Remote Sensing Data Analysis

Boris Dubuisson

MSc by Research in Pattern Analysis and Neural Networks

ASTON UNIVERSITY

September 2005

ASTON UNIVERSITY

Remote Sensing Data Analysis

Boris Dubuisson

MSc by Research in Pattern Analysis and Neural Networks, 2005

Thesis Summary

This thesis documents the study of remote sensing data carried out for xLogos. The allotted task was to separate regions with water present from other parts of the image. Gabor filters were used to create the features. Visualisation, basic classification models and features selection were applied and gave promising results on Digital Globe RGB and Infrared database.

Keywords: Satellite Imagery, Edge detection, Gabor Filter, Feature Selection

Acknowledgements

First, I would like to gratefully thank my supervisor, Prof Ian Nabney, for his general attention and advice all along this project. I would also like to thank my second supervisor, Dr Iris Fermin for her tips about image processing and Dr Dan Cornford for his precious help. I would like to acknowledge Kevan Durrell and Rui Da Silva from the company xLogos for their availability.

I would also like to thank all the NCRG researchers for their constant enthusiasm and the familiar ambiance they keep within the department. Special thanks to Dr Stéphane Bounkong and Borémi Toch, for their numerous stimulating discussions. Many thanks also to Ms Vicky Bond for her efficiency concerning administrative matters.

Obvious thanks to my fellow MSc students for the good time we shared, Etienne Mallard, Mathieu Chatelain, Mathieu Collas, Amy Rostron, Sonia Lester, Ludovic Turlier, Innocent Sizo Duma and the PhD students as well, Rémi Barillec, Thomas Bermudez, Saif Khan, Erik Casagrande and Dharmesh Maniyar.

I would like to gratefully acknowledge all kind of supports of my parents during the whole year, and thank my brothers and my sister as well, to be what they are.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Foreword

This thesis is the report of research project carried out at Aston University in the Neural Computing Research Group from January to September 2005, as part of a Master of Science by Research. The goal of this document is to give account of the work done, as well as to provide a documentation about this project which uses different techniques in image processing and in data classification.

The project was run in collaboration with a Canadian company called xLogos . The goal of the application is to identify regions that are covered by water by analysing remote sensing imagery. [Nabney and Fermin, 2004] reports the initial study made on this problem last year. Promising results were obtained with different classification models using features based on Gabor wavelets. Nevertheless it remained some areas of work to carry out in order to better model accuracy.

The experiments were written in Matlab and largely based on the Netlab toolbox developed by the Neural Computing Research Group at Aston.

1.2 Motivation

Water is surely the most important resource on Earth. All known forms of life depend on water. A considerable part of the ecosystem is based on it and it offers a considerable biodiversity. Therefore it is essential to catalogue and maintain water areas. On the planet, water is continuously moving through the cycle involving evaporation, precipitation, and runoff to the sea. So we need a way to settle this non-trivial issue. Satellite imagery correctly used provides a cost- and time-efficient tool to perform this task.

1.3 Organisation of the work

1.3.1 Architecture and Scenario

The task of identifying and classifying regions of interest in an image can be tackled with several different strategies.

1. Extract regions of interest from the image (i.e. perform an image segmentation) typically using edge detection and other computer vision techniques. Compute region-based features for each region and then classify them.

2. Classify each pixel separately but incorporate information about the surrounding region by using contextual features (that take account of values from a region around the pixel of interest).

3. Classify each pixel separately, but then use an object model to create coherent region classifications. Strictly speaking, this approach should use classifiers based just on pixel values, but in practice contextual features can be used. The best approach is to use probabilistic classifiers (i.e. ones that generate an approximation to the class posterior probability) and use probabilistic models of the regions so that the laws of probability can be used to carry out inference.

The exploratory work of [Nabney and Fermin, 2004] focussed on the second of these approaches, using Gabor filters and run-length encoding as features. But Nabney and Fermin's experiments were based on too few images from RGB DigitalGlobe database. One image was used to train a model and another one to test it. The generalisation performance were subsequently quite poor. This is why this work was extended with larger sets from more image data.

The third approach was also evaluated. The first approach is a computer-vision approach, requiring an initial segmentation of the image, and has been considered briefly in this project.

Thus the three main areas of work that this project addresses are:

- **Improvement in features and feature selection.** Different parameters of the Gabor filter were experimented. The number of features per pixel had to be reduced in order to compute more quickly the results and improve the accuracy model. Indeed it is important that only the features that prove effective are included. We investigated the use of Principal Components Analysis (PCA) as feature extractors and combiners and use Automatic Relevance Determination (ARD) in order to select the optimal feature sets.

- **Improvement of classifiers.** A complete set of experiments to determine the optimal model structure needed to be performed. In addition, more work was

required on data selection to generalise the models. We also investigated unsupervised classification models. Indeed, such models are potentially *more* appropriate for this application because of the large variability of regions not in the class of interest. To train a good supervised model, it is essential to have many examples of both classes (i.e. the class of interest and all other possible regions in the image), which takes a lot of time on the part of the human user, and also requires sampling regions appropriately from many images. Hence a 'novelty detection' approach could give more robust generalisation.

- **Spatial filtering.** Averages over neighbouring patches were experimented.

1.3.2 Thesis outline

Chapter 2 In this chapter, we describe the different databases provided for this project. We focus the research on one of them. A manual data selection is made to create the data sets and we talk about the pre-processing applied on them.

Chapter 3 We present here the extraction of the features that were used for the future models. HSV colour space, Canny edge detection and Gabor filter are introduced.

Chapter 4 In this part, we tackle the supervised classification and evaluate the potential of different models such as the generalised linear regression (GLM), and the multi-layer perceptron (MLP). We also use PCA and ARD to choose the correct values of the parameters of the Gabor filter.

Chapter 5 We evaluate here the potential of an unsupervised classification (Gaussian mixture models) for classification. This is applied in the following way. A single model is trained to approximate the class-conditional probability density of the water class. Then a threshold on its output is used to decide if a new vector is likely to belong to this class.

Chapter 6 In this chapter, the third approach is assessed. We try to incorporate some spatial information in the model to improve its accuracy.

Chapter 7 We end the thesis by summarising the work done and providing some future direction of research.

Chapter 2

Data sets and Pre-processing

2.1 Different databases

Three databases covering small regions were supplied:

Iunctus visible band; this is from the SPOT satellite at 10 m resolution and is panchromatic (the total intensity across the entire visible spectral band).

Figure 2.1: An example of the Iunctus database.

DigitalGlobe ; this is from QuickBird satellite. Launched in October 2001, it acquires colour images (4 bands - **RGB and infrared/IR**) with a resolution of 2.44 m covering a surface area of 16.5 km $\times$ 16.5 km. Our database was composed of 5 full scenes images of the following locations:

- Ottawa;
- Vancouver;
- Boulder, CO;
- Missoula, MT;
- Palm Island, United Arab Emirates.

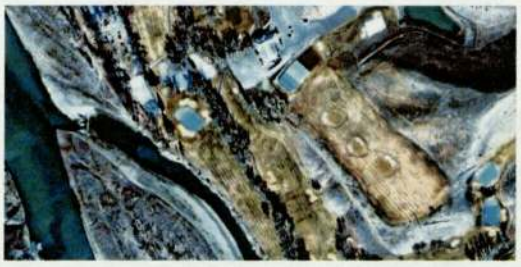


Figure 2.2: RGB channel of an example of the DigitalGlobe database.

Figure 2.3: RGB channel of an example of the DigitalGlobe database.

Known locations of water from manual selection (see Section 2.3) were used to select regions in two classes: water and non-water. The water regions were essentially lakes, rivers and reservoirs. The non-water regions were chosen to cover a range of terrain, including both natural features (such as forest, mountains) and man-made features (such as roads, buildings and cultivated areas).

2.2 Choice of the database

Because of time constraints, all my efforts were focussed on one database. [Nabney and Fermin reports the initial study made on this problem and used visualisation methods to decide which database was the most suitable to this problem among Iunctus, DigitalGlobe RGB and DigitalGlobe IR. The purpose was to see in each database if two classes (water and non-water) are well separable using the provided features. By projecting the input data linearly using PCA (see Section 4.3.1) or non-linearly using Neuroscale [Bishop, 1995] into a two-dimensional space and colouring each class differently, we can see whether a given set of features provides good separation between classes. Normally, if a reasonable class separation is impossible in the feature space, the classes will overlap heavily in the two-dimensional space. From these methods Nabney and Fermin decided that the DigitalGlobe RGB database was most likely to yield good classification models. It was decided subsequently to add the DigitalGlobe IR database as well in my study knowing that actually DigitalGlobe RGB and IR databases come from the single DigitalGlobe database.

