MONITORING NEW ZEALAND'S NATIVE BEES

A COLLABORATIVE APPROACH USING IMAGE ANALYSIS

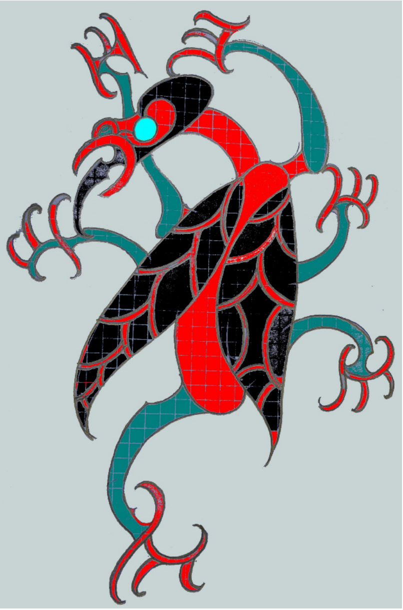
NGAIRE HART

A thesis submitted to Auckland University of Technology in fulfilment of the requirements for the degree of Doctor of Philosophy (PhD)

School of Engineering

February 2016

Ngaire Hart: *MONITORING NEW ZEALAND'S NATIVE BEES,* A Collaborative Approach using Image Analysis, February 2016, Version 3.3



Tradition tells of the time when earth mother, *Papatuanuku*, was clothed in vegetation [1]. After she was adorned, the *Atua* (Gods) turned their attention towards the insects and reptiles of the earth. There were some who viewed the insects as *kutukutu* (vermin); infesting the body of Papa. The *whatukura* (enlightened beings), Ruatau and Rehua intervened. They spoke gently of the creatures to Tāne, the great God of the forests, saying...

Treat kindly the offspring of Torohua and Muhumuhu.
That they may serve as companions for you all.
While some are desirable, others are not.
But they preceded all other things.

This manuscript is dedicated to *ngaro huruhuru*, the native bees of *Aotearoa* (New Zealand); with respect for, and in recognition of, this *whakapapa* (genealogy).

ABSTRACT

New Zealand has around thirty different species of native bees. Much has been discovered about their biology. They are pollinators of wild and cultivated plants. Because of this, they are likely important to the health of ecosystems. Most are solitary ground nesting bees. They are commonly referred to as mining bees, since individual females construct their nests in the ground. During the active flight season, many thousands of bees will nest alongside each other. They form large communities called aggregations.

Studies of native bees can be difficult. As a result, there is much to learn about their diversity and population status. To address this problem, a method to measure populations of native bees using digital images and analysis was proposed. The method capitalised on some unique aspects of their nesting biology. While it was difficult to acquire images of individual bees, it was straightforward to photograph nests. Furthermore, the number of nests in an aggregation could provide an indicator of community health. For these reasons, the methods in this research focused on counting the number of active nests. These data could be used as a proxy for populations.

Surveys were conducted over six years (2009–2014), at three communities of native bees located in Whangarei (Northland, New Zealand). Monitoring data were collected across five years (2010–2014). Fundamental ecological data were collected, including manual nest counts; digital images of active nests were acquired. Open source, biomedical imaging platform FIJI, was used to process images. Pixel-level, interactive image segmentations were achieved, using the random forest classifier. Three classifiers were constructed, one for each monitoring location. They were applied to nest-images in a single batch process and the number of active nests were counted.

The performance of machine learning models were analysed in the workbench WEKA. The random forest classifier performed well on the nest-imaging task. The model compared favourably with similar types of machine learners in WEKA experiments. Nest counts derived from the automated imaging methods were compared with manual-field and manual-image counts. There were good agreements between methods. Results suggested image-centric monitoring methods could replace manual-field nest counting methods. Data by manual and automatic techniques, indicated the number of active nests have decreased over five years.

The imaging methodology presented in this thesis shows good potential. It could be used to help increase baseline knowledge about New Zealand's native bees. The image-centric design was fully documented, based on open source software and off-the-shelf tools. Therefore, the system could be immediately adapted for other solitary ground nesting bees worldwide. This system could help to provide the tools, to gather much needed information about the health of important background pollinators.

ACKNOWLEDGEMENTS

A great many people supported me during this journey. This work would not have been successfully completed without the help, support and trust of mentors, friends and family. Foremost, I would like to acknowledge the guidance from my supervisors, Associate Professors Loulin Huang and Russel Pears.

This research was financially supported by the Ministry of Business, Innovation and Employment (MBIE), under contract number: AITX0901. I was very fortunate to receive wonderful support from the staff at MBIE who helped me to transition from Massey University to Auckland University of Technology (AUT).

Nga mihi nui mō ō manākitanga, tenā rāwā atu koutou, in appreciation of your kindness many thanks, to all the staff at AUT who have guided me along the way. My sincere thanks to Kath Ravenhall from the School of Engineering, Emma Kelly and the staff from the Research and Innovation Office; Tui O'Sullivan and all those who helped to organise and produce the exhibition on native bees.

I am indebted to the developers and maintainers of the open source packages used throughout this work; including the medical imaging package FIJI. I would especially like to thank Ignacio Arganda-Carreras who was generous with his time and resources.

My sincere gratitude to Barry Donnovan and Linda Newstrom-Lloyd for sharing their passion for the natural history of New Zealand with me; and Olivier Ball for his kindness and patience.

...Nō reira, tēnā koutou, tēnā koutou, ā tēnā tātou katoa.

DECLARATION

" I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning. "

Ngaire Hart

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

1 INTRODUCTION

Pollination is a natural process pivotal to life on earth [2]. Any changes in services has the potential to affect food security and human welfare [2, 3, 4]. The importance of pollinators was highlighted by the popular press in 2006. Declines in populations of honey bees, *Apis mellifera* (Hymenoptera: Apoidea), were documented throughout the United States (US) [5]. The term colony collapse disorder was coined to describe the syndrome [5]. The problem caused widespread alarm. This was because, around 80-85% of all cultivated food crops are dependent on pollination from insects; bees are a valuable pollinator group [6, 7]. Studies have estimated the economic cost of pollination services could be as high as $310.9 million US dollars [7]. Others have indicated the economic consequences of pollinator declines, could result in a reduction of food crops amounting to a total of $334.1 billion US dollars [8].

The pollination crisis continues to receive attention throughout Europe and the US. The causes of mass honey bee losses remain largely unknown [5]. An added consequence of this phenomenon, has been an increase in the overall awareness of the role of pollinators [6, 9, 10]. Especially the role of non-*Apis* bees [9, 10]. Many species have been undervalued in the past [11]. In the future, they may provide a buffer against major colony loss events [6, 12]. In the US, native bees are estimated to contribute about $3 billion dollars towards fruit production each year [13]. O'Toole writes, "...native bees occupy keystone positions and without them, ecosystems would eventually collapse..." [9, pg.32].

Around 20,000 different bees have been formally described by science [14]. Most of these species are solitary [14]. Unlike social bees, solitary species do not produce honey or live in colonies. They are not as easily managed for crop pollination. One exception is the alkali bee, *Nomia melanderi* (Hymenoptera: Halictidae) [15]. It is the most intensively managed solitary bee in the world and is a vital pollinator for alfalfa crops [15]. Alkali bees have a tendency to nest alongside each other. Many thousands of bees can form large communities; some persist over decades [15]. One of the most populous and long-lived nesting sites was recorded by Cane [16]. He conducted a monitoring programme on alkali bees over eight years, by measuring their populations across 240 km^2 of agricultural land (Washington, US) [16].

There are at least 40 different species of bees in New Zealand [17]. Around 32 of these are native bees [17]. Most of New Zealand's native bees are solitary ground nesting types. Many have behaviours similar to alkali bees [15]. They can be described as *gregarious* ground nesting bees. At the beginning of the active flight season (around September-October), female native bees start to construct their nests. They prefer to nest alongside each other. Consequently, within a very short period of time, large communities or aggregations are formed by thousands of nesting bees.

In contrast to alkali bees, New Zealand's native bees have not been managed for crop pollination [12]. Baseline knowledge about some species is limited. There are also challenges associated with studies of native bees. Since most species are in flight for only a few months of the year, projects can be drawn out and laborious. Good population data can take longer than three years to collect [16]. As a result, it can be difficult to attract researchers and funding. Most native bees are hard to identify with the naked eye. This is because many species look the same. Expert training is required in order to identify them properly. Even with the aid of taxonomic keys identification can be difficult [18, 19]. As described by Donovan [17], " The general resemblance of many of our native bees to some flies has not only limited their recognition as bees by the public, but within the group of large hairy species the close similarity among a number of species has made identification difficult even by entomologists." [17, pg.6].

Many native species are small. Therefore, when they are in mid-flight they are nearly impossible to *visually* track. Capturing digital images of native bees in mid-flight is equally problematic. Tracking tools have been successfully used on larger insects, but it is likely the technologies cannot be easily adapted for New Zealand's native bees [20, 21]. Looking towards global research, habitat fragmentation from urbanisation and agricultural intensification, is an important issue [6, 22]. In the future, communities of native bees may be progressively more difficult to locate and study.

There are relatively few past or present studies on New Zealand's native bees. Historical research was more often focused on taxonomy, biology and floral relationships. One of the earliest studies was by Rayment [23]. He described the life history of a Māori bee; presenting a range of sketches of larvae, pupae and adult bees. Quinn [24], conducted surveys on native bees found in the Mackenzie Basin (South Canterbury), from 1976 to 1980. He collected 10 different species of native bees. He detailed the natural history records of their foraging and nesting characteristics. In 1964, Godley [25] observed native bees foraging on *Elytranthe flavida*. The bees were prising open the tip of the mistletoe flower buds in search of nectar. Similar observations were confirmed recently. Kelly et al. [26], captured video footage of native bees opening the flowers of an endangered species of mistletoe. These plants have bird-adapted flowers and were generally thought to be *only* visited by birds [26]. In a follow up study, Robertson et al. [27] suggested native bees may "...partially replace birds as pollinators of mistletoes, despite their apparent ornithophilous syndrome." [27, pg.298].

Donovan's [17] *Apoidea* is the most comprehensive body of research; it is a complete taxonomic treatment of New Zealand's bees. Donovan et al. [28], have also made progress towards relocating native bees. They tested methods for establishing new nests using a species of native bee, *Leioproctus huakiwi* (Hymenoptera: Colletidae). They successfully established new populations of *L. huakiwi*; numbers increased by 8-25 times over three years (2007–2009). The results of their research "...opens the way for the development of new populations of bees wherever required for both conservation pollination, and pollination of some economically valuable crops." [28, pg.111].

Hart [21] reviewed different methods for studying the foraging ranges of native bees. In a related study, she collated natural history records from communities around Whangarei (Northland, New Zealand), over three years [29]. Results showed, up to seven different species of native bees were nesting alongside each other [29]. To date, the diversity of species located at a single community on Mt. Parihaka (Whangarei) remain the highest recorded in New Zealand [29]. Newstrom-Lloyd [12] provided a comprehensive account of pollination services in New Zealand. She suggested more could be done to explore the role of native bees as managed pollinators [12]. Newstrom-Lloyd explained, "...permanent sites with populations of these native bee species can be developed, and these could serve as alternatives if nest site areas are not disturbed and are supported with the necessary bee forage plants." [12, pg.419].

According to a recent report, one native bee species *Leioproctus nunui* (Hymenoptera: Colletidae), is critically endangered [30]. However, it is difficult to ascertain the conservation status of other species [30]. Native bees might provide essential pollination services [12, 17, 29]. If so, their role in New Zealand's ecosystems could be critical [4]. Studies from abroad indicate the consequences of loosing pollinator species can be unpredictable and irreversible [4, 31]. Some studies have shown the ecological effects might only be recognised when dependent species of wild and cultivated crops decline [4, 31]. Although the research to date is promising, much more is required [12, 21, 28, 29]. Long term monitoring initiatives, would help to quantify the population abundance and species diversity of New Zealand's native bees. Open source, image-centric tools could also be used, simplifying ecological methods [32, 33, 34]. If the lessons from abroad are considered relevant, monitoring initiatives might even be community driven [35, 36, 37, 38, 39]. Therefore, this thesis outlines the development, design, and application of an image-centric monitoring tool for New Zealand's native ground nesting bees.

1.1 OBJECTIVES

The main aims of this research were to design, apply and verify the performance of an image-centric monitoring system. The system was designed to measure the population abundance of ground nesting bees in New Zealand. The techniques for monitoring bees were tested during field studies in 2009. The types of digital data were not initially limited to images. Audio and video formats were also considered relevant. In 2010 field tests were completed. There was sufficient proof of concept to formulate a clear image-centric design and direction for applied field monitoring. The research thus centred on digital image acquisition and analysis of whole communities of native bees. Five questions were used to refine the research objectives:

1. What indicators could be used to establish the general health of bee communities?
2. Could images be used to capture or quantify *key* indicators?
3. Could image handling, acquisition and analyses be standardised?
4. What pattern recognition, segmentation or classification techniques best suit the image data?
5. What methods can be used to verify the accuracy and precision of imaging methods?

The research questions listed above were used to identify key aims. The objectives were therefore to:

- Investigate the types of image data suitable for proxy measures.
- Develop standard image collection techniques.
- Design manual field sampling methods for comparative analysis against imaging methods.
- Apply the methods proposed: gather manual and image data in field tests, conducted over multiple years and locations.
- Select data handling protocols including: storage, collation, processing and analysis.
- Verify the accuracy and precision of imaging methods.
- Compare manual field and imaging methods; and results.

The key objectives formed the basis of several recursive tasks. These were categorised into stages as described in Table 1.1.

Table 1.1: Development stages.

Stage	Tasks
Review Data	Examine past image data and ascertain design constraints.
	Identify appropriate monitoring sites.
Image Acquisition	Test techniques for standard image acquisition.
Field Methods	Survey locations and select sites for repeat monitoring.
	Design field methods for comparative analysis against imaging methods.
	Collect manual and image field data over multiple seasons.
Image Processing	Collate and prepare images.
	Train and apply classifiers to segment target objects in images.
	Post-process and count key data.
Verification	Verify the precision and accuracy of image classifiers.
Analysis	Examine and compare manual field and imaging results; review final monitoring data.

1.2 THESIS OUTLINE

There are three main sections in this manuscript. In Part I, key literature and background theories are introduced. Chapter 2 examines the field methods used for studying insects and the advances in digital technologies relevant to this research. The rationale for adopting image-centric tools are detailed within the context of New Zealand's native bees. Following on from this theme, Chapter 3 outlines foundation theories relating to image processing methods. Biomedical imaging techniques are well developed, so methodologies can be adapted to a range of life science applications. Open source biomedical software and new algorithmic image segmentation methods are detailed.

The research methods are described in Part II of this manuscript. Chapter 4 outlines the field sampling techniques and tools. Chapter 5 details the image-data methods and tools including: image data management, pre-processing, trainable image segmentations, post processing operations and statistical analyses.

Research outcomes are presented in Part III. The results are detailed in Chapter 6. A comparative analysis between manual-field, manual-image and automated nest counts are presented in this chapter. Chapter 7 provides a critical examination of the research methodology, rationale and outcomes. The limitations of the study are discussed. A summary of outcomes are given in Chapter 8. The research objectives are reviewed with an appraisal of the overall contributions. Suggestions for future developments are detailed.

1.3 PUBLICATIONS AND CONTRIBUTIONS

Publications from this research are listed below. Other outcomes and pathways were also important. They revolved around disseminating knowledge to a wider, more general audience. It was possible to share some of the research objectives and results with a larger community. This was achieved by combining contemporary digital media with Māori design. The images collected during field monitoring were integrated with contemporary designs to create an exhibition. The public exhibition was curated and hosted by the School of Creative Design and Technologies (AUT) (refer to Appendix G).

- [40] Hart N. H. and Huang L. *Monitoring Populations of Solitary Bees using Image Processing Techniques*. International Journal of Computer Applications in Technology, 50(1): 45–50, 2014.

- [41] Hart N. H. and Huang L. *Counting Insects in Fight using Image Processing Techniques*. In Proceedings of the 27th Conference on Image and Vision Computing New Zealand, pages 274–278. ACM, 2012.

- [42] Hart N. H. and Huang L. *Monitoring Nests of Solitary Bees using Image Processing Techniques*. In Mechatronics and Machine Vision in Practice (M2VIP), 2012 19th International Conference, pages 1–4. IEEE, 2012.

- [43] Hart N. H. and Huang L. *An Image Based Approach to Monitor New Zealand Native Bees*. In Robotics, Automation and Mechatronics (RAM), 2011 IEEE Conference, pages 353–357. IEEE, 2011.

Part I

BACKGROUND

2 MONITORING NATIVE BEES

Synopsis

The field monitoring techniques used in studies of native bees, or related insects are examined in this chapter. There are a wide range of sampling methods. Therefore, this review was orientated towards the methodologies that could be used to gather rapid, broad information about the health of communities of native bees. A review of conventional techniques are presented. Two specific approaches for evaluating populations of solitary ground nesting bees are outlined. The use of digital tools to aid ecological studies are introduced. The relevance of technologies and research methods are examined within the context of New Zealand's native bees. The rationale for an image-centric monitoring system is presented in this chapter.

2.1 TRADITIONAL METHODS

Field research methods are often based on real-world data collections, sampling, measurements and observations [44]. Data can be messy, inexact or skewed; particularly when compared to laboratory sampling methods [45, 46]. Field data sometimes requires special statistical treatment [47]. More so, if it does not conform to assumptions about normality, required by common methods [44, 45, 47, 48, 49]. For these reasons, care is given to the formation of hypotheses, sampling design and data analyses in field biology [44, 48, 50]. In the following reviews, methodologies are concerned with multiple species, or groups of insects [48, 51]. Some studies are specifically designed to measure species diversity; others the population abundance [48, 51, 52, 53]. The methods reviewed in this chapter could be used in studies of native bees [1].

2.1.1 *Mark-recapture, survey and passive sampling techniques*

There are several methods used to study insects [52, 53, 54, 55, 56]. Mark recapture techniques are considered the most reliable [56]. Because insects are captured using sweep nets, while foraging or above nest sites, some behavioural information can also be gathered [53, 56]. Mark-recapture methods can be used to quantify species diversity

[1] In reference to New Zealand's species, the terms native bees, solitary bees and ground nesting bees are used synonymously throughout this manuscript.

and population abundance [56]. However, they can be labour intensive. They require individual insects to be captured and uniquely marked, then released and caught again [56]. Scientists need expert skills to handle insects and studies are time consuming. In contrast, survey methods are easier to perform. Protocols depend on visual identification of insects. Repeat observations are made, over a predetermined transect, within a set time-frame. Although surveys are probably less reliable, there are some practical advantages. It is also possible to combine the methods and capitalise on the benefits of both.

For example, Larsson and Franz [57] estimated the overall population size of solitary bees. They used a small sample of mark, recaptured bees with an initial observational survey. According to their findings, there were good correlations between the two methods. They concluded population size could be reliably established using observational survey walks alone. They showed their method could save considerable time and effort. Thus enabling resources to be directed towards long term, large scale monitoring. Nonetheless, survey methods still depend on specialist skills. The scientists carrying out observations, must be completely familiar with the habitat; and the bee fauna found within survey locations.

Passive sampling methods do not depend heavily on specialist skills. Therefore passive techniques have some advantages over other methods. In a comprehensive study, Westphal and Bommarco et al. [58], compared a range of sampling methods including: observational plots, pan traps, standardised and variable transect walks and trap nests. Their results showed pan traps were the most efficient and gave the best indication of *species diversity*. Similar findings were reported by Nielsen and Steffan-Dewenter et al. [59]. Both studies indicated pan traps were easy to use and reliable [58, 59]. However, others have suggested pan traps are sensitive to taxonomic bias [60, 61, 62]. The traps are also fatal for the insects collected. Therefore, the method might not be appropriate for research involving vulnerable species? Several studies have indicated data gathered from pan traps, do not necessarily provide good abundance measurements [60, 62, 63]. They are more suitable for measuring the diversity of species within a habitat [62].

2.1.2 *Population methods for solitary bees*

Population studies for solitary bees are rare so there are only a few examples to draw upon. In the next few paragraphs two key studies are reviewed in turn. In a four year study, Bischoff [64] used mark-recapture methods and nest counts to evaluate populations of a solitary bee, *Andrena vaga* (Hymenoptera: Andrenidae). The estimations using nest counts, corresponded well with mark recapture data.

Bischoff [64] reported minimal differences between the cumulative number of nests (164) and the mean population estimations (160) for data collected in 1997 [64]. He measured an overall decline in the populations of *Andrena vaga*, across four years (1996 to 1999). He attributed some of the population changes to corresponding increases in populations of bee parasitoids. Bischoff [64], also concluded mark recapture methods provided more reliable data.

In an eight year study, Cane [16] recorded the population changes of the alkali bee, *Nomia melanderi* (Hymenoptera: Halictidae). Aerial photographs were used to identify communities, across a 240 km^2 area of agricultural land. Nest densities were surveyed each year using up to twenty, 1 m^2 grids. The grids were scattered randomly on aggregations. The number of nest holes per grid were counted. Using video recordings, Cane [16] checked the entry holes that were being actively used by nesting bees, against those counted. He found around 66% were actual nest entrances. Final counts were averaged across the total nest area, then multiplied by a factor of 2/3. Thus, Cane [16] accounted for the differences between the observed, and actual number of active nests. The final analysis showed the population density varied within and between nest aggregations. Results showed the populations increased by a factor of nine between the years 1999 to 2006.

Taken together, the methods used by Bischoff [64] and Cane [16] support the notion of using nests as a proxy for populations. Their studies highlight the limitations. Manual nest counting methods are likely to produce inflated estimates, according to Cane [16]. Nevertheless, the number of active nests can provide a *broad estimation* of the population abundance of solitary ground nesting bees within aggregations. The same method has been successfully used in at least two other studies to date [65, 66]. The review also highlighted the paucity of available research. Little is known about the current population health of many species of solitary bees around the world; including New Zealand's native bees. In the absence of good data records and with a lack of monitoring tools, straightforward field methods based on the number of active nests might suffice? At least until alternative methods have been developed.

2.2 technologies for field research

A diverse range of technological tools are available to science. More are developed each year. According to Moore's Law, computing power doubles every eighteen months [67]. Thus, the rate of technological developments and scientific discovery is staggering [67, 68]. Combined with mobile devices, crowd sourcing projects and increasing connectivity, the volume of data for analysis is changing the scientific landscapes [39, 69, 70]. The road map is not clearly defined.

Therefore, the following sections review a range of tools that are used, or could be used for field research studies of native bees.

2.2.1 *Digital tools*

An all purpose recognition tool that can be applied to different classification problems, without modification, is the goal which challenges most developers today [71, 72, 73]. Considerable advances have been made; and Gaston and O'Neil [72] have suggested "...the bounds on just what is possible to achieve remain to be established. " [72, p.12]. Several approaches have been used for automated identification of insects. To date, designs have been based around image, audio, and frequency data [72]. There are some novel methods. Therefore, in the first parts of this section some unique approaches are reviewed [74].

The main parts of this section are dedicated to the advances in imaging systems. This is because they have revolutionised many scientific studies. Specialised tools have been developed for bees [75, 76, 77], water insects [78, 79], live moths [80] and sharks [81, 82]. Combined these studies demonstrate the *potential* of digital technologies. They also affirm the benefits of collaborative research [83]. Many projects shared common characteristics. They involved a range of specialists, were developed using open source tools, or were designed to support community science initiatives [70].

2.2.2 *Audio data*

Insects are more often heard, than seen. Consequently, species identification can be difficult. In this application, techniques using digital audio data and frequency-based classifications are promising. For example, Potamitis et al. [84] capitalised on the unique acoustic sounds of crickets. They used audio data to recognise 105 different species. Their system was based on two main steps; acoustic signal parametrisation and classification. During the first stage, they determined the features providing the most information. During the second, they compared the input feature vectors, with a range of predefined models representing target classes. Final results were impressive. They exceeded a 99% recognition accuracy, on the levels of family and subfamily. They achieved a 94% accuracy, on the level of *species* [84].

Cricket ID with digital audio data.

Raman et al. [85] developed a low cost acoustic insect flight detector. It was designed to monitor mosquito activity. The sensor was constructed using off the shelf components; a noise cancelling microphone and digital sound recorder. They classified recordings using the various harmonic ranges of wing beat frequencies. According to Raman et al. [85], the design concept dates back to the 1949's [86]. They reported their system was inexpensive to build and performed well in natural outdoor environments [85].

Mosquito ID with audio windbeat data.

Similarly, Batista et al. [87] developed a low cost optical sensor. They designed the sensor to count and classify flying insects, by using wing beat frequencies. Different species produce distinct wing beat sounds. Therefore, they extracted unique acoustic information from recorded sequences. They matched wing beat frequencies with specific species, using Bayesian classification and probabilities. Large insects produced lower frequencies - and smaller insects produced higher frequencies. Therefore, classifications were straightforward, according to Batista et al. [87]. They reported an accuracy of 96.04%. The insect species they tested included one bumblebee, *Bumble impatiens* (Hymenoptera: Apidae) and two mosquitoes, *Culex quinquefasciatus* and *Aedes aegypti* (Diptera: Culicidae).

Insect ID with optical wing beat data.

2.2.3 *Image data*

Caci et al. [88] used I^{3}S-Contour (Interactive Individual Identification Software[2]) to identify individuals of the threatened beetle, *Rosalia alpina* (Coleoptera: Cerambycidae) [82]. Beetles were identified using natural markings on their backs, as shown below in Figure 2.1. Selected images were annotated with a semi-automated contour tracing function. The photographs were matched with unknown images from a reference library. The results were given as a ranked picture list. They achieved good recognition rates; between 94.5–95.2% using 290 images. Their system saved time and effort; and was appropriate for studies involving vulnerable species.

Beetle image ID and matching with I^{3}S-Contour.

In a similar type of study, Towner et al. [81] replaced mark-recapture methods with an image-centric approach. They monitored shark populations around the coast of South Africa [81]. They used digital images of dorsal fins to identify individual white sharks. They collected images over a four year period (2007–2011). They processed 1683 images of dorsal fins, using open source imaging software, DARWIN [89]. They identified a total of 532 unique individuals. The results of their study indicated shark populations had not recovered since they were nationally protected in 1991 [81, 90].

Shark image ID and matching with DARWIN.

Mayo and Watson [91] developed an automated species identification system for moths. They used open source medical imaging software ImageJ[3] and WEKA[4] [91, 92, 93]. They compared the performance of several classifiers which were applied to the task of moth identification. They automatically classified 35 different species, using images of 774 live individuals. They reported an accuracy of 85% using a support vector machine algorithm implemented via WEKA. Similarly, Lone et al. [80] developed a real-time image processing system. Their system was designed to identify moths in flight in their natural en-

Moth ID with ImageJ & WEKA.

[2] *I^{3}S-Contour*-http://www.reijns.com/i3s/

[3] *ImageJ*- http://imagej.nih.gov/ij and *FIJI*-http://fiji.sc/FIJI

[4] *WEKA*-http://www.cs.waikato.ac.nz/ml/weka/

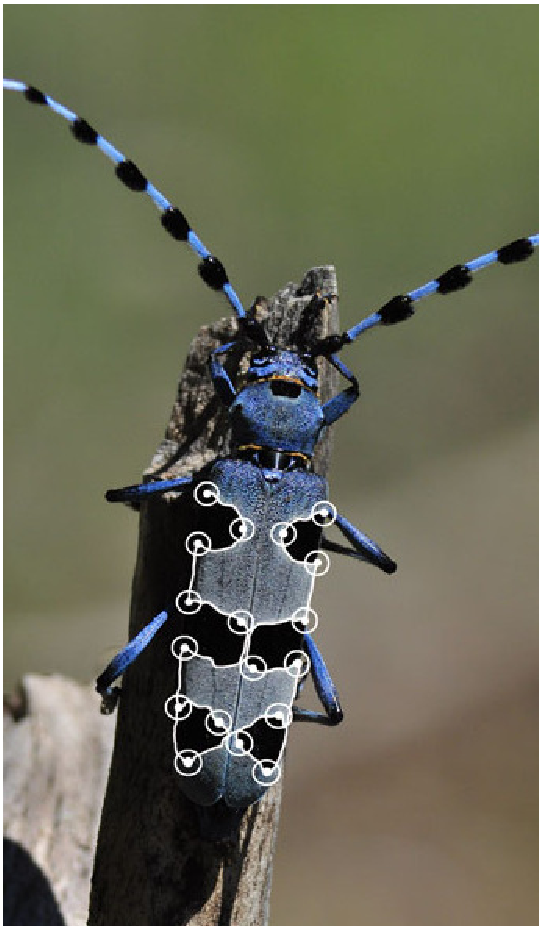


Figure 2.1: I^3S-Contour annotations on the longhorn beetle [88, pg.788]

vironment. They adapted an algorithm based on randomised trees. Their results were good; they achieved an 82% recognition rate, with 10 different species of butterflies.

2.3 RELEVANCE OF STUDIES

There are many different biological methods that can be used to help gather information about native bees [94]. Only the most relevant have been included in this review. The synthesis was directed towards the methods and tools which could be easily applied to, or adapted for, studies of New Zealand's native bees. In the following sections the context of the task is examined closer.

2.3.1 *Data-intensive science: future field methods?*

Multiple disciplines fall within the umbrella of life sciences. Specialists do not always use the same biological methods or field sampling techniques. The studies reviewed in the previous sections could be categorised by using the focus of methodologies. Some were aimed towards an understanding of: 1) individuals within a species e.g. Caci et al. [88], 2) selected species within a group e.g. Towner et al. [81], 3) collections of many different species forming communities e.g. Bischoff et al. [64], 4) larger populations of species on spatio-temporal scales e.g. Cane [16], 5) or an understanding of meta-populations or organisms e.g. Murray et al. [94].

Natural science methodologies could also be broadly classed as *observational* or *experimental*. Observed information is normally required before hypotheses are formed, refined or tested. The scientific method has changed over time; from an emphasis on empirical and observational records, to a focus on hypothesis testing and experimental design. Modern research methods, based on data-intensive techniques, are fundamentally observational studies [69].

Some of the technologies gaining momentum in biology, lend themselves towards metadata sampling. Kelling et al. [69] explored some of the difficulties associated with traditional ecological methods. In the past, methodologies were reliant on specialist knowledge and skilled collection techniques. However, when a spatial or temporal understanding of species is a requirement, there are practical challenges not met by traditional methods [69]. New approaches can draw on a vast network of human *sensors* and thereby increase data capture for the scientific community [69, 95]. Traditional indigenous knowledge systems, might also provide rich insights and valuable tools.

2.3.2 *Tool selection: what are the criteria?*

There are specialised tools that work well on a range of target insects [96]. However, not all tools could be easily adapted for use with New Zealand's native bees or their habitats. For example, radio telemetry tools were excluded from the review [20, 96, 97, 98, 99]. This included technologies such as the harmonic radar [100]; shown in Figure 2.1. This was because New Zealand's native bees might not carry added weights [21, 29]. Some species may be capable of carrying added loads. For example, they are commonly observed covered in clay dust, shown in Figure 2.3. However, evidence suggests they have behavioural characteristics not suitable for tracking technologies [21, 29]. For instance, observations by Hart [29] indicated some species will manoeuvre around tight spaces, as shown in Figure 2.4 (a) below. They were also observed persistently attempting to dislodge added loads as indicated in Figure 2.4 (b). For these reasons, digital identification tools were considered more appropriate for studies on native bees.

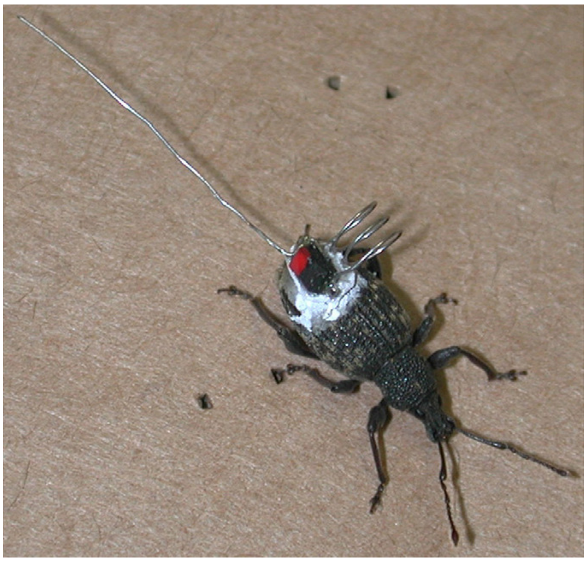


Figure 2.2: Harmonic radar transponder attached to a weevil [100, pg.53].

2.3.3 *Image acquisition: are native bees too small?*

A handful of representative studies reviewed thus far are sufficient to demonstrate the range of imaging technologies available. Most are effectively non invasive mark-release-recapture *type* tools. They could be used to assess diversity, abundance or behavioural characteristics, in a variety of species. This includes New Zealand's native bees. However, there are some aspects, of some imaging tools that could not be used in applications for New Zealand's native bees. In the first instance, the issues revolve around the type and quality of images that

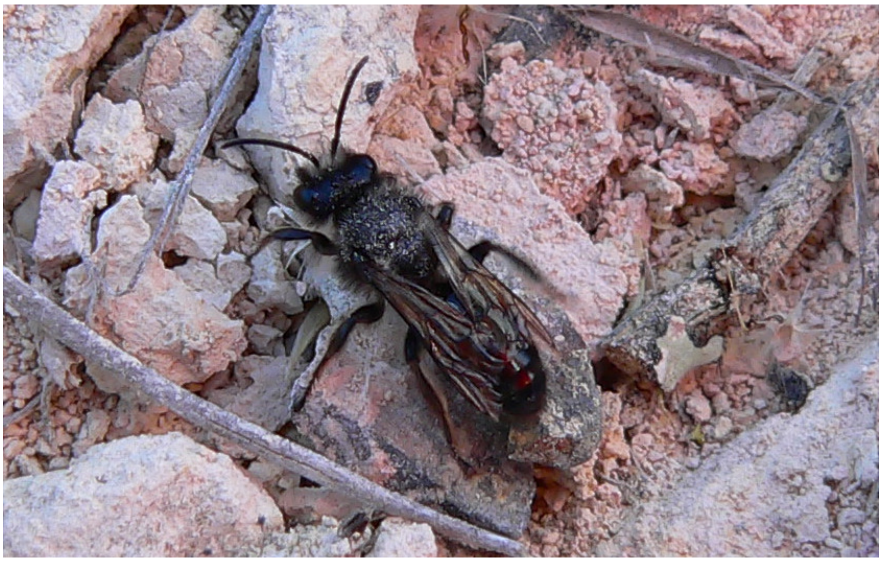
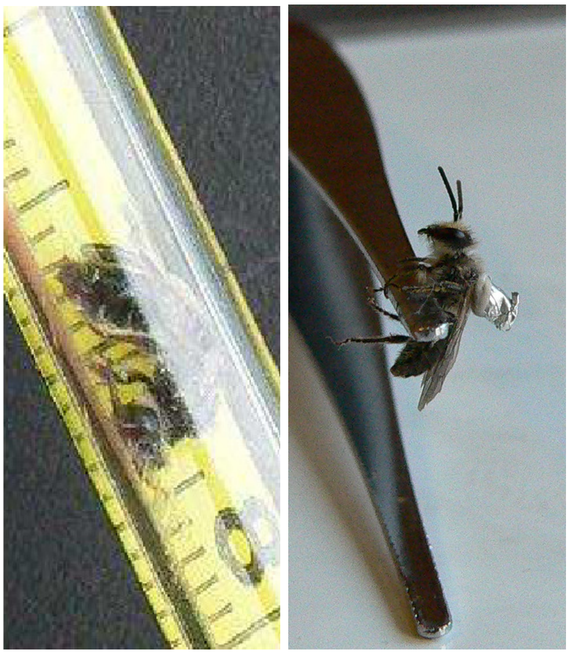


Figure 2.3: A female native bee has groomed herself to remove clay from her body; some soil grains remain on her thorax [29, pg.78].



(a) Bee inside tube.

(b) Male bee with load.

Figure 2.4: Images of New Zealand native bees (a) turning around inside a small (5 mm wide) tube [29, pg.78] and (b) attempting to dislodge an artificial load [29, pg.78].

can be gathered for analysis. Many species of New Zealand's native bees are small; especially compared to other bees [17].

In addition, closely related species are morphologically similar [17]. Thus, they can be difficult to identify properly. Previous records were used to determine the approximate species diversity, at the target monitoring locations in this study [29]. These were combined with Donovan's [17] research. Data were used to examine the biometrics of key species in more detail. The species sampled are displayed in Figures 2.5 (a)–(f), Figure 2.6 and listed in Table 2.1 below.

The beetle identification system described by Caci et al. [88], showcases the benefits of imaging techniques (Figure 2.1). An image-centric sampling method is not fatal for target species. Therefore, the methods have some advantages over traditional techniques; especially for studies involving vulnerable species. However, similar systems might not be easily adapted for monitoring New Zealand's native bees. Image acquisition is a problem because of the size and speed of bees in flight. Also, the distinguishing morphological features of bees, are only perceptible in close-up views.

Collecting good sample images is therefore a fundamental issue. A point and shoot approach, using a typical digital single lens reflex (DSLR) camera, might result in some useful scientific images. But, the method would be difficult to replicate. As shown in Figure 2.5, native bees range in body size from around 3–12 mm. Therefore, if standard images could be collected, size-features could be used to classify bees into broad family, or subfamily groups. Native bees are mostly black in colour. This is unlike the target species in the study by Caci et al. [88]. The *Rosalia longicorn* beetle in their study, is up to 40 mm long. It also has striking physical features; evident in Figure 2.1. In comparison, the facial features of native bees can only be seen in microscopic photographs; as the examples collated in Figure 2.6 below indicate.

If close-up images could be acquired, then shape or colour features from facial markings could be used to identify bees. This could be achieved using any of the techniques outlined by Caci et al. [88], Mayo and Watson [91] or Lone et al. [80]. However, image acquisition would most likely be fatal for the insects; unless they were sedated for close-up imaging and released [76]. Also, for a few species, facial features are not significantly different and could not help to differentiate species. For example, there are no obvious differences between sample images [8] and [10], in Figure 2.6. Both are male bees. However, one belongs to the species *Leioproctus (Leioproctus) huakiwi* (Hymenoptera: Colletidae) and the other to *Leioproctus (Leioproctus) imitatus* (Hymenoptera: Colletidae).

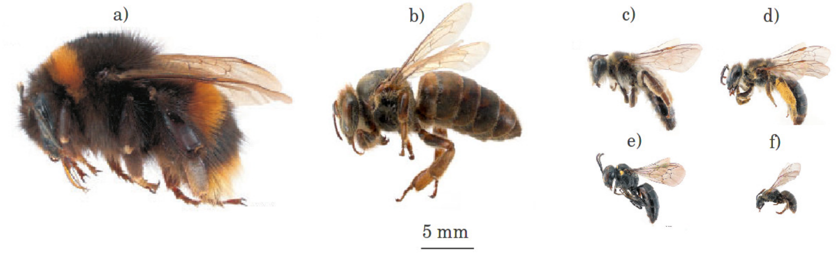


Figure 2.5: Size features to categorise bees into broad family groups. For example, a) bumble bees, b) honey bees, and c)-f) four *different* species of native bees. Images from Donovan [17, pg.130–231].

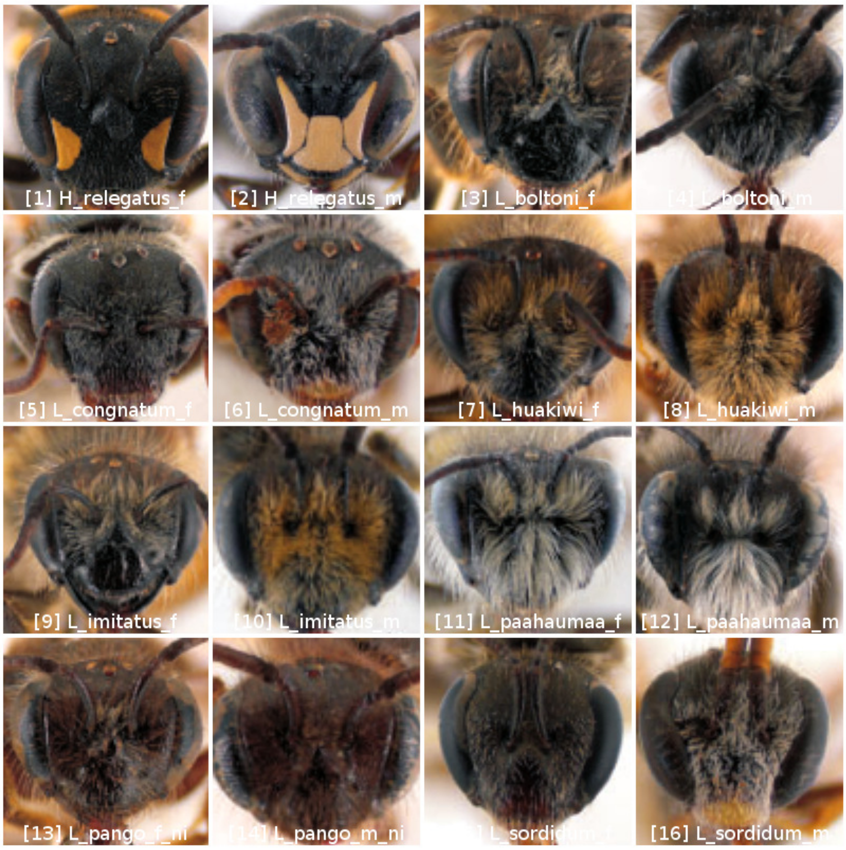


Figure 2.6: Eight species of native bees previously sampled in Whangarei [29]. Represented in close-up sample images [1-16] from Donovan [17, pg.130–231]. Refer to Table 2.1 for species names.

Table 2.1: Taxonomic species list of native bees previously sampled in Whangarei [29]; also shown in Figures 2.5–2.6.

