

**Investigating the potential of a classification algorithm to identify black
wattle (*Acacia mearnsii* De Wild.) trees using imaging spectroscopy**

By

Na'eem Hoosen Agjee

207512769

Submitted in fulfilment of the academic requirements for the degree of
Master of Science in Environmental Science
School of Environmental Sciences, Faculty of Science and Agriculture.

University of KwaZulu-Natal, Westville

March 2012

Declaration One

This study was undertaken in fulfilment of the academic requirements for the degree of Master of Science in Environmental Science and represents the original work of the author. Any work taken from other authors or organisations is duly acknowledged within the text and references chapter.

.....

Na'eem Hoosen Agjee

.....

Mr. A. Pillay

Supervisor

.....

Professor F. Ahmed

Co-supervisor

Declaration Two: Plagiarism

I, Na'eem Hoosen Agjee declare that:

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
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, unless specifically acknowledged, and the source being detailed in the thesis and in the References section.

Signed: _____

Table of Contents

Declaration One	i
Declaration Two: Plagiarism	ii
Table of Contents	iii
List of Figures	v
List of Tables	vi
Abstract	vii
Acknowledgements	viii

Chapter One: Introduction

1.1. Background	1
1.2. Motivation	2
1.3. Aim and objectives	3
1.4. Study area	4
1.5. Outline of thesis	6

Chapter Two: Literature Review

2.1. Introduction	7
2.2. Black wattle (<i>Acacia mearnsii</i> De Wild.)	8
2.2.1. Black wattle as successful invaders	8
2.2.2. Socio-economic and environmental impacts	9
2.2.3. Control methods	12
2.3. Remote sensing	14
2.3.1. Remote sensing of invasive alien plant species	14
2.3.2. Automated algorithms	17
2.3.3. Classification algorithms	19
2.4. Conclusion	20

Chapter Three: Materials and Methods

3.1. Introduction	21
3.2. Image acquisition and pre-processing	21
3.3. Collection of spectral signatures and ground reference data	21

3.4. The classification algorithm.....	22
3.5. The implementation of the classification algorithm	25
3.5.1. BlackWattle class	25
3.5.2. HyperionImage class.....	27
3.5.3. StatisticalTests class	27
3.5.4. RasterImage class	28

Chapter Four: Results and Discussion

4.1. Introduction.....	30
4.2. Automated approach to image classification	30
4.3. Image classification	31
4.4. Age dependent image classification.....	36

Chapter Five: Conclusions and Recommendations

5.1. Introduction.....	42
5.2. Aim and objectives reviewed.....	42
5.3. A synthesis	44
5.4. Limitations of this study	44
5.5. Recommendations for future studies	45

References.....	46
------------------------	-----------

Appendices	57
-------------------------	-----------

List of Figures

Figure 1.1.: Location of study area one and study area two in KwaZulu-Natal, South Africa.	5
Figure 2.1.: A black wattle (<i>A. mearnsii</i>) tree.....	8
Figure 3.1.: Flowchart representing the classification algorithm.	24
Figure 3.2.: Implementation of the classification algorithm.....	26
Figure 3.3.: Results generated from the classification algorithm	29
Figure 4.1.: Classified images indicating the presence and absence of black wattle (<i>A. mearnsii</i>) trees.....	35
Figure 4.2.: Classified images indicating the presence and absence of black wattle (<i>A. mearnsii</i>) trees at three to five, seven to nine and eleven to thirteen years of age for study area one.....	40
Figure 4.3.: Classified images indicating the presence and absence of black wattle (<i>A. mearnsii</i>) trees at three to five, seven to nine and eleven to thirteen years of age for study area two.....	41

List of Tables

Table 4.1.: Results of the accuracy assessment for black wattle trees for study area one and study area two	34
Table 4.2.: Results of the accuracy assessment for all age groups for study area one and study area two	39

Abstract

In South Africa, invasive black wattle trees (*Acacia mearnsii* D. Wild) are a major threat to ecosystem functionality causing widespread social, economic and environmental degradation. It is important that environmental managers are provided with rapid, regular and accurate information on the location of invasive black wattle trees to coordinate removal efforts. This study investigated the potential of an automated image classification algorithm to accurately identify black wattle (*A. mearnsii* De Wild.) trees using imaging spectroscopy. Hyperspectral data acquired by the EO-1 Hyperion sensor was used to identify black wattle trees in two study areas near Greytown, KwaZulu-Natal, South Africa. Image classifications were performed by the classification algorithm to identify black wattle trees using general and age specific spectral signatures (three to five years, seven to nine years, eleven to thirteen years). Results showed that using the general spectral signature an overall accuracy of 86.25% (user's accuracy: 72.50%) and 84.50% (user's accuracy: 69%) was achieved for study area one and study area two respectively. Using age specific spectral signatures, black wattle trees between three to five years of age were mapped with an overall accuracy of 62% (user's accuracy: 24%) and 74.50% (user's accuracy: 49%) for study area one and study area two respectively. The low user's accuracies for the age specific classifications could be attributed to the use of relatively low resolution satellite imagery and not the efficacy of the classification algorithm. It was concluded that the classification algorithm could be used to identify black wattle trees using imaging spectroscopy with a high degree of accuracy.

Acknowledgements

It is with great appreciation that I would like to acknowledge the following people who have provided me with advice, assistance and support throughout the duration of this study.

I would like to thank my supervisor Mr. A. Pillay (UKZN) and co-supervisor Professor F. Ahmed (UKZN) for their guidance over the duration of this study. Without your assistance this study would not have been completed successfully.

To my colleague and friend, Kamleshan Pillay, I am truly grateful for your motivation, support and assistance you provided me throughout this study.

Finally, with gratitude I acknowledge my family. Thank you for your patience, motivation and support throughout this study. This study would not have been completed without you.

Chapter One: Introduction

1.1. Background

Invasive alien plant (IAP) species are a growing threat in South Africa, causing widespread social, economic and environmental degradation (Coetzee *et al.*, 2007; Villamagna and Murphy, 2010). Over the past few decades, a multitude of IAP species have been introduced into South Africa as wind breakers, ornamentals and potentially new species for commercial cultivation (Enright, 2000; van Wilgen *et al.*, 2001). IAP species that have been introduced into South Africa include *Chromolaena odorata*, *Eichhornia crassipes*, *Lantana camara* and *Parthenium hysterophorus*. IAP species have the potential to proliferate and propagate rapidly, extensively expanding the range of their distribution. Their success as invaders lies in the absence of natural enemies and the prevalence of heterogeneous landscape conditions that are suitable for their development (van Wilgen *et al.*, 2004). The invasion of natural ecosystems by IAP species cause severe environmental degradation such as the disruption of ecosystem functionality and ecosystem services (van Wilgen *et al.*, 2004). Of major concern are the negative environmental impacts that IAP species have on South Africa's scarce water resources and rich biodiversity. Consequently, it is imperative to remove and control IAP species in order to maintain the vital ecosystem services that humans depend on. In South Africa, black wattle (*Acacia mearnsii* De Wild.) is one of many IAP species that need to be removed and controlled in order to mitigate further environmental degradation.

Currently, black wattle trees are commercially cultivated extensively in non native areas throughout South Africa (Russel, 2009). Black wattle trees produce large quantities of seeds that are often transported along river systems, negatively impacting on environments downstream (de Neergaard *et al.*, 2005; Holmes *et al.*, 2008). Since its introduction, black wattle trees cover an area of approximately 2.4 million ha of land in South Africa spanning a range of diverse ecosystems (Enright, 2000). Invasive black wattle trees cause severe environmental degradation such as the loss of biodiversity (de Wit *et al.*, 2001; de Neergaard *et al.*, 2005), reduction in streamflow (Scott and Lesch, 1997; Prinsloo and Scott, 1999) and the reduction in catchment water yields (de Wit *et al.*, 2001). The Working for Water (WfW) programme, an initiative of the South African government is at the forefront of removing and controlling the spread of IAP species in non native areas (Zimmermann *et al.*, 2004). The WfW programme successfully integrates a range of social, economic, political and

environmental dimensions to effectively control the spread of IAP species (Richardson and van Wilgen, 2004). Since its inception, the WfW programme has spent more than R 3.2 billion (2005/06 financial year) to combat the spread of IAP species in South Africa (Marais and Wannenburgh, 2008). Invasive black wattle trees form part of the WfW programme in an effort to mitigate their negative environmental impacts. In order for the WfW programme to effectively plan management strategies for combating the spread of invasive black wattle trees, detailed maps of the distribution of the trees are critical for its success (Rowlinson *et al.*, 1999). Remote sensing techniques offer the potential to identify, map, monitor and manage the distribution of IAP species over large spatial scales (Rowlinson *et al.*, 1999; Oumar, 2008).

Remote sensing is the process of acquiring information about the Earth's surface without being in contact with it. Remotely sensed imagery provides a synoptic view of the Earth's surface thereby allowing for complete and accurate information to be acquired over large sometimes inaccessible areas non-destructively (Buerkert *et al.*, 1996; Verma *et al.*, 2003). Hyperspectral imaging spectrometers, for example Earth Observing-1 Hyperion Sensor, capture images in narrow contiguous bands allowing for detailed reflectance spectra to be collected (Vane and Goetz, 1993; Mutanga *et al.*, 2009). This is advantageous as IAP species can be accurately discriminated from the surrounding dense vegetation. The use of remote sensing has been successfully applied to identify and map IAP species. A study conducted by Kimothi *et al.* (2010) mapped the distribution of *Lantana camara* using multispectral imaging spectroscopy in the Rajaji National Park forest in Uttarakhand, India. In addition, a study by Tsai *et al.* (2007) accurately detected the IAP species horse tamarind (*Leucaena leucocephala*) utilizing Hyperion hyperspectral imagery in southern Taiwan. These studies illustrate the potential of mapping the distribution of invasive black wattle trees using hyperspectral imaging spectroscopy. However, the cost and efficiency of proprietary image processing software is a major obstacle to the widespread use of remote sensing techniques to map IAP species.

1.2. Motivation

Traditionally, environmental managers had to visually inspect areas to identify IAP species that can be labour intensive, costly and time consuming. Remote sensing offers a quicker and more efficient method of identifying the spatial distribution of invasive black wattle trees.

Thematic maps produced using remote sensing techniques will provide the spatial location and distribution range of invasive black wattle trees to environmental managers. This will allow for control efforts to be planned and targeted in areas of severe invasion. Further, thematic maps can be incorporated into a geographical information system (GIS) framework to model the distribution of invasive black wattle trees over time (Ghebremicael *et al.*, 2004). This information is vital for the co-ordination of future management strategies that aim to remove and control the spread of invasive black wattle trees in non native areas. However, processing the large volumes of hyperspectral remotely sensed data quickly and efficiently is a major challenge. There is a need for specialist remote sensing applications that are easy to use and cost effective as well as novel algorithms that can achieve greater classification accuracies. Further, there is a need to automate the classification process so that processing hyperspectral satellite imagery can be conducted quickly and efficiently. Automation of the classification process involves carrying out the classification process without user intervention. This will allow routine and repeated classifications to be undertaken thereby providing regular, accurate, timeous and near real time distribution maps of invasive black wattle trees to environmental managers. This study will address these needs through the development of the black wattle classification algorithm and its implementation. It is also hoped that the algorithm could be used, with slight modification, to map other plant species in the future. Finally, as no studies have been conducted on the potential of a classification algorithm to automatically identify black wattle trees using imaging spectroscopy, this study will add to the knowledge base of detecting black wattle trees using hyperspectral remote sensing.

1.3. Aim and objectives

This study aims to investigate the potential of a classification algorithm to identify black wattle (*Acacia mearnsii* De Wild.) trees using imaging spectroscopy. The objectives of this study are:

- To develop an image classification algorithm that will identify black wattle (*Acacia mearnsii* De Wild.) trees using hyperspectral EO-1 Hyperion data.
- To assess the image classification algorithm's ability to automate the classification process.

- To assess the efficacy of the image classification algorithm in identifying black wattle (*Acacia mearnsii* De Wild.) trees using hyperspectral EO-1 Hyperion data.
- To assess the accuracy of the image classification algorithm to identify black wattle (*Acacia mearnsii* De Wild.) trees of varying age groups using hyperspectral EO-1 Hyperion data.

1.4. Study area

The study areas (study area one: 29° 0′ 12″ S, 30° 42′ 29″ E; study area two: 29° 10′ 34″ S, 30° 39′ 9″ E) are located near Greytown, KwaZulu-Natal, South Africa (figure 1.1.). Greytown forms part of the Umzinyathi district municipality and is approximately 70 km north of Pietermaritzburg (figure 1.1.). Greytown and its surrounding areas have a mean annual temperature of 17 °C with an annual rainfall ranging between 400 mm and 836 mm (Babugura, 2010). The area is situated at an altitude of 1038 m above sea level. The topography of the landscape is characterised by deep river gorges, grasslands, wetlands, hills, valleys and bush-velds (Umzinyathi municipality, 2010). The general slope of the land is between 1:5 and 1:6 (Umzinyathi municipality, 2010). Generally, Greytown and its surrounding areas have great agricultural potential owing to a combination of high rainfalls, moderate temperatures, good soils and moderate slopes (Umvoti municipality, 2008). The land uses largely practised include livestock farming, sugarcane farming, dry land crop production and forestry. Commercial farmlands account for more than 70% of the municipal area with forestry plantations dominating the land use (Afrispace consulting, 2009). Species that are commercially cultivated within forestry plantations include *Eucalyptus grandis*, *Pinus patula* and black wattle. Consequently, this study area was selected owing to the possibility of the spread of black wattle trees out of commercial forestry plantations into non native areas.

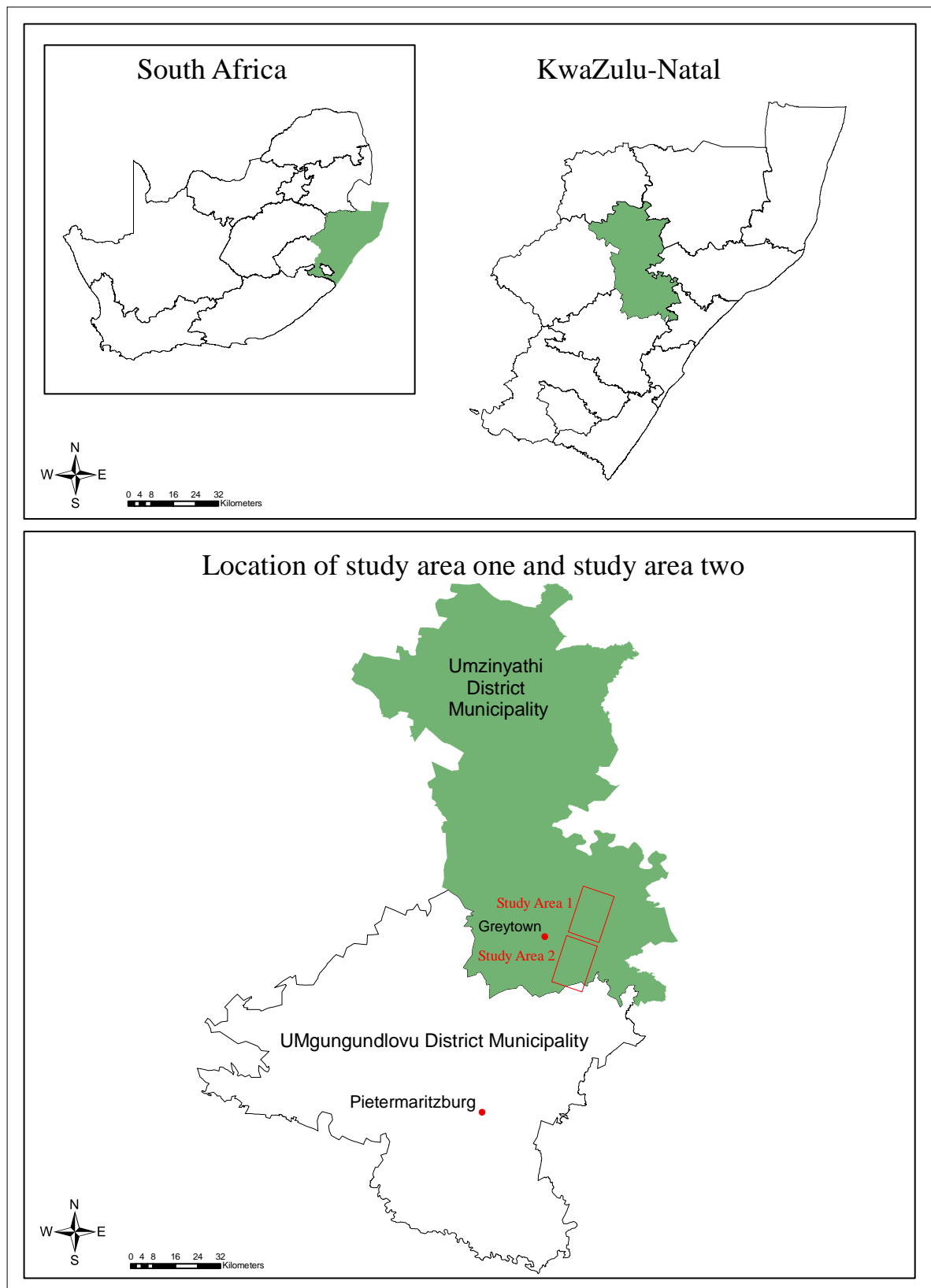


Figure 1.1.: Location of study area one and study area two in KwaZulu-Natal, South Africa.

1.5. Outline of thesis

Chapter two reviews the relevant literature on the potential of a classification algorithm to identify black wattle trees using imaging spectroscopy. Firstly, the characteristics that make black wattle trees successful invaders are reviewed. Further, the socio-economic and environmental impacts associated with black wattle trees in non native areas are discussed and current control methods reviewed. Subsequently, the potential of identifying black wattle trees using remote sensing techniques are explored. The final section reviews the potential use of automated classification algorithms to identify black wattle trees using imaging spectroscopy.

Chapter three provides a detailed account of the methodology employed to carry out this study. The pre-processing techniques performed on the EO-1 Hyperion data sets are outlined. Subsequently, the collection of spectral signatures and ground reference data is presented. This chapter concludes with a detailed description of the classification algorithm and the implementation of the classification algorithm.

Chapter four presents the main results of this study which are discussed and related to other relevant studies. In this chapter, the automated approach employed by the classification algorithm is assessed. The efficacy of the classification algorithm in identifying black wattle trees are presented and discussed. This chapter concludes with a discussion of the performance of the classification algorithm in identifying black wattle trees of varying age groups.

Chapter five concludes this study. The aim and objectives initially outlined are reviewed to establish if they were achieved by this study. Finally, the limitations of this study and recommendations for future studies are presented.

Chapter Two: Literature Review

2.1. Introduction

The invasion of natural ecosystems by IAP species is a major threat to South Africa's water resources, rich biodiversity and ecosystem functionality (Lodge, 1993; Rose and Fairweather, 1997). Transported from Australia, the IAP species black wattle, forms an integral part of South Africa's forestry industry. Black wattle trees have been able to proliferate and spread rapidly; successfully establishing themselves as part of South Africa's landscape (Eldridge *et al.*, 1993; Kull and Rangan, 2008). Invasive black wattle trees threaten freshwater ecosystems throughout South Africa (Poynton, 1979; Dye and Jarman, 2004). Black wattle trees tend to consume large volumes of water from sensitive river systems as they are evergreen and maintain a high leaf area throughout the year (Dye and Jarman, 2004; du Toit and Dovey, 2005). Further, black wattle trees have been found to block water ways, reduce catchment water yields, and reduce the stability and integrity of riparian ecosystems (de Wit *et al.*, 2001). This places great strain on South Africa's limited water resources (Binns *et al.*, 2001; van Wilgen *et al.*, 2001). Currently, various methods are employed to remove IAP species. These include mechanical (Holmes *et al.*, 2008), chemical (Viljoen and Stoltsz, 2008) and biological (Impson *et al.*, 2008) removal methods. However, in an effort to inform removal efforts, IAP species are manually mapped and monitored by field workers that survey invaded areas to assess the extent of invasion.

The inception of remote sensing technologies has been critical in addressing the challenges faced in obtaining information on the spatial distribution of IAP species (Tefamichael *et al.*, 2010). Initially, the potential of remote sensing was limited to multispectral sensors that collected data in three to six broad spectral bands from the visible region (VR) and near infrared region (NIR) of the electromagnetic spectrum (Govender *et al.*, 2007). However, with the inception of hyperspectral sensors, more detailed reflectance spectra can now be collected capable of discriminating of spectrally similar but unique plant species (Mutanga *et al.*, 2009). The use of hyperspectral remote sensing offers the potential for finer temporal, spatial and spectral resolutions that can be used to accurately identify, map, monitor and manage the spread of IAP species in a cost effective manner (Rowlinson *et al.*, 1999; Oumar, 2008). Despite its many benefits, the regular use of hyperspectral satellite imagery is limited by the large data volumes associated with it. Processing large data sets are often time

consuming and limited by the hardware and software capabilities of the computer. The use of automated classification algorithms will facilitate the efficient processing of large volumes of hyperspectral data for maximum feature extraction.

This chapter first reviews black wattle trees as IAP species and then examines the potential of using hyperspectral satellite imagery to identify and map the distribution of black wattle trees. Further, the potential of automated classification algorithms are explored.

2.2. Black wattle (*Acacia mearnsii* De Wild.)

2.2.1. Black wattle as successful invaders

Native to Australia, black wattle trees have since been transported by human agency for cultivation in most countries that are suitable for its development (Kull and Rangan, 2008; Eldridge *et al.*, 1993). Countries that cultivate black wattle trees include Brazil, China, India, South Africa and Zimbabwe (Brown and Ko, 1997; Jones *et al.*, 2008). In South Africa, black wattle trees, are cultivated extensively in forestry plantations throughout the country for a variety of purposes including timber and firewood (Eldridge *et al.*, 1993; de Neergaard *et al.*, 2005). It is the widespread distribution and poor management of forestry plantations in South Africa that has resulted in the invasion of black wattle trees into non native areas (Enright, 2000). However, black wattle trees have many other characteristics that enable them to be highly successful invaders. These include their reproductive ability, canopy structure and their adaptability to a wide range of diverse habitats.



Figure 2.1.: A black wattle (*A. mearnsii*) tree. (Photograph by B. Strong)

Black wattle trees are fast growing evergreen shrubs that form dense thicket in areas that are invaded (Eldridge, 1978; Eldridge *et al.*, 1993; Searle, 1997). A relatively fast growth rate (2 m/yr) ensures that black wattle trees reach maturity and bear seeds early in their life cycle (4-5 years) thereby establishing themselves within an environment (Chaunbi, 1997; Di Stefano, 2002; Bauhus *et al.*, 2004; Christina *et al.*, 2011). Black wattle trees have been found to produce large quantities of seeds. As much as 20 000 seeds/m⁻² have been recorded under mature trees (De Beer, 1986; de Wit *et al.*, 2001). These seeds can lay dormant in the soil for up to 37 years if undisturbed, which ensures the survival of the species in non native areas (Maiden, 1891). Disturbances such as fire events provide instances for the mass germination of seeds allowing for invasive black wattle trees to eliminate all other plant species in the area and to colonize it exclusively (Chaunbi, 1997).

