

UNIVERSITY OF KWAZULU-NATAL

**Gabor Filter Parameter Optimization  
for Multi-textured Images: A Case  
Study on Water Body Extraction from  
Satellite Imagery**

by

Maldean Pillay

Submitted in fulfillment of the academic requirements  
for the degree of Master of Science

in the  
College of Agriculture, Engineering & Science  
School of Mathematics, Statistics and Computer Science

July 2012

*To my family.*

# Preface

The research described in this thesis was performed in the School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville, under the supervision of Dr. Deshendran Moodley. I hereby declare that

1. The research presented in this thesis is my own original work, except where acknowledgement is made by name or in the form of a reference.
2. The work contained herein has not been submitted in part or whole for a degree at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged.
4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
  - (a) Their words have been re-written but the general information attributed to them has been referenced.
  - (b) Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.
5. This thesis does not contain text, graphics or tables copied and pasted from the internet.

Author: ..... Date: .....  
M. Pillay

Certified by: ..... Date: .....  
Dr. Deshendran Moodley, Thesis Supervisor

# Abstract

The analysis and identification of texture is a key area in image processing and computer vision. One of the most prominent texture analysis algorithms is the Gabor Filter. These filters are used by convolving an image with a family of self similar filters or wavelets through the selection of a suitable number of scales and orientations, which are responsible for aiding in the identification of textures of differing coarseness and directions respectively.

While extensively used in a variety of applications, including, biometrics such as iris and facial recognition, their effectiveness depend largely on the manual selection of different parameters values, i.e. the centre frequency, the number of scales and orientations, and the standard deviations. Previous studies have been conducted on how to determine optimal values. However the results are sometimes inconsistent and even contradictory. Furthermore, the selection of the mask size and tile size used in the convolution process has received little attention, presumably since they are image set dependent.

This research attempts to verify specific claims made in previous studies about the influence of the number of scales and orientations, but also to investigate the variation of the filter mask size and tile size for water body extraction from satellite imagery. Optical satellite imagery may contain texture samples that are conceptually the same (belong to the same class), but are structurally different or differ due to changes in illumination, i.e. a texture may appear completely different when the intensity or position of a light source changes.

A systematic testing of the effects of varying the parameter values on optical satellite imagery is conducted. Experiments are designed to verify claims made about the influence of varying the scales and orientations within predetermined ranges, but also to show the considerable changes in classification accuracy when varying the filter mask and tile size. Heuristic techniques such as Genetic Algorithms (GA) can be used to find optimum solutions in application domains where an enumeration approach is not feasible. Hence, the effectiveness of a GA to automate the process of determining optimum Gabor filter parameter values for a given image dataset is also investigated.

The results of the research can be used to facilitate the selection of Gabor filter parameters for applications that involve multi-textured image segmentation or classification, and specifically to guide the selection of appropriate filter mask and tile sizes for automated analysis of satellite imagery.

# Acknowledgements

A special thanks to my family for moral support, Mr. W.C. Naidoo for providing aid with the thesis design and words of encouragement, Prof. J.R Tapamo for his input on the feasibility of this research and NASA for providing access to the EO-1 Hyperspectral imagery used in this research.

Additional thanks to the following technical staff members (SMSCS, UKZN): Mr. S. Greenwood, Mr. J.P. Moodley and Mr. B. Johnston for aiding with the acquisition of data and the provision of additional computing equipment.

I also acknowledge the ICT4EO group (CSIR, Pretoria) for the bursary and support for the period 2008-2009.

---

## Contents

---

<b>Preface</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	2
1.2 Research Objectives . . . . .	2
1.3 Methodology and Expected Impact . . . . .	3
1.4 Thesis Layout . . . . .	4
<b>2 Literature Review</b>	<b>5</b>
2.1 Texture Analysis . . . . .	5
2.1.1 Texture Segmentation and Classification . . . . .	8
2.2 What are Gabor Filters? . . . . .	9
2.2.1 Applying the Gabor Filter . . . . .	12
2.3 Gabor Filter Parameter Optimization . . . . .	15
2.4 Water Body Extraction from Satellite Imagery . . . . .	20
2.4.1 Satellite Imagery . . . . .	20
2.4.2 Water Body Extraction: Sources of Data and Methods . . . . .	23
2.4.3 Using Gabor filters for the Identification of Water Bodies . . . . .	24
2.5 Conclusion . . . . .	26
<b>3 Methodology: Genetic Algorithms</b>	<b>28</b>
3.1 Genetic Algorithms . . . . .	28
3.2 Genetic Representation . . . . .	30
3.3 Selection Operators . . . . .	30

---

3.4	Genetic Operators . . . . .	32
3.5	Conclusion . . . . .	34
<b>4</b>	<b>Design of Experiments</b>	<b>35</b>
4.1	Gabor Filter Design . . . . .	35
4.1.1	Feature Vector Representation and Classification . . . . .	36
4.2	Workflow Design . . . . .	36
4.3	Initial Case Study and Difficulties: . . . . .	38
4.4	The Dataset . . . . .	40
4.4.1	Factors that Influence Sample Selection . . . . .	40
4.4.2	Image Variation and Dataset Preparation . . . . .	41
4.5	Genetic Algorithm Design . . . . .	44
4.6	Summary . . . . .	45
<b>5</b>	<b>Experiments and Results</b>	<b>47</b>
5.1	Experiments . . . . .	47
5.1.1	A Heuristic Approach: Using a Genetic Algorithm . . . . .	48
5.1.2	Linear Enumeration Approach . . . . .	49
5.2	Experiment 1: Identification of Clear and Mildly Turbid Water Bodies . . . . .	49
5.2.1	Heuristic Approach: Results obtained from the Genetic Algorithm . . . . .	49
5.2.2	Results obtained through Linear Enumeration . . . . .	49
5.3	Experiment 1: Summary . . . . .	58
5.3.1	The Genetic Algorithm . . . . .	58
5.3.2	Linear Enumeration . . . . .	59
5.4	Experiment 2: Identification of Clear, Mildly and Highly Turbid Water Bodies . . . . .	59
5.4.1	Heuristic Approach: Results obtained from the Genetic Algorithm . . . . .	59
5.4.2	Results obtained through Linear Enumeration . . . . .	59
5.5	Experiment 2: Summary . . . . .	68
5.5.1	The Genetic Algorithm . . . . .	68
5.5.2	Linear Enumeration . . . . .	69
5.6	Summary . . . . .	69
<b>6</b>	<b>Analysis and Discussion</b>	<b>70</b>
6.1	Effects of Scales and Orientations . . . . .	70
6.1.1	Summary . . . . .	74
6.2	Effects of Filter Mask and Tile Sizes . . . . .	74
6.2.1	Summary . . . . .	76
6.3	Challenges in Water Body Extraction . . . . .	77
6.3.1	Conclusion . . . . .	80
6.4	Automating Gabor Filter Parameter Value Optimization . . . . .	81
6.4.1	Effectiveness of the Genetic Algorithm . . . . .	81
6.4.2	Improving GA Effectiveness and Performance . . . . .	82
6.4.3	Methodology: Recommending the GA . . . . .	83
6.4.4	Summary . . . . .	83
<b>7</b>	<b>Future Work and Conclusion</b>	<b>85</b>
7.1	Future Work . . . . .	85

---

7.2 Conclusion . . . . .	86
--------------------------	----

---

## List of Figures

---

2.1	Types of texture based on structural variation [1]. . . . .	6
2.2	Texture image composed of bricks with differing scale. . . . .	6
2.3	Texture image composed of bricks with differing orientation. . . . .	7
2.4	Effects of illumination on textures [2]. . . . .	7
2.5	Class variation due to illumination [2]. . . . .	7
2.6	Introduction of new textures due to smoothing. . . . .	9
2.7	The half-peak magnitude of the filter responses in the frequency spectrum [3]. . . . .	11
2.8	Real part of a two dimensional Gabor wavelet and its Fourier transform [4].	12
2.9	Two dimensional Gabor filter showing phase offset for the real and imaginary components. . . . .	13
2.10	Gabor filter wavelets of differing scales and orientations. . . . .	13
2.11	The Convolution Process. . . . .	14
2.12	Simple textured image with texture regions of different sizes. . . . .	15
2.13	Selecting filter mask and tile sizes. . . . .	16
2.14	Overlap of Gabor filters in the frequency domain [5]. . . . .	17
2.15	New York Bay Turbid Water [6]. . . . .	25
2.16	New York Bay Turbid Water (Red Band) [6]. . . . .	25
2.17	The difficulty in identifying Highly Turbid water bodies and changes due to sunlight. . . . .	26
4.1	Workflow for the framework. . . . .	37
4.2	Workflow for the framework extended to incorporate the use of a Genetic Algorithm. . . . .	38
4.3	Database Design. . . . .	39
4.4	Sample Selection. . . . .	40
4.5	Images from Zambezi, Mozambique. . . . .	42
4.6	Images from Zambezi with cloud cover. . . . .	43
4.7	Creating Sub-images from the EO-1 Hyperion images. . . . .	44
4.8	Land cover types. . . . .	45
5.1	Identification of Clear and Mildly Turbid Water Bodies. Scales: 3. . . . .	52
5.2	Identification of Clear and Mildly Turbid Water Bodies. Scales: 4. . . . .	52

5.3	Identification of Clear and Mildly Turbid Water Bodies. Scales: 5. . . . .	52
5.4	Identification of Clear and Mildly Turbid Water Bodies. Scales: 6. . . . .	53
5.5	Identification of Clear and Mildly Turbid Water Bodies. Scales: 7. . . . .	53
5.6	Region Images with Scales: 3 and Orientations: 3. . . . .	53
5.7	Region Images with Scales: 3 and Orientations: 5. . . . .	54
5.8	Region Images with Scales: 3 and Orientations: 7. . . . .	54
5.9	Region Images with Scales: 4 and Orientations: 3. . . . .	54
5.10	Region Images with Scales: 4 and Orientations: 5. . . . .	55
5.11	Region Images with Scales: 4 and Orientations: 7. . . . .	55
5.12	Region Images with Scales: 5 and Orientations: 3. . . . .	55
5.13	Region Images with Scales: 5 and Orientations: 5. . . . .	56
5.14	Region Images with Scales: 5 and Orientations: 7. . . . .	56
5.15	Region Images with Scales: 6 and Orientations: 3. . . . .	56
5.16	Region Images with Scales: 6 and Orientations: 5. . . . .	57
5.17	Region Images with Scales: 6 and Orientations: 7. . . . .	57
5.18	Region Images with Scales: 7 and Orientations: 3. . . . .	57
5.19	Region Images with Scales: 7 and Orientations: 5. . . . .	58
5.20	Region Images with Scales: 7 and Orientations: 7. . . . .	58
5.21	Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 3. . . . .	62
5.22	Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 4. . . . .	62
5.23	Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 5. . . . .	62
5.24	Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 6. . . . .	63
5.25	Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 7. . . . .	63
5.26	Region Images with Scales: 3 and Orientations: 3. . . . .	63
5.27	Region Images with Scales: 3 and Orientations: 5. . . . .	64
5.28	Region Images with Scales: 3 and Orientations: 7. . . . .	64
5.29	Region Images with Scales: 4 and Orientations: 3. . . . .	64
5.30	Region Images with Scales: 4 and Orientations: 5. . . . .	65
5.31	Region Images with Scales: 4 and Orientations: 7. . . . .	65
5.32	Region Images with Scales: 5 and Orientations: 3. . . . .	65
5.33	Region Images with Scales: 5 and Orientations: 5. . . . .	66
5.34	Region Images with Scales: 5 and Orientations: 7. . . . .	66
5.35	Region Images with Scales: 6 and Orientations: 3. . . . .	66
5.36	Region Images with Scales: 6 and Orientations: 5. . . . .	67
5.37	Region Images with Scales: 6 and Orientations: 7. . . . .	67
5.38	Region Images with Scales: 7 and Orientations: 3. . . . .	67
5.39	Region Images with Scales: 7 and Orientations: 5. . . . .	68
5.40	Region Images with Scales: 7 and Orientations: 7. . . . .	68
6.1	Experiment 1: Classification Accuracy over Selected Scales. . . . .	71
6.2	Experiment 2: Classification Accuracy over Selected Scales. . . . .	71
6.3	Experiment 1: Classification Accuracy over Selected Orientations. . . . .	72
6.4	Experiment 2: Classification Accuracy over Selected Orientations. . . . .	72
6.5	Experiment 1: Classification Accuracy for Selected Filter Mask and Tile Sizes. . . . .	75
6.6	Experiment 2: Classification Accuracy for Selected Filter Mask and Tile Sizes. . . . .	76

---

6.7	Anomalies in a sub-image from Region 3. . . . .	78
6.8	Anomalies in sub-images from Region 2. . . . .	79
6.9	Anomalies in a sub-image from Region 5. . . . .	80

---

## List of Tables

---

2.1	Gabor Filter Parameters Evaluated in the Literature (adapted from [5] and [7]). . . . .	21
5.1	Experiment 1: Solutions Generated by the GA for the Training and Validation subsets. . . . .	50
5.2	Experiment 1: Solutions Generated by GA <i>Validation</i> subset and <i>Test</i> dataset. . . . .	51
5.3	Experiment 2: Solutions Generated by the GA for the Training and Validation subsets. . . . .	60
5.4	Experiment 2: Solutions Generated by GA for the Validation and Test dataset. . . . .	61
6.1	Experiment 1: Confusion Table for Selected Scales, averaged over all Orientations, and Filter Mask and Tile Size Combinations. . . . .	72
6.2	Experiment 1: Confusion Table for Selected Orientations averaged over all Scale, and Filter Mask and Tile Combinations. . . . .	73
6.3	Experiment 2: Confusion Table for Selected Scales averaged over all Orientations, and Filter Mask and Tile Size Combinations. . . . .	73
6.4	Experiment 2: Confusion Table for Selected Orientations averaged over all Scale, and Filter Mask and Tile Combinations. . . . .	73
6.5	Experiment 1: Confusion Table for Selected Filter Mask and Tile Sizes averaged over all Scales and Orientations . . . . .	75
6.6	Experiment 2: Confusion Table for Selected Filter Mask and Tile Sizes averaged over all Scales and Orientations. . . . .	77
6.7	Experiment 1: Comparison of the best results obtained via the Linear Enumeration and GA Approaches. . . . .	81
6.8	Experiment 2: Comparison of the best results obtained via the Linear Enumeration and GA Approaches. . . . .	81
6.9	Gabor Filter Parameters Updated. . . . .	84

# CHAPTER 1

---

## Introduction

---

Gabor filters are texture analysis algorithms that are prominently used in a variety of image segmentation and classification applications. By trying to imitate the visual system of animals, it is able to discriminate between various textures of different scales and orientations in a multi-textured environment.

Apart from its computational complexity, the Gabor filter does have another major weakness. Its discriminatory ability is influenced by various parameters. These include its centre frequency and the number of scales and orientations. Previous studies provide some guidelines as to what parameter values should be used but are not all consistent. Guidelines for selecting a suitable tile size for window segmentation and the filter mask size of the Gabor filter have also generally been neglected. These parameters are influenced by the texton sizes, as well as the textons sizes in relation to the size of the image tile. Tile sizes that are too small may provide an inadequate texture sample and if too large a single dominant texture can be difficult to isolate for the application of the filter. This also applies to the mask size which should be a suitable fraction of the tile size. Hence, the need to determine the optimum filter mask and tile size for the analysis of a specific multi-textured image dataset.

Though the Gabor filter is widely used, its ability to distinguish conceptually different textures that are very similar in structure from multi-textured images is questionable. A multi-textured image can contain different textures of varying sizes and directions. Optical satellite systems provide multi-textured images that are used in a variety of land

cover classification problems, such as water management and flood detection. Water bodies can be clear, mildly or highly turbid depending on the water contents. Clear water bodies have little or no debris, while mildly turbid water bodies may contain small amounts of soil and silt deposits. Highly turbid water bodies contain large amounts of protruding and surface vegetation. Highly turbid water bodies may therefore exhibit similar properties (pixel intensities and structure) to other land cover types (such as dense vegetation) and are therefore difficult to distinguish.

This thesis aims to verify specific claims made in previous studies about the influence of the number of scales and orientations and investigate the effects of the variation of the filter mask and tile sizes. This is achieved by attempting to extract clear, mildly and highly turbid water bodies from optical satellite imagery which may contain texture samples that are conceptually the same but structurally different.

## 1.1 Problem Statement

The effectiveness of the Gabor filter is influenced by its parameters, independent of the application domain. Previous studies have stressed the need for optimizing the number of scales and orientations and are inconsistent about which values (or range of values) are suitable [8] [5]. The effects of varying the filter mask and tile sizes have also been neglected. There is a clear need for further evaluation of the parameter values and methods to optimize these values. For the purpose of applications such as flood detection and water management, optical satellite imagery contain textures that are often quite complex (are visually difficult to distinguish). The similarity between highly turbid water and natural vegetation will help provide some insight into the effectiveness of the Gabor filter to aid in such applications.

## 1.2 Research Objectives

The broad objective is to investigate the effectiveness of the Gabor filter for multi-textured optical satellite imagery by varying specific parameter values. Specific objectives are to:

1. Identify the effects of varying the values of the following parameters:
  - *Number of scales.*
  - *Number of orientations.*
  - *Filter mask size.*

- *Tile size.*
2. Test the effectiveness of the Gabor filter under different conditions as most guidelines are based on single-textured image databases for the purpose of image retrieval. This is accomplished by trying to identify clear, mildly and highly turbid water bodies from multi-textured optical satellite imagery.
  3. Determine the extent to which the process of determining optimal parameter values for the Gabor filter can be automated using a heuristic approach.

### 1.3 Methodology and Expected Impact

To satisfy the broad objective an experimental framework was designed and implemented to apply the Gabor filter on multi-band and multi-textured satellite imagery, and to support the variation of the filter parameters.

The framework was used to test the different parameter values on two scenarios:

- **Experiment 1:** Identification of Clear and Mildly Turbid Water Bodies.
- **Experiment 2:** This experiment is designed to evaluate the discriminatory ability of the Gabor filter when faced with conceptually different but structurally similar textures. A new texture class representing *highly turbid* water bodies are introduced. These textures exhibit properties similar to that of other vegetation types (non-water bodies).

For each experiment two approaches are used:

- A linear enumeration (brute-force approach).
- A heuristic approach using a Genetic Algorithm.

The findings from these experiments will help determine which of the Gabor filter parameters investigated require optimization and whether a Genetic algorithm can help automate the process with minimal user interaction for a given multi-textured image dataset.

## 1.4 Thesis Layout

This dissertation contains seven chapters. **Chapter 2** provides the background for texture analysis and the Gabor filter: what it is, how it works, and the approaches taken in previous studies to identify the best parameter values. Finally the use of satellite imagery and Gabor filters for water body extraction are discussed. **Chapter 3** provides an overview of Genetic Algorithms while **Chapter 4** describes the design and implementation details of the Gabor filter and Genetic Algorithms, as well as the framework and the data used in the experiments. **Chapter 5** provides a description of the experiments and results. **Chapter 6** provides a critical analysis of the results and describes key findings. **Chapter 7** describes avenues for future work and our conclusions.

This chapter provides background information on the Gabor filter and water body extraction. Section 2.1 introduces key concepts and factors in texture analysis. Section 2.2 describes the Gabor filter, its history and its role in texture analysis. Section 2.3 provides a review of previous studies that evaluate and optimize the parameter values that influence its design, as well as challenges and latest findings. Section 2.4 describes the use of satellite imagery and Gabor filters for water body extraction.

### 2.1 Texture Analysis

Texture analysis is of important interest in the scientific community. Recent advances have allowed for the use of texture analysis to solve a myriad of problems. Being one of the most well known texture analysis algorithms, Gabor filters are used extensively in multiple domains [9] [8] [10]. This section briefly describes the Gabor filter including some of its uses and limitations. Texture analysis will first be discussed to set the context for the rest of this section.

A surface is said to be textured if it contains a large number of texture elements or *texels*, or when placed in an appropriate way it forms a particular texture. The placing of the texels may be random, regular or directional. Generally if a pattern has randomness and regularity then it is probably a texture and not noise or a periodic pattern and can be

characterized by 'busy' microstructures and uniform macrostructures. Different texture types based on structure are illustrated in Figure 2.1 [1].

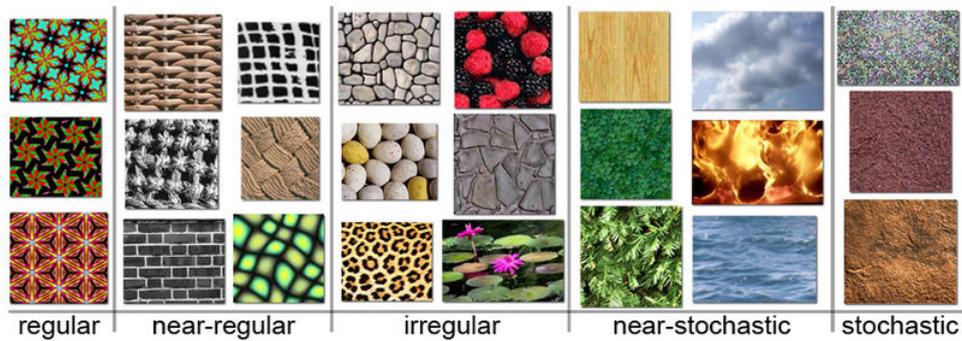


FIGURE 2.1: Types of texture based on structural variation [1].

Another property of a texture is its scale (frequency/coarseness) and orientation (direction). For the purpose of texture classification or segmentation of a multi-textured image, a given image may contain a texture at a different scale as in Figure 2.2 (adapted from a Brodatz image [11]), where the brick texture **B** differs in coarseness but is conceptually the same as **A**. A texture may also occur in a different orientation as shown in Figure 2.3 where the brick texture **C** is the same as texture **A** but lies at a different orientation.

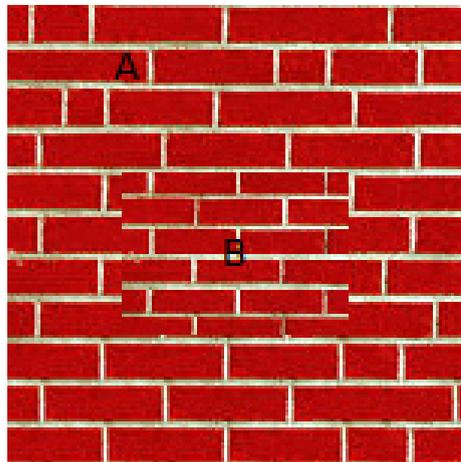


FIGURE 2.2: Texture image composed of bricks with differing scale.

Illumination is also a key factor of texture analysis, as varying intensities of light can alter its appearance as shown in Figure 2.4 [2].

In some cases the variation in illumination can cause large intra-class variation and a smaller inter-class variation as shown in Figure 2.5 and may affect texture classification [12][2].

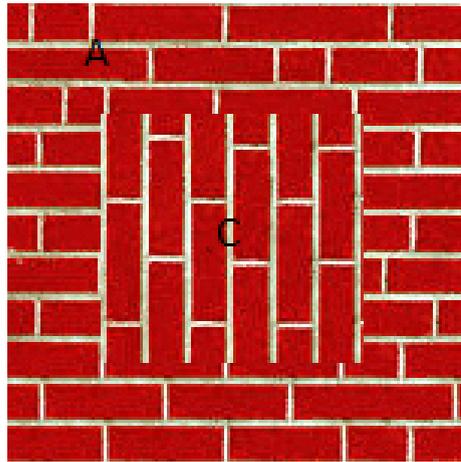


FIGURE 2.3: Texture image composed of bricks with differing orientation.

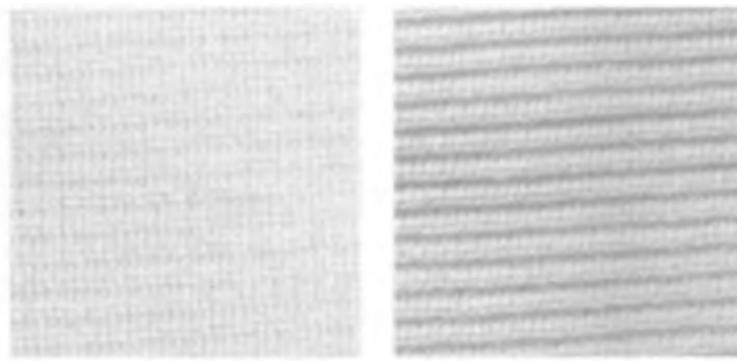


FIGURE 2.4: Effects of illumination on textures [2].

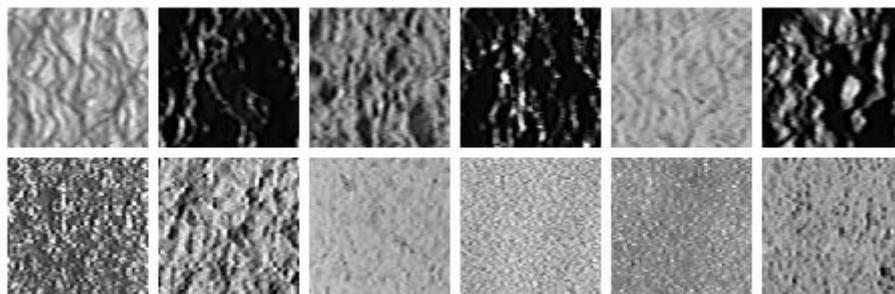


FIGURE 2.5: Class variation due to illumination [2].

Approaches in the field of texture analysis are typically broken down into the following categories [12]:

- **Statistical methods:** Involves the use of statistics to create features that represent textures such as the Gray-Level Co-Occurrence Matrix [13].
- **Spectral methods:** Similar to statistical methods, these collect filter responses at different orientations and scales. The Gabor filter is a good example of a spectral method [14][4].

- **Structural methods:** Textures are viewed as two dimensional patterns composed of textons (*subpatterns*) organized at different positions which are used to model textures. A good example is the application of Laplacian of Gaussian (LoG) masks at different scales for the extraction of more regular textures from an image [15].
- **Stochastic methods:** Textures are assumed to be part of some non-deterministic process under the influence of some parameters which need to be estimated. The Autoregressive (AR) model used for texture segmentation is an example of such a method [16].

### 2.1.1 Texture Segmentation and Classification

For a multi-textured image, texture analysis involves the separation and/or grouping of similar textures into well defined classes. Given some prior knowledge about the number of texture classes in an image, *texture segmentation* generally involves the idea of separating the various texture types in an image by creating a type of boundary map. Similar to segmentation, *texture classification* aims to identify the type of textures in an image given some prior knowledge using texture samples or training sets.

Depending on the nature of the textures in an image, pre or post smoothing (using a filter) is usually done to remove noise and sudden peaks in intensity which prevents considerable variation in the same group of structurally similar textures. This allows a texture analysis algorithm to improve its accuracy when texture segmentation or classification is performed. However, extensive smoothing can lead to very little variance in texture and remove most of the criteria that actually help identify the textures. Most importantly the size of the smoothing filters must also be selected carefully as a large smoothing filter (in pixels) may lead to the unwanted introduction of new texture types as shown in Figure 2.6 [12].

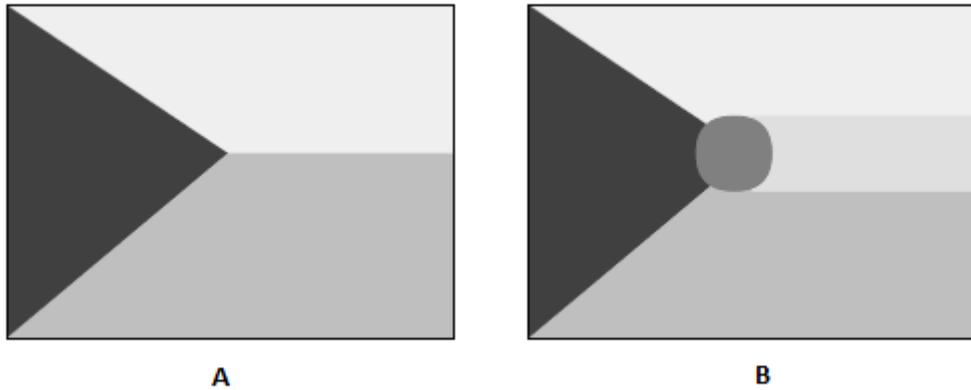


FIGURE 2.6: Introduction of new textures due to smoothing.