Taxonomic tree	Image reference	
Superfamily APOIDEA		
Family Colletidae		
Subfamily Colletinae		
Genus Leioproctus		
Subgenus Leioproctus		
boltoni	[3-4] Figure 2.6	
huakiwi	[7-8]	
imitatus	[9-10]	d)♀ Figure 2.5
pango		
Subgenus Nesocolletes		
paahaumaa	[11-12]	c)♀
Subfamily Hylaeinae		
Genus Hylaeus		
Subgenus Prosopisteron		
relegatus	[11-12]	e)♀
Subfamily Hylaeinae		
Family Halictidae		
Subfamily Halictinae		
Tribe Halictini		
Genus Lasioglossum		
Subgenus Austrevylaeus		
sordidum	[15-16]	f)♀
Subgenus Chilalictus		
cognatum	[5-6]	
Family Apidae		
Subfamily Apinae		
Tribe Bombini		
Genus Bombus		
Subgenus Bombus		
terrestris		a)♀
Tribe Apini		
Genus Apis		
mellifera		b)♀

2.3.4 *Image verse audio data?*

Image acquisition might not be so challenging in the future; especially as the performance of tools continues to improve. Currently however, without good equipment and stationary target insects, imaging live native bees for scientific analysis is impractical. This is not true for audio data. Theoretically, data could be easily collected. An audio-based system could be developed to automatically classify native bees into species. The benefits of such a tool would be considerable. However, there is little evidence the technologies developed by Potamitis et al. [84], Raman et al. [85] and Batista et al. [87] have been used in wider studies. The speed at which new tools and techniques are integrated into ecological methods may depend on other factors [69, 95]. Perhaps, more could be achieved towards building greater collaborative relationships. Currently, there are many promising technologies developed for biological studies, that are not utilised [84, 85, 87]. The benefits of these tools remain largely unknown, since they can only be realised when they are incorporated into larger scientific studies.

This situation might change as the technology becomes more familiar. More so, since many of the tools are designed for open access, community science projects. New methods benefit from crowd sourced data [39, 95]. Referring to their insect sensor, Batista et al. [87] explained, "...within the limits of our budget, we will continue our practice of giving a complete system as shown to any research entomologist who requests one..." [87, pg.764]. An audio-based identification system for New Zealand's native bees could be worth pursuing in the future.

2.4 MEASURING POPULATIONS USING IMAGES

Broad image data, could also help to describe the population dynamics of species over time and space [69]. However, there are few exemplar studies in this area. There is no known research focused on monitoring the populations of native bees using digital images and image analysis. Nevertheless, the progress of imaging technologies for science is supported by open source community initiatives [92, 101]. Furthermore, there are some parallel studies, with closely related problems and possible solutions.

Counting white flies from digital images.

For example, Solis-Sánchez et al. [102] developed a machine vision algorithm to detect white flies, *Bemisia tabaci* (Homoptera: Aleyrodidae). They used the shape of white flies for identification. Geometric features such as solidity, eccentricity and area, were used to identify white flies from other insects caught in sticky traps. An example sticky trap image is given in Figure 2.7 (a) below. They reported a 97–100% accuracy, identifying insects from images of sticky traps and leaves.

Solis-Sánchez et al. [102, 103] expanded their initial system in a related study. A scale invariant feature transformation procedure was added to their original algorithm. Five different types of insects were identified, collected from hunting traps. Their results showed an improvement. Automatic and manual classifications were highly correlated. Pearson's correlation coefficients (r^2) were between 0.96–0.99. Pointing out the benefits, they explained their system could replace time consuming, manual counting methods [103].

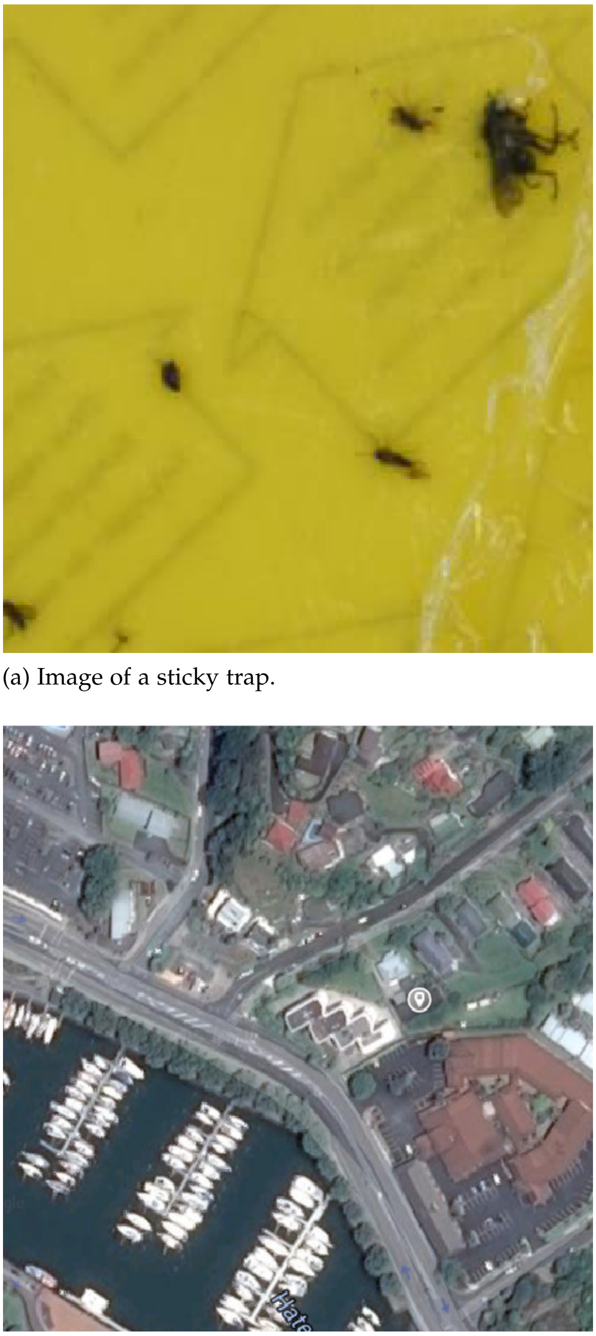
Checchi et al. [104] presented a method to count the displacement of human populations. They used high resolution satellite images similar to that shown in Figure 2.7 (b) below. Checchi et al. [104] explained, "...an estimated 43 million people worldwide are forcibly displaced due to armed conflict or other crises. " [104, pg.2]. Knowing the size of displaced populations is critically important, especially in the first few days of a disaster. According to Checchi et al. [104], good quantitative data is required to help assess the adverse affects on populations. Also to allocate aid resources effectively and mitigate problems. They used manual identification methods by counting the number of building structures in images. They explained their results were *good enough* for estimating human populations and their method was fit for purpose. For example, a ground census conducted by human personnel may not be possible during a natural emergency; but reliable quantitative population data is required as quickly as possible [104].

Displaced human populations with satellite images.

In a final related study, Johansson [105] used satellite imagery for the automatic identification of small marine vessels (an example image is shown in Figure 2.7 (b) below). He explained the levels of piracy on the sea are increasing; detecting and monitoring the activities of small vessels is important [105]. Larger ships are already monitored by automatic identification systems. However, smaller vessels are more difficult to identify, even though high resolution satellite data are now available. Until recently, imaging methodologies employed thresholding methods. Thresholding-based techniques, were traditionally used to segment image data into key components. However, according to Johansson [105], straightforward segmentation of satellite images into *vessels* and *all other objects* was a challenging problem not easily overcome. Photographs were affected by natural variations. Therefore, thresholding methods performed poorly.

Marine vessels with satellite images.

In order to compensate for this, Johansson [105] used a specialised filtering technique to highlight objects in the images. He combined the filtering with a machine learning algorithm. Johansson [105] used a *random forest* model for image-classifications. He implemented the classification methods via WEKA. He described his study as novel within the field of vessel detection. Study results were excellent; with an 85–99% classification accuracy using images obtained from Google maps. He summarised the research explaining, "...as the amount of



(a) Image of a sticky trap.

(b) Satellite image of Whangarei (Northland, New Zealand).

Figure 2.7: Parallel imaging problems. An (a) image of a sticky-trap, similar to those used by Solis-Sánchez et al [102] and; (b) a satellite image of Whangarei (from Google Maps); similar to those used by Checchi et al. [104] and Johansson [105].

data increases so does the need for efficient methods of processing such data. " [105, pg.1].

2.5 MONITORING REVIEW SUMMARY

According to research by Bischoff [64] and Cane [16], it is possible to use the active nests of native bees as a proxy for population abundance. Both studies relied on manual field counting methods; although Cane [16] used digital video data to confirm the number of active nests (i.e. those nests being used by bees). Manual nest counting methods could be substituted with image-centric techniques. Image acquisition could be incorporated into traditional field methods and provide an efficient and reliable method. Image-based techniques would not be heavily dependent on expert sampling or manual field protocols.

The future prospects of automated taxonomic identification, from images of native bees is promising. But there are practical constraints associated with the biology and behaviour of many native species. This limits acquisition of good scientific images. At least where images of live insects are concerned. It may be possible to sedate or fix insects for image acquisition [76]. However, the extra tools, time and effort involved with insect handling, may defeat the overall purpose of imaging methods [76]. This is could be equally true for audio-based collections. In addition, many of the systems reviewed were developed for automated species identification. Although the tools are important for biodiversity research, they were not designed to collect rapid or broad, spatio-temporal population data. An important objective of this thesis was to design a method to collect broad, reliable, base-line information on the health of native bee communities. A *simplified monitoring* method was also paramount.

Given image data can be reliably collected, image collation, management, analysis choices and tasks become central. For instance, Jim Gray summarised, "...People are collecting data either from instruments or sensors, or from running simulations. Pretty soon they end up with millions of files, and there is no easy way to manage or analyze their data." [106, pg.21]. Data intensive research brings many opportunities and great challenges [38, 69]. Johansson [105] and others have indicated, the ways to handle and process such vast amounts of information, are not easily defined [69, 106, 107]. There are rapid advances in scientific knowledge [108]. Technologies continually improve in performance and power; capacities double every year [67]. Increasing connectivity unites large networks of human sensors; e-science tools are rapidly developing [37, 106, 109].

The benefits of interdisciplinary, collaborative research are recognised [70, 83]. A synergy of knowledge can enrich scientific discovery [83]. Yet, while some fields embrace the era of big data, other disciplines are struggling to move toward more open research models [106, 110]. Lindenmayer and Likens [107] point out, "...in our discipline of ecology, there is an increasing number of examples where increased knowledge is missed or even where substantially flawed papers are being published, in part because authors had limited or no understanding of the data sets they were using, nor any experience of the ecosystems or other entities about which they have written." [107, pg.338]. Hampton et al. [110] questioned the role of some experts, data-intensive methods and socially relevant science. These perspectives clarify some of the current issues affecting ecological research [107, 110, 111].

In other areas of natural science, open access, big data archiving and sharing, image processing and analysis, machine learning and data mining, and e-science have been integrated into methods [69, 110, 111]. Biomedical imaging research collaborations have culminated in dynamic and useful open source tools, such as FIJI and ImageJ [92, 101]. The interdisciplinary relationships between biologists, engineers and computer scientists, are well established in the field of medical image analysis [92, 101, 112, 113]. But, there are no such clear paths for ecological research, and there are a limited number of example studies to draw upon. In this research, there were no known examples, either in New Zealand or abroad, where images have been used to monitor populations of ground nesting solitary bees. Nevertheless, there was some inspiration found in a wider domain, by looking at similar problems and parallel studies.

Therefore in the final sections of this chapter, a number of analogous studies were reviewed. By evaluating representative imaging methods and closely related studies, the probability image-centric techniques could be successfully applied in studies of native bees was established. The advances in biomedical imaging systems have had a significant impact in the fields of human biology and medicine. Methodologies can also be applied to a range of other life science research, including those outlined in this thesis. In the next chapter, a review of the developments in open source, biomedical imaging systems is presented.

3 IMAGE ANALYSIS THEORY

Synopsis

The scope of image analysis is wide. Therefore, this review was restricted to biomedical imaging methods, open source software tools, and the advances in machine learning imaging applications. General biomedical methods and generic image-system workflows are introduced first. This is followed by an overview of open source concepts and tools. The topic of outdoor imaging is discussed within the context of image acquisition, formats and pre-processing operations. Image segmentation tasks, challenges and solutions are defined. The advances in machine learning and interactive image segmentation techniques are outlined. The concluding sections in this chapter are dedicated to the methods used to measure the performance of imaging systems.

3.1 IMAGING AND SCIENTIFIC DISCOVERY

Science and photography have a close interdependent history, dating back to early *heliographic* experiments in the 1820's [114]. Imaging technologies are continuing to develop rapidly [109]. The capabilities of modern imaging systems are remarkable. Within the scientific domain new technologies are contributing to all manner of discoveries. From the intricate workings of the human brain, in single photon emission computed tomography imaging systems [115]; to mapping the universe, with the billion-pixel camera on the space probe Gaia [116]. The scope of image analysis is wide. It includes many distinctive strands of knowledge. These are summarised in Table 3.1.

Overview of quantitative analysis of microscopy images.

The most comprehensive reviews regarding imaging methods are often directed towards biomedical image analysis. Microscopy is one of the fundamental tools of biology. Traditionally, scientists relied on visual interpretations of microscopic images [112]. Consequently, many advances in automated imaging methods have been developed by biologists, for biological applications [101]. For example, Ljosa and Carpenter [117] described a range of techniques used in imaging systems for the automated image analysis of microscopy images. They presented a clear overview of current methods; providing a list of imaging resources. They also outlined the pitfalls to consider when using or designing quantitative imaging systems [117].

Across disciplines image-tasks and pipelines are generally similar. Therefore, their review has provided some useful theory for other

Table 3.1: Image analysis scope and topics.

Field of discipline	Main tasks
Image processing:	Designing or applying filters or operators that change the basic aspects of an image (i.e. enhancing contrast and brightness).
Pattern recognition:	Designing or applying the algorithms/-code to automatically identify patterns in data.
Computer vision:	A complete system design, from image acquisition to analysis (i.e robotic vision in manufacturing processes).
Machine learning:	Designing or applying the algorithms used in semi-supervised/supervised machine learning scenarios.
Data-mining:	Scanning for patterns that emerge from analysis of large quantities of data to help knowledge discovery.

novel imaging research [117]. Similarly reviews, by Sharmir et al. [118] and Antony et al. [119] are equally informative. Both have provided summaries of current software tools and techniques for life science image research [118, 119].

A guide to image analysis for ecology and evolution.

A handful of reviews are written for other areas of life science disciplines. For example, Pennekamp and Schtickzelle [120] presented a hands on guide for imaging techniques in experimental laboratory systems. They introduced past, present and future benefits of technologies, with an overview of methodologies for experimental laboratory systems. They explained, "...despite the advantages of image analysis, the technology has not been fully adopted yet, presumably due to the difficulties of technical implementation." [120, pg.485]. They pointed out the range of benefits that could arise if automated image analysis and experimental laboratory systems were integrated [120].

Automated species identification systems for biology.

Gaston and O’Neill [72], reviewed imaging systems for pure and applied biology; they questioned the reasons why automated species identification has not been widely adopted. They investigated some core issues, to determine if automated species identification would be a realistic option looking towards the future. They addressed the suggestions that the tasks are too difficult, too threatening, too different or too costly. Overall they concluded "...vision and enterprise were more limiting...than the practical constraints of technologies." [72, pg.1].

3.2 IMAGE SYSTEMS WORKFLOW

Image analysis workflows consist of two major steps; 1) image acquisition (capturing digital images), and 2) image analysis (manipulating and measuring digital image data). These are generic stages common to most applications and disciplines. Glasbey and Horgan [112] described five distinct stages of image analysis; these are listed below and expanded on, in Figure 3.1.

- *Acquisition and display*: Capturing raw digital images; viewing an array of pixel values as an image on a digital camera or computer monitor.
- *Pre-processing*: Enhancing images by applying filter transformations to groups of pixels.
- *Segmentation*: Dividing an image into regions by sectioning or classifying pixels into different areas of objects.
- *Post-processing*: Applying operators that relate to the size and shape characteristics of objects to extract information from images.

Within medical imaging disciplines, careful attention is given to the quality[1] of raw image data, particularly with regards to method replications [113]. Guidelines on the appropriate use and manipulation of scientific digital images were presented by Cromey [122]. The recommendations are broadly relevant to other scientific imaging applications. However, some points are not applicable when methods are based on natural outdoor images (see points 4.* and 5.**).

1. Manipulation of digital images should only be performed on a copy of the unprocessed image data file.
2. Simple adjustments to an entire image is usually acceptable.
3. Cropping an image is usually acceptable.
4. Digital images that will be compared to one another should be acquired under identical conditions*, and any post-acquisition image processing should also be identical.
5. Avoid the use of lossy compression.**
6. Use care when changing the size (in pixels) of digital images.

[1] By definition, *image fidelity*–is the ability to visually discriminate between two images and *image quality*–is the preference for one image over another [121].

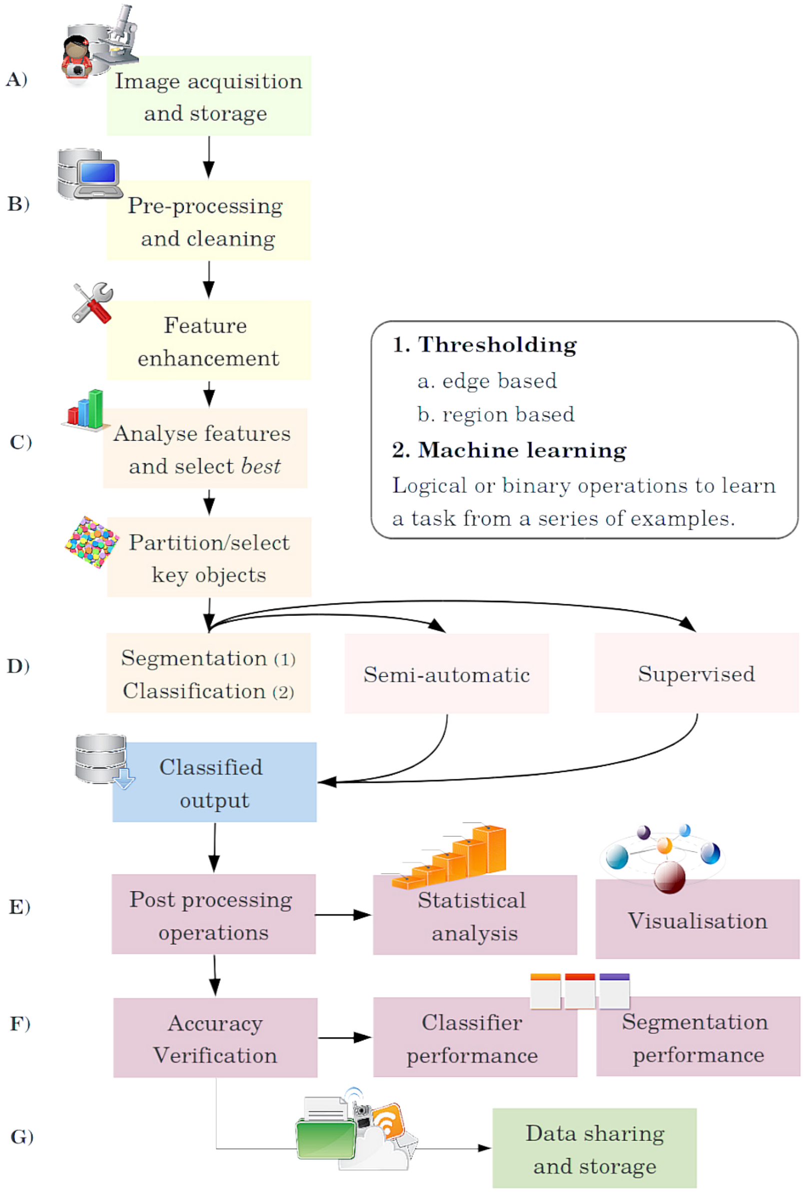


Figure 3.1: Typical image analysis stages and design.

The advances in electronic hardware tools and technologies have impacted the developments within the fields of image analysis. Off-the-shelf digital cameras are increasingly more powerful. They parallel the advances in computers and imaging software tools. Public access to technologies is changing scientific landscapes, since many of the tools and methods are readily available, easy to use, open access and open source. For example, Antony et al. [119], reviewed imaging software tools. They found open source packages offered substantial benefits. Particularly with regards to reproducibility of methods [101, 119, 123]. The importance of open access was declared by a group of Nobel laureates. They wrote, open access "...expands shared knowledge across scientific fields..." and suggested it is the best path for *accelerating multi-disciplinary* breakthroughs in research." [124].

Open source, community driven image systems for biology.

Open source tools, methods, and reporting, can work in unison to advance scientific knowledge [125]. The versatility of software tools, access to code, and replication of methods are important factors that are worth considering during research design. Antony et al. explained "...open source solutions facilitate community driven efforts in the development of image analysis." [119, pg.13]. The extendibility, interoperability and scalability of software platforms are key considerations. They have an impact on overall research methodologies. These are defined as:

1. *Expandability*: A system design principle taking future growth into consideration.
2. *Interoperable*: The capability of different programs to exchange data using a common set of formats to read and write, in the same formats; and to use the same protocols, present or future, without any restricted access or implementation.
3. *Scalability*: The ability of a system, network, or process to handle a growing amount of work in a capable manner; or its ability to be enlarged to accommodate that growth.

Although there are many good proprietary platforms available, open source bio-imaging software solutions are very popular and continue to be at the centre of many scientific advances. Fiji is Just ImageJ (FIJI) is a popular biomedical imaging toolbox [101]. The software has been carefully designed to consider all three factors listed above and is used in a range of applications. There are other packages commonly used for image analysis including: ImageJ, Icy[2], the Matlab plus Image Processing Toolbox and R–EBImage package [92, 126, 127, 128].

Image database design, are becoming vital components for research involving big data analysis. Sharing protocols are well established in

[2] *Icy*- http://icy.bioimageanalysis.org/about

some disciplines. For instance, genomics, astronomy and meteorology. Therefore, substantial care is afforded to database design and access. But, the shift towards more open models have not been adopted by all disciplines [129]. community driven image systems for biology. Similar points have been discussed previously, in relation to the use of big data and the slow adoption of automated image analysis [72, 110]. Data sharing is not always considered virtuous [107]. Some authors have suggested paradigms are slow to change for some fields [110, 130, 131]. However, the progress towards open access, open source and data sharing, is well advanced in a range of interdisciplinary life science disciplines. These disciplines and sub-areas, can provide good examples to follow [34, 132].

Big-data sharing, cloud-services and data-base options.

For example, the Knowledge Network for Bio-complexity (KNB)[3] provides the facilities to store, share, discover, access and interpret complex ecological data. It can be integrated with the open source statistical package R, or used with a desktop application, *Morpho*. Users can create metadata, search, edit or view data collections. There are a range of other generic options which can be used to store data online. For example, cloud storage such as Dropbox [4] can be used for single users; OpenStack is used [5] for large distributed community developments [133]. GitHub[6] is also popular for building online repositories. The platform is often used for collaborative projects and code design. GitHub was not specifically designed for sharing data. Nevertheless, it has been used as a centralised hub for smaller research projects [134, 135].

3.4 Image acquisition, formats and pre-processing

Digital image acquisition, formats and pre-processing methods are important aspects of image pipeline designs. In biomedical imaging fields great attention is placed on the quality of raw image data; particularly with regards to method replication [113, 136]. Final decisions can have a cumulative impact on analyses and validity of results. The quality and consistency of image capture is dependent on the acquisition methods. Under laboratory conditions important factors can be controlled such as lighting (e.g. brightness and background illumination) and the total capture area. Because there is greater control over image acquisition, analyses are more reliable. The methods are easier to replicate [137].

Mitigating affects of complex, natural images

Natural images are complex. Consequently, it is not possible to achieve the same degree of control when imaging under natural outdoor situations. Even so, a number of techniques can be used to help

[3] *KNB*- https://knb.ecoinformatics.org/
[4] *Dropbox*-https://www.dropbox.com/
[5] *Openstack*-http://www.ubuntu.com/cloud/ubuntu-openstack
[6] *GitHub*-https://github.com/

standardise acquisition. For example, Burks et al. [138] used image analysis to identify different types of weeds. They designed a specialised image acquisition system to capture field images. They used a wheel mounted, self contained imaging device. It included four adjustable flood lights, with diffuser covers to eliminate shadows [138]. They described the variable natural conditions which affected image acquisition, saying "...the combination of windy conditions with partly cloudy skies produced rapid changes in ambient light conditions and created significant leaf movement. " [138, pg.444]. They explained, that in order to reduce the affect of changing conditions, a diffuse off-white cotton cover was placed over the camera system. They used supplemental lighting. This was to compensate for the effects of cloud cover and the angle of the sun. A nylon canvas was also used as a wind break. It was placed at the base of the camera system. This was to prevent excessive motion in the plant leaves [138].

Aquisition techniques for complex, natural images.

Over the last decade, image acquisition techniques for agriculture have advanced considerably. For example, unmanned aerial vehicle's (UAV's) are more commonly used to capture hyper-spectral images of crops [139, 140, 141, 142]. The challenges described by Burks et al. [138], are still fundamental constraints for many outdoor imaging applications. Nevertheless, there have also been significant developments in imaging methods. Although core image processing techniques have not changed, there are a range of new image *Classification methods for complex images.*. Some of these can help to mitigate the affects of inconsistent image capture. For instance, Feng et al. [143, 144], employed an off-the-shelf digital camera for UAV imaging. Their system was designed to evaluate the impact of urban flooding in Yuyao, China. Rather than using more sophisticated imaging equipment (e.g. multi and hyper-spectral sensors), they paired the low spectral resolution digital single-lens reflex (DSLR) camera, with a random forest (RF) classifier. They reported good outcomes. Random forest (RF) classifiers offer some advantages over other image segmentation procedures. For example, they are particularly robust to noisy image data [145, 146]. They are a good choice for the studies by Feng et al. [143, 144] and other closely related imaging problems [145, 146].

Classification techniques for complex, natural images.

Data acquisition, format and storage decisions, are interrelated. The affects of image data compression can be an important consideration when designing imaging systems. Large data volumes create storage problems and impact the speed of image processing [147]. In most camera systems, data can be saved as raw uncompressed files; or small compressed files which contain less digital information. Raw files retain full data but they are at least double the size of JPEG (joint photographic experts group) files; they require more on-camera storage space and processing resources. Because smaller JPEG files are quicker to upload to online repositories, they are frequently used in

Image formats, quality, quantity and compression.

remote sensing applications. They continue to be used in a range of image-centric studies [148, 149].

There are no agreed standards for raw files, so proprietary platform dependent software is often bundled with off-the-shelf DSLR cameras for file processing. Open source software can help to mitigate proprietary format issues. There are some flexible solutions, such as XnView[7]. The package has the utilities to open raw files and export to a range of standard image formats [150]. Biomedical imaging software FIJI and ImageJ, also have plug-in modules specifically for opening raw files [92]. However, reading and converting raw files, is an added step in the image processing pipeline. Conversion can be resource intensive compared to ready-to-view formats.

Nonetheless, the quality of image data can impact the reliability of image processing. At least where biological image analysis is concerned, high resolution raw images are preferred; they are easily captured under laboratory conditions. When outdoor image data is required on-camera SD (secure digital device) memory card capacity may be limited. Compressed image formats are more practical and cost effective. Many imaging studies do not specifically outline image format and memory considerations, but there are several comparative studies investigating the affect of compression, on image classification [147].

Remote sensing, image compression and classifications.

For example, Zabala and Pons [148, 149] compared the affects of JPEG and J2P (JPEG 2000) compression on remote sensing image classification, for mapping crops and forest areas. They found overall that J2P compression, was more reliable than JPEG, at least for the specific categories of images tested [148, 149]. Others have shown image compression is not necessarily an issue for some classification tasks. For instance, Paola and Schowengerdt [151] tested three different classification scenarios. They found that high quality classifications could still be achieved with a compression ratio (CR) of 10:1. Image CR is defined as the number of bytes of the original image, over the number of bytes of the compressed image as follows:

$$\mathrm{CR} = \frac{\textit{original image data volume}}{\textit{compressed image data volume}} \tag{1}$$

Levels of compression below 10:1, are within the boundaries considered acceptable for image classification applications [147, 149, 151].

Finally, in most imaging methods data is normalised before analyses are performed. For example, in FIJI, images can be prepared using operators such as *contrast enhancement* or *histogram equalisation*. Other common pre-processing steps include image cropping, transformations such as flipping, or conversion from RGB (red-green and blue) to grey scale. Image conversions are frequently required before segmentation operations can be applied. There are generally several options

[7] *XnView*- http://www. xnview. com

available (e.g. conversion from RGB into 8, 16 or 32 bit grey-scale images). The methods used to achieve image segmentations, are at the centre of current research developments within the field of image analysis. These are discussed in greater detail over the next sections.

3.5 IMAGE SEGMENTATION TASKS

In image analysis, a *region*, is a group of pixels that have similar properties. Regions are used to help image interpretation, but they must be correctly *partitioned* into areas that represent objects, or parts of objects. *Image segmentation* is a process in which regions sharing similar characteristics such as intensity, texture or colour, are grouped together to form multiple segments or collections of pixels[8]. Procedures for image segmentation are multifarious. Some studies employ statistical classification techniques, thresholding, edge and region detection; others use any combination of these techniques. The final segmented output in an imaging system pipeline, is a set of classified elements represented as a binary image.

Image segmentation, global thresholding and alternative methods.

Thresholding is a region-based, direct method; it is used to turn grey scale images, into black and white - binary form. Although it is unambiguous, good segmentation results can be difficult to obtain. When a more generalised imaging approach is required, classification techniques often perform well. In this instance, regions of pixels are sorted into classes by way of statistical methods, or algorithmic machine learning techniques. The terms binarization, segmentation and classification, are closely related. A classifier implicitly segments an image; segmentation implies classification, and the final output of a classification-segmentation process, is a black and white image.

The selection of one technique, over another, is partly subjective. Also, the *performance* of different methods are relative. However, good segmentation techniques are those where 1) pixels in the same category, have similar values and form connected regions; or 2) neighbouring pixels which are in different categories, have dissimilar values. The primary aim of all segmentation techniques, is to quantify aspects of image data using reproducible and objective techniques. This includes a capacity to *generalise* over a given range of image data variability. The final results produced by any segmentation, ultimately depends on the original image content and quality, the specific application constraints and characteristics; and the intended use of the information required to be extracted from images.

Intensity based thresholding methods produce straightforward segmentations. They are simple, direct and easily programmed. If there are foreground objects or features in images that are *defined by intensity*, then threshold procedures can outperform other methods. As

[8] Segmentation can be defined as " the division of an image into spatially continuous, disjoint and homogeneous regions." [152, pg.215]

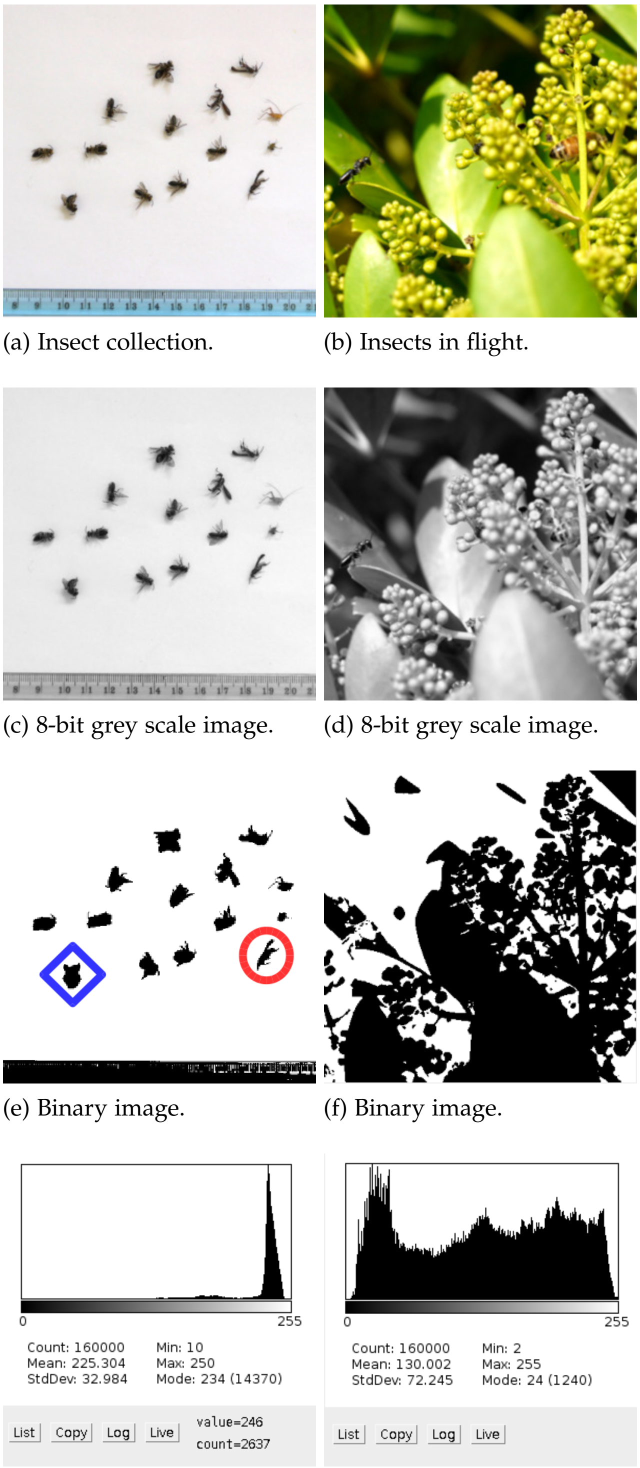
Wang et al.[153] write, "...image segmentation is one of the most important and fundamental tasks in image processing and techniques based on image thresholding are typically simple and computationally efficient." [153, pg.117]. Because there are no extra parameters to tune, thresholding is fast and requires minimum processing resources. Considerable research effort is devoted to the methods used to effectively categorise or classify important parts of an image. Threshold, difference image, edge detection and watershed are among the most widely used image processing techniques for biological image data. But, as Ljosa and Carpenter [117] point out, imaging tasks are increasingly more complex and the volume of data is growing.

In some cases, traditional image analysis methods, such as thresholding, might not suffice. Traditional image processing techniques work well on images that are acquired under controlled conditions. However, when collected from natural environments, the processing tasks can become difficult. For example, Figure 3.2 below shows two imaging pipelines. On the left, a simple thresholding task and on the right, a difficult task. In this pipeline the images were first converted into an 8-bit grey scale format; Figures 3.2 (c)–(d).

In the next step, they were *made* binary. This was achieved by selecting a global threshold value set between 0-130 pixels. In Figures 3.2 (e)–(f), the pixels with values less than 130 are foreground objects, and were converted to a binary value of 255 (black). Pixels above 130 were background objects; they were assigned a binary value of zero (white). The distribution of pixel intensities are evaluated via the histograms for each pipeline. These are shown in Figures 3.2 (g)–(h). The binomial histogram, described in Figure 3.2 (g), indicated this type of image could be successfully segmented by selecting a global threshold value.

In addition, the binary image could be further analysed (morphological operators are frequently used during this stage). For example, insects could be categorised into species. This could be achieved using size and shape characteristics only. Native bees have a rounded body shape. This is highlighted with the blue diamond overlay in Figure 3.2 (e) below; showing a species of native bee (*Leioproctus spp.*). There are some distinguishable differences in body shape, when compared to native parasitic wasps (*Pseudofoenus spp.*). As shown in the same image (Figure 3.2 (e)), using the red circle overlay, native wasps have long slender abdomens.

The insect collection image pipeline on the left, provides a good example of a straightforward segmentation task. In contrast, the image of insects in flight on the right, is an example of a more ambitious assignment. The image in Figure 3.2 (b) shows a native bee in flight, and a honey bee foraging on the flowers of the coastal five finger plant, *Pseudopanax lessonii* (Araliaceae). The multimodal histogram suggested no *single* threshold value could be used to make



(a) Insect collection.

(b) Insects in flight.

(c) 8-bit grey scale image.

(d) 8-bit grey scale image.

(e) Binary image.

(f) Binary image.

```
Count: 160000    Min: 10
Mean: 225.304    Max: 250
StdDev: 32.984   Mode: 234 (14370)

value=246
count=2637
```

(g) Bimodal histogram.

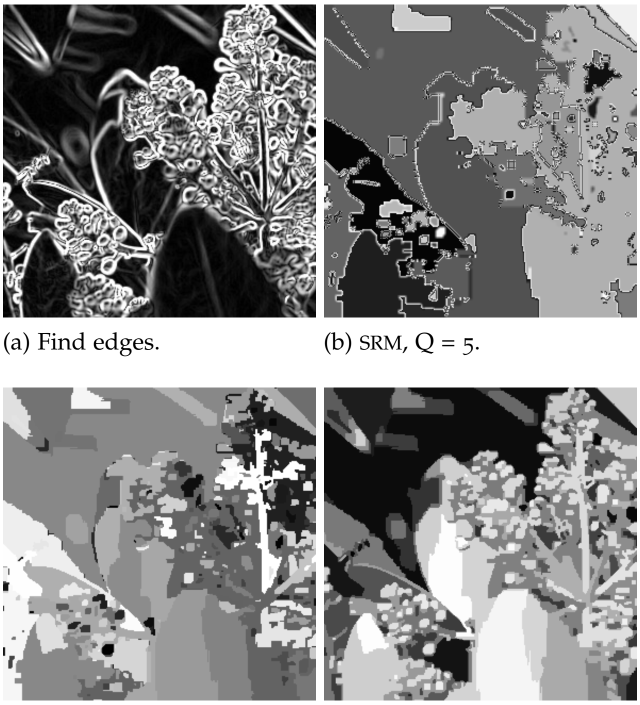
```
Count: 160000    Min: 2
Mean: 130.002    Max: 255
StdDev: 72.245   Mode: 24 (1240)
```

(h) Multimodal histogram.

Figure 3.2: Images of (a) insect collections and (b) insects in flight converted into 8-bit grey scale images, (c)–(d). Global threshold applied to *make* images binary. Insect collections were sufficiently segmented (e) as indicated via the (g) bimodal histogram. Body size and shape to categorise bees (e)–blue diamond, from wasps (e)–red circle. In contrast, no threshold value was suitable for (f) images of insects in flight; as indicated by the (h) multimodal histogram.

the image binary (Figure 3.2 (h)). Thus, it was not possible to select a *global* threshold value to adequately segment the image to show, 1) both species of bees and 2) all other background data. Figure 3.2 (f) demonstrates the images of *both* insects were almost lost during the binary conversion.

In cases where images are not easily segmented, there are some alternative approaches. Frequently, other characteristics can help to define an image. For example, connected structures, outlines, areas or textural qualities. Two alternative methods used for segmentation are edge detection, and region merging. These are briefly introduced in the next paragraph and shown in Figure 3.3 below.


(a) Find edges.

(b) SRM, Q = 5.

(c) SRM, Q = 15.

(d) SRM, Q = 25.

Figure 3.3: Edge detection (a) and SRM (b)–(d) methods to segment the challenging image of insects in flight. SRM segmentation *regions* change as larger values of Q are used.

Canny-Deriche filtering is a popular edge-detection method [154]. The α parameter, controls the degree of smoothing applied; where the default value is 1.0. Greater values suggest less smoothing, but more accurate detection; lower values suggest more smoothing but less accurate detection. *Statistical region merging* (SRM) is a region-based method. It performs well on a range of images [155, 156]. The algorithm examines one region per pixel. A statistical test is applied to neighbouring regions in ascending order of intensity differences. Tests are applied to determine if the mean intensities are sufficiently similar enough to be merged. Segmented regions can be represented by mean grey values, or by an index of the regions. Q – is the setting determining the approximate number of regions to segment. Above in Figure 3.3 (b), when $Q = 5$, images of both bees are lost in the image background. However, in Figure 3.3 (d), the images of the native bee and honey bee, are not *completely* lost.

According to Nock and Nielsen [155], SRM is a fast segmentation algorithm, that copes well with noise. As demonstrated in Figure 3.3 (d), although not ideal, a greater degree of segmentation was achieved compared to the global thresholding method and result, presented in Figure 3.2 (f).

3.6 INTERACTIVE SEGMENTATION METHODS

Few traditional techniques can be used on highly variable images, since methods cannot be applied to generalise over a wide range of pixel intensities. In these circumstances, machine learning techniques are more effective. This issue is also the basis for the growing tendency towards using machine learning in challenging imaging tasks, as proposed by Ljosa and Carpenter [117].

Trainable image segmentation techniques work by utilising human visual knowledge, to provide a machine learning algorithm with a set of *expertly labelled examples*. For instance, in FIJI the TWS plug-in requires a user to provide two sets of labels [157]. Region of interest (ROI) tools are used to select pixels samples belonging to foreground, and background objects. Filters are applied to original image data and used to create a separate *features stack*. In TWS, a user may select any combination of filters from a possible twenty. Image-filters can be grouped according to their main filter functions. For a summary of TWS filters, refer to Table 7.1, in Chapter 7 (pg.129).

During the learning process, a selected algorithm uses the examples provided, and the features stack, to construct a classifier. The classifier can be used to segment similar types of images, including the one it was trained on. It can be ambitious to ascertain what combination of image features best describes *key* objects. Therefore, some aspects of classifier training are subjective. There are a number of other confounding decisions to consider when applying machine learning algorithms to image segmentation tasks. However, providing a user a) selects appropriate representative pixel samples, and b) chooses filters that will provide a rich features stack for analysis, machine learning algorithms work very well. They can surpass other methods. Especially on challenging imaging problems.

3.6.1 *Trainable WEKA Segmentation workbench*

The TWS is a FIJI plug-in designed for pixel level segmentation, based on semi-supervised learning. The workbench was primarily developed for biomedical imaging applications. As previously outlined, a user selects a set of features (e.g. such as edge detectors or texture filters) and using ROI tools, interactively selects pixel traces representing at least two classes. When classifier training is initiated, the features of the input image will be extracted and converted to a set of vectors of float values (the format expected for WEKA classifiers). Based on the samples provided, a classifier is trained. Results are returned as a segmented image. This is shown as a semi-transparent overlay, corresponding to the class colours. The process is normally repeated until satisfactory segmentations are obtained. Good segmentation results are typically achieved over two training sessions. Thus, the process is iterative and *interactive*. The time taken to train and classify data is variable. It depends on 1) the image size, 2) the amount of features selected, 3) the chosen classifier and 4) the number of cores of the processing machine. The default classifier in the TWS plug-in is the *Fast Random Forest*. The algorithm is a multi-threaded version of random forest. The forest is initialised with 200 trees and 2 random features per node. There are five default training features (from a possible twenty). Default filters automatically selected include: Gaussian blur, Hessian, Membrane projections, Sobel filter and Difference of Gaussians.