Black wattle trees grow to heights of up to 20 m and have large crowns that dominate the canopy of forestry plantations and natural habitats (Searle 1997; Eldridge *et al.*, 1993). Bauhus *et al.* (2004) reported an average crown diameter of 3.4 m for black wattle trees after 9.5 years of growth. Generally, rows of black wattle trees in forestry plantations are spaced approximately two to three metres apart (Chaunbi, 1997; Khanna, 1997). The planting of trees very close to each other limits the penetration of light and subsequent growth of native vegetation that may occupy the understorey. Similarly, invasive black wattle trees that grow in dense thicket in natural ecosystems limit the penetration of light and growth of native vegetation. However, distribution patterns of invasive black wattle trees may differ when black wattle propagules are transported through natural distribution mechanisms such as wind and water. Black wattle trees can adapt to a wide range of soil types and have the ability to resist extended dry periods allowing them to be highly successful in a wide range of diverse ecosystems (Eldridge *et al.*, 1993; Chaunbi, 1997). It is apparent that black wattle trees are well suited to the South African climate which encourages them to propagate and proliferate as highly successful invaders. In the following section the socio-economic and environmental impacts associated with the presence of invasive black wattle trees in non native areas of South Africa will be discussed.

2.2.2. Socio-economic and environmental impacts

Since its introduction into the forestry industry, black wattle trees now cover an area of approximately 2.4 million ha of land in South Africa (Enright, 2000). This widespread

distribution of black wattle trees poses a major threat to the integrity and stability of a vast array of different ecosystems throughout South Africa (Poynton, 1979; Dye and Jarman, 2004). Over the past few decades, many studies have documented the negative environmental impacts associated with the presence of invasive black wattle trees in non native areas (Scott and Lesch, 1997; Prinsloo and Scott, 1999; Dye and Jarman, 2004; Richardson and van Wilgen, 2004). Some of these negative environmental impacts include a reduction in streamflow, water yield, biodiversity and an increase in fire hazards to indigenous vegetation (Binns *et al.*, 2001; de Neergaard *et al.*, 2005). However, research has been largely focused on the negative impacts that invasive black wattle trees have on South Africa's limited water resources so that management strategies can be formulated to mitigate this effect.

Invasive black wattle trees are a major threat to freshwater ecosystems throughout South Africa (Poynton, 1979; Dye and Jarman, 2004). Black wattle stands maintain a high leaf area throughout the year (Eldridge, 1978; Eldridge *et al.*, 1993; Searle, 1997) with leaf area indexes of 2-3.5 for black wattle stands been reported (Jarman and Everson, 2002; Landsberg *et al.*, 2003). The high leaf area index of black wattle plantations allow the species to maintain a high rate of evaporation throughout the year causing them to consume large volumes of water from groundwater and river systems (Dye and Jarman, 2004; de Neergaard *et al.*, 2005; du Toit and Dovey, 2005). Dye and Jarman (2004) concluded that total evaporation rates may exceed 1500 mm per year for invasive black wattle trees along riparian systems. This places severe strain on water resources that are used for recreational, domestic and commercial uses. A study conducted by Scott and Lesch (1997) measured streamflow response after the afforestation of grassland regions with invasive *Eucalyptus grandis* and *Pinus patula* at the Mokobulaan research catchment. Importantly, a statistically significant decrease in streamflow was reported after three years of growth and after nine years of growth the stream had completely dried up (Scott and Lesch, 1997). Increased water usage by IAP species can cause instances of extreme fluctuations in physico-chemical parameters (example salinity) of the river system having detrimental effects on the biota that occupy the system (Enright, 2000).

IAP species affect the stability of ecosystem processes such as nutrient cycling, water availability and soil fertility. The alteration of ecosystem processes negatively transforms the structure and composition of indigenous biodiversity. Biodiversity is important in ensuring the resilience of ecosystem services so that human activities are maintained (Diaz *et al.*,

2006). However, the introduction of black wattle trees into non native areas tends to invade grasslands and river systems posing a threat to indigenous biodiversity in South Africa (de Wit *et al.*, 2001; Forsyth *et al.*, 2004). There is a lack of literature on the impact that invasive black wattle trees have on indigenous vegetation. However, a study conducted by Allan *et al.* (1997) assessed the impact of commercial afforestation (*Acacia* spp., *Eucalyptus* spp. and *Pinus* spp.) on bird populations in the Mpumalanga Province, South Africa. It was found that there was a significant negative correlation between the diversity of all grassland birds with the extent of forest plantation cover (Allan *et al.*, 1997). In contrast, there was a significant positive correlation between the diversity of species that benefit from afforestation and the extent of forest plantation cover (Allan *et al.*, 1997). Forestry plantations also offer the potential for the establishment of shade tolerant native forest species (Geldenhuys, 1997). However, once invasive black wattle trees have successfully established themselves in non native areas, they increase the biomass of the area (Enright, 2000). An increase in biomass increases the amount of plant material that can burn which poses a fire hazard to the indigenous vegetation (Binns *et al.*, 2001). Within forestry plantations, this scenario is limited by fire reducing measures that are put in place to reduce the frequency of fires (Geldenhuys, 1997). However, within natural habitats increased fire frequencies will eliminate indigenous vegetation leading to the excessive loss of fertile top soil through surface and rill erosion (de Neergaard *et al.*, 2005). Despite this, the presence of invasive black wattle trees in non native areas offer a range of socio-economic benefits to the surrounding communities in which they occur.

Invasive black wattle trees are felled by local rural communities for firewood, timber and are often sold for income (de Neergaard *et al.*, 2005). The presence of invasive black wattle trees in the natural environment is integral to the upkeep of local communities. However, there is a conflict of interest between maintaining invasive black wattle stands so that communities may benefit from them and removing them so that negative environmental impacts are mitigated. A balance needs to be struck so that both benefits are reaped. The WfW programme can be seen as an effective solution to this conflict. The WfW programme creates temporary employment for approximately 30 000 people to assist in the clearing of IAP species (Marais and Wannenburgh, 2008). This initiative provides vital employment to thousands of disadvantaged people resulting in significant poverty alleviation (de Neergaard *et al.*, 2005). Providing employment to rural communities will reduce their dependency on invasive black wattle trees.

From the available literature, many studies have attempted to quantify the effects that IAP species have on a range of parameters at a range of spatial scales (Enright, 2000; Le Maitre *et al.*, 2000; van Wilgen *et al.*, 2001; Le Maitre *et al.*, 2002; Görgens and van Wilgen, 2004; Samways and Taylor, 2004; Cullis *et al.*, 2007). However, the negative impacts that specific species have within ecosystems should be quantified. Research has been focused on invasive black wattle trees as it is ranked third (total invaded area) in the top ten invading species in South Africa (Le Maitre *et al.*, 2000). However, research has largely been conducted on invasive black wattle trees within the Western Cape and Mpumalanga Provinces (Scott and Lesch, 1997; Prinsloo and Scott, 1999; Forsyth *et al.*, 2004). Future research should be focused on the negative environmental impacts that invasive black wattle trees have within the KwaZulu-Natal Province.

2.2.3. Control methods

The WfW programme is active in managing, controlling and containing the proliferation of IAP species in South Africa (Marais and Wannenburgh, 2008). Since its inception, the WfW programme has cleared approximately 1.66 million ha of land of IAP species in South Africa (Marais and Wannenburgh, 2008). Currently, invasive black wattle trees are manually cleared in an attempt to control its spread (Holmes *et al.*, 2008). Manual removal is preferred as it is perceived as being environmentally sound and safer as compared to chemical and biological removal methods (Mathur and Singh, 2004; Greenfield *et al.*, 2007). The manual removal of IAP species from non native areas have shown marked improvements in streamflow (Prinsloo and Scott, 1999).

A study conducted by Prinsloo and Scott (1999) reported a noticeable improvement in the streamflow after the removal of invasive *A. mearnsii* and *A. longifolia* trees from riparian zones at three sites in the Western Cape Province. Results showed that streamflow increased by 9, 10 and 12 m³/ha/day in Du Toits Kloof, Oaklands and Somerset West riparian zones respectively (Prinsloo and Scott, 1999). Similarly, a study by Scott (1997) reported increased streamflow after one third of a *Pinus radiata* plantation in the Western Cape Province had been cleared. In contrast, the clearfelling of *Eucalyptus* spp. at the Mokobulaan research catchment in the Mpumalanga Province did not improve streamflow until five years after being clearfelled (Scott and Lesch, 1997). This could be attributed to the depletion of groundwater storage that would need to be replenished before streamflow would return (Scott

and Lesch, 1997). Further, the deep penetrating root system of *E. grandis* trees could have altered the flow path of water through the catchment resulting in water leaking out of the catchment (Scott and Lesch, 1997). Therefore, the integration of manual removal with chemical removal/biological removal is imperative to ensure that trees are killed in its entirety so that subsequent negative environmental impacts are mitigated. However, the subsequent environmental impacts of chemical removal methods should be determined. Samways and Taylor (2004) stated that endemic dragonflies (Odonata) are likely to recover after the removal of dense stands of invasive black wattle trees. Therefore, the use of chemical removal methods should ensure that recovering indigenous species are not eliminated. Chemical removal methods offer the advantage that large areas of land can be sprayed quickly and inexpensively with relative success. Richardson *et al.* (2006) reported the death of black wattle trees when applied with metsulfuron-methyl and glyphosate herbicide. Further, a study conducted by Viljeon and Stolsz (2008) illustrated the control of black wattle seedlings using Garlon 4 herbicide. Despite its relative success, the use of chemical herbicides poses an ecological risk to the surrounding environment as opposed to biological control methods.

The use of biological control agents is seen as an inexpensive, effective and sustainable method of controlling IAP species limiting the use of other methods such as chemical removal. In the context of South Africa, the use of biological control agents has reduced the cost of controlling IAP species by 19.80% (Zimmermann *et al.*, 2004). Importantly, biological control agents are host specific to target plant species preventing them from becoming invasive themselves (Julien *et al.*, 1999). However, the success of biological control agents lies in their ability to establish themselves in an environment thereby providing a long term management solution. Currently, there are only a few biological control agents that are being tested to control invasive black wattle trees. These include *Melanterius maculatus* and *Dasineura rubiformis* (Diptera: Cecidomyiidae) (de Neergaard *et al.*, 2005; Impson *et al.*, 2008). These biological control agents have been shown to target seedlings as opposed to targeting the plant by damaging it physiologically and morphologically (de Neergaard *et al.*, 2005; Impson *et al.*, 2008). This is vital in maintaining the integrity of commercial forestry plantations while limiting the spread of invasive black wattle trees. A study by Impson *et al.* (2008) suggested that *Dasineura rubiformis* can be employed as a seed reducing biological control agent for black wattle trees. Despite the success of various control methods there needs to be an effective restoration programme that can capitalize on the

cleared land. Black wattle seeds lay dormant in the soil; therefore, effective restoration will have to be undertaken quickly to avoid further invasion. Remote sensing techniques offer the potential to map and monitor the spread of IAP species before and after removal efforts have been undertaken.

2.3. Remote sensing

2.3.1. Remote sensing of invasive alien plant species

Visually assessing the spatial distribution of IAP species on the ground is often subjective, time consuming, costly and spatially restrictive (Everitt *et al.*, 2002). Remotely sensed imagery provides a synoptic view of the Earth's surface thereby facilitating complete and accurate information to be acquired over large areas, non destructively (Verma *et al.*, 2003). Information over complex geographic terrains that were once inaccessible can now be acquired in a cost effective and timeous manner (Joshi *et al.*, 2004; Mutanga *et al.*, 2009). Further, remote sensing systems can store remotely sensed data over long periods of time creating archival databases that can be used to determine land cover changes over time. Therefore, the merits of remote sensing techniques warrant its use in identifying and mapping IAP species to inform management strategies and removal efforts. Initially, the potential of remote sensing was limited to multispectral imaging spectrometers, such as the Landsat 7 Enhanced Thematic Mapper Plus (ETM+), that collects data in three to six broad spectral bands from the VR and NIR of the electromagnetic spectrum (Govender *et al.*, 2007). However, multispectral imaging spectrometers average reflectance spectra over broad spectral bands lack the detailed reflectance spectra required to accurately identify unique plant species from a complex mixture of scene elements (Carson *et al.*, 1995). Therefore, different land covers can only be classified into broad classes when generating thematic maps.

Many studies have employed the use of multispectral remote sensing to identify IAP species (Carson *et al.*, 1995; Mladinich *et al.*, 2006; Cuneo *et al.*, 2009; Kimothi *et al.*, 2010). However, research focused towards identifying and mapping the distribution of black wattle trees using multispectral satellite imagery has been limited. Despite this, a study conducted by Mladinich *et al.* (2006) mapped leafy spurge (*Euphorbia esula* L.) in the Theodore Roosevelt National Park using Landsat 7 ETM+. Leafy spurge has the potential of being

remotely sensed owing to its distinctive yellow-green colour which is likely to be spectrally unique from the surrounding vegetation (Parker Williams and Hunt, Jr. 2002). However, an unsupervised classification algorithm was used to classify leafy spurge that resulted in an overall classification accuracy of approximately 63% (Mladinich *et al.*, 2006). This relatively low overall accuracy could be attributed to the classification method used and the coarse spectral resolution of multispectral imagery (Mladinich *et al.*, 2006). In contrast, Cuneo *et al.* (2009) detected and mapped the invasive, African Olive (*Olea europaea* L. ssp. *cuspidata* Wall ex G. Don Ciferri) in Sydney, Australia using Landsat 7 Enhanced Thematic Mapper (ETM) data and a supervised classification. It was found that from a total area of 1907 ha of dense African Olive infestation there was an omission error of 7.50% and a commission error of 5.40% (Cuneo *et al.*, 2009). This accurate classification could be attributed to the classification method used as well as the phenology of the plant even though multispectral satellite imagery was used. In some instances phenological stage variation in orientation of leaves, age of leaves, variation in leaf area index and different slopes of the locations where the individuals are found could make the spectral signature of a species difficult to define (Cuneo *et al.*, 2009). This creates intraspecies variation that contributes to overlapping spectral signatures between co-occurring species (He *et al.*, 2011). In contrast, Hestir *et al.* (2008) reported significant spectral variation based on phenology between perennial pepperweed (*Lepidium latifolium*) and water hyacinth (*Eichhornia crassipes*) plants. Image acquisition at key phenological stages may assist in distinguishing between different IAP species (He *et al.*, 2011). Invasive black wattle trees form dense thicket in areas that are invaded (Eldridge, 1978; Eldridge *et al.*, 1993; Searle, 1997). Dominating the stands canopy ensures that only the reflectance spectra of invasive black wattle trees would be measured. Further, black wattle trees are evergreen; therefore, detecting their coverage will not be limited by seasonal variation.

Black wattle trees that form part of commercial forestry plantations occur in large stands. However, black wattle trees that invade natural ecosystems occur in stands of variable sizes that are dependent on the severity of the invasion. The spatial resolution of remotely sensed data used is critical to the level of accuracy of the classification (He *et al.*, 2011). Carson *et al.* (1995) found that Landsat Thematic Mapper (TM) and Satellite Pour l'Observation de la Terre (SPOT) data with a spatial resolution of 30 and 20 meters respectively is inadequate to identify plant species. Unless stands are large enough, the spectral variability within pixels will hamper the classification process and its ability to classify pixels accurately (He *et al.*,

2011). Further, spectral similarity of a pixel to the spectral signature may not necessarily mean that the entire pixel is covered by a plant species and therefore may over estimate the land cover of the plant species. Russel (2009) showed that *A. mearnsii* can be identified and mapped using SPOT 5 imagery. Results showed an overall accuracy of 78.77% with a kappa statistic of 0.7495 in the Fort Nottingham area (Russel, 2009). SPOT 5 imagery has a relatively low spectral resolution but the high spatial resolution (10 m) which could have positively influenced the reflectance spectra measured and the classification process. In contrast, hyperspectral satellite imagery captured by the EO-1 Hyperion sensor has a high spectral resolution but a coarse spatial resolution (30 m) which may limit the accuracy of identifying plant species. A study by Carter *et al.* (2009) compared the effectiveness of Landsat 5 Thematic Mapper (TM5, 30 m), QuickBird (QB, 2.5 m) and EO-1 Hyperion (Hyperion, 30 m) data at different spatial resolutions in discriminating Tamarisk (*Tamarix* spp., saltcedar) stands in Colorado, USA. Results showed that multispectral data at a high spatial resolution (QB, 2.5 m) was more effective than either multispectral (TM5) or hyperspectral (Hyperion) data at a moderate spatial resolution (30 m). This illustrates that the spatial resolution of an image is as vitally important as the spectral resolution of the image. Consequently, a balance needs to be struck between spatial and spectral resolutions to ensure maximum detection accuracy when identifying invasive black wattle trees.

With the inception of hyperspectral imaging spectrometers (Airborne Visible/Infrared Imaging Spectrometer [AVIRIS], HyMap and Hyperion) more detailed reflectance spectra can now be collected. Hyperspectral imaging spectrometers capture images in narrow contiguous bands that allow for the discrimination of spectrally similar but unique materials (Vane and Goetz, 1993; Mutanga *et al.*, 2009). Spectra collected by hyperspectral imaging spectrometers range from 350 - 2500 nm covering the visible, near infrared and shortwave infrared regions of the electromagnetic spectrum. Hyperspectral imaging spectrometers capture detailed reflectance spectra recording subtle changes in reflectance for different scene elements. Differences in reflectance could be attributed to differences in pigments, nutrients and structural properties of the elements in a scene giving each element a unique spectral signature which can be used to distinguish between IAP species (Asner *et al.*, 2008). This is crucial when identifying a single plant species from a mixture of complex scene elements that reflect similar reflectance spectra.

Launched in November 2000, the EO-1 Hyperion sensor is an example of a hyperspectral imaging spectrometer that can be used to identify, map and monitor the spread of IAP species (Pearlman *et al.*, 2003; Lass *et al.*, 2005). The Hyperion sensor is a pushbroom hyperspectral sensor that captures data between 400 – 2500 nm in 242 bands (198 calibrated) at a band width of 10 nm (Pearlman *et al.*, 2003; Lass *et al.*, 2005). The Hyperion sensor operates at an altitude of 705.0 km and captures an area of 7.7 km at a spatial resolution of 30 m (Pearlman *et al.*, 2003; Lass *et al.*, 2005). Hyperspectral remote sensing has been successful in mapping numerous IAP species (Tsai *et al.*, 2007; Asner *et al.*, 2008; Hestir *et al.*, 2008). However, no studies have been carried out on identifying invasive black wattle trees using hyperspectral imaging spectroscopy.

A study by Lass *et al.* (2005) used hyperspectral images from a charged-coupled device (CCD) digital camera (spatial resolution: 2 m, spectral resolution: 400 to 953 nm, band width: 12 nm) to identify locations of spotted knapweed (*Centaurea maculosa*) and babysbreath (*Gypsophila paniculata*). It was found that 57% of known spotted knapweed infestations and 97% of known babysbreath infestations were identified through the use of hyperspectral imagery. A study by Ustin *et al.* (2002) detected IAP species using AVIRIS data. Results included the mapping of *Arundo donax* at accuracies of 90.68% and 97.79% using a spectral angle mapper and maximum likelihood classification respectively in the Camp Pendleton Marine Base in California. Studies by Lass *et al.* (2005) and Ustin *et al.* (2002) illustrate the potential of using remote sensing techniques to identify invasive black wattle trees. This study aims to fill the knowledge gap in the identification of the distribution of invasive black wattle trees using hyperspectral imaging spectroscopy.

2.3.2. Automated algorithms

Traditionally, the spread of IAP species was determined by manually counting each plant on an aerial photograph which proved time consuming, expensive and inefficient (Niu, 2006). Further, IAP species propagate and proliferate rapidly and in this manner dynamically change the land cover of infested areas. This proved a challenge to identifying and updating the spatial distribution of IAP species in non native areas (Agüera and Liu, 2009). Consequently, image classification techniques have been implemented within remote sensing applications to expedite this process. Remote sensing applications provide an interface through which the classification process as well as complex mathematical and computational algorithms can be

performed to identify IAP species such as black wattle trees. Currently, there is a shift in research towards specialist remote sensing applications (Grün, 2000; Zhao *et al.*, 2002; Leckie *et al.*, 2003) that offer an automated approach to image classification. However, there is a lack of literature and research in applying automated approaches to image classification in identifying IAP species and vegetation in general. Despite this, the driving force behind an automated approach is optimizing the processing of large datasets accurately in short time frames with little external expert knowledge in a cost effective manner.

Hyperspectral remotely sensed data is characterised by large volumes of data which is time consuming to process when identifying IAP species. An automated approach to image classification offers a possible solution to process large volumes of data quickly and efficiently (Leckie *et al.*, 2003). Hyperspectral imaging spectrometers such as the EO-1 Hyperion sensor has a high temporal resolution of 18 days ensuring that regular and reliable data over areas of interest is available. Automated image classification can ensure routine and repeated image classifications to be performed over areas of interest with consistent results (Higgins and Harris, 1997). Consequently, a time series analysis can be undertaken to monitor and model the spread of invasive black wattle trees over short and long periods of time. The ability to conduct rapid and routine image processing is instrumental in providing near real time information to inform management strategies and removal efforts of invasive black wattle trees.

There is a growing trend in the development of specialist automated image classification applications. These applications are tailored and provide a few key tools that are required for particular applications. They are inexpensive as compared to comprehensive software packages that offer a range of tools that aren't required on a day to day basis. This allows for remote sensing applications to be accessible and available to a wide range of environmental managers and practitioners. Previously, the classification process required well trained personnel to conduct the classification and interpret its results. However, through the use of automated image classification applications, expert knowledge can be packaged and distributed widely. This allows for image classifications to be undertaken by less skilled personnel that have a basic understanding of remote sensing concepts. This ensures that environmental managers and decision makers can focus primarily on the results without having to worry about technical details behind the classification.

Research has been focused on automated feature extraction such as roads and buildings (Grün, 2008; Zhao *et al.*, 2002; Leckie *et al.*, 2003). This is because these features are larger than one pixel in size. However, with plant species, this may not be the case depending on the size of the stand. Because the spatial resolution of satellite images are still relatively low an object based approach may not be practical. A pixel based approach is still necessary to conduct accurate image classification. This involves the comparison of all the bands for a single pixel to a spectral signature collected in the field or under laboratory conditions.

2.3.3. Classification algorithms

Classification algorithms compare spectral signatures of a feature to each pixel in a multispectral or hyperspectral satellite image to identify features. Spectral signatures can be collected using field spectrometers *in situ* or under laboratory conditions. Spectral signatures can also be collected from the image itself through the use of training sites which are homogenous areas that represent the land cover or pure pixels. An example of a classification algorithm is the spectral angle mapper algorithm. The spectral angle mapper compares the spectral angles between the reflectance spectrum of a pixel and the spectral signature obtained from training sites or captured using a field spectrometer (Kruce *et al.*, 1993). Each pixel is assigned to a class according to the lowest spectral angle value (Kruce *et al.*, 1993).

Internationally, many studies have employed the use of the spectral angle mapper algorithm, to identify IAP species (Lass *et al.*, 2002; Lawrence *et al.*, 2005). However, there are no studies that have applied the use of the spectral angle mapper to identify invasive black wattle trees. Despite this, a study conducted by Narumalani *et al.* (2006) detected saltcedar (*Tamarix* sp.) using the spectral angle mapper algorithm to classify airborne hyperspectral imagery. It was found that saltcedar, cottonwood and other woody species can be spectrally discriminated. Further, the images were classified with an overall accuracy of 83%. In contrast, a study conducted by Lawrence *et al.* (2005), mapped leafy spurge (*Euphorbia esula* L.) and spotted knapweed (*Centaurea maculosa* Lam.) using the spectral angle mapper algorithm. From this study overall accuracies of 40% and 66% for the spotted knapweed and leafy spurge sites, were found respectively. Thus it is clear that there is still some variability in accuracy with regard to spectral matching. This study uses a different approach to spectral matching by using statistical techniques to match spectral signatures to spectral profiles of single pixels.