**A** is a multi-textures image, and **B** shows the introduction of new textures in **A** due to smoothing [12].

## 2.2 What are Gabor Filters?

Although originally created by Dennis Gabor in 1946, it is the two dimensional Gabor filter that was introduced by John Daugman in the 1980s [4] [17] that has lead to its use in a variety of areas such as multi-purpose feature extraction methods in the field of computer vision and image processing, texture analysis (classification and/or segmentation) [18], edge detection [19], motion detection [20], document analysis [21], and image coding and representation [22]. The most notable applications being biometric authentication such as iris recognition [23], facial recognition [10] and finger print matching [24].

The texture discrimination capability of Gabor filters can be compared to the response of cortical cells responsible for the processing visual signals. This means that the properties of Gabor filters can be seen as a way to model the simple visual system of animals [25].

In its continuous case a two dimensional Gabor filter in the spatial domain is a Gaussian kernel function modulated by a sinusoidal plane wave at a certain frequency and orientation and is defined as:

$$h(x, y) = s(x, y)g(x, y) \quad (2.1)$$

where  $s(x, y)$  is a complex sinusoid, or carrier signal and  $g(x, y)$  is a two dimensional Gaussian envelope.

The complex sinusoid is defined as:

$$s(x, y) = e^{-j2\pi(u_0x+v_0y)} \quad (2.2)$$

The two dimensional Gaussian function is defined as follows:

$$g(x, y) = \frac{1}{\sqrt{2\pi\sigma_x\sigma_y}} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} \quad (2.3)$$

Therefore the two dimensional Gabor filter can be written as:

$$h(x, y) = \frac{1}{\sqrt{2\pi\sigma_x\sigma_y}} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} e^{-j2\pi(u_0x + v_0y)} \quad (2.4)$$

The frequency response of the Gabor filter:

$$\begin{aligned} H(u, v) &= G(u - u_0, v - v_0) \\ H(u, v) &= 2\pi\sigma_x\sigma_y [e^{-2\pi^2[(u-u_0)^2\sigma_x^2 + (v-v_0)^2\sigma_y^2]}] \\ H(u, v) &= \frac{1}{2\pi\sigma_u\sigma_v} e^{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^2} + \frac{(v-v_0)^2}{\sigma_v^2}\right]} \end{aligned} \quad (2.5)$$

where,

$$\sigma_u = \frac{1}{2\pi\sigma_x} \quad (2.6)$$

$$\sigma_v = \frac{1}{2\pi\sigma_y} \quad (2.7)$$

and  $u_0$  and  $v_0$  represent the center frequency.  $\sigma_x$  and  $\sigma_y$  refers to the standard deviations of the Gaussian envelope and determine the bandwidth of the filter, which can be used to derive the Gabor filter mask size (size of the Gabor filter in pixels) in the spatial domain.

While there are many approaches to designing Gabor filters in the literature, the two dimensional Gabor filter can be envisioned as a filter bank consisting of Gabor wavelets with various scales (frequency) and orientations (rotations) generated from a mother wavelet, with each wavelet covering a portion of the frequency spectrum. Given an input image  $I(x, y)$  with size  $P \times Q$ , the discrete Gabor Wavelet transform is given by : [3][8][26]:

$$G_{mn}(x, y) = \sum_s \sum_t I(x - s) \psi_{mn}^*(s, t) \quad (2.8)$$

where  $s$  and  $t$  are the filter mask sizes.  $\psi_{mn}^*$  is a complex conjugate (has both a real and imaginary components) and represents a group of similar functions generated from the mother wavelet  $\psi_{mn}$  through the use of various scales ( $m$ ) and orientations ( $n$ ).

$$\psi(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \exp(j2\pi Wx) \quad (2.9)$$

$W$  is known as the modulation frequency.  $\psi(x, y)$  is the Gaussian modulated by a sinusoid. The self-similar Gabor filters are then created through a generating function:

$$\psi_{m,n}(x, y) = a^{-m}\psi(\acute{x}, \acute{y}) \quad (2.10)$$

$m$  and  $n$  are the current scale and orientations respectively.  $m = 0, 1, 2, \dots, M - 1$  and  $n = 0, 1, 2, \dots, N - 1$ , with  $M$  being the number of scales and  $N$  the number of orientations.

Increasing the scale can be viewed as zooming into a region of interest, while increasing the number of orientations allows for the ability to identify textures that lie in different directions. Figure 2.7 [3] shows the scaling and rotation of the various Gabor filters in the frequency domain. The aim of this design is to ensure that the half-peak magnitude of the filter responses touch each other and hence no bias is introduced due to overlap. Equations 2.11 and 2.12 are used to rotate the Gabor filter using a rotation angle  $\theta$ .

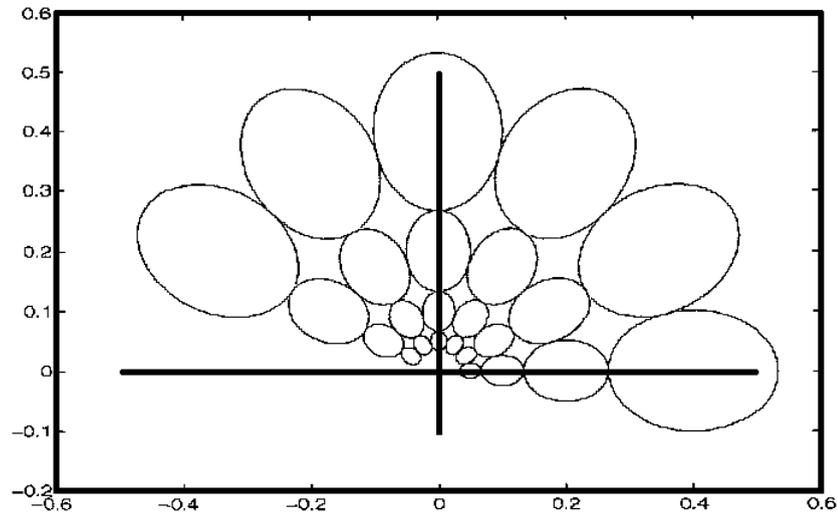


FIGURE 2.7: The half-peak magnitude of the filter responses in the frequency spectrum [3].

The parameter values used are:  $U_h = 0.4$ ,  $U_l = 0.05$ ,  $M = 6$  and  $N = 4$

$a > 1$  and  $\theta = n\pi/N$  and  $a^{-m}$  represents the scaling factor and is used to ensure that the energy is independent of the scale  $m$ .

$$\acute{x} = a^{-m}(x\cos\theta + y\sin\theta) \quad (2.11)$$

$$\acute{y} = a^{-m}(-x\sin\theta + y\cos\theta) \quad (2.12)$$

The wavelet function is a band-pass filter, it passes frequencies within a certain range and rejects frequencies outside that range. The problem is that the entire spectrum needs to be covered which leads to the use of a scaling function to filter to the lowest

level of the transform and ensure that most of the frequency spectrum is covered.

$$a = (U_h/U_l)^{\frac{1}{M-1}} \quad (2.13)$$

$$W_{mn} = a^m U_l \quad (2.14)$$

The standard deviations of the Gaussian envelope which aid in the detection of lines and edges represent the bandwidth and spatial extent of the filters in the  $x$  and  $y$  directions are defined as:

$$\sigma_{x,m,n} = \frac{(a+1)\sqrt{2\ln 2}}{2\pi a^m(a-1)U_l} \quad (2.15)$$

$$\sigma_{y,m,n} = \frac{1}{2\pi \tan(\frac{\pi}{2N}) \sqrt{\frac{U_h^2}{2\ln 2} - (\frac{1}{2\pi\sigma_{x,m,n}})^2}} \quad (2.16)$$

$U_h$  denotes the highest center frequency and  $U_l$  is the lowest center frequency.

Figure 2.8 illustrates the real part of a two dimensional Gabor wavelet centred at the origin (0,0) and its corresponding Fourier Transform (filter response) in the spatial domain [4] while Figure 2.9 shows the real and imaginary components of a two dimensional Gabor filter in the spatial domain. The effects of scaling and rotating the real component of a two dimensional Gabor filter can be seen in Figure 2.10.

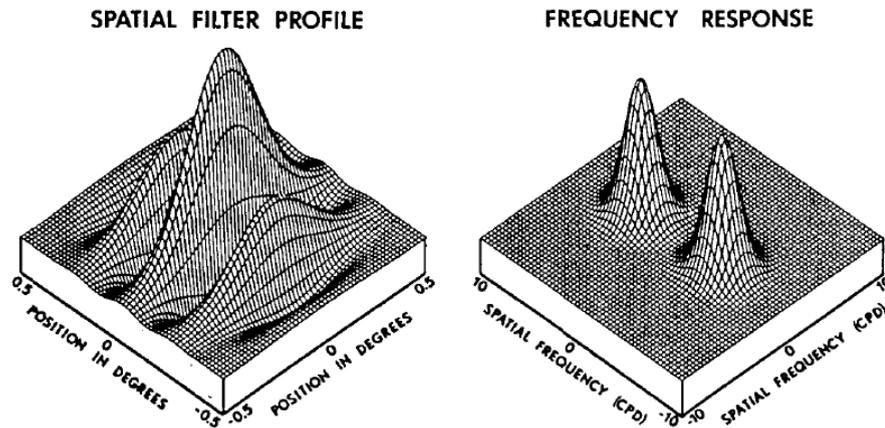


FIGURE 2.8: Real part of a two dimensional Gabor wavelet and its Fourier transform [4].

### 2.2.1 Applying the Gabor Filter

Firstly, when creating and applying the Gabor filter, the *filter mask size* refers to the size of the Gabor filter wavelets in pixels. Tiles refer to windows that are created during *window segmentation* prior to applying the Gabor filter and are used to isolate a single dominant texture. The *tile size* therefore refers to the size of the tiles in pixels.

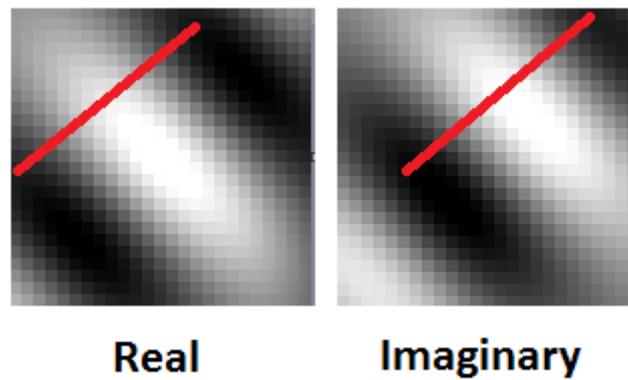


FIGURE 2.9: Two dimensional Gabor filter showing phase offset for the real and imaginary components.

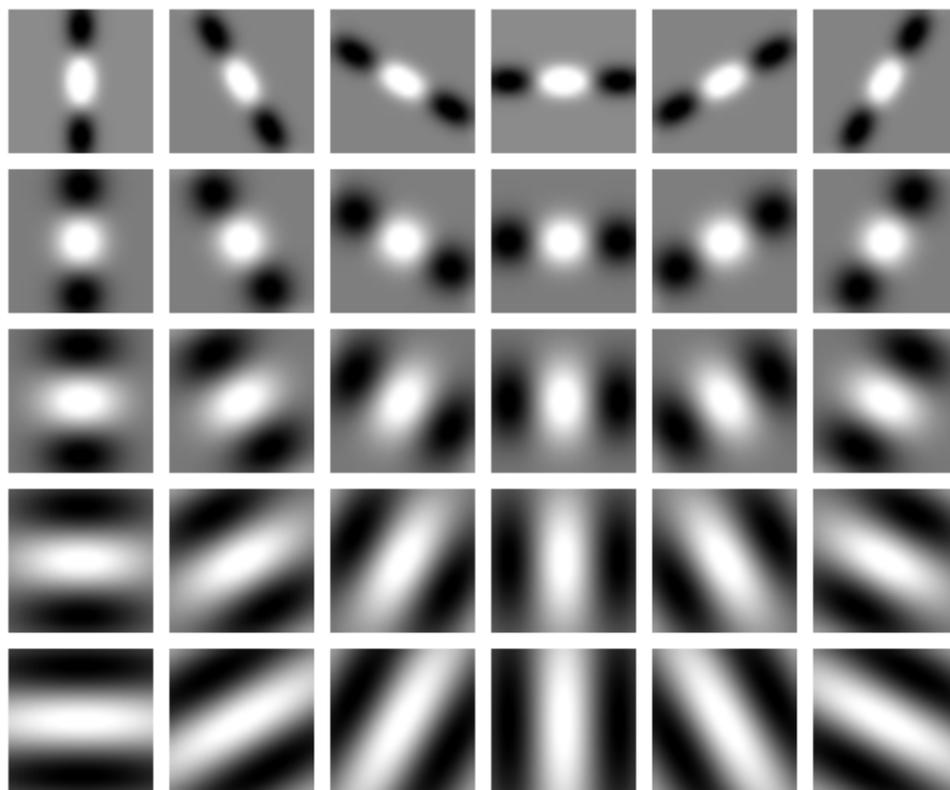


FIGURE 2.10: Gabor filter wavelets of differing scales and orientations.

There are two ways of applying the Gabor filter, either in the *spatial domain* using a discrete convolution in the spatial domain or using the Fast Fourier Transform (FFT) and the Inverse Fourier Transform (IFFT) in the *frequency domain* to improve speed. For simplicity its use in the spatial domain will be discussed.

As illustrated in Figure 2.11 (adapted from [27]), the discrete convolution for an image  $I$  with size  $P \times Q$  and filter mask with size  $S \times T$  with weighted averaging in the spatial

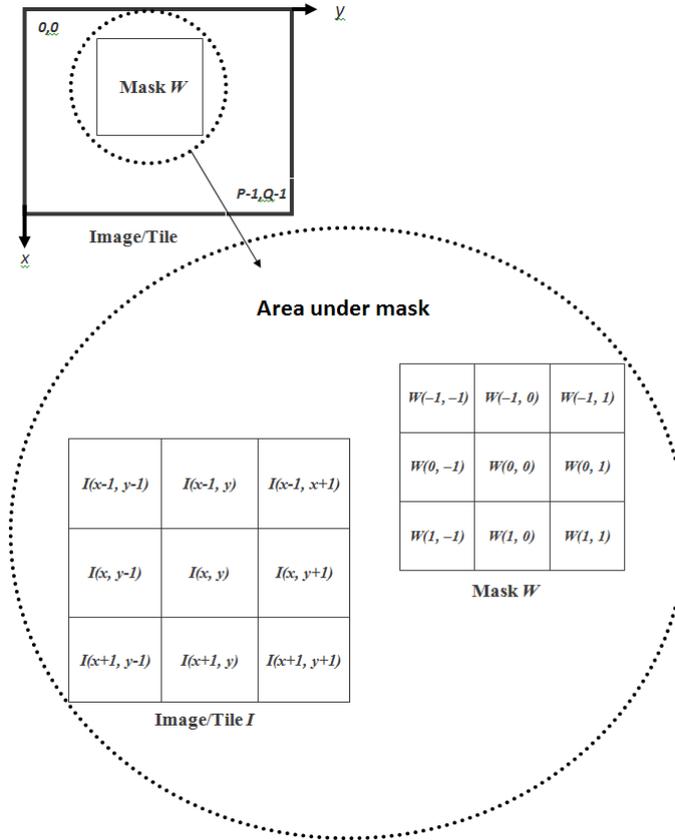


FIGURE 2.11: The Convolution Process.

domain can be expressed as [27]:

$$G(x, y) = \frac{\sum_{j=-a}^a \sum_{k=-b}^b W(j, k) I(x + j, y + k)}{\sum_{j=-a}^a \sum_{k=-b}^b w(j, k)} \quad (2.17)$$

This process is done for all pixels of the image  $I$ , where  $x = 0, 1, 2, \dots, P - 1$  and  $y = 0, 1, 2, \dots, Q - 1$ ,  $S = 2a + 1$  and  $T = 2b + 1$ , with  $S$  and  $T$  odd and,  $a$  and  $b$  are non-negative integers.

From the convolution process two challenges arise when performing texture classification:

- The selection of a suitable filter mask size.
- The selection of a suitable tile size for window segmentation.

The filter mask size and tile size are generally selected based on the size of textons in the image and the size of the image or tile.

Figure 2.12 shows a multi-textured image taken from the Brodatz database [11] containing textures with differing sizes. For the selection of a suitable tile size for texture  $A$  for example, one would assume that the smallest occurrence of texture  $A$  would be suitable

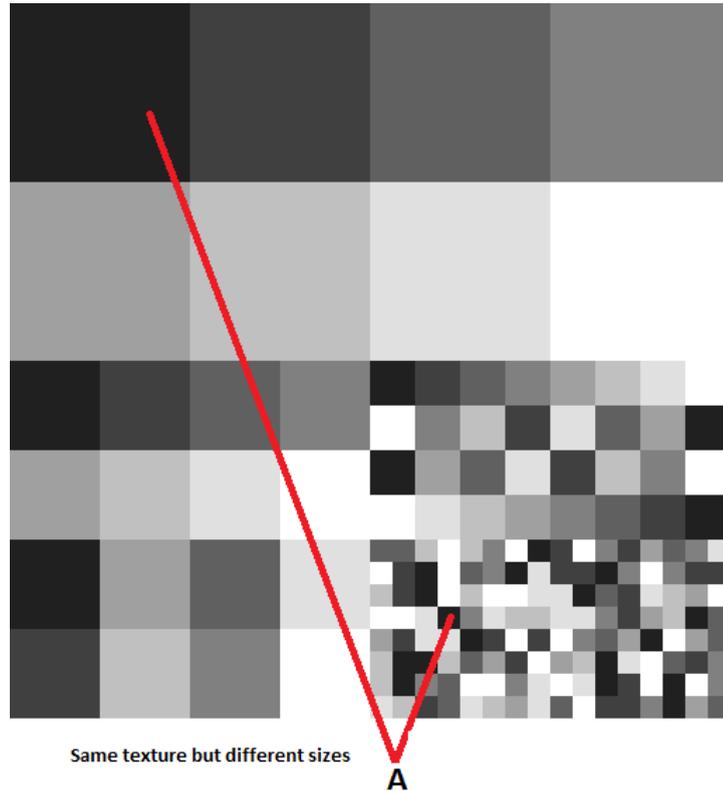


FIGURE 2.12: Simple textured image with texture regions of different sizes.

for determining a *tile* size, but depending on the nature of the textures the smallest occurrence of texture *A* may not have enough discriminatory detail.

The aim of using these tiles is isolate a single texture but it is still possible for no single dominant texture to exist in a given tile as illustrated in Figure 2.13.

### 2.3 Gabor Filter Parameter Optimization

Gabor filters provide superior performance over other known techniques such as the Pyramid-structured Wavelet Transform, Tree-structured Wavelet Transform and Multi-resolution Simultaneous Autoregressive Models [3]. Although widely used in various domains, Gabor filters do have limitations, they depend on a number of parameters, certain assumptions and in some cases contradicting information about their influence during filter design [5] [7].

For parameter optimization researchers have primarily investigated the effects of the varying the following [5] [8] [26]:

- Number of *Scales*  $M$ .

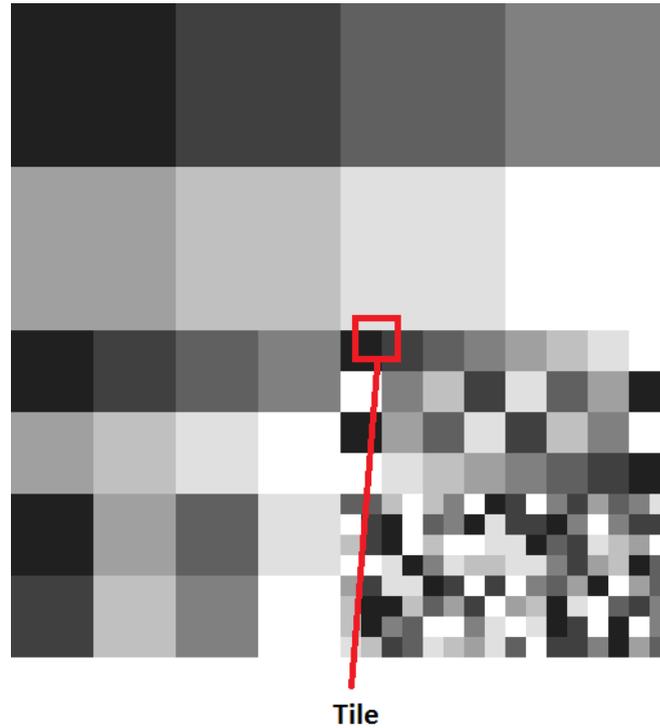


FIGURE 2.13: Selecting filter mask and tile sizes.

- Number of *Orientations*  $N$ .
- *Standard deviations* of the Gaussian envelope ( $\sigma_x$  and  $\sigma_y$ ) in the x and y directions; which can be derived.
- *Mask size*, which represents the size of the Gabor filter kernels in pixels, with the aim of capturing variation of the original signal.
- *Tile size*, which refers to the size of the tiles in pixels, created when performing segmentation, with the aim to isolate a region with a predominant texture.
- *Centre Frequency* of the Gabor filter.

Another factor in filter design is that the scaling and rotating of the Gabor filter in the image space domain has an inverse relation with the radial and angular dimensions of the filter in the frequency domain. The consequences of adjusting Gabor filter scales are explained in detail by Daugman [4]. Assuming the modulation frequency is facing the  $y$  – *axis*, a higher spatial resolution in the  $y$  direction can be achieved at the expense of orientation selectivity, or a higher spatial resolution can be achieved in the  $x$  direction at the expense of spatial-frequency selectivity. This means that scaling has a direct impact on the orientation selectivity of the filter and vice-versa. By reducing the standard deviations along both axes in the spatial domain a greater spatial resolution can be

achieved. This results in the increase in size of the filter in the frequency domain and as a result reduces its scale and orientation selectivity [26].

The latest research with regards to recommended values for aiding in the selection of optimal Gabor filter parameter values will now be discussed with a brief description about the parameters selected for optimization and experiments.

F. Bianconi and A. Fernández [5] used statistical analysis to identify the effects of various Gabor filter parameters on texture classification. They conducted two experiments: one with no overlap of filters in the frequency domain and the other with radial and circumferential overlap as shown in Figure 2.14.

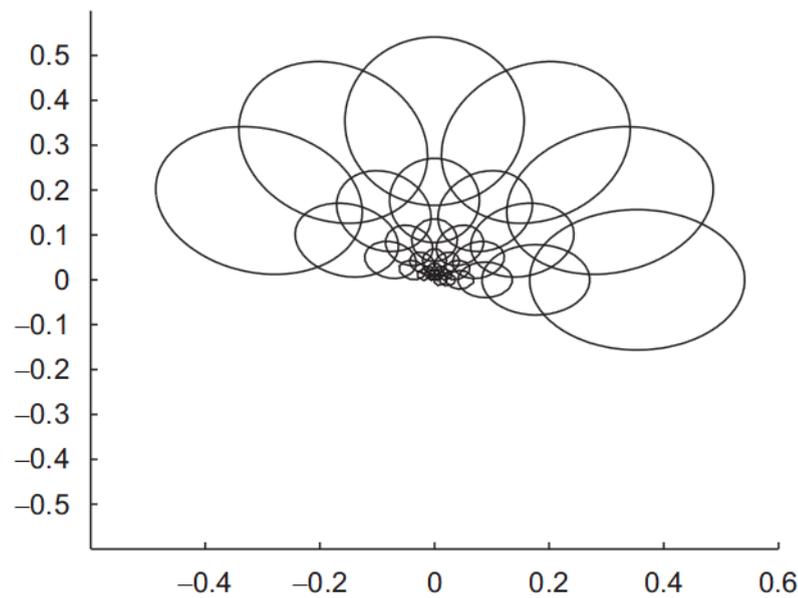


FIGURE 2.14: Overlap of Gabor filters in the frequency domain [5].

The effects of the following parameters were investigated:

- Centre Frequency ( $U_l$  and  $U_h$ ): values chosen are  $\sqrt{2}/4$  and  $\frac{\sigma_x}{2\sigma_x + 2\sqrt{\log 2/\pi}}$  for two experiments, respectively.
- Scales ( $M$ ): values chosen are 4, 5, 6.
- Orientations ( $N$ ): values chosen are 4, 6, 8.
- Standard Deviation in the x direction ( $\sigma_x$ ): values chosen are 0.5, 1.0, 1.5.
- Standard Deviation in the y direction ( $\sigma_y$ ): values chosen are 0.5, 1.0, 1.5.

Though they contain natural but not necessarily real world textures, 40 images from the Brodatz [11] and the Outex [28] databases were used to create eight groups of similar

textures. For each group, a hundred tests were conducted with the use of training and test sets created from the 80 images. They found significant change when varying the centre frequencies and standard deviations in both experiments and no significance change when varying the number of scales and orientations.

For the second experiment, allowing for radial overlap of the filters (non-orthogonal) were found to make a significant difference in classification accuracy. One concern is that by allowing the size of the standard deviations of the Gabor filter to vary the filter banks become too large, thus increasing the computational complexity [7].

Gabor filters are computationally expensive due to the possibility of large filter banks generated by using multiple scales and orientations. Li et al. [7] stressed the point that by taking the parameters  $\sigma_x$  and  $\sigma_y$  into consideration the resultant filter bank becomes even larger and therefore more computationally expensive. They proposed a *supervised approach* using *Mahalanobis* separability to create a compact bank of non-redundant filters. This means that if the distance between one filter and another filter (based on some user defined value) is small (similar), one of the filters is discarded from the initial family of filters. The parameters were predefined and chosen as follows:

- Centre Frequency ( $U_l$  and  $U_h$ ):  $\frac{\sigma_x}{2\sigma_x + 2\sqrt{(\log 2/\pi)}}$ .
- Scales ( $M$ ): 4.
- Orientations ( $N$ ): 4.
- Standard Deviation in the x direction ( $\sigma_x$ ): 0.5, 1.0.
- Standard Deviation in the y direction ( $\sigma_y$ ): 0.5, 1.0.

The initial filter bank consists of 288 filters with the final set of filters reduced to a range of 2.6 – 14.4. After numerous runs a classification accuracy higher than previous methods were observed using the Outex and Brodatz databases. Both the Fisher-ratio and Mahalanobis distance metrics were tested with the latter providing better results.

Zhang et al. [8] and Chen et al. [26] evaluated the effects of specific Gabor filter parameters and highlighted the lack of attention placed on the filter *mask* size of the Gabor filter. If the imagery used contains large texture samples with respect to the image size, then a larger mask size can be utilized. Conversely, smaller filter sizes are used in areas where the dominant textures are quite small.

The following were investigated:

- Scales ( $M$ ): values chosen are 4, 6, 8.

- Orientations ( $N$ ): values chosen are 3, 4, 16.
- Filter Mask size ( $S$ ): 9 x 9, 33 x 33, 61 x 61 and 81 x 81.

All Brodatz database textures were used, each cut into sixteen 128 x 128 sub-images. Some images were then rotated to provide a total of 1852 textures for single-textured image retrieval. An experiment based approach was used to determine the best values for the parameters. Suitable values for the lower and upper centre frequencies are said to lie between 0 and 0.5 respectively. The lower and upper centre frequencies are set to 0.05 and 0.4 respectively and the standard deviations are derived using equations 2.15 and 2.16 [3].

They concluded that the selection of a suitable number of scales and orientations is required and that the filter mask size is dependent on the image size and resolution and hence the best filter mask size must be determined.