2.3 Creation of the data sets

So satellite images from DigitalGlobe were chosen. Each image had 4 channels: the 3 common RGB channels and an infrared (IR) channel. The database that was supplied contained 8 Gb of satellite images from 5 full scenes of different parts of the world. These images were partitioned into strips of one thousand rows. They were originally

stored in the representation with four 11-bit bands (RGB and Infrared). xLogos created a program to partition these images into 1000×500px tiles in order to limit the use of memory required for the experiments and to split the RGB channels and the IR one in two images.

We selected manually 51 satellite images that we found interesting and representative of the database. Indeed data sets of pixels were required for the experiments to train and test classification models. Besides these data had to be labelled to be used for supervised learning and to assess the models. However it takes time to label each image by hand; that is why the whole database was not considered. Only two classes were used: water and non-water. Then for every selected images some areas were labelled such as lakes, rivers, fields, roads, mountains, buildings. Thus a lot of pixels coming from different textures could be extracted easily. And the more we had different types of textures, the more the models would be generalised. The limit in this method was that only the regions that we were sure of the class were labelled. For example it is difficult to label the border of a river or of a lake visually with a good accuracy. So some parts of the images stayed unlabelled and could likely be not represented in the next models.

From this labelled data, three data sets were built: training set to train the model, the validation set to select the architecture of the model, and the test set to determine the generalisation performance. For each one 17 different satellite images were used. Thus taking different parts of regions of the world for each set, we could be sure that these sets were independent. However to give the same weight for each class, each data set had to be balanced (same number of pixels for each class). So finally we obtained:

- A balanced training data set of 96 518 pixels from 17 satellite images;
- A balanced validation data set of 66082 pixels from 17 satellite images;
- A balanced test data set of 117818 pixels from 17 satellite images.

2.4 Histogram equalisation

Before extracting features, some image pre-processing was performed. This consisted of histogram equalisation on each image separately in order to produce a more consistent distribution of pixel intensities and reduce variability of lighting [Petrou, 1999]. We assumed that this processing enhanced the separation between the water and not-water classes, particularly because we used, as you will see farther, gray-level images to extract the textures. Nevertheless there was an edge-effect since each image was processed separately, but we did not notice a decrease of performance. Thus we did not experiment without the histogram equalisation.

Feature Extraction

3.1 HSV colour space

The images from DigitalGlobe are using the classic RGB (Red, Green, Blue) colour space to represent the colour of a pixel. We will talk farther in this thesis about experiments using an alternative colour space called HSV[1] for Hue, Saturation and Value. This defines a colour space composed of three coordinates:

- **The hue** is an angle from 0 to 360 degrees, typically 0 is red, 60 degrees yellow, 120 degrees green, 180 degrees cyan, 240 degrees blue, and 300 degrees magenta.
- **The saturation** typically ranges from 0 to 1 and defines how grey the colour is, 0 means grey and 1 is the pure primary colour.
- **The value** corresponds to the brightness of the colour.

The HSV colour space is often represented in the most of the softwares as a wheel (See Figure 3.1).

Transformation from RGB to HSV

A series of formulas make the transformation from RGB colour space to HSV colour space easy. Let us define a colour defined by its coordinates in RGB space (R, G, B) and in HSV space by (H, S, V). Let Max be the maximum of the (R, G, B) values, and Min be the minimum of those values. The formulas can then be written as:

$$H = \begin{cases} \left(0 + \frac{G-B}{Max-Min}\right) \times 60, \text{ if } R = Max \\ \left(2 + \frac{B-R}{Max-Min}\right) \times 60, \text{ if } G = Max \\ \left(4 + \frac{R-G}{Max-Min}\right) \times 60, \text{ if } B = Max \end{cases}$$

[1]NB: The HSV model was created in 1978 by Alvy Ray Smith, future co-founder of *Pixar*.

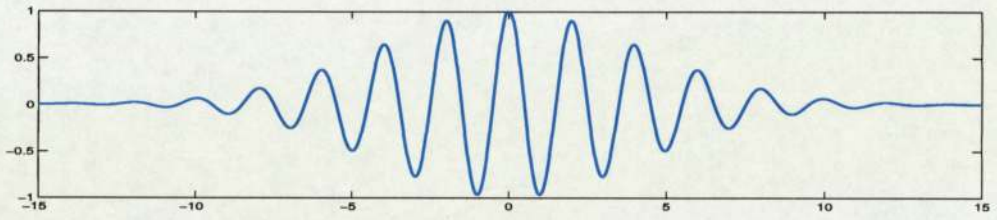


Figure 3.3: An example of a Gabor wavelet function in 1 dimension.

So, when a function is convolved with a Gabor wavelet, the frequency information near the centre of the Gaussian is captured, while frequency far away from the centre of the Gaussian has a negligible effect.

$$\mathcal{W}(x, y, \theta, \lambda, \phi, \sigma, \gamma) = \exp\left(-\frac{\tilde{x}^2 + \gamma^2\tilde{y}^2}{2\sigma^2}\right)\cos\left(2\pi\frac{\tilde{x}}{\lambda} + \phi\right) \tag{3.2}$$

where $\tilde{x} = x\cos\theta + y\sin\theta$ and $\tilde{y} = -x\sin\theta + y\cos\theta$. The parameters have the following interpretations:

- θ specifies the wavelet orientation in range $[0, \pi]$;
- λ specifies the wavelength of sine wave;
- ϕ specifies the phase of the sine wave in range $[0, \pi]$;
- σ specifies the radius of the Gaussian;
- γ specifies the aspect ratio of the Gaussian (ratio of major and minor axes).

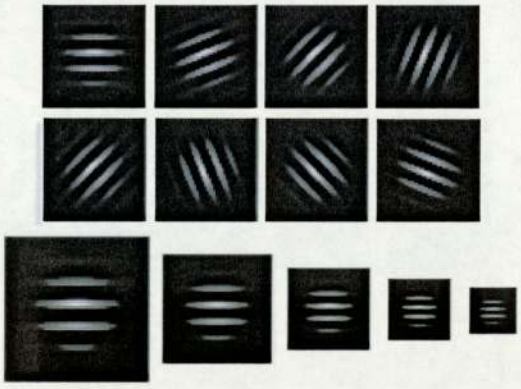


Figure 3.4: Different orientations and scales of Gabor wavelets were used to analyse the textures of the images.

In fact we can add a sixth parameter which is the size of the Gabor mask. However it is completely linked with the radius of the Gaussian σ. Indeed we can see on the Figure 3.4 that the coefficients too far away from the centre of the mask are drawn in

Actual	Predicted		
	non-water	water	Total
non-water	46411	12498	58909
water	8659	50250	58909
Total	55070	62748	117818

Table 4.5: Confusion matrix for the classification of the test set using the MLP with 150 hidden nodes and 160 Gabor features. Accuracy of 82.0%

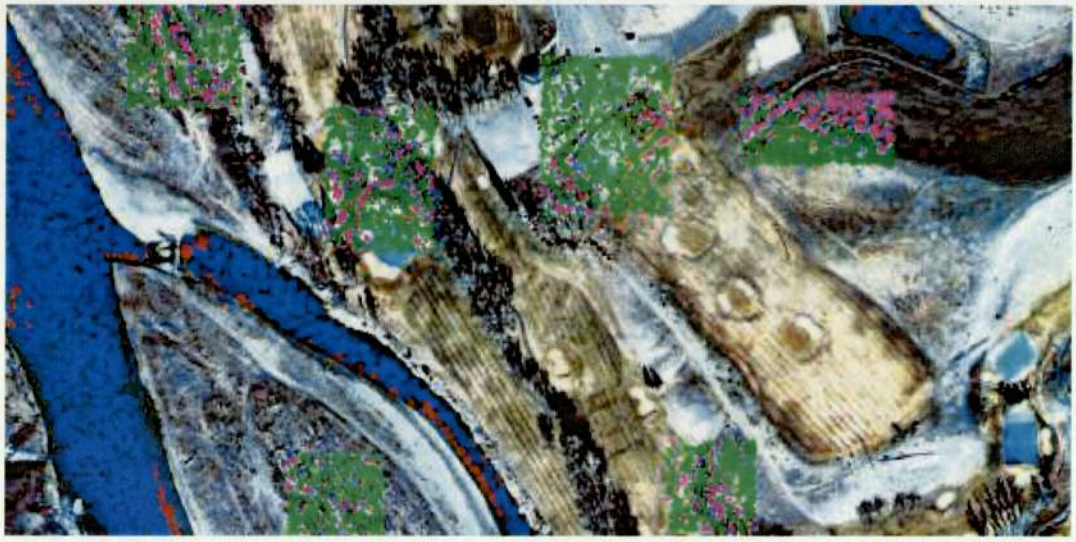


Figure 4.4: The results with the MLP using the 160 Gabor features on an image of the test set. The result image is coloured as follows: green pixels for True-Positive non-water, red pixels for False-Positive non-water, blue pixels for True-Positive water, and magenta pixels for False-Positive water.

4.3 Feature selection

To perform it we investigated the use of PCA as extractors and combiners, and ARD to select the optimal feature sets.