3.6.2 *Training procedure*

The general procedure used for classifier training in TWS is outlined in Table 3.2 below. Most training consists of a least two runs. The first returns broad segmentations; the second is optimised by selecting a few previously miss-classified ROI areas for re-classification. The main concept, is to supply the RF classifier with a small representative sample of classes. This enables the classifier to *learn* to recognise, rather than *memorise* target pixels. When the results of classifications are satisfactory, it is possible to save a fully constructed classifier for use in similar tasks. Trainable Weka Segmentation (TWS) provides both the option to save as a standard WEKA model file, or immediately apply a constructed classifier via the main graphical user interface (GUI). The training annotations can also be saved as an ARFF (attribute-relation file format) file. The ARFF file contains all the feature vectors derived from the pixels, belonging to each trace during training. During classification on new previously unseen images, the classifier can also be re-trained to incorporate new information based on other images and saved.

3.6.3 *Random forest machine leaner*

The default classifier in the TWS workbench is the RF. However, there are at least fifty other WEKA classifiers available [93]. Each machine learning algorithm is equipped with a set of parametrisation tools.

Table 3.2: Interactive segmentation method with TWS.

1 Select a minimum of two traces – representing two classes.

2 Train the base classifier and check output image.

3 Tune classifier by selecting incorrectly segmented pixels:

   a) Assign pixels to the correct classes for re-classifying.

   b) Train final classifier.

4 If segmentation is satisfactory then apply to all other *similar* images.

5 OR optimise classifier to reduce errors, increase processing speed by adjusting the:

   a) Features provided.

   b) Classifier type.

   c) Tuning parameters.

   d) Training set-up.

These settings are used to tune the classifiers for specific tasks. It is also possible to custom design WEKA classifiers and create new models for specific applications [123]. Traditionally, support vector machines were considered state-of-the-art. They have been used successfully in a range of classification applications using real-world data [158, 159, 160]. Until recently, few machine learners have surpassed the performances of support vector machines or neural networks.

But according to a recent study, RF's are most likely to perform the best [161]. Fernandez et al. [161], evaluated 179 classifiers from 17 families of learners. They undertook comprehensive evaluations by implementing the classifiers in WEKA, R, C and Matlab; using the whole UCI (University of California, Ivine) data base[9] (121 data sets). Their objectives were to determine which of the classifiers were most likely to perform the best on *any* data set. The results were extensively reported, covering all aspects of classifier optimisations data set partitioning, and test configurations. Concluding that three out of the five best classifiers were from the random forest family; closely followed by support vector machines, neural networks and boosting ensembles [161].

A range of other studies have reported the on benefits of RF's. They continue to be developed and used in a range of seemingly unrelated applications. For instance, RF's have been applied to studies in: astronomy [162], biomedical imaging [163], economic forecasting [159], genetics [164], pharmacology [165], species identification [78], ecological modelling [166, 167, 168], remote sensing for land-cover classifica-

[9] https://archive.ics.uci.edu/ml/datasets.html

tion [169], carbon mapping [131], forest classification [146], UAV flood mapping [143, 144] and invasive species mapping [170, 171].

The breadth and range of studies suggests the RF is a flexible all-round classifier, suited to a range of natural real-world data. However, the study by Fernandez et al. [161], is the first to properly quantify the performances of RF classifiers. It is likely the developers of FIJI and TWS selected the RF classifier as the default model, based on anecdotal evidence of segmentation performances in a range of biomedical imaging applications.

3.7 PERFORMANCE ANALYSIS

Biomedical imaging methods frequently employ three quantitative segmentation metrics. They include the pixel, rand and warping errors. For image segmentation analysis the ideal metrics for machine-human disagreement should tolerate minor differences in boundary location, penalise topological disagreements, and serve as a cost function for supervised learning [172, 173]. The three metrics are defined as:

1. *Pixel Error*: the squared Euclidean distance between the original and resulting images. The lower the error, the greater the agreement is between images.

2. *Rand Error*: a measurement of similarity between clusters of pixels. The lower the error the greater the agreement is between images.

3. *Warping Error*: a measurement that penalises topological disagreements and is used as a direct cost function of segmentations. The lower the error the greater the agreement is between images.

In many applications segmentation metrics are more informative. This is because they can indicate how well images are partitioned into categories [172, 173]. This is important for biomedical imaging problems. Performance measures have often been based on visual-image checks, using ground-truth techniques. Until recently, *classifier performance* has been difficult to quantify. Unnikrishnan et al. [174] point out, "...the evaluation of segmentation algorithms thus far has been largely subjective, leaving a system designer to judge the effectiveness of a technique based only on intuition and results in the form of a few example segmented images. This is largely due to image segmentation being an ill-defined problem? There is no unique ground-truth segmentation of an image against which the output of an algorithm may be compare." [174, pg.929].

In machine learning research, other parameters are used to determine the predictive accuracy of a classifiers. These include the overall number of correctly classified instances – often in the form of a confusion matrix, the number of true positives and negatives, precision, recall, and the F-measure. Even a trivial classifier that incorrectly predicts every case as the target class, can still achieve a high accuracy depending on the measures used. Therefore in machine learning research, it is important to use the appropriate performance measures and analytical set-up. Unbalanced datasets can further complicate analyses. However, in many real-world classification problems, data are unbalanced. It is more common to use the Kappa statistic in such cases. It is more informative because it is a *chance-corrected* measure of agreement between classifications and true classes [175].

The statistic is calculated by taking the agreement expected by chance away from the actual observed agreements Then dividing this value by the maximum possible agreement, as follows:

$$\kappa = \frac{P(A) - P(E)}{1 - P(A)} \tag{2}$$

Where $P(A)$ is the observed agreement and $P(E)$ is the expected agreement. The values of κ are constrained to the interval $[-1, 1]$. A value of $\kappa = 0$ indicates a complete absence of agreement; $\kappa = 1$ shows a very strong agreement. Any value above zero indicates the classifier is, at the very least, performing better than by chance alone.

There are a number of other performance techniques that can be used to verify the validity of image analysis methods. The specifics of verification methods, are highly application dependent. For example, in biomedical imaging applications, visual inspection of images provides a ground-truth test for segmentation performances. In contrast, in machine learning applications, methods for testing classifier performances can take the form of the percentage of correctly classified pixels.

3.7.1 *Agreement measures*

Measurements of agreement are frequently needed to assess the acceptability of new or generic processes. This includes agreements between methodologies, formulations (in areas of laboratory performance), instruments, assay validations, methods, statistical control processes, goodness of fit and individual bio-equivalences [176]. In some applications it is possible to visually inspect *real data*, without the aid of equipment. In these situations, the dependence on ground-truth images, segmentation metrics, and/or classifier performance testing, is not as critical. For example, Stewart and McDonald [177], used imaging methods to measure the impact of the pathogen *Zymoseptoria tritici* on wheat plants. Although visual estimates from images are relatively straightforward, they are not always reliable or repeatable. Explaining this issue, Stewart and McDonald [177] wrote, "Manually counting individual *pycnidia* on a large scale is prohibitively time consuming and visual estimation, although much quicker, is subjective and prone to error. "[177, pg.989]. They concluded that automated digital image analysis, could address all of the issues encountered. At least one other study has reported similar issues [178]. In order to evaluate the measures further, Stewart and McDonald [177] tested the *manual-scorer* and *automated-imaging* counts directly. They used Lin's concordance of correlation coefficient [179]. They found the counts derived from automated image analysis, were more reliable, precise and accurate, compared to those given by visual scorers [177].

Verification methods are also dependent on the mechanisms used in an imaging pipeline. For example, the verification tests used in an interactive segmentation method (i.e. in the TWS workbench) are dependent on the classifier model selected. In contrast to other machine learners, the RF algorithm has an internal error mechanism. The RF takes a bootstrap of all the data provided for training. During construction, individual training sets for each tree are generated from the original set, using sampling with replacement. The samples, which are not chosen for training, are called the *out of bag samples*. They are used to calculate the out of bag error (oob). It is an unbiased estimate of the generalisation-error. In contrast to other classifiers, there is no need to perform cross-validation tests to get an unbiased estimate of the true test set error.

3.8 SUMMARY OF IMAGE ANALYSIS REVIEW

The scope of image analysis is wide and technologies are continually developing. Many of the advances have evolved from the fields of biomedical imaging. Image analysis is central to scientific methods, particularly where microscopy is concerned. Therefore, techniques are well established and provide a good basis for reference. Imaging methodologies for other life sciences, such as agriculture and ecology have not been as widely reported. This situation is changing, as the hardware tools and software methods are becoming increasingly more powerful. Open source technologies are vital. Many of the developments within the field of image analysis have depended on, or contributed towards reproducible scientific methods.

A review of some key aspects of image analysis theory and design were outlined in this chapter. The literature was discussed within the context of an image-centric method to monitor New Zealand's native bees using images of active nests. Machine learning options work well for studies that depend on variable images which are difficult to segment using conventional techniques. Outdoor imaging applications, off-the-shelf camera equipment and JPEG compression are factors that impact image quality. New machine learning tools that can mitigate the affects of poor quality images and have resulted in some innovative solutions for a diverse range of applications. From Johansson's [105] novel small vessel detection study, to the UAV flood mapping system proposed by Feng et al. [143, 144].

Part II

RESEARCH METHODS

4 FIELD COLLECTION METHODS

Synopsis

Monitoring sites were selected using previous records in conjunction with single surveys. When locations were identified a standard programme was initiated. A range of static and dynamic image data were collected. The main body of this research focuses on a method to monitor broad populations of native bees using images of their active nests. This chapter describes the study locations, the methods used to determine the start and end of monitoring seasons, the types of environmental data collected and the tools used to collect data. This chapter also covers the techniques used to manually count active nests and methods used for collecting monitoring images.

4.1 MONITORING LOCATIONS

Several locations around the greater Whangarei district were identified as potential monitoring sites. Selection priority was given to areas that were easy to access. Study sites had to be reasonably safe and unlikely to be modified in the foreseeable future (e.g. by local government developments or by the general public).

Table 4.1: Monitoring locations and map co-ordinates.

Identifier	Name	Location
Site 1	Mt. Tiger	35°44′ 31.9" S, 174°25′ 18.8" E
Site 2	Mt. Parihaka	35°42′ 41.4" S, 174°20′ 19.3" E
Site 3	Memorial Drive	35°42′ 59.8" S, 174°20′ 25.4" E

Secondary priority was given to sites known to have supported large numbers of native bees in the past [29]. Communities with different biological structures were chosen. This was in order to collect a range of image data and test the validity of image-centric monitoring system. Two geographically separated locations were identified. Communities on Mt. Tiger and Mt. Parihaka were selected for repeat monitoring. In 2010, Memorial Drive, was included in the monitoring programme. The map coordinates are listed in Table 4.1 above, and shown in Figure 4.1 (pg. 47).

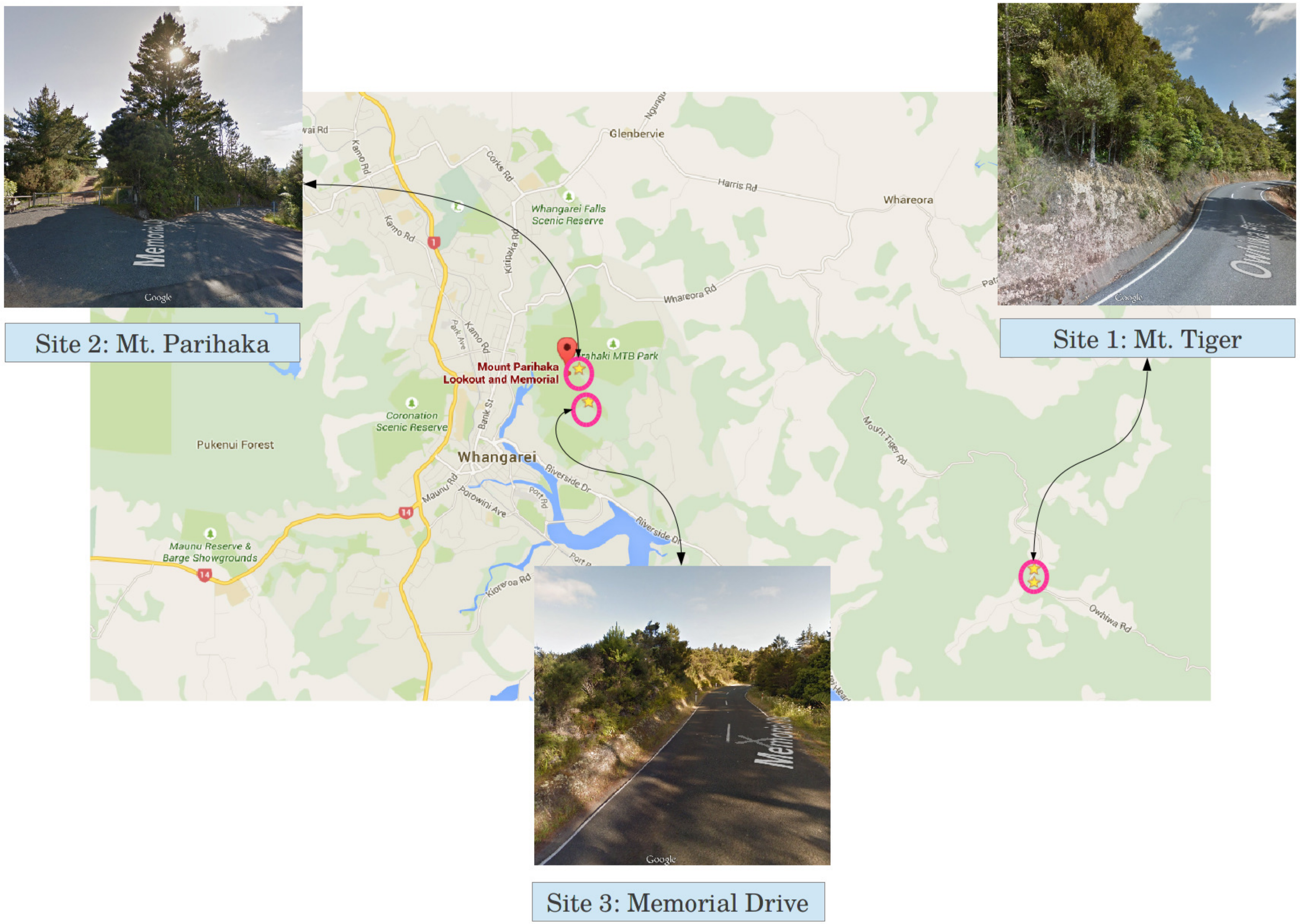


Figure 4.1: Google Map showing the site locations and photographs of each monitoring area.

4.1.1 *Species diversity*

Existing records were used to determine baseline data [29]. This included the known species composition and approximate population densities at locations selected for monitoring [29]. As shown in Figure 4.2 (pg. 49), six species were identified at these locations previously [29]. Five species from the Colletidae family as follows: *L.boltoni, L.huakiwi, L.imitatus, L.paahaumaa, L.pango* and a single species from the Halictidae family, *Lasioglossum sordidum*. Historically, the greatest numbers of bees were collected from Mt. Parihaka and surrounding areas; 826 individuals were collected between the years 2005–2006. The highest species diversity was also recorded at communities located around Mt. Parihaka; 522 bees were collected, representing six different species during the same period. Four species were identified at the communities located along Memorial Drive, and two species from the communities on Mt. Tiger.

4.1.2 *Description of monitoring sites*

SITE 1 Mt. Tiger is located around 20 minutes drive, east of Whangarei central. The monitoring site was located just opposite 510 Owhiwa Road. It encompassed a roadside bank around 15 meters long by 10 meters high; with a slope of around 60-80°. The bank consisted of exposed reddish clay soil. There were areas of sparse ground cover and areas of dense vegetation. It was bordered by some native shrubs (e.g. maunuka and kanuka), as well as introduced plants (e.g. ox-eye daisy). The main bank area backed onto agricultural land, which appeared to be mainly used for dry live-stock. There were some larger areas of natural bush within a few meters of the nest site. These were located across the main road but almost exactly opposite to the majority of active nests.

SITE 2 Mt. Parihaka[1] is a historical pā. It is located a few minutes drive from Whangarei central. The monitoring site was situated just through a gated forestry area. It encompassed a small bank, around 5 meters long by 8 meters high; with a slope of around 60-80°. The bank consisted mainly of white clay soils. It was covered in weedy, shrubby vegetation. The area is managed as pine plantation but much of the land has already been logged. It is currently regenerating back into natural bush. Gorse and different types of shrubs, such as manuka and kanuka, dominated the area. The location had been actively managed in the past (e.g. the Whangarei District Council regularly tended the area by periodically spraying with herbicides)

[1] In the past the summit was frequently *but incorrectly* called Mt. Parahaki. The original Māori spelling and proper name of *Parihaka* was reinstated in 2005. Refer to http://www.beehive.govt.nz/node/23727.

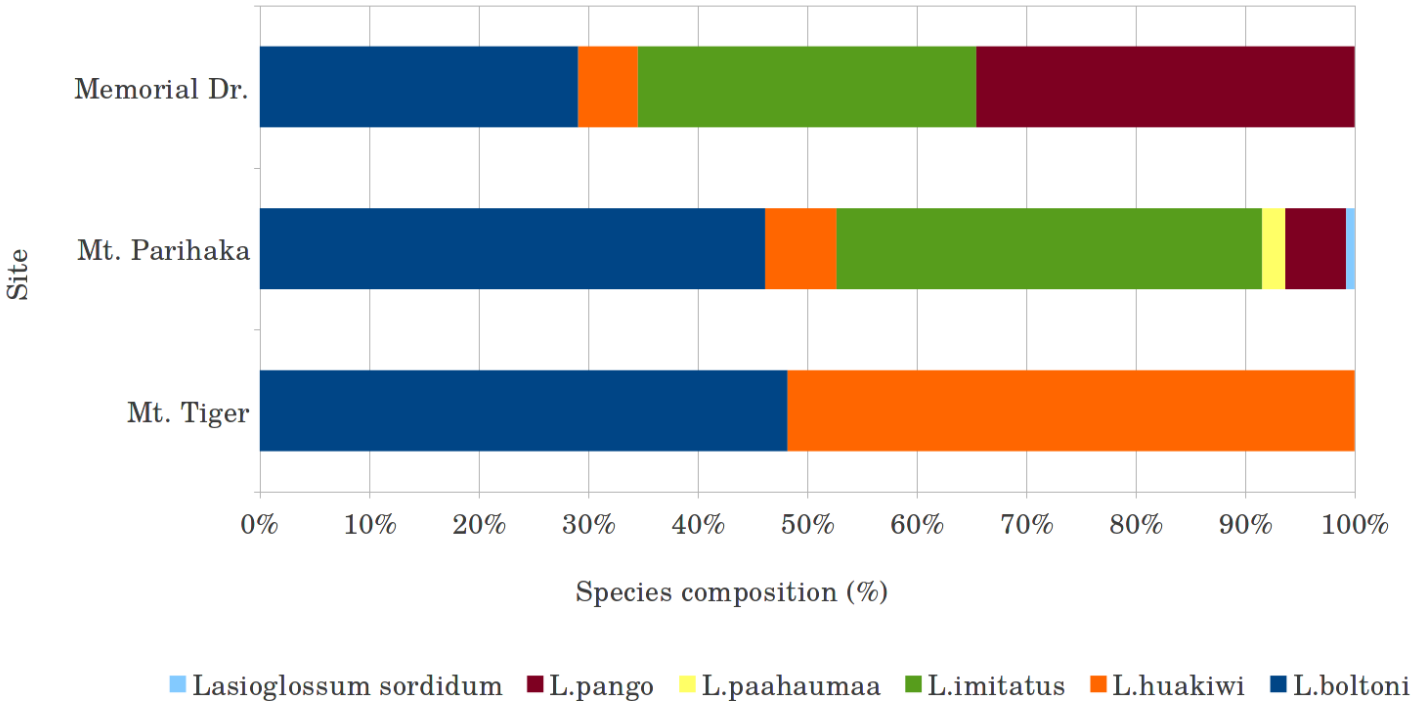


Figure 4.2: Site-species composition data from Hart [21, 29].

Table 4.2: Weather measurements and general observations.

1	Observations:
	a) Cloud-cover as a percentage of total cover (%).
	b) Flight/foraging: L-low, M-medium, H-high.
	c) Nest construction activity: L-low, M-medium, H-high.
2	Ambient temperature (°C).
3	Relative humidity (%).
4	Wind-speed (meters/sec).

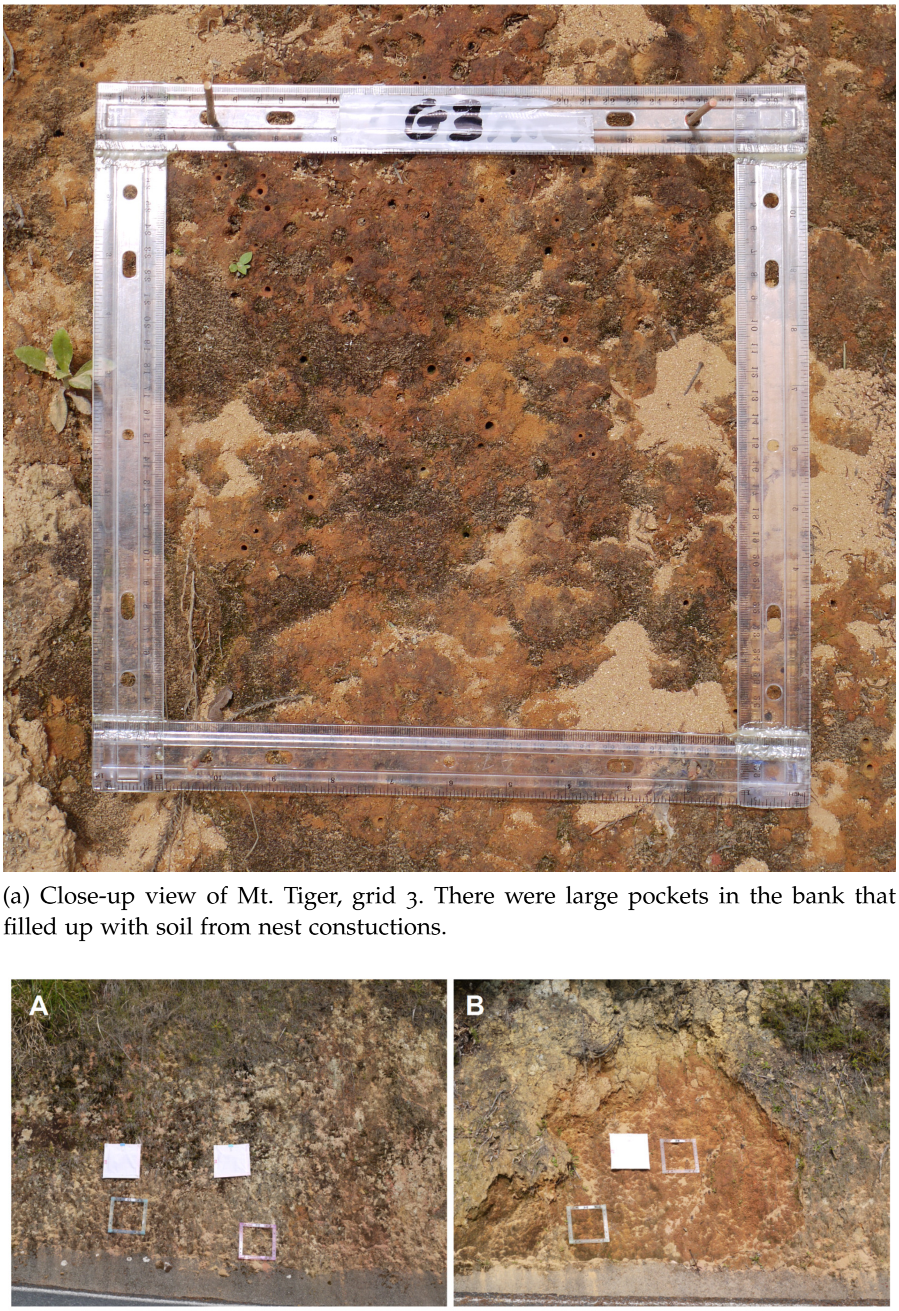
but there was no evidence of this during monitoring. The area where active nests were monitored, became overgrown with gorse, young pine and other shrubs. In 2014 the active nest areas were cleared to remove gorse seedlings before data could be collected.

SITE 3 Memorial Drive monitoring site was located about half way down from the summit of Mt. Parihaka. The nest site spanned a road-side bank approximately 5 meters long by 2 meters high; with a slope of around 60-80°. The bank consisted of white clay soils covered in weedy vegetation. The bank was bounded by the Parihaka reserve on either side. The reserve areas were covered in a variety of introduced plants such as ox-eye daisy, wild carrot and grasses.

4.2 FIELD MONITORING METHODS

Surveys were conducted weekly in the months of September and October (2009–2014). When signs of nest construction or native bees in flight were observed, monitoring was initiated. For examples of nest sites, see Figures 4.3 – 4.4 below. Sampling continued each fine day, until bees were no longer active. Daily monitoring and seasonal surveys were conducted by site order starting with site 1: Mt. Tiger, site 2: Mt. Parihaka and finishing with site 3: Memorial Drive. Monitoring started at Mt. Tiger between the hours of 0800–1000, followed by Mt. Parihaka between 1000–1200 hours. Sampling finished at Memorial Drive between 1100–1300 hours.

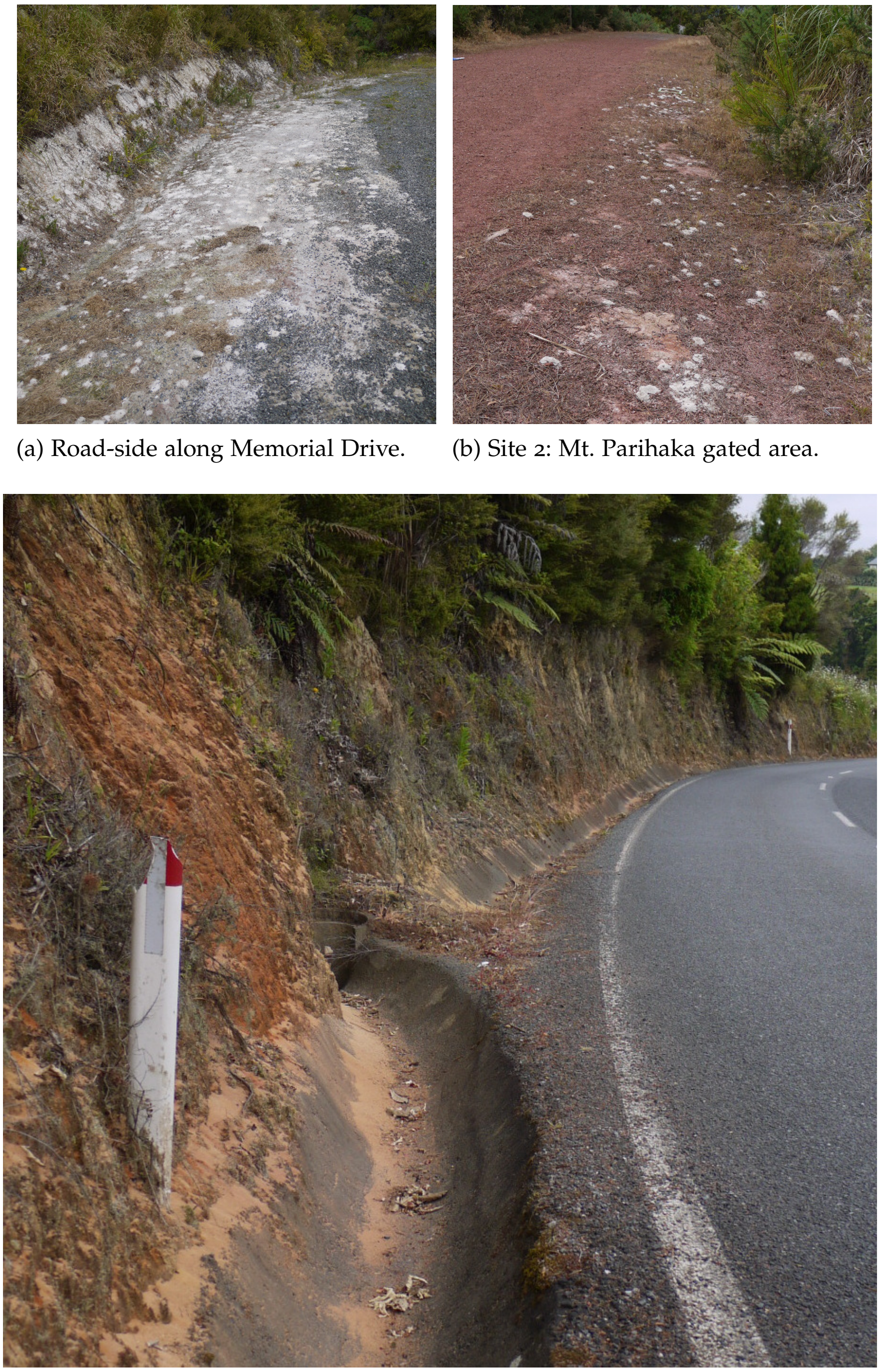
Standard weather and observational data were collected. These are shown in Table 4.2. At each monitoring location, local micro-weather measurements were collected using a Kestrel-1000 Wind Meter. Three measurements were taken. The average results were recorded. Subjective quantities were recorded. This included the percentage cloud-cover and estimations of the flight-foraging and nesting-activity of native bees.



(a) Close-up view of Mt. Tiger, grid 3. There were large pockets in the bank that filled up with soil from nest constuctions.

(b) Wide-views of Mt. Tiger: A) grids 1-2 and B) grids 3-4. The white grids were used to collect video data.

Figure 4.3: Mt. Tiger monitoring images showing (a) the set-up for grid 3 (b) the overall structure of the road-side bank.



(a) Road-side along Memorial Drive.

(b) Site 2: Mt. Parihaka gated area.

(c) Site 1: Tumulii in the road-side drains along Mt. Tiger

Figure 4.4: Signs marking the beginning of the active season. Mounds of white (or reddish) clay soil started to accumulate around horizontal ground nest entrances. This was visible from a moving car along (a) Memorial Drive and (b) Mt. Parihaka. The excess reddish clay soils from nest constructions could be seen accumulating in the roadside drains along (c) the Mt. Tiger site.

Some faunistic surveys were made in 2011 by using sweep net collections of insects in flight around nesting communities. The bees (and related insects) were collected while they were in flight over active nests, using five sweeps. They were immediately transferred to a killing jar of ethyl acetate until they expired. They were placed into containers. Specimen jars were labelled with date-time, site and collection details. Collections were processed immediately after field monitoring was completed. They were placed on standard graph paper and photographed under natural indoor lighting conditions. For an example image, refer to Figure 7.2 (a), (Chapter 7, pg. 116). Specimens were re-packaged and couriered to Dr. B. J. Donovan[2] for taxonomic identification; and inclusion in national entomological records.

In 2009, Northland was affected by drought. Surveys were conducted weekly, but native bees did not emerge as expected. Most sites were only slightly active in late December. Some bees emerged in early January. Within a week, most had expired. For these reasons, little monitoring data were collected during this time.

4.2.1 *Manual nest counts and nest image collections*

Images of active nests were collected at every site, in the same manner, each year. A standard of measure was constructed using four plastic rulers, each 300 mm long. They were fixed with glue at each end to form a grid. Four separate grids were made. When constructed, the internal dimension of each grid was 245 x 245 mm (0.06 m^2). When monitoring began, four grid locations were chosen at each site.

Locations were selected so exactly the same nests, at the same locations, could be monitored across seasons. Grids were placed over nests areas that were easy to access and observe. Locations were selected in areas that were unlikely to be disturbed. Grids were spaced out to accommodate the distribution of active nests; and within the practical boundaries of the entire nest aggregation. When the exact locations for the grids were chosen, they were fixed into place with fine wooden skewers. Grids were aligned and removed by using existing holes through the centre of each ruler, through which four skewers were staked. The set-up is shown in Figure 4.3 (pg. 51). Each 15 mm skewer was pushed into the ground, to a depth of around 8 mm. This was so they would not be easily dislodged. Skewers were nearly undetectable. Therefore, florescent nail polish was used to mark the tips.

The four grids were set up at the start of each monitoring session. They were hooked onto the skewers. Images were collected using an off-the-shelf DSLR camera (Panasonic G1 Lumix / 35 mm). The camera was set to *active sports mode* and a *single shot automatic focus*. The viewfinder was set to *show a guide*. Each nest-grid image was

[2] Donovan Scientific Insect Research, Private Bag 4704, Christchurch, New Zealand.

approximately centred on, and acquired at right angles to the grid. When possible the distance between the nest-grid and camera was at least 2.5 meters. The zoom was used to help keep the grid dimensions proportionally square. Around ten images of each grid were taken in sequential order. Photographs were saved in a high quality JPEG format. Each image was 2,816 x 2,112 pixels, with a resolution of 180 dots per inch (DPI) and a bit depth of 24.

When image collections were completed, the number of active nests within each grid were counted. When possible manual nest counts were collected by two observers. The number of active nests in each grid was counted three times. The average number of nests per grid was recorded for each observer. If nest entrances were obscured by debris, the number of active nests were roughly estimated.

At the end of each monitoring day, nest-image collections were immediately transferred from the camera's SD memory onto an external hard-drive. A copy of the monitoring images were backed up to a second external hard-drive. The camera batteries were recharged. The SD memory was cleared. This procedure was repeated until the active flight season was completed. At the start of each new monitoring year, the camera date and time stamp settings were checked. The image file numbering was reset to zero.

4.2.2 *Control and inactive nest images*

A clay bank, located on Mt. Tiger, was cleared from vegetation to simulate active nests. The plastic grid was set-up. Sixteen holes were bored into the grid-area of the bank. Each hole was the approximate size of a nest entrance. Some holes included small disturbances in soil; this was to mimic the mounds of dirt typically observed around the active nests of native bees. Photographs of the artificial nest were collected. They were processed alongside monitoring images. In May 2013 inactive nest images were gathered from each site-grid. These were processed alongside monitoring images. Refer to Chapter 6 for images of training stacks, shown in Figures 6.7–6.9 (pg. 94–96).

4.3 SUMMARY OF FIELD METHODS

Surveys were conducted over six years (2009–2014). Monitoring data were collected across five years (2010–2014). Data were collected on each fine day, each year until there was a clear indication that the bees were no longer active. The active flight season was regarded as complete when bees were no longer constructing nests, were not observed foraging, or were not seen in flight around their nests.

Manual counting methods have been detailed in this chapter. As each season progressed, the active nest entrances were increasingly more difficult to identify. Consequently the number of active nests measured on the first few monitoring days were the most important and the most reliable.

Image acquisition techniques have been described. The monitoring equipment was readily acquired, low budget, off-the-shelf or easily constructed. The monitoring method was designed so the images of active nests could be collected at approximately the same time, at each location, every monitoring day. The method was consistently used each season. The image capture area was broadly estimated by using a square grid standard of measure (245 x 245 mm). Images of an artificially constructed nest and of inactive nests were acquired. This chapter completes the data collection methods. In the next chapter, image data handling, processing and analysis techniques are outlined.

5 IMAGING METHODS

Synopsis

Image data handling, database management and image analysis procedures are outlined in this chapter. This includes the techniques used to sort and pre-process images of active nests. Biomedical imaging package FIJI and the interactive image segmentation methods developed within the TWS workbench are detailed. The RF machine learner was used to construct nest-image classifiers. Therefore, the methods developed to tune and optimise the models using TWS and WEKA are outlined. The final sections in this chapter are dedicated to verification methods; including the statistical tests used in comparative analyses of methods.

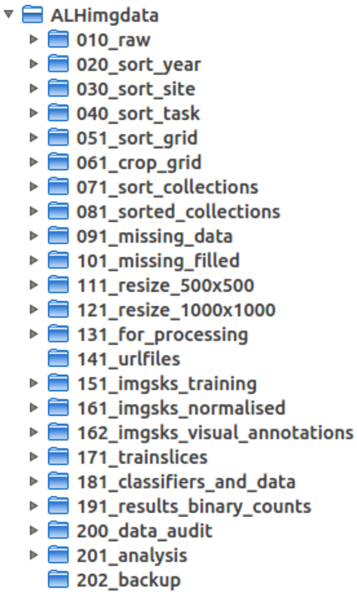
5.1 DATA MANAGEMENT

In 2013 monitoring image data were corrupted. Images were recovered using forensic tools; TestDisk[1] and PhotoRec[2]. When the recovery process was finished, bash scripts were used to restore images. For code examples, refer to Appendix E, Listing E.1 (pg.167). A folder structure was created for monitoring image data, as shown in Figure 5.1 below. The database was copied onto separate hard-drives using Grsync[3]. The workflow operations for each folder were documented. Records were saved in readme files which were attached to base folders for reference. The folder names were chosen to be as short as possible, while remaining descriptive enough to follow. Batch processing techniques were applied to a copy of a working folder, before they were used on final folders.

5.2 COMPUTING ENVIRONMENT

The system specifications are listed below in Table 5.1. A Linux operating system was used. It was configured for the specific image processing and classification tasks. Ubuntu (14.04) swap memory was turned off during image processing tasks. This was to prevent processes from swapping out of physical memory. Virtual memory[4] options were passed to the Java Virtual Machine from FIJIs main configu-

[1] *TestDisk*- http://www.cgsecurity.org/wiki/TestDisk
[2] *PhotoRec*-http://www.cgsecurity.org/wiki/PhotoRec
[3] *Grsync*-http://www.opbyte.it/grsync
[4] *Java*- http://docs.oracle.com/cd/E13222_01/wls/docs81/perform/JVMTuning.html



```
ALHimgdata
    010_raw
    020_sort_year
    030_sort_site
    040_sort_task
    051_sort_grid
    061_crop_grid
    071_sort_collections
    081_sorted_collections
    091_missing_data
    101_missing_filled
    111_resize_500x500
    121_resize_1000x1000
    131_for_processing
    141_urlfiles
    151_imgsks_training
    161_imgsks_normalised
    162_imgsks_visual_annotations
    171_trainslices
    181_classifiers_and_data
    191_results_binary_counts
    200_data_audit
    201_analysis
    202_backup
```

Figure 5.1: Folder structure created for the monitoring image database.

Table 5.1: Computer hardware/software and operating system specifications.

Operating system	Ubuntu 14.04
Linux kernel	3.13.0-48-generic
Computer	CQ1-1240IN Personal computer
Monitor	46.99 cm Liquid crystal display
Ram	7.2 GB
Processor	2 x AMD E-350
Graphics card	Gallium 0.4 on AMD PALM
External hard-drive	2 x 2 TB Seagate
Java	java.runtime.version 1.6.0_24-b07
WEKA	Version 3.7.11
FIJI/ImageJ	1.49u

ration file (see Listing E.2, pg. 167). This was to increase the memory heap size.

5.3 IMAGE PREPARATION

More than one category of images were collected during monitoring For examples, refer to Chapter 7, Figure 7.2 (pg.116). These included photographs of a) bees in flight or foraging, b) active nests with and without a standard grid measure, c) active nests acquired with a Smart Phone, d) a range of monitoring videos. Raw data were therefore sorted by year, site and image monitoring-*task*. Images of active nests, using a grid as a standard of measure, were used in the analysis outlined in this thesis. Nest images were sorted by year, site, and grid number as shown in Figure 5.2 (a) below. Images were renamed using XnView[5]. As shown in Figure 5.2 (b), image EXIF metadata was combined with the sub-folder grid identifier and used to create unique file names.

5.3.1 *Overview of file and folder procedures*

The following paragraphs provide a general overview of practical procedures used throughout the imaging pipeline. The important processing tasks will be discussed in greater detail in the proceeding sections.

FOLDER 061_CROP_GRID: Image data were manually cropped to grids using XnView.

FOLDERS 081_SORTED_COLLECTIONS: Image data were sorted into separate image collections.

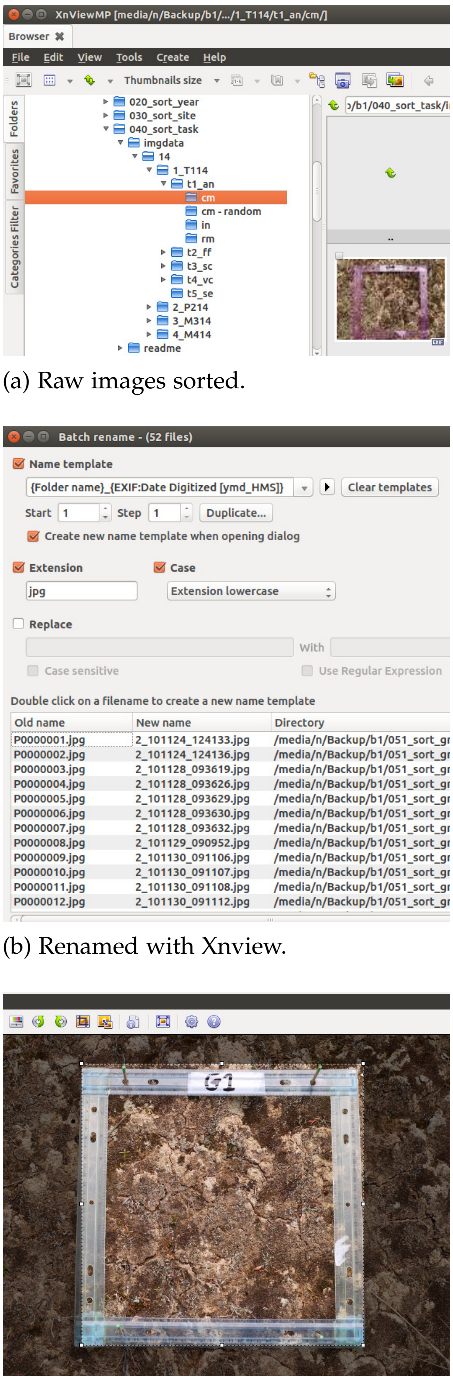
FOLDER 101_MISSING_FILLED: Filler image data were added to folders to retain the sequential order of image collection monitoring days.

FOLDER 111_RESIZE_500X500: Images were resized to upload to the GitHub database.

FOLDER 121_RESIZE_1000X1000: Images were resized for visual annotation and checking.

FOLDER 131_FOR_PROCESSING: Image data for pre-processing were stored in this folder.