Afshang et al. [29] used Genetic Algorithms (GA) with the aim of optimizing some of the Gabor filter parameter values. These include:

- Standard Deviations of the Gaussian envelope in the  $x$  and  $y$  directions.
- Centre Frequency.

A Genetic Algorithm was used with criteria to maximize classification accuracy for the type of image dataset used. Forty images from the Brodatz and forty from the Outex databases were used with each image divided into sixteen non-overlapping sub-images. The resultant set of 160 images was used to form training and test sets. Experiments were then performed on eight groups of images, each containing textures samples from ten classes of similar textures. As expected, for each group of textures the optimum values for the parameters differed across the tests due to the different types of textures contained within.

From these studies a suitable range can be identified for the selection of the number of scales and orientations for the Gabor filter. There is however some confusion as to which parameters (and their respective ranges of values) are the most influential for filter bank design. Parameters such as the scales and orientations are inversely related and need to be chosen carefully. Other studies deem the standard deviations of the Gaussian envelope more important to filter design than the number of scales and orientations.

The Brodatz and Outex databases are useful for testing texture analysis algorithms for single-textured image retrieval as they are of considerable resolution and variation.

There is however no clear evidence that the filters chosen for these databases would provide the same level of performance on real-world and multi-textured image datasets. The filter mask and tile sizes may have a considerable effect when segmenting or classifying *multi-textured* images such as optical satellite imagery with limited spatial resolution and non-homogenous textures of varying sizes that exist in close vicinity to each other. This can make it difficult to select sample textures for class creation and more importantly determine an appropriate mask and tile size for the entire image set.

Table 2.1 has been taken from [5] and [7] and has been expanded to cater for the evaluation of the filter mask and tile sizes.

## 2.4 Water Body Extraction from Satellite Imagery

The identification of water bodies allows for the ability to manage water resources and aid in preventing and monitoring disasters such as floods. Certain types of water bodies such as those containing traces of other substances (protruding or surface vegetation or soil) are much more difficult to distinguish from other land cover types. A suitable source of data and relevant techniques are therefore required to extract the various water bodies.

The ideal sources of data would be the placement of in-situ sensors with wireless telemetry but coverage is generally sparse and maintenance is difficult due to damage caused by debris or vandalism. Instead, this has led to the need for remotely sensed imagery to acquire the relevant data. Airborne or spaceborne sensors provide a myriad of remotely sensed imagery that can be used in a variety of applications.

Airborne missions use aircraft and are quite expensive leading to the use satellite systems as the only viable solution. Satellites can cover the entire globe and frequent regions of interest in a matter of days providing suitable imagery for a variety of applications.

### 2.4.1 Satellite Imagery

Imagery that can be acquired from satellite systems include optical (multispectral or hyperspectral) and Synthetic Aperture Radar (SAR) imagery. Multispectral and hyperspectral images are produced by sensors that use the visible light (red, green and blue) and infrared frequencies of the electromagnetic spectrum [37]. Using these light waves multiple image bands are created, with each band providing unique descriptions of the various phenomena being observed. For example, vegetation appears much brighter than soil or water in an infrared image. However, sunlight and reflective surfaces can

TABLE 2.1: Gabor Filter Parameters Evaluated in the Literature (adapted from [5] and [7]).

	Scales ( $M$ )	Orientations ( $N$ )	Upper Central Frequency ( $\bar{U}_h$ )	$\sigma_x$	$\sigma_y$	Mask Size ( $S$ )
M.R. Turner [30]	4	4	1/4			
A.K. Jain and F. Farrokhnia [31]	7	4	$\sqrt{2}/4$			
B.S. Majunath and Y. Ma [3]	4	6	0.4			
A.K. Jain et al. [32]	5	4	$\sqrt{2}/4$			
P. Kruizinga and N. Petkov [33]	3	8	1/5.47			
Y. Rubner [34]	4	6	0.3			
S. Li and J. Shawe-Taylor [35]	4 - 6	4 - 6	0.4			
D.A. Clausi and H. Deng [36]	4	4	$\sqrt{2}/4$			
Chen et al. [26]	6	4	0.4			
F. Bianconi and A. Fernández [5]	4, 5, 6 <sup>(1)</sup>	5, 6, 8 <sup>(1)</sup>	$\sqrt{2}/4$ and $\frac{\sigma_x}{2\sigma_x+2\sqrt{\log 2/\pi}}$	0.5	0.5	
W. Li et al. [7]	4	4	$\frac{\sigma_x}{2\sigma_x+2\sqrt{\log 2/\pi}}$	0.5, 1.0 <sup>(2)</sup>	0.5, 1.0 <sup>(2)</sup>	13x13 <sup>(3)</sup>
M. Afshang et al. [29]			varies	varies	varies	

<sup>1</sup> Statistically no significant change in classification accuracy.<sup>2</sup> Used to create initial set of Gabor filters.<sup>3</sup> Optimum mask size in pixels for Brodatz images.

cause classification errors. These types of imagery are cheaper than SAR imagery and more widely available.

SAR sensors use radar waves that bounce or reflect off surfaces and provide a good description of the topography of an area. A key property of SAR for this case study, is that the incident radar waves are reflected away from water bodies such as rivers and lakes unlike other land cover types. This results in these water bodies appearing much darker than the surrounding vegetation and soil in the resultant SAR image. By using two images of the same region taken at different angles, Digital Elevation Models (DEMs) can be created and provide details of the elevation of the terrain. Perhaps the most important property of a SAR sensor is its ability to penetrate cloud cover unlike optical sensors, and can therefore be used during adverse atmospheric conditions.

While there are many public and military satellite systems available, their capabilities can differ considerably. Imagery produced can differ by the [37]:

- *Spatial resolution*: the area (in square metres) on the ground that corresponds to a single pixel. Low to medium resolution satellites provide 30 – 1000 m and more modern high resolution satellites of with 1m or less.
- *Image bands*: images produced by the sensor using various frequency ranges.
- *Revisit time*: how often the satellite system produces images for a specific area. Some satellites such as EO-1 have side-look capabilities which allows for a more frequent revisit, but the scenes may have different viewing angles.
- *Availability*: whether the satellite is for military (such as Cosmos 2441 and Helios 2B) or civil (such as LANDSAT, MODIS and EO-1) purposes. Low cost missions or more readily available public data sets from newer systems may allow in the improvement of existing applications.

### **Selecting Satellite Imagery**

Even though some satellites may provide high resolution data, the choice is dependent on the image bands they provide and how often the images are produced. For example, a satellite may provide multi-band and high resolution imagery to allow one to clearly identify water bodies, but has a low revisit time and hence is not suitable for applications like flood detection.

The National Point of Contact in the Netherlands [38] provides information on the capabilities of all earth observation satellites. The information is gathered from public sources (the Internet) and is regularly updated. The tables they provide contain satellite systems that have completed their missions and some that are yet to be launched, and as a result may not reflect their current status and availability.

### 2.4.2 Water Body Extraction: Sources of Data and Methods

Techniques for the identification of land cover and hence water body extraction for use with either SAR and/or optical imagery will now be discussed. The simplest of methods is the use of pixel intensities from single or a combination of bands. This can be fairly straight forward such as user defined pixel value ranges for classes or better techniques such as the *mean shift algorithm*, which is a type of clustering algorithm that tries to isolate homogenous regions by determining the most common values (the mode) within a local region.

For optical imagery, one of the most well known “techniques” is the use of composite imagery computed from multiple bands. Composite imagery such as Normalized Vegetation Difference Index (NDVI) and Normalized Water Difference Index (NDWI) use the near-infrared and visible red bands, and near-infrared and visible green bands respectively to generate an image with pixels intensities in the range  $[-1, 1]$ . For NDVI highly positive values represent vegetation with soils varying in pixel intensity due to water content. Clear water bodies appear much darker with values close to zero (or negative values), with the opposite true for NDWI.

Each satellite sensor has its own characteristics. For indices such as NDVI, NDWI and the mean shift algorithm, user defined ranges of pixel intensities that represent classes of land cover will vary depending on the type of satellite images used. The resultant pixel intensities can also vary due to atmospheric conditions. There is also no guarantee that a land cover class will not exhibit similar pixel intensities to other classes. These techniques may work well for clear water types but not for mildly turbid (muddy) water and highly turbid waters (surface vegetation) and as a result texture analysis may be more suitable.

The ideal situation would be the use of SAR imagery and/or suitable DEMs imagery together with optical imagery for the identification and monitoring of water bodies of varying sizes and types. However, publicly available 90 m resolution DEMs are just not good enough for the extraction of “thin” water bodies such as rivers and the acquisition of data for developing countries are few and far between. Suitable times series are also difficult to acquire, even for optical imagery.

High resolution DEMs are difficult to acquire, SAR imagery is expensive and publicly available datasets are also almost impossible to find. Therefore the only viable option is to use optical imagery.

Unfortunately for the coverage of developing countries, there is also limited access and availability of optical satellite images with a suitable time series (even for research purposes) and this is still a major concern.

### **2.4.3 Using Gabor filters for the Identification of Water Bodies**

Gabor filters have been used on satellite imagery for a variety of land cover segmentation and classification problems [39][40]. This involves selecting and categorizing texture samples into well defined classes with the aims of identifying various land cover types such as mountains, dense vegetation, shrubs, rivers and urban areas, etc.

The focus of this case study is the identification of not just clear water bodies commonly performed using spectral based classifiers (mentioned earlier) but the identification of turbid water bodies that are more difficult to extract.

Nath and Deb [41] provide a list of techniques used for the extraction of water bodies from some optical and SAR satellite imagery. These include Neural networks and Maximum-Likelihood classifiers. They concluded that while many techniques exist, none of them are universally accepted and none are applicable to all available satellite images. Gabor filters are mentioned but not used and the identification of turbid and highly turbid water bodies were also not tackled.

Figures 2.15 and 2.16 show colour and red band (TM3) LANDSAT images of New York Bay used by Hellweger et al. [6] for identifying turbid waters. These rivers contain a green tinge with noticeable texture and are somewhat distinguishable from the surrounding urban environment. The main concern in this research were the shadows that were being cast from buildings which affected classification results.



FIGURE 2.15: New York Bay Turbid Water [6].



FIGURE 2.16: New York Bay Turbid Water (Red Band) [6].

As illustrated in Figure 2.17, the data used in this research (discussed in more detail in Chapter 3) however, have no shadows from urban structures that distort the land cover types. Some of the major concerns are:

- Highly turbid water containing vegetation that exhibit spectral properties similar to that of other vegetation types. Their appearance also differs due to varying illumination caused by sunlight and natural changes in land cover. This is evident in the images marked **A**.
- Mildly turbid water with soil deposits (**B**).
- The similarity in “smoothness” of the dense vegetation textures (**C**) to that of clear and mildly turbid water bodies (**D**).

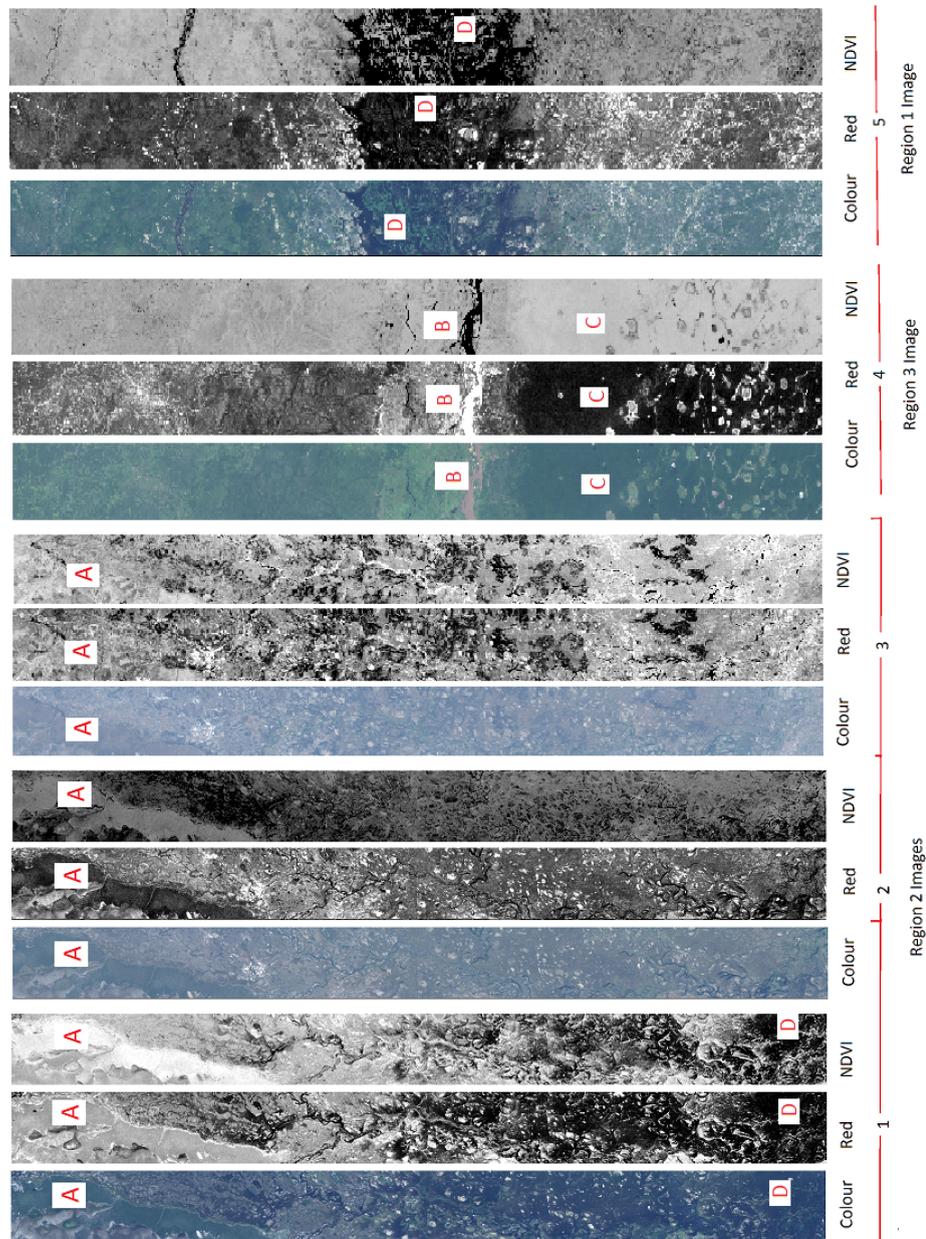


FIGURE 2.17: The difficulty in identifying Highly Turbid water bodies and changes due to sunlight.

## 2.5 Conclusion

From various studies, it can be concluded that while the Gabor filter is a widely adopted technique for texture analysis its effectiveness is dependent on the selection of suitable parameters values. Studies into the effects of varying these parameters (centre frequency, scales, orientations and standard deviations) are usually based on experiments using the Brodatz and Outex databases. However, these findings do have limitations, there is still some contradiction as to which values to use and which parameters need to be optimized.

Only a few of the studies have stressed the importance of the Gabor filter mask size and the effects of varying the tile sizes for window segmentation has not been investigated. Suitable values for the filter mask and tile sizes are image set dependent and there are limited studies that use multi-textured images for the optimization of these parameters.

There are a variety of techniques used for water body extraction from satellite imagery, with SAR imagery generally being preferable as the source of data. However, optical imagery is more commonly available than SAR imagery and texture analysis algorithms such as the Gabor filter have already been used in other land cover classification problems that use optical satellite imagery.

Mildly and highly turbid water bodies is a concern, as there is limited research on how to extract these from optical satellite imagery alone. Spectral based classifiers cannot be used for these water bodies since most of their pixel intensities are similar to those of other land cover types.

---

## Methodology: Genetic Algorithms

---

In this research, the Genetic Algorithm (GA) is investigated as a heuristic tool for Gabor filter parameter optimization. This chapter provides a description of Genetic Algorithms. The first section provides an introduction and the basic requirements of the GA, while Section 3.2 describes encoding schemes. Section 3.3 highlights some of the selection operators used and Section 3.4 provides a list of GA operators.

### 3.1 Genetic Algorithms

Heuristic algorithms are those algorithms that find solutions among all possible solutions. It is not guaranteed that the best solution will be found but it is possible if the optimum solution or solution space is known.

Genetic Algorithms (GA) are heuristics that are inspired by and aims to mimic the natural evolution process by using techniques such inheritance, mutation, selection and crossovers [42]. They have been used in a variety of fields such as bioinformatics, phylogenetics, computational science, engineering, mathematics and physics, etc.

GAs have proved to be capable of solving large, complex problems in areas which other techniques have experienced difficulties. One of its major advantages is its ability to tackle problems with multiple local optima. For a given numerical search and optimization problem, a list of possible solutions is searched in order to find the solution that best

fits a given problem. This is known as the *search space*, and could possibly be of infinite size. Enumeration can also be used to evaluate the search space for a few parameters but if the search space growth is complex, or the algorithm for whose parameters are being optimized is computationally expensive, then GAs are deemed more suitable.

The basic requirements of a Genetic Algorithm are:

1. A population of individuals or estimates (guesses) of the solution  $p$ .
2. A genetic representation of the solution, usually an array, string of bits or trees.
3. A fitness function to evaluate the solution.
4. Methods for mixing fragments to form new solutions such as a cross-over operator.
5. A Mutation operator to prevent a possible loss of diversity within a given population.

A typical Genetic Algorithm involves the iteration through multiple generations  $g$ , ( $g = 0, 1, 2, \dots, G - 1$ ) of a population  $p$  of  $N$  individuals. The GA can be terminated after a certain number of predetermined generations  $G$  has been reached. It can also be terminated if no change in the best fitness ( $f_{max}$ ) over a given number of generations is observed or until a generation contains an individual with a suitable fitness. For each generation the fitness  $f$  of all individuals  $i$  ( $i = 0, 1, 2, \dots, N - 1$ ) of the current population  $p$  are evaluated. Individuals that meet a certain criteria (or randomly chosen) are then *selected* as *cross-over* pairs. The offspring of these individuals have a chance of undergoing *mutation*, the resultant individuals (and possibly their parents) are then placed into a new population  $p + 1$ . The remaining individuals or weakest individuals from the current population  $p$  are not allowed to form part of the new population  $p + 1$ .

The above is just the basic case. The way in which individuals are chosen for cross-overs and mutation and the manner in which new populations are created can differ depending on the type of selection operators used.

Based on various books and tutorials [42][43][44], a brief description of Genetic Representations, Selection Operators and Genetic Operations are explained below.

## 3.2 Genetic Representation

A suitable method for representing the genes and chromosomes of individuals are required. Whilst there is no single best *encoding scheme*, one usually selects the appropriate scheme depending on the type of application used. The following are some of the more popular encoding schemes used:

**Bit strings** are the most commonly used encoding, the main reason being Holland's work in 1975 [45] which many have tended to follow. Most of the literature on Genetic Algorithms are also based on bit strings of fixed length and fixed-order. Research on parameter settings for cross-over and mutation rates are mostly done for binary encodings.

**Many-Character and Real-Valued Encodings:** One way to form chromosomes is the use of alphabets of many characters or real numbers. Applications include representations of graph-generation grammars, and real-valued representation for neural-network weights.

**Tree encoding** schemes are mainly used for evolving programs or expressions, for genetic programming. Every chromosome is a tree of some objects, such as functions or commands in programming language. There is a possibility that the trees can grow uncontrollably large, and can also be very difficult to simplify.

Unlike Binary Genetic Algorithms (BGM), Real-valued Genetic Algorithms (RGM) use real values, no encoding schemes are used [46].

## 3.3 Selection Operators

The next part in the design of a Genetic Algorithm is the manner in which individuals from a population are chosen to create offspring for the next generation, including the number of offspring each pair produces. A balance has to be found to gain the best results, as very strong selection results in a population saturated with suboptimal highly fit individuals, thus reducing the diversity needed for further progress. Conversely a very weak selection will result in a very slow evolution rate. Various selection schemes have been proposed, seven of which are described below [42] and [43].

- **Fitness-Proportionate Selection**

In fitness-proportionate selection, the *expected* number of times an individual  $I$  will be selected as a parent to produce offspring, is the fitness  $f$  of  $I$  divided by the sum of the population fitness  $P$ .

The *roulette wheel* is the most common type: each individual  $I$  is assigned a slice of the *roulette wheel*, with each slice being proportional to the individual's fitness. For each spin a random number  $r$  ( $0 \leq r \leq S$ , where  $S$  is the sum of all the chromosome fitness values) is generated. An individual with an *expected value* that exceeds  $r$  is added to be in the collection of parents for the next generation. Another variation, proposed by James Baker (1987) [47] is a sampling method called *stochastic universal sampling* (SUS), where a roulette wheel with  $N$  equally spaced pointers is spun  $N$  times to select  $N$  individuals (parents). However both fitness-proportionate selection and Stochastic Universal Sampling are both susceptible to *premature convergence*, due to limited variation in fitness in subsequent generations.

- **Sigma Scaling**

Sigma Scaling is designed to keep the extent at which highly fit individuals are allowed to produce many offspring relatively constant. It does so by preventing the selection pressure from being strong too early, to prevent premature convergence, and not too weak once the population has stabilized.

- **Elitism**

When a new population is created by crossover and mutation, there is a high possibility that some of the best chromosome/(s) will be lost. Elitism allows for the retention of a few of the best individuals from a population by copying some of these individuals to a new population, and can substantially increase the performance of the GA.

- **Boltzmann Selection**

Unlike *sigma scaling*, sometimes different amounts of selection pressure is needed at different times in a run. Less fit individuals can reproduce at rates similar to that of fitter individuals. This may be accomplished by having selection occur slowly while maintaining a lot of variation in the population early on, but increasing the selection of highly fit individuals (higher probability) later on. *Boltzmann selection* is one of such approaches that uses a temperature to adjust the rate at which individuals produce offspring throughout the run.

- **Rank Selection**

In *Rank selection* individuals in the population are ranked according to *fitness* and the *expected value* of each individual depends on its *rank* rather than its fitness. Since the use of absolute fitness can lead to convergence problems, ranking prevents the possibility of the largest share of offspring being produced by the fittest individuals. Selection pressure is also reduced when the fitness variance is high and keeps up selection pressure when the fitness variance is low.

- **Tournament Selection**

Unlike the fitness-proportionate methods that require two passes through the population at each generation, one pass is used to compute the *mean fitness* and another pass is required to compute the *expected value* of each individual. Similar to rank selection in selection pressure, *tournament selection* is computationally more efficient. Two individuals are chosen at random from the population. A random number  $r$  is then chosen between 0 and 1, if  $r < p$  ( $p$  is a parameter) and the individual with the greater fitness is selected to be a parent, otherwise the less fit individual is selected. The two individuals are then returned to the original population and can be selected again.

- **Steady-State Selection**

In Steady-State Selection, in every generation a few individuals with a high fitness are selected as parents, those with poor fitness values are replaced by the new resultant offspring. The rest of the population survives to new generation with the offspring as well as their parents.

### 3.4 Genetic Operators

Genetic operators are techniques that try to mirror the natural manipulation of chromosomes via crossovers and mutations. Listed below are some of the commonly used crossover techniques as well as the what mutation is and its purpose [42] [43].

- **Single-point Crossover** is the simplest form of crossover. A single crossover point  $p$  is chosen at random and the parts of two parents  $I_1$  and  $I_2$  after the crossover position are exchanged to form two offspring. One major limitation of the single-point crossover is its inability to combine all possible schemas and main long length schemas. It is also noted that in single-point crossover the segments exchanged between the two parents always contain the endpoints of the chromosomes of the individuals  $I_1$  and  $I_2$ .
- **Two-point or Multi-point Crossover** is used to reduce positional bias and endpoint bias. Two positions are chosen at random and the segments between them are exchanged. Two-point crossover is less likely to disrupt schemas with large defining lengths and can combine more schemas than single-point crossover. Also, the exchanged segments do not necessarily contain the endpoints of the strings of the parents.

- **Cut and splice** is similar to single-point crossover, but the difference is that each chromosome has a separate crossover point and may result in a change of length of the child chromosomes.
- **Uniform Crossover** allows the parent chromosomes to contribute at the gene level rather than at the segment level. A mixing ratio for the crossover is chosen so that a certain number of randomly chosen genes are chosen from the respective parents. Bits are typically swapped with a probability of 0.2.
- **Half Uniform Crossover:** half of the non matching bits are swapped. The Hamming distance is first calculated and then divided by two and is used to determine how many of the genes that *do not* match between the two parents will be swapped.

The suitability of a particular crossover operator depends on the type of fitness function and encoding schemes used. Even with various crossover operators (with key criteria being positional bias, degree of disruption potential and the creation of different schemas), there is still no comprehensive evidence as to which type of crossover is to be used [42].

For real valued chromosomes (not used in this research), known techniques include: Intermediate recombination, Line recombination and Extended line recombination. Explanations below are provided by the Genetic and Evolutionary Algorithm Toolbox Website [48]:

- For **Intermediate recombination**, variable values of the offspring are chosen around and between the variable values of their parents ( $P_1$  and  $P_2$ ). Offspring are produced according to the rule the following rule:

$$Var_i^0 = Var_i^{P_1} \cdot a_i + Var_i^{P_2} \cdot (1 - a_i) \quad (3.1)$$

$1 \leq i \leq N$ ,  $-d \leq a_i \leq 1 + d$  and  $d = 0.25 \cdot a_i$  for each new  $i$ .  $a$  is the scaling factor that is selected uniformly at random over an interval  $[-d, 1 + d]$  for each new variable.  $d$  defines the size of the area for possible offspring. A value of  $d = 0$  defines an area for the offspring which is the same size as the area spanned by the parents. The area for the variables shrinks over each generations. A value of  $d = 0.25$  statistically ensures that the variable area of the offspring is the same as the variable area spanned by the variables of the parents.

- **Line recombination** is similar to intermediate recombination but instead only a single  $a$  value is used for all variables.

- For **Extended line recombination** offspring are generated on a line defined by the variable values of the parents but is not restricted to the line between the parents and a small area outside. The parents are only used to define a line where possible offspring may be created. The domain of the variables are used to define the area for possible offspring. The offspring are not uniform at random and there is a high chance of them being created near their parents. Offspring are produced according to the following rule:

$$Var_i^0 = Var_i^P \cdot s_i \cdot r_i \cdot a \frac{Var_i^{P_2} - Var_i^{P_1}}{\|Var_i^{P_1} - Var_i^{P_2}\|} \quad (3.2)$$

$1 \leq i \leq N$ ,  $a = 2^{-ku}$  is the relative step size.  $k$  is the mutation precision and typical values range from 4 – 20.  $0 \leq u \leq 1$  (uniform at random).  $r = r$ . domain of  $r$ , and represents the range of recombination steps and is usually 10% of the domain of the variable.  $s_i \in \{-1, 1\}$  and represents the direction of the recombination.

### Mutation

The mutation operator is used to maintain and introduce genetic diversity from one generation to the next and to avoid local minima by preventing premature convergence or slowing down the GA. The basic concept is choosing an arbitrary position in a chromosome and altering it from its original state, the most common method involves the generation of a random variable for each allele in the chromosome. A given probability is then used to determine if a random variable indicates a particular allele will be modified and is known as single point mutation. Other types of mutation include inversion, floating point mutation swaps and scrambles.

For real valued GAs, a type of “swingometer” crossover operator can be use. For a given parameter  $c$ , a certain number is added or extracted from  $c$  the resultant number that is generated is restricted by some defined upper limit  $U$  and lower limit  $L$  [49].

## 3.5 Conclusion

Genetic Algorithms can be used as a means of determining some of the best Gabor filter parameter values for a given image dataset. As mentioned in Chapter 2, Afshang et. al [29] have already used a GA to find optimum values of the standard deviations of the Gaussian envelope and the centre frequency. For the purpose of determining the optimum filter mask and tile sizes, a well constructed GA will save computation time for large solution spaces and datasets. There is potential to design a tool that uses a GA to automate the process of determining the best parameter values for the Gabor filter for any given multi-textured image dataset.