4.3.1 Principal Components Analysis

Principal Components Analysis (PCA) is a technique that can be used for extracting structure from high-dimensional data sets. Basically, PCA is a linear transformation that chooses a new coordinate system for the data set such that the greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis, and so on. So PCA can be used for reducing dimensionality in a data set. Indeed, in general, a reduction in the dimensionality of the data set will be accompanied by a loss of some information. But in the case of the PCA, by projecting the data set onto the first

principal components, the characteristics of the data set that contribute the most to its variance will be retained. The principal components are found by diagonalising the covariance matrix of the data set [Bishop, 1995]. However there is no general technique for deciding how many principal components should be used to represent the data adequately. Commonly, we list the principal components in descending order of eigenvalues to analyse the participation of each component in the variance of the data set.

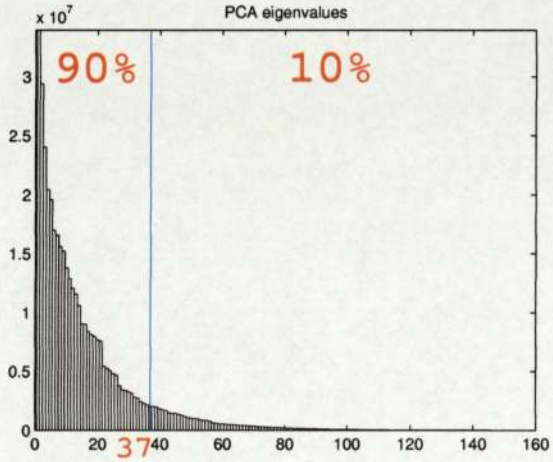


Figure 4.5: Principal components analysis on the 160 Gabor features (5 scales). 37 principal components are kept corresponding to 90% of the variance of the training set.

Results

PCA was applied on the data training set. Figure 4.5 depicts the plots of the principal eigenvalues and shows no significant structure [1]. So to determine the number of the most significant components we chose to keep empirically 90% of the variance of the data. Thus we calculated the limit where 90% of the sum of the eigenvalues were represented. Thereby 37 principal components were kept. Nevertheless instead of computing these features as a linear combination of the Gabor features, it is possible to compute them directly by applying a linear combination of the Gabor masks and so reduce the time of computation. Afterwards different models using these 37 inputs were trained. The performance obtained was a bit lower than previously with the models using all the features (see Table 4.6 and Table 4.7). However it was promising since we reduce considerably the dimension of the data and even so an accuracy of around 70% was obtained with GLM.

[1] We can see a fast decrease after the 22th feature, but we did not consider this 'elbow' because the eigenvalues after this limit is still non negligible.

Model	No hidden nodes	Training set	Validation set
GLM	-	**76.8**	**70.7**
MLP	60	72.6	67.9
MLP	80	75.0	70.2
MLP	100	73.3	69.3
MLP	120	74.5	70.2
MLP	150	74.7	69.7
MLP	180	75.6	70.4
MLP	200	75.8	70.6

Table 4.6: Classification accuracy on the training and validation sets with the 37 Gabor features computed using PCA. GLM was considered here as the most appropriate model.

Actual	Predicted		
	non-water	water	Total
non-water	43808	15101	58909
water	20679	38230	58909
Total	64487	53331	117818

Table 4.7: Confusion matrix for the classification of the test set using a GLM and 37 Gabor features computed using PCA. Accuracy of 69.6%.

4.3.2 Automatic Relevance Determination

Selecting the best input variables is a non-trivial problem.

The goal of the Automatic Relevance Determination (ARD) is the detection of the relevant component of the input vector: this can be achieved by associating one hyper-parameter to the group of weights which connects one input unit to all of the units in the next layer [MacKay, 1995]. These hyperparameters α_i control the size of the groups of weights through a prior distribution. The prior is a Gaussian function with 0 mean and standard deviation $\sigma_i = \frac{1}{\sqrt{\alpha_i}}$. From the values of the hyperparameters it is possible to figure out which inputs are more relevant than others. A large hyperparameter value means that the weights are constrained near zero, and hence the corresponding input is less important [Nabney, 2002]. This naturally prunes irrelevant features in the data.

In the case under study, the set of hyperparameters is composed by 180 elements, one for each input unit (α_i, $i = 1..180$) and one for the bias unit (α_0). For the MLP used in this work, each one of the α_i is controlling 20 weights connecting one input to the 20 hidden units.

During the learning, the hyperparameters are modified using the evidence procedure; we find their optimal value, subject to some simplifying assumptions about the network function to make the analysis tractable.

Results

ARD was applied on the data training set. Figure 4.6 shows the plot of the hyperparameters α. We can so conclude according to Figure 4.3 to know the properties of the features that the bigger the scale of a feature is, the more irrelevant the feature will be.

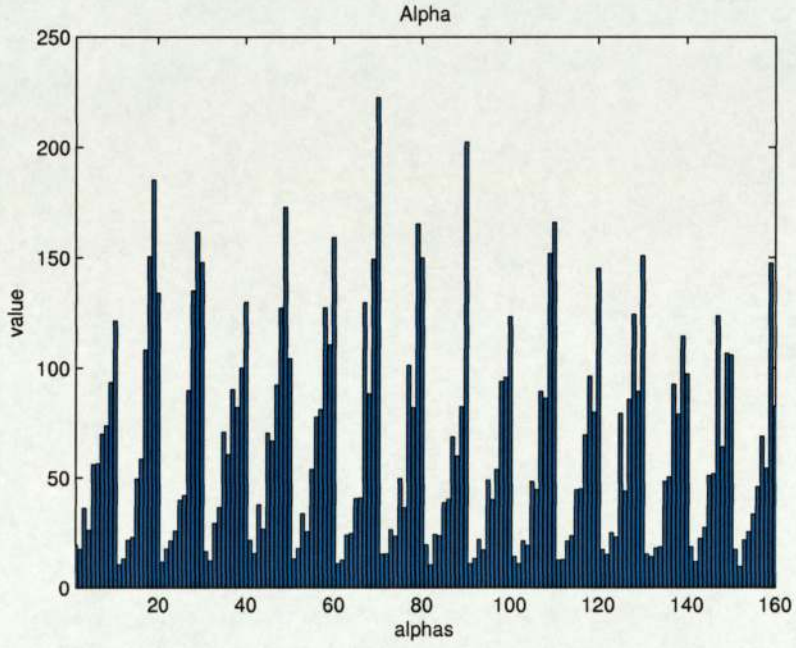


Figure 4.6: Automatic Relevance Determination on the 160 Gabor features (5 scales, 8 orientations, 2 phases, 2 layers). The bigger the value of alpha, the less relevant the corresponding feature is.

Besides regarding the 32 most relevant inputs we can say that we have approximately as many RGB Gabor features as IR Gabor features (respectively 14 and 18 features). Each direction θ is well-represented (around 4 features for each). This seems actually reasoning because we can assume that the analysed textures are randomly oriented and that their variance are not based on only one direction. The influence of the phase is also real because the both phases are represented for a same orientation and a same frequency/scale (around 14 features for $\phi = 0$ and 18 features for $\phi = \pi/2$). Nevertheless for the wavelength λ some values seem to play a bigger role than some others especially the smallest value. Indeed we have 29 features with $\lambda = 4$ and a size of the Gabor mask = 11 and 3 features with $lambda = 2\sqrt{2}$ and size of the Gabor mask = 13. The rest of the scales are not represented among these 32 inputs. This confirms that the smallest scale makes the variance of the data set increase. As a result we decided to only keep the features with the size of the Gabor mask equal to 11 ($lambda = 2\sqrt{2}$).

Then different models were experimented using these 32 Gabor features actually

based on a one-scale Gabor filter. The obtained results were better than with the previous models using 5 scales (see Table 4.8 and Table 4.9) especially with the MLP. So the ARD method seems to have been efficient even if the number of hidden nodes required for the MLP is relatively large.

Model	No hidden nodes	Training set	Validation set
GLM	-	75.7	72.4
MLP	20	79.5	79.5
MLP	60	80.1	81.8
MLP	80	80.5	81.5
MLP	100	81.1	82.0
MLP	**120**	**81.5**	**82.4**
MLP	150	82.0	82.5
MLP	180	81.6	82.6
MLP	200	82.3	83.1

Table 4.8: Classification accuracy on the training and validation sets with 32 Gabor features (1 scale). The MLP with 120 hidden units is chosen, because the performance is stable around this value.

Actual	Predicted		
	non-water	water	Total
non-water	45968	12941	58909
water	7956	50953	58909
Total	53924	63894	117818

Table 4.9: Confusion matrix for the classification of the test set using a MLP with 120 hidden nodes and 32 Gabor features (1 scale). Accuracy of 82.3%.