[5] *XnView*-http://www.xnview.com/en/



(a) Raw images sorted.

Batch rename - (52 files)

☑ Name template

{Folder name}_{EXIF:Date Digitized [ymd_HMS]}

Clear templates

Start 1 Step 1 Duplicate...

☑ Create new name template when opening dialog

☑ Extension: jpg

☑ Case: Extension lowercase

□ Replace

With

□ Case sensitive

□ Use Regular Expression

Double click on a filename to create a new name template

Old name	New name	Directory
P0000001.jpg	2_101124_124133.jpg	/media/n/Backup/b1/051_sort_gr
P0000002.jpg	2_101124_124136.jpg	/media/n/Backup/b1/051_sort_gr
P0000003.jpg	2_101128_093619.jpg	/media/n/Backup/b1/051_sort_gr
P0000004.jpg	2_101128_093626.jpg	/media/n/Backup/b1/051_sort_gr
P0000005.jpg	2_101128_093629.jpg	/media/n/Backup/b1/051_sort_gr
P0000006.jpg	2_101128_093630.jpg	/media/n/Backup/b1/051_sort_gr
P0000007.jpg	2_101128_093632.jpg	/media/n/Backup/b1/051_sort_gr
P0000008.jpg	2_101129_090952.jpg	/media/n/Backup/b1/051_sort_gr
P0000009.jpg	2_101130_091106.jpg	/media/n/Backup/b1/051_sort_gr
P0000010.jpg	2_101130_091107.jpg	/media/n/Backup/b1/051_sort_gr
P0000011.jpg	2_101130_091108.jpg	/media/n/Backup/b1/051_sort_gr
P0000012.jpg	2_101130_091112.jpg	/media/n/Backup/b1/051_sort_gr

(b) Renamed with Xnview.

(c) Nest images cropped.

Figure 5.2: Preparation of monitoring data. Nest images (a) acquired using a constructed standard of measure are used in this analysis. Active nest images were (b) sorted to grids, renamed and (c) cropped to the outside grid using XnView.

FOLDER 141_URLFILES: Image data were renamed for uploading to online databases.

FOLDER 151_IMGSKS_TRAINING: Images were collated by FIJI macro into image stacks for processing and saved as TIFF files. ImageJ macro files (.ijm) and image stacks (.tiff) were stored in this folder.

FOLDER 161_IMGSKS_NORMALISED: Stacks were enhanced with 0.4% saturated pixels.

FOLDER 162_IMGSKS_VISUAL_ANNOTATIONS: Image data used for visual annotations were saved in this folder, including image overlays.

FOLDER 171_TRAINSLICES: Training stacks were stored in this folder, including the *control image* and the *inactive nest* images for respective grids.

FOLDER 181_CLASSIFIERS_DATA: Training stacks and output files were stored in this folder. This included final annotation data files (.arff), WEKA classifier model files (.model), training log files (.csv and .txt) and final classified images (.tiff). The folder was structured per site and grid. With subfolders for each grid (G1–G4) within each of the site folders M3, P2 and T1. These processes are explained further in Section 5.6 (pg. 71).

FOLDER 191_RESULTS_BINARY_COUNTS: Post-processing image data (.tiff), macro files (.ijm) and results (.txt and .csv) were stored in this folder. Post processing operations included: cropping images, applying morphological operators and counting binary image objects based on their size and shape. These processes are explained further in Section 5.7 (pg.73).

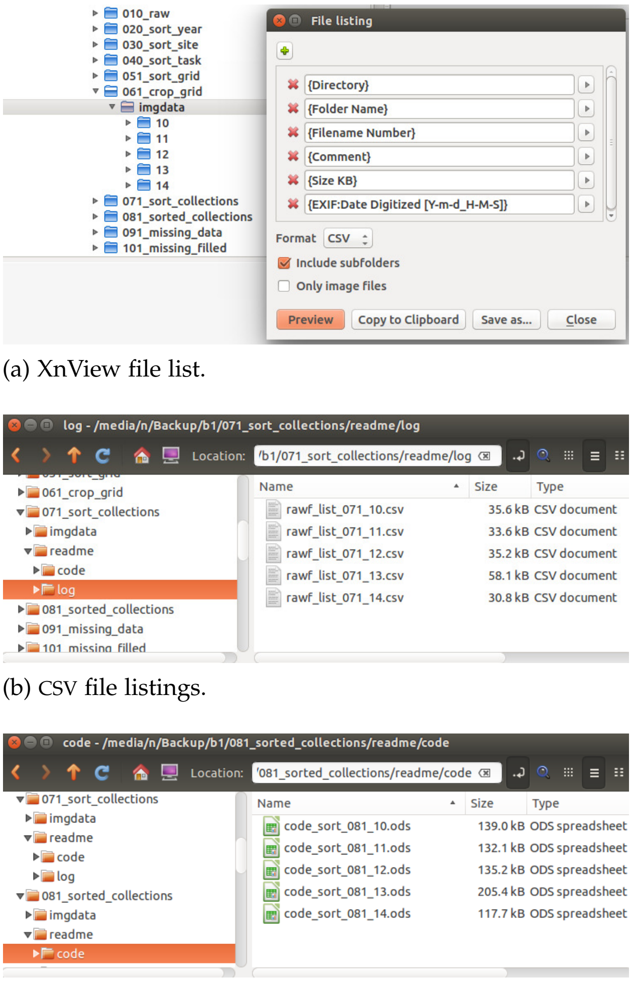
5.3.2 *Sorting image collections*

The images of active nests were manually cropped to the outside grid edge using XnView. This is shown in Figure 5.2 (c). Images were sorted into separate *collections*; these were sequences of the same nest, separated by minutes-seconds. Collections were sorted into four folders; C1–C3 were used in analysis and C4 contained any other extra images. The sorting process was semi-automated.

XnView catalogue feature, Create > File listing, was applied to build a list.csv file. An example of this is shown in Figure 5.3 (a) (pg. 61). The list.csv file was imported into Apache OpenOffice[6] spreadsheets as outlined in Figure 5.3 (b) (pg. 61). Image data were sorted

[6] *Apache OpenOffice*-https://www.openoffice.org/

into four separate collections based on date-time attributes. The final sorted list.csv was written into a bash script. The script automatically copied images from the unsorted folder (/071 sort_collections) to sorted folders (/081 sorted_ collections). Metadata information was retained during the copying process using ExifTool[7]. When the process was completed, the file numbers were reviewed via XnView (e.g. Create > File listing). The file listings were checked to ensure the correct number of files were copied.



- 010_raw
- 020_sort_year
- 030_sort_site
- 040_sort_task
- 051_sort_grid
- 061_crop_grid
  - imgdata
    - 10
    - 11
    - 12
    - 13
    - 14
- 071_sort_collections
- 081_sorted_collections
- 091_missing_data
- 101_missing_filled

File listing

- {Directory}
- {Folder Name}
- {Filename Number}
- {Comment}
- {Size KB}
- {EXIF:Date Digitized [Y-m-d_H-M-S]}

Format: CSV

☑ Include subfolders

☐ Only image files

(a) XnView file list.

log - /media/n/Backup/b1/071_sort_collections/readme/log

Location: /b1/071_sort_collections/readme/log

- 061_crop_grid
- 071_sort_collections
  - imgdata
  - readme
    - code
    - log
- 081_sorted_collections
- 091_missing_data
- 101_missing_filled

Name	Size	Type
rawf_list_071_10.csv	35.6 kB	CSV document
rawf_list_071_11.csv	33.6 kB	CSV document
rawf_list_071_12.csv	35.2 kB	CSV document
rawf_list_071_13.csv	58.1 kB	CSV document
rawf_list_071_14.csv	30.8 kB	CSV document

(b) CSV file listings.

code - /media/n/Backup/b1/081_sorted_collections/readme/code

Location: '081_sorted_collections/readme/code

- 071_sort_collections
  - imgdata
  - readme
    - code
    - log
- 081_sorted_collections
  - imgdata
  - readme
    - code

Name	Size	Type
code_sort_081_10.ods	139.0 kB	ODS spreadsheet
code_sort_081_11.ods	132.1 kB	ODS spreadsheet
code_sort_081_12.ods	135.2 kB	ODS spreadsheet
code_sort_081_13.ods	205.4 kB	ODS spreadsheet
code_sort_081_14.ods	117.7 kB	ODS spreadsheet

(c) Sorting image collections.

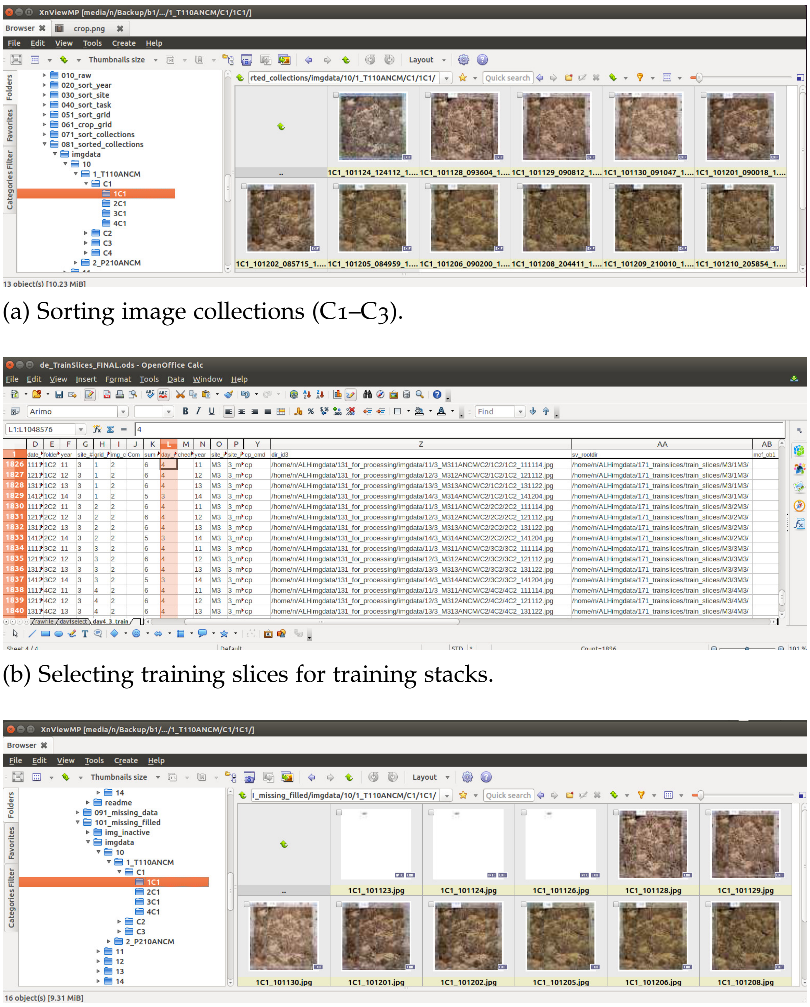
Figure 5.3: Image data were organised and sorted using (a) XnView file listing features (b) CSV file lists and (c) spreadsheet sorting tools.

The sorting process was repeated when training images were selected. This is shown in Figure 5.4 (a) below. The list.csv was imported into spreadsheets. The sheets were used to select the third or fourth monitoring day for each year, site and grid. The final list.csv was written into a bash script.

[7] *ExifTool*-http://www.sno.phy.queensu.ca/ phil/exiftool

The script automatically copied images from the processing folder (/131_for_ processing) to the training stacks folder (/151_imgsks_ training). A sample is shown in Figure 5.4 (b).

Filler data (ND) were created for the image database. An example is shown in Figure 5.4 (c). A white image was labelled with ND and copied. The EXIF date-time, and name of each filler image was changed to match the monitoring collections. Filler images were selected, copied and pasted into the folder /101_missing_ filled using Ubuntu *merge* function. Existing images were not overwritten. Thus, only data representing *missing* files were copied.



(a) Sorting image collections (C1–C3).

de_TrainSlices_FINAL.ods - OpenOffice Calc

File Edit View Insert Format Tools Data Window Help

Arimo

L1:L1048576 = 4

	D	E	F	G	H	I	J	K	L	M	N	O	P	Y	Z	AA	AB
1	date_	folder	year	site_#	grid_	img_c	Com	sum	day_	chec	year	site_	site_	cp_cmd	dir_id3	sv_rootdir	mcf_ob1
1826	1111	1C2	11	3	1	2		6	4		11	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/11/3_M311ANCM/C2/1C2/1C2_111114.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/1M3/	
1827	1211	1C2	12	3	1	2		6	4		12	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/12/3_M312ANCM/C2/1C2/1C2_121112.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/1M3/	
1828	1311	1C2	13	3	1	2		6	4		13	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/13/3_M313ANCM/C2/1C2/1C2_131122.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/1M3/	
1829	1412	1C2	14	3	1	2		5	3		14	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/14/3_M314ANCM/C2/1C2/1C2_141204.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/1M3/	
1830	1111	2C2	11	3	2	2		6	4		11	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/11/3_M311ANCM/C2/2C2/2C2_111114.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/2M3/	
1831	1211	2C2	12	3	2	2		6	4		12	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/12/3_M312ANCM/C2/2C2/2C2_121112.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/2M3/	
1832	1311	2C2	13	3	2	2		6	4		13	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/13/3_M313ANCM/C2/2C2/2C2_131122.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/2M3/	
1833	1412	2C2	14	3	2	2		5	3		14	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/14/3_M314ANCM/C2/2C2/2C2_141204.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/2M3/	
1834	1111	3C2	11	3	3	2		6	4		11	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/11/3_M311ANCM/C2/3C2/3C2_111114.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/3M3/	
1835	1211	3C2	12	3	3	2		6	4		12	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/12/3_M312ANCM/C2/3C2/3C2_121112.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/3M3/	
1836	1311	3C2	13	3	3	2		6	4		13	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/13/3_M313ANCM/C2/3C2/3C2_131122.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/3M3/	
1837	1412	3C2	14	3	3	2		5	3		14	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/14/3_M314ANCM/C2/3C2/3C2_141204.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/3M3/	
1838	1111	4C2	11	3	4	2		6	4		11	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/11/3_M311ANCM/C2/4C2/4C2_111114.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/4M3/	
1839	1211	4C2	12	3	4	2		6	4		12	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/12/3_M312ANCM/C2/4C2/4C2_121112.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/4M3/	
1840	1311	4C2	13	3	4	2		6	4		13	M3	3_m	cp	/home/n/ALHimgdata/131_for_processing/imgdata/13/3_M313ANCM/C2/4C2/4C2_131122.jpg	/home/n/ALHimgdata/171_trainslices/train_slices/M3/4M3/	

(b) Selecting training slices for training stacks.

(c) Filler image files added to monitoring data.

Figure 5.4: File methods for (a) sorting image collections, (b) selecting training slices, and (c) filler images (ND) for missing data.

5.3.3 *Pre-processing procedures*

Images were resized to 500 x 500 pixels using Bicubic interpolation. They were collated into time based stacks. The stacks were saved in TIFF format; for code, refer to Listing E.4 (pg. 168). Saturated pixels were set to 0.4%; each slice was enhanced.

5.4 TRAINABLE SEGMENTATION INTERFACE

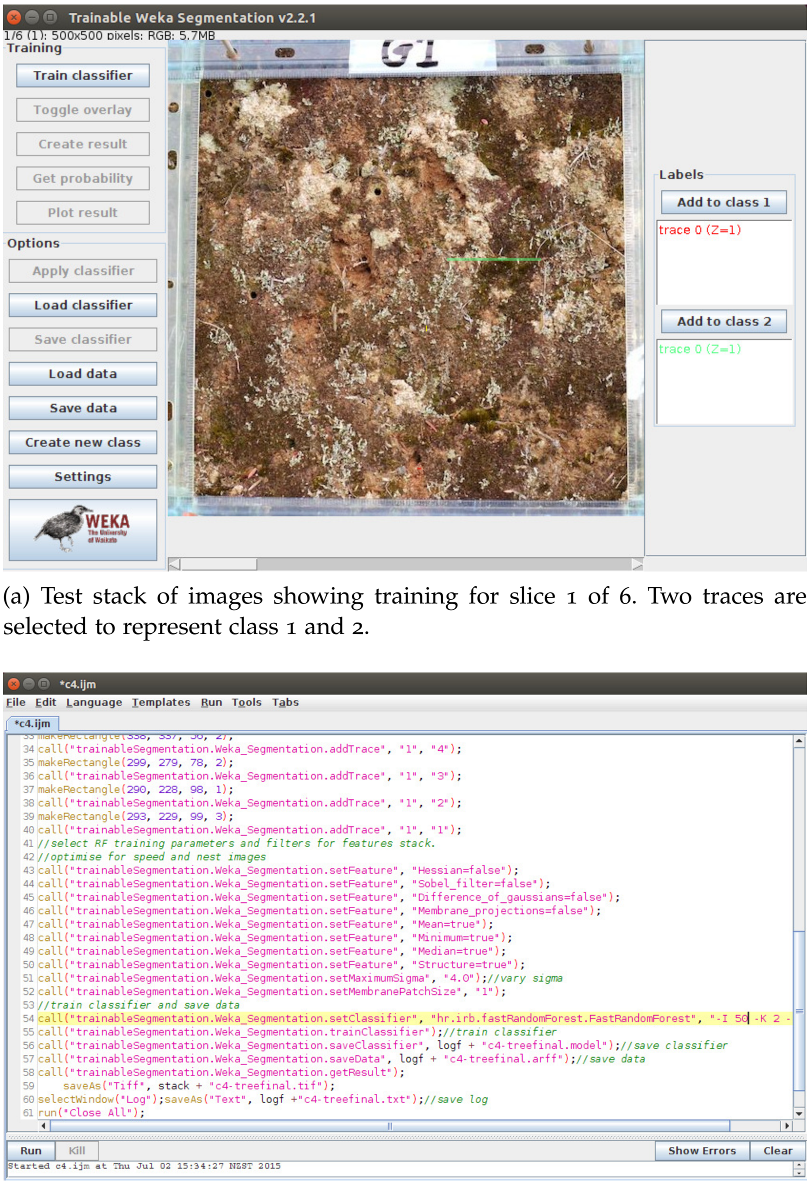
Trainable Weka Segmentation (TWS) user interface and options are shown in the Figures 5.5, 5.6 and 5.7 below. They are referred to frequently in the proceeding sections. The main user interface and script editor is shown in Figures 5.5 (a)–(b). Feature settings and options for classifiers are shown in Figure 5.6. Finally WEKA model options and model information interfaces are shown in Figures 5.7 (a)–(b) (pg. 65).

5.4.1 *Filter descriptions and setting options*

The filters listed below are used in classifier tuning tests. They were selected from a possible twenty. They were chosen to highlight the textural information in images of active nests. A brief description of each is included here for reference.

1. *Mean, Variance, Median, Minimum, Maximum*: The pixels within a radius of $1, 2, 4 ... 2^n$ pixels from a target pixel location, are subjected to the mean, variance, median, minimum and maximum operation. The target pixel is then set to that value. The radius is the neighbourhood area; for example a 3 x 3 matrix. The median filter reduces the noise in an active image by replacing each pixel with the neighbourhood median. The mean filter smooths the image by replacing each pixel with the neighbourhood mean. The minimum filter performs grey scale erosion by replacing each pixel with the smallest pixel value in the neighbourhood. The maximum filter performs grey scale dilation by replacing each pixel with the largest pixel value in the neighbourhood.

2. *Structure filter*: For all elements in the input image this filter calculates the eigenvalues (smallest and largest) of the structure tensor. This is also referred to as the second-moment matrix. The matrix is derived from the gradient of a function. It summarizes the predominant directions of the gradient in a specified neighbourhood of a point, and the degree to which those directions are coherent [180]. It uses a smoothing scale set by $\sigma = 1, 2, 4 ... 2^n$ and integration scales 1 and 3.



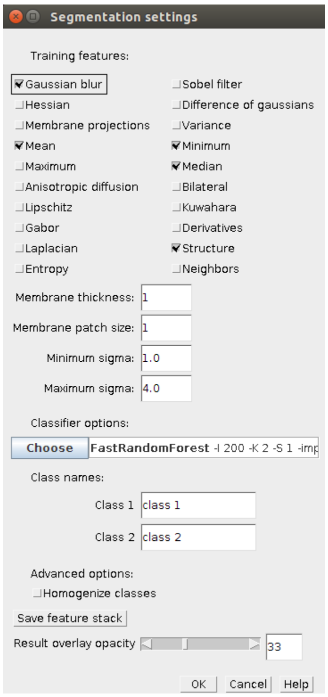
(a) Test stack of images showing training for slice 1 of 6. Two traces are selected to represent class 1 and 2.

```
call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "4");
makeRectangle(299, 279, 78, 2);
call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "3");
makeRectangle(290, 228, 98, 1);
call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "2");
makeRectangle(293, 229, 99, 3);
call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "1");
//select RF training parameters and filters for features stack.
//optimise for speed and nest images
call("trainableSegmentation.Weka_Segmentation.setFeature", "Hessian=false");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Sobel_filter=false");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Difference_of_gaussians=false");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Membrane_projections=false");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Mean=true");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Minimum=true");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Median=true");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Structure=true");
call("trainableSegmentation.Weka_Segmentation.setMaximumSigma", "4.0");//vary sigma
call("trainableSegmentation.Weka_Segmentation.setMembranePatchSize", "1");
//train classifier and save data
call("trainableSegmentation.Weka_Segmentation.setClassifier", "hr.irb.fastRandomForest.FastRandomForest", "-I 50 -K 2 -
call("trainableSegmentation.Weka_Segmentation.trainClassifier");//train classifier
call("trainableSegmentation.Weka_Segmentation.saveClassifier", logf + "c4-treefinal.model");//save classifier
call("trainableSegmentation.Weka_Segmentation.saveData", logf + "c4-treefinal.arff");//save data
call("trainableSegmentation.Weka_Segmentation.getResult");
   saveAs("Tiff", stack + "c4-treefinal.tif");
selectWindow("Log");saveAs("Text", logf +"c4-treefinal.txt");//save log
run("Close All");
```

```
Started c4.ijm at Thu Jul 02 15:34:27 NZST 2015
```

(b) FIJI script editor interface with the classifier test macro C4.ijm loaded.

Figure 5.5: TWS main (a) graphical user interface (b) macro script editor.



Training features:

☑ Gaussian blur	□ Sobel filter
□ Hessian	□ Difference of gaussians
□ Membrane projections	□ Variance
☑ Mean	☑ Minimum
□ Maximum	☑ Median
□ Anisotropic diffusion	□ Bilateral
□ Lipschitz	□ Kuwahara
□ Gabor	□ Derivatives
□ Laplacian	☑ Structure
□ Entropy	□ Neighbors

Membrane thickness: 1

Membrane patch size: 1

Minimum sigma: 1.0

Maximum sigma: 4.0

Classifier options:

Choose FastRandomForest -I 200 -K 2 -S 1 -imp

Class names:

Class 1 class 1

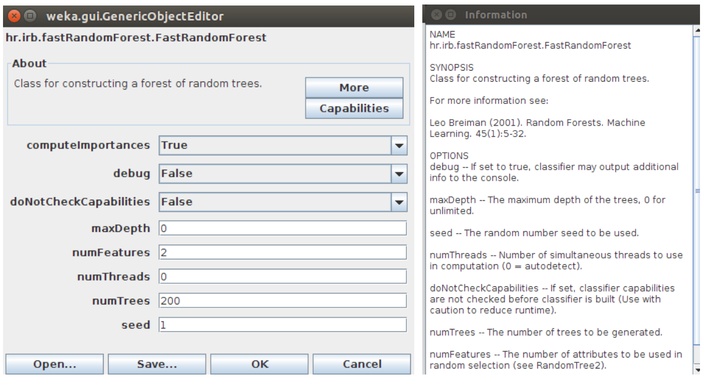
Class 2 class 2

Advanced options:

□ Homogenize classes

Result overlay opacity 33

Figure 5.6: TWS features settings interface.



hr.irb.fastRandomForest.FastRandomForest

About

Class for constructing a forest of random trees.

computeImportances True

debug False

doNotCheckCapabilities False

maxDepth 0

numFeatures 2

numThreads 0

numTrees 200

seed 1

NAME
hr.irb.fastRandomForest.FastRandomForest

SYNOPSIS
Class for constructing a forest of random trees.

For more information see:

Leo Breiman (2001). Random Forests. Machine Learning. 45(1):5-32.

OPTIONS
debug -- If set to true, classifier may output additional info to the console.

maxDepth -- The maximum depth of the trees, 0 for unlimited.

seed -- The random number seed to be used.

numThreads -- Number of simultaneous threads to use in computation (0 = autodetect).

doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is built (Use with caution to reduce runtime).

numTrees -- The number of trees to be generated.

numFeatures -- The number of attributes to be used in random selection (see RandomTree2).

(a) Model options.

(b) Model information.

Figure 5.7: TWS, (a) the model option settings and (b) information about the WEKA model.

3. *Sobel filter*: The gradient at each pixel are calculated after Gaussian blurs with $\sigma = 1, 2, 4...2^n$ are applied.

4. *Gaussian blur*: Operator performs n individual convolutions using Gaussian kernels with $\sigma = 1, 2, 4...2^n$. The larger the radius (σ) the more blurred the image becomes until the pixels are homogeneous. By default $n = 4$.

5. *Difference of gaussians*: calculates two Gaussian blur images from the original image and subtracts one from the other. The values for $\sigma = 1, 2, 4...2^n$.

6. *Anisotropic diffusion*: Filtering with 20 iterations, $1, 2, 4, ...2^n$ with the smoothing per iterations, $a_1 = 0.10, 0.35, a_2 = 0.9$ and an edge threshold set to the membrane size [181].

7. *Bilateral filter*: Preserves edges while averaging other parts of the image. It is similar to the mean filter [182]. It accomplishes blurring by only averaging the values around the current pixel that are close in colour value to the current pixel. The closeness of other neighbourhood pixels to the current pixels is determined by the specified threshold. For example with a value set to 10, each pixel that contributes to the current mean has to be within 10 values of the current pixel. In TWS this is a combination of spatial radii of 5, 10 and 20, with a range radius of 50 and 100.

8. *Lipschitz filter*: A Lipschitz cover of an image is equivalent to a grey scale opening by a cone. The cover can be applied for the elimination of a slowly varying image background by subtraction of the lower Lipschitz cover (a top-hat procedure) [183].

9. *Kuwahara filter*: Is a noise-reduction filter that preserves edges. In FIJI, the version of Kuwahara filter uses linear kernels rather than square. With a membrane patch size, as the kernel size and 30 angles and 0, 1 and 2.

There are four other filter options used in TWS. These are briefly described below. Only the values of sigma (σ) were adjusted for nest segmentations and classifier testing. The other settings relate to biological image analysis and were not used for nest image processing.

- *Membrane thickness*: Is the expected value of the membrane thickness; it is 1 pixel by default. This setting was not used for nest analysis and pertains to the membrane thickness of biological cells (e.g. brain image scans).

- *Membrane patch size*: Represents the size (n x n) of the field of view for the membrane projection filters. This setting was not used for nest analysis and pertains to the membrane thickness of biological cells.

- *Minimum sigma*: Is the minimum radius of the filters used to create the features. By default is 1 pixel.
- *Maximum sigma*: Is the maximum radius of the filters used to create the features. By default is 16 pixels.

5.5 CLASSIFIER TUNING

Six images of active nests were selected for classifier optimisation tests. Nest images were chosen to represent the full variation of monitoring data; as demonstrated in Figure 5.8 (pg. 67). A macro script was written to automate the testing. For an example script refer, to Figure 5.5 (b) (pg. 64). User-traces were saved as rep_nest.arff files. This was so the same feature vectors could be applied to all classifier models during evaluations. Small repetitive tests were applied to the stack of representative nest images. This was to examine the affects of filters on classifications and image segmentations. Tests were also designed to investigate classifier performance and the affects on segmentations when model parameters were adjusted. Tests are outlined in Table 5.2 below. They are detailed in the proceeding sections.

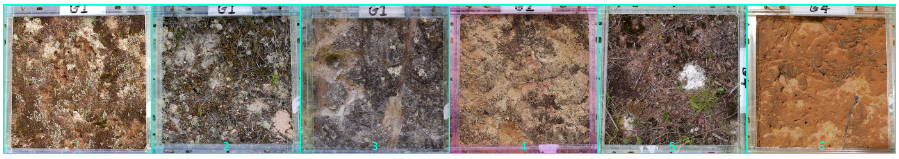


Figure 5.8: Stack of representative images (slices 1–6) were used to tune RF classifiers.

5.5.1 *Testing feature importances*

These tests were primarily conducted via FIJI, in the TWS workbench. To evaluate the contribution of each filter *compute importances* was selected in the TWS segmentation settings dialogue. This is shown previously in Figure 5.7 (a) (pg. 65).

The model was permitted to grow to the maximum depth during training (i.e. *max-depth* was not checked). For each test the TWS model output performance parameters including the: 1) feature importances, 2) time to create features stack and 3) the out of bag errors were saved as Test$_n$.csv files. Results were imported into spreadsheets for analysis.

During testing the filters that did not obviously contribute to the accuracy of final classifiers were removed. The training was re-run. The segmentation results were checked by comparing post-processed binary nest counts against raw RGB images. They were compared with manual nest counts taken in the field. This process was repeated until images from each representative slice were sufficiently segmented

Table 5.2: Tests on a range of RF classifier models.

Test	ID	F_n	N	M	σ_{max}	Comment
T1	C1	79	200	2	16	Results from TWS tests when default RF settings are used.
T2	C2	46	200	2	4	Results from TWS tests when filters are optimised for best segmentation of nest images
T3	C3	20	200	2	2	Results from TWS tests when filters are removed to reduce processing time.
T4	C4	20	10–1000	2	2	Results from WEKA tests used to optimise number of trees.
T5	C5	20	50	0–20	2	Results from WEKA tests used to optimise number of random features.

Table key

The number of features	F_n
The number of trees	N
The number of random features	M
The maximum value of sigma	σ_{max}

using the *very minimum* number of features possible. The following tests were performed:

1. Default features (test 1):
   - Classifier C1 was not tuned, the default parameters were used.
   - The filters selected were: Guassian blur, Sobel filter, Hessian, Difference of gaussians, and Membrane projections.
   - $F_n = 79$, $N = 200$, $M = 2$ and $\sigma_{max} = 16$.
2. Optimised features (test 2):
   - Classifier C2 was tuned for features to enhance textural information.
   - The filters selected were: Guassian blur, Mean, Minimum, Median, Anisotropic diffusion, Bilateral, Lipschitz, Kuwahara and Structures.
   - $F_n = 46$, $N = 200$, $M = 2$ and $\sigma_{max} = 4$.
3. Optimised features (test 3):
   - Classifier C3 was tuned to optimise the speed of feature stack creation, classifier training, construction and application.
   - The filters selected were: Gaussian blur, Mean, Minimum, Median and Structures.
   - $F_n = 20$, $N = 200$, $M = 2$ and $\sigma_{max} = 2$.

5.5.2 *Testing random forest parameters*

These tests were primarily conducted in WEKA Experimenter. The number of trees in a forest should be initially set to 200; the initial number of random features is the square root of the maximum number of features. In the feature optimised test (C3) there were 20 features that were important. Tests indicated they all contributed towards final classifications. Based on the tests, the ideal number of random features was five. RF model parameters were adjusted to incorporate these ranges. The rep_nest.arff dataset was used to evaluate the performances of models in WEKA Experimenter.

In test 4, twenty-two RF models were loaded into Algorithms for testing. The experiment was saved for repeat investigations as trees.exp. A 10 fold cross validation with a maximum of 10 iterations were selected for tests and used in the analysis. The number of trees were adjusted $N = 10 - 1000$ in each of the RF models; with $F_n = 20$, $M = 2$ and $\sigma_{max} = 2$. The analysis was run. The results from the experiment were saved in trees.arff and analysed in WEKA. The out of bag error and overall time to complete processing, was evaluated in

TWS. The output performance results were saved as tws_trees.csv for comparison.

In test 5, twenty RF models were added to Algorithms for testing. The experiment was saved for repeat investigations as random_features.exp. A 10 fold cross validation with a maximum of 10 iterations were selected for tests and used in the analysis. The number of random features were adjusted between $M = 0 - 20$ in each of the RF models; with $F_n = 20$, $N = 50$ and $\sigma_{max} = 2$. The analysis was run. The results from the experiment were saved as randomfeature.arff and analysed in WEKA. The out of bag error and overall time to complete processing was evaluated via TWS. The output performance results were saved as tws_random_feature.csv for comparison. Calibration experiments are summarised below:

1. Optimal number of trees (test 4_N):
   - Classifier C4, twenty-two RF models were loaded into WEKA experimenter algorithms.
   - The number of trees for each model was varied between $N = 10 - 1000$.
   - With the other RF parameters set at $F_n = 20$, $M = 2$ and $\sigma_{max} = 2$.
2. Optimal number of random features (test 5_M):
   - Classifier C5, twenty RF models were loaded into WEKA experimenter algorithms.
   - The number of random features for each model was varied between $M = 0 - 20$.
   - With the other RF parameters set at $F_n = 20$, $N = 50$ and $\sigma_{max} = 2$.

5.5.3 *Classifier benchmarks*

These tests were primarily conducted in WEKA Experimenter. The test classifier CF, was compared against other well known machine learners in final evaluations. Several RF classifiers were tested, including: the WEKA default model (M5), the FIJI default model (M7), a random feature optimised model (M6) and the final classifier CF model (M1). Six other common machine learners were tested, including: the zero rules model (M2), the J48 decision tree (M3), a random tree (M4), a Naive Bayes model (M8), a Voted Perception neural network (M9), and the SMO support vector model (M10).

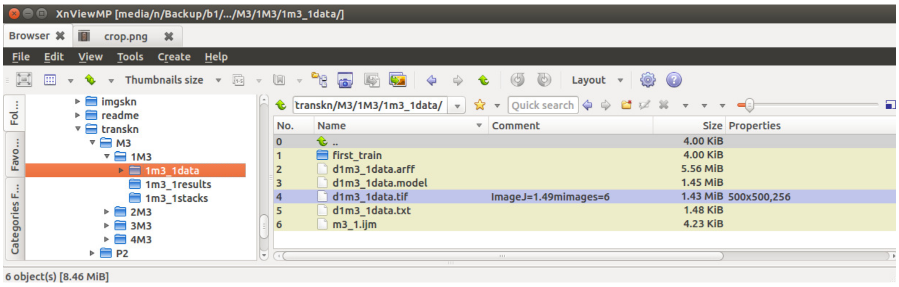
A 10 fold cross validation with a maximum of 10 iterations were selected for tests. Two RF models were tuned: M7) classifier CF ($N = 50$, $M = 2$) and M6) the random feature optimised classifier ($N = 50$, M = 8). All other classifiers were left at WEKA or FIJI default settings. The percentage of correctly classified instances were used for the analysis, using $n = 1000$ data results; with a confidence of 0.05 in a paired-corrected two tailed test. Classifier CF was used as the test base-classifier. The results from WEKA statistical analysis were saved.

5.5.4 *Segmentation performance of test classifier*

The representative image stack of six images, shown in Figure 5.8 (pg. 67), were fully processed using the final test classifier CF. A single trace was added and the model was trained again (CF2). The segmentation results were checked and another trace was added. The model was trained a final time (CF3). The out of bag error and binary image results were post-processed with morphological operators. The final counts were visually checked against raw RGB images and the manual-field counts for verification.

5.6 CLASSIFIER TRAINING

Classifiers were optimised for speed. This was to reduce the overall time required to process the monitoring images. There was a total of 1896 slices in the monitoring stack. Each slice was a 32-bit RGB image, 500 x 500 pixels in size. Classification processes, files and results were stored in Folders 181_classifiers_and_data. Training stacks and data (subfolder _transkn) and, monitoring image stacks and results (subfolder _imagkn) were stored in separate folders for processing. The folder structure provides an overview of the procedures used for training classifiers. This is outlined below, in Figure 5.9.



- imgskn
- readme
- transkn
  - M3
    - 1M3
      - 1m3_1data
      - 1m3_1results
      - 1m3_1stacks
    - 2M3
    - 3M3
    - 4M3
  - P2

transkn/M3/1M3/1m3_1data/

No.	Name	Comment	Size	Properties
0	..		4.00 KiB	
1	first_train		4.00 KiB	
2	d1m3_1data.arff		5.56 MiB	
3	d1m3_1data.model		1.45 MiB	
4	d1m3_1data.tif	ImageJ=1.49mimages=6	1.43 MiB	500x500,256
5	d1m3_1data.txt		1.48 KiB	
6	m3_1.ijm		4.23 KiB	

6 object(s) [8.46 MiB]

Figure 5.9: Classifier training and data folders.

Four training stacks were collated for each site. Each contained an image of the control grid (e.g. slice 2 for Mt. Tiger and Mt. Parihaka, and slice 1 for Memorial Drive) and an image of the inactive nest (acquired May 21, 2013). Slices were arranged in date sequence. Each site-grid classifier was trained separately using respective stacks. There were twelve stacks in total, four per site.

The first grid was used to train initial classifiers for each site. A *minimum* number of pixels were selected for each class label, starting with the control grid, and inactive grid slices. The segmentation results were checked and when they were satisfactory the grid G_n.arff results were saved. The grid G_n.arff files were loaded into the second grid classifier. Training was run *before* any new annotations were added. Results were checked and only where minimally necessary, new class traces were assigned to correct for segmentation inaccuracies. A new classifier was trained based on added class traces, saved and loaded into the next grid-batch classifier training. This process was repeated until all four grids were trained and checked across all site-grid training stacks. The final classifier for each site included the training across all four grids loaded as single final.arff data file. This classifier was applied to all image stacks for respective sites for final segmentation results. The training procedures are summarised in the sections below.

5.6.1 *General training set-up procedures*

FIJI *Record Macro* was initiated to record all trace samples during training. *Record Macro* files from each training session were saved for records (e.g. train.ijm) and stored in site classification data folders. Only areas that could be clearly identified as active nests were used for class_1 traces. Only areas that were clearly backgrounds were used for class_2 traces. No attempt was made to try and equalise the class data by providing the classifier with the same number of traces for each class. The site-classifiers were trained using exactly the same procedures. These are detailed further below.

5.6.2 *Grid-training procedures*

Pixels were selected from the control and inactive images slices. The traces were automated with a FIJI macro script. The script was applied to train an initial classifier. No other traces were added. The base-classifier was trained using only the automated annotations. After the first training-run the segmentation results were checked. If required, a *very small* number of inaccurate segmentations, on a single image slice, were assigned to the correct classes with new traces. The classifier was retrained and segmentation results were checked. If segmentations were still not accurate then a very small number of traces were reassigned, on a single slice. The classifier was retrained. If segmentation results deteriorated then the traces added during the previous training-run were removed. Another trace was added, at a different location, and the classifier was retrained. This process was repeated sequentially until all slices were properly segmented.

5.6.3 *Site-training procedures*

Two training runs were used. First-train data were saved to grid-data subfolders. The processes are outlined in Figure 5.9 above. Each site-grid classifier were trained sequentially. Data from the initial grid-training (G_1.arff) were loaded into training for the next grid (G_2.arff). The procedure was repeated until all site-grid classifiers were trained. Final data were saved (e.g. final.arff and final-site.model).

5.6.4 *Application of final site-classifiers*

Data files (final-site.arff) and classifier models (final-site.model) from site-training were loaded into TWS. The classifiers were applied in a batch process. For each site, grid monitoring stacks were selected in sequential order. The binary results from classifications and data were saved to respective site-grid folders.

5.7 POST-PROCESSING

Several morphological operations and pipeline combinations were empirically tested using the small test stack. When the morphological operations were completed, the counted results were visually checked against the original RGB images. They were checked against the values recorded for manual field counts. The post-processing method was automated in a script and applied as a batch process (for example code, see Listing E.5, pg. 168). The procedures used in the post-processing pipeline are summarised in the sections below.

5.7.1 *Binary options*

There were four main binary options, as shown in Figure 5.10 (pg. 75). These were, 1) the number of *Iterations* specified the number of times operations were performed. 2) The number of the *Count* specified the number of adjacent background pixels necessary, before they were removed from the edge of objects during the erosions. It also specified the number of adjacent foreground pixels necessary before pixels were added to the edge of objects during dilations. 3) *Pad edges* was checked. This specified the erosions and closing operations (i.e. edge erosion was not performed during *Erode* or *Close* operations). 4) Finally, EDM output was checked to overwrite the 8-bit input images.

5.7.2 *Post-processing pipeline*

There were four morphological operators used, as outlined in Figure 5.10 (pg. 75). These were, 1) *Fill Holes* was applied to fill holes in binary objects. This was followed by 2) *Close*. Close-operations were set to 10 iterations and 5 counts. This applied dilations[8] followed by erosions[9]. This connected any disconnected parts of the binary images. The process joined breaks, closed holes and smoothed out contours. 3) *Fill Holes* was re-applied to fill any remaining holes. Finally, 4) *Open* was set to 2 iterations and 3 counts. This applied erosions followed by dilations. This separated connected objects in the images. It smoothed out contours by removing isolated objects.

5.7.3 *Analyze Particles for three binary schemes*

Three counting schemes were implemented in the Analyze Particles utility demonstrated in Figure 5.10 (b) (pg. 75). The *Circularity* parameter for counted objects was set between $0.10 - 1.00$ for all schemes. The *Pixel sizes* varied. For the first scheme $p^2 = 10 - \infty$, the second $p^2 = 15 - \infty$ and the third $p^2 = 20 - \infty$. The *Image Overlay* option was checked.

This created a separate image stack of overlays; showing the outlines of counted objects. For an example, refer to Figure 5.10 (c) (pg. 75). The final binary counts were taken as the average count over the three binary schemes and the median values over three image collections. An example post-processing macro snippet is given in Appendix E, Listing E.5 (pg. 168).