2.4. Conclusion

The propagation and proliferation of invasive black wattle trees in non native areas is detrimental to ecosystem functionality. Therefore, it is imperative that invasive black wattle trees are mapped so that removal efforts are targeted in areas of severe invasion. However, mapping and monitoring the spatial distribution of invasive black wattle trees is a challenge because it is costly, time consuming and labour intensive. Remote sensing offers the technology to map and monitor the spread of invasive black wattle trees. Many studies have employed the use of hyperspectral remote sensing to map IAP species. However, future studies should focus on identifying invasive black wattle trees using hyperspectral imaging spectroscopy in KwaZulu-Natal, South Africa. Currently, there is an urgent need to provide near real time information on the spatial distribution of invasive black wattle trees to inform removal efforts. Various factors such as time constraints, large data sets and the cost of remote sensing specialists are hindering this process. A possible solution involves the use of automated image classification algorithms that provide quick, efficient, routine and repeated image classification. In general, this review has illustrated the potential of an automated classification algorithm to identify black wattle (*Acacia mearnsii* De Wild.) trees using imaging spectroscopy.

Chapter Three: Materials and Methods

3.1. Introduction

This study assesses a novel classification algorithm to automatically identify black wattle trees using imaging spectroscopy. This chapter outlines the methodology followed to achieve the aim of the study. A detailed explanation of the image acquisition and pre-processing techniques employed is provided. Subsequently, the methodology used to collect spectral signatures of black wattle trees and ground reference data is outlined. The chapter concludes with a detailed description of the classification algorithm and its implementation.

3.2. Image acquisition and pre-processing

The EO-1 Hyperion sensor captures images in 242 spectral bands in the 400-2400 nm spectral range at a spectral resolution of 10 nm and a spatial resolution of 30 m (Pearlman *et al.*, 2003). A single Hyperion image was captured (path 175/82) on the 19th March 2006 covering the entire study area (study area one: 29° 0' 12" S, 30° 42' 29" E; study area two: 29° 10' 34" S, 30° 39' 9" E). The image was provided as Hyperion level 1R data which was calibrated to at-sensor radiance ($\text{W m}^{-2} \text{sr}^{-1} \mu\text{m}^{-1}$); only 196 out of 242 spectral bands were calibrated. The radiance image was spectrally subsetting removing un-calibrated (1-7, 58-78, 225-242) and bad spectral bands (80-82, 120-132, 165-182, 185-187, 220-225). The subsetting radiance image was atmospherically corrected and transformed to canopy reflectance using the FLAASH (Fast Line-of-Sight Atmospheric Analysis of Spectral Hypercubes) atmospheric correction algorithm which is built into the ENVI (Environment for Visualising Images: ENVI, 2006) software package. The reflectance image was then ortho-rectified and georeferenced (Universal Transverse Mercator, zone 36 South) according to a Landsat 7 ETM+ image (6th March 2006). An overall total root mean square error (RMSE) of less than one pixel was used as an indication of a good geometric correction. The resultant image was re-sampled to the new grid system using a cubic convolution algorithm.

3.3. Collection of spectral signatures and ground reference data

Spectral signatures of healthy black wattle trees (general signature, three to five years, seven to nine years, eleven to thirteen years) were collected from the Hyperion images within ENVI for each study area. Homogenous areas that represent black wattle trees of varying age

groups were identified within each study area. Black wattle trees of varying age groups were identified from rasterized polygons of commercial forestry plantations of known ages that are maintained by the Mondi group within each study area. The general signature was generated through the combination of spectral profiles of pure pixels that represent black wattle trees of varying age groups. To generate age dependent spectral signatures, the spectral profiles of four samples each of more than thirty pure pixels that represent black wattle trees (three, four, five, seven, eight, nine, eleven, twelve, and thirteen years) were averaged. The spectral profiles of black wattle trees between three to five, seven to nine and eleven to thirteen years of age were averaged to generate three age dependent spectral signatures. For each spectral signature ground reference data was collected. For the general spectral signature, ground reference data was collected from a land cover map of the study areas. In ENVI, a set of random points were generated from the land cover map indicating the presence (400 points) and absence (400 points) of black wattle trees. For each age group, reference data was collected from rasterized polygons of commercial forestry plantations of known ages that are maintained by the Mondi group. In ENVI, a set of random points were generated from the polygons indicating the presence (100 points) and absence (100 points) of black wattle trees. The four spectral signatures (general signature, three to five years, seven to nine years, eleven to thirteen years) together with its associated ground reference data were used as input to the classification algorithm.

3.4. The classification algorithm

The classification algorithm developed is based on the concept of spectral matching. Spectral matching involves comparing the spectral profile of a single pixel to a reference spectrum to determine if the reference object is present (Kruse *et al.*, 1993). The classification algorithm implements a novel statistically based comparison technique to determine the presence or absence of black wattle trees within areas of interest. Figure 3.1. illustrates a flowchart representing the black wattle classification algorithm. The subsetted EO-1 Hyperion image and the reference spectrum of healthy black wattle trees are input to the classification algorithm (figure 3.1.). The classification algorithm compares the reference spectrum to the spectral profile of every single pixel in the Hyperion image. At each band a statistical measure, the z-score is calculated using the reflectance value, the mean and standard deviation values from the reference spectrum at the corresponding band (figure 3.1.). A z-score is calculated using the following formula:

$$z \text{ score} = ((A-B)/C)$$

Where:

A = reflectance value for a single pixel for a single band (image)

B = mean reflectance value for a single band (reference spectrum)

C = standard deviation of B

From the z-score, two probability values within 95% confidence intervals are determined from a z-table at each band. The final probability at each band is calculated using the following formula:

$$\text{probability} = (A-B)$$

Where:

A = probability value from the positive half of the z-table

B = probability value from the negative half of the z-table

However, if the probability values do not lie within 95% confidence intervals, the final probability is assigned a zero for that band (figure 3.1.). The above procedure is repeated until probability values are determined for all bands for a single pixel. If the probabilities of more than 90% of the total number of bands for a single pixel fall within 95% confidence intervals, the probabilities are averaged and the mean stored (figure 3.1.). If this is not the case the pixel is assigned a zero. The above procedure is repeated for each pixel for the entire Hyperion image resulting in a two dimensional raster of stored probability values. The classified image is generated from the stored probability values indicating the probability of presence of black wattle trees for each pixel (figure 3.1.). The higher the probability the greater the chance that black wattle will be present within that pixel area.

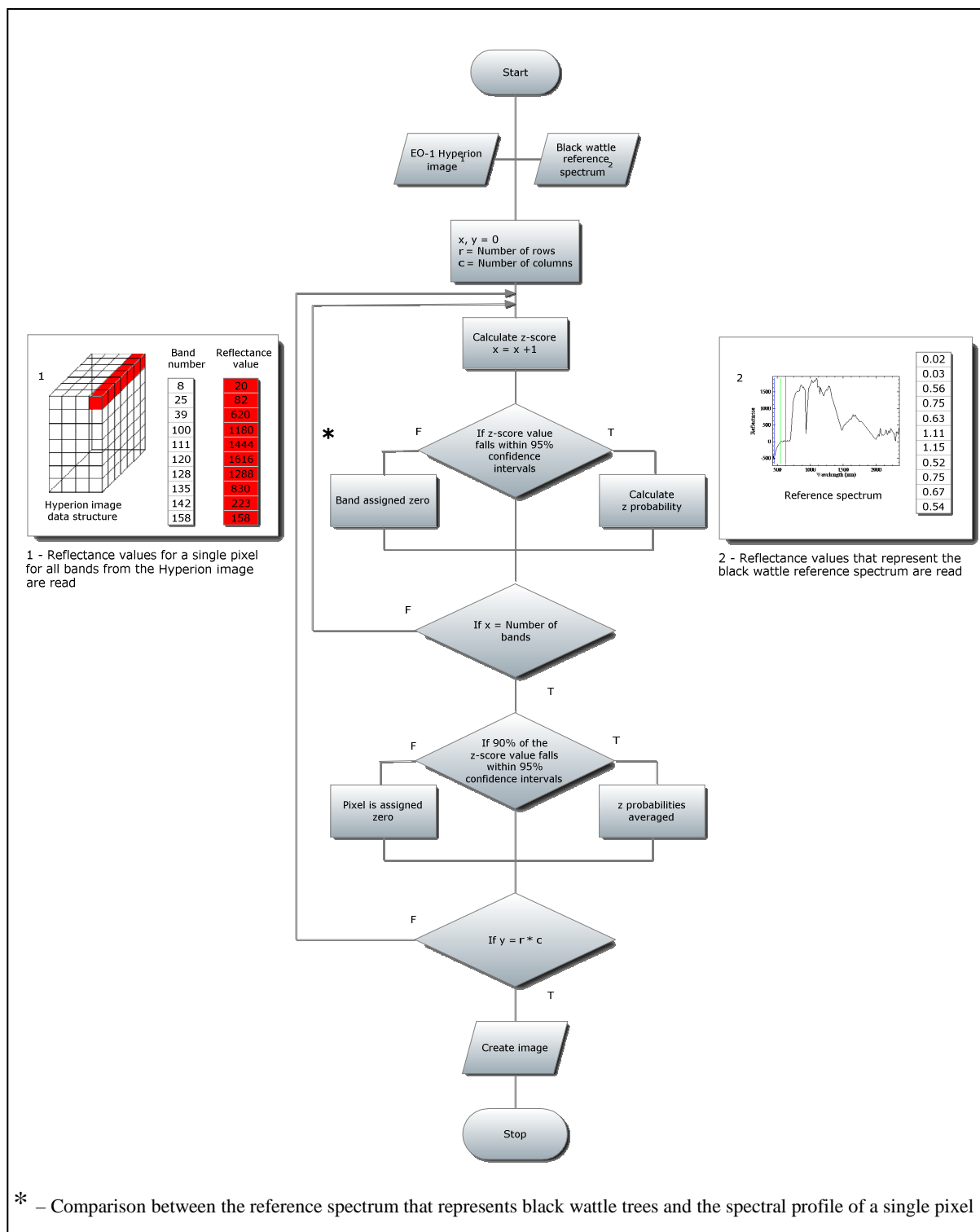


Figure 3.1.: Flowchart representing the classification algorithm.

3.5. The implementation of the classification algorithm

The application was written in the Java programming language and was designed using an object orientated approach. The application consists of four independent classes namely the BlackWattle class, HyperionImage class, StatisticalTests class and RasterImage class. Each class consists of many methods that provide the necessary functionality. In the following sections, each class is described and discussed in greater detail outlining its role within the application. The complete source code for all classes is provided in Appendix A.

3.5.1. BlackWattle class

The BlackWattle class manages the graphical user interfaces (GUI) that an end user interacts with. The main methods that constitute the BlackWattle class are the blackWattleHome, displayClassifiedImage, exportClassifiedImage and blackWattleClassification methods. The blackWattleHome method creates and displays the main user interface through which the various functions may be accessed (figure 3.2.). The displayClassifiedImage method creates an interface through which classified images may be viewed. The exportClassifiedImage method creates an interface through which classified images can be exported from .png to the .tiff data format. Further, the exportClassifiedImage method reads the classified image in .png data format and writes the classified image in .tiff data format to the specified disk location. The blackWattleClassification method creates and displays an interface through which the relevant files for input into the classification algorithm may be selected.



Figure 3.2.: Implementation of the classification algorithm.

3.5.2. HyperionImage *class*

The HyperionImage class processes the Hyperion image header file and the Hyperion image data file. The two main methods within this class are the readHeaderFile and readDataFile methods. The readHeaderFile method reads each line of the Hyperion header file and stores essential information required during the processing. This information includes samples, lines, bands, header offset, interleave and map info. The readDataFile method reads a single pixel from the Hyperion data file given the row and column values. This method calculates the position of the pixel within the file and reads, stores and returns the reflectance values for each band for that pixel.

3.5.3. StatisticalTests *class*

The StatisticalTests class implements the classification algorithm. The two main functions performed by the StatisticalTests class are the image classification and assessing the accuracy of the classified image.

The pixel based classification process is undertaken by comparing the spectral signature of black wattle trees to the spectral profile of a single pixel. The setImgVal method reads each line of the spectral signature file and stores the mean reflectance values and standard deviation values in separate one dimensional arrays. Conversely, the readDataFile method returns reflectance values for a single pixel for all the bands from the Hyperion image file. The zTable method reads and stores z-score values from a standard z-table into a two dimensional array. The zTest method implements the classification algorithm (see 3.4., Chapter 3) by comparing the spectral profile of a single pixel to the spectral signature of black wattle trees to determine if black wattle trees are present or absent.

The accuracy assessment is performed by comparing the land cover (presence or absence) of ground reference data with the land cover (presence or absence) at corresponding locations within the classified image. The setObservedExpectedVal method sets the observed and expected values for the chi-squared test. The setObservedExpectedVal method reads through the file and at each record the geographical co-ordinates are read. The corresponding row and column values within the image for the geographical co-ordinates are calculated and returned by the getRCVal method. If the row and column values fall within the boundary of the study area the observed value (presence or absence) is stored; otherwise the record is

discarded and the next record is processed. Using the row and column values the expected values (presence or absence) are retrieved and stored. The frequency of expected (presence or absence) and observed (presence or absence) values are calculated. From the observed and expected values an error matrix is created as well as the overall accuracy, kappa statistic and chi-squared p values calculated. The `getErrorMatrix` method creates and returns an error matrix for the classification including the user's and producer's accuracies. The `getOverallAccuracy` method returns the calculated overall accuracy value whilst the `getKappaStatistic` method returns the calculated kappa statistic. The `chiSquaredTest` method calculates the chi-square probability for the classified image based on the observed and expected values.

3.5.4. *RasterImage class*

The `RasterImage` class creates and displays the classified image indicating the presence or absence of black wattle trees. Further, this class creates and displays the accuracy assessment of the classification. The `rasterVal` method returns the probability value given the row and column co-ordinates for a single pixel. The `createRaster` method stores the probability values returned from the `rasterVal` method in a two dimensional array. The `createClassifiedImage` method reads the probability values from the two dimensional array and creates the final classified image. The `createClassifiedImage` method creates a buffered image instance. Thereafter, using the graphics API each pixel is colour coded based on the probability value (white [0], blue [0.01-0.25], green [0.25-0.50], yellow [0.50-0.75], red [0.75-0.99]). The classified image file is then written in .png data format to the hard disk. On completing the classification process, the `classifiedImageFrame` method displays the final classified image (figure 3.3.). This method also displays the co-ordinates and associated probability value of a pixel selected by the end user. The co-ordinates are calculated by the `coOrdinates` method while the associated probability value is returned by the `getRaster` method. The `accuracyAssessmentFrame` method creates a frame and displays the error matrix and contingency table as well as the overall accuracy, kappa statistic and chi-squared p values (figure 3.3.).

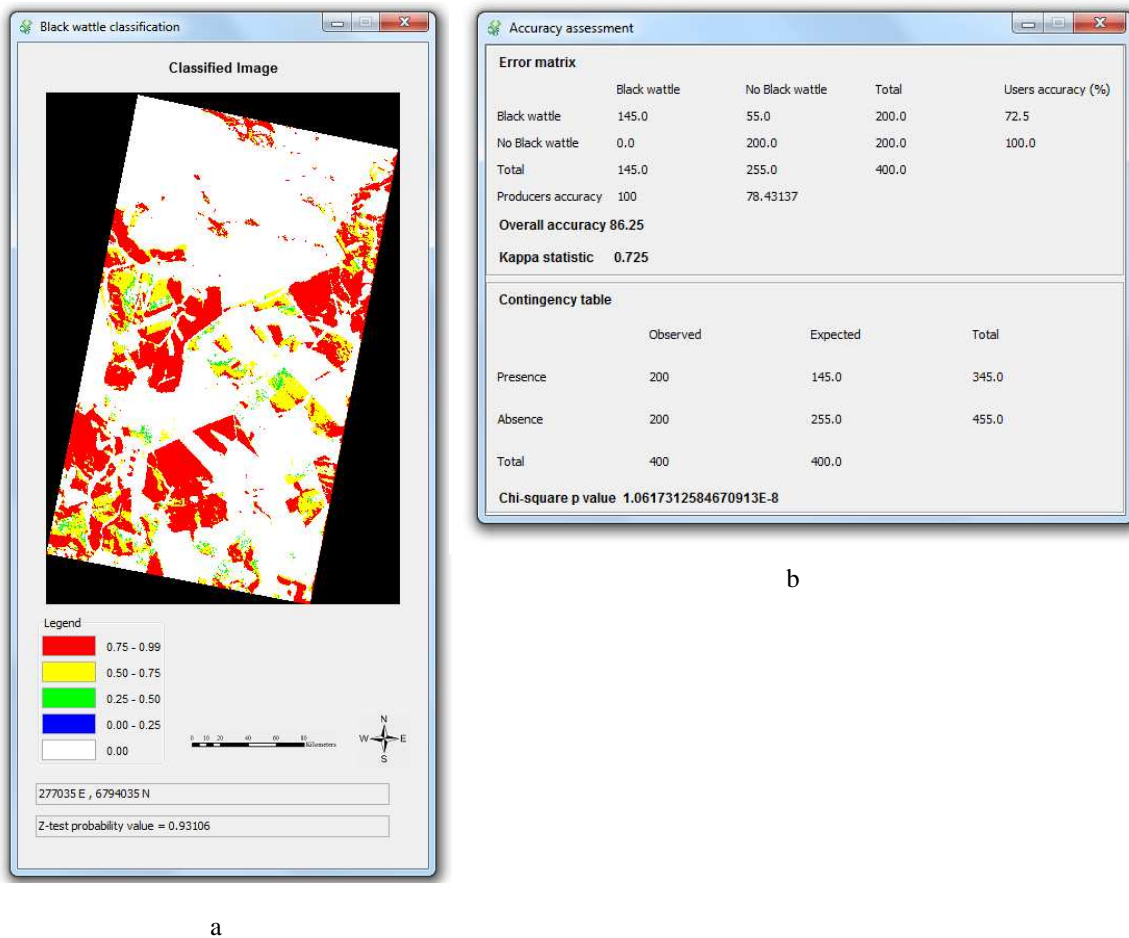


Figure 3.3.: Results generated from the classification algorithm: a) Classified image; b) Accuracy assessment.

In the following chapter an assessment of the automated approach to image classification is presented. Additionally, the results of the classifications using the general signature and age dependent signatures are presented and critically discussed.

Chapter Four: Results and Discussion

4.1. Introduction

This chapter presents the results and a detailed discussion thereof in light of the aim and objectives of this study. An assessment of the automated approach to image classification employed by the application and the implementation of the classification algorithm is presented. Subsequently, the efficacy of the classification algorithm in correctly identifying black wattle trees in general are presented and discussed. Finally, this chapter concludes with the results of the classification algorithm to identify black wattle trees of varying age groups.

4.2. Automated approach to image classification

A cost effective and easy to use specialist remote sensing application that provides an automated approach to image classification was developed for this study. The application is a stand alone program with specialist functionality i.e. the implementation of the classification algorithm (figure 3.2.). The application automatically processes the large volumes of data associated with EO-1 Hyperion satellite imagery ensuring the classification process is undertaken quickly and efficiently. Importantly, the automated approach to image classification provides access to vital information on the spatial distribution of black wattle trees quickly that would otherwise not have been available. This enables the application to be used as a monitoring tool performing routine and repeated classifications quickly. Routinely mapping and monitoring the spatial distribution of black wattle trees is essential in providing environmental managers with near real time information on the spatial distribution of black wattle trees. This information is critical in informing environmental managers where removal efforts should be targeted and how effective previous removal efforts were.

Special care was taken to ensure that the application provides simple and easy to use interfaces. Each of the interface elements were designed to be clear ensuring that user interaction with the application was kept to the minimum. This enables new end users and less skilled personnel to operate the application whilst still producing highly accurate classification maps. The application was designed to allow end users to input predefined spectral signatures of black wattle trees rather than defining training sites. This design decision obviates the need for using well trained personnel to identify training sites for

generating spectral signatures during the classification process which is time consuming and expensive to conduct.

4.3. Image classification

Through visual interpretation of the classifications, it is apparent that the classification algorithm is effective in mapping the spatial distribution of black wattle trees using imaging spectroscopy (figure 4.1.). The overall spatial distribution of black wattle trees is apparent with black wattle trees identified and dispersed throughout both study areas indicating the extent of the invasions (figure 4.1.). The classifications illustrate clearly defined homogenous areas which are the same as black wattle forestry plantations which are found throughout the study areas (figure 4.1.). Outside of the defined boundaries of forestry plantations, black wattle trees which exhibit a fragmented spatial distribution were mapped within complex land uses (figure 4.1.). A fragmented distribution in some areas was expected owing to the pixel based approach to classification employed by the classification algorithm and the unpredictable natural dispersal mechanisms that distribute black wattle trees into non native areas. Particularly, smaller pockets of black wattle trees were mapped near commercial forestry plantations indicating possible invasion into non native areas from commercial forestry plantations. Further, black wattle trees were mapped with co-occurring vegetation in regions known to be dense bush land cover in both study areas. These regions are particularly vulnerable because of the accessibility from forestry plantations and ideal growth conditions of the region. These smaller areas of invasion provide the location of invasive black wattle trees and the severity of invasion at a local spatial scale so that removal and control efforts can be targeted.

The classification algorithm was able to accurately identify and map black wattle trees throughout both study areas. An overall accuracy of 86.25% and 84.50% was achieved for study area one and study area two respectively (Table 4.1.). The kappa co-efficient was 0.72% and 0.69% for study area one and study area two respectively, which is a moderate agreement between the classification and reference data (Table 4.1.). This level of accuracy could be attributed to the stringent conditions placed upon by the classification algorithm. The statistical test was carried out within 95% confidence intervals as well as pixels with only greater than 95% of the bands per pixel being similar to the spectral signature were classified as black wattle being present. Similar results were seen by Russel (2009) in which

black wattle trees were identified and mapped using SPOT 5 imagery. An overall accuracy of 78.77% with a kappa statistic of 0.7495 was achieved in the Fort Nottingham area (Russel, 2009). Further, a study by Ustin *et al.* (2002) detected IAP species using AVIRIS data. Results included the mapping of *Arundo donax* at accuracies of 90.68% and 97.79% using a spectral angle mapper and maximum likelihood classification respectively in the Camp Pendleton Marine Base in California. The high level of accuracy achieved by the classification algorithm could be due to large concentrations of black wattle trees ensuring their detection. Further, the use of detailed spectral signatures and high spectral resolution Hyperion imagery provides more rigorous spectral matching. Smaller spectral variations are considered thereby discriminating invasive black wattle trees from complex co-occurring land uses. Within image classification, there is a constant motivation to achieve the highest level of accuracy possible. The level of accuracy of classifications can be improved by identifying invasive black wattle trees using unique morphological or physiological properties or at key phenological stages. Hestir *et al.* (2008) identified invasive vegetation (Perennial pepperweed, Water hyacinth, submerged aquatic vegetation) using hyperspectral remote sensing in the California Delta ecosystem. Results showed that both Perennial pepperweed and Water hyacinth exhibited significant spectral variation related to plant phenology. Despite this, the high overall accuracy warrants the use of the black wattle classification algorithm to identify invasive black wattle trees using hyperspectral satellite imagery.