However the analysis of Figure 4.6 is not totally finished. Indeed if we concluded that the features of the smallest scale were the most relevant, i.e. for a size of the Gabor mask equal to 11 (resp. $\lambda = 4$), we have also to try value lower than 11 (resp. 4). This is why we created a different Gabor filter changing the values of the scales to fit better the data. Table 4.10 defined these new values.

New frequencies	2	$2\sqrt{2}$	4	$4\sqrt{2}$
New sizes of the masks	5×5	7×7	11×11	13×13

Table 4.10: New parameters used for the Gabor filter. Only the values of the scale are modified. The values for the orientations and the phases are kept.

Then ARD is applied on the data training set newly processed from the modified Gabor filter. The hyperparameters α were plotted, but, this time, it is not as readable

as previously. So another method was used to visualise the results. First the features were listed in ascending order of α (and so in descending order of relevance). Then to emphasize the importance of a scale, we can count the number of features using this scale among the first M relevant features. We make this number M vary from 1 to 128 (the number total of features) and repeat this process for each scale. Finally Figure 4.7 depicts the importance of each scale among the first relevant features.

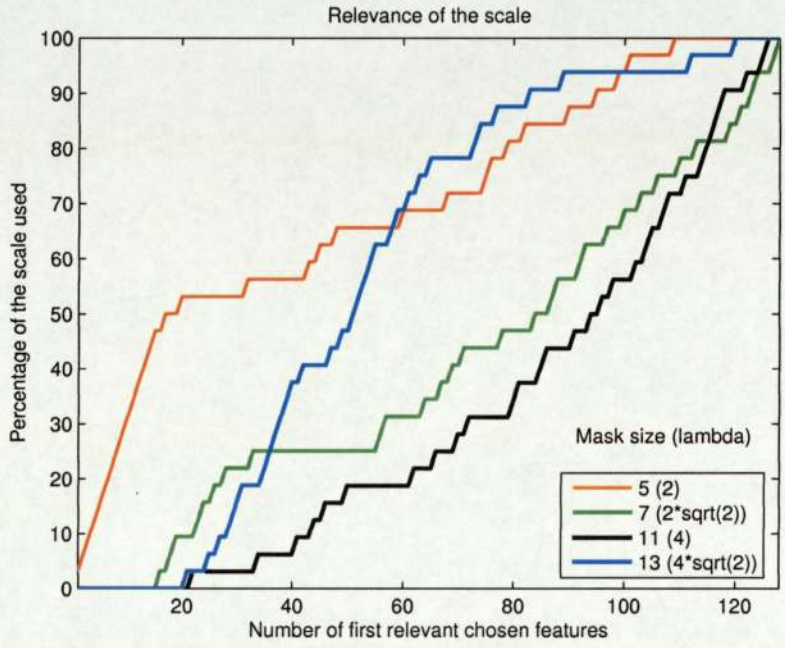


Figure 4.7: Importance of each scale among the first relevant features. The Y-coordinate describes the percentage of features using the scale among the X-coordinate relevant features.

We can notice clearly that the smallest ($\lambda = 2$) and the biggest mask ($\lambda = 4\sqrt{2}$) are the most appropriate to classify the data. The other scales seem to have the same importance, with a small preference for $lambda = 2\sqrt{2}$. As a result we decided to keep 3 scales with $\lambda = 2, 2\sqrt{2}, 4\sqrt{2}$ and the corresponding sizes of the mask of 5×5, 7×7, 13×13. Finally we had 96 features. Different models were trained again and better results were obtained with an accuracy around 85% (see Table 4.11 and Table 4.12). So it seems that the selected scales suit better to our data set.

4.4 Combining Gabor features with colour components

An accuracy around 85% has been reached with the model using the 96 Gabor features and the one using the 4 colour components. But one question can be raised: what

Model	No hidden nodes	Training set	Validation set
GLM	-	77.6	74.2
MLP	100	84.3	84.1
MLP	120	84.2	83.9
MLP	**150**	**85.0**	**84.5**
MLP	180	85.4	84.6
MLP	200	84.9	84.4

Table 4.11: Classification accuracy on the training and validation sets with 96 Gabor features (3 scales). The MLP with 150 hidden units is chosen, because the performance is stable around this value.

Actual	Predicted		
	non-water	water	Total
non-water	47546	11333	58909
water	5542	53367	58909
Total	53118	64700	117818

Table 4.12: Confusion matrix for the classification of the test set using a MLP with 150 hidden nodes and 96 Gabor features (3 scales). Accuracy of 85.6%.

would be the performance of a model using all these features? Thus the purpose of this section is to combine the 96 Gabor features from the three-scale Gabor filter with the 4 colour components (RGB and infrared).

4.4.1 Mixing the features

So a set of GLM and MLP models were created with these 100 features. Table 4.13 shows reasonable results for the GLM since the accuracy was increased by around 3% compared to the GLM using only the colours. 88.5% of the test data were correctly classified. Moreover as we can see on Figure 4.10(a), the ROC curve is more convex than the one of GLM using only the colour components. This means that the classification is less random. So the classification through GLM has been consolidated.

On the other hand the MLP does not give satisfying results because it is even less than the MLPs using only the colours. However we can notice as well on Figure 4.10(b) that the ROC curve of the MLP using only the colour components is not convex and so its classification can be random. But it is not really surprising.

Nevertheless it is hard to evaluate these results from the accuracy since the difference of percentage is low. This is why the ROC curve is more adequate to compare the performance between the GLM and the MLP. Indeed we can see quickly on Figure 4.9 that the GLM give a best classification than the MLP.

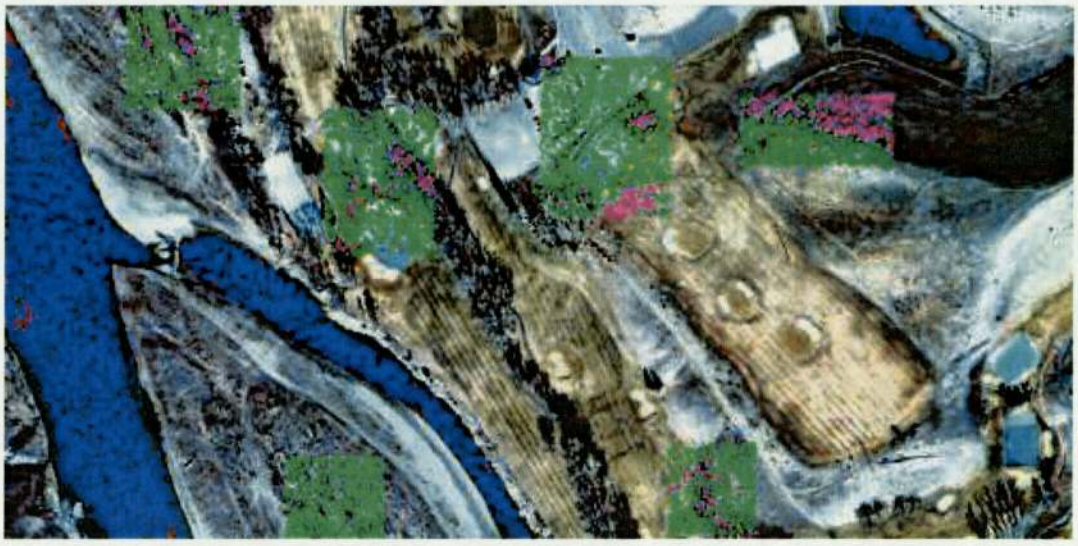


Figure 4.8: Results with the MLP using 96 Gabor features (3 scales) on an image of the test set. The result image is coloured as follows: green pixels for True-Positive non-water, red pixels for False-Positive non-water, blue pixels for True-Positive water, and magenta pixels for False-Positive water.

4.4.2 Committee

We experimented the use of a *committee* from these both models. It simply consists in a weighted sum of their outputs. A committee with M models can be written in the form:

$$P(\mathrm{x}|C_i) = \sum_{i=1}^{M} \alpha_i P_i(\mathrm{x}|C_i) \tag{4.5}$$

where α_i are the weights of the models (which must be positive and sum to one), $P_i(\mathrm{x}|C_i)$ the probability of the input vector x to belong to the class C_i through the model i.

Model	No hidden nodes	Training set	Validation set
GLM	-	87.8	88.2
MLP	50	86.5	85.5
MLP	60	85.5	84.7
MLP	80	86.2	85.2
MLP	100	86.3	85.4
MLP	120	86.7	85.8
MLP	150	87.9	86.2
MLP	180	87.0	85.5
MLP	200	88.0	86.2

Table 4.13: Classification accuracy on the training and validation sets with 96 Gabor features (3 scales) and the 4 colour components (RGB and infrared).

As only two models were integrated in our committee, two weights had to be evaluated. They were determined from the validation set. We made them vary until the optimal accuracy of our set was reached. Besides we found a weight of 0.37 for the MLP with the Gabor features and so 0.63 for the MLP with the colour components. Table 4.14 presents the confusion matrix of the test set. According to Figure 4.9, the committee have similar results as GLM.