---

## Design of Experiments

---

This chapter describes the design of the Gabor filter, the Genetic Algorithm and an experimental framework that was developed for the experiments. Section 4.1 describes the design of the Gabor filter, representation of the feature vectors and classification. The framework is described in Section 4.2. The assumptions and constraints on the research, description of the dataset and finally the Genetic Algorithm design are discussed in Sections 4.4 and 4.5.

### 4.1 Gabor Filter Design

The values of the following parameters are investigated:

- *Number of scales.*
- *Number of orientations.*
- *Filter mask size.*
- *Tile size.*

Masks ranging from 3 x 3 pixels to about a third of the tile size were considered. In order to centre the mask only odd sized masks were investigated. True overlapping of tiles for window segmentation is not used for this research, however, even sized tiles

were allowed. When referring to a filter mask or tile of size  $m$  it is assumed that the dimensions are  $m \times m$  pixels.

The lower and upper values for the centre frequencies are kept constant at  $0.05$  and  $0.4$  (as used in the literature) with the filter bandwidth ( $\sigma_x$  and  $\sigma_y$ ) being derived using the equations 2.15 and 2.16 [8].

### 4.1.1 Feature Vector Representation and Classification

Statistics are required to represent the filtered image  $\mathbf{G}$  of size  $P \times Q$ , firstly an array of magnitudes  $\mathbf{E}$  that represent the energy at different scales  $m$  and orientations  $n$  is computed:

$$E(m, n) = \sum_{x=0}^{P-1} \sum_{y=0}^{Q-1} |G_{mn}(x, y)| \quad (4.1)$$

$m = 0, 1, \dots, M - 1$  and  $n = 0, 1, \dots, N - 1$ . The first and second order statistics, namely the mean and standard deviation are generally used, and are given by:

$$\mu_{mn} = \frac{E(m, n)}{P \times Q} \quad (4.2)$$

$$\sigma_{mn} = \frac{\sqrt{\sum_{x=0}^{P-1} \sum_{y=0}^{Q-1} (|G_{m,n}(x, y)| - \mu_{mn})^2}}{P \times Q} \quad (4.3)$$

For a Gabor filter with  $m$  scales and  $n$  orientations, the mean  $\mu$  and standard deviation  $\sigma$  are used to create a *feature vector*  $f$  [3] of the form:

$$f = (\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \dots, \mu_{mn}, \sigma_{mn}).$$

Given an input image  $\mathbf{A}$  and target image  $\mathbf{B}$ , the Euclidean distance is used as a texture similarity measurement [50][29] is given by:

$$D(A, B) = \sum_m \sum_n d_{mn} \quad (4.4)$$

where,

$$d_{mn} = \sqrt{(\mu_{mn}^A - \mu_{mn}^B)^2 + (\sigma_{mn}^A - \sigma_{mn}^B)^2} \quad (4.5)$$

For the classification of texture types, 1-nearest neighbour is used.

## 4.2 Workflow Design

Given a single input image for texture segmentation or classification, a workflow is shown in Figure 4.1 and is further extended for the utilization of a Genetic Algorithm

as illustrated in Figure 4.2.

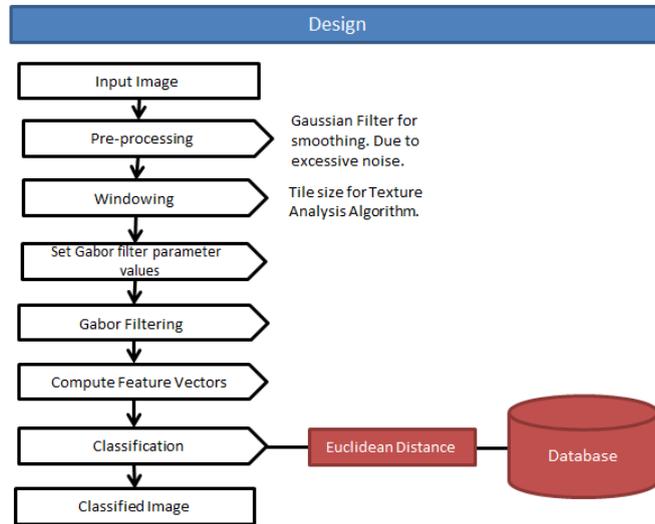


FIGURE 4.1: Workflow for the framework.

This workflow is used to create a framework that is intended to be extendable at the code level for the use of other texture analysis algorithms and distance metrics other than the Euclidean distance (which is used here). The aim is to implement the algorithms using a low-level language such as *c++* and use a scripting language (in this case Python) to call these algorithms as well as manage their respective meta-data. The database caters for a specific dataset only and Figure 4.3 illustrates these design choices. Metadata stores descriptions of the algorithms used, and for each algorithm- the type of feature vectors and parameter combinations used. The *Satellite Image Properties* file is used to store metadata about the satellite system and the data it provides. Finally a *colour map* file stores corresponding colours for each class of textures/concepts used. For a given image dataset, it is assumed that corresponding ground-truth images that comply with the colour map are available .

For storing data created and used by the texture analysis algorithms, the following is required:

- A *Class* represents the concepts that we need to identify, independent of the type of texture algorithm used and may contain subclasses (which are also classes ). There is also an option of having a single feature vector computed from a specific texture algorithm (in this case the Gabor filter) to represent all samples that fall immediately within that class.
- A *Sample* refers to individual texture samples and contain features for each of the texture algorithms used.

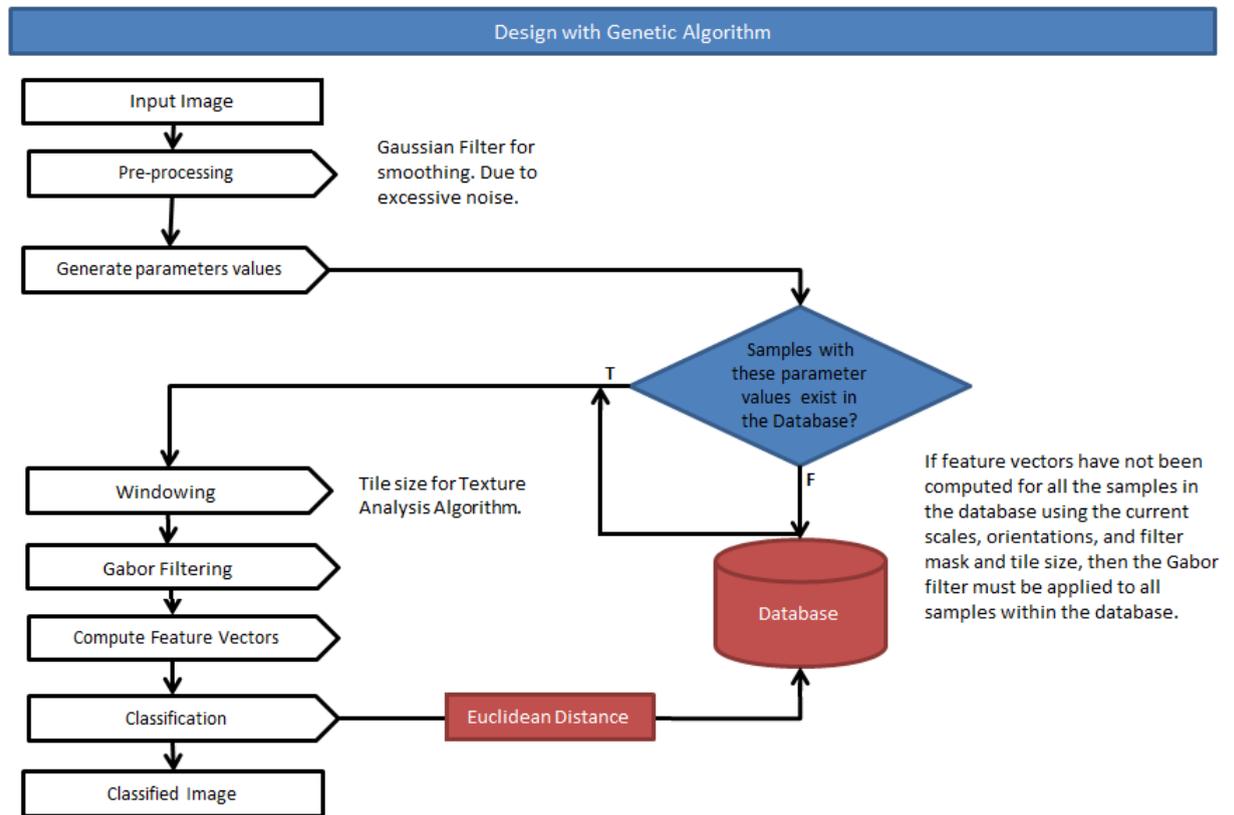


FIGURE 4.2: Workflow for the framework extended to incorporate the use of a Genetic Algorithm.

- A *Feature* contains feature vectors from the texture analysis algorithms. For each algorithm there are corresponding feature vectors composed of statistics for every parameter value combination used. For example, if the Gabor filter is used, there are corresponding feature vectors for all scale, orientation, filter mask, and tile sizes.

### 4.3 Initial Case Study and Difficulties:

The initial case study for this research was *Flood Detection from Satellite Imagery* with the area of study being Mozambique during December 2007 and February 2008 <sup>1</sup>. Requirements were to identify remote sensing satellites used for earth observation and to select those most suitable for this type for flood detection. This meant acquiring a time series of imagery containing rivers in Mozambique for change detection. The following constraints led to a new direction for the research:

<sup>1</sup>A project funded by the ICT4EO research group at CSIR, South Africa

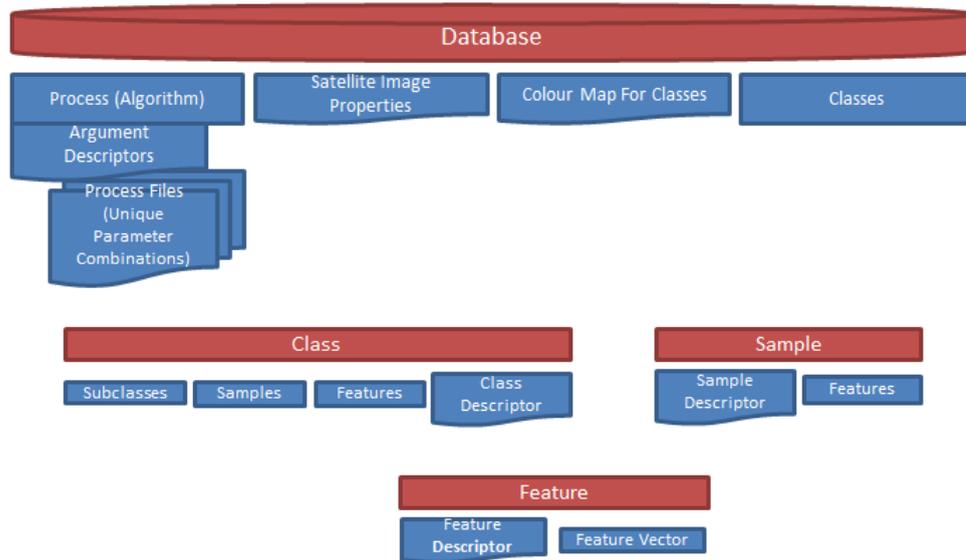


FIGURE 4.3: Database Design.

- While there are many satellites available for earth observation, there are very few publicly available datasets that are free.
- With high revisit times, numerous MODIS satellite imagery are available but their low spatial resolution is more suited for weather monitoring than the identification of rivers.
- LANDSAT imagery are also widely available but the limited number of imagery available for African countries and hence Mozambique was and still is a problem, especially for change detection applications.
- SAR imagery is expensive and difficult to acquire.
- The Hyperion EO-1 sensor provides hyperspectral imagery that have a 30 m spatial resolution and can easily be used for identifying rivers. Unfortunately most of the Hyperion EO-1 imagery provided by NASA has a considerable amount of cloud cover and had to be discarded and the change detection component of this research had to be abandoned.

The EO-1 Hyperion dataset of the Zambezi River does however contain images with differing water types and sizes, namely: clear, mildly turbid (muddy) and highly turbid (with surface vegetation).

## 4.4 The Dataset

The EO-1 Hyperion dataset consists of hyper-spectral imagery of the Zambezi River taken between 2006 and 2008 with dimensions:  $256 \times 3241$  or  $256 \times 3176$  pixels, a 30 m spatial resolution and a revisit time of 200 days. Due to subtle noise and sudden peaks in pixel intensities, preprocessing involved the use of a  $5 \times 5$  Gaussian filter (standard deviation of 1.0) to smoothen the images and retain the edges. The initial experiments consisted of the selection of the blue, green, red, and infrared bands for land cover classification. Through visual inspection these preliminary results were similar to that of those obtained using only a single NDVI composite band. Due to most satellite systems have differing instruments and actual pixel intensity values not being used, the NDVI image was selected as suitable used for dimension reduction with little to no loss in accuracy. The NDVI is a composite band that can be created from any satellite sensor that provides the red and near-infrared bands. The use of the NDVI band *may* serve to improve the applicability of the research to other satellite systems which have different sensors but similar spatial resolutions.

### 4.4.1 Factors that Influence Sample Selection

The manner in which texture samples are selected depends on the goals of the application and how the ground truth images are interpreted and created. This means that texture samples for classification are selected depending on the context in which they are used. Samples can have one dominant texture but may include traces of other textures. For example, in Figure 4.4, the area (**A**) contains soil (brown) and sparse vegetation (light green) textures, but in its entirety one can view (**A**) as a single texture representing an *arid climate* (contains both sparse vegetation and vast amounts of sand).

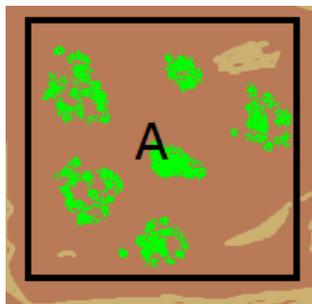


FIGURE 4.4: Sample Selection.

The classification accuracy of satellite imagery (using the same dataset) may yield different results when used on the same or different application domains. This means that results can not only vary due to parameter value selection but also due to ground truth creation and sample selection.

#### 4.4.2 Image Variation and Dataset Preparation

EO-1 Hyperion images from 5 regions of Zambezi river are used as shown in the Figure 4.5. Images from each region were used from different years, with some containing major changes in land cover. They contain water bodies and non-water bodies, each of different size and texture. To minimize the effects that atmospheric conditions have on the appearance of the land cover and hence the results obtained when applying the Gabor filter, images with any cloud cover and high levels of surface reflection due to sunlight were discarded.

- *Region 1* contains mostly clear and mildly turbid waters but also water based vegetation occupying neighbouring areas. Three images were available, one of the images was used for the selection of clear water, mildly turbid water and non-water bodies.
- *Region 2* contains some highly turbid water which consists of vegetation on the surface of the river and an intricate network of moist soil and clear water at the base of the image. Between the three images available, there are considerable changes in appearance and hence texture due to land cover change and sunlight. To provide a good representation of how the Gabor filter performs, only a single image was used for the selection of non-water and highly turbid water bodies.
- *Region 3* contains a mildly turbid water body with soil deposits. Only one image was available and hence only a single sample of turbid water was taken. Note the smooth vegetation textures located at the bottom half of the image. No samples of these textures are taken in order to determine which class these textures are assigned to during classification.
- *Region 4* contains two images from the same area as *Region 1* but the revisit point is slightly different. There is considerable change in land cover in the area occupied by the river. No samples were taken from these images.
- *Region 5*: only a single image was available and contains a large body of water with sunlight reflection. No samples were taken.

As illustrated in Figure 4.7, the Hyperion EO-1 images were windowed to create a dataset consisting of 135 sub-images with a dimension of  $256 \times 256$  pixels. As mentioned in Chapter 2, Section 2.2, it is possible to have more intra-class variation than inter-class variation, so instead of creating a single feature vector to represent a single class, multiple samples (hence multiple feature vectors) are used. To avoid experiential bias 50 of the sub-images are used to create a *Sample dataset* and are not used in the experiments



FIGURE 4.5: Images from Zambezi, Mozambique.

(except for a single sample of mildly turbid water (muddy)). The remaining **85** sub-images were then used to create a *Test dataset* for the experiments.

Figure 4.8 shows some of the land cover types under investigation. Three classes of textures are used for the experiments conducted in this research (discussed in detail in Section 5.1), namely:

- **Clear or Mildly Turbid Water:** consists of water bodies that are clear or those with soil deposits.

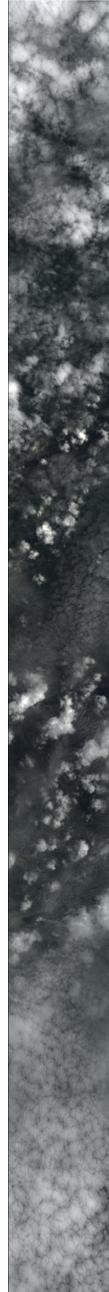


FIGURE 4.6: Images from Zambezi with cloud cover.

- **Highly Turbid Water:** consist of vegetation protruding from, or growing on the water surface. Even before post processing (smoothing), these highly turbid waters appear similar in structure (texture) to that of other vegetation types.
- **Non-water Bodies:** these include: soil, mud, rock and vegetation; with the latter being the most common land cover. These do not contain urban textures as none were available and may affect results in similar studies.

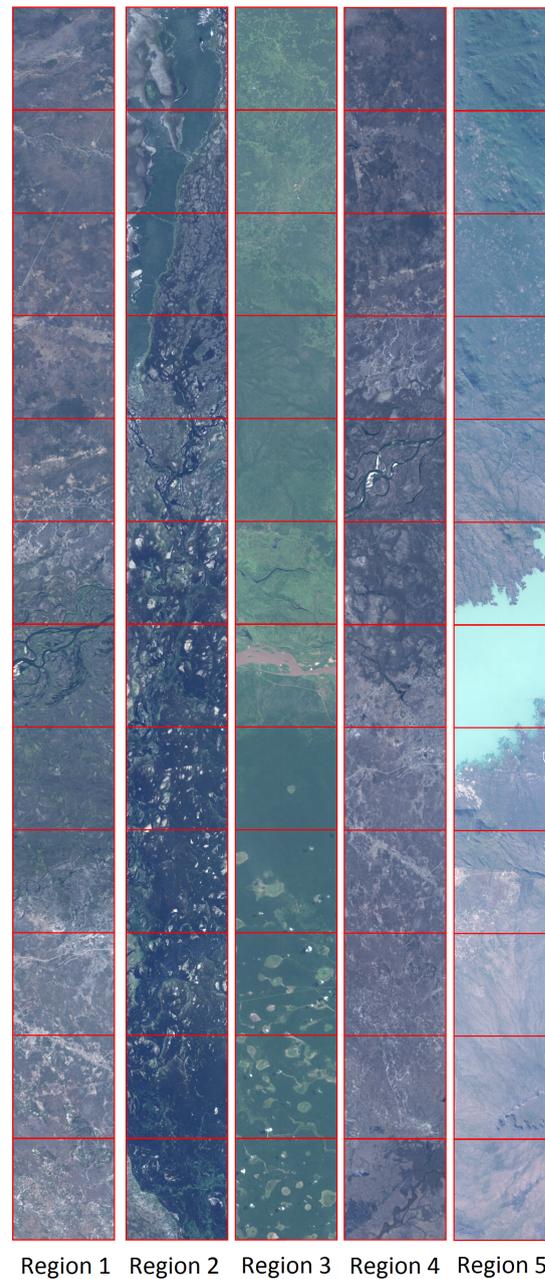


FIGURE 4.7: Creating Sub-images from the EO-1 Hyperion images.

## 4.5 Genetic Algorithm Design

For the heuristic approach suitable design choices were made to use the Genetic Algorithm effectively. *Bit strings* are used to represent chromosomes of the individuals with fixed lengths. The length has an upper limit defined by the user (upper values set for the number of scales, orientation, and filter mask and tile sizes respectively). For example, if the highest number of scales and orientations are 8 then the length is 4 (4 bits) and

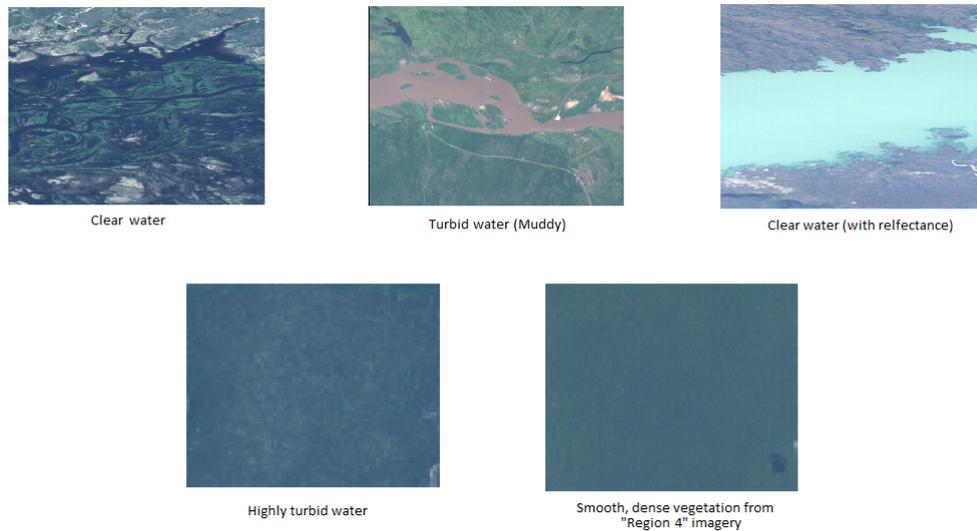


FIGURE 4.8: Land cover types.

if the maximum tile size is 31 then the length is 6 (6 bits). For the genetic operators, a *two point crossover* and *mutation* with a probability of  $0.2$  is used.

*Elitism* works well for high dimensional data and *Steady-State* for low dimensions [51]. For this research pure elitism is not used, but instead, the concept of elitism is used with a variation of Steady-State selection. The aim is to ensure that most of the individuals survive to the next generation but also allow a certain percentage of the best individuals to survive. Individuals are randomly selected as parents from the top 20 percent of the population. Two offspring are produced from a two point crossover and both then undergo *mutation* if necessary. Each of the chromosomes that represent the scales, orientations, mask size and tiles sizes respectively are treated independently of each other and have random chances of being selected for mutation.

With a fixed population size of 50 individuals, the GA is terminated when no considerable change over 10 consecutive generations is observed. Classification accuracy serves as the fitness function for the GA, with the aim of maximizing this value.

## 4.6 Summary

An experimental framework was designed to cater for the application of the Gabor filter on multi-band and multi-textured imagery. To meet the objectives of this research it had to allow for the variation of the Gabor filter parameters and the use of a Genetic Algorithm. The following Gabor filter parameters were then selected for investigation: the *scales* and *orientations*, *filter mask size* and the *tile size* for window segmentation.

Due to the large size of the Hyperion EO-1 images, 135 sub-images were created to form a Test dataset of 85 images and a Sample dataset of 50 images. Three texture classes were created to represent clear or mildly turbid water, highly turbid water and non-water bodies. Texture samples for the different classes were carefully selected to avoid experimental bias.

This framework, dataset and GA provides the basis for conducting the experiments described in the next chapter.

---

## Experiments and Results

---

This chapter provides a description of two experiments (scenarios) and the two approaches used for determining the best Gabor filter parameter values for the respective experiments. Sections 5.2 and 5.4 provide results for the GA and linear enumeration approaches for both experiment 1 and experiment 2, respectively, while Sections 5.3 and 5.5 provide summaries of these results.

For all results presented in this chapter, *Accuracy* refers to the classification accuracy in percentage. Filter mask size  $S$  refers to a filter of size  $S \times S$ , the same applies to the tile sizes ( $T$ ).

### 5.1 Experiments

The following experiments are conducted:

- **Experiment 1:** *Identification of Clear and Mildly Turbid Water Bodies.* The reason for adding *mildly turbid* waters such as muddy water, is that during flooding (especially flash floods) there are large amounts of sediment deposits in overflowing and fast moving rivers. Two classes of textures are used for this experiment: A single class for *clear or mildly turbid* water body textures, and one for *non-water* bodies (mostly vegetation).

- **Experiment 2:** *Identification of Clear or Mildly and Highly Turbid Water Bodies.* Highly turbid waters have vast amounts of protruding and/or surface vegetation. Three classes of textures are used for this experiment. The first class represents *clear or mildly turbid* water bodies, the second represents *highly turbid water* bodies and the last class represents *non-water* bodies.

For the approaches described below, for the Gabor filter (see Chapter 4, Section 4.1), the *lower* and *upper* centre frequencies are kept constant at 0.05 and 0.4, the mean and standard deviations are used for feature vector representation. For classification, 1-nearest neighbour with Euclidean distance is used.

### 5.1.1 A Heuristic Approach: Using a Genetic Algorithm

The first approach involves the use of GA to find the best solutions under parameter derived constraints to eliminate unsuitable or spurious parameter value combinations for the Gabor Filter, otherwise the GA would take considerable time to find the best solutions.

#### Parameter derived Constraints for the Genetic Algorithm:

- *Number of scales:* values between 3 and 8 (inclusive) are used to cover the ranges provided in the literature as shown in Table 2.1 from Chapter 2, Section 2.3.
- *Number of orientations:* values between 3 and 8 (inclusive) are selected for the same reason mentioned above.
- *Filter mask size* (restricted by tile size): at 3 x 3 to a third of the tile size.
- *Tile size* (restricted by texture sample size): 9 x 9 – 32 x 32, with odd and even sizes allowed.

Recall from Chapter 4, Section 4.5, the GA uses a population size of 50 individuals, a mutation probability of 0.2, and is terminated when no considerable change over 10 consecutive generations are observed. The *test* dataset of 85 images is randomly split into static *training* and *validation* subsets consisting of 42 and 43 images, respectively. To reiterate the 50 sub-images used from the original set of 135 sub-images in the creation of samples are not used in any of the experiments.

With these restrictions there are 6480 possible solutions with the Gabor filter being theoretically (GA is constrained) applied 550800 times to a dataset consisting of 85 images. For example, for a Gabor filter bank consisting of 7 scales and 7 orientations

(49 filters), for all filter mask and tile sizes (60 combinations), 249900 convolutions will be performed for the Test dataset.

### 5.1.2 Linear Enumeration Approach

For the second approach, the following parameters are investigated:

- *Number of scales*: 3, 4, 5, 6, 7.
- *Number of orientations*: 3, 5, 7.
- *Filter mask size* (restricted by tile size- at 3 x 3 to a third of the current tile size).
- *Tile size* (restricted by texture sample size- 9 x 9 - 32 x 32, with odd and even sizes allowed).

This provides 2700 parameter value combinations and results in a family/bank of Gabor filters being applied 229500 times to a dataset consisting of 85 images.

## 5.2 Experiment 1: Identification of Clear and Mildly Turbid Water Bodies

The first objective as stated in Section 5.1 is to conduct an experiment based on the identification of clear and mildly turbid water bodies.

### 5.2.1 Heuristic Approach: Results obtained from the Genetic Algorithm

Table 5.1 shows the results obtained for all sub-images in the training and validation subsets, while Table 5.2 *contrasts* the results obtained from the validation subset and test datasets. In Tables 5.1 and 5.2 the first 2 entries are the best solutions produced by the GA after multiple runs and the remaining entries show some of the best solutions encountered over different generations.

### 5.2.2 Results obtained through Linear Enumeration

For the evaluation of Gabor filter behavior using the enumeration approach the *Test* dataset is used. The following figures show the results obtained by using scale values:

TABLE 5.1: Experiment 1: Solutions Generated by the GA for the Training and Validation subsets.