5.8 CLASSICAL SEGMENTATION METHODS

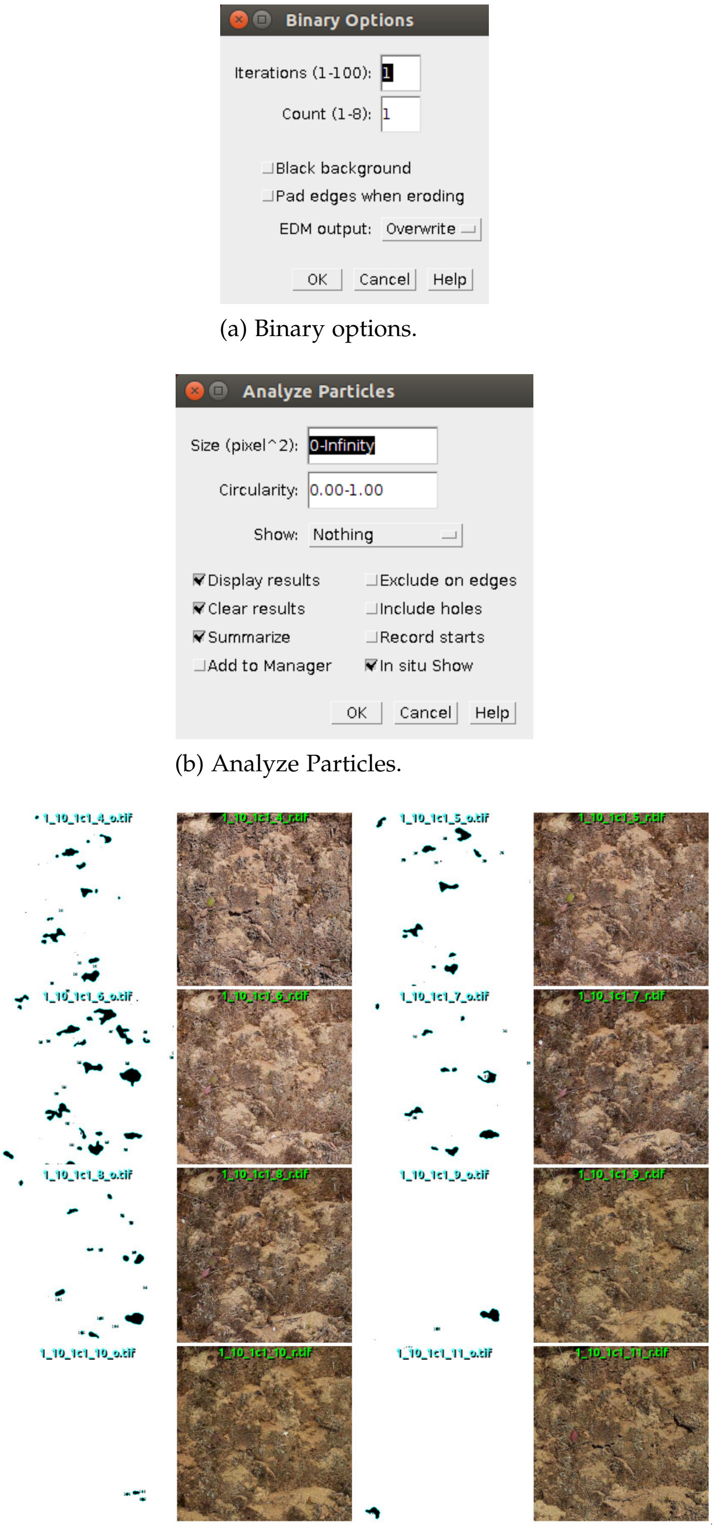
Raw monitoring images were collated into a single stack. The default binary threshold was applied to the stack and it was post-processed using the operators and settings outlined above. The count results were imported into the monitoring spreadsheet. These were used for comparative analysis between automatic counts derived from thresholding and those from segmentations using the CF classifier. Post-processed images with count overlays were collated alongside raw image slices, into a single stack. The stack was saved as a pdf document for manual checking and reporting.

5.9 MANUAL COUNTS FROM IMAGES

Images for manual counts were selected from image data using the spreadsheet list. Images were randomly selected; but only from paired-monitoring data. For example, filler images (ND) were not included. One hundred and seventy images were used for manual im-

[8] *Dilate*: enlarges object borders so holes become smaller.

[9] *Erode*: shrinks the image so holes became larger and small details are deleted.



Binary Options

Iterations (1-100): 1

Count (1-8): 1

□ Black background

□ Pad edges when eroding

EDM output: Overwrite

OK Cancel Help

(a) Binary options.

Analyze Particles

Size (pixel^2): 0-Infinity

Circularity: 0.00-1.00

Show: Nothing

☑ Display results

☑ Clear results

☑ Summarize

□ Add to Manager

□ Exclude on edges

□ Include holes

□ Record starts

☑ In situ Show

OK Cancel Help

(b) Analyze Particles.

(c) Example of binary count overlay images (left) alongside raw RGB images (right) in a combined pdf file.

Figure 5.10: Post-processing options in FIJI. The (a) Binary Options and (b) Analyze Particles user interface. Final overlay results (c) were manually checked against raw RGB images to review segmentations.

age counts. They were collated into a single pdf stack. Another stack was created which included counted overlay results.

This stack was used for verification, as demonstrated in Figure 5.10 (c) above. Two observers conducted counts on the raw RGB images. After a trial run, scorers reviewed which objects could realistically be identified as active nests. Final counts were conducted in a complete run. There were no further discussions between scorers. The first sequence was carried out by the first scorer, followed by the second. Data were entered directly into a spreadsheet by one scorer, as counts were made by the other. Counts were made as quickly as possible. The process was repeated three times by each scorer. The mean counts for each observer were taken and rounded up to whole numbers. These were used in final comparative analysis between the three methods.

5.10 DATA PREPARATION AND ANALYSES

Each image collection was comprised of near-replica images. They were data collected from the same location, grid and day but separated by minutes and seconds. Separate collections were therefore comprised of different images. Each single image was acquired under varying natural conditions. Therefore *median counts* were taken across three image collections. Count data for all methods were prepared in spreadsheets as follows:

- *For automatic-count data*. The mean nest counts from three binary schemes were calculated. Values were rounded up to whole numbers to provide automatic-count subtotals. The median nest counts from three image collections were taken on the automatic-count subtotals. This provided the final automatic-count data used in comparative method analysis (i.e. the CF classifier–*ac* and threshold method–*at*).

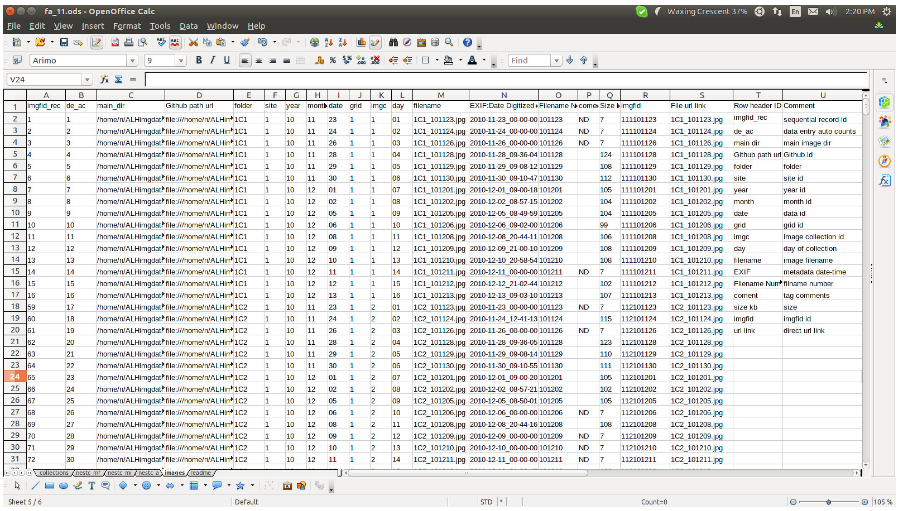
- *For manual-image count data*. The mean of three replica counts were taken for each observer and rounded up to whole numbers. This provided manual-image count data (*mic_ob1 and mic_ob2*). Data were used for inter-observational analysis. The mean count-values from two observers were taken and rounded to up whole numbers. This provided the final manual-image data used for comparative method analysis (*mic_t*).

- *For manual-field count data*. The mean of three replica counts were taken for both observers. They were rounded up to whole numbers. These provided the manual-field data used in comparative method analysis (*mfc_t*).

5.10.1 *Methods comparisons*

Verification of the image-centric method focused on five primary comparative assessments:

1. Automated counts from the RF optimised model (*ac*) and classical thresholding (*at*).
2. Manual counts from the images by two observers (*mic_ob1* and *mic_ob2*).
3. Manual counts from the images (*mic*) and from the field (*mfc*).
4. Automated counts from the RF optimised model (ac) and manual counts from the images (*mic*).
5. Automated counts from the RF optimised model (*ac*) and manual counts from the field (*mfc*).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	imgfid_rec	de_ac	main_dir	Github path url	folder	site	year	montl	date	grid	imgc	day	filename	EXIF:Date Digitized	Filename N	come	Size	imgfid	File url link	Row header ID	Comment
2	1	1	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	11	23	1	1	01	1C1_101123.jpg	2010-11-23_00-00-00	101123	ND	7	111101123	1C1_101123.jpg	imgfid_rec	sequential record id
3	2	2	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	11	24	1	1	02	1C1_101124.jpg	2010-11-24_00-00-00	101124	ND	7	111101124	1C1_101124.jpg	de_ac	data entry auto counts
4	3	3	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	11	26	1	1	03	1C1_101126.jpg	2010-11-26_00-00-00	101126	ND	7	111101126	1C1_101126.jpg	main dir	main image dir
5	4	4	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	11	28	1	1	04	1C1_101128.jpg	2010-11-28_09-36-04	101128		124	111101128	1C1_101128.jpg	Github path url	Github id
6	5	5	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	11	29	1	1	05	1C1_101129.jpg	2010-11-29_09-08-12	101129		108	111101129	1C1_101129.jpg	folder	folder
7	6	6	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	11	30	1	1	06	1C1_101130.jpg	2010-11-30_09-10-47	101130		112	111101130	1C1_101130.jpg	site	site id
8	7	7	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	01	1	1	07	1C1_101201.jpg	2010-12-01_09-00-18	101201		105	111101201	1C1_101201.jpg	year	year id
9	8	8	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	02	1	1	08	1C1_101202.jpg	2010-12-02_08-57-15	101202		104	111101202	1C1_101202.jpg	month	month id
10	9	9	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	05	1	1	09	1C1_101205.jpg	2010-12-05_08-49-59	101205		104	111101205	1C1_101205.jpg	date	data id
11	10	10	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	06	1	1	10	1C1_101206.jpg	2010-12-06_09-02-00	101206		99	111101206	1C1_101206.jpg	grid	grid id
12	11	11	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	08	1	1	11	1C1_101208.jpg	2010-12-08_20-44-11	101208		106	111101208	1C1_101208.jpg	imgc	image collection id
13	12	12	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	09	1	1	12	1C1_101209.jpg	2010-12-09_21-00-10	101209		108	111101209	1C1_101209.jpg	day	day of collection
14	13	13	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	10	1	1	13	1C1_101210.jpg	2010-12-10_20-58-54	101210		108	111101210	1C1_101210.jpg	filename	image filename
15	14	14	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	11	1	1	14	1C1_101211.jpg	2010-12-11_00-00-00	101211	ND	7	111101211	1C1_101211.jpg	EXIF	metadata date-time
16	15	15	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	12	1	1	15	1C1_101212.jpg	2010-12-12_21-02-44	101212		102	111101212	1C1_101212.jpg	Filename Num	filname number
17	16	16	/home/n/ALHimgdat	file:///home/n/ALHim	1C1	1	10	12	13	1	1	16	1C1_101213.jpg	2010-12-13_09-03-10	101213		107	111101213	1C1_101213.jpg	coment	tag comments
18	59	17	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	11	23	1	2	01	1C2_101123.jpg	2010-11-23_00-00-00	101123	ND	7	112101123	1C2_101123.jpg	size kb	size
19	60	18	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	11	24	1	2	02	1C2_101124.jpg	2010-11-24_12-41-13	101124		115	112101124	1C2_101124.jpg	imgfid	imgfid id
20	61	19	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	11	26	1	2	03	1C2_101126.jpg	2010-11-26_00-00-00	101126	ND	7	112101126	1C2_101126.jpg	url link	direct url link
21	62	20	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	11	28	1	2	04	1C2_101128.jpg	2010-11-28_09-36-05	101128		123	112101128	1C2_101128.jpg		
22	63	21	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	11	29	1	2	05	1C2_101129.jpg	2010-11-29_09-08-14	101129		110	112101129	1C2_101129.jpg		
23	64	22	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	11	30	1	2	06	1C2_101130.jpg	2010-11-30_09-10-55	101130		111	112101130	1C2_101130.jpg		
24	65	23	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	12	01	1	2	07	1C2_101201.jpg	2010-12-01_09-00-20	101201		105	112101201	1C2_101201.jpg		
25	66	24	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	12	02	1	2	08	1C2_101202.jpg	2010-12-02_08-57-21	101202		102	112101202	1C2_101202.jpg		
26	67	25	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	12	05	1	2	09	1C2_101205.jpg	2010-12-05_08-50-01	101205		105	112101205	1C2_101205.jpg		
27	68	26	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	12	06	1	2	10	1C2_101206.jpg	2010-12-06_00-00-00	101206	ND	7	112101206	1C2_101206.jpg		
28	69	27	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	12	08	1	2	11	1C2_101208.jpg	2010-12-08_20-44-16	101208		108	112101208	1C2_101208.jpg		
29	70	28	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	12	09	1	2	12	1C2_101209.jpg	2010-12-09_00-00-00	101209	ND	7	112101209	1C2_101209.jpg		
30	71	29	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	12	10	1	2	13	1C2_101210.jpg	2010-12-10_00-00-00	101210	ND	7	112101210	1C2_101210.jpg		
31	72	30	/home/n/ALHimgdat	file:///home/n/ALHim	1C2	1	10	12	11	1	2	14	1C2_101211.jpg	2010-12-11_00-00-00	101211	ND	7	112101211	1C2_101211.jpg		

Figure 5.11: The image data sheet shown was one of six used for monitoring data entry: 1. collections (basic records), 2. nestc_mf (manual-field counts), 3. nestc_mi (manual-image counts), 4. auto_c (automatic counts), 5. images (image data files and hard links) and 6. readme (row header descriptions).

Statistical analysis were performed in RStudio[10]. They were formatted in spreadsheets to accommodate easy import. Data were organised using a single header row and arranged using date-time or grid-image sequences, in single long columns. Examples are shown above in Figure 5.11. The first column for each sheet contained a sequential base reference identification. Zero count data were input as 0; all *no-data* entries were identified by a text value entered as ND. This is also demonstrated above in Figure 5.11, under column P–comments.

[10] *RStudio*-http://www.rstudio.com

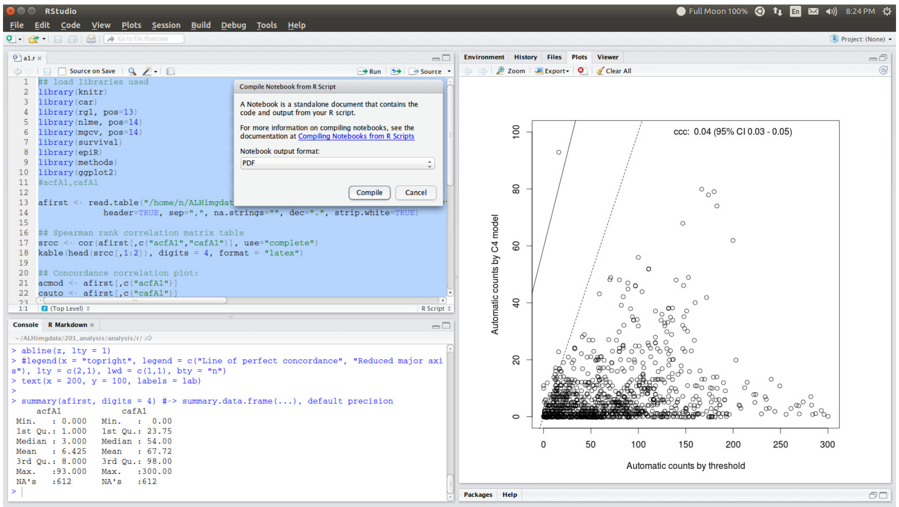
5.10.2 *Statistical tests*

Lin's Concordance of Correlation (ρ_c) was used to compare methods. The statistic combines measures of accuracy and precision to test the agreement between two observations [179]. Pearson's correlation coefficient (r) indicates how scattered the data points are around the line of best fit [179]. This gives a *measure of precision*. The value of μ defines the scale shift and measures systematic bias compared with actual values; ν defines the location shift and measures the difference between actual and measured values; and C_b is a bias correction factor calculated using ν and μ. C_b gives a *measure of accuracy*. A *perfect concordance* between actual and measured values would return $\rho_c = 1$, $r = 1$, $C_b = 1$, $\nu = 1$, and $\mu = 0$. In the equation $\rho_c = rC_b$ and r is the correlation coefficient. C_b is determined by:

$$C_b = [\frac{(\nu + 1/\nu + u^2)}{2}]^{-1} \tag{3}$$

Where:
σ is the variance of x and y
$\nu = \sigma_x/\sigma_y$
μ is the mean value of x and y respectively
$u = (\mu_x - \mu_y)/\sqrt{\sigma_c \sigma_y}$

The value ρ_c was calculated using the epi.ccc function in the epiR package [184]. The RStudio graphical user interface is shown below, in Figure 5.12. The epi.ccc function is provided in Listing E.15 (pg. 181). Graphical analyses were saved as results.tiff for records. A notebook was compiled from the R script associated with each analysis. These data were saved as a pdf.



```
## load libraries used
library(knitr)
library(car)
library(rgl, pos=13)
library(nlme, pos=14)
library(mgcv, pos=14)
library(survival)
library(epiR)
library(methods)
library(ggplot2)
#acfA1,cafA1

afirst <- read.table("/home/n/ALHimgdat
              header=TRUE, sep=",", na.strings="", dec=".", strip.white=TRUE)

## Spearman rank correlation matrix table
srcc <- cor(afirst[,c("acfA1","cafA1")], use="complete")
kable(head(srcc[,1:2]), digits = 4, format = "latex")

## Concordance correlation plot:
acmod <- afirst[,c("acfA1")]
cauto <- afirst[,c("cafA1")]
```

Compile Notebook from R Script

A Notebook is a standalone document that contains the code and output from your R script.

For more information on compiling notebooks, see the documentation at Compiling Notebooks from R Scripts

Notebook output format: PDF

Compile Cancel

~/ALHimgdata/201_analysis/analysis/r/

```
> abline(z, lty = 1)
> #legend(x = "topright", legend = c("Line of perfect concordance", "Reduced major axi
s"), lty = c(2,1), lwd = c(1,1), bty = "n")
> text(x = 200, y = 100, labels = lab)
>
> summary(afirst, digits = 4) #-> summary.data.frame(...), default precision
     acfA1            cafA1
 Min.   : 0.000   Min.   :  0.00
 1st Qu.: 1.000   1st Qu.: 23.75
 Median : 3.000   Median : 54.00
 Mean   : 6.425   Mean   : 67.72
 3rd Qu.: 8.000   3rd Qu.: 98.00
 Max.   :93.000   Max.   :300.00
 NA's   :612      NA's   :612
>
```

Figure 5.12: RStudio graphical user interface showing the operation of the script in Listing E.15.

5.11 SUMMARY OF IMAGING METHODS

The imaging tools and methods used to process the images of active nests were outlined in this chapter. This included the management of the digital data and the design of the image collections database. Image analysis tools and methods were fully detailed so they can be replicated for use in similar applications. Open source biomedical image analysis software FIJI was used for most of the imaging tasks. The main task for nest monitoring photographs, was delineating the areas in images that corresponded to active nests from other background areas or objects. Therefore this chapter included the tests used to investigate the performances of a range of image segmentation tools.

The images of active nests were sufficiently segmented by using an interactive trainable tool in the TWS plug-in which is included in the FIJI package. TWS utilises human knowledge for image segmentations by combining the traces selected by a user which represent key objects. The TWS procedures used to classify images of active nests were more dynamic compared to than classical segmentation techniques. Since TWS uses an interactive process, replicating the exact nest image segmentations could be problematic. Thus the specific methods used to train, construct and apply monitoring classifiers, were fully detailed in this chapter. This included the methods used to test, optimise and verify the performance of the RF machine learner, which was the selected classifier used for segmenting monitoring images via TWSs. The final sections in this chapter were dedicated to verification of results and statistical methods.

Part III

RESEARCH OUTCOMES

6 RESULTS

Synopsis

The results are presented in this chapter. Active nest classifiers were based on RF models which were optimised for speed and accuracy. The results from feature engineering investigations, the number of forest-trees and the number of random forest-features were examined before final nest-classifiers were constructed and applied to monitoring images. Classifier benchmarks and segmentation performance were evaluated and are summarised. The number of active nests by method and by year are outlined in the final sections of this chapter.

6.1 CLASSIFIER TUNING

A summary of classifier tuning tests are shown in Table 6.1 below. Five main tests (T1–T5) were performed. The results were used to evaluate the optimal parameters for the final RF monitoring classifier (CF).

Table 6.1: Results of tests for classifier models C1–C5. Performance parameters were compared using the out of bag error (oo_b%) and the time taken to construct the features stack (ms).

Test	Classifier	F_n	N	M	σ_{max}	oo_b%	ms
T1	C1	79	200	2	16	1.98	147138
T2	C2	46	200	2	4	0.93	402974
T3	C3	20	200	2	2	0.698	23092
T4	C4	20	50	2	2	1.279	12503
T5	C5	20	50	2	2	0.628	12503

Model parameters key	
Number of features used to construct the features stack	F_n
Number of trees	N
Number of random features	M
Maximum sigma	σ_{max}

6.1.1 *Default features*

In Test 1, the following filters were selected: Guassian blur, Sobel filter, Hessian, Difference of gaussians, and Membrane projections. The number of features: $F_n = 79$, trees: $N = 200$, random features: $M = 2$ and $\sigma_{max} = 16$.

The TWS default model and features settings produced model with a features stack of 79. The number of pixels selected as $class_1 = 44$ and $class_2 = 816$. The feature stack for six slices with 79 features took $147138ms$ to create and the $oo_b = 1.98\%$.

Of the 79 filters provided for classifier construction, 46 did not provide any additional information. Feature importances were less than zero for 58% of the filters. The top twenty most important features for test classifier C1 are shown in Figure 6.1, pg.83.

6.1.2 *Optimised features*

In Test 2, the following filters were selected: Guassian blur, Mean, Minimum, Median, Anisotropic diffusion, Bilateral, Lipschitz, Kuwahara and Structures. The number of features: $F_n = 46$, trees: $N = 200$, random features: $M = 2$ and sigma: $\sigma_{max} = 4$.

The TWS settings for classifier C2 produced a features stack of 46. The number of pixels selected as $class_1 = 44$, $class_2 = 816$ and the out of bag error improved to $oo_b = 0.93\%$. The feature stack for six slices with 46 features took $402974ms$ to create.

Of the 46 filters provided for classifier construction around 22 did not provide any additional information. Feature importances were less than zero for 48% of the filters. However, the oo_b error reduced when the excess filters were removed and the new textural features were added. The top twenty most important features for test classifier C2 are shown in Figure 6.2, pg.84.

6.1.3 *Optimised speed*

In Test 3, the following filters were selected: Gaussian blur, Mean, Minimum, Median and Structures. The number of features: $F_n = 20$, trees: $N = 200$, random features: $M = 2$ and sigma: $\sigma_{max} = 2$.

The TWS settings for classifier C3 produced a features stack of 20. The number of pixels selected as $class_1 = 44$, $class_2 = 816$ and the out of bag error improved to $oo_b = 0.698\%$. The feature stack for six slices with 20 features took $23092ms$ to create. The total model processing was around 6 x faster than C1 and 17 x C2. All 20 filters provided information as shown in Figure 6.3, pg.85.

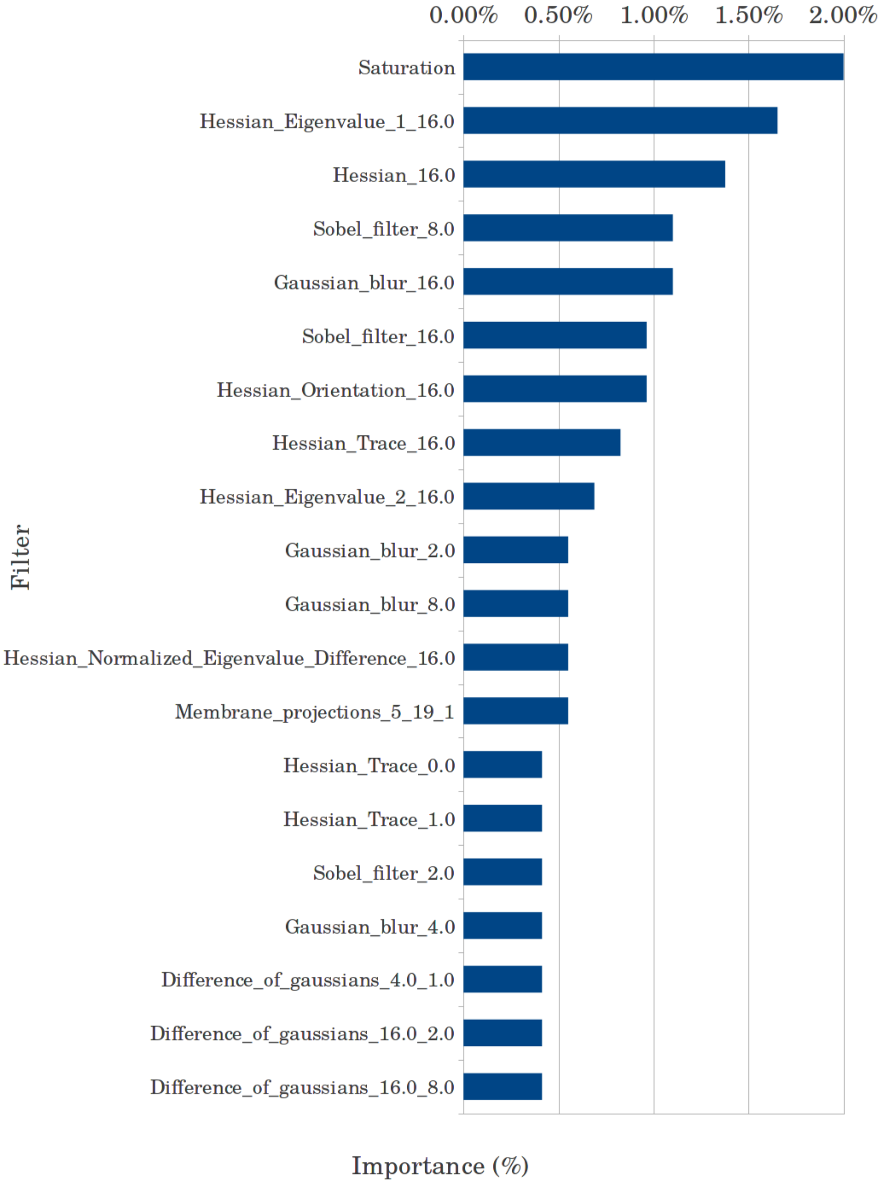


Figure 6.1: *Test 1*, feature importance results for classifier C1. The model used default settings. The feature stack for six slices with 79 features took 147138ms to create and the $oo_b = 1.98\%$. The highest ranking filters and respective importances are listed.

.

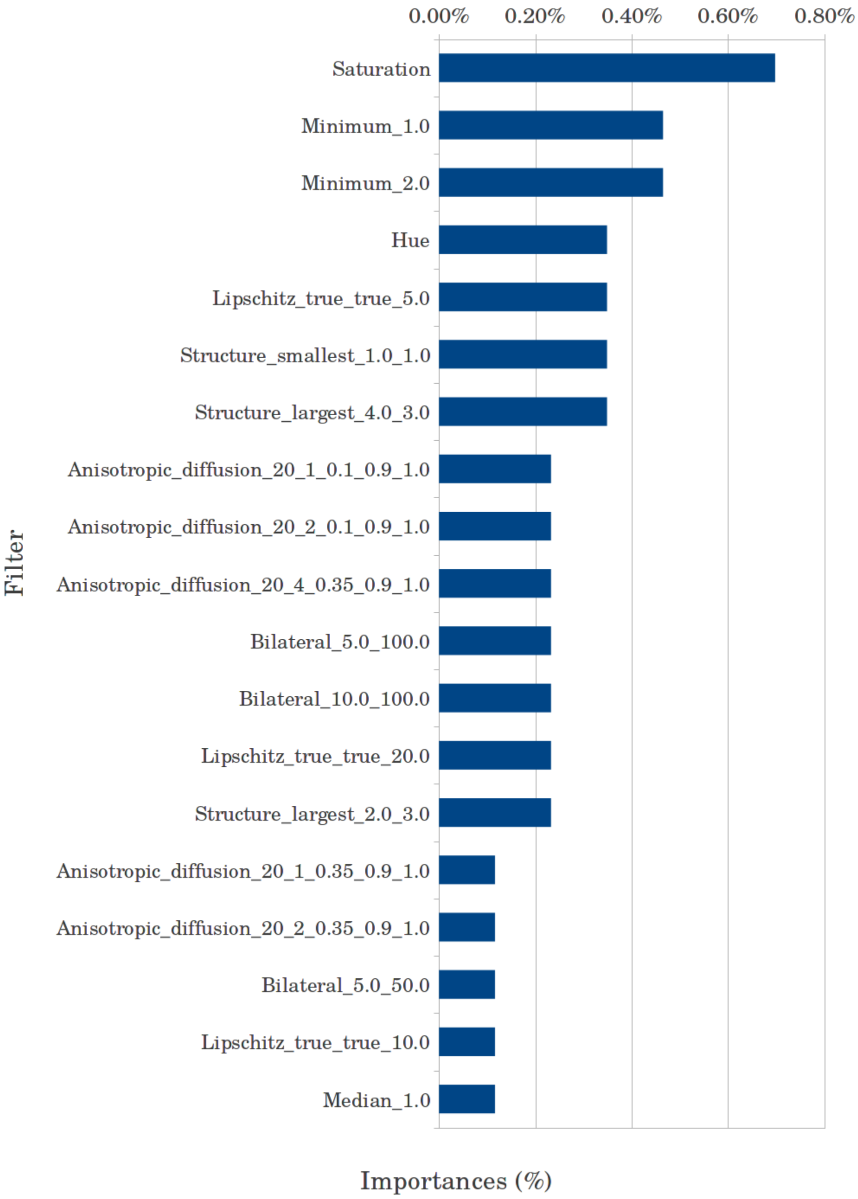


Figure 6.2: *Test 2*, feature importance tests on classifier C2. The model was optimised to produce the best segmentation of nest images. The feature stack for six slices with 46 features took $402974ms$ to create and the $oo_b = 0.93\%$. The highest ranking filters and respective importances are listed.

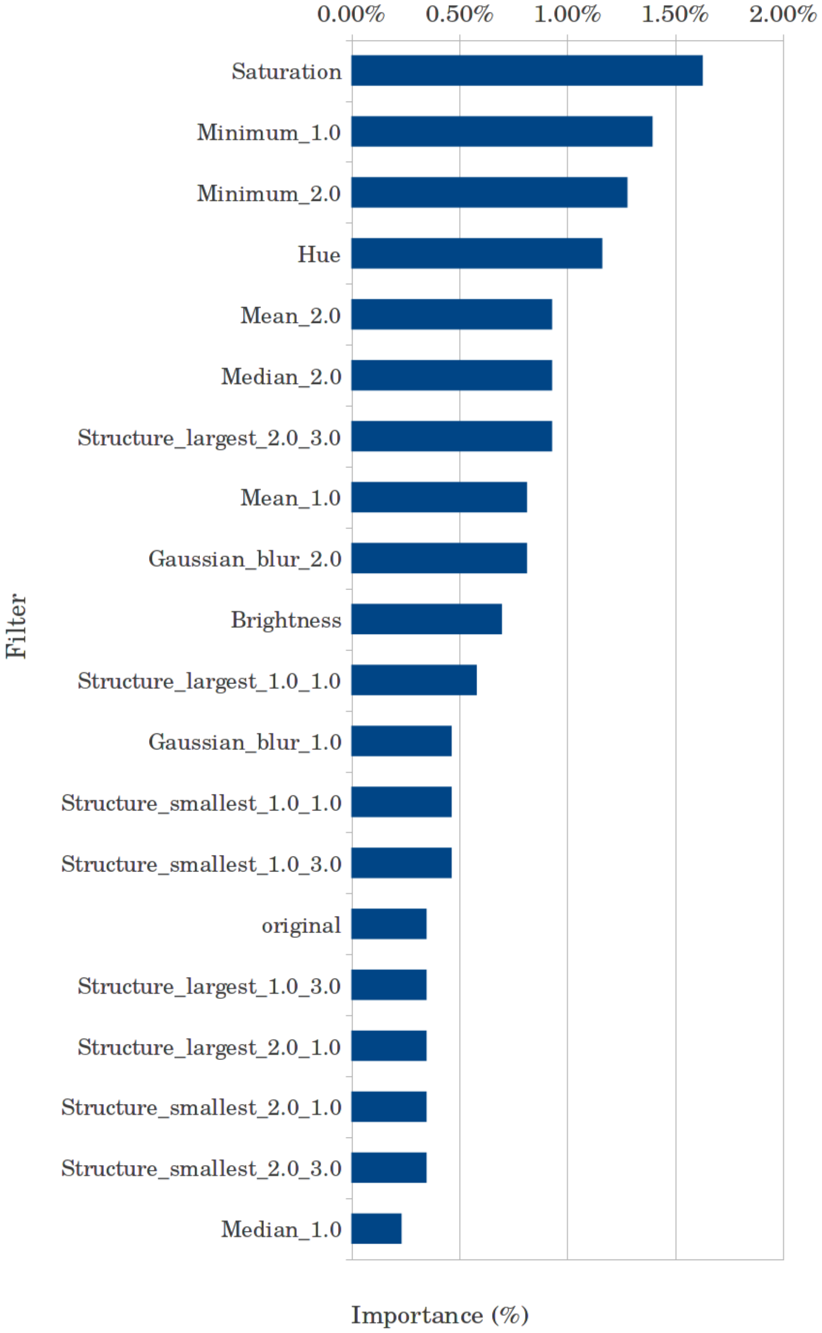


Figure 6.3: *Test 3*, feature importance tests on classifier C3. The model was optimised to speed up processing. The feature stack for six slices with 20 features took 23092ms to create and the $oo_b = 0.698\%$. All twenty filters were important and ranked as listed.

6.1.4 *Number of trees*

In Test 4, twenty-two RF models were loaded into WEKA Experimenter; configured for a 10 fold cross validation, 10 iteration test. The number of trees were adjusted $N = 10 - 1000$ for each of the RF models. This is shown along the bottom axis in Figure 6.4, pg.87. Processing time exponentially increased with the number of trees. Performance gains did not appear significant beyond $N = 150$.

Classifier C4 was re-run via TWS with $N = 50$, $M = 2$ and $\sigma_{max} = 2$. The TWS settings for classifier C4 produced a features stack of 20. The number of pixels selected as $\mathtt{class}_1 = 44$, $\mathtt{class}_2 = 816$. The feature stack for six slices with 20 features took $17503ms$ to create; and the $oo_b = 1.279\%$. The total model processing was around 12 x faster than C1; 32 x faster than C2 and 2 x faster than C3.

The oo_b from TWS was different than reported in WEKA Experimenter for the RF model with $N = 50$, $M = 2$ and $\sigma_{max} = 2$. WEKA reported the $oo_b = 1.083\%$. This is highlighted in orange on Figure 6.4 (pg.87). Tests were repeated using the rep_nest.arff dataset. An alternative set-up was loaded into WEKA Experimenter. A 66.67% split between training and tests (using randomised data) were applied in 500 iterations ($n = 500$). The error changed slightly to $oo_b = 1.151\%$; indicating the differences in oo_b were most likely a consequence of the test configuration (i.e. evaluations run with a cross validation or % data splits) [185, 186].

6.1.5 *Number of random features*

In Test 5, twenty RF models were added to WEKA Experimenter; configured for a 10 fold cross validation, 10 iteration test. The number of random features were adjusted between $M = 0 - 20$, for each of the RF models. This is shown along the bottom axis on Figure 6.5, pg.88.

Classifier C5 was re-run via TWS with $N = 50$, $M = 8$ and $\sigma_{max} = 2$. The TWS settings for classifier C5 produced a features stack of 20. The number of pixels selected as $\mathtt{class}_1 = 44$, $\mathtt{class}_2 = 816$. The feature stack for six slices with 20 features took $17503ms$ to create; $oo_b = 0.628\%$. The total model processing was around 12 x faster than C1; 32 x faster than C2, 2 x faster than C3 and was equal to C4.

The test results showed the performance improved when more random features were provided to the model for training and construction. The out of bag error for C5 ($oo_b = 0.628\%$) was lower than C5 ($oo_b = 1.279\%$). The final monitoring classifier (CF) used a RF with $F_n = 20$, $N = 50$, $M = 8$ and $\sigma_{max} = 2$. This is highlighted in orange on Figure 6.5 (pg.88).

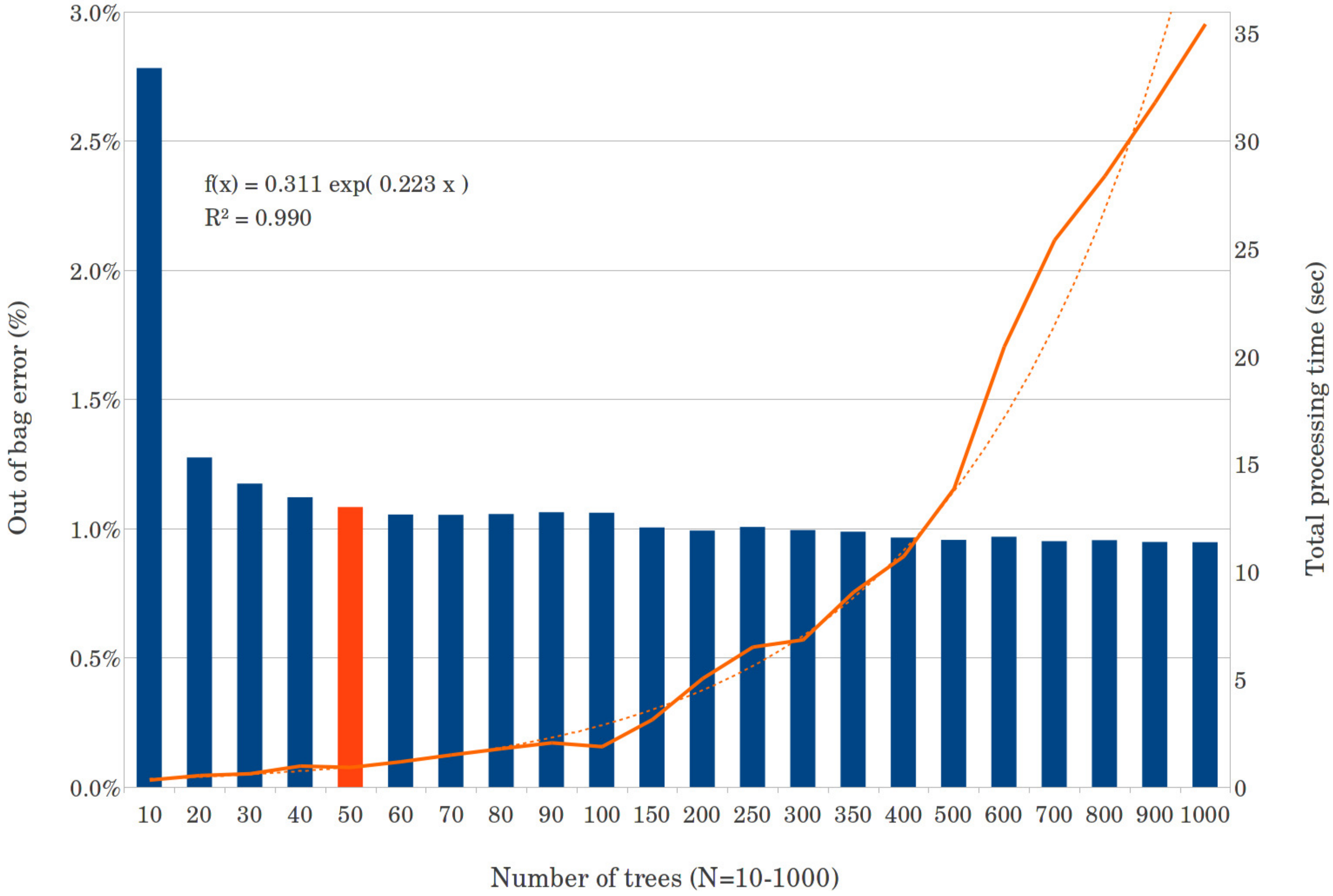


Figure 6.4: *Test 4*, 50 trees ($N = 50$) were selected for the random forest model final monitoring classifier (CF). The out of bag error from WEKA was $oo_b = 1.083\%$ (orange bar) and total processing time $t_{total} = 0.925(sec)$ using the rep_nest.arff dataset.