A user's accuracy indicates the probability that a pixel classified on the map represents that category on the ground (Story and Congalton, 1986). A producer's accuracy indicates the probability of a reference pixel being correctly classified (Congalton, 1991). Both, the user's and producer's accuracy are critical in ensuring that the extent of invasive black wattle trees are not under or over estimated. Results from this study have shown that the user's accuracy for black wattle trees were 72.50% and 69.00% for study area one and study area two respectively (Table 4.1.). The classification algorithm has underestimated the presence of black wattle trees. The producer's accuracy for non black wattle trees were 78.43% and 76.33% for study area one and study area two respectively (Table 4.1.). Results from the chi-squared test showed that there was a significant difference ($p < 0.05$) between the observed and expected values for study area one and study area two (Table 4.1.). The discrepancy between the classification and reference data could be attributed to spectral confusion when classifying certain pixels. The EO-1 Hyperion sensor has a low spatial resolution. Black

wattle trees may not occupy the full extent of a pixel therefore incorporating spectra from a range of land covers into a single spectral profile resulting in spectral confusion. To detect smaller patches of invasive black wattle trees the use of higher spatial resolution may be advantageous. Using high spatial resolution imagery, land covers can be spatially separated limiting the extent of spectral confusion between land covers. Despite this, the black wattle classification algorithm provides the probability that black wattle is present within a pixel.

Results from the classification show that generally the probabilities of black wattle trees being present are moderate (0.25-0.50) to very high (0.75-0.99) within areas of potential invasion. Probabilities are very high at the centre of potential stands of invasion with probabilities decreasing from very high to moderate towards the outside of these areas. As black wattle trees become sparsely distributed towards the edges of dense clumps the probability decreases. This is due to possible spectral confusion as the land cover changes from predominantly black wattle to other land covers. The black wattle trees that have invaded non native regions occur in sparsely distributed patches of a few pixels. These regions particularly in study area two have a moderate probability. Moderate and high probability values can be misleading. Probability values could be influenced by spectral confusion resulting in lower probability values or that the probability of black wattle trees being present is actually low. Environmental managers will need to assess if valuable resources can be utilized in these areas to remove and control invasive black wattle trees. However, these areas provide insight into areas that may be experiencing the early stages of invasions. Therefore environmental managers can combat the propagation of invasive black wattle trees at the inception of the invasion.

It is concluded that the classification algorithm is adequate in identifying and mapping the spatial distribution of invasive black wattle trees using Hyperion data.

Table 4.1.: Results of the accuracy assessment for black wattle trees for study area one and study area two

Study area one						
Overall accuracy(%)	Kappa statistic	User's* accuracy(%)	Producer's* accuracy(%)	User's# accuracy(%)	Producer's# accuracy(%)	Chi square
86.25	0.72	72.50	100	100	78.43	<0.05
Study area two						
Overall accuracy(%)	Kappa statistic	User's* accuracy(%)	Producer's* accuracy(%)	User's# accuracy(%)	Producer's# accuracy(%)	Chi square
84.50	0.69	69.00	100	100	76.33	<0.05

* - indicates black wattle (*A. mearnsii*) presence

- indicates black wattle (*A. mearnsii*) absence

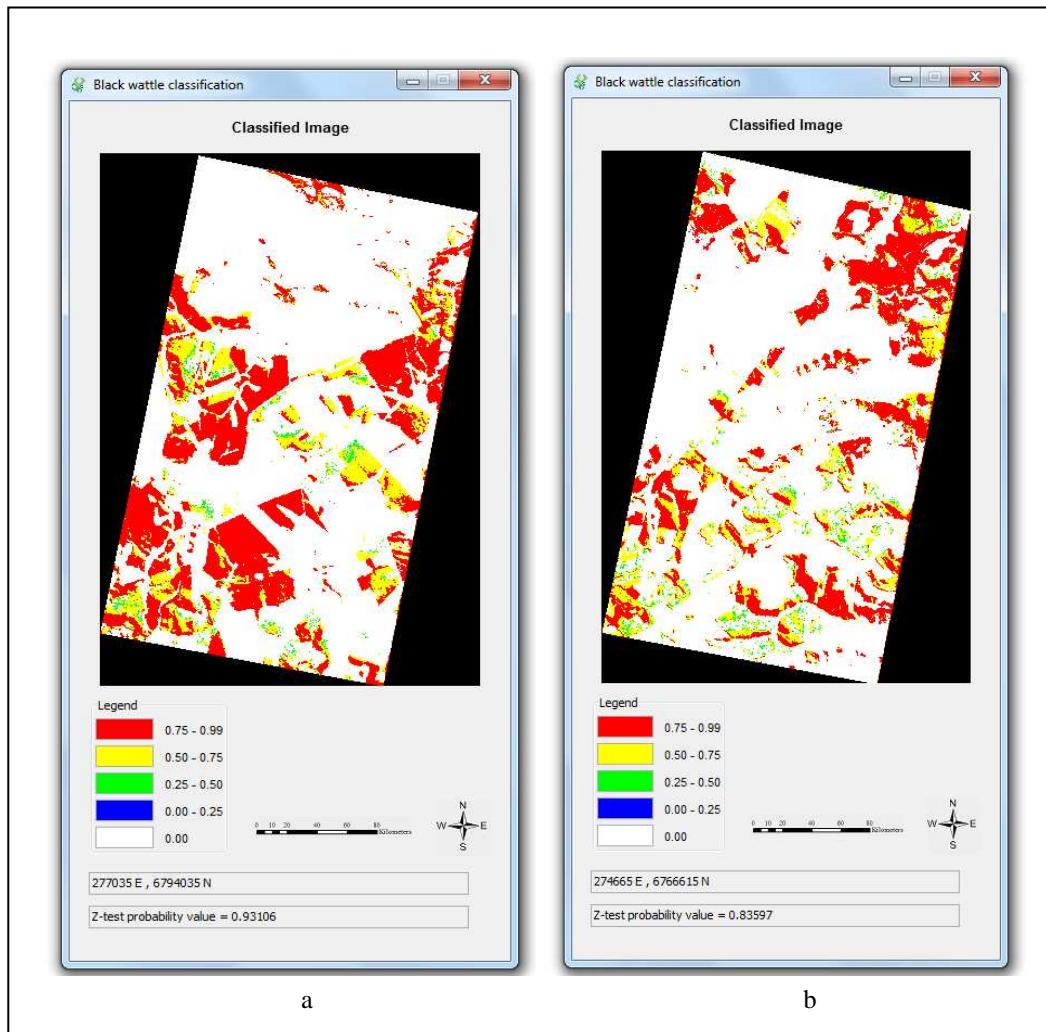


Figure 4.1.: Classified images indicating the presence and absence of black wattle (*A. mearnsii*) trees: a) Study area one; b) Study area two.

4.4. Age dependent image classification

Invasive black wattle trees are a major threat to natural ecosystems therefore removing them at a young age is critical to mitigating future negative environmental impacts. However, it is a challenge to identify young black wattle trees within non native areas using remote sensing techniques. Young black wattle trees are small in size and lack the spatial dominance at a micro spatial scale to ensure its detection. The appropriate satellite imagery (high spatial resolution, high spectral resolution) and classification algorithm must be utilized to ensure that classifications are undertaken accurately. Over time, the ageing of a tree changes its morphology, physiological status and presence within an environment; positively influencing its spectral reflectance and ability to be detected. Essentially, there is a need to assess the accuracy of age specific classifications to ensure that the youngest black wattle tree is identified with the highest of accuracies.

The classification algorithm was able to identify and map age specific black wattle trees in both study areas (figure 4.2. and 4.3.). Results from the accuracy assessment are illustrated in Table 4.2. Results show that the overall accuracy of black wattle trees between three to five years of age was 62% (kappa statistic: 0.24) for study area one and 74.50% (kappa statistic: 0.49) for study area two (Table 4.2.). Young invasive black wattle trees have small crowns that do not cover a large area which exposes the ground and surrounding vegetation to detection. EO-1 Hyperion data has a low spatial resolution (30 m) capturing reflectance from a range of scene elements within a complex environment. Therefore, the spectral profiles of pixels are not pure which could have resulted in spectral confusion. However, the high spectral resolution of collected spectral profiles would assist in discriminating between different co-occurring vegetation species. Despite this, the moderately high overall accuracy is a positive indication that young black wattle trees between three to five years of age can be identified and mapped. The overall accuracy increased for trees of three to five and seven to nine years of age which could be attributed to the growth of the plant. Results showed that the overall accuracy achieved when identifying black wattle trees between seven to nine years of age for study area one was 78.50% (kappa statistic: 0.57) (Table 4.2.). Older invasive black wattle trees proliferate and propagate rapidly establishing themselves within an environment. Black wattle trees cover a much larger area and occur in dense clumps within natural ecosystems. Their large coverage tends to eliminate non native vegetation through shade and competition. This allows them to be the dominant scene element within a pixel scene

ensuring its detection using low resolution satellite imagery. Invasive black wattle trees (seven to nine years) were detected with a moderately high accuracy; however, it is detection of emerging invasive black trees that are required. Conversely, young black wattle trees were detected with relatively lower accuracy. A balance would need to be struck between the age of the trees and an acceptable level of accuracy. Over time, invasive black wattle trees completely dominate an area which should result in detection with the highest of accuracies. However, results from this study showed that the overall accuracies and kappa statistics decreased from seven to nine years (overall accuracy: 78.50%, kappa statistic: 0.57) to eleven to thirteen years (overall accuracy: 69.50%, kappa statistic: 0.39) of age for study area one (Table 4.2.). Similarly, overall accuracies and kappa statistics decreased from seven to nine years (overall accuracy: 75%, kappa statistic: 0.50) to eleven to thirteen years (overall accuracy: 72.50%, kappa statistic: 0.45) of age for study area two (Table 4.2.). Older trees are more susceptible to environmental stress such as lack of water availability and nutrients supply at a local scale. These physiological stresses would alter the spectral profile of the trees even though they may appear healthy, consequently negatively influencing its detection. Despite this, the overall accuracy of black wattle trees between eleven to thirteen years age was still higher than the overall accuracy achieved by the three to five years age group classification. Generally, the relatively high overall accuracies are a positive indication that the classification algorithm can identify black wattle trees of varying age groups.

Although the overall accuracy of the classifications were relatively high the user's accuracy were very low (20%) to moderately (60%) high for all age groups for both study areas. A user's accuracy of 24%, 56.99% and 39% was achieved for black wattle presence for the three to five, seven to nine and eleven to thirteen age groups respectively for study area one (Table 4.2.). Further, a user's accuracy of 49%, 50% and 49% was achieved for black wattle presence for the three to five, seven to nine and eleven to thirteen age groups respectively for study area two (Table 4.2.). Results from the chi-squared test showed that there was a significant difference ($p < 0.05$) between the observed and expected values for both study areas for all age groups (Table 4.2.). The low user's accuracy for black wattle trees between three to five years of age could be attributed to spectral confusion. Young invasive black wattle trees are small in size which exposes the ground and surrounding vegetation to detection. EO-1 Hyperion data has a low spatial resolution (30 m) capturing reflectance from a range of scene elements within a complex environment. Therefore, the spectral profiles of pixels are not pure which could have resulted in spectral confusion. However, the user's

accuracy should have been higher for black wattle trees between seven to nine and eleven to thirteen years of age. Variability in the spatial distribution, density and percentage cover that black wattle trees occupy within a pixel could have negatively influenced its detection. The use of higher spatial resolution satellite imagery would be better suited to limit the variability of scene elements within a pixel thus enhancing the detection of black wattle trees. Therefore, the moderate results achieved by this study are attributed to inappropriate satellite imagery used for age specific classifications and not the efficacy of the classification algorithm. A producer's accuracy of 56.81%, 69.93% and 62.11% was achieved for black wattle absence for the three to five, seven to nine and eleven to thirteen age groups respectively for study area one (Table 4.2.). Further, a producer's accuracy of 66.22%, 66.66% and 64.51% was achieved for black wattle absence for the three to five, seven to nine and eleven to thirteen age groups respectively for study area two (Table 4.2.). Indirectly, this level of accuracy could be attributed to the stringent conditions placed upon by the classification algorithm. Pixels may not have been classified owing to the rigorous spectral matching employed by the classification algorithm leading to a great number of pixels being classified as black wattle absence.

In summary, the classification algorithm can identify and map age specific black wattle trees with a relatively high overall accuracy. However, the user's accuracy for age specific black wattle classifications was poor. The low spatial resolution of EO-1 Hyperion data could have resulted in spectral confusion and increased variability within pixels. Therefore, the low user's accuracies could be attributed to the use of relatively low resolution satellite imagery and not the efficacy of the classification algorithm. However, the high producer's accuracies for black wattle absence could be attributed to the stringent conditions of the classification algorithm resulting in the misclassification of black wattle trees. It is concluded that the classification algorithm is adequate in identifying and mapping the spatial distribution of age specific black wattle trees using imaging spectroscopy.

Table 4.2.: Results of the accuracy assessment for all age groups for study area one and study area two

Study area one							
Age group	Overall accuracy(%)	Kappa statistic	User's [*] accuracy(%)	Producer's [*] accuracy(%)	User's [#] accuracy(%)	Producer's [#] accuracy(%)	Chi square
3-5	62.00	0.24	24.00	100	100	56.81	<0.05
7-9	78.50	0.57	56.99	100	100	69.93	<0.05
11-13	69.50	0.39	39.00	100	100	62.11	<0.05
Study area two							
Age group	Overall accuracy(%)	Kappa statistic	User's [*] accuracy(%)	Producer's [*] accuracy(%)	User's [#] accuracy(%)	Producer's [#] accuracy(%)	Chi square
3-5	74.50	0.49	49.00	100	100	66.20	<0.05
7-9	75.00	0.50	50.00	100	100	66.66	<0.05
11-13	72.50	0.45	49.00	100	100	64.51	<0.05

^{*} - indicates black wattle (*A. mearnsii*) presence

[#] - indicates black wattle (*A. mearnsii*) absence

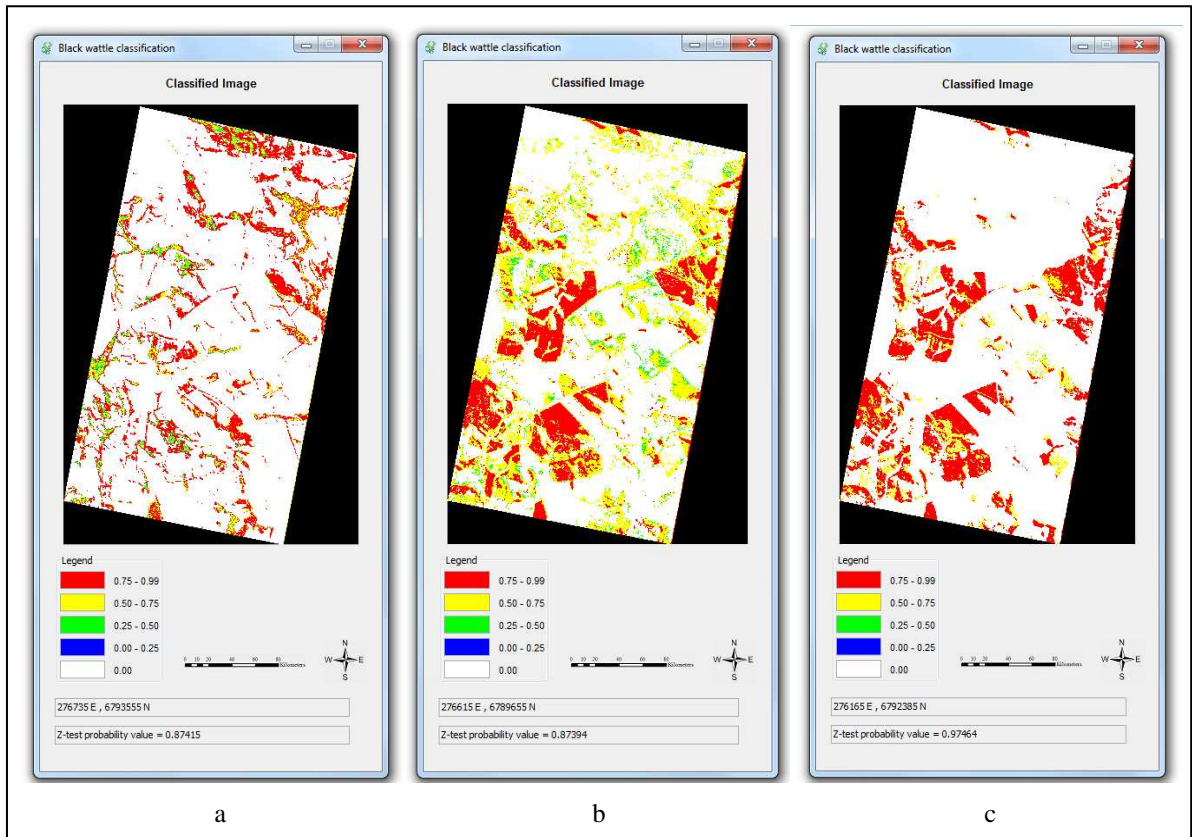


Figure 4.2.: Classified images indicating the presence and absence of black wattle (*A. mearnsii*) trees at three to five, seven to nine and eleven to thirteen years of age for study area one: a) Three to five; b) Seven to nine; c) Eleven to thirteen.

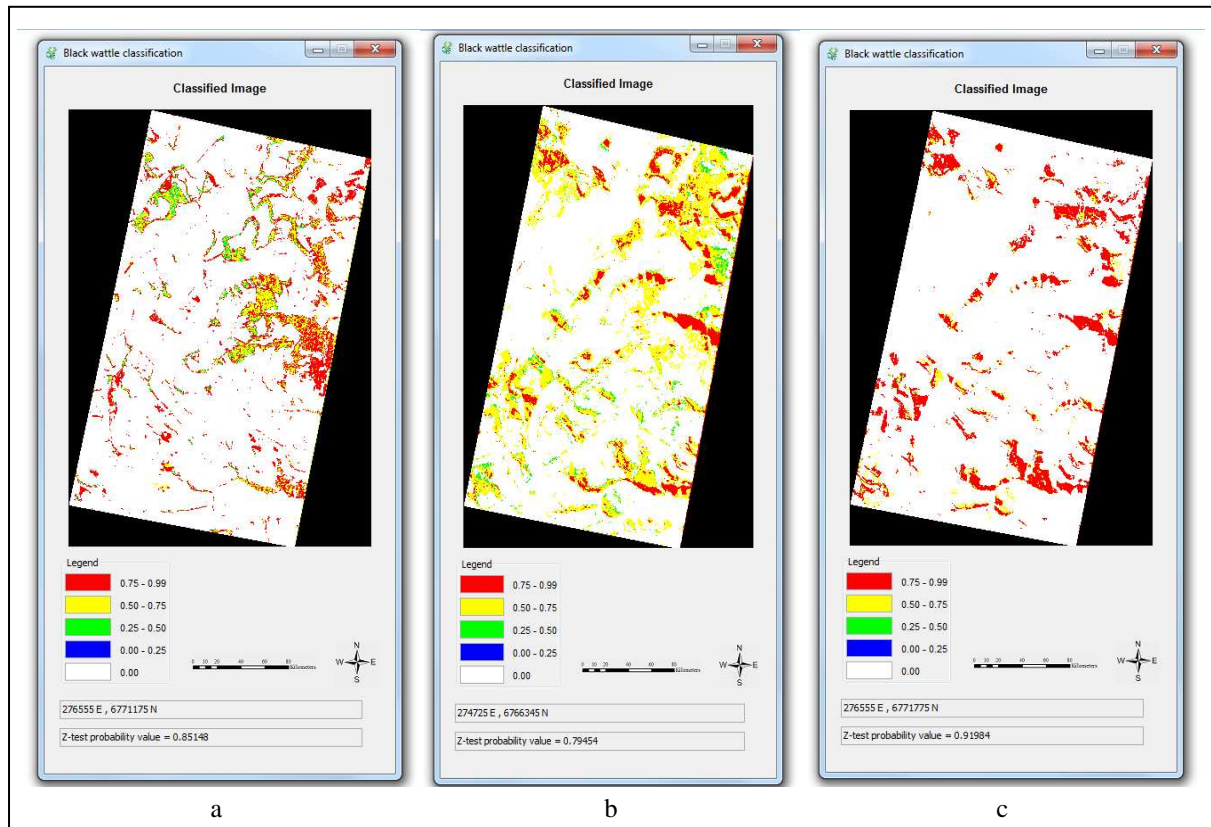


Figure 4.3.: Classified images indicating the presence and absence of black wattle (*A. mearnsii*) trees at three to five, seven to nine and eleven to thirteen years of age for study area two: a) Three to five; b) Seven to nine; c) Eleven to thirteen.

Chapter Five: Conclusions and Recommendations

5.1. Introduction

This study aimed to determine the potential of a classification algorithm to identify black wattle (*Acacia mearnsii* De Wild.) trees using imaging spectroscopy. This chapter assesses the aim and related objectives to determine if they were achieved within the framework of this study. Subsequently, the limitations of this study are presented and evaluated. This chapter concludes with the recommendations for future studies.

5.2. Aim and objectives reviewed

This study aims to investigate the potential of a classification algorithm to identify black wattle (*Acacia mearnsii* De Wild.) trees using imaging spectroscopy. In order to achieve this aim the following objectives were identified:

- To develop an image classification algorithm that will identify black wattle (*Acacia mearnsii* De Wild.) trees using hyperspectral EO-1 Hyperion data.

This was achieved by developing a specialist remote sensing application in the Java programming language. The application was programmed to process EO-1 Hyperion header files, EO-1 Hyperion data files, spectral signature files (.txt) and ground reference data files (.txt). A pixel based classification algorithm was developed based on the z test statistical test. The classification algorithm was a pixel based classifier for hyperspectral EO-1 Hyperion data. The classification algorithm was implemented within the application to produce classified images that indicate the presence or absence of black wattle trees over the area of interest.

- To assess the image classification algorithm's ability to automate the classification process.

This was achieved by creating a simple, clear and easy-to-use user interface that facilitated the input of data into the classifier quickly. Throughout this process, the user interface ensured that user interaction was kept to the minimum. Importantly, the application

implemented the classification algorithm which automatically conducted the classification process quickly and efficiently. Further, the classification algorithm automatically creates a map indicating the presence and absence of black wattle trees with associated accuracy assessment for users to interpret. It was concluded that the classification algorithm can automatically identify and map black wattle trees using EO-1 Hyperion data.

- To assess the efficacy of the image classification algorithm in identifying black wattle (*Acacia mearnsii* De Wild.) trees using hyperspectral EO-1 Hyperion data.

This was achieved by applying the classification algorithm to EO-1 Hyperion data (study area one, study area two) using a general spectral signature of black wattle trees. The application produced a classified image with an associated accuracy assessment of the classification. Results showed an overall accuracy of 86.25% (users accuracy: 72.50%; kappa 0.725) and 84.50% (users accuracy: 69%; kappa: 0.69) for study area one and study area two respectively. It was concluded that the classification algorithm is adequate in identifying and mapping the spatial distribution of invasive black wattle trees using imaging spectroscopy in KwaZulu-Natal, South Africa.