Actual	Predicted		
	non-water	water	Total
non-water	52823	6086	58909
water	7899	51010	58909
Total	60722	57096	117818

Table 4.14: Confusion matrix for the classification of the test set using committee of a MLP with 150 hidden nodes and the 96 Gabor features (3 scales) and a MLP with 10 hidden nodes using the 4 colour components (RGB and infrared). Accuracy of 88.1%.

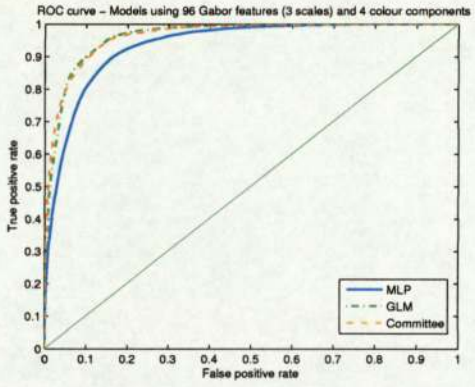
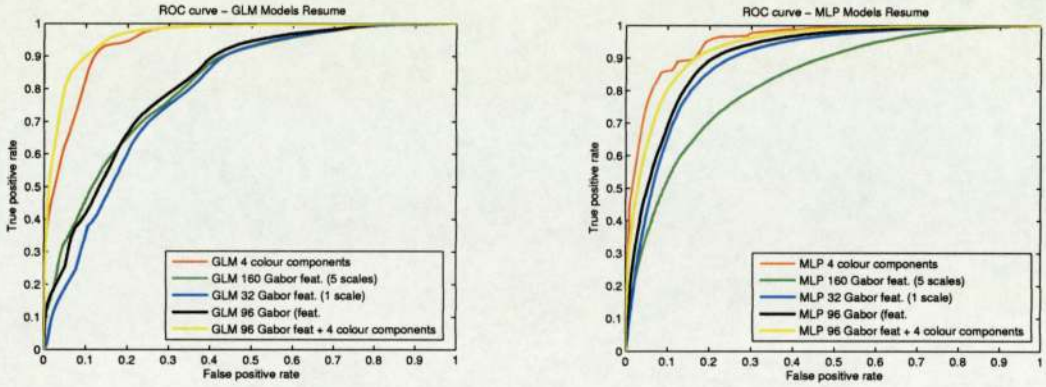


Figure 4.9: ROC curves of the models using the Gabor features and the colour components. The MLP has 150 hidden units.

4.5 Conclusion

Thanks to ARD the parameters of the Gabor filter have been improved to suit better to the textures of the DigitalGlobe images. Concerning supervised classification, simple models such as the GLM are actually not really convincing when they only use the Gabor features. However they perform quite well with all the features. More sophisticated techniques such as Bayesian learning for the MLP are more efficient to use only the Gabor features than GLM. But the use of the combination colour features and Gabor features does not promote the MLP. Nevertheless with the committee the results are as efficient as the one with the GLM, though it is more complex.



(a) ROC curves of GLMs.

(b) ROC curves of the MLPs.

Figure 4.10: ROC curves of the models using the different set of features.

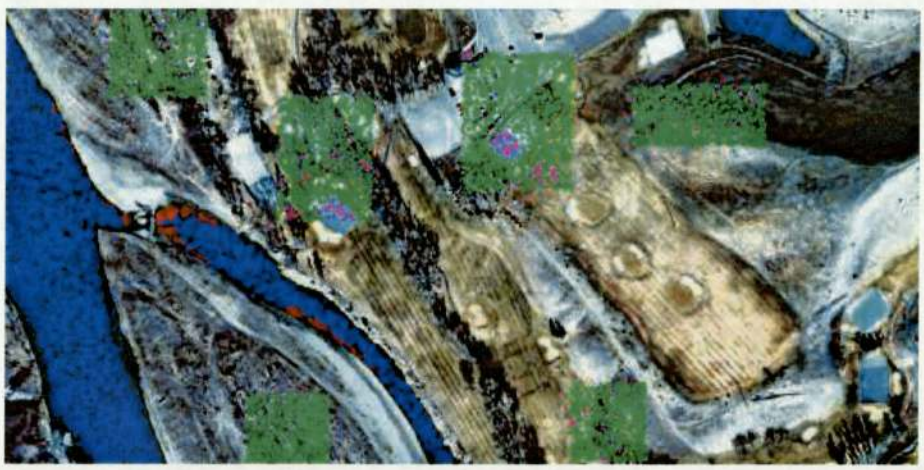


Figure 4.11: Results with the committee on an image of the test set. The result image is coloured as follows: green pixels for True-Positive non-water, red pixels for False-Positive non-water, blue pixels for True-Positive water, and magenta pixels for False-Positive water.

Chapter 5

Unsupervised Classification

Unsupervised classification is investigated in this chapter with the use of Gaussian Mixture Models. In this approach a model is trained to fit the water-classed data. As the non-water class owns a widest variety of textures than the water class, it seems reasonable to focus our study on the water class and establish its density distribution rather than on the non-water class. As a result this approach is likely to be of interest for our problem because of the wide variability of regions. After setting up a *probability mapping* of the water class, a threshold on its output is used to decode if a new input vector is likely to belong to this class. This can be viewed as a 'novelty detection' approach [Bishop, 1994]: anything that looks new does not belong to the class of interest.

5.1 Density modelling

Mixture Model is a common model for clustering and density modelling. It consists in a linear combination of different components. Here, we consider models in which the density function is formed from a finite linear combination of basis functions. A model with M components can be written in the form:

$$p(x) = \sum_{j=1}^{M} p(\mathrm{x}|j)P(j) \tag{5.1}$$

where the M functions $p(\mathrm{x}|j)$ are the components density functions, with

$$\int_{-\inf}^{+\inf} p(\mathrm{x}|j)\mathrm{dx} = 1. \tag{5.2}$$

The probabilities $P(j)$ are the mixing coefficients, which must be positive and sum to one.

5.1.1 Gaussian Mixture Model

Because of their probabilistic nature, Gaussian mixtures are in general preferred over models. The density components are now Gaussian distributions:

$$p(\mathrm{x}|j) \sim \mathcal{N}(\mu_j, \Sigma_j) \tag{5.3}$$

The finite mixture model will then be expressed by extending Equation 5.1 to:

$$p(x|\mu, \Sigma, \alpha) = \sum_{j=1}^{M} \alpha_j \mathcal{N}(\mu_j, \Sigma_j) \tag{5.4}$$

where $\alpha = \{\alpha_1, ..., \alpha_M\}$ are the mixing weights (which must be positive and sum to one), $\mathcal{N}(\mu_j, \Sigma_j)$ are the Gaussian density functions, $\mu = \{\mu_1, ..., \mu_M\}$ are the means and $\Sigma = \{\Sigma_1, ..., \Sigma_M\}$ are the covariance matrices.

5.1.2 Expectation-Maximisation Algorithm

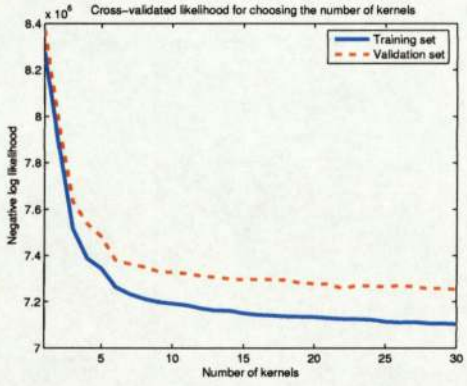
Assuming that a data set is generated by a certain Gaussian mixture, the task is to fit a model to these data, and thus to estimate the parameters of the generating mixture. The most popular algorithm for training a Gaussian mixture is the Expectation-Maximisation (EM) algorithm. The EM algorithm iteratively modifies the Gaussian Mixture Model (GMM) parameters, the means μ_j, the covariance matrices Σ_j, and the mixing coefficients α_j for each components j, to maximise the likelihood of the data [Bishop, 1995].

5.2 Determination of the number of components

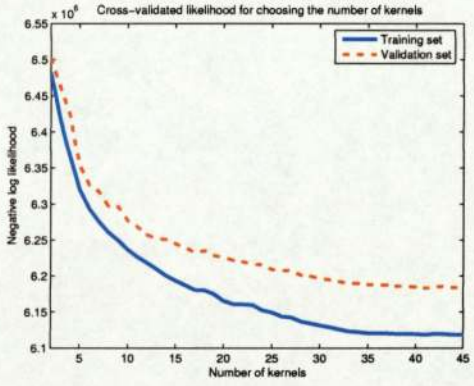
Finding the "right" number of components for a data set is a difficult task. The likelihood of the data will reach a maximum when the number of mixtures will be equal to the number of training data, and in this case the model will overfit. This number of components should reflect the number of *populations* in which a pixel can be labelled. In our case it should express the number of main textures of the data set in which a pixel can be classified. There exist several solutions to determine this number of components [Smyth, 1996, Salvador and Chan, 2004]. [Smyth, 1998] proposes a efficient and simple solution based on the cross-validation to select a model from a family of candidate models. The cross-validated likelihood is used for choosing the number of mixture components.