Training Subset					Validation Subset				
Scales	Orientations	Mask Size	Tile Size	Accuracy	Scales	Orientations	Mask Size	Tile Size	Accuracy
4	3	3	21	<b>85.62801</b>	8	6	3	10	<b>81.04677</b>
4	3	3	22	<b>85.62212</b>	3	7	3	10	<b>81.02847</b>
5	3	3	22	85.424	5	6	3	10	81.0211
4	3	3	19	85.31415	7	6	3	10	81.02073
4	3	3	20	85.28216	4	6	3	10	81.01909
5	3	3	23	85.25583	6	8	3	10	80.99066
6	3	3	25	84.93375	6	6	3	10	80.98389
5	3	3	24	84.90422	4	8	3	10	80.98195
4	3	9	28	84.90039	7	5	3	10	80.97808
6	3	3	23	84.86984	6	7	3	10	80.97488
6	3	3	24	84.78518	6	5	3	10	80.97354
4	3	9	27	84.67899	5	7	3	10	80.97198
6	3	3	22	84.61927	4	7	3	10	80.96327
6	3	3	26	84.56834	8	5	3	10	80.96164
3	7	3	9	84.51009	7	7	3	10	80.96
7	3	3	24	84.5062	8	7	3	10	80.96
6	6	3	9	84.45404	5	5	3	10	80.94437
6	3	3	21	84.44489	8	4	3	10	80.903
5	6	3	10	84.4349	3	5	3	10	80.77589
4	5	7	23	84.43217	7	4	3	10	80.74679
5	5	3	9	84.42652	4	3	3	21	80.73313
4	6	3	9	84.42018	4	5	3	10	80.72707
7	3	3	22	84.41615	5	3	3	23	80.52952
7	3	3	26	84.41502	3	7	3	9	80.50637
3	7	3	10	84.41322	4	3	3	20	80.4916
8	3	3	24	84.411	4	6	3	9	80.37066
6	5	3	9	84.40612	5	5	3	9	80.33455
7	5	3	9	84.40323	5	6	3	9	80.32872
4	6	3	10	84.38922	8	6	3	9	80.31521
7	6	3	10	84.38914	6	5	3	9	80.30961

TABLE 5.2: Experiment 1: Solutions Generated by GA Validation subset and Test dataset.

Validation Subset				Test Dataset					
Scales	Orientations	Mask Size	Tile Size	Accuracy	Scales	Orientations	Mask Size	Tile Size	Accuracy
8	6	3	10	81.04677	5	3	3	23	82.86487
3	7	3	10	81.02847	4	3	3	20	82.8587
5	6	3	10	81.0211	4	3	3	22	82.84947
7	6	3	10	81.02073	5	6	3	10	82.70792
4	6	3	10	81.01909	4	3	3	19	82.7023
6	8	3	10	80.99066	3	7	3	10	82.70093
6	6	3	10	80.98389	8	6	3	10	82.68892
4	8	3	10	80.98195	5	3	3	22	82.68689
7	5	3	10	80.97808	7	6	3	10	82.68512
6	7	3	10	80.97488	4	6	3	10	82.68433
6	5	3	10	80.97354	6	8	3	10	82.66615
5	7	3	10	80.97198	4	8	3	10	82.66103
4	7	3	10	80.96327	4	7	3	10	82.65466
8	5	3	10	80.96164	7	5	3	10	82.6521
7	7	3	10	80.96	5	7	3	10	82.65199
8	7	3	10	80.96	6	6	3	10	82.64958
5	5	3	10	80.94437	6	5	3	10	82.64883
8	4	3	10	80.903	6	7	3	10	82.64548
3	5	3	10	80.77589	8	5	3	10	82.6448
7	4	3	10	80.74679	8	7	3	10	82.63994
4	3	3	21	80.73313	5	5	3	10	82.63784
4	5	3	10	80.72707	7	7	3	10	82.63418
5	3	3	23	80.52952	8	4	3	10	82.60232
3	7	3	9	80.50637	6	3	3	23	82.52918
4	3	3	20	80.4916	3	5	3	10	82.505
4	6	3	9	80.37066	3	7	3	9	82.48468
5	5	3	9	80.33455	7	4	3	10	82.42665
5	6	3	9	80.32872	4	5	3	10	82.42189
8	6	3	9	80.31521	4	6	3	9	82.3716
6	5	3	9	80.30961	6	6	3	9	82.35692

3, 4, 5, 6, and 7 but allowing the filter mask size and tile size to vary. Within each of the figures, three graphs also depict the effects of orientation values: 3, 5, and 7.

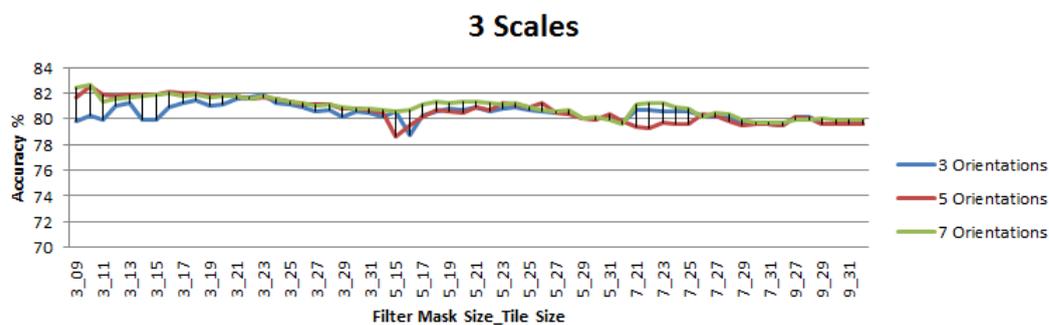


FIGURE 5.1: Identification of Clear and Mildly Turbid Water Bodies. Scales: 3.

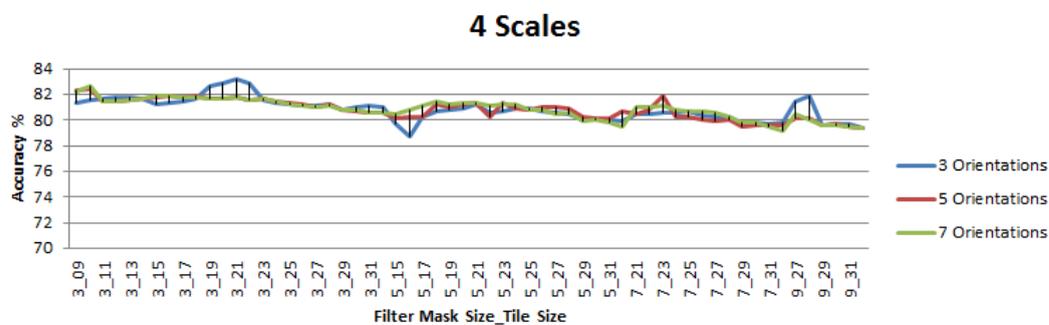


FIGURE 5.2: Identification of Clear and Mildly Turbid Water Bodies. Scales: 4.

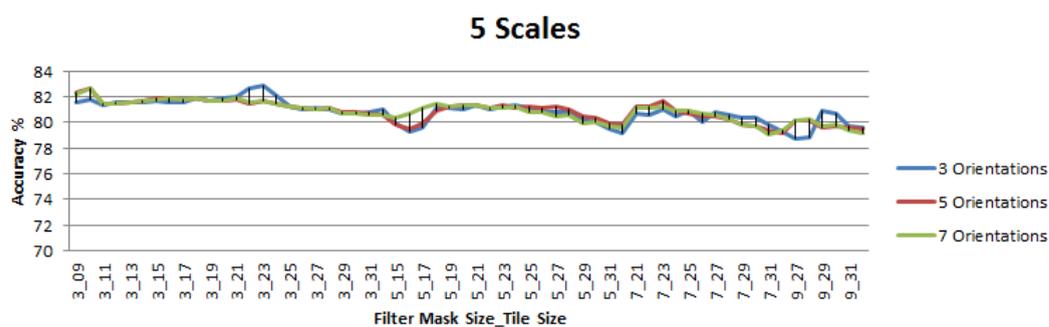


FIGURE 5.3: Identification of Clear and Mildly Turbid Water Bodies. Scales: 5.

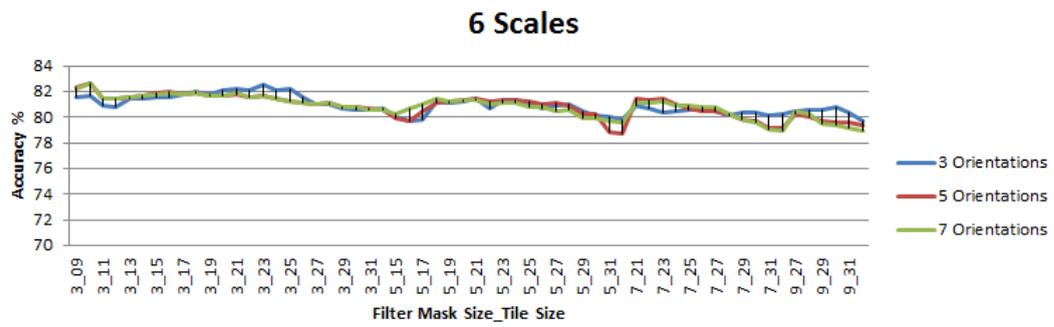


FIGURE 5.4: Identification of Clear and Mildly Turbid Water Bodies. Scales: 6.

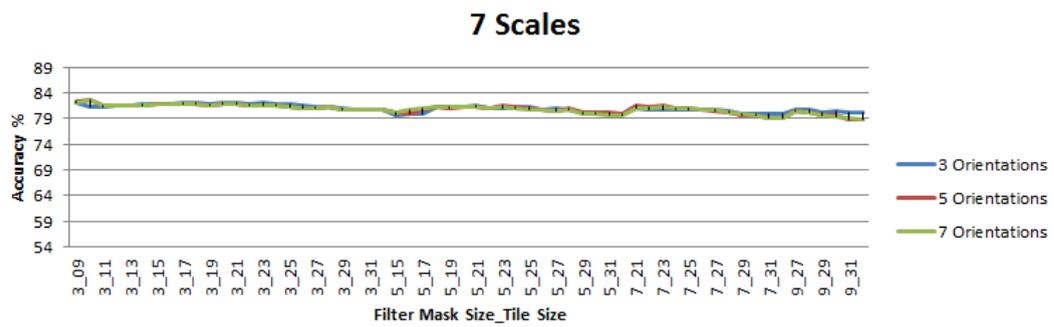


FIGURE 5.5: Identification of Clear and Mildly Turbid Water Bodies. Scales: 7.

The next task is to identify how well the parameter value combinations worked for all of the images in each “Region” (see Chapter 4, Figure 4.5). Each figure depicts the effects of the filter mask and tile sizes, while maintaining a fixed scale and orientation value.

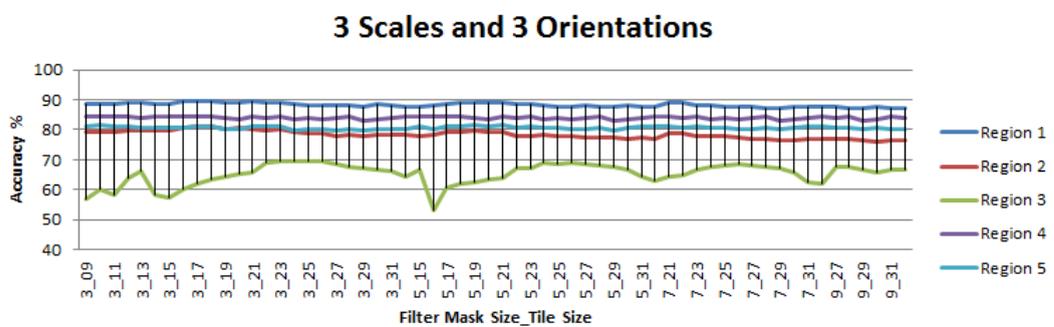


FIGURE 5.6: The effects of filter mask and tile sizes on images taken from different regions. Scales: 3 and Orientations: 3.

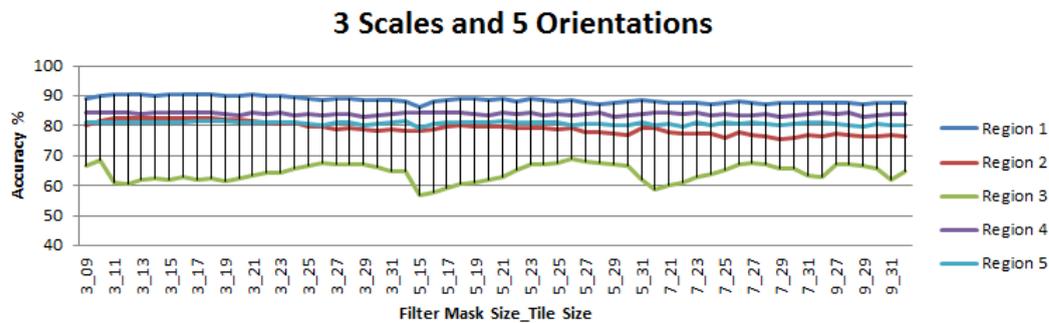


FIGURE 5.7: The effects of filter mask and tile sizes on images taken from different regions. Scales: 3 and Orientations: 5.

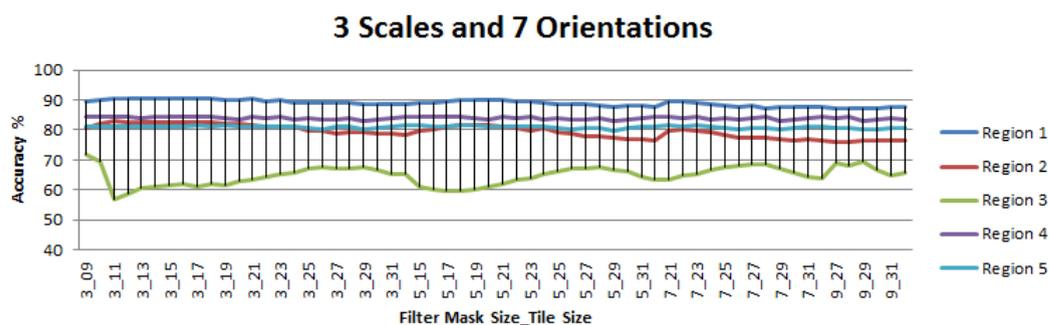


FIGURE 5.8: The effects of filter mask and tile sizes on images taken from different regions. Scales: 3 and Orientations: 7.

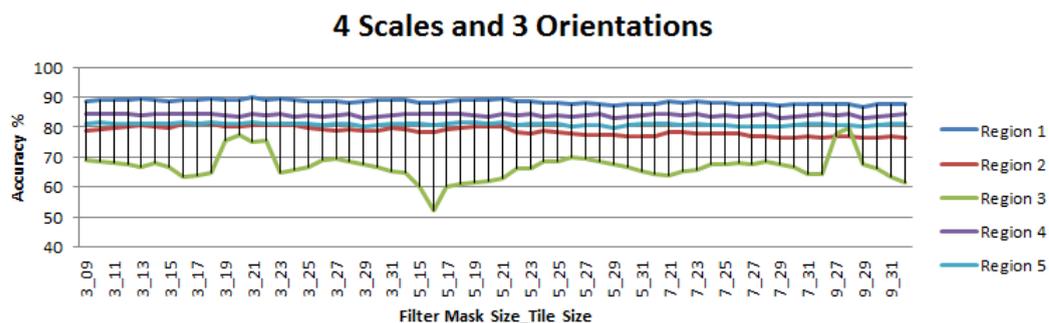


FIGURE 5.9: The effects of filter mask and tile sizes on images taken from different regions. Scales: 4 and Orientations: 3.

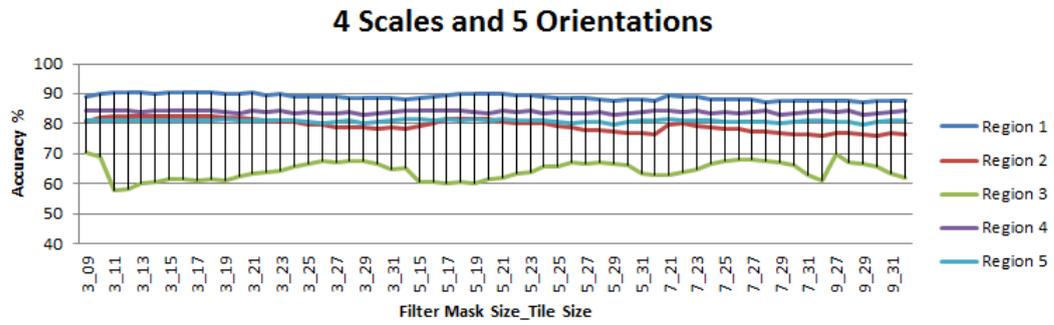


FIGURE 5.10: The effects of filter mask and tile sizes on images taken from different regions. Scales: 4 and Orientations: 5.

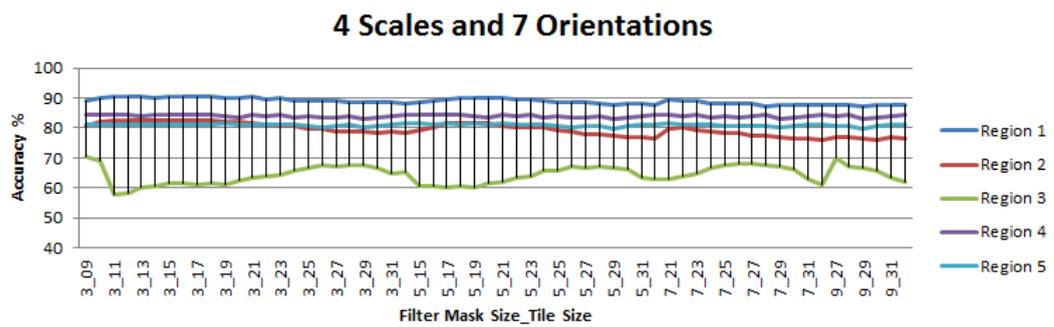


FIGURE 5.11: The effects of filter mask and tile sizes on images taken from different regions. Scales: 4 and Orientations: 7.

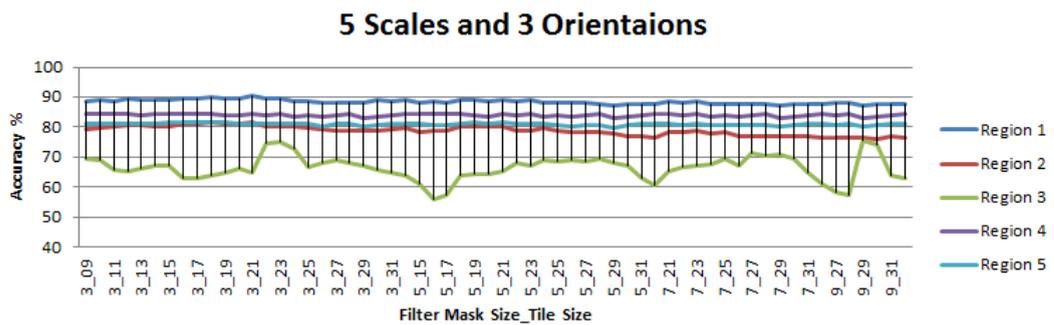


FIGURE 5.12: The effects of filter mask and tile sizes on images taken from different regions. Scales: 5 and Orientations: 3.

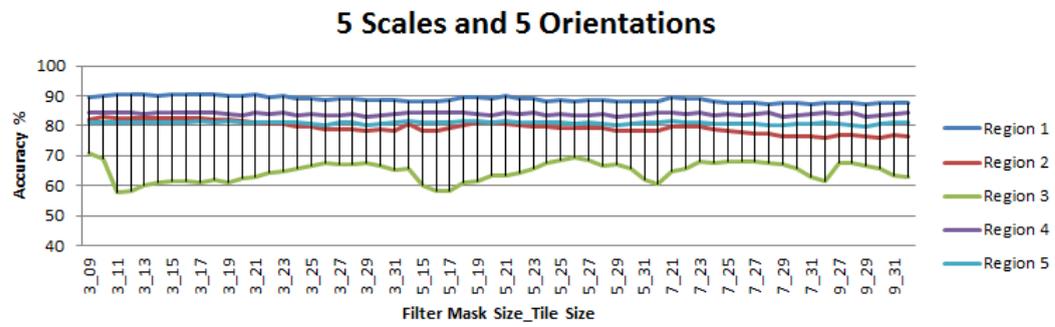


FIGURE 5.13: The effects of filter mask and tile sizes on images taken from different regions. Scales: 5 and Orientations: 5.

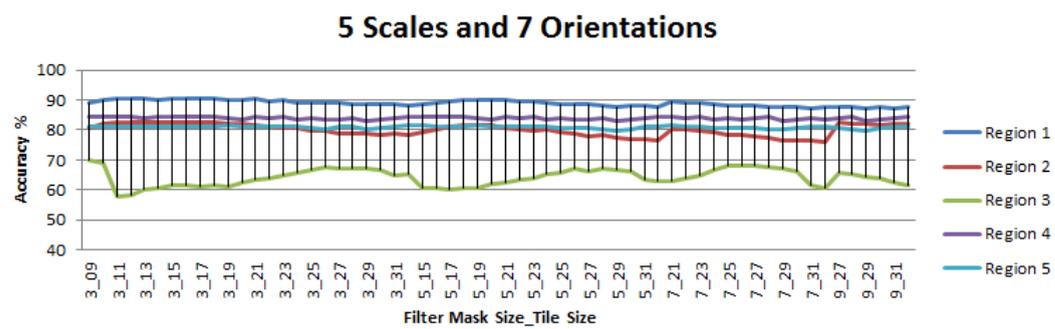


FIGURE 5.14: The effects of filter mask and tile sizes on images taken from different regions. Scales: 5 and Orientations: 7.

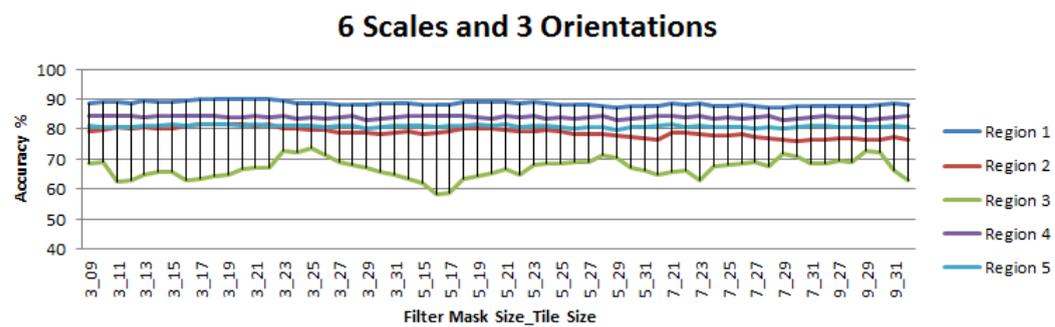


FIGURE 5.15: The effects of filter mask and tile sizes on images taken from different regions. Scales: 6 and Orientations: 3.

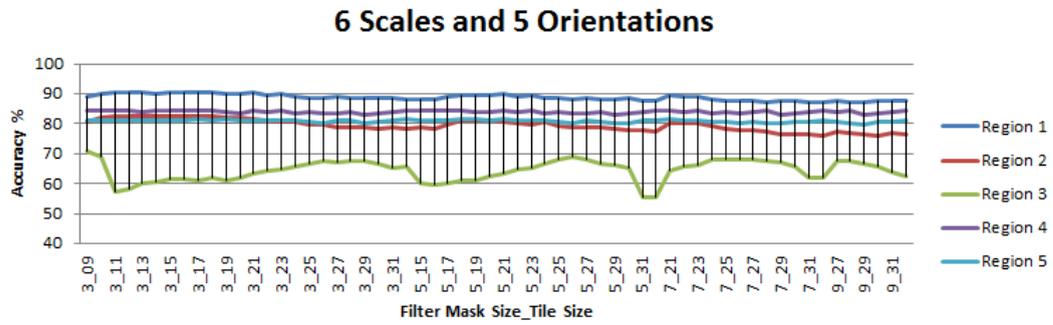


FIGURE 5.16: The effects of filter mask and tile sizes on images taken from different regions. Scales: 6 and Orientations: 5.

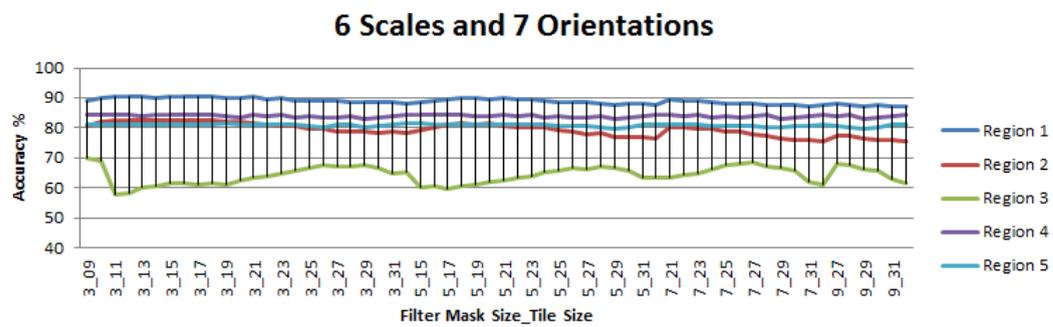


FIGURE 5.17: The effects of filter mask and tile sizes on images taken from different regions. Scales: 6 and Orientations: 7.

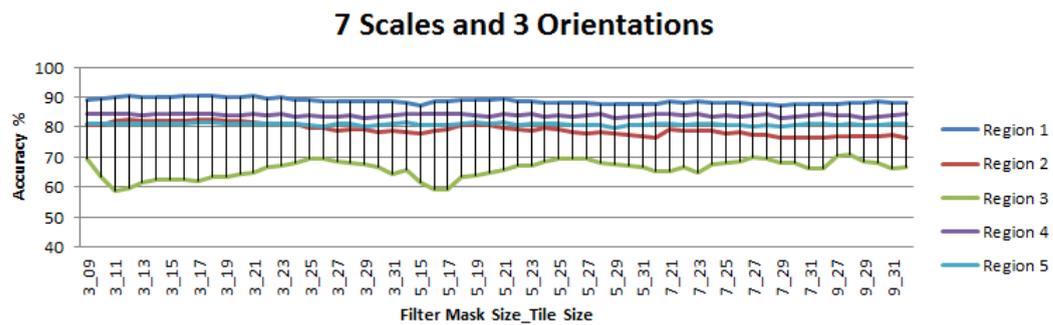


FIGURE 5.18: The effects of filter mask and tile sizes on images taken from different regions. Scales: 7 and Orientations: 3.

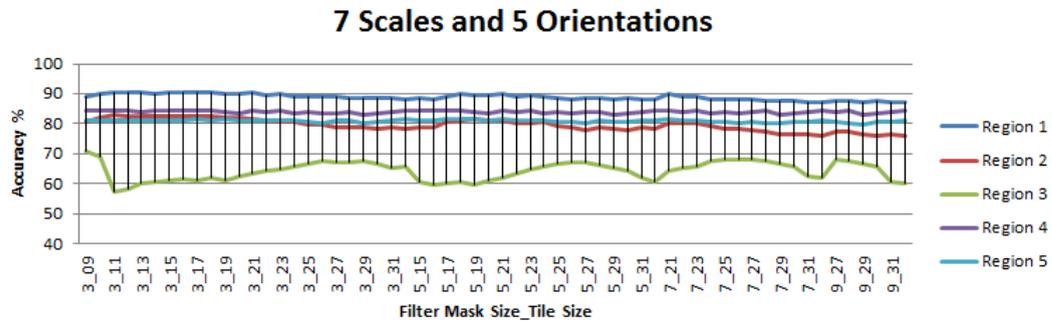


FIGURE 5.19: The effects of filter mask and tile sizes on images taken from different regions. Scales: 7 and Orientations: 5.