- To assess the accuracy of the image classification algorithm to identify black wattle (*Acacia mearnsii* De Wild.) trees of varying age groups using hyperspectral EO-1 Hyperion data.

This was achieved by applying the classification algorithm to EO-1 Hyperion data (study area one, study area two) using age specific spectral signatures of black wattle trees. Black wattle trees between three to five years of age were classified with the lowest overall accuracy and user's accuracy. The highest overall accuracy and user's accuracy was achieved when identifying black wattle trees between seven to nine years of age. It was concluded that the classification algorithm is adequate in identifying and mapping the spatial distribution of age specific black wattle trees using imaging spectroscopy. It was also concluded that black wattle trees between seven and nine years of age are optimal for remote sensing with a high accuracy.

5.3. A synthesis

This study has illustrated the potential of a classification algorithm to identify black wattle trees using imaging spectroscopy. The application provides an automated approach to image classification, through the implementation of the classification algorithm. This study has shown that the classification algorithm can identify black wattle trees with a very high overall accuracy (>84%) using a general spectral signature for both study areas. Further, this study has shown that the classification algorithm can identify black wattle trees of varying age groups using age specific spectral signatures of black wattle trees for both study areas. However, black wattle trees between three to five years of age were classified with the lowest overall accuracy and user's accuracy. The low user's accuracies could be attributed to the use of relatively low resolution satellite imagery and not the efficacy of the classification algorithm. This is disadvantageous as it prevents young black wattle trees from being detected and removed before causing significant negative environmental impacts in non native areas. However, the highest overall accuracy and user's accuracy was achieved when identifying black wattle trees between seven to nine years of age. Black wattle trees between seven and nine years of age are optimal for remote sensing with a high accuracy. Importantly, this study has illustrated that invasive black wattle trees can be identified accurately using remote sensing techniques. There is also great potential for using this developed algorithm for the identification and mapping of other plant species. This may require the capture of the spectral signature of those plants and their incorporation into the algorithm.

5.4. Limitations of this study

One of the limitations of this study was that the application and classification algorithm was developed to process EO1-Hyperion data only. Further, EO-1 Hyperion data has a relatively low spatial resolution and invasive black wattle trees may not occur in large enough clumps to be detected thus resulting in spectral confusion. The classification algorithm did not incorporate subpixel spectral un-mixing to ensure classifications of the highest of accuracies. During the classification process, ground reference data was collected from thematic maps of the study areas. Inherent errors that were present within the thematic map could have propagated through the classification process negatively affecting the accuracy assessment of the classification.

5.5. Recommendations for future studies

This study has illustrated the potential of a classification algorithm to identify black wattle trees using EO1-Hyperion data. However, the application should be developed further to process a range of satellite images from different platforms. The application should be able to process multi-spectral data as it is freely and widely available. The application should also incorporate subpixel spectral un-mixing ensuring classifications of the highest of accuracies. Future studies should focus on assessing the classification algorithms performance classifying different satellite images from different sensors. Studies should focus on detecting invasive black wattle trees using hyperspectral imagery of a higher spatial resolution so that spectral confusion can be minimized. This study employed the use of general and age specific spectral signatures for black wattle trees. Future studies should identify unique spectral signatures to identify black wattle trees at key morphological and phenological stages of plant development thus enhancing its detection. Further, future studies should focus on developing predictive models to model the spread of invasive black wattle trees. This would allow environmental managers to target and implement control efforts in areas before they become severely invaded. This study has taken the first steps in achieving this goal as black wattle trees can be identified and mapped using imaging spectroscopy.

References

- Afrispace consulting, 2009. Umvoti spatial development framework. Umvoti Local Municipality, Durban, available online at: http://devplan.kzntl.gov.za/idp_reviewed_2010_11/IDPS/KZ245/Adopted/Revised%20Mvoti%20SDF.pdf (accessed on 15 June 2011).
- Agüera, F., Liu, J.G., 2009. Automatic greenhouse delineation from Quickbird and Ikonos satellite images. *Computers and Electronics in Agriculture* 66. 191-200.
- Allan, D.G., Harrison, J.A., Navarro, R.A., van Wilgen, B.W., Thompson, M.W., 1997. The impact of commercial afforestation on bird populations in Mpumalanga Province, South Africa-insights from bird-atlas data. *Biological Conservation* 79, 173-185.
- Asner, G.P., Jones, M.O., Martin, R.E., Knapp, D.E., Hughes, R.F., 2008. Remote sensing of native and invasive species in Hawaiian forests. *Remote Sensing of Environment* 112, 1912-1926.
- Babugura, A., 2010. Gender and climate change: South Africa case study. Heinrich Böll Foundation Southern Africa, Cape Town, South Africa.
- Bauhus, J., van Winden, A.P., Nicotra, A.B., 2004. Aboveground interactions and productivity in mixed-species plantations of *Acacia mearnsii* and *Eucalyptus globulus*. *Canadian Journal of Forest Research* 34, 686-694.
- Binns, J.A., Illgner, P.M., Nel, E.L., 2001. Water shortage, deforestation and development: South Africa's working for water programme. *Land Degradation and Development* 12, 341-355.
- Brown, A.G., Ko, H.C., 1997. Black wattle and its Utilisation – Abridged English Version. Rural Industries Research and Development Corporation, Barton Act, Australia.

- Buerkert, A., Mahler, F., Marschner, H., 1996. Soil productivity management and plant growth in the Sahel: potential of an aerial monitoring technique. *Plant and Soil* 180, 29-38.
- Carson, H.W., Lass, L.W., Callihan, R.H., 1995. Detection of yellow hawkweed (*Hieracium pratense*) with high resolution multispectral digital imagery. *Weed technology* 9, 477-483.
- Carter, G.A., Lucas, K.L., Blossom, G.A., Lassitter, C.L., Holiday, D.M., Mooneyhan, D.S., Fastring, D.R., Holcombe, T.R., Griffith, J.A., 2009. Remote sensing and mapping of tamarisk along the Colorado River, USA: a comparative use of summer-acquired Hyperion, Thematic Mapper and QuickBird data. *Remote Sensing* 1, 318-329.
- Chaunbi, G., 1997. Black wattle plantations in South Africa: introduction, silviculture and management. In: Brown, A.G., Ko, H.C. (Eds.), *Black wattle and its Utilisation – Abridged English Version*. Rural Industries Research and Development Corporation, Barton Act, Australia, pp. 33-38.
- Christina, M., Laclau, J.P., Gonçalves, J.L.M., Jourdan, C., Nouvellon, Y., Bouillet, J.P., 2011. Almost symmetrical vertical growth rates above and below ground in one of the world's most productive forests. *Ecosphere* 2, 1-10.
- Coetzee, J.A., Byrne, M.J., Hill, M.P., 2007. Impact of nutrients and herbivory by *Eccritotarsus catarinensis* on the biological control of water hyacinth, *Eichhornia crassipes*. *Aquatic Botany* 86, 179-186.
- Congalton, R.G., 1991. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment* 37, 35-46.
- Cullis, J.D.S., Görgens, A.H.M., Marais, C., 2007. A strategic study of the impact of invasive alien plants in the high rainfall catchments and riparian zones of South Africa on total surface water yield. *Water SA* 33, 35-42.
- Cuneo, P., Jacobson, C.R., Leishman, M.R., 2009. Landscape scale detection and mapping of

- invasive African Olive (*Olea europaea* L. ssp. *cuspidata* Wall ex G. Don Ciferri) in SW Sydney, Australia using satellite remote sensing. *Applied Vegetation Science* 12, 145-154.
- De Beer, H., 1986. Black wattle. *Forest molecular genetics South Africa Weeds A.24*. In: Viljoen, B.D., Stoltz, C.W., 2008. Control of black wattle (*Acacia mearnsii* De Wild.) seedlings with Garlon herbicide applied by backpack mistblower. *South African Journal of Plant and Soil* 25, 242-244.
- de Neergaard, A., Saarnak, C., Hill, T., Khanyile, M., Berzosa, A.M., Birch-Thomsen, T., 2005. Australian wattle species in the Drakensberg region of South Africa-an invasive alien or a natural resource? *Agricultural Systems* 85, 216-233.
- de Wit, M.P., Crookes, D.J., van Wilgen, B.W., 2001. Conflicts of interest in environmental management: estimating the costs and benefits of a tree invasion. *Biological Invasions* 3, 167-178.
- Di Stefano, J., 2002. River red gum (*Eucalyptus camaldulensis*): a review of ecosystem processes, seedling regeneration and silvicultural practice. *Australian Forestry* 65, 14-22.
- Díaz, S., Fargione, J., Chapin III, F.S., Tilman, D., 2006. Biodiversity loss threatens human well-being. *PLoS Biology* 4, e277.
- du Toit, B., Dovey, S.B., 2005. Effect of site management on leaf area, early biomass development, and stand growth efficiency of a *Eucalyptus grandis* plantation in South Africa. *Canadian Journal of Forest Research* 35, 891-900.
- Dye, P., Jarman, C., 2004. Water use by black wattle (*Acacia mearnsii*): implications for the link between removal of invading trees and catchment streamflow response. *South African Journal of Science* 100, 40-44.
- Eldridge, K.G., 1978. Genetic improvement of eucalypts. *Silvae Genetica* 27, 205-209.

- Eldridge, K., Davidson, J., Harwood, C., van Wyk, G., 1993. Eucalypt domestication and breeding. Oxford University Press Inc., New York, United States.
- Enright, W.D., 2000. The effect of terrestrial invasive alien plants on water scarcity in South Africa. *Physics Chemistry Earth (B)* 25, 237-242.
- ENVI, 2006. Environment for Visualising Images. Release 4.7 (Boulder, USA. ITT industries, Inc).
- Everitt, J.H., Yang, C., Helton, R.J., Hartmann, L.H., Davis, M.R., 2002. Remote sensing of giant salvinia in Texas waterways. *Journal of Aquatic Plant Management* 40, 11-16.
- Forsyth, G.G., Richardson, D.M., Brown, P.J., van Wilgen, B.W., 2004. Rapid assessment of the invasive status of *Eucalyptus* species in two South African provinces. *South African Journal of Science* 100, 75-77.
- Geldenhuys, J.C., 1997. Native forest regeneration in pine and eucalypt plantations in Northern Province, South Africa. *Forest Ecology and Management* 99, 101-115.
- Ghebremicael, S.T., Smith, C.W., Ahmed, F.B., 2004. Estimating the leaf area index (LAI) of black wattle from Landsat ETM+ satellite imagery. *Southern African Forestry Journal* 201, 3-12.
- Görgens, A.H.M., van Wilgen, B.W., 2004. Invasive alien plants and water resources in South Africa: current understanding, predictive ability and research challenges. *South African Journal of Science* 100, 27-33.
- Govender, M., Chetty, K., Bulcock, H., 2007. A review of hyperspectral remote sensing and its application in vegetation and water resource studies. *Water SA* 33, 145-151.
- Greenfield, B.K., Siemering, G.S., Andrews, J.C., Rajan, M., Andrews, S.P., Spencer, D.F., 2007. Mechanical shredding of water hyacinth (*Eichhornia crassipes*): effects on water quality in the Sacramento-San Joaquin River Delta, California. *Estuaries and Coasts* 30, 627-640.

- Grün, A., 2000. Semi-automated approaches to site recording and modeling. *International Archives of Photogrammetry and Remote Sensing* 33, 309-318.
- He, K.S., Rocchini, D., Neteler, M., Nagendra, H., 2011. Benefits of hyperspectral remote sensing for tracking plant invasions. *Diversity and Distributions* 17, 381-392.
- Hestir, E.L., Khanna, S., Andrew, M.E., Santos, M.J., Viers, J.H., Greenberg, J.A., Rajapakse, S.S., Ustin, S.L., 2008. Identification of invasive vegetation using hyperspectral remote sensing in the California Delta ecosystem. *Remote Sensing of Environment* 112, 4034-4047.
- Higgins, J., Harris, A., 1997. VAST: a program to locate and analyse volcanic thermal anomalies automatically from remotely sensed data. *Computers and Geosciences* 23, 627-645.
- Holmes, P.M., Esler, K.J., Richardson, D.M., Witkowski, E.T.F., 2008. Guidelines for improved management of riparian zones invaded by alien plants in South Africa. *South African Journal of Botany* 74, 538-552.
- Impson, F.A.C., Kleinjan, C.A., Hoffmann, J.H., Post, J.A., 2008. *Dasineura rubiformis* (Diptera: Cecidomyiidae), a new biological control agent for *Acacia mearnsii* in South Africa. *South African Journal of Science* 104, 247-249.
- Jarmain, C., Everson, C.S., 2002. Comparative evaporation measurements above commercial forestry and sugarcane canopies in the KwaZulu-Natal midlands. CSIR Report ENV-C-S 2002 005. CSIR, Pretoria. In: Dye, P., Jarmain, C., 2004. Water use by black wattle (*Acacia mearnsii*): implications for the link between removal of invading trees and catchment streamflow response. *South African Journal of Science* 100, 40-44.
- Jones, M.E., Shepherd, M., Henry, R., Delves, A., 2008. Pollen flow in *Eucalyptus grandis* determined by paternity analysis using microsatellite markers. *Tree Genetics and Genomes* 4, 37-47.

- Joshi, C., de Leeuw, J., van Duren, I.C., 2004. Remote sensing and GIS applications for mapping and spatial modeling of invasive species. *Proceedings of the International Society for Photogrammetry and Remote Sensing* 35, 669-677.
- Julien, M.H., Griffiths, M.W., Wright, A.D., 1999. Biological control of water hyacinth. The weevils *Neochetina bruchi* and *N. eichhorniae*: biologies, host ranges, and rearing, releasing and monitoring techniques for biological control of *Eichhornia crassipes*. ACIAR Monograph No. 60, 87 pp.
- Khanna, P.K., 1997. Comparison of growth and nutrition of young monocultures and mixed stands of *Eucalyptus globulus* and *Acacia mearnsii*. *Forest Ecology and Management* 94, 105-113.
- Kimothi, M.M., Anitha, D., Vasistha, H.B., Soni, P., Chandola, S.K., 2010. Remote sensing to map the invasive weed, *Lantana camara* in forests. *Tropical Ecology* 51, 67-74.
- Kruce, F., Lefkoff, A., Boardman, J., Heidebrecht, K., Shapiro, A., Barloon, P., Goetz, A., 1993. The spectral image processing system (SIPS) interactive visualization and analysis of imaging spectrometer data. *Remote Sensing of Environment* 44, 145-163.
- In: Govender, M., Chetty, K., Bulcock, H., 2007. A review of hyperspectral remote sensing and its application in vegetation and water resource studies. *Water SA* 33, 145-151.
- Kull, C.A., Rangan, H., 2008. *Acacia* exchanges: wattles, thorn trees, and the study of plant movements. *Geoforum* 39, 1258-1272.
- Landsberg, J.J., Waring, R.H., Coops, N.C., 2003. Performance of the forest productivity model 3-PG applied to a wide range of forest types. *Forest Ecology and Management* 172, 199-214.
- Lass, L.W., Prather, T.S., Glenn, N.F., Weber, K.T., Mundt, J.T., Pettingill, J., 2005. A review of remote sensing of invasive weeds and example of the early detection of spotted knapweed (*Centaurea maculosa*) and babysbreath (*Gypsophila paniculata*) with a hyperspectral sensor. *Weed Science* 53, 242-251.

- Lawrence, W., Lass, T.S., Prather, N.F., Glenn, K.T., Weber, J.T., Mundt, J.P., 2005. A review of remote sensing of invasive weeds and example of the detection of spotted knapweed (*Centaurea maculosa*) and babysbreath (*Gypsophila paniculata*) with a hyperspectral sensor. *Weed Science* 53, 242-251.
- Le Maitre, D.C., Versfeld, D.B., Chapman, R.A., 2000. Impact of invading alien plants on surface water resources in South Africa: a preliminary assessment. *Water SA* 26, 397-408
- Le Maitre, D.C., van Wilgen, B.W., Gelderblom, C.M., Bailey, C., Chapman, R.A., Nel, J.A., 2002. Invasive alien trees and water resources in South Africa: case studies of the costs and benefits of management. *Forest Ecology and Management* 160, 143-159.
- Leckie, D.G., Gougeon, F.A., Walsworth, N., Paradine, D., 2003. Stand delineation and composition estimation using semi-automated individual tree crown analysis. *Remote Sensing of Environment* 85, 355-369.
- Lodge, D.M., 1993. Biological invasions: lessons for ecology. *Trends in Ecology and Evolution* 8, 133-137.
- Maiden, J.H., 1891. Wattles and wattle-barks being hints on the conservation and cultivation of wattles. Department of Public Instruction, Technical Education Branch. Government Printer, Sydney. In: Searle, S.D., 1997. *Acacia mearnsii* De Wild. (black wattle) in Australia. In: Brown, A.G., Ko, H.C. (Eds.), *Black wattle and its Utilisation – Abridged English Version*. Rural Industries Research and Development Corporation, Barton Act, Australia, pp. 1-13.
- Marais, C., Wannenburgh, A.M., 2008. Restoration of water resources (natural capital) through the clearing of invasive alien plants from riparian areas in South Africa-costs and water benefits. *South African Journal of Botany* 74, 526-537.
- Mathur, S.M., Singh, P., 2004. Development and performance evaluation of a water hyacinth chopper *cum* crusher. *Biosystems Engineering* 88, 411-418.

- Mladinich, C.S., Bustos, M.R., Stitt, S., Root, R., Brown, K., Anderson, G.L., Hager, S., 2006. The use of Landsat 7 Enhanced Thematic Mapper Plus for mapping leafy spurge. *Rangeland Ecology and Management* 59, 500-506.
- Mutanga, O., van Aardt, J., Kumar, L., 2009. Imaging spectroscopy (hyperspectral remote sensing) in southern Africa: an overview. *South African Journal of Science* 105, 193-198.
- Niu, X., 2006. A semi-automated framework for highway extraction and vehicle detection based on a geometric deformable model. *ISPRS Journal of Photogrammetry and Remote Sensing* 61, 170-186.
- Oumar, Z., 2008. Field spectroscopy of plant water content in *Eucalyptus grandis* forest stands in KwaZulu-Natal, South Africa. MSc thesis, School of Environmental Sciences, University of KwaZulu-Natal, Pietermaritzburg, South Africa.
- Parker Williams, A., Hunt, Jr., E.R., 2002. Estimation of leafy spurge cover from hyperspectral imagery using mixture tuned matched filtering. *Remote Sensing of Environment* 82, 446-456.
- Pearlman, J.S., Barry, P.S., Segal, C.C., Shepanski, J., Beiso, D., Carman, S.L., 2003. Hyperion, a space-based imaging spectrometer. *IEEE Transactions on Geoscience and Remote Sensing* 41, 1160-1173.
- Poynton, R.J., 1979. Tree planting in Southern Africa. Vol. 2. The Eucalyptus. Department of Forestry, Pretoria. In: Forsyth, G.G., Richardson, D.M., Brown, P.J., van Wilgen, B.W., 2004. Rapid assessment of the invasive status of *Eucalyptus* species in two South African provinces. *South African Journal of Science* 100, 75-77.
- Prinsloo, F.W., Scott, D.F., 1999. Streamflow responses to the clearing of alien invasive trees from riparian zones at three sites in the Western Cape Province. *The Southern African Forestry Journal* 185, 1-7.
- Richardson, D.M., van Wilgen, B.W., 2004. Invasive alien plants in South Africa: how well

- do we understand the ecological impacts? South African Journal of Science 100, 45-52.
- Richardson, W., McGuire, D., Grey, N., Preston, C., Watts, J., Crossman, N., 2006. Woody weed control using metsulfuron-methyl and glyphosate applied by boomless spray technology prior to clear fell of radiata pine (*Pinus radiata*). In: Preston, C., Watts, J.H., Crossman, N.D. (Eds.), 15th Australian Weeds Conference Papers and Proceedings: Managing Weeds in a Changing Climate, Adelaide, South Australia, 24–28 September 2006, Weed Management Society of South Australia Proceedings, 890-893.
- Rose, S., Fairweather, P.G., 1997. Changes in floristic composition of urban bushland invaded by *Pittosporum undulatum* in northern Sydney, Australia. Australian Journal of Botany 45, 123-149. In: Lake, J.C., Leishman, M.R., 2004. Invasion success of exotic plants in natural ecosystems: the role of disturbance, plant attributes and freedom from herbivores. Biological Conservation 117, 215-226.
- Rowlinson, L.C., Summerton, M., Ahmed, F., 1999. Comparison of remote sensing data sources and techniques for identifying and classifying alien invasive vegetation in riparian zones. Water SA 25, 497-500.
- Russel, C., 2009. Investigating the utility of SPOT 5 imagery and artificial neural networks, in the identification and mapping of *Acacia mearnsii* within environments of varying complexity. MSc thesis, School of Environmental Sciences, University of KwaZulu-Natal, Pietermaritzburg, South Africa.
- Samways, M.J., Taylor, S., 2004. Impacts of invasive alien plants on red-listed South African dragonflies (Odonata). South African Journal of Science 100, 78-80.
- Scott, D.F., 1997. The contrasting effects of wildfire and clearfelling on the hydrology of a small catchment. Hydrological Processes, In press. In: Scott, D.F., Lesch, W., 1997. Streamflow responses to afforestation with *Eucalyptus grandis* and *Pinus patula* and to felling in the Mokobulaan experimental catchments, South Africa. Journal of Hydrology 199, 360-377.
- Scott, D.F., Lesch, W., 1997. Streamflow responses to afforestation with *Eucalyptus grandis*

- and *Pinus patula* and to felling in the Mokobulaan experimental catchments, South Africa. *Journal of Hydrology* 199, 360-377.
- Searle, S.D., 1997. *Acacia mearnsii* De Wild. (black wattle) in Australia. In: Brown, A.G., Ko, H.C. (Eds.), *Black wattle and its Utilisation – Abridged English Version*. Rural Industries Research and Development Corporation, Barton Act, Australia, pp. 1-13.
- Story, M., Congalton, R.G., 1986. Accuracy assessment: a user's perspective. *Photogrammetric Engineering and Remote Sensing* 52, 397–399.
- Tesfamichael, S.G., van Aardt, J.A.N., Ahmed, F., 2010. Estimating plot-level tree height and volume of *Eucalyptus grandis* plantations using small-footprint, discrete return lidar data. *Progress in Physical Geography* 34, 515-540.
- Tsai, F., Lin, E.K., Yoshino, K., 2007. Spectrally segmented principal component analysis of hyperspectral imagery for mapping invasive plant species. *International Journal of Remote Sensing*. 28, 1023-1039.
- Umvoti municipality, 2008. Umvoti municipality 2008/2009 integrated development plan. Umvoti Local Municipality, Greytown, available online at: <http://kzntopbusiness.co.za/site/umvoti-municipality> (accessed on 15 June 2011).
- Umzinyathi municipality, 2010. Umzinyathi district municipality: annual report 2009/2010. Umzinyathi District Municipality, Dundee, available online at: http://www.umzinyathi.gov.za/umzinyathi/index.php?option=com_docman&Itemid=131 (accessed on 15 June 2011).
- Ustin, S.L., DiPietro, D., Olmstead, K., Underwood, E., Scheer, G.J., 2002. Hyperspectral remote sensing for invasive species detection and mapping. *Proceedings of the 2002 IEEE International Geoscience and Remote Sensing Symposium* 3, 1658-1660.
- van Wilgen, B.W., Richardson, D.M., Le Maitre, D.C., Marais, C., Magadlela, D., 2001. The economic consequences of alien plant invasions: examples of impacts and approaches to sustainable management in South Africa. *Environment, Development and*

Sustainability 3, 145-168.