Cross validation

So we decided to use the Cross-Validation method, which selects the number of components by maximising the average likelihood over the training and validation data sets simultaneously. A first experiment was launched using the 32 Gabor features processed from the one-scale Gabor filter. Only the pixels labelled as water were kept for the learning. Indeed the purpose here is to create a model, which fits the water-class data distribution. The water-class was preferred since it has certainly less textures than the non-water class. Figure 5.1(a) depicts the plot of the negative log-likelihood in function of the number of kernels. Actually the curve should go up again when the model starts to overfit and we should be able to see a minimum for the validation set curve, which would correspond to the appropriate value. But as the training sets are relatively large for the training (around 48000 and 33000 pixels simultaneously), it is hard to make a model overfit with so few kernels. Nevertheless we can see that it is quite stable around 15 kernels. We assumed that the variety of textures of the water is reduced and can be modelised with these 15 kernels. Another experiments was made using the 96 Gabor features of the three-scale Gabor filter and in this case, 35 kernels was likely adequate for the data (see Figure 5.1(b)).



(a) GMM using 32 Gabor features from the one-scale Gabor filter. The model with 15 components seems to be a good trade-off between efficiency and complexity.



(b) GMM using 96 Gabor features from the three-scale Gabor filter. The model with 35 components was chosen.

Figure 5.1: Determination of the number of components for the GMM of the water class using the cross-validation method. These plots represent the negative log-likelihood of the data in function of the number of components used.

5.3 Determination of the novelty threshold

Once the probability density functions was modelled, the next step was to fix a threshold on the output of the model to decide if a pixel is likely to belong to the water class (see Figure 5.2).

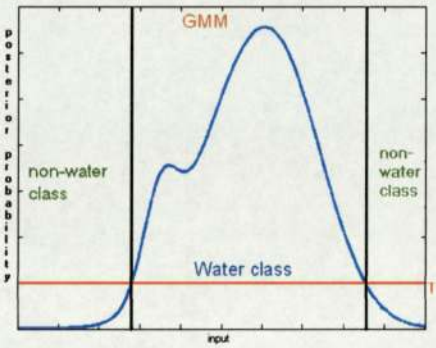


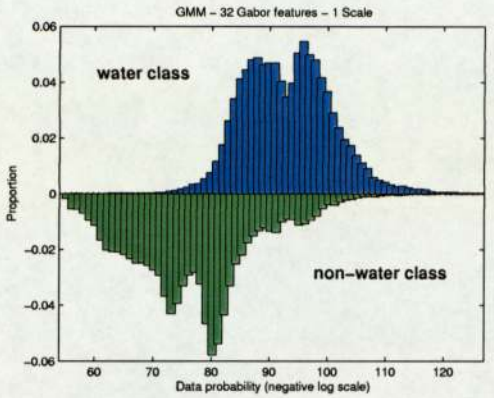
Figure 5.2: Determination of the threshold to separate the classes (1-D example).

To decide the value of this parameter, the output of the whole validation set was analysed. The histogram in Figure 5.3 bins the outputs into equally spaced containers to visualise the output distribution. Thereby we can notice clearly in both case that the distributions of the classes overlap each other. However the features of the one-scale Gabor filter seem to be the more appropriate for the GMM since the overlap of the distributions is even so lesser. On the other hand the GMM using the three-scales Gabor filter must give a doubtful classification. So it seems reasonable to prefer to trust the GMM using the one-scale Gabor filter rather than the other one. Then we had to determine a threshold to decide the class of a pixel from the output. The dichotomy method was applied to calculate the threshold which optimise the accuracy of the validation set. A lower and a upper thresholds were defined, and then the interval between them was divided by 2 keeping the lower and upper thresholds giving the best accuracy. This step was restarted until the interval between the 2 thresholds is close to zero. Afterwards this threshold was applied on the output of the test set.

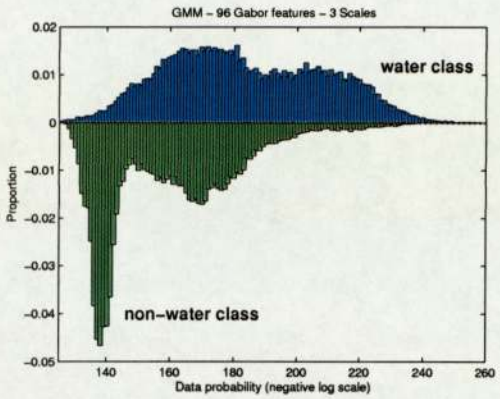
5.4 Results

Promising results were obtained with an accuracy around 83.0% with the test set with GMM using the one-scale Gabor filter (see Table 5.1). However the plot of the ROC curve makes the performance of the model easier to analyse (see Figure 5.4). Indeed we can clearly notice the poor performance of the model using the 96 Gabor features compared to the one using 32 Gabor features.

Figure 5.5 depicts the ROC curves of the MLP using the features from the three-scales Gabor filter and the GMM using the features from the one-scale Gabor filter.



(a) GMM using one-scale Gabor filter. The separation between the two classes is acceptable so far.



(b) GMM using three-scale Gabor filter. The overlap between the two classes is too important to hope a accurate classification.

Figure 5.3: Output distributions of the validation set through the models. On the top the histogram represents the output distribution of the water class, and on the bottom the one of the non-water class. Y-axis represents the proportion of the pixels which are contained in the bin, and which have a posterior probability in the interval of the bin.

We can notice that the water pixels are better classified with the MLP than with the GMM, while the non-water pixels are better identified with the GMM.

Actual	Predicted		
	non-water	water	Total
non-water	46378	12531	58909
water	7558	21351	58909
Total	53936	63882	117818

Table 5.1: Confusion matrix for the classification of the test set using a Gaussian Mixture Model with 15 kernels and 32 Gabor features (1 scale). Accuracy of 83.0%.

5.5 Conclusion

The results obtained with the Gaussian Mixture Model are finally disappointing. GMM was expected to be more appropriate for our problem since we have a wide variety of regions. However the poor performance of the GMM using the 96 Gabor features is surprising in the light of the results with the MLP. Indeed it seemed that the three-scales Gabor filter could give a good separation of the classes. So it remains some work to carry out to improve this unsupervised approach.

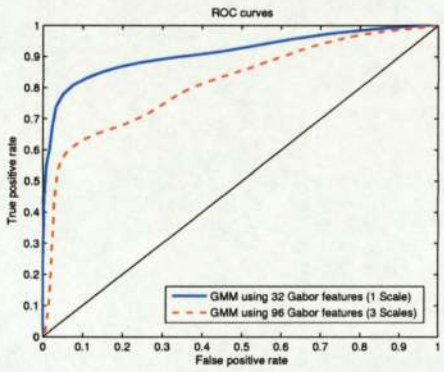


Figure 5.4: Comparison between the GMM using the one-scale Gabor filter and the one using the three-scale Gabor filter.

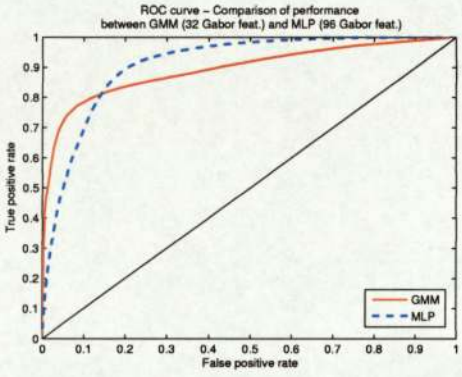


Figure 5.5: Comparison between the performance of the MLP using the three-scale Gabor filter and GMM using the one-scale Gabor filter.

Chapter 6

Spatial filtering

This chapter presents the concise study made to improve the general accuracy of the models. The purpose was to provide some spatial context for a pixel-level classifier. The processing was applied here on the whole image to make the visualisation of the performance of the models easier.

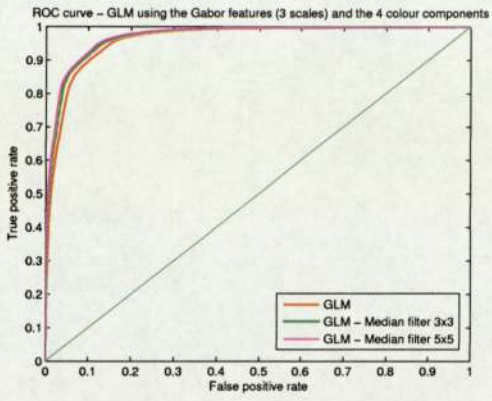
6.1 Median filtering

The idea was to experiment the use of median filtering to better the classification. The median filter is normally used to reduce noise in an image somewhat like the mean filter. However it often does a better job than the mean filter of preserving useful detail in the image. Some pixels can be classify, for example, as water though they are only surrounded by pixels classified as non-water. Median filtering consists in the application of the median over a neighbouring patch to determine the class of a pixel. Therefore this technique homogenise the classification by decreasing the number of isolated pixels in term of class. We can compare it to a blur processing on the output of the analysed image.