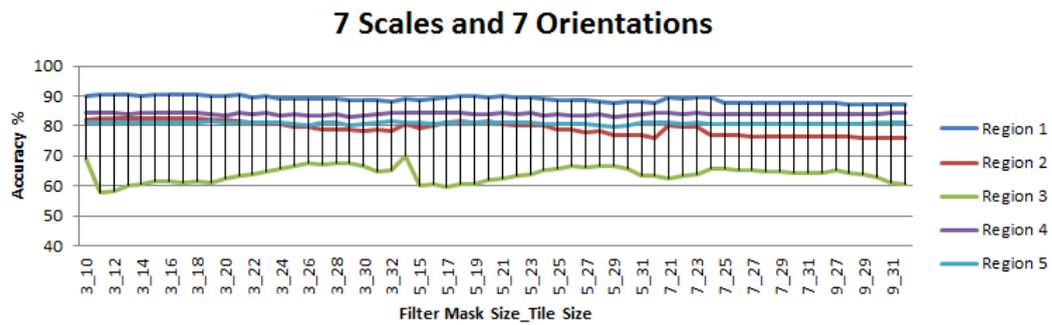


FIGURE 5.20: The effects of filter mask and tile sizes on images taken from different regions. Scales: 7 and Orientations: 7.

## 5.3 Experiment 1: Summary

### 5.3.1 The Genetic Algorithm

The Genetic Algorithm was applied to the training and validation subsets and the test dataset. The best solutions from Tables 5.1 and 5.2 are summarized below:

- *Training subset:* 4 scales, 3 orientations, mask size 3 x 3, and tile size 21 x 21 pixels.
- *Validation subset:* 8 scales, 6 orientations, mask size 3 x 3, and tile size 10 x 10 pixels.
- *Test dataset:* 5 scales, 3 orientations, mask size 3 x 3, and tile size 23 x 23 pixels.

The best scale and orientation values vary across the datasets. However, a filter mask size of 3 x 3 pixels is consistent for all three data datasets. Table 5.2 does however indicate that most solutions favoured tile sizes of 10 x 10.

### 5.3.2 Linear Enumeration

#### Results obtained from the Test dataset

From Chapter 4, the graphs in Figures 5.1 to 5.5 depict very little change over scales: 3, 4, 5, 6 and 7 but also individual orientations: 3, 5 and 7. There is a slight increase in classification accuracy when more scales are chosen. For 7 scales (Figure 5.5), the difference in classification accuracy over *orientations*: 3, 5 and 7 is almost indistinguishable.

#### Analysis of Region images

For selected scale values, Figures 5.6 to 5.20 illustrate the effects of the performance of the Gabor filter using various filter mask and tile size combinations on the different “Region” images. When larger filter mask and tile size combinations are used a slight drop in classification accuracy is observed. For individual regions similar behavior is observed for all the scale and orientation value combinations.

## 5.4 Experiment 2: Identification of Clear, Mildly and Highly Turbid Water Bodies

The next objective as stated in Section 5.1 is to conduct an experiment based on the identification of clear, mildly and highly turbid water bodies.

### 5.4.1 Heuristic Approach: Results obtained from the Genetic Algorithm

As with the previous experiment a Genetic Algorithm was first applied. These are the averaged results for all sub-images, and the first 3 entries in Tables 5.3 and 5.4 are the best solutions generated by the GA over multiple runs. The remaining entries are some of the best solutions generated after different generations. Table 5.3 provides the results obtained from the training and validation subsets and Table 5.4 contrasts the results obtained from the validation subset and test dataset.

### 5.4.2 Results obtained through Linear Enumeration

The following figures show the results obtained by using scale values: 3, 4, 5, 6, and 7 but allowing the filter mask size and tile size to vary. Within each of the figures three graphs depict the effects of orientation values: 3, 5, and 7 as well.

TABLE 5.3: Experiment 2: Solutions Generated by the GA for the Training and Validation subsets.

		Training Subset			Validation Subset					
Scales	Orientations	Mask Size	Tile Size	Accuracy	Scales	Orientations	Mask Size	Tile Size	Accuracy	
7	6	3	9	72.57716	7	6	3	9	74.03906	
7	7	3	9	72.34887	6	7	3	9	73.98600	
6	7	3	9	72.34144	7	7	3	9	73.97794	
6	8	3	9	72.29417	6	8	3	9	73.97472	
4	7	3	9	72.20201	8	7	3	9	73.94619	
8	7	3	9	72.17456	8	6	3	9	73.93304	
8	6	3	9	72.11105	4	7	3	9	73.91623	
4	8	3	9	72.02650	5	7	3	9	73.87408	
5	7	3	9	71.99850	4	8	3	9	73.85211	
5	6	3	9	71.94121	5	6	3	9	73.83695	
6	6	3	9	71.80203	6	6	3	9	73.80626	
3	5	3	9	71.75712	3	5	3	9	73.77162	
4	6	3	9	71.72405	4	6	3	9	73.77162	
4	3	3	9	71.44551	7	6	3	10	73.46995	
3	7	3	9	71.41079	3	5	3	10	73.39959	
8	5	3	9	71.28681	3	7	3	9	73.39739	
5	5	3	9	71.23739	6	6	3	10	73.33391	
8	3	3	9	71.14403	5	7	3	10	73.31177	
7	5	3	9	71.07043	6	7	3	10	73.30180	
7	3	3	27	70.95535	8	5	3	9	73.29192	
6	5	3	9	70.74383	7	7	3	10	73.25488	
7	3	3	9	70.72370	6	8	3	10	73.21975	
7	6	3	10	70.69672	8	7	3	10	73.20587	
6	6	3	10	70.60632	4	8	3	10	73.20111	
4	7	3	10	70.52015	4	7	3	10	73.18883	
6	7	3	10	70.51584	8	6	3	10	73.15002	
8	7	3	10	70.50213	8	4	3	10	73.14842	
7	7	3	10	70.46396	4	6	3	10	73.10455	
5	7	3	10	70.40952	8	3	3	9	73.09541	
6	8	3	10	70.40826	3	7	3	10	73.08919	
8	3	3	27	70.39754	5	6	3	10	73.08777	
4	5	7	31	70.38570	7	5	3	9	73.08175	

TABLE 5.4: Experiment 2: Solutions Generated by GA for the Validation and Test dataset.

Validation Subset						Test Dataset					
Scales	Orientations	Mask Size	Tile Size	Accuracy		Scales	Orientations	Mask Size	Tile Size	Accuracy	
7	6	3	9	<b>74.03906</b>		7	6	3	9	<b>73.31671</b>	
6	7	3	9	<b>73.98600</b>		6	7	3	9	<b>73.17339</b>	
7	7	3	9	<b>73.97794</b>		7	7	3	9	<b>73.17299</b>	
6	8	3	9	73.97472		6	8	3	9	73.14433	
8	7	3	9	73.94619		8	7	3	9	73.07080	
8	6	3	9	73.93304		4	7	3	9	73.06920	
4	7	3	9	73.91623		8	6	3	9	73.03276	
5	7	3	9	73.87408		4	8	3	9	72.95005	
4	8	3	9	73.85211		5	7	3	9	72.94732	
5	6	3	9	73.83695		5	6	3	9	72.90023	
6	6	3	9	73.80626		6	6	3	9	72.81594	
3	5	3	9	73.77162		3	5	3	9	72.77622	
4	6	3	9	73.77162		4	6	3	9	72.75988	
7	6	3	10	73.46995		3	7	3	9	72.41578	
3	5	3	10	73.39959		8	5	3	9	72.30116	
3	7	3	9	73.39739		5	5	3	9	72.16845	
6	6	3	10	73.33391		8	3	3	9	72.13120	
5	7	3	10	73.31177		7	6	3	10	72.09965	
6	7	3	10	73.30180		7	5	3	9	72.08792	
8	5	3	9	73.29192		4	3	3	9	71.98692	
7	7	3	10	73.25488		6	6	3	10	71.98616	
6	8	3	10	73.21975		6	7	3	10	71.92521	
8	7	3	10	73.20587		5	7	3	10	71.87772	
4	8	3	10	73.20111		7	7	3	10	71.87584	
4	7	3	10	73.18883		4	7	3	10	71.87019	
8	6	3	10	73.15002		8	7	3	10	71.86991	
8	4	3	10	73.14842		6	8	3	10	71.83055	
4	6	3	10	73.10455		6	5	3	9	71.81880	
8	3	3	9	73.09541		3	5	3	10	71.78864	
3	7	3	10	73.08919		4	8	3	10	71.75858	
5	6	3	10	73.08777		8	6	3	10	71.73268	
7	5	3	9	73.08175		3	7	3	10	71.72244	

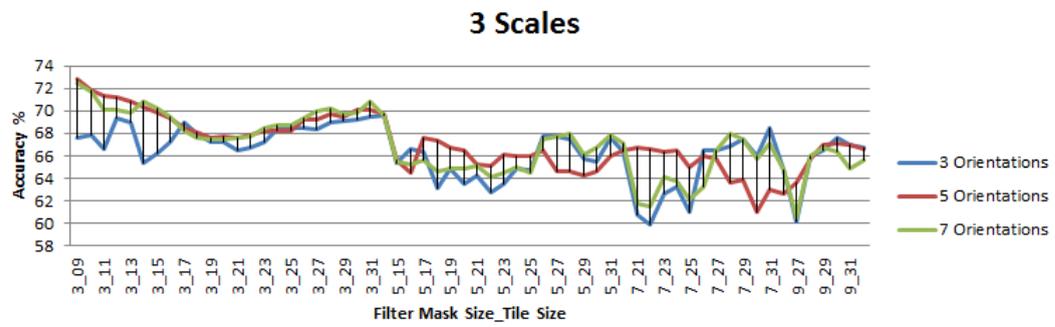


FIGURE 5.21: Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 3.

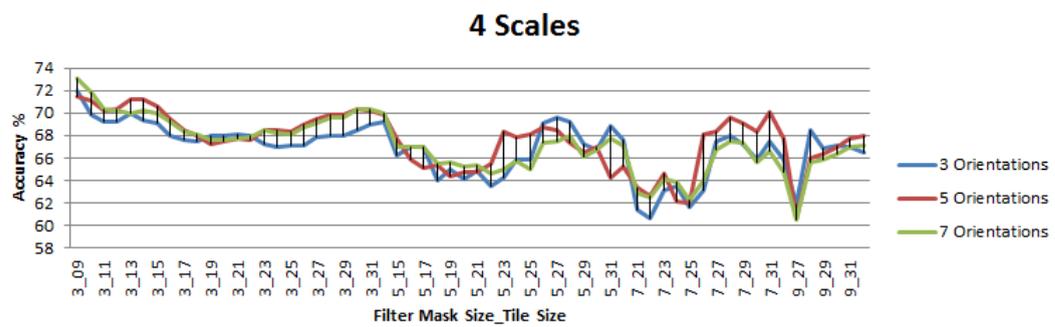


FIGURE 5.22: Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 4.

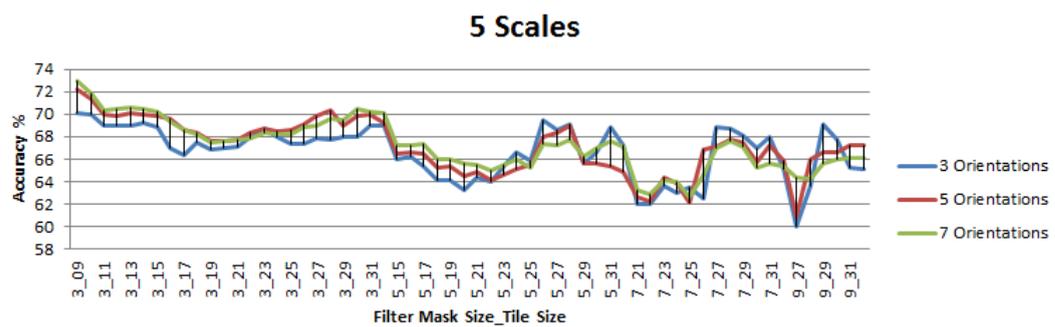


FIGURE 5.23: Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 5.

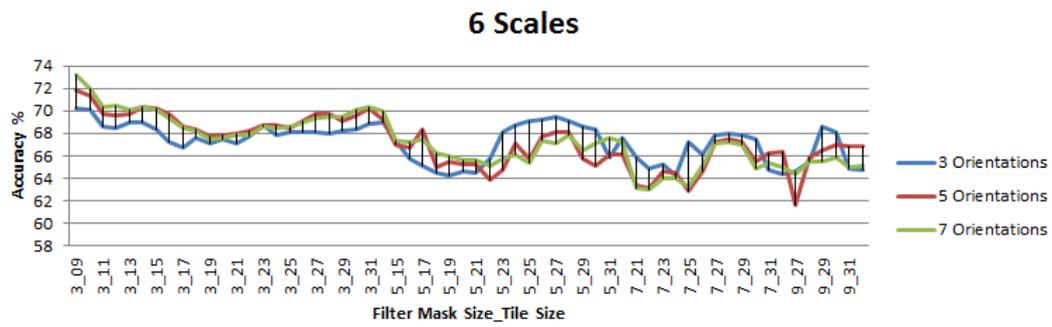


FIGURE 5.24: Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 6.

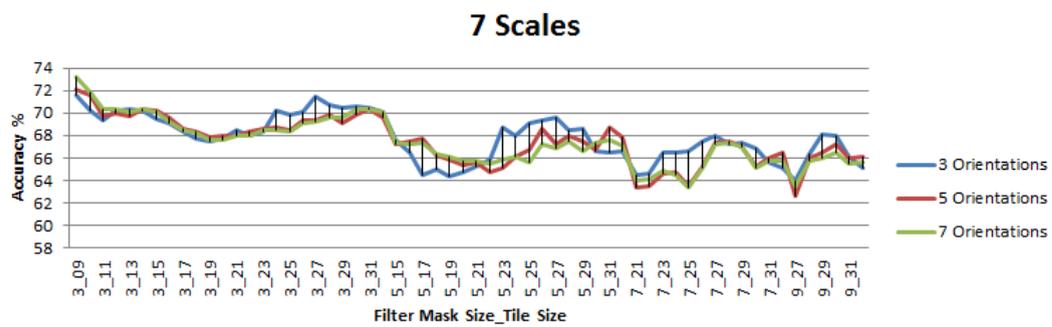


FIGURE 5.25: Identification of Clear, Mildly and Highly Turbid Water Bodies. Scales: 7.

Again, the next task was to identify how well the parameter value combinations worked for all of the images in each “Region” (see Chapter 4, Figure 4.5). Each figure depicts the effects of the filter mask and tile sizes while maintaining a fixed scale and orientation value.

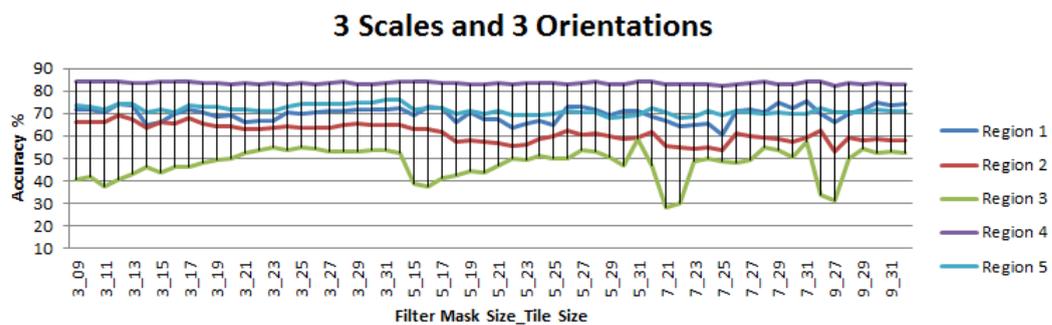


FIGURE 5.26: The effects of filter mask and tile sizes on images taken from different regions. Scales: 3 and Orientations: 3.

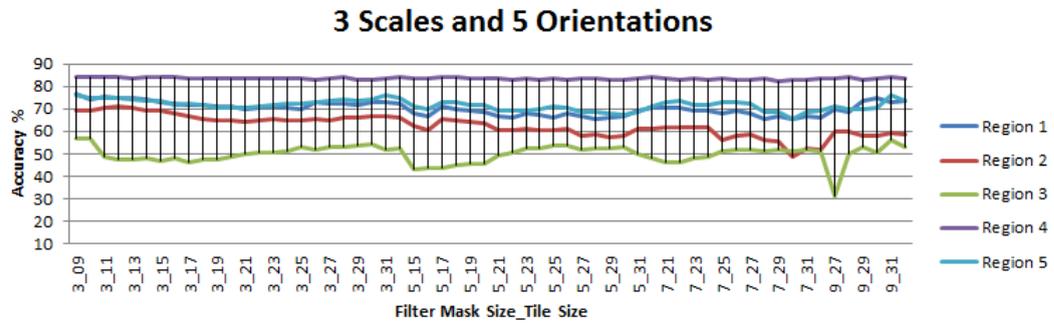


FIGURE 5.27: The effects of filter mask and tile sizes on images taken from different regions. Scales: 3 and Orientations: 5.

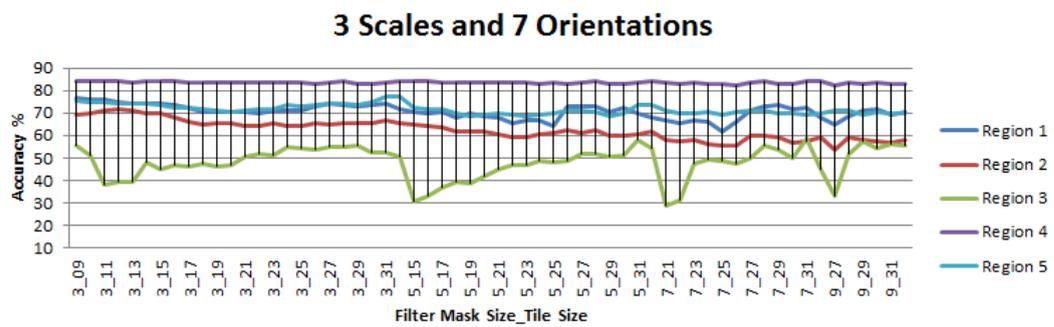


FIGURE 5.28: The effects of filter mask and tile sizes on images taken from different regions. Scales: 3 and Orientations: 7.

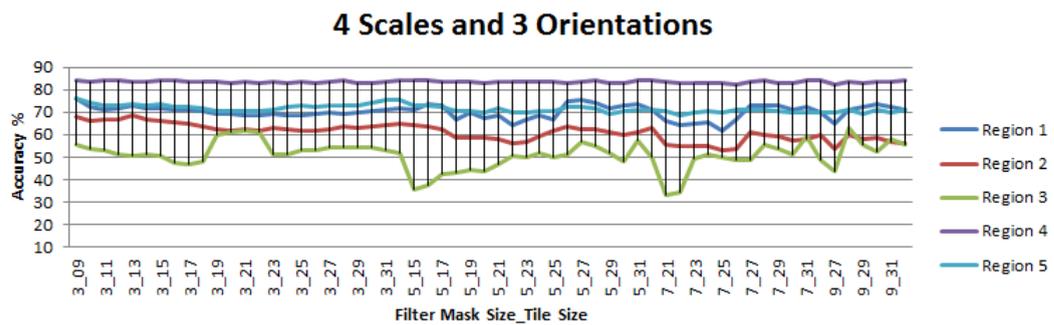


FIGURE 5.29: The effects of filter mask and tile sizes on images taken from different regions. Scales: 4 and Orientations: 3.

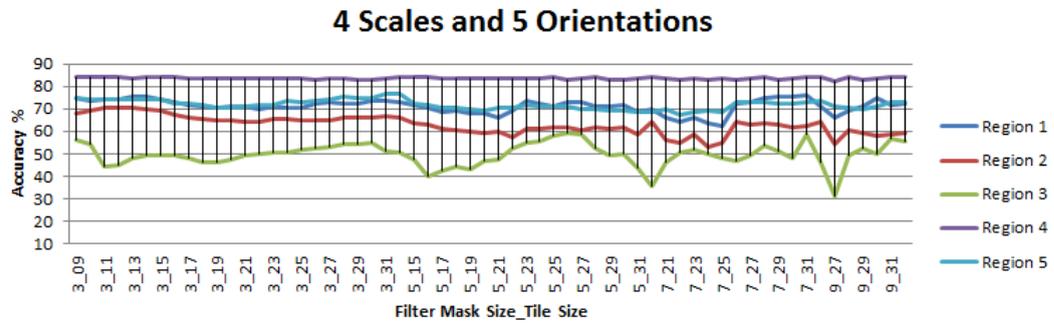


FIGURE 5.30: The effects of filter mask and tile sizes on images taken from different regions. Scales: 4 and Orientations: 5.

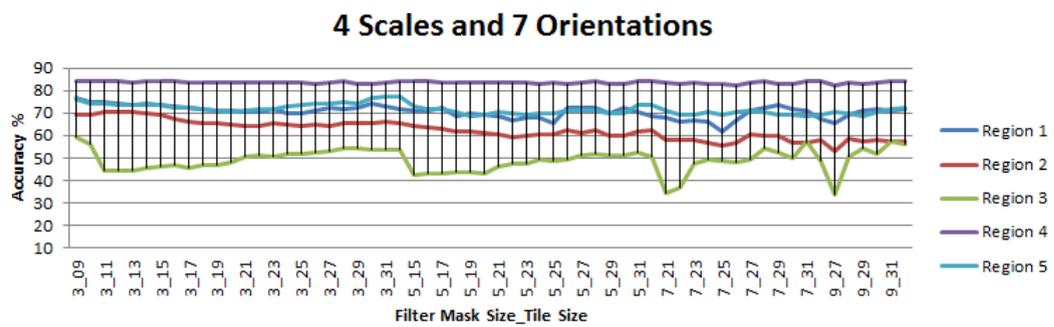


FIGURE 5.31: The effects of filter mask and tile sizes on images taken from different regions. Scales: 4 and Orientations: 7.

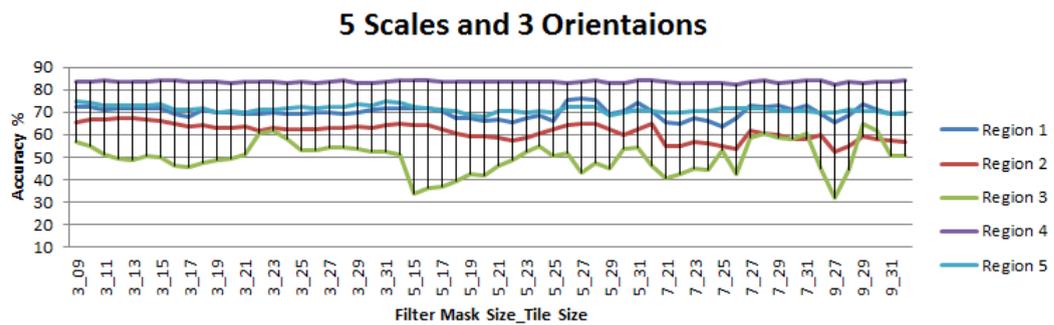


FIGURE 5.32: The effects of filter mask and tile sizes on images taken from different regions. Scales: 5 and Orientations: 3.

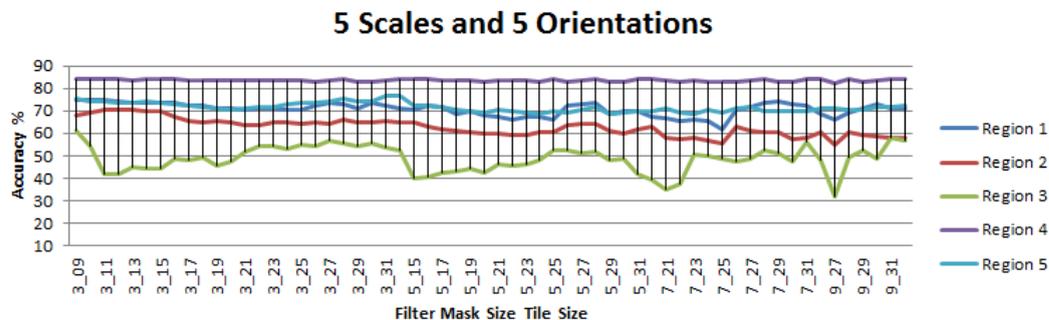


FIGURE 5.33: The effects of filter mask and tile sizes on images taken from different regions. Scales: 5 and Orientations: 5.

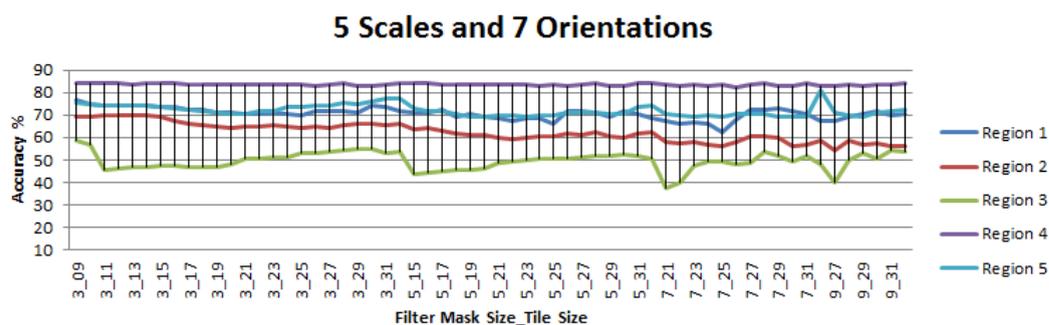


FIGURE 5.34: The effects of filter mask and tile sizes on images taken from different regions. Scales: 5 and Orientations: 7.

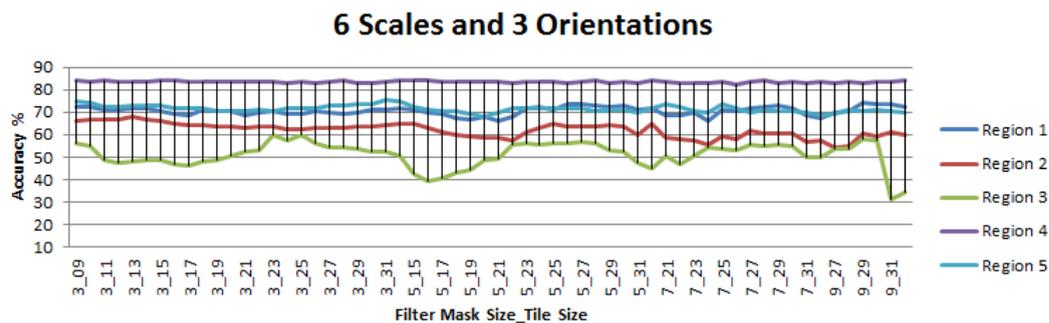


FIGURE 5.35: The effects of filter mask and tile sizes on images taken from different regions. Scales: 6 and Orientations: 3.

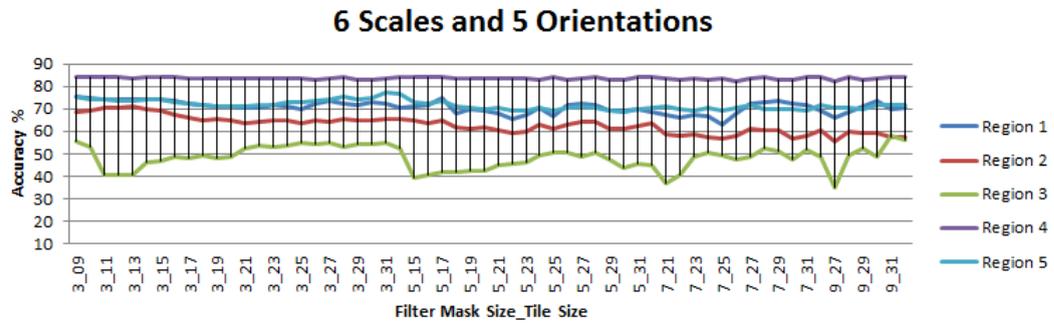


FIGURE 5.36: The effects of filter mask and tile sizes on images taken from different regions. Scales: 6 and Orientations: 5.