- van Wilgen, B.W., de Wit, M.P., Anderson, H.J., Le Maitre, D.C., Kotze, I.M., Ndala, S., Brown, B., Rapholo, M.B., 2004. Costs and benefits of biological control of invasive alien plants: case studies from South Africa. *South African Journal of Science* 100, 113-122.
- Vane, G., Goetz, A.F.H., 1993. Terrestrial imaging spectroscopy. *Remote Sensing of Environment* 24, 1–29. In: He, K.S., Rocchini, D., Neteler, M., Nagendra, H., 2011. Benefits of hyperspectral remote sensing for tracking plant invasions. *Diversity and Distributions* 17, 381-392.
- Verma, R., Singh, S.P., Ganesha, R.A.J., 2003. Assessment of changes in water hyacinth coverage of water bodies in northern part of Bangalore city using temporal remote sensing data. *Current Science* 84, 795-804.
- Viljoen, B.D., Stoltz, C.W., 2008. Control of black wattle (*Acacia mearnsii* De Wild.) seedlings with Garlon herbicide applied by backpack mistblower. *South African Journal of Plant and Soil* 25, 242-244.
- Villamagna, A.M., Murphy, B.R., 2010. Ecological and socio-economic impacts of invasive water hyacinth (*Eichhornia crassipes*): a review. *Freshwater Biology* 55, 282–298.
- Zhao, H., Kumagai, J., Nakagawa, M., Shibasaki, R., 2002. Semi-automatic road extraction from high-resolution satellite image. *International Archives Of Photogrammetry Remote Sensing and Spatial Information Sciences* 34, 406-411.
- Zimmermann, H.G., Moran, V.C., Hoffmann, J.H., 2004. Biological control in the management of invasive alien plants in South Africa, and the role of the Working for Water programme. *South African Journal of Science* 100, 34-40.

APPENDICES

Appendix A

Source code implementing the black wattle classification algorithm

A1.1. BlackWattle.java

```
package bw2;

import java.awt.*;
import java.awt.event.*;
import java.awt.geom.Rectangle2D;
import java.awt.image.BufferedImage;
import java.awt.image.ImageObserver;
import java.awt.print.*;
import java.io.*;
import java.util.Iterator;

import javax.imageio.ImageIO;
import javax.imageio.ImageWriter;
import javax.imageio.stream.ImageOutputStream;
import javax.swing.*;
import javax.swing.border.*;
import org.apache.commons.math.MathException;

/**
 * A class that creates and displays the graphical user interface
 * for the application.
 */
public class BlackWattle implements Printable {

    private static final ImageObserver observer = null;

    /**
     * Main method
     * @throws IOException
     */
    public static void main (String[] args)
        throws IOException {

        try {

            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            e.printStackTrace();
        }
        BlackWattle gui = new BlackWattle();
        gui.blackWattleHome();
    }

    private JMenuItem jmibwclass, jmiopen, jmiexport, jmiclose, jmiprint;
    private JButton jbtih, jbtid, jbts, jbto, jbtr, jbtc, jbtv, jbtci;
    private JTextField direct1, direct2, direct3, direct4, direct5;
    private String pathifh = null, pathifd = null, pathis = null, pathiv = null, pathio = null;
    static String path;
```

```

/**
 * Creates a new BlackWattle instance.
 */
public BlackWattle() {
}

/**
 * Creates and displays the first graphical user interface
 * that is seen on startup. This graphical user interface
 * comprises of a frame with two menu items that is
 * "File" and "Classification" menu items. It is the
 * classification menu item that leads to the "Black
 * wattle classification" graphical user interface.
 * @throws IOException
 */
public void blackWattleHome()
    throws IOException {

    final ImageIcon leaf = new ImageIcon("resources/leaf.png");
    Font font = new Font ("ArialNarrow", 0, 12);

    /**Creates a frame**/
    JFrame frame = new JFrame("ESDA");
    frame.setSize(744, 365);
    frame.setResizable(false);
    frame.setLocationRelativeTo(null);
    frame.setIconImage(leaf.getImage());
    frame.setVisible(true);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    /**Creates and adds a menu bar to the frame**/
    JMenuBar jmb = new JMenuBar();
    frame.setJMenuBar(jmb);

    /**Adds a "File" menu to the menu bar**/
    JMenu filemenu = new JMenu("File");
    filemenu.setFont(font);
    jmb.add(filemenu);

    /**Adds a "Classification" menu to the menu bar**/
    JMenu classificationmenu = new JMenu("Classification");
    classificationmenu.setFont(font);
    jmb.add(classificationmenu);

    /**Adds menu items to the "File" menu**/
    filemenu.add(jmiopen = new JMenuItem ("Open file"));
    jmiopen.setFont(font);
    filemenu.add(jmiexport = new JMenuItem ("Export"));
    jmiexport.setFont(font);
    filemenu.addSeparator();
    filemenu.add(jmiclose = new JMenuItem ("Exit"));
    jmiclose.setFont(font);
    filemenu.add(new JSeparator(JSeparator.VERTICAL));
    classificationmenu.add(jmibwclass = new JMenuItem("Black wattle classification"));
    jmibwclass.setFont(font);

    /**Adds the startup image to the frame**/
    BufferedImage intro = ImageIO.read(new File("resources/intro.png"));
    JLabel introlabel = new JLabel(new ImageIcon(intro));
    frame.add(introlabel);
}

```

```

/**Register listeners**/
jmiopen.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        path = chooseFiles();
        try {
            displayClassifiedImage(path);
        } catch (NullPointerException e1) {
            JOptionPane.showMessageDialog(null, "ERROR select a classified
            image(.png) to display.", "Error", JOptionPane.ERROR_MESSAGE);
        } catch (FileNotFoundException e1) {
            JOptionPane.showMessageDialog(null, "ERROR classified image not
            found.", "Error", JOptionPane.ERROR_MESSAGE);
        } catch (IOException e1) {
            JOptionPane.showMessageDialog(null, "ERROR reading classified
            image file.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});

jmiexport.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        try {
            exportClassifiedImage();
        } catch (IOException e1) {
            JOptionPane.showMessageDialog(null, "ERROR creating exported
            image.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});

jmiclose.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        System.exit(0);
    }
});

jmibwclass.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        blackWattleClassification();
    }
});
}

/**
 * Creates and displays the classification algorithm graphical user
 * interface. The user is prompted to select the relevant files
 * for each field.
 * <p>
 * > Hyperion Image Header File: Select the Hyperion image header file(.HDR).
 * <p>
 * > Hyperion Image Data File: Select the Hyperion image data file.
 * <p>
 * > Spectral Signature File: Select the spectral signature file(.txt) captured
 * from the image.
 * Select the "Image" checkbox.
 * <P>
 * > Ground Truth Data File: Select the ground truth data file(.txt).
 * Select the "Validation" checkbox.
 * <p>
 * > Save Classified Image As: Select the location where the classified image
 * will be save be to.
 */

```

```

public void blackWattleClassification() {

    ImageIcon leaf = new ImageIcon("resources/leaf.png");
    Font font = new Font ("ArialNarrow", 0, 11);
    Border loweredetched = BorderFactory.createEtchedBorder(EtchedBorder.LOWERED);

    /**Panel to hold the Hyperion image header and data file**/
    final JPanel p1a = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));

    p1a.add(jbtih = new JButton("Hyperion Image Header File"));
    jbtih.setPreferredSize(new Dimension(166, 20));
    jbtih.setFont(font);
    jbtih.setHorizontalAlignment(SwingConstants.LEFT);
    p1a.add(direct1 = new JTextField());
    direct1.setPreferredSize(new Dimension(518, 21));
    direct1.setFont(font);

    final JPanel p1b = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
    p1b.add(jbtid = new JButton("Hyperion Image Data File"));
    jbtid.setPreferredSize(new Dimension(166, 20));
    jbtid.setFont(font);
    jbtid.setHorizontalAlignment(SwingConstants.LEFT);
    p1b.add(direct2 = new JTextField());
    direct2.setPreferredSize(new Dimension(518, 21));
    direct2.setFont(font);

    final JPanel p1 = new JPanel(new GridLayout(2, 2, 0, 0));
    p1.add(p1a);
    p1.add(p1b);
    p1.setBorder(loweredetched);

    /**Creates the checkboxes**/
    final JCheckBox jimg = new JCheckBox("Image File");
    jimg.setFont(font);
    final JCheckBox jval = new JCheckBox("Validation");
    jval.setFont(font);

    /**Panel to hold the spectral signature and ground truth data files**/
    final JPanel p2a = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));

    p2a.add(jbts = new JButton("Spectral Signature File "));
    jbts.setPreferredSize(new Dimension(166, 20));
    jbts.setFont(font);
    jbts.setHorizontalAlignment(SwingConstants.LEFT);
    p2a.add(direct3 = new JTextField());
    direct3.setPreferredSize(new Dimension(518, 21));
    direct3.setFont(font);

    JPanel p2b = new JPanel(new FlowLayout(FlowLayout.RIGHT, 130, 5));
    p2b.add(jimg);

    final JPanel p2c = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
    p2c.add(jbtv = new JButton("Ground Truth Data File "));
    jbtv.setPreferredSize(new Dimension(166, 20));
    jbtv.setFont(font);
    jbtv.setHorizontalAlignment(SwingConstants.LEFT);
    jbtv.setEnabled(false);
    p2c.add(direct4 = new JTextField());
    direct4.setPreferredSize(new Dimension(518, 21));
    direct4.setFont(font);
    direct4.setEditable(false);

```

```

JPanel p2d = new JPanel(new FlowLayout(FlowLayout.RIGHT, 128, 5));
p2d.add(jval);

final JPanel p2 = new JPanel(new GridLayout(4, 2, 0, 0));
p2.add(p2a);
p2.add(p2b);
p2.add(p2c);
p2.add(p2d);
p2.setBorder(loweredetched);

/**Panel to hold the output classified image file**/
final JPanel p3 = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));

p3.add(jbto = new JButton("Save Classified Image As"));
jbto.setPreferredSize(new Dimension(166, 20));
jbto.setFont(font);
p3.add(direct5 = new JTextField());
direct5.setPreferredSize(new Dimension(518, 21));
direct5.setFont(font);
p3.setBorder(loweredetched);

/**Panel to hold the apply and cancel buttons**/
final JPanel p4 = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));

p4.add(jbtr = new JButton("Apply"));
jbtr.setPreferredSize(new Dimension(70, 20));
jbtr.setFont(font);
p4.add(jbtc = new JButton("Cancel"));
jbtc.setPreferredSize(new Dimension(70, 20));
jbtc.setFont(font);
p4.setBorder(loweredetched);

/**Panel to hold all created panels**/
final JPanel p5 = new JPanel(new FlowLayout(FlowLayout.RIGHT, 10, 14));
p5.add(p1);
p5.add(p2);
p5.add(p3);
p5.add(p4);

/**Creates the Black wattle classification frame**/
final JFrame bwframe = new JFrame("Black wattle classification");
bwframe.setSize(744, 365);
bwframe.setResizable(false);
bwframe.setLocationRelativeTo(null);
bwframe.setIconImage(leaf.getImage());
bwframe.setVisible(true);
bwframe.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
bwframe.add(p5);

/**Register listeners**/
jbtih.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        if (e.getSource() == jbtih) {
            pathifh = chooseFiles();
            direct1.setText(pathifh);
            direct1.setCaretPosition(0);
        }
    }
});

jbtid.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {

```



```

        if (e.getSource() == jbtid) {
            pathifd = chooseFiles();
            direct2.setText(pathifd);
            direct2.setCaretPosition(0);
        }
    });

    jbts.addActionListener(new ActionListener() {
        public void actionPerformed (ActionEvent e) {
            if (e.getSource() == jbts) {
                pathis = chooseFiles();
                direct3.setText(pathis);
                direct3.setCaretPosition(0);
            }
        }
    });

    jimg.addActionListener(new ActionListener() {
        public void actionPerformed (ActionEvent e) {
        }
    });

    jval.addActionListener(new ActionListener() {
        public void actionPerformed (ActionEvent e) {
            if (jval.isSelected() == true) {
                direct4.setEditable(true);
                jbtv.setEnabled(true);
            } else if (jval.isSelected() == false) {
                direct4.setEditable(false);
                jbtv.setEnabled(false);
            }
        }
    });

    jbtv.addActionListener(new ActionListener() {
        public void actionPerformed (ActionEvent e) {
            if (e.getSource() == jbtv) {
                pathiv = chooseFiles();
                direct4.setText(pathiv);
                direct4.setCaretPosition(0);
            }
        }
    });

    jbto.addActionListener(new ActionListener() {
        public void actionPerformed (ActionEvent e) {
            if (e.getSource() == jbto) {
                JFileChooser jc = new JFileChooser();
                jc.showSaveDialog(null);
                try {
                    pathio = jc.getSelectedFile().getCanonicalPath() + ".png";
                } catch (NullPointerException e1) {
                    JOptionPane.showMessageDialog(null, "Input classified
                    image file name.", "Error", JOptionPane.ERROR_MESSAGE);
                } catch (IOException e1) {
                    JOptionPane.showMessageDialog(null, "ERROR creating
                    classified image.", "Error", JOptionPane.ERROR_MESSAGE);
                }
                direct5.setText(pathio);
                direct5.setCaretPosition(0);
            }
        }
    });

```

```

    }
});

jbtr.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        if (e.getSource() == jbtr) {
            try {
                HyperionImage hyperionimage = new HyperionImage(pathifh,
                    pathifd);
                StatisticalTests spectralsignature = new
                    StatisticalTests(pathis, hyperionimage);
                if (jimg.isSelected() == false) {
                    throw new IllegalArgumentException();
                } else if (jimg.isSelected() == true) {
                    spectralsignature.setImgVal();
                }
                spectralsignature.zTable();
                if (jval.isSelected() == true) {
                    StatisticalTests validation = new
                        StatisticalTests(pathiv, hyperionimage);
                    validation.setObservedExpectedVal();
                    validation.chiSquaredTest();
                }
                if (pathio == null) {
                    throw new NullPointerException();
                }
                RasterImage rasterimage2 = new RasterImage(pathio,
                    hyperionimage);
                rasterimage2.createRaster();
                rasterimage2.classifiedImageFrame();
                if (jval.isSelected() == true) {
                    rasterimage2.accuracyAssessmentFrame();
                }
            } catch (IllegalArgumentException e1) {
                JOptionPane.showMessageDialog(null, "Select spectral
                    signature file." + "\nSelect either ASD file or Image
                    file checkbox.", "Error", JOptionPane.ERROR_MESSAGE);
            } catch (NullPointerException e1) {
                JOptionPane.showMessageDialog(null, "Select spectral
                    signature file." + "\nInput classified image file name.",
                    "Error", JOptionPane.ERROR_MESSAGE);
            } catch (FileNotFoundException e1) {
                JOptionPane.showMessageDialog(null, "ERROR spectral
                    signature file not found.", "Error",
                    JOptionPane.ERROR_MESSAGE);
            } catch (IOException e1) {
                e1.printStackTrace();
                JOptionPane.showMessageDialog(null, "ERROR reading
                    spectral signature file.", "Error",
                    JOptionPane.ERROR_MESSAGE);
            } catch (MathException e1) {
                e1.printStackTrace();
            }
        }
    }
});

jbtcl.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        if (e.getSource() == jbtcl) {
            bwframe.setVisible(false);
        }
    }
});

```

```

        });
    }
}

/**
 * Creates and displays the graphical user interface to display the
 * classified image.
 * @param pathio Pathname of the classified image
 * @throws FileNotFoundException
 * @throws IOException
 */
public void displayClassifiedImage(String pathio)
    throws FileNotFoundException, IOException {

    try {
        ImageIcon leaf = new ImageIcon("resources/leaf.png");
        Font font = new Font ("ArialNarrow", 0, 11);
        Font font2 = new Font ("ArialNarrow", Font.BOLD, 13);
        BufferedImage image = null;
        String pathci = pathio;

        /**Reads the classified image**/
        image = ImageIO.read(new File(pathci));
        JLabel classifiedimage = new JLabel(new ImageIcon(image));
        int height = image.getWidth();
        int width = image.getHeight();

        /**Panel to hold the title**/
        final JPanel p1 = new JPanel(new FlowLayout(FlowLayout.CENTER, height / 2 -
        25, 5));
        JLabel classifiedimage1 = new JLabel("Classified Image");
        classifiedimage1.setFont(font2);
        p1.add(classifiedimage1);

        /**Panel to hold the classified image**/
        final JPanel p2 = new JPanel(new FlowLayout(FlowLayout.CENTER, 0, 0));
        p2.add(classifiedimage);

        /**Panel to hold the legend**/
        final JPanel p3 = new JPanel(new GridLayout(5, 2, 10, 5));
        p3.setBorder(new TitledBorder("Legend"));

        final JTextField red = new JTextField();
        red.setBackground(Color.RED);
        red.setEditable(false);
        p3.add(red);
        JLabel legend1 = new JLabel("0.75 - 0.99");
        legend1.setFont(font);
        p3.add(legend1);

        final JTextField yellow = new JTextField();
        yellow.setBackground(Color.YELLOW);
        yellow.setEditable(false);
        p3.add(yellow);
        JLabel legend2 = new JLabel("0.50 - 0.75");
        legend2.setFont(font);
        p3.add(legend2);

        final JTextField green = new JTextField();
        green.setBackground(Color.GREEN);
        green.setEditable(false);
        p3.add(green);
    }
}

```

```

JLabel legend3 = new JLabel("0.25 - 0.50");
legend3.setFont(font);
p3.add(legend3);

final JTextField blue = new JTextField();
blue.setBackground(Color.BLUE);
blue.setEditable(false);
p3.add(blue);
JLabel legend4 = new JLabel("0.00 - 0.25");
legend4.setFont(font);
p3.add(legend4);

final JTextField wht = new JTextField(1);
wht.setBackground(Color.WHITE);
wht.setEditable(false);
p3.add(wht);
JLabel legend5 = new JLabel("0.00");
legend5.setFont(font);
p3.add(legend5);

/**Panel to hold the scale bar and north arrow**/
final JPanel p4a = new JPanel(new BorderLayout(10, 10));

BufferedImage scale = ImageIO.read(new File("resources/scalebar2.jpg"));
JLabel scalelabel = new JLabel(new ImageIcon(scale));
p4a.add(scalelabel, BorderLayout.WEST);

BufferedImage northarrow = ImageIO.read(new File("resources/northarrow.jpg"));
JLabel northarrowlabel = new JLabel(new ImageIcon(northarrow));
p4a.add(northarrowlabel, BorderLayout.EAST);

final JPanel p4 = new JPanel(new BorderLayout(10, 10));
p4.add(p4a, BorderLayout.SOUTH);

/**Panel to hold the panels containing the legend, scale bar, north arrow, co-
ordinate and statistical information**/
final JPanel p8 = new JPanel(new BorderLayout(10, 10));
p8.add(p3, BorderLayout.WEST);
p8.add(p4, BorderLayout.EAST);

/**Panel to hold all created panels**/
final JPanel p9 = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));
p9.add(p1);
p9.add(p2);
p9.add(p8);

/**Creates a frame for the the classified image**/
final JFrame frame = new JFrame("Classified image");
frame.setSize(height + 60, width + 270);
frame.setResizable(true);
frame.setLocationRelativeTo(null);
frame.setIconImage(leaf.getImage());
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
frame.add(p9);

/**Creates and adds a menu bar to the frame**/
JMenuBar jmb = new JMenuBar();
frame.setJMenuBar(jmb);

/**Adds a menu "File" to the menu bar**/
JMenu filemenu = new JMenu("File");

```

```

        filemenu.setFont(font);
        jmb.add(filemenu);

        /**Adds menu items to the "File" menu**/
        filemenu.add(jmiprint = new JMenuItem ("Print"));
        jmiprint.setFont(font);
        filemenu.addSeparator();
        filemenu.add(jmiclose = new JMenuItem ("Close"));
        jmiclose.setFont(font);

        /**Register listeners**/
        jmiprint.addActionListener(new ActionListener() {
            public void actionPerformed (ActionEvent e) {
                PrinterJob printJob = PrinterJob.getPrinterJob();
                printJob.setPrintable(new BlackWattle());
                boolean doPrint = printJob.printDialog();
                if (doPrint) {
                    try {
                        printJob.print();
                    } catch (PrinterException e1) {
                        /* The job did not successfully complete*/
                    }
                }
            }
        });

        jmiclose.addActionListener(new ActionListener() {
            public void actionPerformed (ActionEvent e) {
                frame.setVisible(false);
            }
        });

    } catch (IllegalArgumentException e1) {
        JOptionPane.showMessageDialog(null, "ERROR displaying the classified image.",
        "Error", JOptionPane.ERROR_MESSAGE);
    }
}

/**
 * Converts the classified image from .png to .tiff data format.
 * @throws IOException
 */
public void exportClassifiedImage()
    throws IOException {

    ImageIcon leaf = new ImageIcon("resources/leaf.png");
    Font font = new Font ("ArialNarrow", 0, 11);
    Border loweredetched = BorderFactory.createEtchedBorder(EtchedBorder.LOWERED);

    /**Panel to hold the Hyperion image header and data file**/
    final JPanel p1a = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));

    p1a.add(jbtci = new JButton("Classified Image"));
    jbtci.setPreferredSize(new Dimension(166, 20));
    jbtci.setFont(font);
    jbtci.setHorizontalAlignment(SwingConstants.LEFT);
    p1a.add(direct1 = new JTextField());
    direct1.setPreferredSize(new Dimension(500, 21));
    direct1.setFont(font);

    final JPanel p1b = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
    p1b.add(jbto = new JButton("Save File As"));

```

```

jbto.setPreferredSize(new Dimension(166, 20));
jbto.setFont(font);
jbto.setHorizontalAlignment(SwingConstants.LEFT);
p1b.add(direct2 = new JTextField());
direct2.setPreferredSize(new Dimension(500, 21));
direct2.setFont(font);

final JPanel p1 = new JPanel(new GridLayout(2, 2, 0, 0));
p1.add(p1a);
p1.add(p1b);
p1.setBorder(loweredetched);

/**Panel to hold the apply and cancel buttons**/
final JPanel p2 = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));

p2.add(jbtr = new JButton("Apply"));
jbtr.setPreferredSize(new Dimension(70, 20));
jbtr.setFont(font);
p2.add(jbtc = new JButton("Cancel"));
jbtc.setPreferredSize(new Dimension(70, 20));
jbtc.setFont(font);
p2.setBorder(loweredetched);

final JPanel p5 = new JPanel(new FlowLayout(FlowLayout.RIGHT, 10, 10));
p5.add(p1);
p5.add(p2);

/**Creates a frame for the the classified image**/
final JFrame frame = new JFrame("Export classified image");
frame.setSize(744, 170);
frame.setResizable(false);
frame.setLocationRelativeTo(null);
frame.setIconImage(leaf.getImage());
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
frame.add(p5);

/**Register listeners**/
jbtci.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        if (e.getSource() == jbtci) {
            pathifd = chooseFiles();
            direct1.setText(pathifd);
            direct1.setCaretPosition(0);
        }
    }
});

jbto.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        if (e.getSource() == jbto) {
            JFileChooser jc = new JFileChooser();
            jc.showSaveDialog(null);
            try {
                pathio = jc.getSelectedFile().getCanonicalPath() +
                    ".tiff";
            } catch (NullPointerException e1) {
                JOptionPane.showMessageDialog(null, "ERROR input exported
                image file name.", "Error", JOptionPane.ERROR_MESSAGE);
            } catch (IOException e1) {
                JOptionPane.showMessageDialog(null, "ERROR reading
                classified image.", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
});