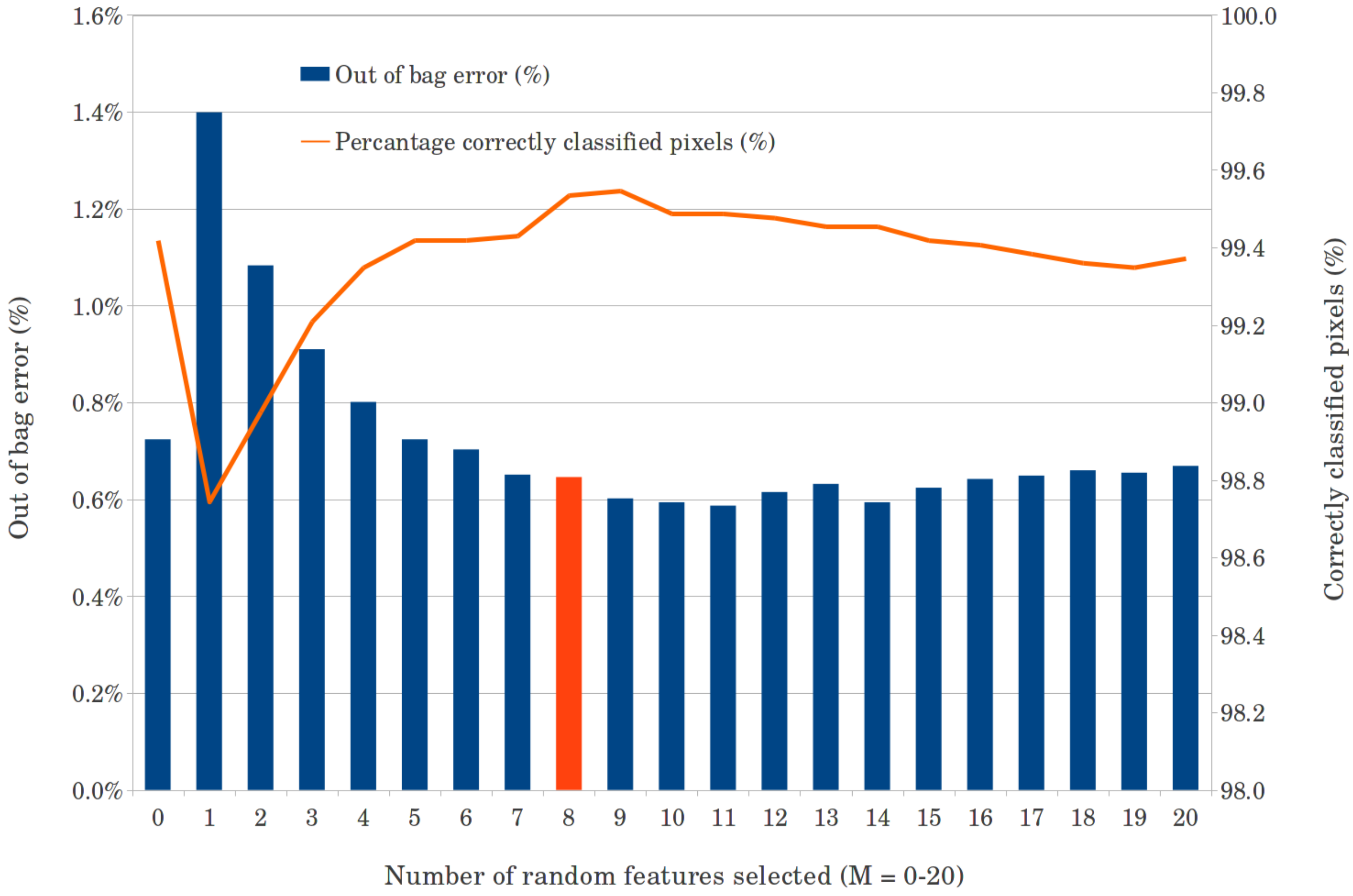


Figure 6.5: *Test 5*, E8 random features (M = 8) were selected for the random forest model final monitoring classifier (CF). The out of bag error from WEKA was $oo_b = 0.628\%$ (orange bar) and the percentage of correctly classified instances (pixels) = 99.4% using the rep_nest.arff dataset.

The final test classifier CF was compared against nine machine learners in WEKA Experimenter. The results are listed in Table 6.2 below. A 10 fold cross validation with 10 iterations was used in the experiment. The percentage of correctly classified instances were tested for $n = 1000$, with a confidence of 0.05 in a paired-corrected two tailed test.

Results showed there were no significant improvements by any other models over CF; four gave results that were statistically worse. The naive Bayes model (M8) did not perform as well as CF on the test data-set with 90.35% correct. The VotedPerceptron (M9) and SMO (M10) models also returned a lower number of correctly classified instances; 95.71% and 96.74% respectively. The ZeroR (M2) gave 94.88% correctly classified instances, slightly higher than naive Bayes model (M8).

Table 6.2: Results of performance evaluations conducted in WEKA Experimenter. The final test classifier CF (M1) was benchmarked against nine other common models on the rep_nest.arff dataset.

<table>
<tr><th>Model</th><th>Correct (%)</th><th>WEKA model code</th></tr>
<tr><td>M1</td><td>99.42</td><td>hr.irb.fastRandomForest.FastRandomForest-I|50|-K|8|-S|1</td></tr>
<tr><td>M2</td><td>94.88 ●</td><td>rules.ZeroR</td></tr>
<tr><td>M3</td><td>98.37</td><td>trees.J48-C 0.25 -M|2</td></tr>
<tr><td>M4</td><td>98.02</td><td>trees.RandomTree-K|0|-M|1.0|-V|0.0010|-S|1</td></tr>
<tr><td>M5</td><td>99.19</td><td>trees.RandomForest-I|10|-K|0|-S|1|-num-slots|1</td></tr>
<tr><td>M6</td><td>98.95</td><td>hr.irb.fastRandomForest.FastRandomForest-I|50|-K|2|-S|1</td></tr>
<tr><td>M7</td><td>99.07</td><td>hr.irb.fastRandomForest.FastRandomForest-I|200|-K|2|-S|1</td></tr>
<tr><td>M8</td><td>90.35 ●</td><td>bayes.NaiveBayes</td></tr>
<tr><td>M9</td><td>95.71 ●</td><td>functions.VotedPerceptron-I|1|-E 1.0|-S|1|-M|10000</td></tr>
<tr><td>M10</td><td>96.74 ●</td><td>functions.SMO-C|1.0|-L|0.0010|-P|1.0E-12|-N|0|-V|-1|-W|1|-K|
functions.supportVector.PolyKernel-E|1.0|-C|250007</td></tr>
</table>

○, ● statistically significant improvement or degradation

6.3 SEGMENTATION PERFORMANCE

The representative image stack of six images were fully processed using the final test classifier (CF). A trace was added and the model was re-trained (CF2). Another trace was added and the model was trained again (CF3). The complete processing pipeline is shown in Figure 6.6 below. When the second trace was included in training (CF3), the out of bag error decreased slightly, to $oo_b = 0.98\%$. The final counts were visually checked against raw RGB images and against the manual field counts.

Five classical threshold methods were tested. They included: Huang, Mean, MinError, Min and Otsu. Six SRM parameters were tested. Each test with varying segmentation regions (Q) set as follows: Q = 1, 2, 3, 6, 8 and 16. Six edge-based Canny-Deriche filtering methods were tested. Each test with varying smoothing values (α) as follows: α = 1.0, 0.9, 0.7, 0.6, 0.45 and 0.15. Results are detailed in Table 6.3 (pg. 93).

6.4 TRAINING STACKS

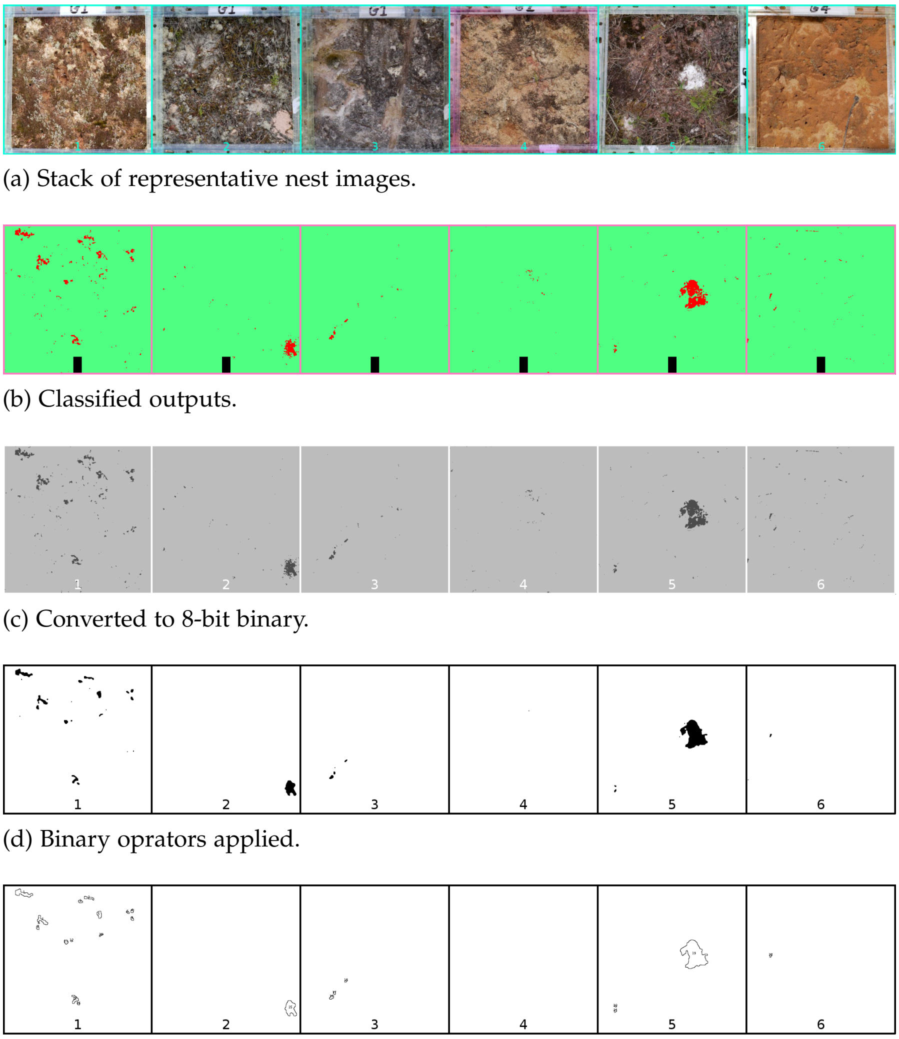
The four training stacks were collated for each site. They are shown in Figures 6.7, 6.8, and 6.9 (pg. 94–96). They demonstrate the wide variation in images between and within monitoring sites. All training stacks contained an image of the control grid, (e.g. slice 2 for Mt. Tiger and Mt. Parihaka, and slice 1 for Memorial Drive) and the images of inactive nests, acquired May 21, 2013. Slices were arranged in date sequence. These were used for classifier training; final classifiers were applied to all monitoring stacks.

6.5 MONITORING DATA SUMMARY

A summary of collections are shown in the next few sections, including a statistical summary of data, and preliminary distribution evaluations. Once the CF classifiers were trained, processing (classifications and post-processing) was completed in less than 24 hours. Field monitoring was conducted in a single two and a half hour round trip, beginning at site 1 and ending at site 3. The time to conduct manual nests counts was approximately 15 minutes per site. The time taken to acquire images was around 5 minutes per site.

6.5.1 *Collections*

A total of 158 monitoring days resulted in 632 images, per image collection. Around 1/3 of image data were unrecoverable (612 images). A summary of collections over five years (2010–2014) for Mt. Tiger, Mt. Parihaka and Memorial Drive are shown Table 6.4, pg. 97



(a) Stack of representative nest images.

(b) Classified outputs.

(c) Converted to 8-bit binary.

(d) Binary oprators applied.

(e) Particle Anlysis.

Figure 6.6: Representative images in (a) the test stack used to train initial classifiers (C1–C5). The classified results from CF3 (b) were post-processed (c)-(d) to give the (e) final counts. These were compared against classical segmentation methods, visual nest counts from images, and the manual counts taken in the field.

Table 6.3: The final automatic count results on representative images (slices 1–6) using CF(1–3) compared to classical segmentation methods, manual-image and manual-field counts.

Method	Q/or α	slice 1	slice 2	slice 3	slice 4	slice 5	slice 6
Manual-image		3	3	3	2	2	2
Manual-field		2	4	4	3	4	9
CF1		21	11	14	22	4	64
CF2		21	2	3	13	3	14
CF3		7	3	8	4	3	8
Haung4		87	89	51	50	51	62
Mean5		87	89	51	50	51	62
MinError6		87	89	51	50	51	62
Min7		87	89	51	50	51	62
Otsu8		87	89	51	50	51	62
Srm9	1	1	2	0	0	2	2
Srm10	2	1	2	1	1	1	1
Srm11	3	4	3	2	2	5	5
Srm12	6	41	57	10	28	99	14
Srm13	8	60	72	24	42	114	20
Srm14	16	93	62	41	70	95	25
Cedge15	0.15	88	92	48	62	39	10
Cedge16	0.45	170	202	71	91	133	20
Cedge17	0.6	191	230	63	99	149	22
Cedge18	0.7	179	225	49	96	169	19
Cedge19	0.9	175	225	64	103	143	22
Cedge20	1.0	189	226	46	103	179	16

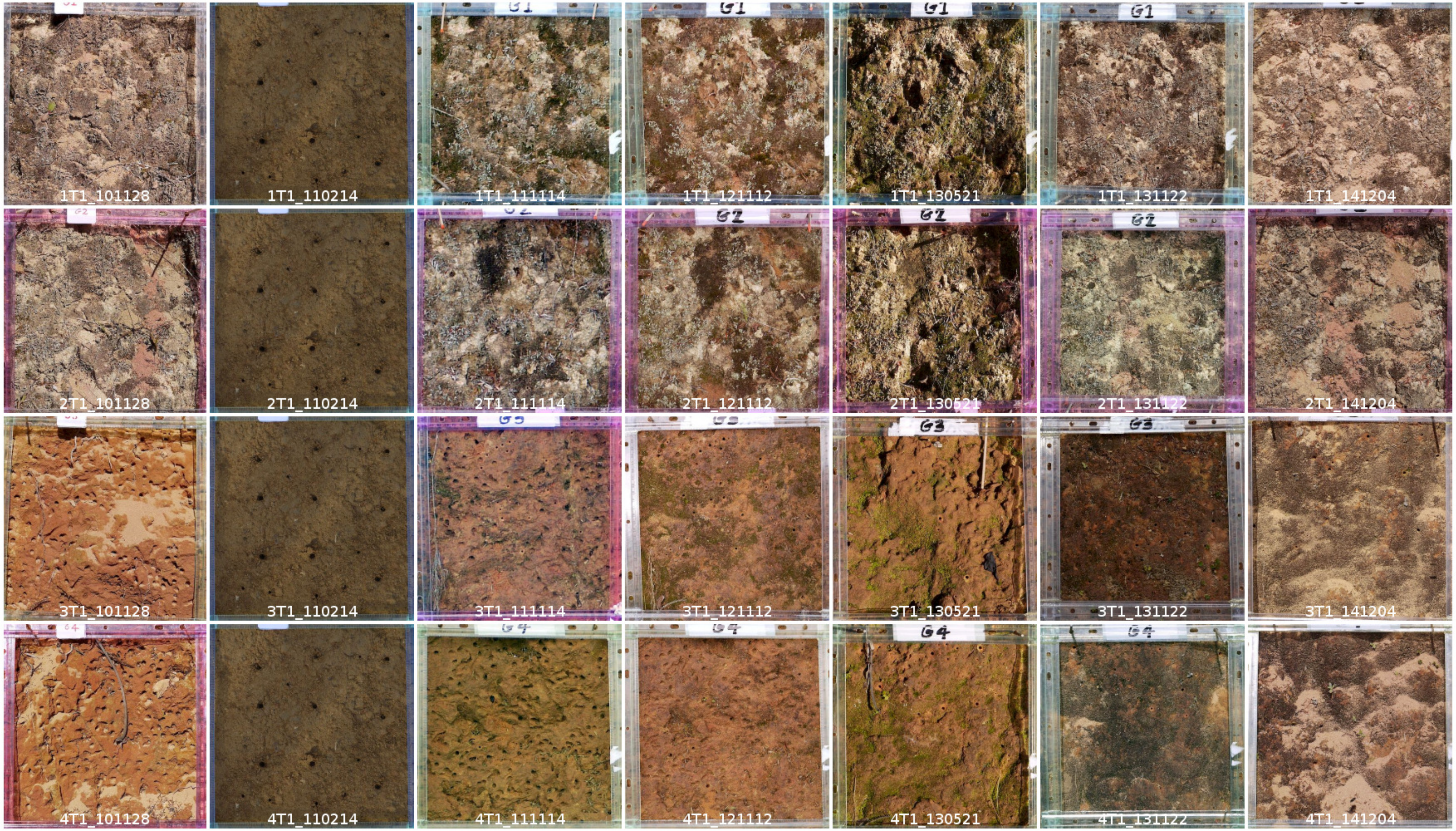


Figure 6.7: *Site 1* training stack for Mt. Tiger. Slices of collection years from left to right and grid numbers 1–4, from top to bottom. Slice 1 = control and slice 4 = inactive nest. All nests were on a roadside bank with an approximate slope ranging between 60–80°. There were no horizontal ground nests at this location. Images from grids 1–2 and grids 3–4 were similar in appearance.

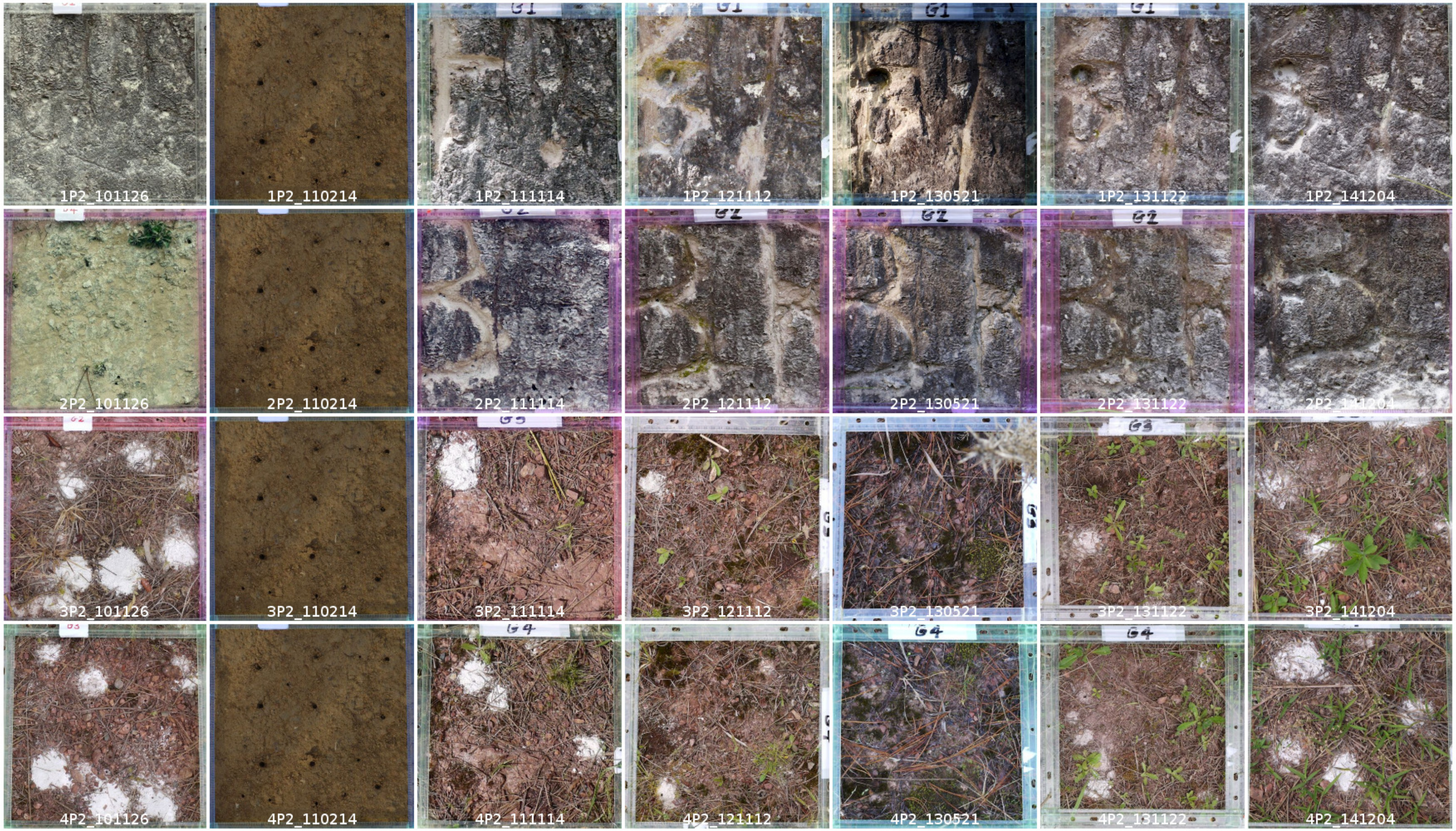


Figure 6.8: *Site* 2 training stack for Mt. Parihaka. Slices of collection years from left to right and grid numbers 1–4, from top to bottom. Slice 1 = control and slice 4 = inactive nest. The nests in grids 1–2 were on a large bank (5m long x 8m high) with an approximate slope ranging between 60–80°; grids 3–4 were horizontal ground nests. Images from grids 1–2 and grids 3–4 were similar in appearance.

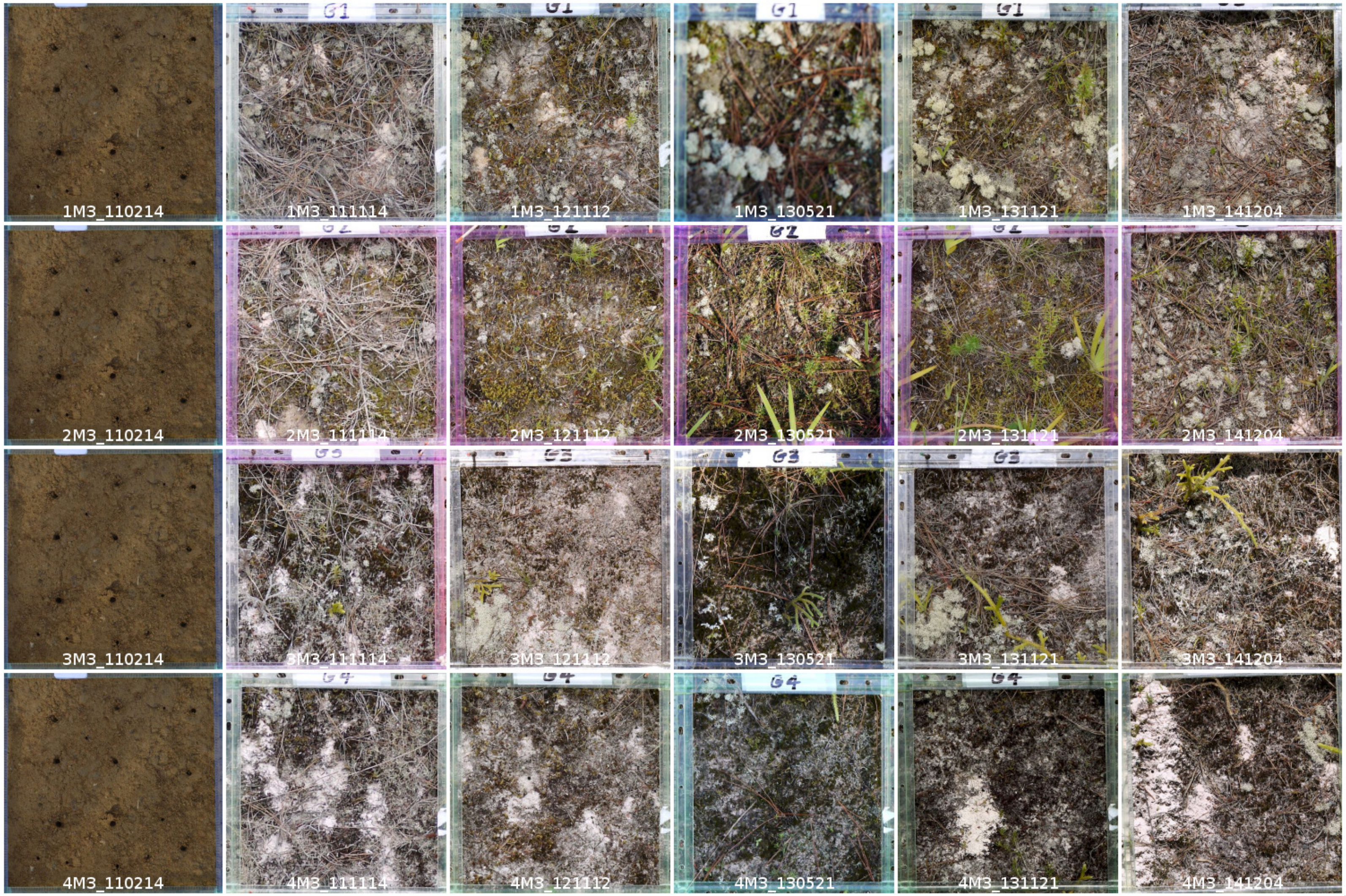


Figure 6.9: *Site 3*, training stack for Memorial Drive. Slices of collection years from left to right and grid numbers 1–4, from top to bottom. Slice 2 = control and slice 5 = inactive nest. All nests were on a roadside bank with an approximate slope ranging between 60–80°. There were no horizontal ground nests at this location. All grid images were similar in appearance.

Table 6.4: Field collection summary of monitoring days for three locations, Mt. Tiger (S1), Mt. Parihaka (S2) and Memorial Drive (S3) over five years (2010–2014).

Site ID	2010	2011	2012	2013	2014	Total days
S1	16	16	13	10	3	59
S2	16	16	13	10	3	59
S3	ND	16	13	10	3	43
Yearly totals	32	48	39	30	9	158

*Total processed images = 1896

*Monitoring images = 1284

*Field samples = 632

6.5.2 *Descriptive statistics*

The mean, standard deviation and variances for manual field and automatic counts (derived from classifier segmentations) are shown in Table 6.5 below. The variances were larger than the means for both methods. The dispersal index (DI) suggests clustering; data fits a negative binomial model.

Table 6.5: Descriptive statistics summary.

Method	n	μ	SE	σ	SD	DI	CV
Automatic	1284	6.424	0.284	103	10	16	1.58
Manual	632	4.545	0.136	35	5	7	1.31

Table key

Sample	n
Mean	μ
Standard error of the mean $(SD/\sqrt{n})$	SE
Variance	σ
Standard deviation	SD
Dispersal index $(\sigma/mean)$	DI
Coefficient of variation $(SD/mean)$	CV

6.5.3 *Frequency distribution*

The frequency distribution of automatic and manual field nest counts using observed and expected values are shown in Figure 6.10. The observed data agrees with the expected frequency distribution of a negative binomial model for the automatic method. The observed and predicted frequency distribution for the manual method, does not show the same strength of agreement with a negative binomial model.

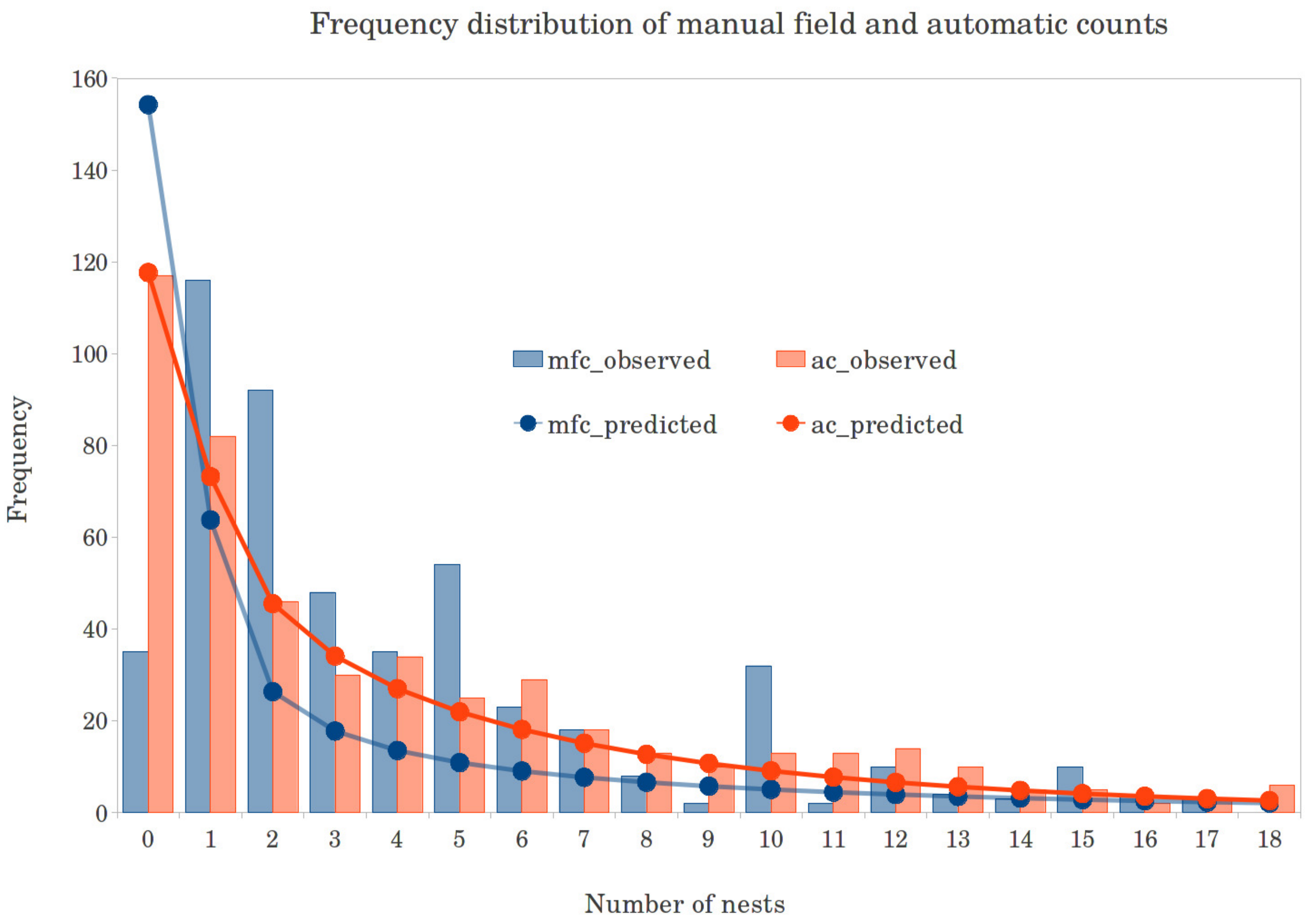


Figure 6.10: Frequency distribution of automatic (ac in red) and manual field (mfc in blue) nest counts. Histograms show the observed frequency distributions for manual field counts and automatic counts. Matching coloured lines, joined with closed circles are the predicted frequencies expected from a negative binomial model.

6.6 NUMBER OF ACTIVE NESTS BY METHOD

Lin's concordance of correlation was used to compare methods by measuring the precision (r), agreement (ρ_c) and accuracy (C_b). Data were calculated using the epi.ccc function within the epiR package. Plots were generated in epiR to display the line of perfect concordance (dashed) and the line of best fit (solid) for each analysis (1–5). These are summarised in Table 6.6.

Table 6.6: Five comparative analyses. A summary of the most important descriptive parameters used to measure precision (r), agreement (ρ_c) and accuracy (C_b) between methods.

Analysis	Method 1	Method 2	n	r	ρ_c	C_b
A1	a-ths	a-CF	1284	0.244	0.040	0.164
A2	m-image ob1	m-image ob2	170	0.891	0.867	0.973
A3	m-field	m-image	170	0.641	0.622	0.97
A4	m-image	a-CF	170	0.705	0.679	0.963
A5	m-field	a-CF	520	0.828	0.738	0.891

Table key

Automatic (a-) and manual (m-) methods

Images segmented by monitoring classifier CF	a-CF
Images segmented by default thresholds	a-ths
Nests counted from images by two scorers	ob1-2
Number of (paired) samples	n
Pearson's correlation coefficient (precision)	r
Lin's Concordance of Correlation (agreement)	ρ_c
Bias correction factor (accuracy) $= \rho_c/r$	C_b

6.6.1 *Classical threshold and machine learner*

An evaluation of nest counts derived from thresholding and CF model segmentations are shown in Figure 6.11 below. The methods showed poor agreement ($\rho_c = 0.040$) and accuracy ($C_b = 0.164$); with a very weak positive correlation ($r = 0.214$, $P < 0.05$).

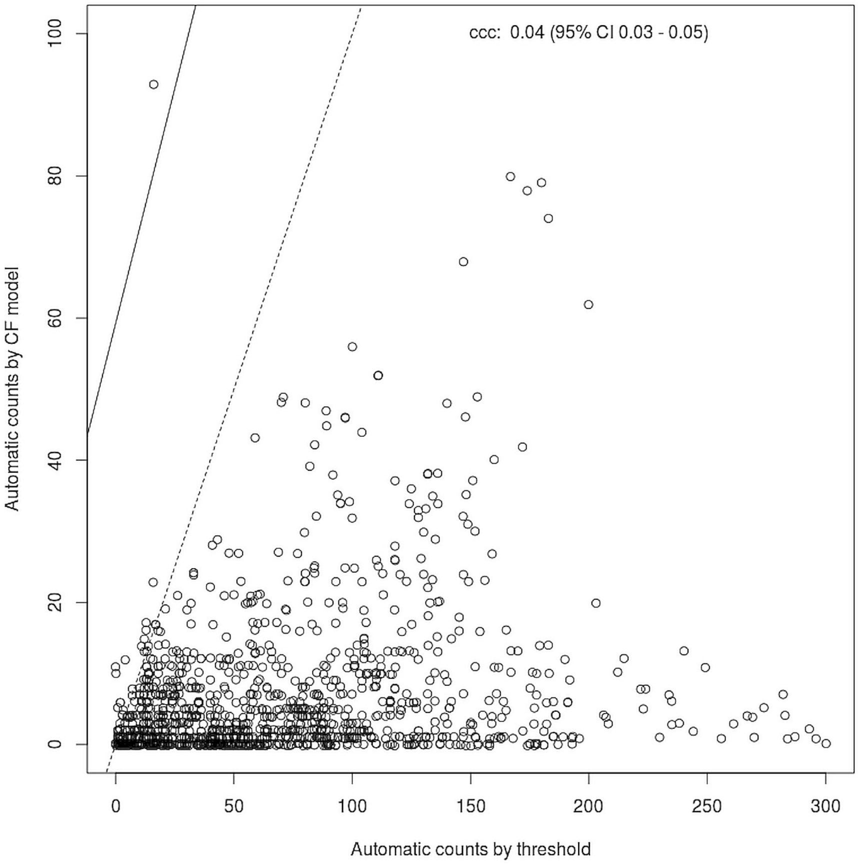


Figure 6.11: *Analysis 1*, a comparison of nest counts by different image analysis methods. Automatic counts from monitoring image stacks using classical thresholding (default intensity histogram) and the CF model (RF classifier). The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.244$ (precision), $\rho_c = 0.040$ (agreement) and $C_b = 0.164$ (accuracy).

6.6.2 *Manual-field and manual-image counts*

Manual nest counts estimated from images between two scorers and the actual numbers taken in the field counts were compared. The variability between manual-image counts by two scorers, are outlined first. The nest counts estimated from images, by two scorers were compared to analyse the variability between observers. Figure 6.12 below, shows the output results from epiR.

There was close agreement between the manual-image estimates from two scorers ($\rho_c = 0.867$), with good precision ($r = 0.891$) and accuracy ($C_b = 0.973$).

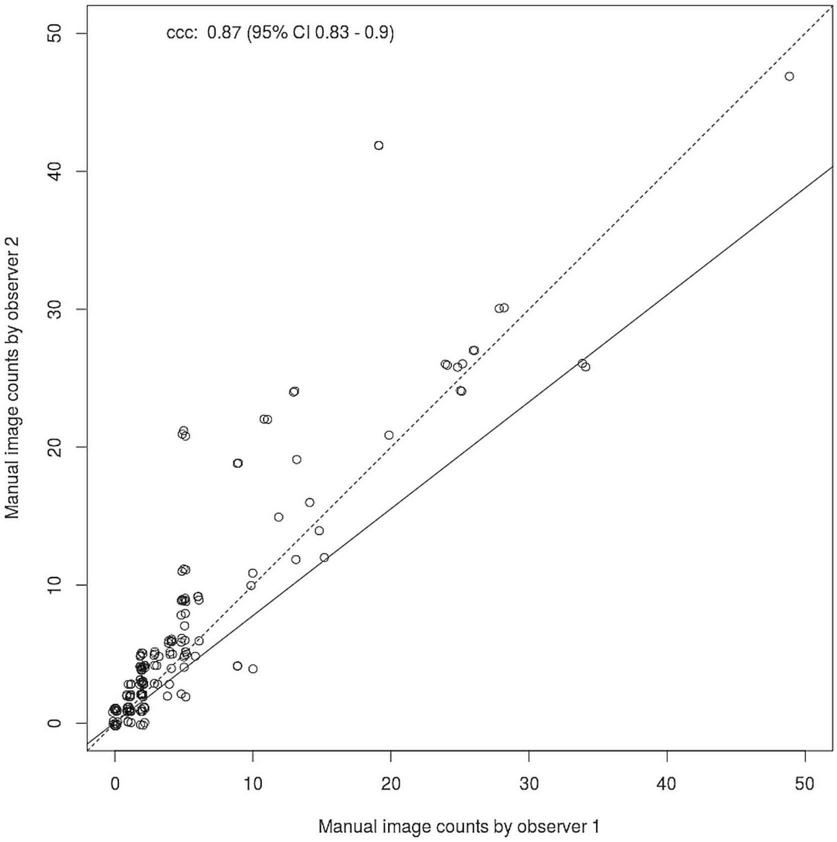


Figure 6.12: *Analysis 2*, a comparison of nest counts estimated from images by different scorers (observer 1 and 2) image analysis methods. The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.891$, $\rho_c = 0.867$ and $C_b = 0.973$.

In Figure 6.13 below, the mean estimated counts from two scorers ($n = 170$), were compared against manual-field counts. Results showed there was close agreement between the counts from two methods ($\rho_c = 0.622$), with good precision ($r = 0.641$) and accuracy ($C_b = 0.97$).

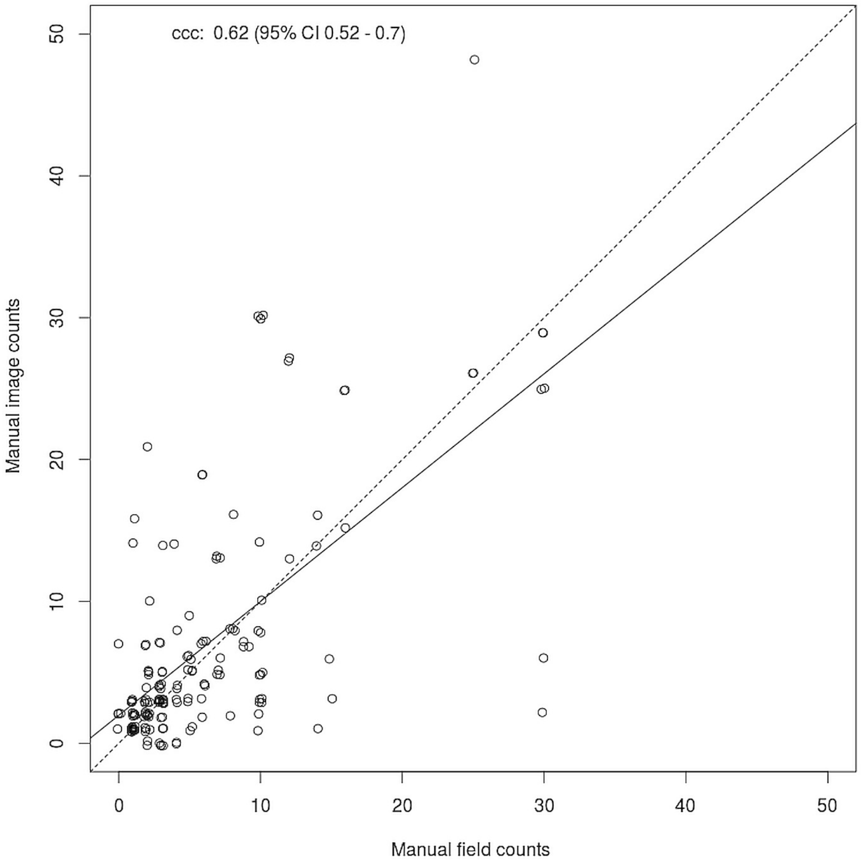


Figure 6.13: *Analysis 3*, a comparison of manual-field and manual-image (mean counts from two observers $n = 170$) nest counts. The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.641$, $\rho_c = 0.622$ and $C_b = 0.97$.

6.6.3 *Automatic and manual counts*

A comparison of nests counts estimated from images (the mean from two scorers $n = 170$) and automatic counts derived from segmentations using the CF model were analysed. Manual-image and automatic-CF counts are compared in Figure 6.14 below. Manual-field counts were compared to automatic counts derived using the CF model. These are compared in Figure 6.15, pg. 105. Both results are discussed here.

There was closer agreement between manual-field and automated-CF counts ($\rho_c = 0.738$, Figure 6.15), compared to manual-image and automated-CF counts ($\rho_c = 0.679$, Figure 6.14). The manual-image and automated-CF counts were slightly more accurate ($C_b = 0.963$, Figure 6.14) than manual-field and automatic-CF counts ($C_b = 0.891$, Figure 6.15). Both had similar values of precision (manual-image: $r = 0.705$, manual-field: $\rho_c = 0.738$).

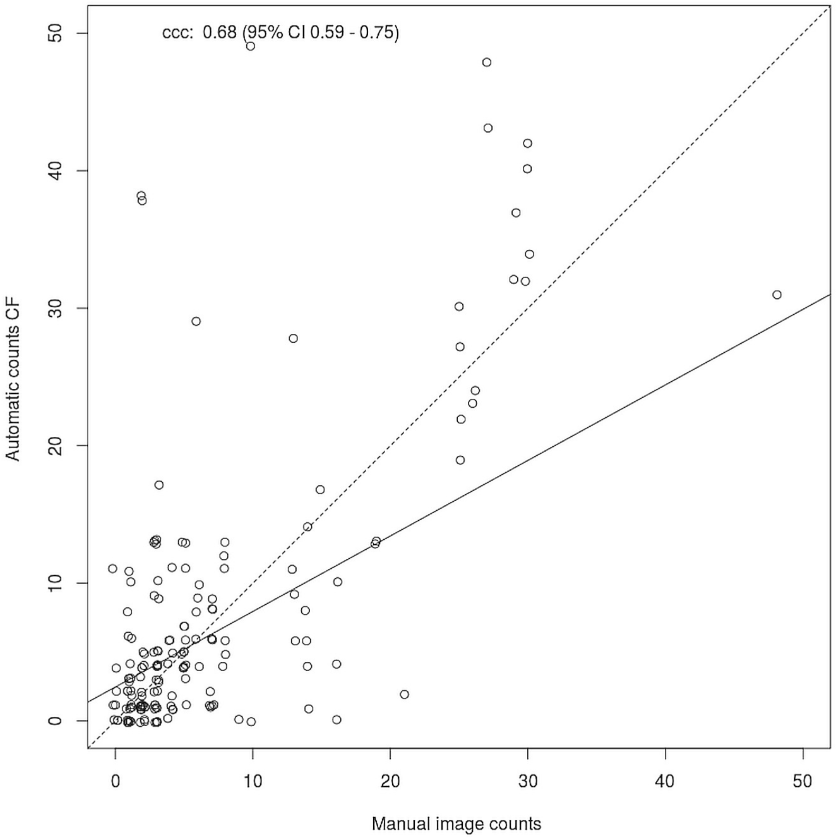


Figure 6.14: *Analysis 4*, a comparison of manual-image nests counts (the mean from two scorers $n = 170$) and automatic counts derived from the CF model. The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.705$, $\rho_c = 0.679$ and $C_b = 0.963$.