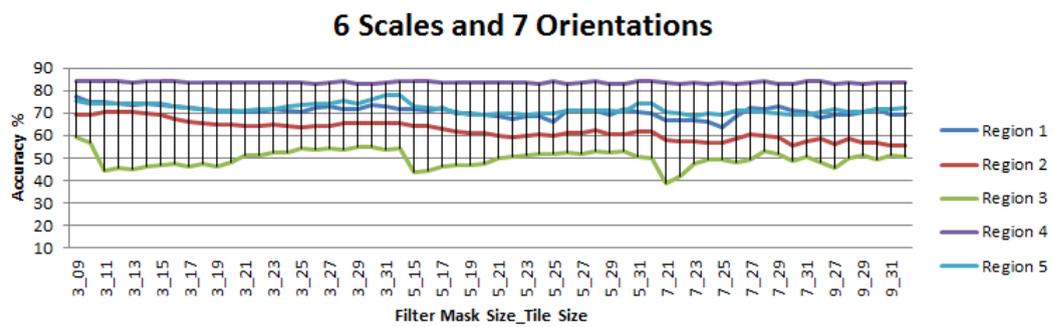


FIGURE 5.37: The effects of filter mask and tile sizes on images taken from different regions. Scales: 6 and Orientations: 7.

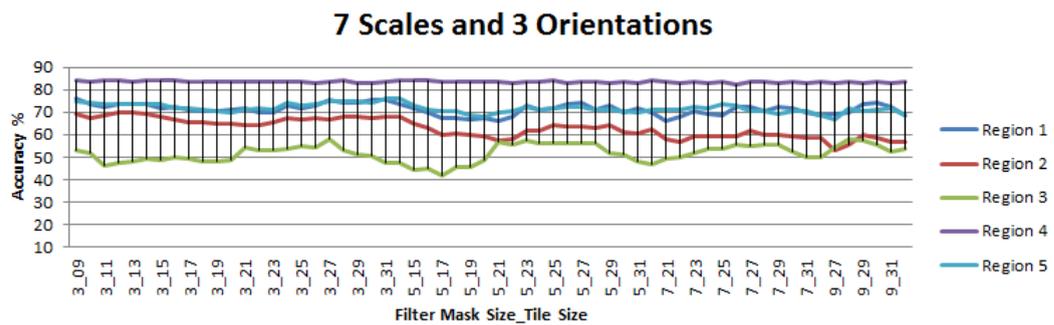


FIGURE 5.38: The effects of filter mask and tile sizes on images taken from different regions. Scales: 7 and Orientations: 3.

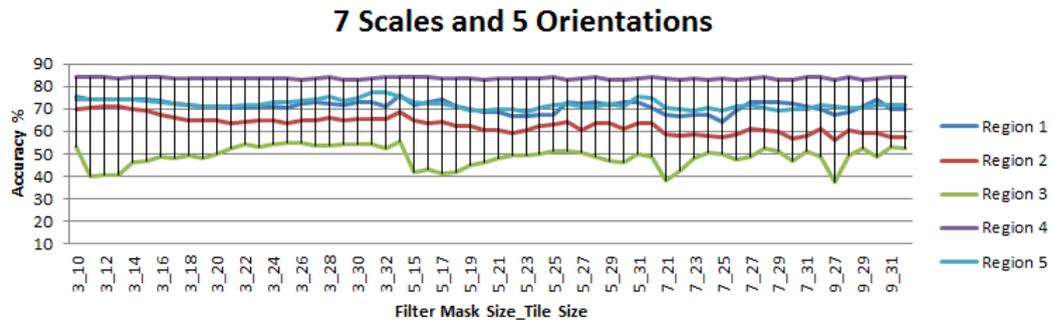


FIGURE 5.39: The effects of filter mask and tile sizes on images taken from different regions. Scales: 7 and Orientations: 5.

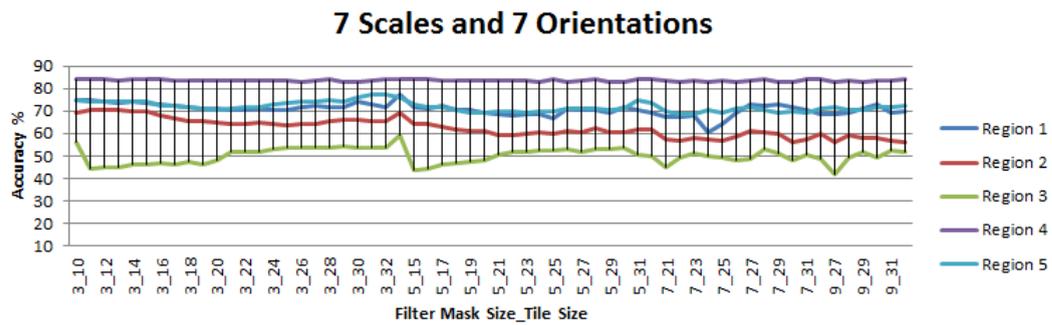


FIGURE 5.40: The effects of filter mask and tile sizes on images taken from different regions. Scales: 7 and Orientations: 7.

## 5.5 Experiment 2: Summary

### 5.5.1 The Genetic Algorithm

The best solutions from Tables 5.3 and 5.4 are summarized below:

- 7 scales, 7 orientations, with mask size 3 x 3 and tile size 9 x 9.
- 6 scales, 7 orientations, with mask size 3 x 3 and tile size 9 x 9.
- 7 scales, 6 orientations, with mask size 3 x 3 and tile size 9 x 9.

The best values for the scales and orientations for the training, validation and test datasets are in fact more consistent for this experiment, with the most common parameter values being the filter mask and tile size of 3 x 3 and 9 x 9, respectively.

### 5.5.2 Linear Enumeration

#### Results obtained from the Test dataset

The addition of an extra class of textures that represent the highly turbid waters resulted in a drop in classification accuracy due to the similarity of the highly turbid waters to other vegetation types as shown in Figures 5.21 to 5.25. For all graphs within these figures the results show more erratic behaviour than those of experiment 1. For each scale value investigated, this similarity between textures also affects the accuracy for orientation selectivity as well and a noticeable difference between orientations: 3, 5 and 7 is observed.

#### Region images

For selected scale values, the Figures 5.26 to 5.40 illustrate the effects of the performance of the Gabor filter using various filter mask and tile size combinations on the different “Region” images. The graphs depict similar behavior for the “Region” images for all the scale and orientation values just like experiment 1 but a drop in classification accuracy is also apparent.

## 5.6 Summary

Two experiments were conducted to determine the effectiveness of Gabor filter for the identification of clear, mildly and highly turbid water bodies. For each of the experiments two approaches were used to determine optimum Gabor filter parameters values: the first is a linear enumeration approach and the other a heuristic approach using a GA. By taking 8 – 9 generations to find optimal solutions, the constrained GA is good for quickly finding optimal values for parameters in large solutions spaces and datasets, but the linear enumeration approach is better for identifying trends by varying different parameter values.

For the enumeration approach, the behavior of the Gabor filter on the various Hyperion EO-1 “Region” images shows that each image can have its own optimum Gabor filter parameter values. This implies that the optimal Gabor filter parameter values for a given dataset can be influenced by the bias introduced through the addition of more images that favour specific parameter values.

---

### Analysis and Discussion

---

This chapter provides an analysis of the results obtained from the experiments. Section 6.1 provides an analysis of varying the number of scales and orientations, while Section 6.2 analyses the effects of varying the filter mask and tile sizes. Section 6.3 provides some insight into the behaviour of the Gabor filter when attempting to classify some of the images used in this research. Most importantly Section 6.4 evaluates the suitability of a GA for the purpose of Gabor filter parameter value optimization.

#### **6.1 Effects of Scales and Orientations**

Figures 6.1 and 6.2 show that there is very little change in classification accuracy when the number of scales are varied.

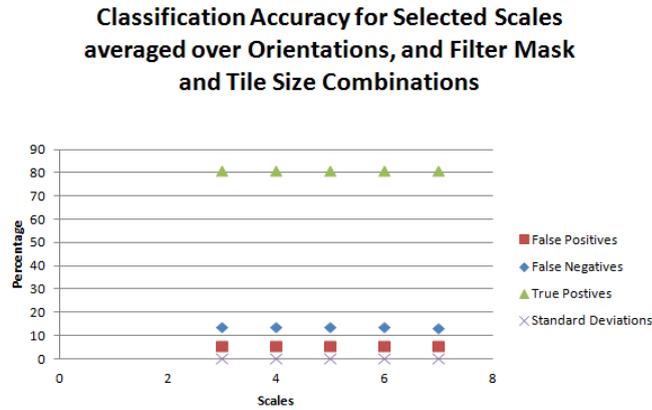


FIGURE 6.1: Experiment 1: Classification Accuracy over Selected Scales.

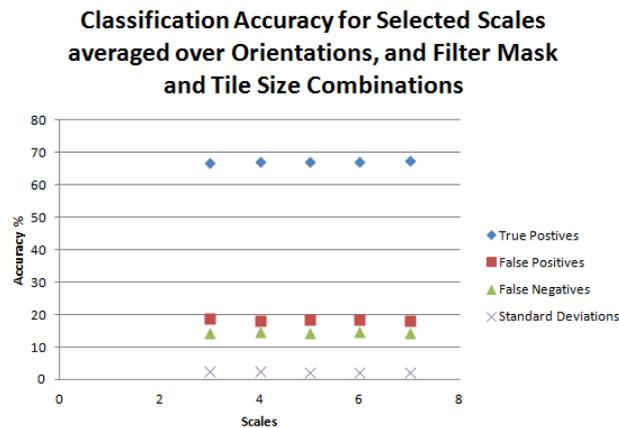


FIGURE 6.2: Experiment 2: Classification Accuracy over Selected Scales.

As with the scales, a similar observation can be made for orientation values: 3, 5, and 7 averaged over all scale, filter mask and tile size values in Figures 6.3 and 6.4, where the increase in classification accuracy is also negligible.

Gabor filters are created using multiple scales and orientations and are used to identify textures with different directions and scales (coarseness). The main concern here is that in general, the selection of too few scale and orientation values (less than 3) may result in an inadequate representation of the underlying texture when applying the Gabor filters.

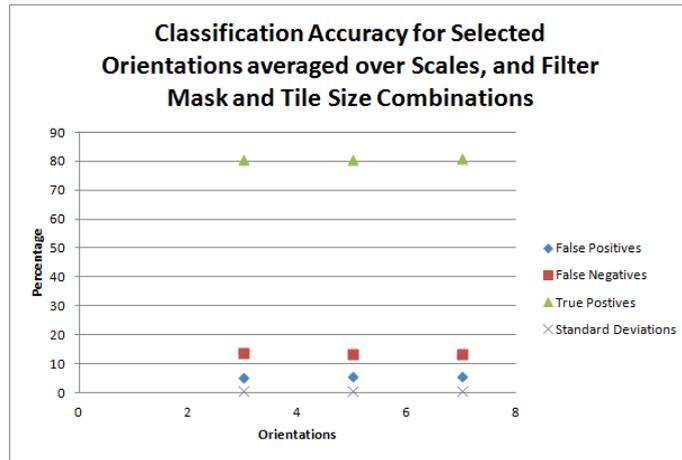


FIGURE 6.3: Experiment 1: Classification Accuracy over Selected Orientations.

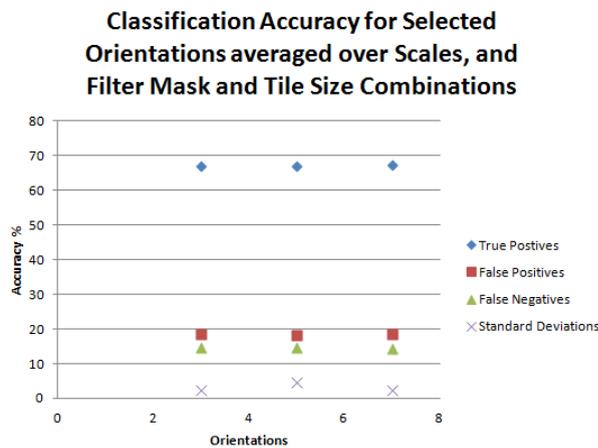


FIGURE 6.4: Experiment 2: Classification Accuracy over Selected Orientations.

Due to the nature of this research (multiple solutions), the False Positives (FP) and False Negatives (FN) are evaluated instead of False Positive Rates (FPR) and False Negative Rates (FNR). For experiment 1, Tables 6.1 and 6.2 show a consistent level of FPs and FNs. However, this means that for the purpose of identifying water bodies, the Gabor filter has in fact failed to identify approximately 13 – 14 percent of the occurrences of water bodies.

TABLE 6.1: Experiment 1: Confusion Table for Selected Scales, averaged over all Orientations, and Filter Mask and Tile Size Combinations.

Scales	True Positives	False Positives	False Negatives	Standard Deviations
3	80.67491017	5.672354526	13.6527353	0.245536734
4	80.82463105	5.368387685	13.80698127	0.020251657
5	80.85170135	5.572808237	13.57549041	0.02163964
6	80.87256105	5.595069859	13.53236909	0.053636603
7	80.88426997	5.715520047	13.40020998	0.066647601

TABLE 6.2: Experiment 1: Confusion Table for Selected Orientations averaged over all Scale, and Filter Mask and Tile Combinations.

Orientations	True Positives	False Positives	False Negatives	Standard Deviations
3	80.80090077	5.307327577	13.89177165	0.784456446
5	80.80620469	5.739413814	13.4543815	0.853828021
7	80.86530709	5.78795847	13.34673444	0.803491364

For experiment 2, Tables 6.3 and 6.4 show that the results are also consistent for scale and orientation selectivity (with slight increase in FNs). However, there are now considerably more FPs than FNs and this is due to more vegetation textures (non-water bodies) being incorrectly classified as turbid water. This phenomenon is explained in detail in Section 6.3.

TABLE 6.3: Experiment 2: Confusion Table for Selected Scales averaged over all Orientations, and Filter Mask and Tile Size Combinations.

Scales	True Positives	False Positives	False Negatives	Standard Deviations
3	66.77405223	18.86565903	14.36028874	2.487856
4	67.18322934	18.1986773	14.61809336	2.418944
5	67.01748361	18.53697725	14.44553914	2.327136
6	67.24256691	18.31158139	14.4458517	2.048832
7	67.55559542	18.03900015	14.40540443	2.069859

TABLE 6.4: Experiment 2: Confusion Table for Selected Orientations averaged over all Scale, and Filter Mask and Tile Combinations.

Orientations	True Positives	False Positives	False Negatives	Standard Deviations
3	66.97498915	18.54229398	14.48271687	2.211209695
5	67.06446499	18.28382981	14.6517052	4.516850445
7	67.20075416	18.40199699	14.39724885	2.325763619

As mentioned in Chapter 2, the Gabor filter is generally applied on the Brodatz and Outex databases for research experiments pertaining to single-textured images. Similarly, when applied to multi-texture satellite imagery, any number of scales and orientations values within the range is suitable.

The results from the experiments also comply with the findings from Bianconi and Fernández [5], and Li et al. [7], that no conclusive evidence is available to warrant the need for optimization of the number of scales and orientations. It can therefore be concluded that determining the best scale and orientation values for the Gabor filter can be seen as a minor objective.

### 6.1.1 Summary

This research has used a different dataset and experimental setup using the ranges of parameter values deduced from previous studies. The findings from the experiments conducted indicates that the number of scales and orientations have no considerable effect on Gabor filter's discriminatory ability. This complies with some of the recent studies that utilized single-texture image databases.

## 6.2 Effects of Filter Mask and Tile Sizes

For selected filter mask and tile sizes averaged over all scales and orientations, Figures 6.5 and 6.6 show a general decline for each progressive mask size and tile size combination. It can be deduced that for this experiment, *larger* tile sizes negatively impact the classification accuracy of the Gabor filter, and that a *smaller* mask and tile size is preferred. One must also consider that these results are specific to the nature of the image set used. Other image sets where the smallest distinct representation of a texture is quite large when compared to the size of the image may have differing unique solutions as noted with some of the "Region" images (see Chapter 4, Section 4.4.2, Figure 4.5).

As stated in Chapter 4, though an image set may come from the same satellite sensor, individual images may contain phenomena of different sizes which have their own optimum solutions. The more images added to a dataset that favour a specific filter mask and tile size combination, the more biased the "optimum" parameter values will become to those images.

For experiment 1, from Table 6.5, the filter mask and size combinations show some slightly erratic behaviour, however a similar observation can be made to that of scales and orientations, whereby the FNs are much higher than FPs due to more vegetation textures (non-water bodies) being incorrectly classified as turbid water.

For experiment 2 however, Table 6.6 clearly shows the increase in FPs, due to vegetation being misclassified as highly turbid waters.

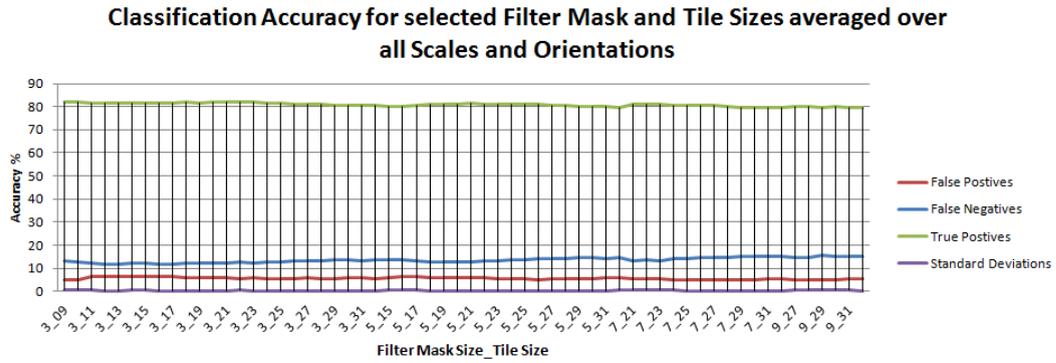


FIGURE 6.5: Experiment 1: Classification Accuracy for Selected Filter Mask and Tile Sizes.

TABLE 6.5: Experiment 1: Confusion Table for Selected Filter Mask and Tile Sizes averaged over all Scales and Orientations

Mask Size	Window Size	True Positives	False Positives	False Negatives	Standard Deviations
3	9	81.91189439	4.795006848	13.29309876	0.688371621
3	10	82.18099702	5.146151623	12.67285136	0.710853849
3	11	81.317017	6.444273222	12.23870978	0.430306413
3	12	81.46235408	6.532276037	12.00536988	0.235734527
3	13	81.57634602	6.453218315	11.97043567	0.158503688
3	14	81.54369155	6.344248677	12.11205977	0.458944704
3	15	81.64336673	6.341940062	12.01469321	0.514173321
3	16	81.77135632	6.328517949	11.90012573	0.312612587
3	17	81.74244174	6.361585437	11.89597282	0.183588217
3	18	81.8524311	6.09153194	12.05603696	0.132349604
3	19	81.70952984	6.056536028	12.23393413	0.332910156
3	20	81.80606345	5.859136272	12.33480028	0.351530234
3	21	81.92716222	5.910557068	12.16228071	0.365517064
3	22	81.79120546	5.698589517	12.51020502	0.43171775
3	23	81.8394227	5.785410053	12.37516725	0.373709002
3	24	81.5589597	5.673122131	12.76791817	0.25466916
3	25	81.36400314	5.663379661	12.9726172	0.270478408
3	26	81.16504879	5.600440093	13.23451112	0.152711355
3	27	81.01914827	5.73265621	13.24819552	0.143155186
3	28	81.13051393	5.629396926	13.24008914	0.147732165
3	29	80.72744574	5.66067371	13.61188055	0.182976309
3	30	80.74746596	5.732305547	13.52022849	0.099902033
3	31	80.68433872	5.871340313	13.44432097	0.145455126
3	32	80.62114162	5.635165059	13.74369332	0.205065001
5	15	80.02148073	6.121866251	13.85665302	0.486498717
5	16	79.92297483	6.435731177	13.64129399	0.725091566
5	17	80.46612057	6.364343126	13.1695363	0.532967023
5	18	81.12062603	6.060978712	12.81839526	0.29245154
5	19	81.08034205	6.121209375	12.79844858	0.202358024
5	20	81.14517974	5.97399646	12.8808238	0.251819188
5	21	81.31064648	5.91803388	12.77131964	0.166982151
5	22	80.9226735	5.734913103	13.3424134	0.280814888
5	23	81.17552954	5.508559918	13.31591054	0.200940355
5	24	81.1832353	5.319872004	13.4968927	0.172930119
5	25	80.97938071	5.290646588	13.7299727	0.19932298
5	26	80.82945517	5.151575832	14.018969	0.173386283
5	27	80.6961822	5.345145391	13.95867241	0.262841412
5	28	80.7037988	5.224593541	14.07160766	0.214769402
5	29	80.12893585	5.330742463	14.54032169	0.206957325
5	30	80.09581836	5.416659858	14.48752178	0.114577949
5	31	79.84886429	5.752924089	14.39821162	0.350137089
5	32	79.67684398	5.808981503	14.51417452	0.417396666
7	21	80.88384119	5.674923056	13.44123575	0.523751609
7	22	80.83015196	5.595321224	13.57452682	0.533126541
7	23	81.02632013	5.525050246	13.44862962	0.558623543
7	24	80.6858624	5.053629569	14.26050803	0.395816588
7	25	80.65246118	5.053891765	14.29364706	0.326940398
7	26	80.48259587	4.975129036	14.54227509	0.276896083
7	27	80.47663307	4.891075176	14.63229175	0.237479307
7	28	80.24026648	4.861615634	14.89811789	0.183184091
7	29	79.83092753	4.884695362	15.28437711	0.246465759
7	30	79.8033061	4.984160539	15.21253336	0.248799231
7	31	79.48567158	5.387675088	15.12665333	0.338037244
7	32	79.43057154	5.441421103	15.12800736	0.364244964
9	27	80.27390213	4.928704276	14.79739359	0.550967499
9	28	80.23562141	4.90631623	14.85806236	0.608208259
9	29	79.82480147	4.747458674	15.42773986	0.405615676
9	30	79.85195	4.855058174	15.29299183	0.429081175
9	31	79.59570445	5.279464775	15.12483078	0.381846151
9	32	79.4362289	5.420175331	15.14359577	0.351047897

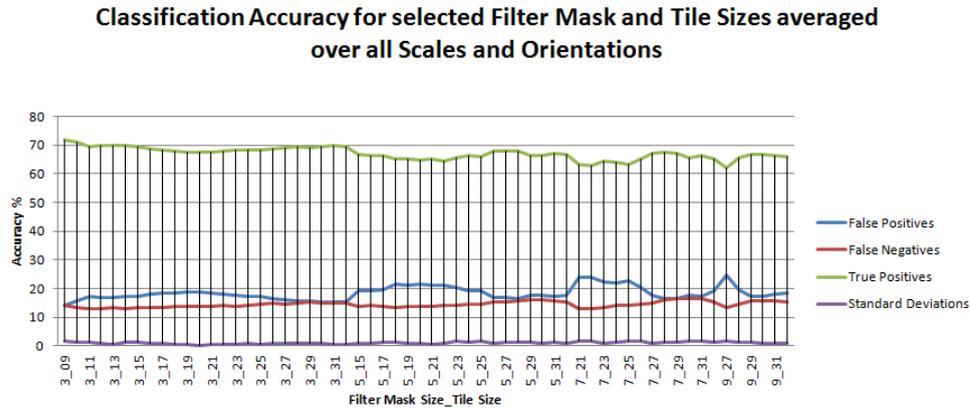


FIGURE 6.6: Experiment 2: Classification Accuracy for Selected Filter Mask and Tile Sizes.

The tables, specifically those pertaining to experiment 2 show that the dataset favour Gabor filter mask sizes of  $3 \times 3$ . This finding complies with statements made by Zhang et al. [8] and Chen et al. [26] who stressed the lack of attention placed on effects of the Gabor filter mask sizes and the need for determining its optimum value.

In this research, not only were the effects of varying the filter mask size investigated but the tile size for window segmentation as well. The tables 6.5 and 6.6, and figures 6.5 and 6.6 clearly show a trend that the dataset favors *specific* filter mask and tile size combinations. On average the smaller filter mask ( $3 \times 3$ ) and tile sizes ( $9 \times 9$  and  $10 \times 10$ ) produced better classification accuracy and a lower number of false positives which is important for applications like water management and flood detection. It can be concluded that for the classification or segmentation of multi-textured images, the need for the optimization of Gabor filter masks and tile sizes supersedes the requirement of optimum scale and orientation values.

### 6.2.1 Summary

Unlike the evaluation of the number scales and orientations, the experiments clearly showed that the Gabor filter favours specific filter mask and tile sizes. When suitable constraints for these parameters were applied, it was much easier to identify good solutions (those with high true positives and low false positives). These constraints *may* therefore be used by a Genetic Algorithm to further improve its effectiveness when trying to find optimum filter mask and tile size combinations.

TABLE 6.6: Experiment 2: Confusion Table for Selected Filter Mask and Tile Sizes averaged over all Scales and Orientations.

Mask Size	Window Size	True Positives	False Positives	False Negatives	Standard Deviations
3	9	71.77490008	14.09918041	14.12591951	1.492960912
3	10	70.95780267	15.55110149	13.49109584	1.157964295
3	11	69.68268494	17.24220078	13.07511428	1.090291874
3	12	69.92876904	17.02115499	13.05007597	0.699263983
3	13	69.95272379	16.82242514	13.22485107	0.63368981
3	14	69.82983934	17.05429669	13.11586397	1.357330691
3	15	69.56481956	17.27472128	13.16045916	1.088932582
3	16	68.8652607	17.91908614	13.21565316	0.968684178
3	17	68.19801615	18.45141348	13.35057037	0.73582549
3	18	67.98483714	18.45916572	13.55599714	0.316209217
3	19	67.50207282	18.76535958	13.7325676	0.273036066
3	20	67.60198529	18.63410338	13.76391133	0.256637953
3	21	67.6933193	18.58904226	13.71763844	0.478124286
3	22	67.85788767	18.17890146	13.96321087	0.388259187
3	23	68.36399661	17.81540461	13.82059878	0.47878662
3	24	68.40988834	17.29646106	14.2936506	0.681284016
3	25	68.33821239	17.12636775	14.53541986	0.624853106
3	26	68.79325607	16.40068072	14.80606321	0.760909879
3	27	69.18383569	16.1133802	14.70278411	0.929551473
3	28	69.43399953	15.84631914	14.71968133	0.880373306
3	29	69.19014333	15.59356767	15.216289	0.702115057
3	30	69.68977397	15.37435326	14.93587277	0.8230305
3	31	69.97529636	15.22234962	14.80235402	0.616949621
3	32	69.59505328	15.33586676	15.06907996	0.412755807
5	15	66.74273635	19.34520438	13.91205927	0.811622696
5	16	66.47039795	19.37050291	14.15909914	0.836969723
5	17	66.5573621	19.66724078	13.77539712	1.169614304
5	18	65.24042827	21.39213882	13.36743291	1.106081676
5	19	65.26127196	21.15595494	13.5827731	0.783108097
5	20	64.9035512	21.47738237	13.61906643	0.859562384
5	21	65.09722161	21.20961224	13.69316615	0.478861058
5	22	64.64188801	21.20751556	14.15059643	0.899124762
5	23	65.68979179	20.17907687	14.13113134	1.551370392
5	24	66.34938726	19.01501714	14.6355956	1.119639694
5	25	66.15802479	19.20456011	14.6374151	1.451533564
5	26	68.07911032	16.6976994	15.22319028	0.927631946
5	27	67.89733086	16.7199136	15.38275554	1.269523642
5	28	67.96099304	16.53493538	15.50407158	1.115015166
5	29	66.46185826	17.48537219	16.05276955	1.137872862
5	30	66.52601579	17.47567171	15.9983125	0.955060545
5	31	67.09405542	17.1368629	15.76908168	1.373405081
5	32	66.78768741	17.75256527	15.45974732	0.860935922
7	21	63.26339054	23.80992775	12.92668171	1.590291703
7	22	62.94683745	23.92222207	13.13094048	1.694440224
7	23	64.46269229	22.26361305	13.27369466	1.022259979
7	24	64.16634909	21.83555293	13.99809798	1.130938204
7	25	63.26520722	22.80059956	13.93419322	1.770554576
7	26	65.24732396	20.37160253	14.38107351	1.629915833
7	27	67.27856526	17.64102402	15.08041072	0.771881784
7	28	67.49401551	16.56217959	15.9438049	1.256205895
7	29	67.22419039	16.35930242	16.41650719	1.086013073
7	30	65.69769472	17.71228814	16.59001714	1.628065544
7	31	66.47927187	17.20433026	16.31639787	1.701769202
7	32	65.33495177	19.36393713	15.3011111	1.15739024
9	27	62.20731347	24.55548041	13.23720612	1.731166805
9	28	65.76338407	19.59767713	14.6389388	1.045749022
9	29	66.78385653	17.36438289	15.85176058	1.052512052
9	30	66.96493502	17.42585538	15.6092096	0.682116764
9	31	66.25158201	18.17703862	15.57137937	0.959215133
9	32	66.1560827	18.38127144	15.46264586	0.945070261

### 6.3 Challenges in Water Body Extraction

From the experiments conducted on the different Hyperion EO-1 “Region” images, the “Region 3” image shows very erratic behavior compared to that of the other images. As stated earlier in Section 4.4.2, the sub-images created from the bottom of the full satellite image contained smooth vegetation textures that were not used for samples. Although these smooth vegetation textures have different pixel intensities similar to that of water bodies (for NDVI in this case), the smooth vegetation textures were incorrectly classified as water bodies as illustrated in Figure 6.7.