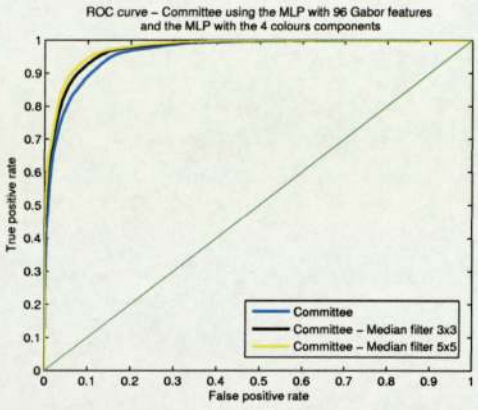
Two size of filters were used: 3×3 and 5×5. Figure 6.1 shows a small amelioration of the performance for the GLM and the committee. The committee seems to give lightly a best response with an accuracy around 91% for the test set with the optimal threshold and the biggest median filter.

6.2 Conclusion

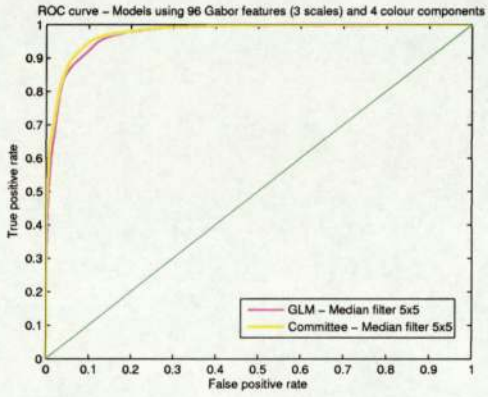
As the processing was applied on the whole image, we realised that there were still some work to carry out. Indeed some areas were completely misclassified (See Figure A.1(g)). The median filtering improved airily the classification for certain image (See Figure 6.2). But if a full area is misclassified and only one well-classified pixel among



(a) Median filtering on the GLM.

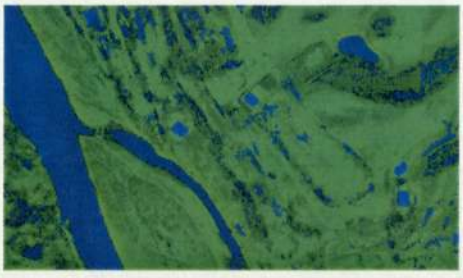


(b) Median filtering on the committee.

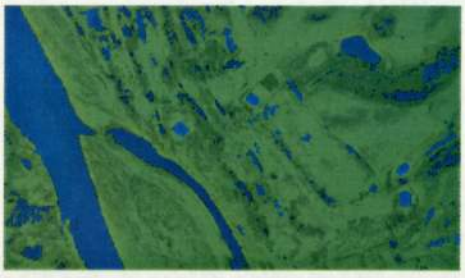


(c) Comparison between the two models after the post-processing.

Figure 6.1: Median filtering applied on the best models.



(a) Application of the committee on an whole image of the test set.



(b) Median filtering on the output of the committee on an whole image of the test set. The size of the filter used here was 5×5.

Figure 6.2: Light enhancement of the performance due to the median filter.

this area, the result will be the opposite of what we want after the median filtering. This pixel will be likely misclassified because all its neighbours are originally misclassified. So median filtering is an interesting method for our problem, but do not settle the large misclassified areas.

Chapter 7

Conclusion

7.1 Summary of the work done

This thesis has given account of the work produced by conducting research on the problem of remote sensing data analysis. Below are the main points that have been covered in this thesis.

After choosing the most appropriate database for our problem among Iunctus and DigitalGlobe, we picked out different images, which looked representative, and built three data sets by classifying different parts of these images.

Then we tried a first approach, which consisted in the extraction of regions of interest from the image. We used different techniques of edge detection such as Canny edge detection. We did not obtain an *image segmentation* that could be exploitable. So we decided to focus directly on the second approach.

We started with an experiment with GLM and MLP models using only the colour components (RGB and Infrared). We found that these features gave surprisingly a good separation of the two classes. Then we experimented the HSV colour space to finally conclude that the RGB colour space was more appropriate for our task.

Nevertheless the model could not rely on only the colour to detect the regions, which are covered by water. This is why we investigated the use of the Gabor filter in order to incorporate information about the surrounding region of the pixel of interest. However it was not easy to carry out since the parameters of the Gabor filter were not accurately fixed, especially the scale. Then, we tried to classify, via different models such as GLM and MLP, the pixels from a five-scale Gabor filter and its 160 corresponding Gabor features. Promising results were obtained and convinced us to carry on in this way.

PCA was therefore applied on the 160 Gabor features to reduce the dimensionality of the input space, while the most of the variance of the data was maintained. The results were correct if we consider the fact that we reduced the number of features from 160 to 37. Yet the accuracy was lower than the one with the five-scale Gabor filter. So

we used ARD to select the optimal features. After several tries, we finally found that three smaller scales were adequate to discriminate the different textures. The results were satisfying for the MLP unlike the GLM.

The next step was to combine the 96 Gabor features with the 4 colour components into a single model. We first mixed the features and used them all as input. GLM gave a unexpected good response, while the MLP did not outperform the accuracy of a model using only the colour features. Then we built a committee from the MLP using the 96 Gabor features and the MLP using the 4 colour components. The performance was similar as the previous GLM. So it seems that GLM are here more adequate because of its simple structure compared to the committee.

Afterwards, we investigated the use of the unsupervised classification with the use of GMM. It was expected to be more appropriate for our problem since we have a wide diversity of regions. But the results were disappointing with a poor performance of the GMM using the 96 Gabor features.

We finally finished our experiment by using the median filtering in order to *blur* the output of a pixel-level classifier by incorporating some spatial context. The results were lightly improved. And the committee seems to be, at last, a bit more efficient than the GLM.

Nevertheless, we realised with the classification of whole images that there are still some work to carry out, because some areas were completely misclassified.

7.2 Further work

Improvement of edge detection The extraction of segments of interest from the image can be improved. Some promising results have been done to perform a useful image segmentation with closed areas [S Wang, 2003, Krupnik and Elder, 2002].

Improvement of unsupervised classifiers We did not investigate the potential of the Gaussian mixture model using a probability density model for each class. Once they are developed, we can combine them with Bayes' theorem to compute the *posterior* probability of each class [Bishop, 1995]. This approach also has the advantage that it can detect if an object does not belong to any class seen in training.

Spatial Inference The incorporation of spatial information, such as typical shape of rivers and lakes into the inference of regions based on pixel-level classifiers can improve the performance of the models.

Adaptive detection The techniques that we used can actually be applied to anything in a satellite picture, not only to classify water regions. Indeed if we train a model

to detect roads, for example, the approach can be similar.

7.3 Afterword

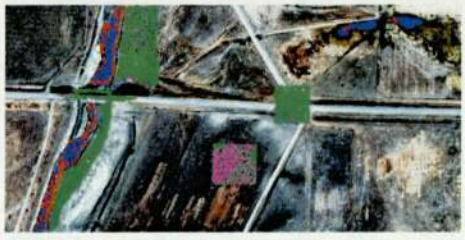
This project has been interesting in many aspects, from the involvement of abstract theory it involves to its practical detection application. It has given us the opportunity to use different areas of expertise such as applied mathematics, statistics, computer science and image processing to achieve the presented results. It certainly is a little frustrating not to have had enough time to take the project to a complete operational state, which was our initial objective, but this is one of the aspects of a research work. It is an interesting piece of work, and to conclude this thesis, let us just express our hope that further research will be conducted to improve the machine learning approach concerning concerning remote sensing data analysis.

Appendix A

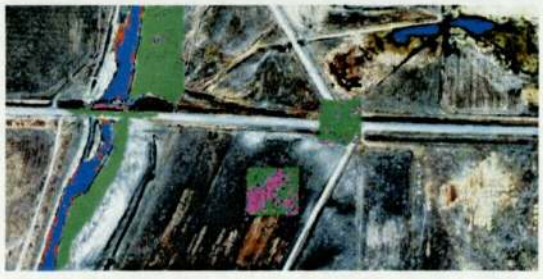
Additional results

(a) RGB image.

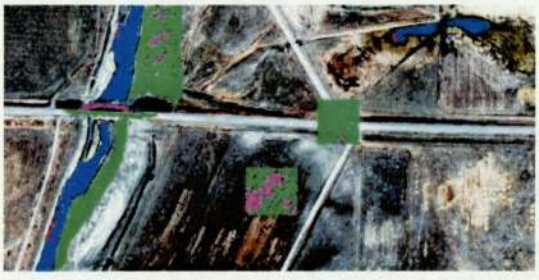
(b) IR image.