```

```

        }
        direct2.setText(pathio);
        direct2.setCaretPosition(0);
    }
});

jbtr.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        if (e.getSource() == jbtr) {
            try {
                final BufferedImage png = ImageIO.read(new File(pathifd));

                Iterator writers =
                    ImageIO.getImageWritersByFormatName("png");
                ImageWriter writer = (ImageWriter) writers.next();
                if (writer == null) {
                    throw new RuntimeException("PNG not supported?!");
                }
                File file = new File(pathio);
                ImageOutputStream out =
                    ImageIO.createImageOutputStream(file);
                writer.setOutput(out);
                writer.write(png);
                out.close();
                displayClassifiedImage(pathio);
            } catch (NullPointerException e1) {
                JOptionPane.showMessageDialog(null, "ERROR select a
                    classified image(.png) to export or \ninput a file name
                    for the exported image", "Error",
                    JOptionPane.ERROR_MESSAGE);
            } catch (IOException e1) {
                JOptionPane.showMessageDialog(null, "ERROR classified
                    image file not found.", "Error",
                    JOptionPane.ERROR_MESSAGE);
            }
        }
    }
});

jbtc.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        if (e.getSource() == jbtc) {
            frame.setVisible(false);
        }
    }
});

}

/**
 * Creates a file chooser dialog and returns the pathname of a selected file.
 * @return path: The pathname of the selected file.
 */
public String chooseFiles() {
    String path = null;
    JFileChooser jfilechooser = new JFileChooser();
    jfilechooser.setVisible(true);
    if (jfilechooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {
        java.io.File file = jfilechooser.getSelectedFile();
        path = (file.getPath());
    }
}

```

```

        }
        return path;
    }

/**
 * Prints a map layout of the classified image.
 * @param graphics The context into which the page is drawn
 * @param pageformat The size and orientation of the page being drawn
 * @param pageindex The zero based index of the page to be drawn
 * @return PAGE_EXISTS: If the page is rendered successfully
 * @throws PrinterException
 */
public int print(Graphics graphics, PageFormat pageformat, int pageindex)
    throws PrinterException {

    if (pageindex > 0) {
        return NO_SUCH_PAGE;
    }
    Graphics2D g2d = (Graphics2D)graphics;
    g2d.translate(pageformat.getImageableX(), pageformat.getImageableY());
    BufferedImage image;
    try {
        Rectangle2D.Double border = new Rectangle2D.Double(0, 0,
            pageformat.getImageableWidth(), pageformat.getImageableHeight());
        g2d.draw(border);
        image = ImageIO.read(new File(path));
        int x = ((int) pageformat.getWidth() / 2) - (image.getWidth() / 2);
        String title = new String("Classified Image");
        graphics.drawString(title, 20 + x + image.getWidth() / 4, 50);
        graphics.drawImage(image, x, (int) pageformat.getHeight() / 8, observer);
    } catch (IOException e) {
        e.printStackTrace();
    }
    return PAGE_EXISTS;
}

}

```


A1.2. HyperionImage.java

```
package bw2;

import java.io.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 * A class that reads the Hyperion image header file (.HDR)
 * and the Hyperion image data file which is in band
 * interleaved pixel(BIP) interleaving.
 */
public class HyperionImage {

    private int bandsnumber, columnsnumber, rowsnumber;
    private double latitude, longitude, pixelsize;
    private String interleave;
    private RandomAccessFile file;
    private int headeroffset = 0;
    private long position = 0;
    private final Logger logger = Logger.getLogger(HyperionImage.class.getCanonicalName());
    private final int TYPESIZE = 2;

    /**
     * Creates a new HyperionImage instance.
     */
    public HyperionImage() {
    }

    /**
     * Creates a new HyperionImage instance given the header file(.HDR) pathname and data file
     * pathname.
     * @param headerfile Pathname of the header file
     * @param datafile Pathname of the data file
     * @throws FileNotFoundException
     * @throws IOException
     */
    public HyperionImage(String headerfile, String datafile)
        throws FileNotFoundException, IOException {

        try {
            this.readHeaderFile(headerfile);
            this.file = new RandomAccessFile(datafile, "r");
        } catch (NullPointerException e1) {
            JOptionPane.showMessageDialog(null, "Select Hyperion image header/data file.",
                "Error", JOptionPane.ERROR_MESSAGE);
        } catch (FileNotFoundException e1) {
            JOptionPane.showMessageDialog(null, "ERROR Hyperion image header/data file not
                found.", "Error", JOptionPane.ERROR_MESSAGE);
        } catch (IOException e1) {
            JOptionPane.showMessageDialog(null, "ERROR reading Hyperion image header/data
                file.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```

/**
 * Gets the number of bands in the Hyperion image.
 * @return bandsnumber: Number of bands
 */
public int getBandsNumber() {
    return bandsnumber;
}

/**
 * Gets the number of columns in the Hyperion image.
 * @return columnsnumber: Number of columns
 */
public int getColumnsNumber() {
    return columnsnumber;
}

/**
 * Gets the number of rows in the Hyperion image.
 * @return rowsnumber: Number of rows
 */
public int getRowsNumber() {
    return rowsnumber;
}

/**
 * Gets the Hyperion image data file.
 * @return file: Hyperion image data file
 */
public RandomAccessFile getFile() {
    return file;
}

/**
 * Gets the position of the file pointer in the Hyperion image data file.
 * @return position: Position in the data file
 */
public long getPosition() {
    return position;
}

/**
 * Sets the position of the file pointer after reading
 * a single digital number value of type short.
 */
public long setPosition() {
    return getPosition() + TYPESIZE;
}

/**
 * Iterator method that checks if there's any pixel left to read.
 * @return True if there's an unread pixel in the data file, false otherwise.
 */
public boolean hasNext() {
    return position < getRowsNumber() * getColumnsNumber();
}

/**
 * Gets the latitude of pixel 0,0.
 * @return latitude
 */
public double getLatitude() {
    return latitude;
}

```

```

}

/**
 * Gets the longitude of pixel 0,0.
 * @return longitude
 */
public double getLongitude() {
    return longitude;
}

/**
 * Gets the offset between lines of latitude and longitude.
 * @return pixelsize
 */
public double getPixelSize() {
    return pixelsize;
}

/**
 * Converts digital number values from small endian to big endian.
 * @param s: Small endian value
 * @return big endian value
 */
public short toBigEndian(short s) {
    return (short) ((short) ((s >>> 8) & 0xff) | ((s << 8) & 0xff00));
}

/**
 * Reads the Hyperion image attribute information(dimensions of the image, data format
 * etc.) from the Hyperion image header file.
 * @param headerfile Pathname of the header file
 * @throws NullPointerException
 * @throws IOException
 */
public void readHeaderFile(String headerfile)
    throws IOException {
    String line;
    logger.log(Level.INFO, "Processing ENVI header file " + headerfile + ".");
    try {
        final BufferedReader header = new BufferedReader(new FileReader(headerfile));
        int setValues = 0x0;
        line = header.readLine().trim();
        if (!line.equals("ENVI")) {
            throw new IllegalArgumentException();
        }
        header.readLine();
        header.readLine();
        header.readLine();
        while ((line = header.readLine()) != null) {
            final String[] pair = line.split("=");
            pair[0] = pair[0].trim();
            pair[1] = pair[1].trim();
            if (pair[0].equals("samples")) {
                columnsnumber = Integer.parseInt(pair[1]);
                setValues |= 0x1;
            } else if (pair[0].equals("lines")) {
                rowsnumber = Integer.parseInt(pair[1]);
                setValues |= 0x2;
            } else if (pair[0].equals("bands")) {
                bandsnumber = Integer.parseInt(pair[1]);
                setValues |= 0x4;
            }
        }
    }
}

```

```

        } else if (pair[0].equals("header offset")) {
            headeroffset = Integer.parseInt(pair[1]);
            setValues |= 0x8;
        } else if (pair[0].equals("interleave")) {
            interleave = pair[1];
            setValues |= 0x12;
        } else if (pair[0].equals("map info")) {
            String[] mapinfo = pair[1].split(",");
            longitude = Double.parseDouble(mapinfo[3]);
            latitude = Double.parseDouble(mapinfo[4]);
            pixelsize = Double.parseDouble(mapinfo[5]);
            setValues |= 0x14;
            break;
        }
    }
    if (setValues != 31) {
        throw new IllegalArgumentException();
    }
    header.close();
    logger.log(Level.INFO, "ENVI header file " + headerfile + " processed successfully.");
} catch (IllegalArgumentException e1) {
    JOptionPane.showMessageDialog(null, "ERROR Invalid Hyperion image header file.", "Error", JOptionPane.ERROR_MESSAGE);
}
}

/**
 * Returns digital number values from the Hyperion image data file
 * for a single pixel at a given time given the row and column values.
 * @param r Row number
 * @param c Column number
 * @return value: Array of digital number values
 * @throws IOException
 */
public double[] readDataFile(int r, int c)
    throws IOException {

    double[] value = new double[getBandsNumber()];
    if (!hasNext()) {
        throw new NoSuchElementException();
    }
    RandomAccessFile file = this.getFile();
    short value0;
    int i = (r * getColumnsNumber() + c);
    file.seek(headeroffset + (long) (TYPE_SIZE * getBandsNumber()) * i);
    for (int j = 0; j < getBandsNumber(); j++) {
        value0 = file.readShort();
        setPosition();
        value0 = toBigEndian(value0);
        value[j] = ((double)value0);
    }
    return value;
}
}

```

A1.3. StatisticalTests.java

```
package bw2;

import java.io.*;
import javax.swing.JOptionPane;
import org.apache.commons.math.*;
import org.apache.commons.math.stat.inference.ChiSquareTestImpl;

/**
 * A class that carries out a novel statistically based comparison technique and Chi-squared test.
 * A novel statistically based comparison technique is carried out between the
 * spectral signature inputted by the user and digital number values for a single pixel
 * from the Hyperion image data file. A Chi-squared test is carried out between the
 * ground truth data and the outputted classified image. This class also creates an
 * error matrix and contingency table.
 */
public class StatisticalTests {

    private HyperionImage hyperionimage = null;
    private double[] value = null, z = null, pval = null;
    private static double[] expectedvalues = new double[2];
    private static double[][] ztable = new double[83][11];
    private String path;
    private double zscore, zscore1;
    private static double chipval;
    static long[] observedvalues = new long[2];
    private static double[] specmean = new double[250], specstdev = new double[250];
    private double errormatrix[] = new double[12];

    /**
     * Creates a new StatisticalTests instance.
     */
    public StatisticalTests() {
    }

    /**
     * Creates a new StatisticalTests instance given the pathname and the HyperionImage object.
     * @param path Pathname of file used by statistical test
     * @param hyperionimage HyperionImage object
     */
    public StatisticalTests(String path, HyperionImage hyperionimage) {
        this.path = path;
        this.hyperionimage = hyperionimage;
    }

    /**
     * Creates a new StatisticalTests instance given the digital number values
     * for a single pixel from the Hyperion image data file and the HyperionImage
     * object. This instance should be used when carrying out a novel statistically
     * based comparison technique.
     * @param value Digital number values for a single pixel
     * @param hyperionimage HyperionImage object
     */
    public StatisticalTests(double[] value, HyperionImage hyperionimage) {
        this.hyperionimage = hyperionimage;
        this.value = new double[hyperionimage.getBandsNumber()];
        this.value = value;
        z = new double[hyperionimage.getBandsNumber()];
        pval = new double[hyperionimage.getBandsNumber()];
    }
}
```

```

/**
 * Sets the mean and standard deviation values for the spectral signature;
 * values are read and stored from the spectral signature file inputed by
 * the user that was obtained from the image.
 * @throws IOException
 */
public void setImgVal()
    throws IOException {

    int j = 0, mean = 0, std = 0;
    String stread = null;
    String[] splitarray = new String[2];
    BufferedReader rb = new BufferedReader(new FileReader(path));
    try {
        stread = rb.readLine().trim();
        if (!stread.startsWith("Spectral")) {
            throw new IllegalArgumentException();
        }
        stread = rb.readLine();
        splitarray = stread.split("\t");
        if (((splitarray[0].startsWith("mean")) ||
            (splitarray[0].startsWith("Mean")))) {
            mean = 0;
            std = 1;
        }
        else {
            mean = 1;
            std = 0;
        }
        while ((stread = rb.readLine()) != null) {
            stread = stread.trim();
            splitarray = stread.split("\t");
            specmean[j] = Double.parseDouble(splitarray[mean]);
            specstdev[j] = Double.parseDouble(splitarray[std]);
            j++;
        }
        rb.close();
    } catch (IllegalArgumentException e1) {
        JOptionPane.showMessageDialog(null, "ERROR Invalid spectral signature file.",
            "Error", JOptionPane.ERROR_MESSAGE);
    }
}

/**
 * Carries out a novel statistically based comparison technique between the
 * spectral signature inputed by the user and digital number values for a single
 * pixel from the Hyperion image data file.
 * @return zprobability: z probability for a single pixel
 */
public double zTest() {

    int percent = 0, sum = 0;
    double sumprob = 0, zprobability = 0;
    for (int i = 0; i < hyperionimage.getBandsNumber(); i++) {
        sum += value[i];
    }
    if (sum != 0) {
        for (int i = 0; i < hyperionimage.getBandsNumber(); i++) {
            z[i] = (((this.value[i]) / 10000) - specmean[i]) / specstdev[i];
            double a = z[i] * 100;
            int a1 = (int) Math.round(a);
            double b = a1 / 10;

```

```

        double x1 = b / 10;
        double x2 = -1 * x1;
        double b2 = a1 % 10;
        double y = Math.abs(b2 / 100);
        if (x1 < 1.9 & x2 > -1.9) {
            for (int k = 0; k < 83; k++) {
                if (x1 == ztable[k][0]) {
                    int y1 = (int) (y * 100) + 1;
                    for (int j = 0; j < 11; j++) {
                        if (y1 == j) {
                            zscore = ztable[k][j];
                        }
                    }
                }
                if (x2 == ztable[k][0]) {
                    int y2 = (int) (y * 100) + 1;
                    for (int j = 0; j < 11; j++) {
                        if (y2 == j) {
                            zscore1 = ztable[k][j];
                        }
                    }
                }
            }
            pval[i] = (zscore - zscore1) * -1;
        }
        else {
            pval[i] = 0.0;
        }
    }

    for (int k = 0; k < hyperionimage.getBandsNumber() - 1; k++) {
        if (pval[k] < 0.0) {
            pval[k] = pval[k] * -1;
        }
        if (pval[k] > 0.0) {
            percent += 1;
            sumprob = sumprob + pval[k];
        }
    }

    double bn = ((double)hyperionimage.getBandsNumber());
    double result = (percent / bn) * 100;
    if (result > 90) {
        zprobability = sumprob / percent;
    }
    else {
        zprobability = 0;
    }
}
else {
    zprobability = 1;
}
return zprobability;
}

/**
 * Reads and stores z-score values from a standard Z Table.
 * @throws IOException
 */
public void zTable()
    throws IOException {

```

```

String strRead = null;
BufferedReader rb = new BufferedReader(new FileReader("resources/ztable.txt"));
rb.readLine();
int i = 0;
while ((strRead = rb.readLine()) != null && i < 83) {
    String splitarray2[] = strRead.split("\t");
    i++;
    for (int j = 0; j < 11; j++) {
        double val = Double.parseDouble(splitarray2[j]);
        ztable[i][j] = val;
    }
}

}

/**
 * Sets the observed and expected values for the Chi-squared test.
 * Observed presence or non presence values for specific geographical locations
 * are read and stored from the ground truth data file inputed by the user.
 * Expected presence or non presence values are extracted from the classified
 * image based on the specific geographical locations read from the ground
 * truth file inputed by the user.
 * @throws MathException
 * @throws IOException
 */
public void setObservedExpectedVal()
    throws MathException, IOException{

    String strread = null;
    String[] splitarray = new String[3];
    BufferedReader rb = new BufferedReader(new FileReader(path));
    int j = 0, x = 0, y = 0, z = 0, ovfreqpres = 0, ovfreqnonpres = 0, evfreqpres = 0,
    evfreqnonpres = 0;
    int[] XYcoordinates = new int[2];
    try {
        strread = rb.readLine().trim();
        if (!strread.startsWith("Validation")) {
            throw new IllegalArgumentException();
        }
    };
    strread = rb.readLine();
    splitarray = strread.split("\t");
    if ((splitarray[0].startsWith("Latitude") &&
        splitarray[1].startsWith("Longitude"))) {
        x = 0;
        y = 1;
        z = 2;
    } else if ((splitarray[0].startsWith("Longitude") &&
        splitarray[1].startsWith("Latitude"))){
        y = 0;
        x = 1;
        z = 2;
    } else if ((splitarray[0].startsWith("Observed") &&
        splitarray[1].startsWith("Latitude"))){
        z = 0;
        x = 1;
        y = 2;
    } else if ((splitarray[0].startsWith("Observed") &&
        splitarray[1].startsWith("Longitude"))){
        z = 0;
        y = 1;
        x = 2;
    } else if ((splitarray[0].startsWith("Latitude") &&
        splitarray[1].startsWith("Observed"))){

```



```

        x = 0;
        z = 1;
        y = 2;
    } else if ((splitarray[0].startsWith("Longitude") &&
        splitarray[1].startsWith("Observed"))){
        y = 0;
        z = 1;
        x = 2;
    }
    while ((strread = rb.readLine()) != null) {
        strread = strread.trim();
        splitarray = strread.split("\t");
        double lati = Double.parseDouble(splitarray[x]);
        double longi = Double.parseDouble(splitarray[y]);
        long ov = Long.parseLong(splitarray[z]);
        XYcoordinates = getRCVal(lati, longi);
        if ((XYcoordinates[0] < hyperionimage.getRowsNumber()) &&
            (XYcoordinates[1] < hyperionimage.getColumnsNumber())) {

            if (ov == 0) {
                ovfreqnonpres++;
            }
            else {
                ovfreqpres++;
            }
            RasterImage rasterimage = new RasterImage(hyperionimage);
            double ev = rasterimage.rasterVal(XYcoordinates[0],
                XYcoordinates[1]);
            if (ev == 0) {
                evfreqnonpres++;
            }
            else if (ev < 1 && ev > 0){
                evfreqpres++;
            }
            j++;
        }
    }
    observedvalues[0] = ovfreqpres;
    observedvalues[1] = ovfreqnonpres;
    expectedvalues[0] = evfreqpres;
    expectedvalues[1] = evfreqnonpres;
    rb.close();
} catch (IllegalArgumentException e1) {
    JOptionPane.showMessageDialog(null, "ERROR Invalid ground truth data file.",
        "Error", JOptionPane.ERROR_MESSAGE);
}
}

/**
 * Returns the row and column values for the expected pixel location
 * given the latitude and longitude co-ordinates of the observed
 * geographical location.
 * @return rc: Row and column values
 * @param lati Latitude co-ordinate
 * @param longi Longitude co-ordinate
 */
public int[] getRCVal(double lati, double longi) {

    int[] rc = new int[2];
    rc[0] = (int) ((lati - hyperionimage.getLatitude()) /
        (hyperionimage.getPixelSize()));
    if (rc[0] < 0) {

```

```

        rc[0] = rc[0] * -1;
    }
    rc[1] = (int) ((longi - hyperionimage.getLongitude()) /
(hyperionimage.getPixelSize()));
    if (rc[1] < 0) {
        rc[1] = rc[1] * -1;
    }
    return rc;
}

/**
 * Carries out a Chi-square test between the observed (ground truth data) and expected
 (classified image) values.
 * @return chipval: Chi-square probability for the output classified image.
 * @throws MathException
 * @throws IOException
 */
public double chiSquaredTest()
    throws IOException, MathException {

    ChiSquareTestImpl chi = new ChiSquareTestImpl();
    chipval = chi.chiSquareTest(expectedvalues, observedvalues);
    if (chipval < 0) {
        chipval = chipval * -1;
    }
    return chipval;
}

/**
 * Returns the error matrix for the accuracy assessment.
 * @return errormatrix: 1D array of error matrix values
 */
public double[] getErrorMatrix() {

    if (expectedvalues[0] < observedvalues[0]) {
        errormatrix[0] = expectedvalues[0];
        errormatrix[1] = observedvalues[0] - expectedvalues[0];
    }
    if (expectedvalues[0] > observedvalues[0]) {
        double a = expectedvalues[0] - observedvalues[0];
        a = expectedvalues[0] - a;
        errormatrix[0] = 0;
        errormatrix[1] = a;
    }
    if (expectedvalues[1] < observedvalues[1]) {
        errormatrix[2] = expectedvalues[1];
        errormatrix[3] = expectedvalues[1] - observedvalues[1];
    }
    if (expectedvalues[1] > observedvalues[1]) {
        double a = expectedvalues[1] - observedvalues[1];
        a = expectedvalues[1] - a;
        errormatrix[2] = 0;
        errormatrix[3] = a;
    }
    errormatrix[6] = ((errormatrix[0]) / (errormatrix[0] + errormatrix[1])) * 100;
    errormatrix[7] = ((errormatrix[0]) / (errormatrix[0] + errormatrix[2])) * 100;
    errormatrix[10] = ((errormatrix[3]) / (errormatrix[3] + errormatrix[2])) * 100;
    errormatrix[11] = ((errormatrix[3]) / (errormatrix[1] + errormatrix[3])) * 100;

    return errormatrix;
}

```

```

/**
 * Gets the overall accuracy of the classified image.
 * @return calculated overall accuracy
 */
public double getOverallAccuracy() {
    return (((errormatrix[0] + errormatrix[3]))/(errormatrix[0] + errormatrix[1] +
        errormatrix[2] + errormatrix[3])) * 100;
}

/**
 * Gets the calculated kappa statistic for the classified image.
 * @return calculated kappa statistic
 */
public double getKappaStatistic() {

    double a = errormatrix[0] + errormatrix[3];
    double b = ((errormatrix[0] + errormatrix[1]) * (errormatrix[0] + errormatrix[2])) +
        ((errormatrix[2] + errormatrix[3]) * (errormatrix[1] + errormatrix[3]));
    double n = ((errormatrix[0] + errormatrix[1]) + (errormatrix[2] + errormatrix[3]));

    return ((n * a) - b) / ((n * n) - b);
}

/**
 * Gets the frequency of the observed black wattle trees.
 * @return observed present and observed absent black wattle trees
 */
public long[] getObservedValues() {
    return observedvalues;
}

/**
 * Gets the frequency of the expected black wattle trees.
 * @return expected present and expected absent black wattle trees
 */
public double[] getExpectedValues() {
    return expectedvalues;
}

/**
 * Gets the Chi-squared probability for the classified image.
 * @return chipval
 */
public double getChiPVal() {
    return chipval;
}
}