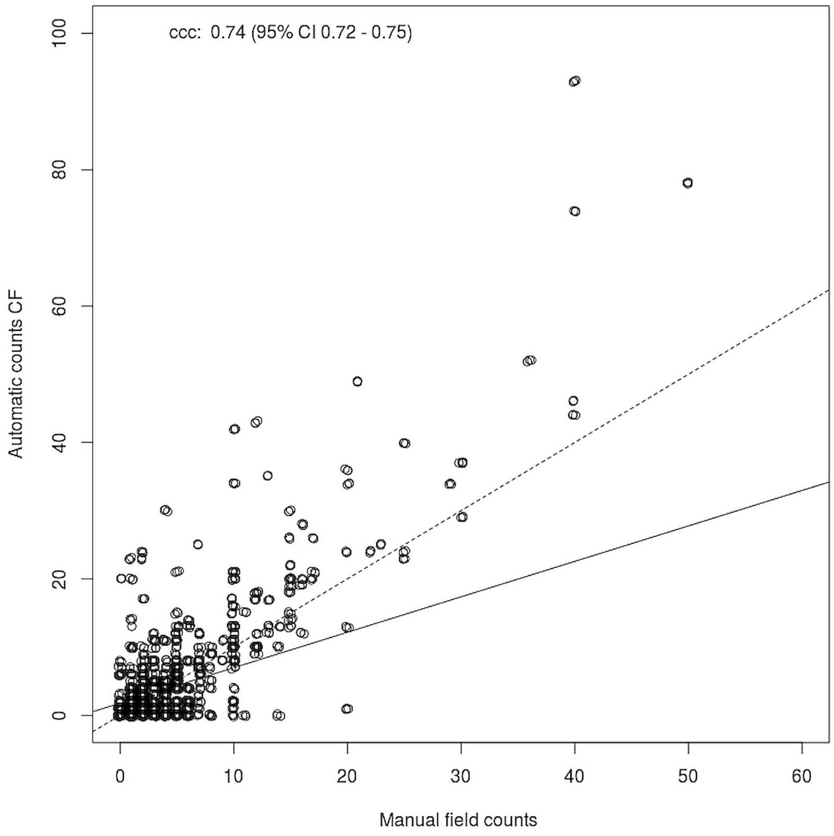


Figure 6.15: *Analysis 5*, a comparison of manual-field and automatic nest counts derived from segmentations using the CF model (median of three image collections $n = 520$). The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.828$, $\rho_c = 0.738$ and $C_b = 0.891$

.

6.7 NUMBER OF ACTIVE NESTS BY SITE AND YEAR

The number of active nests counted by three methods, are organised by year, in Figures 6.16, 6.17 and 6.18 (pg.108–110). For each site, the changes in the number of active nests over time are shown. Methods indicated similar trends. Yearly mean values included error bars (showing the standard error of the means). Field samples varied between sites and over time. This is summarised in Table 6.7 below, for reference.

Different methods were based on more, *or less* data samples (i.e. collections of near-replica images). Increased errors and/or variability, were associated with the number of monitoring samples collected. For instance, see Figure 6.16 (pg. 108). The variability of results for site 1, 2014 where higher than other years, since there were minimal data collected. Samples were collected over three monitoring days, from three sites, and four grids (i.e. $n_m = 3$ and $n_{S1} = 36$). Imaging methods used three image collections. Therefore, the total image data used in analysis was $n_{total} = 108$; while the total number of data used in manual count analysis was $n = 36$.

6.7.1 *Mt. Tiger*

The trends in active nests from data collected over five years (2010–2014) on Mt. Tiger are shown in Figure 6.16, pg. 108. There was a sharp decline in the mean number of active nests between the years 2010–2011. There were minimal changes between the years 2011–2013. An increase in the mean number of active nests was indicated in 2014 (n_{S1}= 12). Similar trends were displayed by all methods. The automatic counts derived from image segmentation using the CF model (red square), on average produced slightly higher mean counts than manual-image (green circle) and manual-field counts (blue triangle).

6.7.2 *Mt. Parihaka*

The trends in active nests from data collected over five years (2010–2014) on Mt. Parihaka are shown in Figure 6.17, pg. 109. There was a moderate decline in the mean number of active nests between the years 2010–2012. There was a gradual increase the mean number of active nests recorded between the years 2012–2014. Similar trends were displayed by all methods. The automatic counts derived from image segmentation using the CF model (red square), on average produced slightly higher mean counts than manual-image counts (blue triangle) but lower than manual- field counts (green circle).

Table 6.7: The number of yearly samples taken. The total number of monitoring days (n_m), for three sites (S1, S2 and S3) and four grids. In 2014 n_m = 3 therefore the total number of samples for analysis was 12.

		Samples		
Year	Method	n_{S1}	n_{S3}	n_{S3}
2010	ac	126	130	
	mic	14	13	
	mfc	64	64	
2011	ac	112	115	89
	mic	13	14	12
	mfc	64	64	64
2012	ac	114	116	74
	mic	13	21	5
	mfc	52	52	52
2013	ac	105	105	90
	mic	17	14	8
	mfc	40	40	40
2014	ac	36	36	36
	mic	10	7	7
	mfc	12	12	12
$n_{ac} =$	1284			
$n_{mic} =$	170			
$n_{mfc} =$	632			

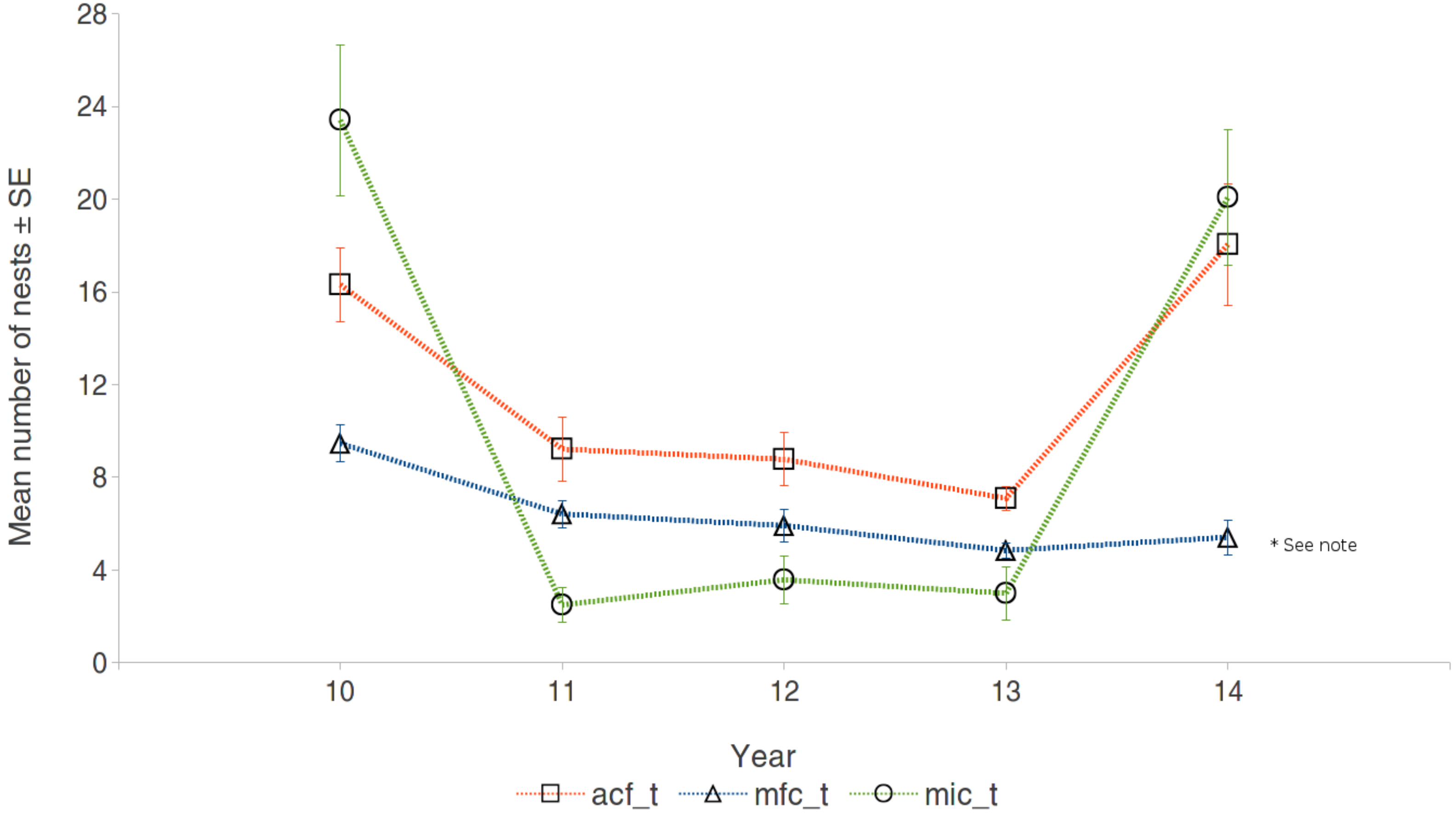


Figure 6.16: *Site 1*, yearly mean nest counts ± SE (error bars) by three methods. Automatically derived segmentations from CF model (red square) - $\mathrm{act\_t}, n = 493$, manual-field counts (blue triangle) - $\mathrm{mfc\_t}, n = 232$ and manual-image counts (green circle) - $\mathrm{mic\_t}, n = 67$.
**Note*: Data for 2014 were collected over three days ($n_{days} = 3$). Results indicate instability; the $\mathrm{mfc\_t}$ outlier is likely due to experimental error associated with the small sample size.

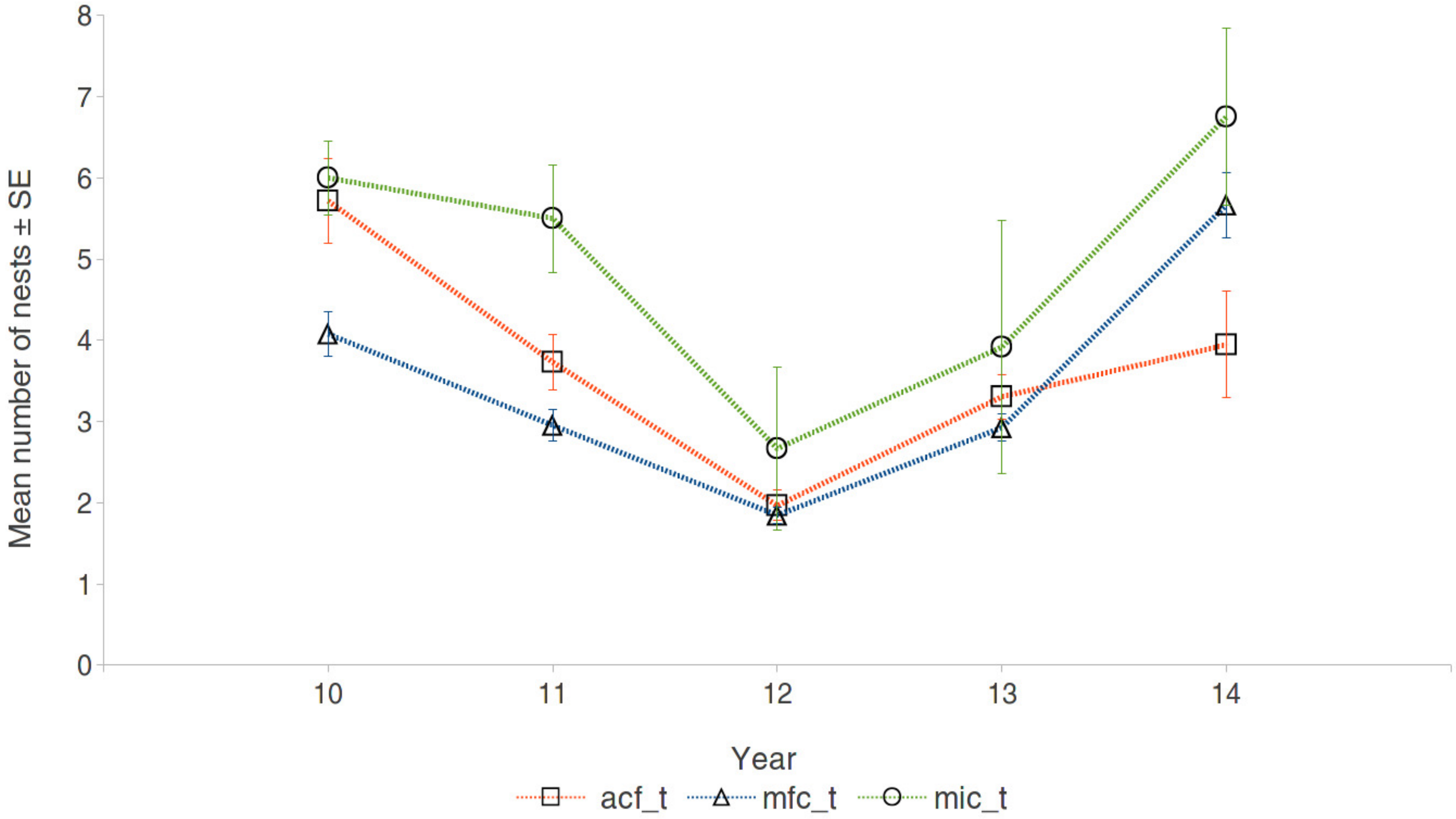


Figure 6.17: *Site 2*, yearly mean nest counts ± SE (error bars) by three methods. Automatically derived segmentations from CF model (red square) - $act\_t, n = 502$, manual-field counts (blue triangle) - $mfc\_t, n = 232$ and manual-image counts (green circle) - $mic\_t, n = 69$.

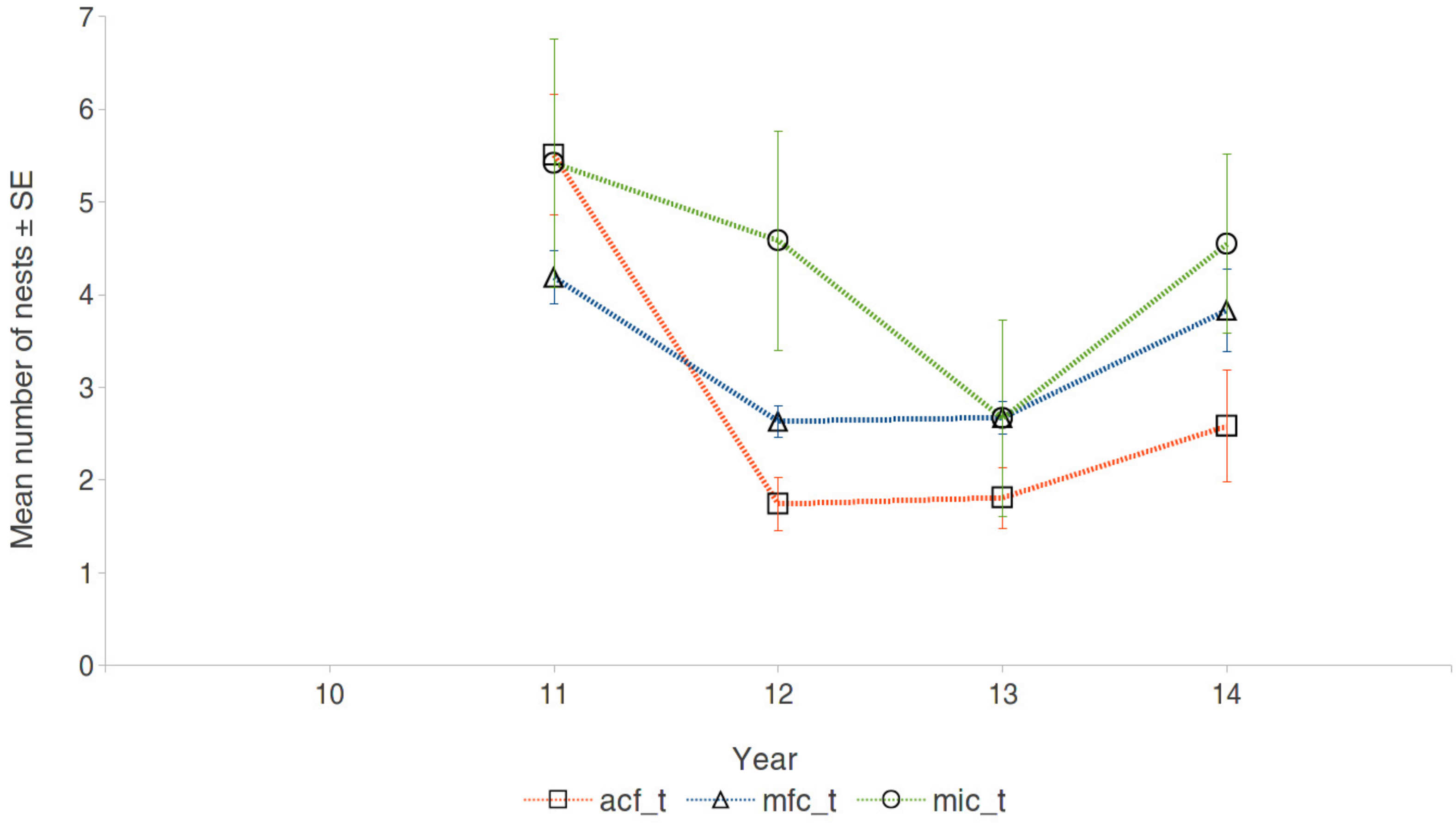


Figure 6.18: *Site 3*, yearly mean nest counts ± SE (error bars) by three methods. Automatically derived segmentations from CF model (red square) - $\mathtt{act\_t}, n = 289$, manual-field counts (blue triangle) - $\mathtt{mfc\_t}, n = 168$ and manual-image counts (green circle) - $\mathtt{mic\_t}, n = 34$.

6.7.3 *Memorial Drive*

The trends in active nests from data collected over four years (2011–2014) on Memorial Drive are shown in Figure 6.18 above (pg. 110). There was a moderate decline in the mean number of active nests between the years 2011–2013. A gradual increase the mean number of active nests was recorded in 2014. Similar trends were displayed by all methods. The automatic counts derived from image segmentation using the CF model (red square), on average produced slightly lower mean counts than manual-image (blue triangle) and manual-field counts (green circle).

7 DISCUSSIONS

SYNOPSIS

If the populations of solitary ground nesting bees can be estimated by the number of active nests in a community, then the number of active nests can be manually counted. Over time, manual counts can provide a measure of the changes in populations within a community. This method is comparatively straightforward and easy to replicate. Given this, it is reasonable to expect the number of active nests can also be counted from digital images of actual nests, in much the same manner as manual estimations. If the number of active nests in an image can be counted, it is therefore also feasible to use image analysis to automatically identify the objects in the images that represent active nests. This reasoning was tested by using a practical monitoring programme. It was designed to collect manual nest counts and images of active nests for comparative analyses and proof of concept. Monitoring was conducted over five years, at three communities of native bees in Whangarei (New Zealand). A total of 1896 images were collected, representing 158 monitoring days. They were processed and used in a comparative analyses against manual field nest counts. This chapter discusses the design, implementation and final performance evaluation of the image-centric nest monitoring system.

7.1 RESEARCH OVERVIEW

Background context, study rationale, design methodology and research outcomes.

This study was based on the assumptions that, (1) the number of active nests could provide a proxy for populations and, (2) it was possible to design methods to reliably count the number of active nests at communities of native bees, over space and time, (3) that digital images could be used in the place of manual visual counts and (4) it was possible to process digital images to reliably count active nests. The evidence supporting the central ecological hypothesis that the *active nests of solitary ground nesting bees can provide a good proxy for populations* was taken from several studies [16, 64, 65, 66]. This could have been tested in the field. For example, emergence traps, nest core samples, or mark release recapture methods could have been implemented [54]. Sampling could have been conducted alongside manual nest counts and digital image acquisition. This could have provided sufficient data to confirm the *active nest population proxy* hypothesis. However, the techniques were beyond the scope of this research. Field sampling methods were designed for a comparative analyses, between manual and image-centric nest counts. Therefore, exactly the same nests were monitored over time.

An image-based active nest counting pipeline could have been developed using pre-existing digital image samples [21, 29]. Some of the preliminary aspects of the system design, did not depend on actual monitoring; manual nest counts or new image collections. However, it was essential to test and apply the monitoring system. This was to demonstrate the potential of the system, as a viable tool to aid field research. Many promising tools developed for ecological research, are not used in practical applications [84, 85, 87]. The reasons for this are not clear. But, as a consequence, it was important to evaluate all aspects of the monitoring system in the field. The image-centric design was founded on natural history observations of communities of native bees [21, 29]. Thousands of hours of empirical data, contributed towards an understanding of the challenges involved with monitoring bees. Also, a consideration for the types of technologies required to increase scientific understanding of native bees and their communities. Therefore, the field methods were designed to test the validity of the image-centric approach. The remaining discussions in this chapter are concerned with design aspects of the image-centric monitoring system; as it was applied, to measure the number of active nests.

7.2 ACTIVE NESTS

Design of field methods, images of active nests and the alternative approaches?

There were other *types* of images that could have been used for monitoring. Some image-types could be used to reflect the population changes over time. Other images, could be used to describe the species diversity of native bees within selected communities. This thesis has focused on the monitoring images of active nests. But, there were a range of image types collected during field monitoring. This was to test image acquisition techniques and image analysis procedures. This included capturing images of insects foraging on single flower heads (e.g. Figure 7.1 (a)–(b)), insect clusters in flight around shrubs (e.g. Figure 7.2 (b)), insect sweep net collections (e.g. Figure 7.2), active nests captured with and without a grid quadrant (e.g. Figure 4.3 and Figure 4.4). Active nests were the basis for the image-centric method presented in this thesis. The reasons are summarised:

1. The images of active nests were easy to *capture*.

2. The image area was easy to *regulate*.

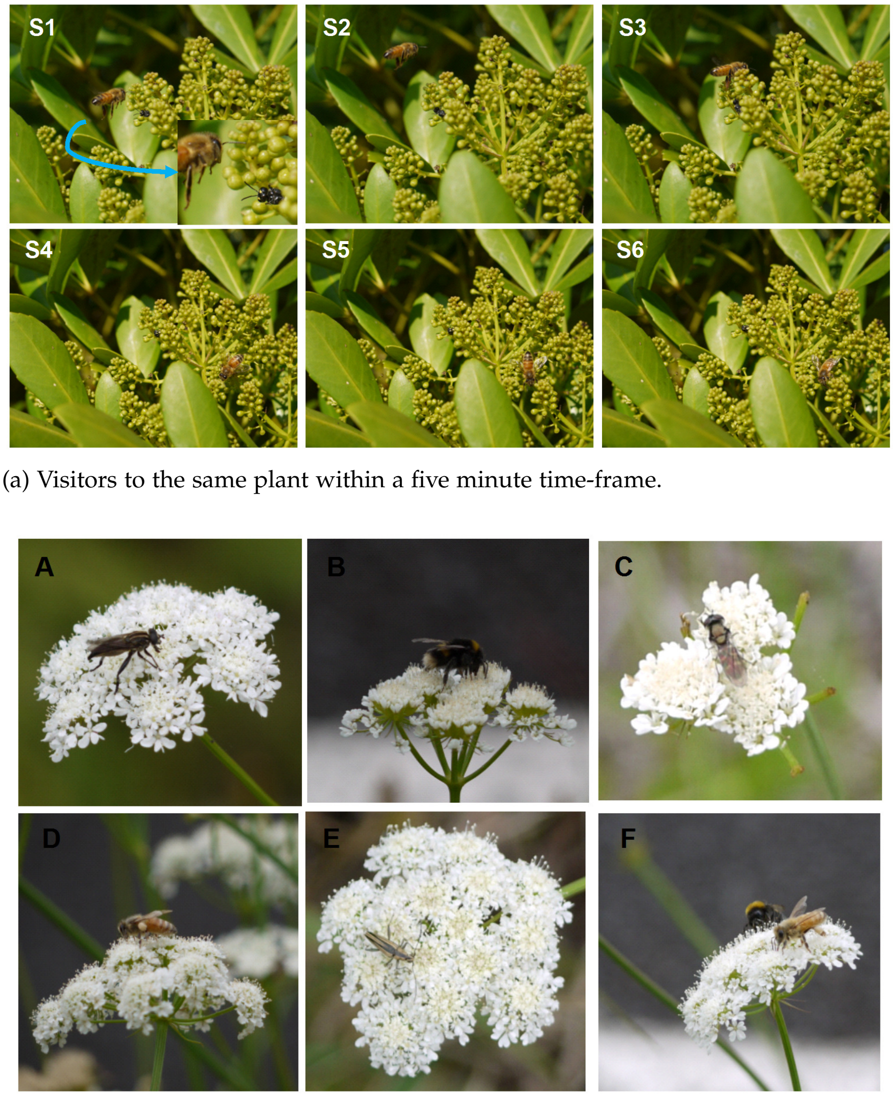
3. The image analyses required only *two objects* to be segmented.

4. The results from analysis provided nest counts.

5. Nest counts could be *directly* used to measure populations.

Nest images were easy to capture. Particularly compared to acquiring photographs of moving insects; as demonstrated below in Figure 7.1 (b). Image acquisition was easier to control for active nest data. This was achieved by using a standard of measure. The grids were placed over areas of active nests. They helped to determine the focal boundaries of digital images. The imaging methods required to segment active nest images, were not as complicated as they were for other images. Some of the images would have required several target features to be segmented. For example, in Figure 7.1 (b) below, six different insects are shown. They are foraging on single flower heads. The images were captured within a set time-frame. They equate to survey-type methods and could be used for biodiversity sampling. The imaging pipeline for these types of images, would have involved identifying at least six types of insects. Therefore, the segmentation task would have been to partition the images into six key objects. In contrast to this, there were only two categories in the images of active nests. It was necessary to segment areas of *active nests* from all other *backgrounds*. Furthermore, data from image analyses based on the number of active nests in the images, *directly relates* to an estimate of the populations of bees. Interpretation of results were straightforward; and achieved with minimal post-processing and statistical analyses. Thus, compared to other image-types, the images of active nests were much easier to capture, process and analyse.

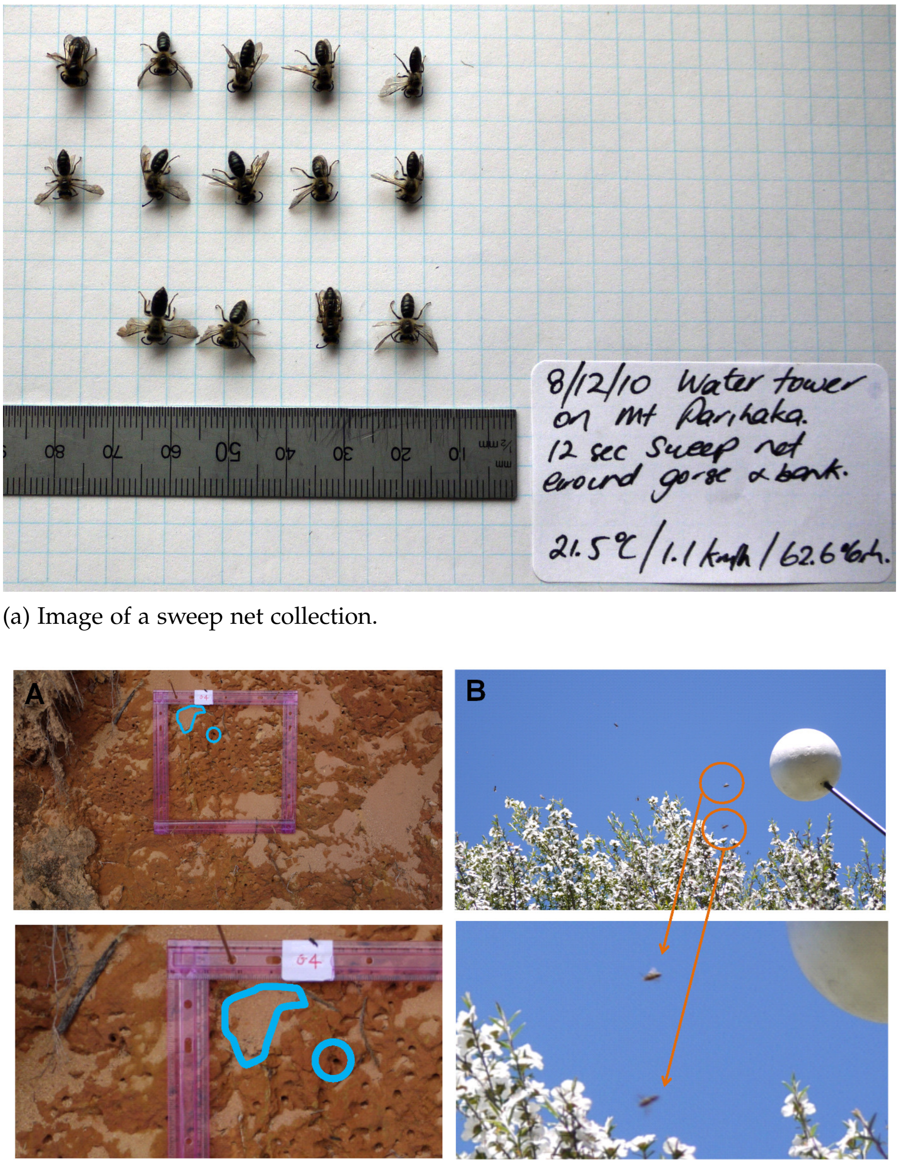
Finally, the field method for counting active nests were not overly complicated. Surveys were conducted each year around September. Monitoring was initiated when there was clear evidence the bees were emerging (e.g. either by signs of nest constructions or observations of bees in flight). At each monitoring location, the number of nests were counted. Active nests were identified by using evidence of soil excavations, or entry holes; or both. The nest counts taken at the start of each season were important. The nest entry holes were more defined. They were easier to count. Because of this, they were more reliable than counts taken towards the peak of active seasons. Mid-season, entry holes to active nests could become obscured by masses of soil from nest excavations. It was almost impossible to determine where the entry holes were, in some cases. This was a particular issue at Mt. Tiger. For an example, refer to Figure 4.3 (Chapter 4, pg.46). The nesting community at Mt. Tiger was established along a roadside bank. Because of the structure and angle of the bank, the soil from nesting bees rapidly accumulated in some pockets. Soil completely covered some nest entrances. In contrast to Mt. Tiger, there were communities of bees established along a flat area of ground at Mt. Parihaka. The horizontal nests were much easier to count; at the start and throughout the season. This was because, the mounds of white clay soil indicating nesting bees, were easily separated from one another; and



(a) Visitors to the same plant within a five minute time-frame.

(b) All the visitors to plants in a meter square.

Figure 7.1: Image sequences of foraging bees and other insects; focusing on (a) the same plant in a set time-frame and on, (b) all plants and insects in a set area.



(a) Image of a sweep net collection.

(b) Grids for active nests (left) and a sphere for bees in flight (right).

Figure 7.2: Images of (a) sweep net collections and (b) standard of measures for monitoring – grids for active nests (left) and a sphere for bees in flight (right).

the ground surface vegetation. An example is shown in Figure 4.4 (Chapter 4, pg.46).

7.3 MONITORING IMAGES

Image formats, quality and variability.

Digital image formats and acquisition techniques, are important aspects of an imaging system design. The decisions regarding the digital camera hardware and image format in this research were based on costs. An off-the-shelf DSLR camera[1] was used for acquisition of nest images. They were saved in compressed JPEG format. Off-the-shelf digital cameras and encrypted image formats are not recommended for biological image analysis [112]. However, they are used in some environmental imaging, and remote sensing applications [142, 143, 144, 148, 149, 151].

7.3.1 *Image format*

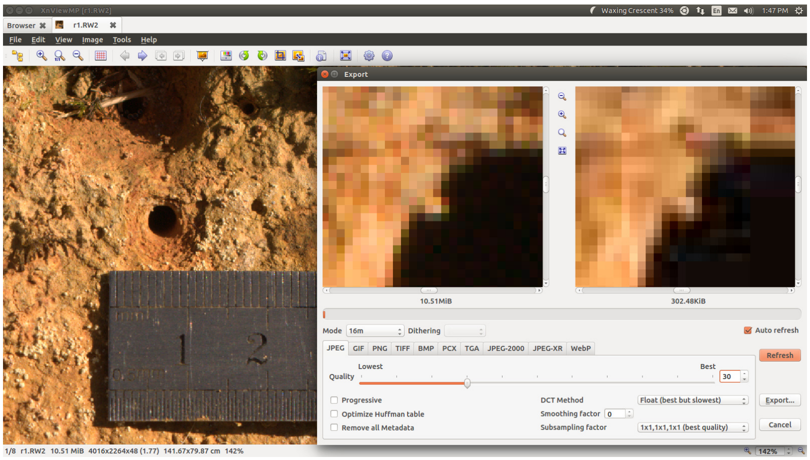
There are no agreed standards for raw files. Therefore, proprietary platform dependent software is often bundled with off-the-shelf digital cameras. This is to enable file processing. Silkypix™ Software was bundled with the DSLR camera used in field monitoring. The software was platform dependent, so use was restricted. Open-source software helped to mitigate proprietary format issues; but with variable success. Raw files were not easily imported into FIJI or XnView. During early tests, the added file work was resource intensive and time consuming. Thus, raw files were not collected during monitoring.

Nonetheless, the quality of image data can impact the reliability of image processing. At least where biological image analysis is concerned, high resolution raw images are preferred. However, biological images are normally captured under laboratory conditions. Also, memory capacity is not usually an issue. On-board camera memory was a consideration for field collections in this research. Because the field camera had a limited amount of memory, compressed high quality JPEG formats were more practical. They were also more cost effective. Some studies have shown image compression is not always an issue. Paola and Schowengerdt [151] for example, tested three different classification scenarios. They found that, high quality classifications could still be achieved with a compression ratio (CR) of 10:1 [151].

The CR of the field camera was closely examined. This was to determine the qualitative affects on monitoring images. Active bee nests are ill defined. Therefore, a close-up image of an associated insect, the common New Zealand tiger beetle–*Cicindela tuberculata* (Coleoptera: Carabidae), was used in the evaluations. The image was converted to a JPEG file using XnView software. The file had a CR of 34.71:1. After the image was compressed, the affects were visually checked. The

[1] Panasonic DMC-G1

close-up view shows there are minimal affects from the encryption. This can be seen in Figure 7.3 below, as slight *blocking*. When images were converted on-camera, the CRs were substantially lower, at around 2.75:1. This level of compression was less than 10:1 and well within the boundaries considered acceptable for image classification applications [147, 149, 151].



Mode 16m

Dithering

☑ Auto refresh

Quality 30

□ Progressive

□ Optimize Huffman table

□ Remove all Metadata

DCT Method Float (best but slowest)

Smoothing factor 0

Subsampling factor 1x1,1x1,1x1 (best quality)

1/8 r1.RW2 10.51 MiB 4016x2264x48 (1.77) 141.67x79.87 cm 142%

Figure 7.3: A raw image of a tiger beetle's nest (background image) and a close-up example of file compression using XnView.

7.3.2 *Outdoor images*

A number of techniques can be used to help standardise image acquisition and to reduce the affects of outdoor variations [138]. Several techniques were tested during the prototyping stages of the monitoring system design. For example, flood lights were tested on monitoring nests. This was to determine if the extra illumination would improve image quality. However, the challenges associated with outdoor imaging methods are fundamental constraints. They were not easily overcome in this research. Because the images of active nests were acquired outdoors, they were variable and complex. It was not possible to control the quality of images. For example, the lighting conditions changed dramatically on cloudy days, sometimes within seconds. On windy days, objects could be blown across the nest sites, and the field of view of the camera during image acquisition. This is common to many outdoor imaging applications, as a review of current literature confirmed [138, 143, 144]. There are few practical methods that can be used to improve the quality of natural outdoor images. But, there are some novel approaches (e.g. see the design by Burks [138]). A number of other studies reported the problems associated with analysis of complex natural images could be mitigated by using RFs for image classifications [143, 144]. Research showed the performance of the classifier was good, even when compressed, outdoor images were acquired using off-the-shelf digital cameras [143, 144].

The quality of monitoring images was a key issue. The constraints had an critical impact on the design of the image-centric monitoring system. A closer examination of nest images are outlined below.

7.3.3 *Examining nest images*

Biomedical imaging package FIJI was the selected tool used for monitoring image analysis. The reasons for this have been mentioned previously. However, because FIJI is open source the methodology used throughout this thesis can be easily replicated. FIJI macro scripts are included in Appendix E (raw data are attached in Appendix F). Also, a range of resources are available online for development of FIJI and ImageJ scripts[2]. Nest images were examined in FIJI. A test stack was compiled using representative nest images. Images were selected from each site-grid to reflect the range and variation typically encountered. The test stack shown below in Figure 7.4, or a single slice from the stack, were used to test a range of processing techniques. These are examined over the next few sections.

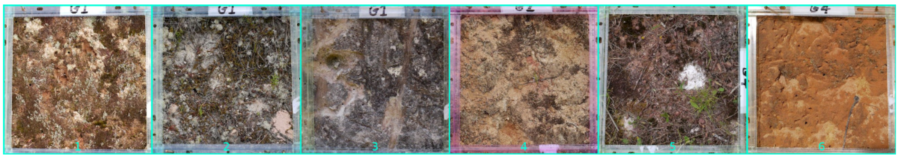


Figure 7.4: Test stack of representative nest images in raw RGB format. Slice 1 = Mt. Tiger (bank grid 1), slice 2 = Memorial Drive (bank grid 1), slice 3 = Mt. Parihaka (bank grid 1), slice 4 = Mt. Tiger (bank grid 2), slice 5 = Mt. Parihaka (horizontal ground grid 4), slice 6 = Mt. Tiger (bank grid 4).

7.3.4 *Preliminary investigations*

In most analysis, workflows images are prepared by using *contrast enhancement* or *histogram equalisation* operations. The operations are generally called image normalisations. A single image of a horizontal ground nest was examined in FIJI. Different contrast enhancement settings were applied to the image, to investigate the most appropriate levels for monitoring images. The *enhance contrast toolbox* was used in tests. There were four different functions tested: 1) the percentage of *saturated pixels*, 2) equalise histogram, 3) *process all* and 4) stack equalise histogram. The test stack was processed using three different values of saturated pixels: 0.4%, 40% and 90%. The changes in contrast can be visually inspected in Figure 7.5 (b). The third scheme was selected for pre-processing monitoring images. The number of

[2] *ImageJ*-http://imagej.nih.gov/ij/

saturated pixels was set to 0.4%. This setting was useful for improving the visual quality of nest images on the LCD monitor used in this research. Histogram equalisation was not used for nest images. The operation is not recommended for quantitative tasks, since image data are fundamentally changed [187].

7.4 SEGMENTATION METHODS

Segmentation challenges

Good segmentation techniques are those where, (1) pixels in the same category have similar values and form connected regions or (2) neighbouring pixels, which are in different categories have dissimilar values [112, 188]. The primary aim of all segmentation techniques are to quantify aspects of image data [112, 188]. Preferably by using reproducible and objective techniques and with operations that have some capacity to *generalise* over a given range of image data variability. The performance of any segmentation method depends on several factors. Firstly, on the original image content and quality. Secondly, on the specific application constraints and characteristics. Thirdly, on the intended use of the information extracted from images. To examine these issues further, three segmentation techniques were tested on a sample of representative nest images. A classical thresholding method; an edge and region-based method. These tests are discussed in greater detail in the sections below.

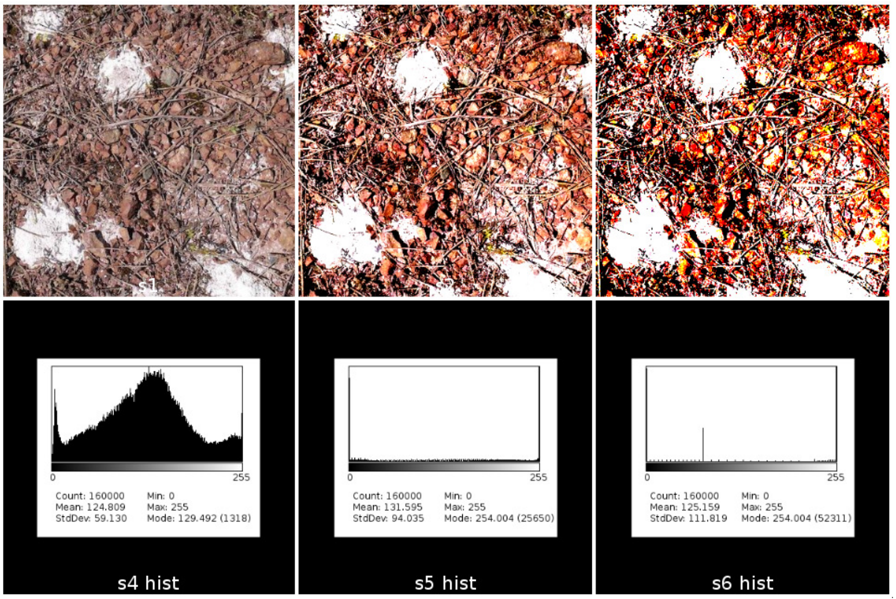
7.4.1 *Thresholding by intensity*

Intensity based thresholding methods produce straightforward segmentations [188, 189]. They are simple, direct and easily programmed. If there are foreground objects or image features that are *defined by intensity*, then threshold procedures can outperform other methods.