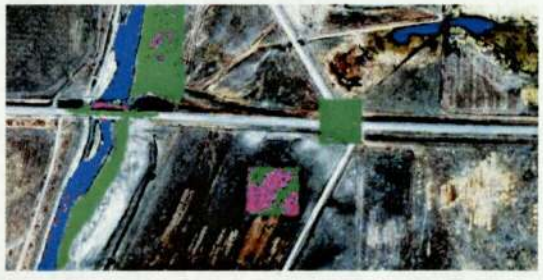
(c) Classification with the MLP using the five-scales Gabor filter.



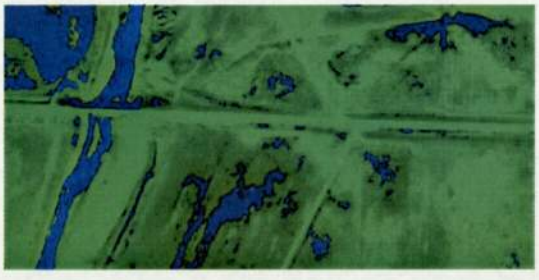
(d) Classification with the MLP using the three-scales Gabor filter.



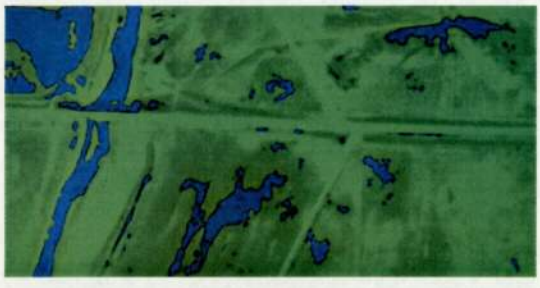
(e) Classification with the committee of the MLP using the three-scales Gabor filter and the MLP using the 4 colour components.



(f) Classification with the GLM using the three-scales Gabor filter and the 4 colour components.



(g) Classification of the whole image with the committee.



(h) Median filtering on the output of the committee on the whole image.

Figure A.1: Set of experiments with an image from the test set.

(a) RGB image.

(b) IR image.

(c) Classification with the MLP using the five-scales Gabor filter.

(d) Classification with the MLP using the three-scales Gabor filter.

(e) Classification with the committee of the MLP using the three-scales Gabor filter and the MLP using the 4 colour components.

(f) Classification with the GLM using the three-scales Gabor filter and the 4 colour components.

(g) Classification of the whole image with the committee.

(h) Median filtering on the output of the committee on the whole image.

Figure A.2: Set of experiments with an image from the test set.

Appendix B

Image classification tool

We implemented a small tool in Matlab to make the classification of the image easier. This application contains the most of the models experimented during the project. The user takes an image from the DigitalGlobe database, then selects the model of his choice and launch the processing. A new image is created based on the output of the image through the model. The pixels classified as water are displayed in blue, while the pixels classified as non-water are represented in green colour. The user has the possibility to save the image.

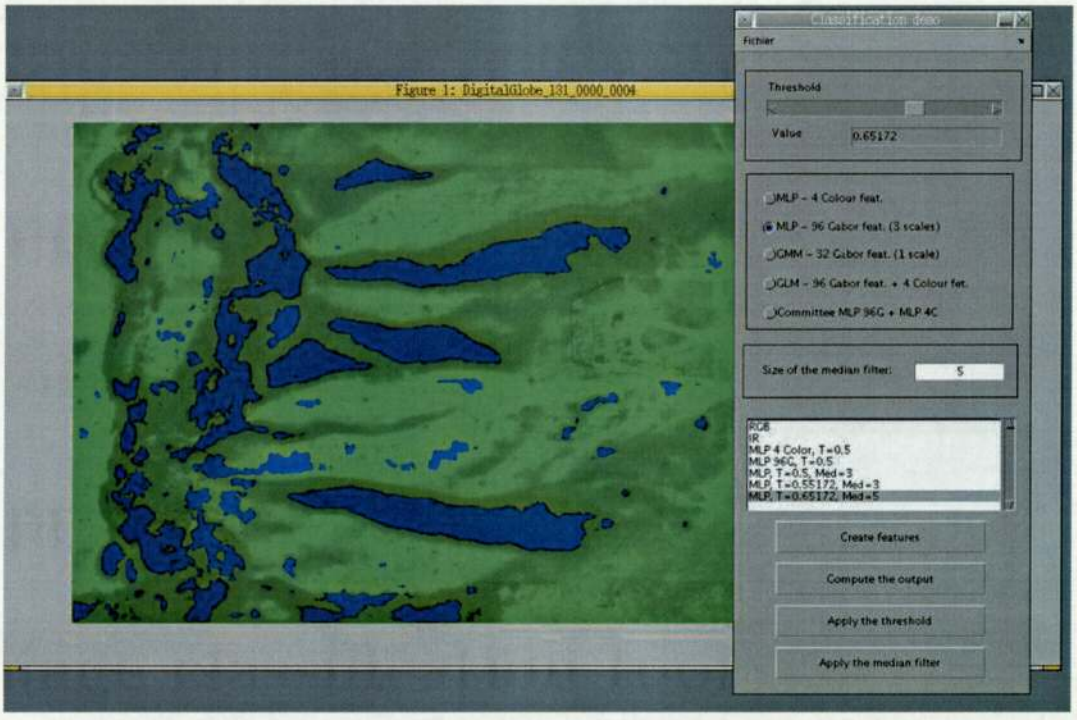


Figure B.1: Interface of the demo application.

Bibliography

[Bishop, 1994] Bishop, C. M. (1994). Novelty detection and neural network validation. In *IEE Proceedings, Special Issue on Applications of Neural Networks*, pages 576–584. Neural Computing Research Group, Department of Computer Science, Aston University.

[Bishop, 1995] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

[Bolme, 2003] Bolme, D. S. (2003). Elastic bunch graph matching. Technical report.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698.

[Chen et al., 2004] Chen, L., Lu, G., and Zhang, D. (2004). Effects of different gabor filter parameters on image retrieval by texture. In *MMM '04: Proceedings of the 10th International Multimedia Modelling Conference*, page 273, Washington, DC, USA. IEEE Computer Society.

[Ji et al., 2004] Ji, Y., Chang, K. H., and Hung, C.-C. (2004). Efficient edge detection and object segmentation using gabor filters. In *Proceedings of 42nd annual Southeast regional conference, Huntsville, Alabama.*

[Krupnik and Elder, 2002] Krupnik, A. and Elder, J. H. (2002). Extraction of lakes from satellite imagery. In *Symposium on Geospatial Theory, Processing and Applications.* Israel Institute of Technology.

[Lachiche and Flach, 2003] Lachiche, N. and Flach, P. A. (2003). Improving accuracy and cost of two-class and multi-class probabilistic classifiers using roc curves. In *ICML*, pages 416–423.

[Li Ma, 2002] Li Ma, Yunhong Wang, T. T. (2002). Iris recognition based on multichannel gabor filtering. In *Proceedings of the 5th Asian Conference on Computer Vision, Melbourne, Australia.* National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.

[MacKay, 1995] MacKay, D. J. C. (1995). Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems.*

[Mallat, 1998] Mallat, S. (1998). *A wavelet tour of signal processing.* Academic press.

[Manjunath, 1992] Manjunath, B. S. (1992). Gabor wavelet transform and application to problems in early vision. In *Proceedings of Twenty-Sixth Asilomar Conference*, volume 2, pages 796–800. University of California.

[McCane, 2001] McCane, B. (2001). Edge detection. Technical report, University of Otago, Dunedin, New Zealand.

[Moller, 1993] Moller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.*, 6(4):525–533.

[Nabney, 2002] Nabney, I. T. (2002). *NETLAB: algorithms for pattern recognition.* Springer-Verlag New York, Inc., New York, NY, USA.

[Nabney and Fermin, 2004] Nabney, I. T. and Fermin, I. (2004). Remote sensing data analysis. Technical report, Aston University.

[Petrou, 1999] Petrou, M. (1999). *Image Processing - The Fundamentals.* Wiley.

[S Wang, 2003] S Wang, T Kubota, J. S. (2003). Salient boundary detection using ratio contour. In *Proceedings of Neural Information Processing Systems Conference.* University of South Carolina and Purdue University.

[Salvador and Chan, 2004] Salvador, S. and Chan, P. (2004). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *ICTAI '04: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)*, pages 576–584, Washington, DC, USA. IEEE Computer Society.

[Smyth, 1996] Smyth, P. (1996). Clustering using monte-carlo cross validation. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.* University of California.

[Smyth, 1998] Smyth, P. (1998). Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 10(1):63–72.

[Sun et al., 2003] Sun, Z., Bebis, G., and Miller, R. (2003). Evolutionary gabor filter optimization with application to vehicle detection. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 307, Washington, DC, USA. IEEE Computer Society.

Index