```

A1.4. RasterImage.java

```
package bw2;

import java.awt.*;
import java.awt.event.*;
import java.awt.image.BufferedImage;
import java.io.*;
import java.text.DecimalFormat;
import java.util.Iterator;
import javax.imageio.*;
import javax.imageio.stream.ImageOutputStream;
import javax.swing.*;
import javax.swing.border.Border;
import javax.swing.border.EtchedBorder;
import javax.swing.border.TitledBorder;

/**
 * A class that creates and displays the output classified image
 * displaying the presence and absence of Black wattle trees.
 */
public class RasterImage {

    private HyperionImage hyperionimage = null;
    private double[] value = null;
    private double[][] raster = null;
    private int height = 0, width = 0, x = 0, y = 0;
    private String pathio = null;
    private double zvalue;

    /**
     * Creates a new RasterImage instance.
     */
    public RasterImage() {
    }

    /**
     * Creates a new RasterImage instance given the hyperionimage object.
     * @param hyperionimage HyperionImage object
     */
    public RasterImage(HyperionImage hyperionimage) {
        this.hyperionimage = hyperionimage;
        this.value = new double[hyperionimage.getBandsNumber()];
        this.raster = new
            double[hyperionimage.getColumnsNumber()][hyperionimage.getRowsNumber()];
        height = hyperionimage.getColumnsNumber();
        width = hyperionimage.getRowsNumber();
    }

    /**
     * Creates a new RasterImage instance given the pathname of the classified image and
     * hyperionimage object.
     * @param pathio Pathname of the classified image
     * @param hyperionimage HyperionImage object
     */
    public RasterImage(String pathio, HyperionImage hyperionimage) {
        this.hyperionimage = hyperionimage;
        this.value = new double[hyperionimage.getBandsNumber()];
        this.raster = new
            double[hyperionimage.getColumnsNumber()][hyperionimage.getRowsNumber()];
        height = hyperionimage.getColumnsNumber();
    }
}
```

```

        width = hyperionimage.getRowsNumber();
        this.pathio = pathio;
    }

    /**
     * Returns the z probability for a single pixel for the classified image.
     * @param r Row number
     * @param c Column number
     * @return zprobability: z probability for a single pixel
     * @throws IOException
     */
    public double rasterVal(int r, int c)
        throws IOException {

        this.value = this.hyperionimage.readDataFile(r, c);
        StatisticalTests ztest = new StatisticalTests(this.value, this.hyperionimage);
        double zprobability = ztest.zTest();
        return zprobability;
    }

    /**
     * Stores the z probabilities for the classified image in a two dimensional array.
     * @throws IOException
     */
    public void createRaster()
        throws IOException {

        for(int r = 0; r < hyperionimage.getRowsNumber(); r++) {
            for(int c = 0; c < hyperionimage.getColumnsNumber(); c++) {
                this.raster[c][r] = rasterVal(r,c);
            }
        }
    }

    /**
     * Gets a 2D array of z probabilities for the classified image.
     * @return raster: 2D array of z probabilities
     */
    public double[][] getRaster() {
        return this.raster;
    }

    /**
     * Returns the co-ordinates of the pixel that is clicked on.
     * @param x X co-ordinate
     * @param y Y co-ordinate
     * @return xy: Longitude and latitude of the pixel
     */
    public String[] coOrdinates(int x, int y) {

        String[] xy = new String[2];
        DecimalFormat formatter = new DecimalFormat("#.#####");
        xy[0] = String.valueOf(formatter.format((hyperionimage.getPixelSize() * x) +
            hyperionimage.getLongitude()));
        xy[1] = String.valueOf(formatter.format(((hyperionimage.getPixelSize() * y) -
            hyperionimage.getLatitude()) * -1));

        return xy;
    }
}

```

```

/**
 * Creates a frame and adds the classified image to the frame.
 * @throws IllegalArgumentException
 * @throws IOException
 */
public void classifiedImageFrame()
    throws FileNotFoundException, IOException {

    try {

        ImageIcon leaf = new ImageIcon("resources/leaf.png");
        Font font = new Font ("ArialNarrow", 0, 11);
        Font font2 = new Font ("ArialNarrow", Font.BOLD, 13);

        /**Panel to hold the title**/
        final JPanel p1 = new JPanel(new FlowLayout(FlowLayout.CENTER, height / 2 -
        25, 5));
        JLabel classifiedimage = new JLabel("Classified Image");
        classifiedimage.setFont(font2);
        p1.add(classifiedimage);

        /**Panel to hold the classified image**/
        final JPanel p2 = new JPanel(new FlowLayout(FlowLayout.CENTER, 0, 0));
        createClassifiedImage(getRaster());
        p2.add(getClassifiedImage(pathio));

        /**Panel to hold the legend**/
        final JPanel p3 = new JPanel(new GridLayout(5, 2, 10, 5));
        p3.setBorder(new TitledBorder("Legend"));

        final JTextField red = new JTextField();
        red.setBackground(Color.RED);
        red.setEditable(false);
        p3.add(red);
        JLabel legend1 = new JLabel("0.75 - 0.99");
        legend1.setFont(font);
        p3.add(legend1);

        final JTextField yellow = new JTextField();
        yellow.setBackground(Color.YELLOW);
        yellow.setEditable(false);
        p3.add(yellow);
        JLabel legend2 = new JLabel("0.50 - 0.75");
        legend2.setFont(font);
        p3.add(legend2);

        final JTextField green = new JTextField();
        green.setBackground(Color.GREEN);
        green.setEditable(false);
        p3.add(green);
        JLabel legend3 = new JLabel("0.25 - 0.50");
        legend3.setFont(font);
        p3.add(legend3);

        final JTextField blue = new JTextField();
        blue.setBackground(Color.BLUE);
        blue.setEditable(false);
        p3.add(blue);
        JLabel legend4 = new JLabel("0.00 - 0.25");
        legend4.setFont(font);
        p3.add(legend4);
    }
}

```

```

final JTextField wht = new JTextField(1);
wht.setBackground(Color.WHITE);
wht.setEditable(false);
p3.add(wht);
JLabel legend5 = new JLabel("0.00");
legend5.setFont(font);
p3.add(legend5);

/**Panel to hold the scale bar and north arrow**/
final JPanel p4a = new JPanel(new BorderLayout(10, 10));

BufferedImage scale = ImageIO.read(new File("resources/scalebar2.jpg"));
JLabel scalelabel = new JLabel(new ImageIcon(scale));
p4a.add(scalelabel, BorderLayout.WEST);

BufferedImage northarrow = ImageIO.read(new File("resources/northarrow.jpg"));
JLabel northarrowlabel = new JLabel(new ImageIcon(northarrow));
p4a.add(northarrowlabel, BorderLayout.EAST);

final JPanel p4 = new JPanel(new BorderLayout(10, 10));
p4.add(p4a, BorderLayout.SOUTH);

/**Panel to hold the co-ordinates**/
final JPanel p5 = new JPanel(new FlowLayout(FlowLayout.LEFT, 0, 5));
final JTextField coord = new JTextField();
coord.setPreferredSize(new Dimension(height, 21));
coord.setEditable(false);
p5.add(coord);

/**Panel to hold the Z-test and Chi-square test probabilities**/
final JPanel p6 = new JPanel(new FlowLayout(FlowLayout.LEFT, 0, 5));
final JTextField statistics = new JTextField();
statistics.setPreferredSize(new Dimension(height, 21));
statistics.setEditable(false);
p6.add(statistics);

final JPanel p7 = new JPanel(new BorderLayout(10, 0));
p7.add(p5, BorderLayout.NORTH);
p7.add(p6, BorderLayout.SOUTH);

/**Panel to hold the panels containing the legend, scale bar, north arrow, co-
ordinate and statistical information**/
final JPanel p8 = new JPanel(new BorderLayout(10, 10));
p8.add(p3, BorderLayout.WEST);
p8.add(p4, BorderLayout.EAST);
p8.add(p7, BorderLayout.SOUTH);

/**Panel to hold all created panels**/
final JPanel p9 = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));
p9.add(p1);
p9.add(p2);
p9.add(p8);

/**Creates a frame for the the classified image**/
final JFrame frame = new JFrame("Black wattle classification");
frame.setSize(height + 60, width + 330);
frame.setResizable(false);
frame.setLocationRelativeTo(null);
frame.setIconImage(leaf.getImage());
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
frame.add(p9);

```

```

    /**Register listeners**/
    p2.addMouseMotionListener(new MouseMotionAdapter() {
        public void mouseDragged(MouseEvent e) {

            String[] xy1 = new String[2];
            x = e.getX();
            y = e.getY();
            if ((x > 0 && x < height) && (y > 0 && y < width)) {
                xy1 = coOrdinates(x, y);
                DecimalFormat formatter = new DecimalFormat("#.#####");
                coord.setText(xy1[0] + " E" + " , " + xy1[1] + " N");
                if (raster[x][y] == 1) {
                    zvalue = 0;
                } else {
                    zvalue = raster[x][y];
                }
                statistics.setText("Z-test probability value = " +
                    formatter.format(zvalue));
            }
            else {
                coord.setText("0E" + " , " + "0S" + " " + "Probaility
                    = 0");
            }
        }
    });

} catch (IllegalArgumentException e1) {
    JOptionPane.showMessageDialog(null, "ERROR displaying the classified image.",
        "Error", JOptionPane.ERROR_MESSAGE);
} catch (FileNotFoundException e1) {
    JOptionPane.showMessageDialog(null, "ERROR classified image not
        found/created.", "Error", JOptionPane.ERROR_MESSAGE);
} catch (IOException e1) {
    JOptionPane.showMessageDialog(null, "ERROR writing classified image file to
        disk.", "Error", JOptionPane.ERROR_MESSAGE);
}

}

/**
 * Creates a frame that displays the accuracy assessment.
 */
public void accuracyAssessmentFrame() {

    ImageIcon leaf = new ImageIcon("resources/leaf.png");
    Font font = new Font ("ArialNarrow", Font.BOLD, 12);
    Border loweredetched = BorderFactory.createEtchedBorder(EtchedBorder.LOWERED);
    DecimalFormat formatter = new DecimalFormat("#.#####");
    StatisticalTests statisticaltests = new StatisticalTests();
    long[] observedvalues = statisticaltests.getObservedValues();
    double[] expectedvalues = statisticaltests.getExpectedValues();
    double[] errormatrix = statisticaltests.getErrorMatrix();

    /**Panel to hold the error matrix**/
    final JPanel p1a = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
    JLabel headingem = new JLabel("Error matrix");
    headingem.setFont(font);
    p1a.add(headingem);

    final JPanel p1b = new JPanel(new GridLayout(5, 5, 10, 0));
    JLabel blank11 = new JLabel("");
    p1b.add(blank11);
    JLabel bw11 = new JLabel("Black wattle");

```



```

p1b.add(bw11);
JLabel nbw11 = new JLabel("No Black wattle");
p1b.add(nbw11);
JLabel total11 = new JLabel("Total");
p1b.add(total11);
JLabel ua1 = new JLabel("Users accuracy (%)");
p1b.add(ua1);

JLabel bw12 = new JLabel("    Black wattle");
p1b.add(bw12);
JLabel em10 = new JLabel("" + errormatrix[0]);
p1b.add(em10);
JLabel em11 = new JLabel("" + errormatrix[1]);
p1b.add(em11);
JLabel totalval11 = new JLabel("" + (errormatrix[0] + errormatrix[1]));
p1b.add(totalval11);
JLabel em16 = new JLabel("" + (errormatrix[6]));
p1b.add(em16);

JLabel nw12 = new JLabel("    No Black wattle");
p1b.add(nw12);
JLabel em12 = new JLabel("" + errormatrix[2]);
p1b.add(em12);
JLabel em13 = new JLabel("" + errormatrix[3]);
p1b.add(em13);
JLabel totalval12 = new JLabel("" + (errormatrix[2] + errormatrix[3]));
p1b.add(totalval12);
JLabel em110 = new JLabel("" + (errormatrix[10]));
p1b.add(em110);

JLabel total12 = new JLabel("    Total");
p1b.add(total12);
JLabel totalval13 = new JLabel("" + (errormatrix[0] + errormatrix[2]));
p1b.add(totalval13);
JLabel totalval14 = new JLabel("" + (errormatrix[1] + errormatrix[3]));
p1b.add(totalval14);
JLabel totalval15 = new JLabel("" + (errormatrix[0] + errormatrix[1] +
errormatrix[2] + errormatrix[3]));
p1b.add(totalval15);
JLabel blank12 = new JLabel("");
p1b.add(blank12);

JLabel pa = new JLabel("    Producers accuracy");
p1b.add(pa);
JLabel em17 = new JLabel("" + formatter.format(errormatrix[7]));
p1b.add(em17);
JLabel em111 = new JLabel("" + formatter.format(errormatrix[11]));
p1b.add(em111);

/**Panel to hold the overall accuracy value and kappa statistic*/
final JPanel p1d = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
    JLabel overallaccuracy = new JLabel("Overall accuracy " +
statisticaltests.getOverallAccuracy());
overallaccuracy.setFont(font);
p1d.add(overallaccuracy);

final JPanel p1e = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 8));
JLabel kappastatistic = new JLabel("Kappa statistic      " +
statisticaltests.getKappaStatistic());
kappastatistic.setFont(font);
p1e.add(kappastatistic);

```

```

final JPanel p1c = new JPanel(new BorderLayout(0, 0));
p1c.add(p1d, BorderLayout.NORTH);
p1c.add(p1e, BorderLayout.SOUTH);

final JPanel p1 = new JPanel(new BorderLayout(0, 0));
p1.add(p1a, BorderLayout.NORTH);
p1.add(p1b, BorderLayout.CENTER);
p1.add(p1c, BorderLayout.SOUTH);
p1.setBorder(loweredetched);

/**Panel to hold the contingency table**/
final JPanel p2a = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
JLabel headingct = new JLabel("Contingency table");
headingct.setFont(font);
p2a.add(headingct);

final JPanel p2b = new JPanel(new GridLayout(4, 4, 10, 0));
JLabel blank21 = new JLabel("");
p2b.add(blank21);
JLabel observed = new JLabel("Observed");
p2b.add(observed);
JLabel expected = new JLabel("Expected");
p2b.add(expected);
JLabel total21 = new JLabel("Total");
p2b.add(total21);

JLabel presence = new JLabel("    Presence");
p2b.add(presence);
JLabel obpresence = new JLabel("" + observedvalues[0]);
p2b.add(obpresence);
JLabel exppresence = new JLabel("" + expectedvalues[0]);
p2b.add(exppresence);
JLabel totalval21 = new JLabel("" + (observedvalues[0] + expectedvalues[0]));
p2b.add(totalval21);

JLabel absence = new JLabel("    Absence");
p2b.add(absence);
JLabel obabsence = new JLabel("" + observedvalues[1]);
p2b.add(obabsence);
JLabel exabsence = new JLabel("" + expectedvalues[1]);
p2b.add(exabsence);
JLabel totalval22 = new JLabel("" + (observedvalues[1] + expectedvalues[1]));
p2b.add(totalval22);

JLabel total22 = new JLabel("    Total");
p2b.add(total22);
JLabel totalval23 = new JLabel("" + (observedvalues[0] + observedvalues[1]));
p2b.add(totalval23);
JLabel totalval24 = new JLabel("" + (expectedvalues[0] + expectedvalues[1]));
p2b.add(totalval24);

/**Panel to hold the chi-square value**/
final JPanel p2d = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
JLabel chisquare = new JLabel("Chi-square p value " +
    statisticaltests.getChiPVal());
chisquare.setFont(font);
p2d.add(chisquare);

final JPanel p2c = new JPanel(new BorderLayout(0, 0));
p2c.add(p2d, BorderLayout.NORTH);

final JPanel p2 = new JPanel(new BorderLayout(0, 0));

```

```

p2.add(p2a, BorderLayout.NORTH);
p2.add(p2b, BorderLayout.CENTER);
p2.add(p2c, BorderLayout.SOUTH);
p2.setBorder(loweredetched);

final JPanel p3 = new JPanel(new GridLayout(2, 1, 5, 5));
p3.add(p1);
p3.add(p2);

/**Creates a frame for the accuracy assessment**/
final JFrame frame = new JFrame("Accuracy assessment");
frame.setSize(580, 450);
frame.setResizable(false);
frame.setLocationRelativeTo(frame);
frame.setIconImage(leaf.getImage());
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
frame.add(p3);
}

/**
 * Creates and stores the classified image(.png).
 * @param raster 2D array of z probabilities
 * @throws IOException
 */
@SuppressWarnings("rawtypes")
public void createClassifiedImage(double[][] raster)
    throws IOException {

    BufferedImage img = new BufferedImage(height, width, BufferedImage.TYPE_INT_RGB);
    Graphics2D g = img.createGraphics();
    g.setColor(Color.BLACK);
    g.setBackground(Color.WHITE);
    g.clearRect(0, 0, height, width);
    for (int r = 0; r < height; r++) {
        for (int c = 0; c < width; c++) {
            if (raster[r][c] == 0.0000) {
                g.setColor(Color.WHITE);
                g.fillRect(r, c, 1, 1);
            } else if (raster[r][c] >= 0.01 & raster[r][c] <= 0.25) {
                g.setColor(Color.BLUE);
                g.fillRect(r, c, 1, 1);
            } else if (raster[r][c] >= 0.25 & raster[r][c] <= 0.50) {
                g.setColor(Color.GREEN);
                g.fillRect(r, c, 1, 1);
            } else if (raster[r][c] >= 0.50 & raster[r][c] <= 0.75) {
                g.setColor(Color.YELLOW);
                g.fillRect(r, c, 1, 1);
            } else if (raster[r][c] >= 0.75 & raster[r][c] <= 0.99) {
                g.setColor(Color.RED);
                g.fillRect(r, c, 1, 1);
            } else if (raster[r][c] == 1) {
                g.setColor(Color.BLACK);
                g.fillRect(r, c, 1, 1);
            }
        }
    }
    g.dispose();
    Iterator writers = ImageIO.getImageWritersByFormatName("png");
    ImageWriter writer = (ImageWriter) writers.next();
    if (writer == null) {

```

```

        throw new RuntimeException("PNG not supported?!");
    }
    File file = new File(pathio);
    ImageOutputStream out = ImageIO.createImageOutputStream(file);
    writer.setOutput(out);
    writer.write(img);
    out.close();
}

/**
 * Gets the classified image.
 * @param pathio Pathname of classified imae
 * @return classifiedimage: Classified image
 * @throws IOException
 */
public Component getClassifiedImage(String pathio)
    throws IOException {

    BufferedImage image = ImageIO.read(new File(pathio));
    JLabel classifiedimage = new JLabel(new ImageIcon(image));
    return classifiedimage;
}

}

```

Appendix B

The black wattle classification algorithm: user manual

1. Introduction

1.1. About

The black wattle classification algorithm is a research initiative that aimed to determine the potential of a classification algorithm to identify invasive black wattle trees using imaging spectroscopy. The algorithm allows the user to identify invasive black wattle trees that are present within an area of interest using hyperspectral satellite imagery. Hyperspectral satellite imagery collected by the EO-1 Hyperion sensor offer a high spectral resolution. Consequently, invasive black wattle trees can be discriminated from the surrounding vegetation accurately. The algorithm allows the user to utilize a spectral signature for the classification process. Based on the spectral signature inputted into the algorithm, the algorithm can be used to identify invasive black wattle trees of any age group or any invasive alien plant species. Classified images produced by the algorithm can be exported into other data formats for incorporation into geographical information systems or printed. Consequently, the algorithm can be used as an instrumental tool to inform and target removal efforts of invasive black wattle trees so that negative environmental impacts can be mitigated.

1.2. System requirements

The minimum system requirements:

- 3.02 gigahertz (GHZ) processor or higher
- 4 gigabyte (GB) RAM or higher
- 100 megabytes (MB) hard drive space
- Windows 7 32bit operating system

2. Installation

Install the Java Runtime Environment found on the provided disk if you have not already done so. Thereafter, copy the file blackwattleclassifier.jar to a location on the local hard disk. To begin the application, double click on the blackwattleclassifier.jar file.

3. Interfaces

The home screen is the first interface that appears on the screen after the end user has successfully installed and initialized the program. The home screen contains a menu bar with two menu items that is “*File*” and “*Classification*” menu items.

3.1. File

3.1.1. Open file

Open file prompts the user to select a previously classified image (.png) that is found on the hard disk for display on to the screen. After selecting the classified image, a classified image frame containing the classified image, legend, scale bar and north arrow will be displayed on to the screen. To print the displayed classified image select File > Print from the menu bar in the classified image frame.

3.1.2. Export

Export allows the user to export the classified image from a .png data format to a .tiff data format. On selecting *Export*, the export classified image frame is displayed on to the screen with two fields for input that is *Classified Image* and *Save File As*. *Classified Image* prompts the user to select a previously classified image (.png) that is found on the hard disk for exporting. *Save File As* prompts the user to select a location and input a name for the exported file. After clicking *Apply*, the converted classified image in .tiff format will be displayed on to the screen.

3.1.3. Exit

Exit allows the user to exit the black wattle classification algorithm.

3.2. Classification

3.2.1. Black wattle classification

Black wattle classification allows the user to conduct an image classification to identify invasive black wattle trees based on the data inputted by the user. On selecting *Black wattle classification*, the black wattle classification frame is displayed on to the screen with the following fields for input:

Hyperion Image Header File

The Hyperion image header file (.HDR) contains the attributes of the associated Hyperion image data file. It is essential that the correct header file is used with the appropriate data file. *Hyperion Image Header File* prompts the user to select a Hyperion image header file that is found on the hard disk.

Hyperion Image Data File

Only Hyperion image data files can be processed by the algorithm. The Hyperion image data file (.File) contains the spectral information. The Hyperion image must be spatially subsetting, spectrally subsetting, atmospherically corrected and orthorectified prior to input into the algorithm. The Hyperion image must be in band interleaved pixel data format. *Hyperion Image Data File* prompts the user to select a Hyperion image data file that is found on the hard disk.

Spectral Signature File

The spectral signature file (.txt) should contain the spectral signature of a black wattle tree or invasive alien plant species. The spectral signature can be captured from the image itself. The spectral signature should be derived by taking an average of more than three reflectance curves with an associated standard deviation. The spectral signature file must be formatted appropriately to be processed by the algorithm. The first line of the file must contain *Spectral signature*. The second line must contain *Mean* and *Standard deviation* in two columns with the relevant values in each column. *Spectral Signature File* prompts the user to select a spectral signature file that is found on disk. The user must select the *Image File* checkbox.

Ground Truth Data File

The ground truth data file (.txt) should contain three columns: *Latitude*, *Longitude* and *Observed value*. The latitude and longitude values should either be in geographical or projected co-ordinate system corresponding to the co-ordinate system of the Hyperion image data file. The observed value should be a boolean value of either 1 or 0 indicating the presence or absence of black wattle trees respectively. The user must select the *Validation* checkbox first in order to select the ground truth data file. *Ground Truth Data File* prompts the user to select a ground truth data file that is found on the hard disk.

Save File As

Save File As prompts the user to select a location and input a name for the classified image.

After clicking *Apply*, the classification process will begin. Once the classification has ended, a classified image frame containing the classified image, legend, scale bar and north arrow will be displayed on to the screen. There are two fields at the bottom of the classified image frame which are only activated if the user clicks and scrolls over the classified image. The first field displays the co-ordinates of the pixel on the ground. The second field displays the probability that black wattle trees are present as well as the Chi-square probability if validation was conducted.

Thank you for choosing the black wattle classification algorithm as your choice of remote sensing software. For further information and support contact agjee@cybersmart.co.za.