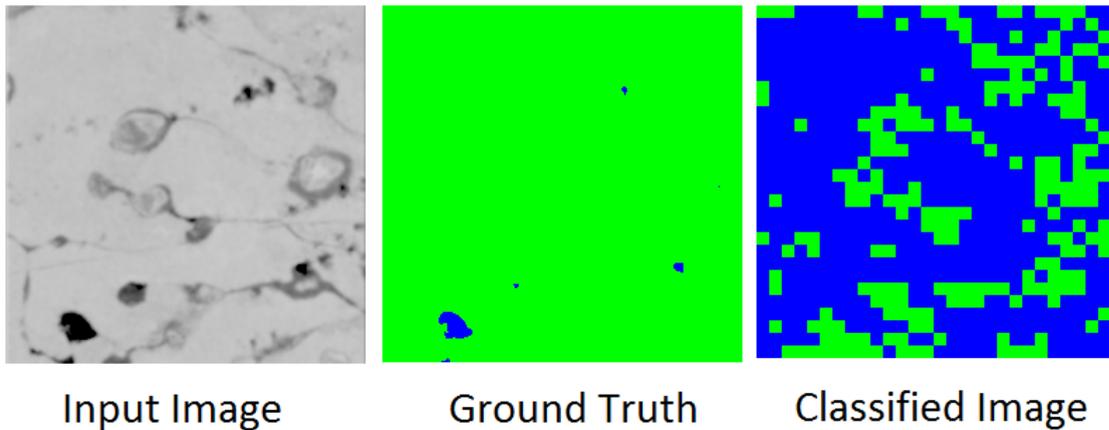
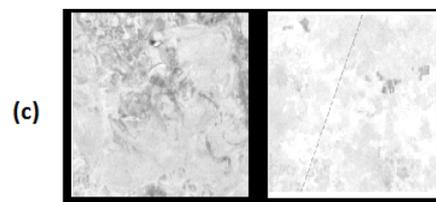
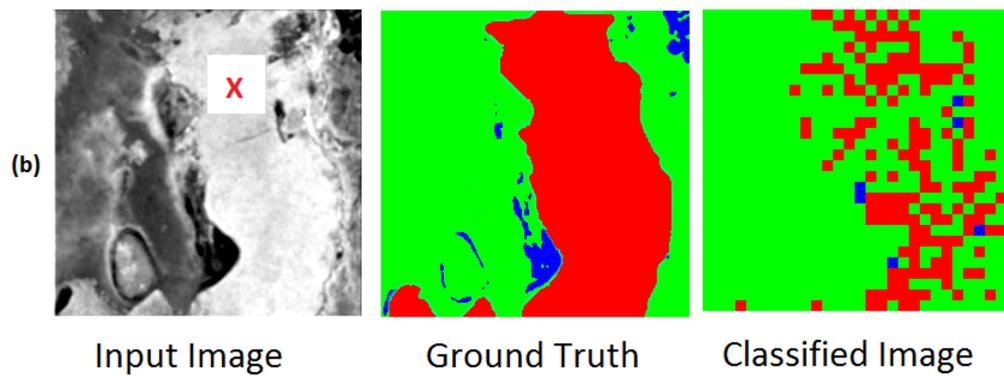
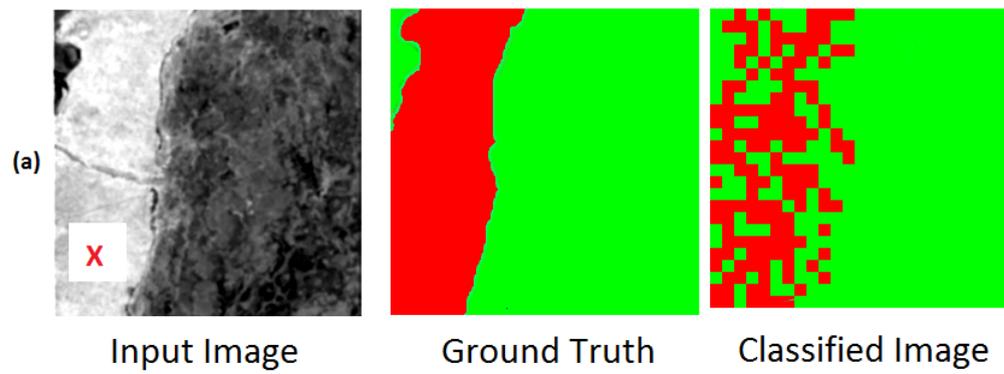


FIGURE 6.7: Anomalies in a sub-image from Region 3.

This sub-image from Region 3 depicts the effects of deliberately leaving out some of the smooth vegetation samples.

For experiment 2, two examples of textures in Figure 6.8(c) taken from images in Region 1 clearly show how similar in appearance the dense vegetation on land is to the highly turbid waters (**X**) in 6.8(a). Figures 6.8(a) and (b) depict highly turbid waters (red) being misclassified as vegetation (green). This resulted in a small percentage increase in false negatives.

The massive increase in false positives in experiment 2 shown in Table 6.6 can only be equated to more numerous instances where vegetation types (green) were misclassified as highly turbid water (red), as illustrated in 6.8(d)). For the purpose of flood detection and monitoring applications, this provides a misconception that there are numerous areas with large water bodies that have protruding or surface vegetation. The limitations in trying to prevent experimental bias (using minimal textures for samples) in a ever changing environment (that is the texture of highly turbid waters changed over time) is apparent.



Vegetation with similar properties to HighlyTurbid Water Bodies

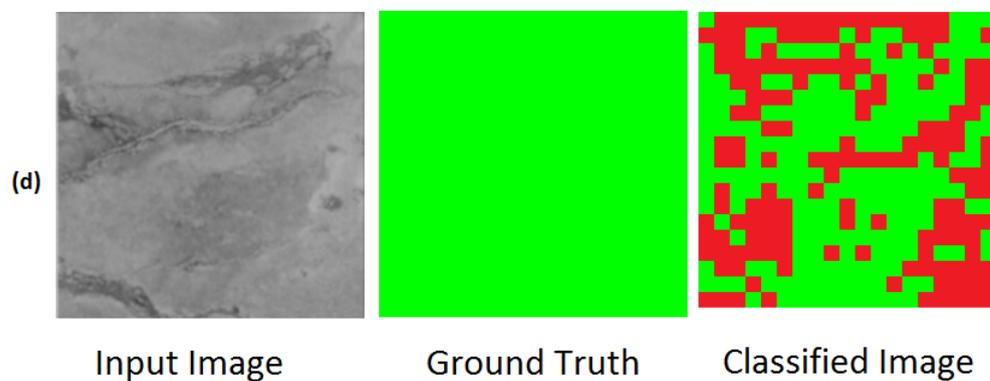


FIGURE 6.8: Anomalies in sub-images from Region 2.

This sub-image from Region 2 depicts the similarity of some of the highly turbid waters to that of vegetation textures. Recall, the highly turbid water is surface vegetation.

For the identification of phenomena that are small, the effects of performing region based (Gabor filter) rather than spectral/pixel based classification is shown in Figure 6.9. The large water body can be easily identified but the smaller water bodies are impossible to extract. Thus, water bodies smaller than the smallest tile size (in this research  $9 \times 9$ ) will not be easily identified.

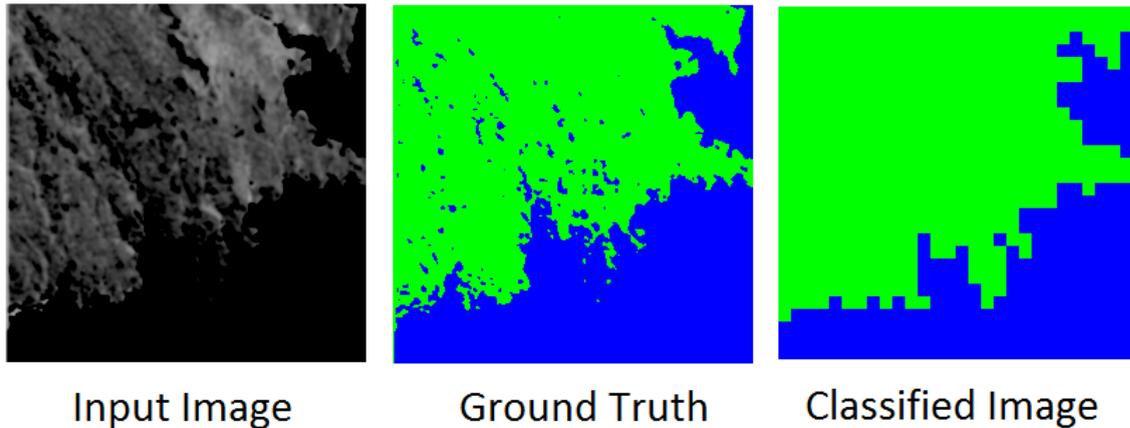


FIGURE 6.9: Anomalies in a sub-image from Region 5.

As mentioned previously, the imagery provided by optical satellite sensors can differ considerably and while the simple framework used was aimed at combining multiple bands for texture classification, only the NDVI band was selected. NDVI is applicable to any optical satellite that has a red and near-infrared band, such as LANDSAT. It can therefore be safely assumed that similar results may be obtained when using satellite images provided by sensors that have similar properties as this research is based on texture and not pixel intensities.

For the identification of phenomena using textures that are conceptually different but exhibit similar properties to other phenomena (such as the highly turbid waters and vegetation in this research), higher resolution imagery maybe required to provide a better delineation.

As stated by Nath and Deb [41] in Chapter 2, Section 2.4.3 while various methods do exist, not all methods will work with the various types of satellite imagery available and there is also no universal standard for water body extraction.

### 6.3.1 Conclusion

The Gabor filter being a region based classifier is inferior to pixel based classifiers for the identification of small objects that are clearly delineated by pixels. However, for concepts that are meaningful only by structure or patterns, the Gabor filter is one

the best techniques to utilize. Although, even its discriminatory ability cannot truly overcome all instances where textures that are conceptually different but structurally similar (inter-class similarity) exist. For land cover classification problems, access to freely available or less costly high resolution optical satellite imagery may provide better results when using the Gabor filter when SAR and/or high resolution Digital Elevation Models are not available.

## 6.4 Automating Gabor Filter Parameter Value Optimization

The Genetic Algorithm was used as one of the two approaches for the scenarios (experiments) to determine its suitability as a means to automate the process of determining the best parameter values. The aim was to design a tool driven by a GA with recommended values and compare its effectiveness to the enumeration approach.

### 6.4.1 Effectiveness of the Genetic Algorithm

For the scales and orientations, although not all the values between the range 3 and 8 for the linear enumeration approach were investigated, the Tables 6.7 and 6.8 contrast the best results using the test dataset from experiment 1 and 2. For these experiments however, the best solutions provided by the GA coincide with the best solutions obtained via the enumeration approach.

TABLE 6.7: Experiment 1: Comparison of the best results obtained via the Linear Enumeration and GA Approaches.

	Linear Enumeration	GA
Scales	5	5
Orientations	3	3
Mask Size	3 x 3	3 x 3
Tile Size	23 x 23	23 x 23
Accuracy	82.86487	82.86487

TABLE 6.8: Experiment 2: Comparison of the best results obtained via the Linear Enumeration and GA Approaches.

	Linear Enumeration	GA
Scales	7	7
Orientations	6	6
Mask Size	3 x 3	3 x 3
Tile Size	9 x 9	9 x 9
Accuracy	82.86487	82.86487

While performance (time complexity) benchmarks are not used in this research, the GA is a much more attractive option. It has provided the same optimum solutions within 8 or 9 generations. This is extremely promising since the preliminary results obtained via the enumeration approach meant evaluating more than 1300 solutions, whereas the GA needed to evaluate about 400 possible solutions. The GA is therefore more suitable, especially for large datasets and when using computationally expensive algorithms like the Gabor filter.

### 6.4.2 Improving GA Effectiveness and Performance

From its very definition, the GAs primary purpose is to be as “natural” as possible when finding solutions from large search spaces, but at the same time prevent premature convergence. However, one also needs to consider when it is not feasible to use a Genetic Algorithm. With regards to this area of research, the following provides some guidelines as to when a GA is problematic to use:

- A GA cannot produce results that will identify trends when varying parameter values.
- If the solution space is small then there is possibly no need for a GA.
- Design choices are generally made to suit the application. A natural GA (no constraints) may produce many solutions that do not exist in the context of the application. If constraints are not placed, the cost overhead induced together with an already computationally expensive algorithm, such as the Gabor filter, is not feasible.
- If the GA takes a considerable time to complete its task then fitness approximation techniques are required to build a model of the fitness function. Therefore some enumeration (using only certain parameter values) maybe required to acquire some real world fitness values so that a fitness approximation technique can then be used to “fill in the gaps”.

By taking into account the statements presented above, for the initial tests the GA was unrestricted but this produced many solutions that were not valid or too far fetched for this application. For example:

- Scale and orientations values in the thousands that are too computationally expensive.
- Filter mask sizes larger than the tile sizes.

- Filter mask and/or tile sizes that are larger than the largest textures/phenomena in the image.

Constraints were deduced and placed on the GA to better suit the Gabor filter and the image dataset. By doing this, significant improvement in performance can be made. For both experiments all sub-images were segmented and stored on disk and loaded into RAM only when necessary. The primary reason for this was to ensure use the GA and/or Gabor filter scripts could continue from a previous state so that no solutions were lost, which unfortunately slowed down the process.

With dropping RAM costs and the availability of 64 bit processors that are able to address and utilize more RAM, the sub-image can be kept in RAM for a significant performance boost and makes the GA approach even more desirable.

### 6.4.3 Methodology: Recommending the GA

One of the main objectives was to automate the process of determining the best Gabor filter parameter values for a given a dataset (complemented by suitable ground-truth images). From the experiments, the following are recommended values when using the GA for parameter value optimization:

- A range of 3 - 8 for both the *scales* and *orientations*.
- Filter mask sizes ranging from 3 x 3 pixels to a third of current tile size.
- Tile sizes ranging from three times the size of the filter mask size up to the size of the image (or some user-defined range which depends on the size of the largest distinct textures relative to the size of the image; for this research it was set to 32 x 32 pixels).

### 6.4.4 Summary

To summarize, through a once-off enumerated approach conducted for the two experiments, this research has provided suitable ranges of values aid in the identification of the optimum Gabor filter parameter values. These values can be used in an automated tool driven by a Genetic Algorithm to significantly improve the rate at which optimal values can be determined for any given multi-textured image database.

Recall Table 2.1 from Chapter 2. Table 6.9 incorporates the results from this research and can be used as constraints on a Genetic Algorithm to automatically determine optimal Gabor filter parameter values for other image datasets.

TABLE 6.9: Gabor Filter Parameters Updated.

	Scales ( $M$ )	Orientations ( $N$ )	Upper Frequency ( $U_h$ )	$\sigma_x$	$\sigma_y$	Mask Size ( $S$ )	Tile Size ( $T$ )	Image Dataset
M.R. Turner [30]	4	4	1/4					
A.K. Jain and F. Farrokhnia [31]	7	4	$\sqrt{2}/4$					Few natural multi-textured images supplied with Brodatz
B.S. Majunath and Y. Ma [3]	4	6	0.4					Brodatz
A.K. Jain et al. [32]	5	4	$\sqrt{2}/4$					Multi-textured visual and infrared images
P. Kruizinga and N. Petkov [33]	3	8	1/5.47					Brodatz and supplied natural multi-textured images
Y. Rubner [34]	4	6	0.3					Brodatz
S. Li and J. Shawe-Taylor [35]	4 – 6	4 – 6	0.4					Brodatz and MIT Vision Texture Database
D.A. Clausi and H. Deng [36]	4	4	$\sqrt{2}/4$					Brodatz
Chen et al. [26]	6	4	0.4					Brodatz
F. Bianconi and A. Fernández [5]	4, 5, 6 (1)	5, 6, 8 (1)	$\sqrt{2}/4$ $\frac{\sigma_g}{2\sigma_x+2\sqrt{(\log 2/\pi)}}$ and $\frac{\sigma_g}{2\sigma_x+2\sqrt{(\log 2/\pi)}}$	0.5	0.5			Brodatz and Outex
W. Li et al. [7]	4	4	varies	0.5, 1.0 (2)	0.5, 1.0 (2)	13x13 (3)		Brodatz and Outex
M. Afshang et al. [29]			varies	varies	varies			Brodatz and Outex
<b>M. Pillay</b>	<b>3 – 8</b>	<b>3 – 8</b>	<b>0.4</b>	<b>derived</b>	<b>derived</b>	<b>3 x 3 up to a third of the tile size.</b>	<b>From three times the size of mask size up to a user-defined range, which depends on the size of the largest distinct texture relative to the size of a multi-textured image.</b>	<b>Hyperion EO-1 Satellite Imagery</b>

<sup>1</sup> Statistically no significant change in classification accuracy.

<sup>2</sup> Used to create initial set of Gabor filters.

<sup>3</sup> Optimum mask size in pixels for Brodatz images.

---

## Future Work and Conclusion

---

### 7.1 Future Work

- As stated by [5], the radial and circumferential superposition of the filters in the frequency domain have not been fully explained, they did however identify a significant improvement on classification due to radial overlap. These therefore need to be further investigated.
- Conduct experiments using fixed values for the lower and upper centre frequency, suitable ranges of values for the number of scales and orientations, but vary the following:
  - Standard deviations of the Gaussian envelope (due to the limited amount of investigation of these parameters).
  - Filter Mask size and Tiles Sizes. Overlapping tiles may provide significantly different results.
- Application of the Gabor filter on higher resolution satellite imagery (if available).
- Finally the use of other distance metrics other than the Euclidean distance.

## 7.2 Conclusion

Gabor filters can be a very effective tool for discriminating textures in complex and multi-textured optical satellite imagery. The filter mask and tile size values have a great influence on the classification of multi-textured image datasets, more so than that of the number of scale and orientations. These parameters must be optimized to allow the Gabor filter to provide the best possible descriptions of textures when using multi-textured images.

By using a different dataset to those used in previous studies, the variation of the number of *scales* and *orientations* had no major impact on the results. This outcome complies with some of the findings in the literature.

For the purpose of optimizing parameters for algorithms that are computationally expensive such as the Gabor filter, the *Genetic Algorithm* is deemed a more suitable alternative than a linear enumeration approach. However in this study, prior evaluation through linear enumeration will allow for a well designed and constrained GA that will eliminate or alleviate the computational complexity caused by large datasets or algorithms with large time complexities.

The results of this study can allow for the creation of a GA driven tool that can automate the selection of suitable Gabor filter parameter values for a given multi-textured image dataset.

---

## Bibliography

---

- [1] W.C. Lin, J. H. Hays, C. Wu, V. Kwatra, and Y. Liu. A comparison study of four texture synthesis algorithms on regular and near-regular textures. Technical report, Carnegie Mellon University, 2004. URL <http://www.cs.nctu.edu.tw/~wclin/nrt.htm>.
- [2] K.J. Dana, B. Van-Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and Texture of Real World Surfaces. *ACM Transactions on Graphics (TOG)*, 18:1–34, 1999.
- [3] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18:837–842, 1996.
- [4] J.G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vis. Res.*, 20:847–856, 1980.
- [5] F. Bianconi and A. Fernàndez. Evaluation of the effects of gabor filter parameters on texture classification. *Pattern Recognition*, 40:3325–3335, 2007.
- [6] F.L. Hellwegera, P. Schlossera, U. Lalla, and J.K. Weissel. Use of satellite imagery for water quality studies in new york harbor. *Estuarine, Coastal and Shelf Science*, 61:437–448, 2004.
- [7] W. Li, K Mao, H. Zhang, and T. Chai. Selection of gabor filters for improved texture feature extraction. In *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, 2010.

- 
- [8] D. Zhang, A.I. Wong, M. Indrawan, and G. Lu. Content-based image retrieval using gabor texture features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1:13–15, 2000.
- [9] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection using evolutionary gabor filter optimization. *IEEE Transactions on Intelligent Transportation Systems*, 2005.
- [10] S.M. Mohsin, M. Younus, and J.A. Anjum. Face recognition using bank of gabor filters. *IEEE-ICET 2nd International Conference on Emerging Technologies*, 1: 144–150, 2006.
- [11] P. Brodatz. Textures: A photographic album for artists and designers. Dover Publications, New York, 1999.
- [12] M. Mirmehdi, X. Xie, and J. Suri, editors. *Handbook of Texture Analysis*. Imperial College Press, 2008.
- [13] I. Dinstein R.M Haralick, K. Shanmugam. Texture features for image classification. *IEEE Transactions on Systems Science, Man and Cybernetics*, 3:610, 1973.
- [14] G. N. Srinivasan and G. Shobha. Statistical texture analysis. *Proceedings of World Academy of Science, Engineering and Technology*, 36:1264–1269, 2008.
- [15] C. H. Chen and L. F. Pau. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*. World Scientific Publishing Co., 1998.
- [16] J. Zhang and T.Tan. Brief review of invariant texture analysis methods. *International Journal of Pattern Recognition*, 35:735747, 2002.
- [17] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Opt. Soc. Amer*, 2: 1160–1169, 1985.
- [18] A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:55–73, 1990.
- [19] V. Kyrki, J.-K. Kamarainen, and H. Kälviäinen. Invariant shape recognition using global gabor features. *12th Scandinavian Conference on Image Analysis*, 1:671–678, 2001.
- [20] H. Cheng, N. Zheng, and J. Qin. Pedestrian detection using sparse gabor filter and support vector machine. *IEEE Proceedings on Intelligent Vehicles Symposium*, 1: 583–587, 2005.

- [21] P.B. Pati, S. Sabari Raju, N. Pat, and A.G. Ramakrishnan. Gabor filters for document analysis in indian bilingual documents. *Intelligent Sensing and Information Processing*, 1:123–126, 2004.
- [22] T.S. Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:959–971, 1996.
- [23] J.G. Daugman. Biometric personal identification system based on iris analysis, March 1994.
- [24] A.K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, 9:846 – 859, 2000.
- [25] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture in two nonstriate visual areas 18 and 19 of the cat. *The Journal of Physiology*, 28:229–289, 1995.
- [26] L. Chen, G. Lu, and D. Zhang. Effects on gabor filter parameters on image retrieval by texture. *10th International Proceedings on Multimedia Modelling Conference*, 1: 273–278, 2004.
- [27] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, Incorporated, Upper Saddle River, New Jersey 07458, 2nd edition, 2002.
- [28] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*, Washington, DC, USA, 2002. IEEE Computer Society.
- [29] M. Afshang, M.S. Helfroush, and A. Zahernia. Gabor filter parameter optimization for texture classification based on genetic algorithm. *IEEE Second International Confernece on Machine Vision*, 2:199–203, 2009.
- [30] M. R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55:71–82, 1986.
- [31] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.
- [32] A.K. Jain, N.R Ratha, and S. Lakhshmanan. Object detection using gabor filters. *Pattern Recognition*, 30:295–309, 1997.
- [33] P. Kruizinga and N. Petkov. Nonlinear operator for oriented texture. *IEEE Transactions on Image Processing*, 8:1395–1407, 1999.

- [34] Y. Rubner. Perceptual metrics for image database navigation. Technical report, Stanford, CA, USA, 1999.
- [35] S. Li and J. Shawe-Taylor. Comparison and fusion of multiresolution features for texture classification. *Pattern Recognition Lett*, 26:633–638, 2007.
- [36] D.A. Clausi and H. Deng. Design-based texture feature fusion using gabor filters and co-occurrence probabilities. *IEEE Transactions on Image Processing*, 14:925–936, 2005.
- [37] R. L. Miller, C. E. Del Castillo, and B. A. McKee, editors. *Remote sensing of coastal aquatic environments*. Springer, 2005.
- [38] Information on earth observation- the national point of contact, 2009. URL <http://www.npoc.nl/EN-version/index.html>.
- [39] C. Wang and Z. Zhao. Land cover change detection based on multi-temporal spot5 imagery. *Urban Remote Sensing Event*, 1:1–5, 2009.
- [40] U. Bayram, G. Can, B. Yuksel, S. Duzgun, and N. Yalabik. Unsupervised land use - land cover classification for multispectral images. *IEEE 19th Conference on Signal Processing and Communications Applications (SIU)*, 1:574–577, 2011.
- [41] R. K. Nath and S.K. Deb. Water-body area extraction from high resolution satellite images- an introduction, review, and comparison. *International Journal of Image Processing (IJIP)*, 3:353–372, 2010.
- [42] M. Melanie. *An Introduction to Genetic Algorithms*. A Bradford Book The MIT Press., Fifth printing, 1999. First MIT Press paperback edition, 1998.
- [43] D.A. Coley. *An Introduction to Genetic Algorithms for Scientists and Engineers*. 1997.
- [44] T. Blickle and L. Thiele. A comparison of selection schemes used in genetic algorithms. Technical report, Gloriastrasse 35, CH-8092 Zurich: Swiss Federal Institute of Technology (ETH) Zurich, Computer Engineering and Communications Networks Lab (TIK), 1995.
- [45] J.H. Holland (1975). Adaptation in natural and artificial systems. *University of Michigan Press, Ann Arbor. Republished by the MIT press*, 1992.
- [46] Y. Goo Wen. Fang C. Wu, G. Tzeng. A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert Systems with Applications*, 32:397–408, 2007.

- 
- [47] J.E Baker. Reducing bias and efficiency in the selection of algorithm. *Proceedings of the Second International Conference on Genetic Algorithms*, 1:14–21, 1987.
- [48] H. Pohlheim. Geatbx: Genetic and evolutionary algorithm toolbox for use with matlab documentation, December 2006 . URL <http://www.geatbx.com/docu/algindex-03.html>.
- [49] S. Lee and H. Rowlands. Introducing the swingometer crossover and mutation operators for floating-point encoded genetic algorithms.
- [50] Y. Rubner, J.Puzicha, C. Tomasi, and J.M. Buhmann. Imperical evaluation of dissimilarity measures for colour and texture. *Computer Vision Image Understanding*, 84:25–43, 2001.
- [51] ShiZhen and C.T. ZhouYang. Comparison of steady state and elitist selection genetic algorithms. *International Conference on Intelligent Mechatronics and Automation*, 1:495–499, 2004.