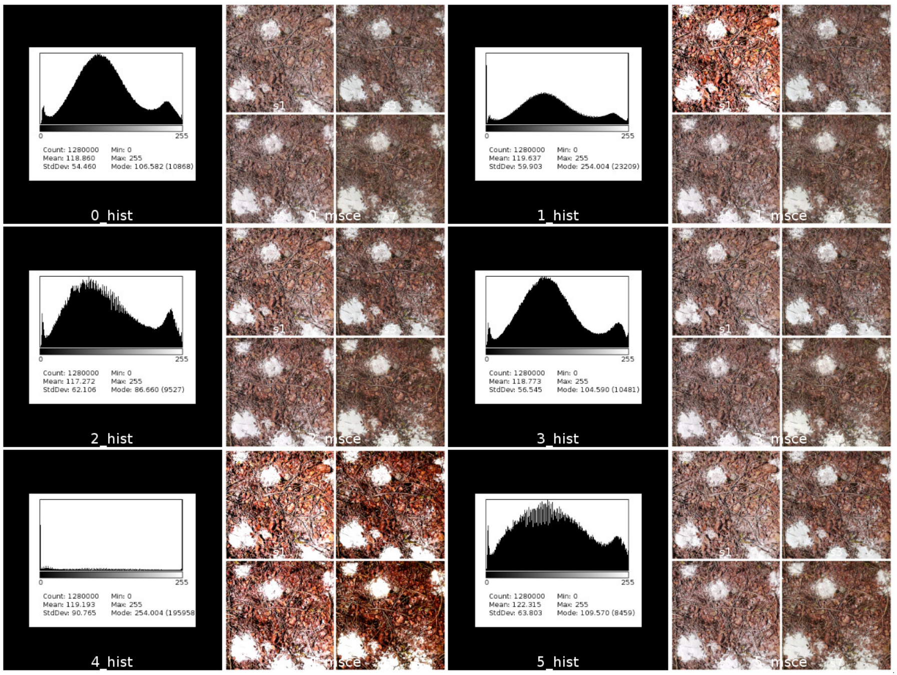
Suitable threshold levels were tested using slice 5 of the test stack. FIJI Auto threshold Try All[3] [190] function was selected. This was to determine which methods best suited the nest image. The active nest in image slice 5, is visually noticeable. The white clay soil, indicates at least one or two active nests. The intensity of soil means the image is relatively easy to make binary. The output stack results for automatic thresholds on slice 5, are shown in Figure 7.6 (a) below.

The results demonstrate that from 16 possible schemes, *Minimum* returned the highest quality segmentation. In this case, the image was adequately segmented *before* any post-processing operations were applied. When post-processing operators were applied, the results confirmed an automatic threshold method using the *Minimum* scheme would produce satisfactory image segmentations. At least for horizontal ground nests. These are shown in Figure 7.7 (a)–(e) (pg. 123). However, when the auto threshold Try All function was applied to

[3] *Try All*- http://fiji.sc/Auto_Threshold

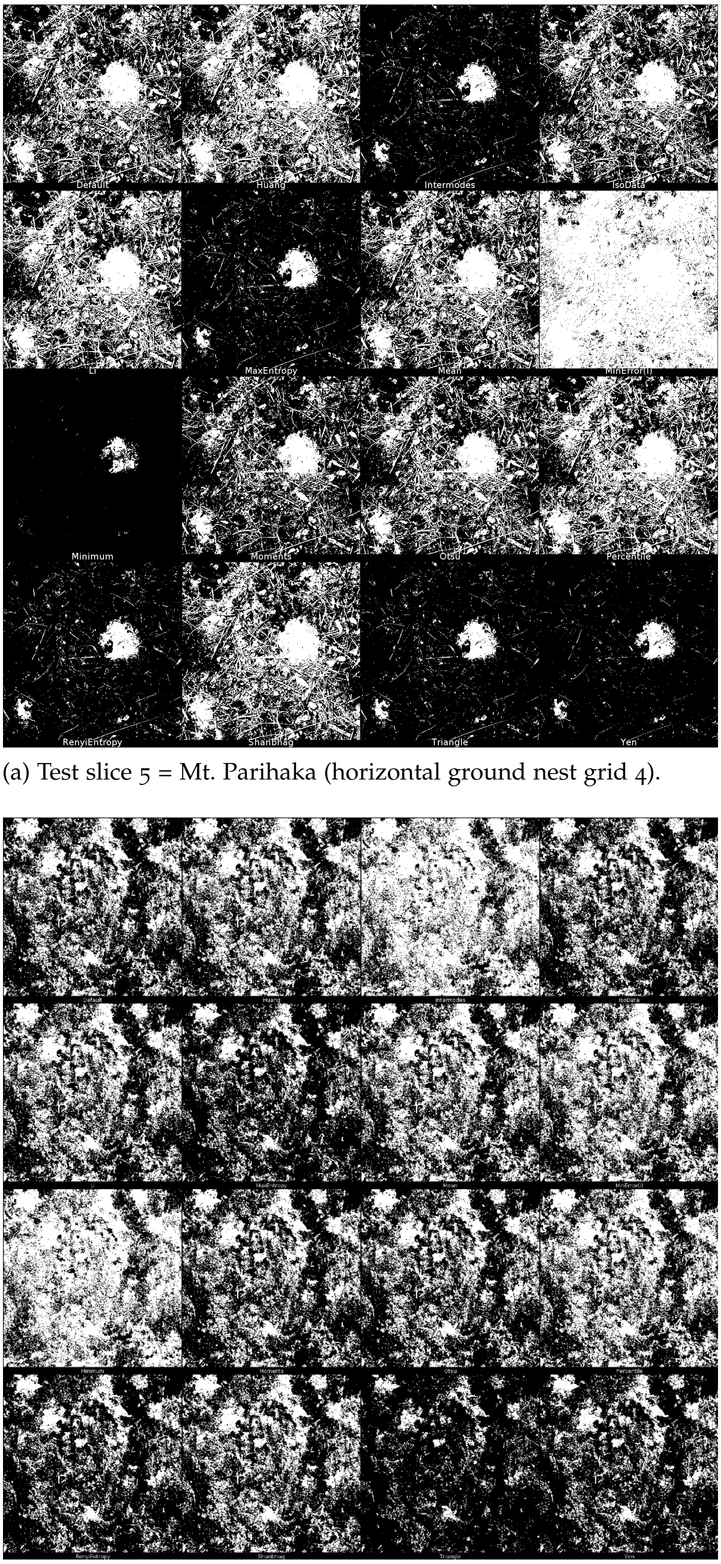


(a) Contrast stretch on horizontal image slice.



(b) *Enhance contrast* pre-processing tests on horizontal image slice.

Figure 7.5: The image of a horizontal nest was used to test the affects of (a) contrast stretching using 0.4%, 40% and 90% saturated pixels, and (b) Enhance contrast toolbox stack functions. Scheme 3 was used for monitoring images. The number of saturated pixels was set to 0.4% and the *Process all slices* function was checked.

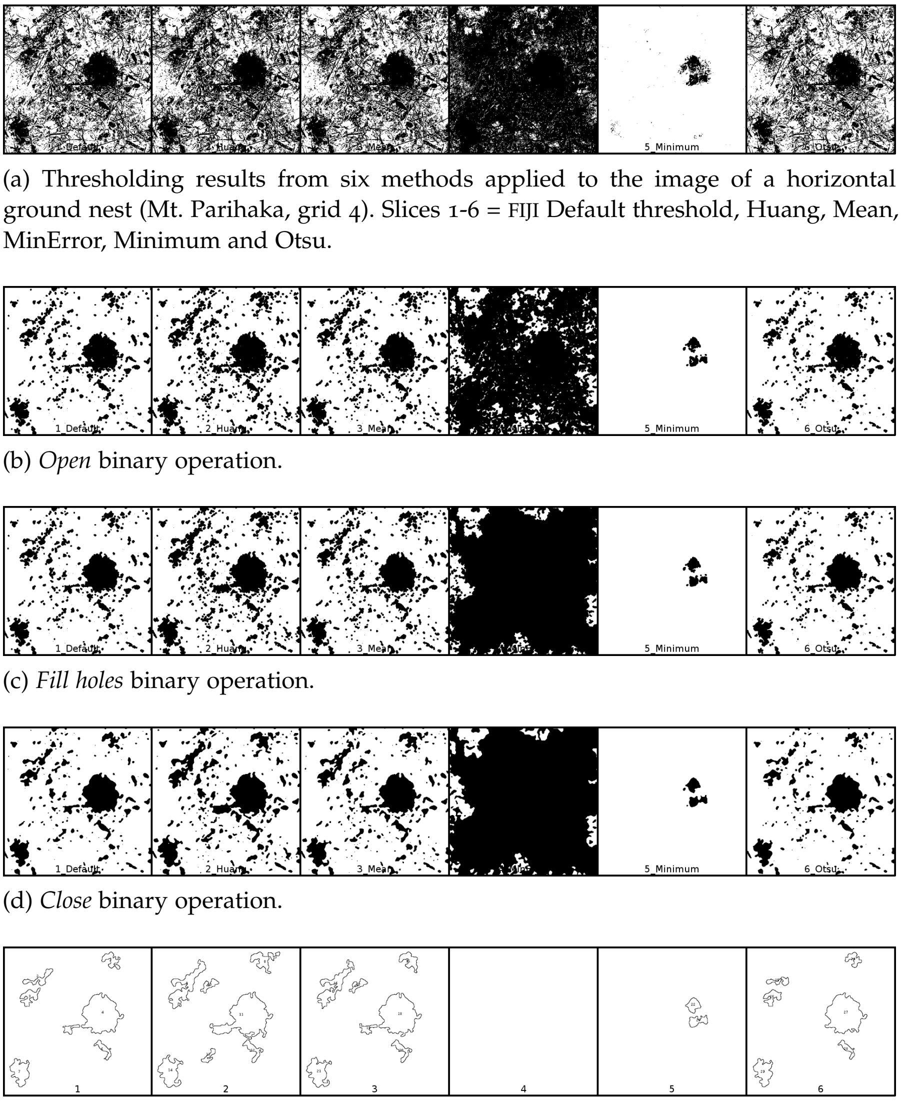


(a) Test slice 5 = Mt. Parihaka (horizontal ground nest grid 4).

(b) Test slice 1 = Mt. Tiger (bank nest grid 1).

Figure 7.6: Auto thresholds for monitoring images of (a) a horizontal ground nest (b) a roadside bank nest.

test slice 1, the nest image was not properly segmented. This is also demonstrated in Figure 7.6 (b) (pg. 122). Test results helped to define the constraints of segmentation methods. They showed the *complete range* of monitoring images could not be segmented using intensity-based threshold levels. They also confirmed automatic thresholds could be used to segment images of horizontal ground nests. Therefore, they would be suitable for segmenting 1/6 th of the monitoring images collected.



(a) Thresholding results from six methods applied to the image of a horizontal ground nest (Mt. Parihaka, grid 4). Slices 1-6 = FIJI Default threshold, Huang, Mean, MinError, Minimum and Otsu.

(b) *Open* binary operation.

(c) *Fill holes* binary operation.

(d) *Close* binary operation.

(e) *Analyze particles* plug-in: set to count all image objects between the sizes of 500–∞ pixels with a circularity morphology between 0.1–0.9.

Figure 7.7: Binary results from six thresholding methods (a) applied to a horizontal ground nest image. Common post-processing pipeline morphological operators were applied to the test image including (b) open, (c) fill holes, (d) close and (e) count particles.

7.4.2 *Canny-Deriche filtering*

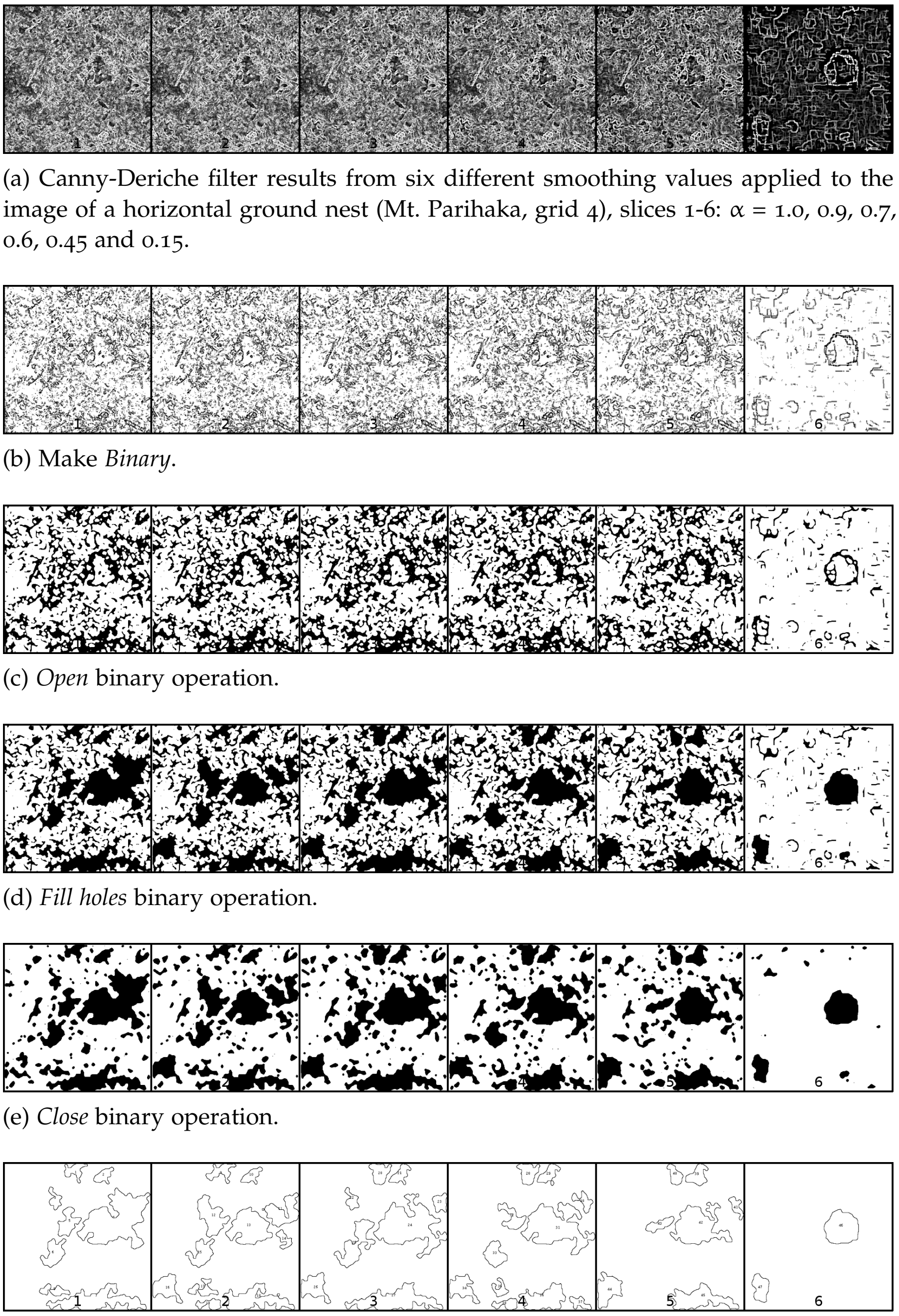
Automatic thresholds work well when intensity is a feature that can be used to identify objects. This was demonstrated in the previous example. Sometimes there are other characteristics that can be used to define images. For example, the connected structures, outlines, areas or textural qualities of objects. Since it was not possible to use automatic thresholds on all monitoring images, two alternative methods for segmentation of active nests were tested. The first of these was an edge-detection based thresholding method–*Canny-Deriche filtering* [154].

Figure 7.8 demonstrates the affects of the Canny-Deriche filter. Different smoothing factors (α) were applied to the image of a horizontal ground nest. Six smoothing factors were applied as follows: slices 1–6: $\alpha = 1.0, 0.9, 0.7, 0.6, 0.45$ and 0.15. The image results from each smoothing were compiled into a stack. This was for visual inspection and comparison. The results from the edge-based thresholds indicated the method could be used to segment horizontal ground nest images. This is demonstrated in Figures 7.8 (a)–(f). The lower smoothing values returned slightly better results. This can be seen from the final image segmentations in Figure 7.8 (e), slice 6. Overall however, edge detection did not provide any obvious additional benefits, over the straightforward thresholds previously tested.

7.4.3 *Statistical region merging*

In a final examination, a region-based method, SRM (statistical region merging) was tested on the ground nest test image [155, 156]. The results from SRM investigations are shown in Figure 7.9 (a)–(f). Six different values were selected for the maximum number of regions. The parameter is given by Q. For slices 1–6, the regions were set as follows: Q = 1, 2, 3, 6, 8 and 16. The method worked very well on test slice 5. Nonetheless, as was the case with the other methods tested, segmenting the range of monitoring images was more challenging. Ground truth labels were added to the representative stack of nest images. This was for comparison with binary results. Examination of Figure 7.10 (a), confirms most of the test images were over-segmented. They were over-segmented to degrees that could not be easily managed by post-processing operations. For instance, there were no active nests in slice 1, yet edge and region methods produced binary outputs resulting in multiple segmentations. This is demonstrated in Figures 7.10 (b)–(d) and Figures 7.10 (e)–(g) respectively.

After applying post-processing operations, a number of objects were detected that did not correspond to ground truth labels. Generally, *any* aspect of the imaging pipeline can be adjusted. This is done to accommodate image characteristics or to highlight the key



(a) Canny-Deriche filter results from six different smoothing values applied to the image of a horizontal ground nest (Mt. Parihaka, grid 4), slices 1-6: α = 1.0, 0.9, 0.7, 0.6, 0.45 and 0.15.

(b) Make *Binary*.

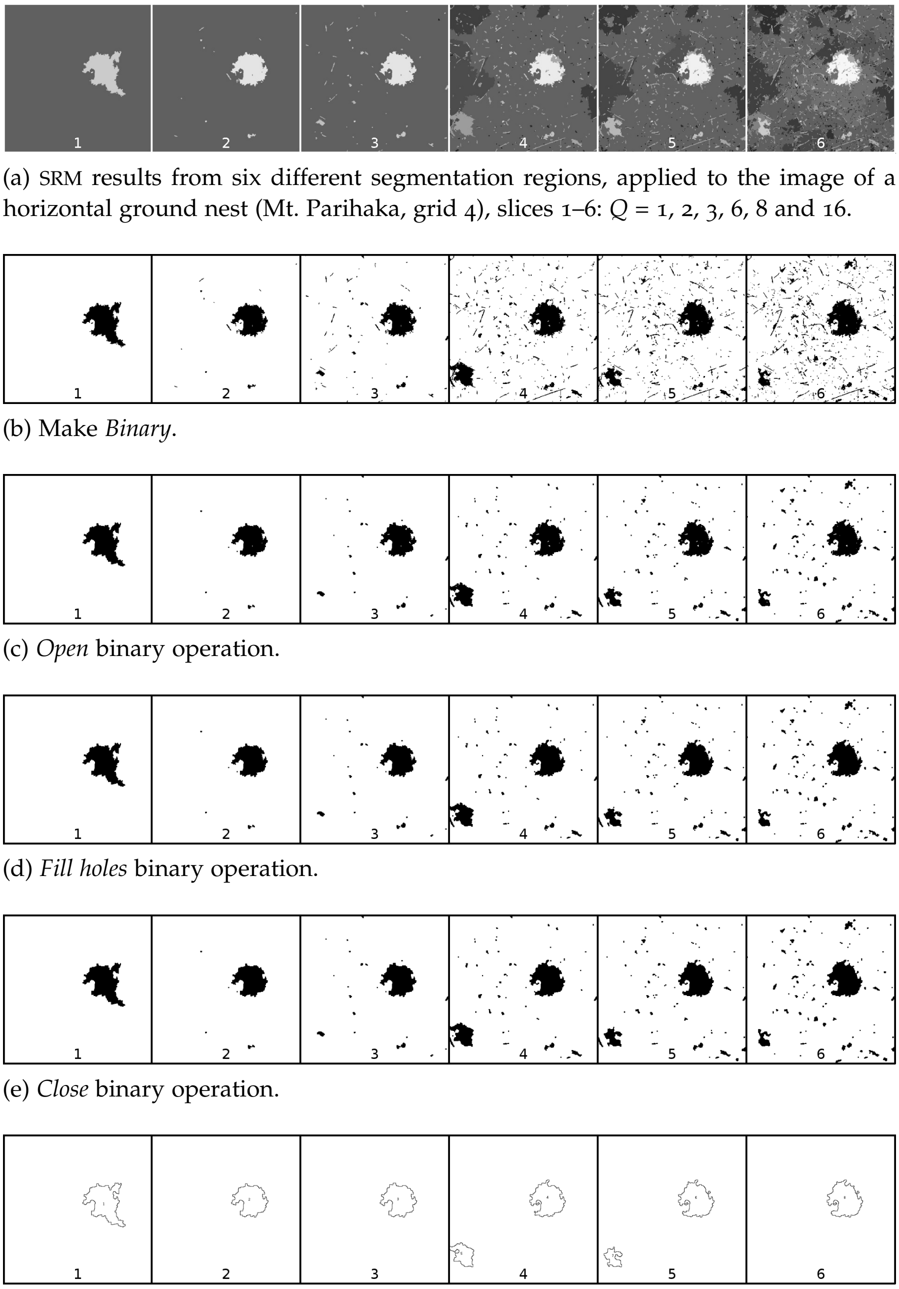
(c) *Open* binary operation.

(d) *Fill holes* binary operation.

(e) *Close* binary operation.

(f) *Analyze particles* plug-in: set to count all image objects between the sizes of 500–∞ pixels, with a circularity morphology between 0.1–0.9.

Figure 7.8: Segmentation by edge detection (a) applied to a horizontal ground nest image with a pipeline of common post-processing operations (b)–(f).



(a) SRM results from six different segmentation regions, applied to the image of a horizontal ground nest (Mt. Parihaka, grid 4), slices 1–6: Q = 1, 2, 3, 6, 8 and 16.

(b) Make *Binary*.

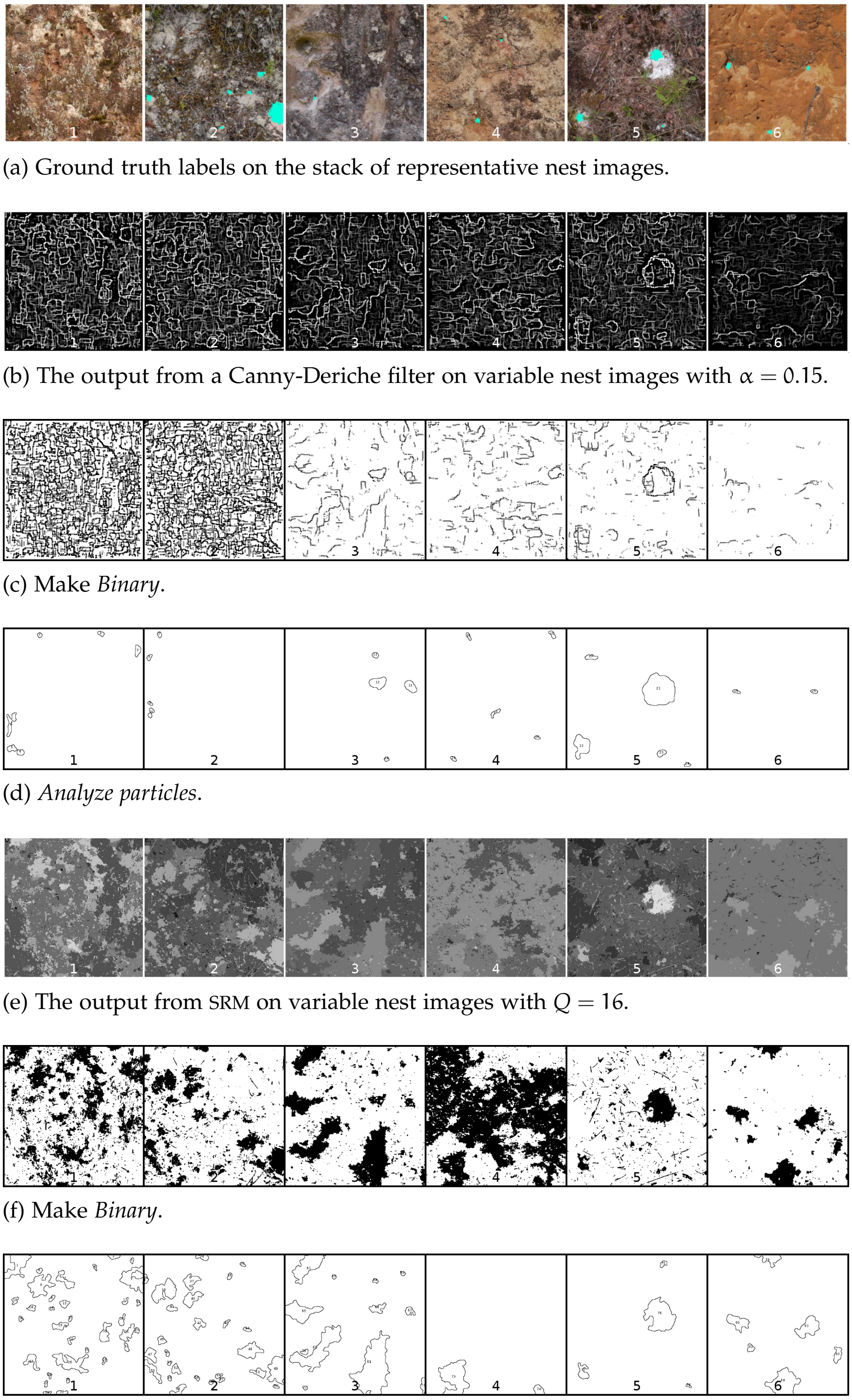
(c) *Open* binary operation.

(d) *Fill holes* binary operation.

(e) *Close* binary operation.

(f) *Analyze particles* plug-in: set to count all image objects between the sizes of 100-∞ pixels, with a circularity morphology between 0.1-0.9.

Figure 7.9: Segmentation by SRM (a) applied to a horizontal ground nest image with a pipeline of common post-processing operations (b)–(f).



(a) Ground truth labels on the stack of representative nest images.

(b) The output from a Canny-Deriche filter on variable nest images with $\alpha = 0.15$.

(c) Make *Binary*.

(d) *Analyze particles*.

(e) The output from SRM on variable nest images with $Q = 16$.

(f) Make *Binary*.

(g) *Analyze particles*.

Figure 7.10: Stack of variable nest images (a) with ground truth annotations in blue. The binary results from the (b)–(d) Canny-Deriche filter edge detection and (e)–(g) SRM and final post-processing operations.

data required for specific applications. However, there were *no* techniques that could improve the variability of monitoring images. There were few image processing techniques that could improve image segmentations. It was concluded that neither of the techniques could appropriately segment the full *range* of nest images. This was because, there was no single intensity value that could properly define the foreground active nests from backgrounds. Traditional methods were not suitable for generalising over the range of monitoring images.

7.5 TRAINABLE SEGMENTATIONS

There were other characteristics in the monitoring images that could help define areas of active nests. For example, the colour and texture of tumulii. Overall, the monitoring images were highly changeable across time and space. Therefore image segmentations were challenging. The images of horizontal nests would be adequately segmented by other methods. In the previous tests SRM performed well; it was easy to apply and produced a properly segmented image of the horizontal ground nest. However, few if any of the other image types were properly segmented. At least not to the degree necessary to have confidence in the post-processed binary results. Therefore, although SRM worked well on images of horizontal ground nests, the method would not have performed well on the other 83% of monitoring images collected.

When other segmentation methods fail, machine learning techniques provide alternative solutions [172, 188]. Some of these were tested on the stack of representative nest images using FIJI and the TWS toolbox [101]. As a consequence, semi-supervised machine learning tools were considered the *only valid option*, for nest image segmentations. The remaining sections are therefore dedicated to Trainable Weka Segmentation (TWS). This includes an overview of training methods and optimisation techniques. The discussions outlined also include a review of the classifiers designed for the image-centric monitoring system.

7.5.1 *General operation and applied techniques*

Trainable image segmentation techniques work by using human visual knowledge [101]. A set of expertly labelled examples, can be passed to a machine learning algorithm. In TWS a user provides two sets of example labels. ROI tools are used to select traces of pixels belonging to foreground (class 1) and background (class 2) objects. A range of filters are applied to original image data. These are used to create a separate features stack. A user may select any combination of filters, from a possible twenty. They can be grouped according to their main filter functions as described below in Table 7.1 [101]. During the learning process a WEKA machine algorithm uses the examples provided, and the features stack to construct a classifier. The classifier can be used to segment similar types of images. Including the image it was trained on. This is a brief summary of the overall procedure, but it is relatively straightforward.

Table 7.1: Filters available in the TWS plug-in grouped by type.

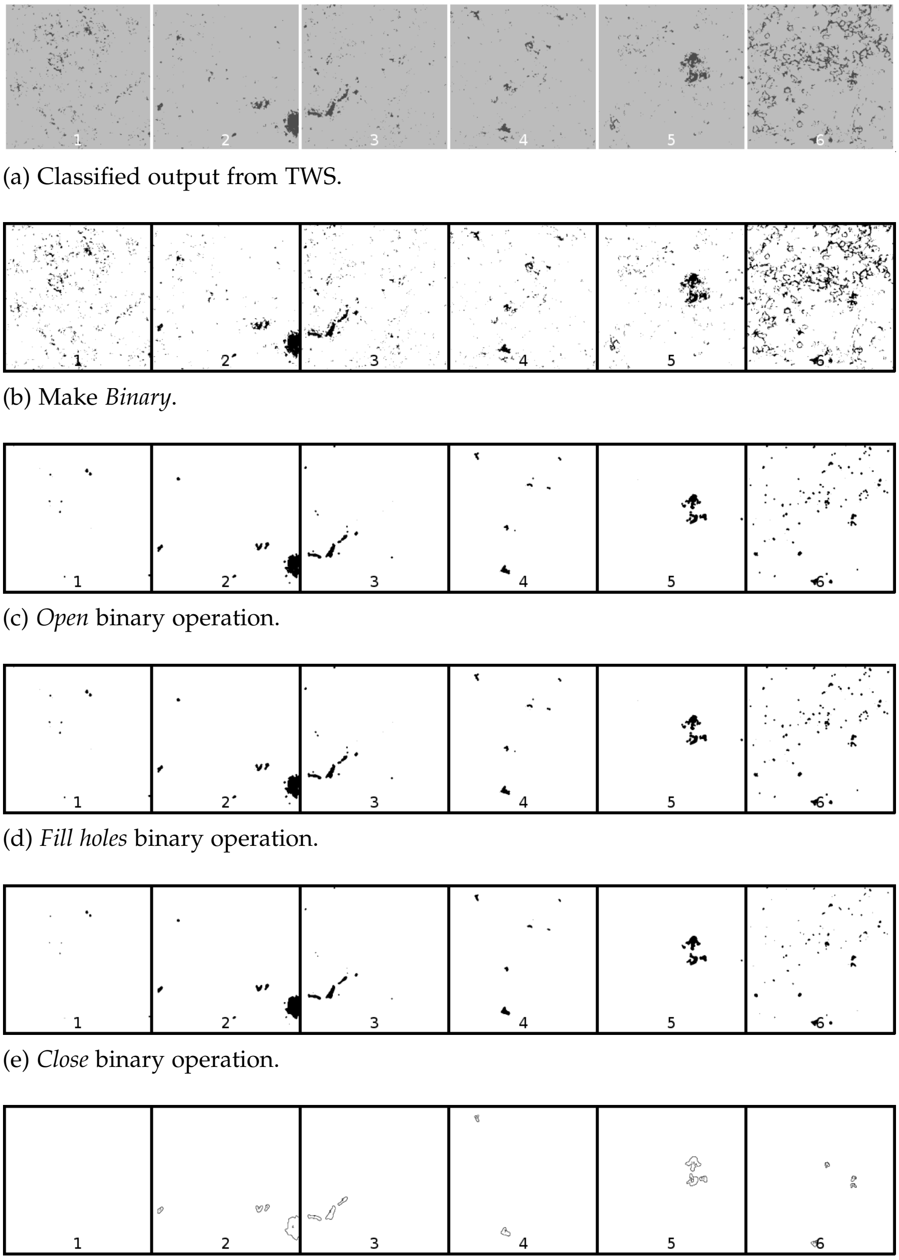
Edge detectors.	Indicate boundaries: Laplacian, Sobel, difference of Gaussians, Hessian matrix eigenvalues and Gabor filters.
Texture filters.	Extract textural information: Minimum, maximum, median, mean, variance, entropy, structure tensor.
Noise reduction filters.	Smooth images: Gaussian blur, bilateral filter, Anisotropic diffusion, Kuwahara and Lipschitz.
Membrane detectors.	Localised membrane-like structures of a certain size and thickness.

7.5.2 *Benefits and drawbacks*

There are aspects of trainable segmentations that are more difficult to quantify [73]. For example, it can be hard to determine what combination of image features best describes key objects. Consequently, some aspects of classifier training are intuitive and subjective. There are also other confounding decisions to consider before applying machine learning for image segmentations [73]. These will be discussed further in later sections. However, providing a user selects appropriate representative pixel samples, and chooses filters that will provide a rich features stack for classifier construction, machine learning algorithms work very well [93]. They can surpass other methods, especially on challenging image segmentations. Preliminary tests on the stack of representative nest images confirmed their effectiveness.

7.5.3 *Preliminary tests*

A classifier was constructed to segment the stack of representative images, using TWS. The results from the classifications were a stack of binary images. These are shown in Figure 7.11 (a). The machine learner classified every pixel in the test images, as belonging to class 1 or 2. The classifier adequately segmented all the slices of the nest images; where the previous methods were inadequate. This is further demonstrated, by comparing the test images in Figure 7.10 (pg. 127) with the classified output from a machine learner, in Figure 7.11 below. The variable image stack was used in comparative checks. The final number of nests counted from segmentations derived from the machine learning method was in much closer agreement with those visually counted and annotated on images. Results are demonstrated in Figures 7.11 (a)–(f).



(a) Classified output from TWS.

(b) Make *Binary*.

(c) *Open* binary operation.

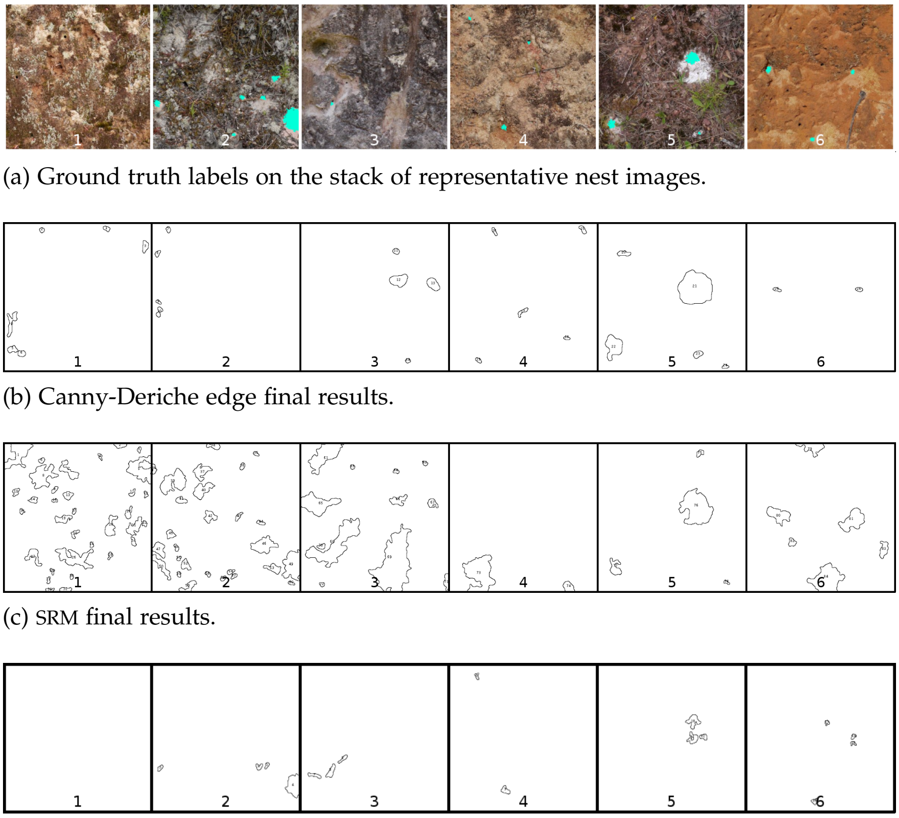
(d) *Fill holes* binary operation.

(e) *Close* binary operation.

(f) *Analyze particles* plug-in: set to count all image objects between the sizes of 100-∞ pixels, with a circularity morphology between 0.1-0.9.

Figure 7.11: The binary results from (a) TWSs and a pipeline of common post-processing operations (b)–(f).

The segmentation results on the variable nest images were *significantly* different to those produced by edge detection and SRM techniques (refer to Figure 7.10, pg. 127). The final particle count results from all techniques were visualised alongside each other. The differences between the methods were obvious, as shown in Figures 7.12 (a)–(d) below.



(a) Ground truth labels on the stack of representative nest images.

(b) Canny-Deriche edge final results.

(c) SRM final results.

(d) TWS final results.

Figure 7.12: The ground truth labels of the (a) stack of variable nest images are compared to the final post-processed outputs from segmentations using the: (b) edge detection, (c) SRM and (d) TWSs. Results from TWSs are in closer agreement with the ground truth labels for each slice.

7.6 Empirical lessons

TWS experiments, rules-of-thumb and optimal parameters for classifiers.

The TWS workbench has undergone significant developments since 2010 [101]. Each year the package has become more versatile. For example, early tests were performed on 8-bit grey-scale images. Initially, TWS was not integrated with WEKA [93]. Also, RF *feature importances* were not easily examined during early tests. However, for the final analysis feature importances were easily examined. For this reason, over the course of this research, each successive classifier testing training and analysis changed [21, 29, 40, 41, 43]. The methods used for nest classifications have steadily improved. The following sections discuss the final results from base-classifier investigations; tuning options and training methods.

7.6.1 *Concepts important for classifier training*

The performance of the final monitoring classifiers was dependent on the quality of classifier training. This included: A) the images selected for training, B) the traces supplied to represent classes during training and C) the filters used to construct the features stack, which were provided to the RF algorithm for training. Some of the important questions considered were:

1. How well do the training images represent the monitoring images?
2. Which pixel traces best contribute towards classifier learning?
3. What image features best help to describe active nests?
4. Can the key features of active nests be enhanced?

In order to determine the optimal parameters for TWSs and the RF classifier, small repetitive tests were applied to the stack of representative nest images. There were two main aspects to consider. The first, was to determine which features were the most important for nest-classifications. This process is referred to as *feature engineering* [73]. The second, was to determine the best tuning settings for the RF model; generally referred to as *classifier optimisation* [73]. These are discussed in the next sections.

7.6.2 *Feature engineering for nest classifiers*

Feature engineering was one of the more difficult processes to evaluate [73]. This was because it was the *total combination* of features, that contributed to the accuracy of the machine learning models used for nest classifications. Generally, the more features provided during training, the better the final classifiers are. However, there were computational costs associated with the construction of the features stacks used for monitoring nest-classifiers. Stack creation and processing was fundamentally limited by the amount of desktop RAM available. This was around 7.2 GBs in total (see Table 5.1).

Therefore, it was important to examine the affects of removing *seemingly* irrelevant features and replacing them with the most *obviously* relevant ones. Textural and noise reduction filters were considered the most appropriate for describing active nest objects in the monitoring images (Table 7.1). It was hypothesised, that these filters would be the most important for classifying images of active nests. Initial tests were performed. The classifiers were re-tuned based on the points below.

- The feature importances output during training.
- The total processing times: the time to construct features stack, time to train the classifier and time to complete classifications.
- The classified binary image results: fully post-processed for comparison against ground truth labels.
- The out of bag error during training.

The filters that did not obviously or significantly contribute to the accuracy of final classifiers, were removed. Training was re-run. The segmentation results were checked. This was achieved by comparing post-processed binary nest counts against raw RGB monitoring images and manual-field nest counts. This process was repeated until the images from each representative slice were sufficiently segmented.

During the first test, classifier C1 was set to default parameters. Of the 79 filters around 58% did not provide any additional information (refer to Figure 6.1, pg. 83). Therefore, the second test classifier, C2, was tuned to provide the optimal nest image features (refer to Figure 6.2, pg. 84). Similarly, at least half of the feature data provided to C2, did not obviously contribute to the final classifications. Furthermore, the time taken to construct the features stack increased. This was due to the iterative processes from the Anisotropic diffusion filter [101]. The out of bag error was lower for test classifier C2, but at the expense of increased processing resources and time. Therefore, classifier C3 was tuned to optimise the speed of feature stack creation, classifier training, construction and application (refer to Figure 6.3, pg. 85). Only the most important textural filters were selected. The final evaluations were good. They showed all the features provided for the training of classifier C3, were important for classifications.

7.6.3 *Calibrating random forest models*

Generally there are no classifier performance penalties for having a RF with more trees [191]. This normally reduces the out of bag error and increases the number of correctly classified pixels. However, as described in the previous section, there were *resource* issues to consider; training and classifications would have taken longer. Also, according to Breiman [192] the number of trees in a RF should be set to 200 initially. It can then be tuned as required. The initial number of random features for RF is given by the square root of the maximum number of features [192]. In the final feature-speed optimised test, there were 20 features used to construct the test classifier, C3. Thus, following Breiman's [192] rule-of-thumb, the ideal number of random features provided to the model for training should have been around five. These two parameters were evaluated in WEKA Experimenter, using the approximate guidelines suggested by Breiman [192].

In the first analysis, twenty-one RF models were loaded into WEKA Experimenter for testing. The results showed the processing time exponentially increased as the number of trees increased (refer to Figure 6.4, pg. 87). The tests also indicated beyond a certain number of trees, there was little improvement in the out of bag error. For example, the gains were not significant after $N = 150$. Therefore, on the balance between processing time, resource usage and error reduction the total number of trees for the test classifier C4, was set at $N = 50$. The test classifier C4 was re-run via TWS. The total model processing was around 11 x faster than C1; 30 x faster than C2 and 2 x faster than C3. Thus, a considerable improvement in processing time was achieved by optimising the features and number of trees.

In the second analysis, twenty RF models were added to WEKA Experimenter. The results indicated, the classifier performance *improved* when more *random features* were provided to the model for training (refer to Figure 6.5, pg.88). These evaluations also confirmed of Breiman's [192] rule-of-thumb for selecting the ideal number of random features. For example, at $M = 8$ the $oo_b = 0.62\%$. This was slightly better than the out of bag error returned for classifier C4, $oo_b = 1.083\%$. These improvements were gained at little processing expense. Therefore, the final monitoring classifiers were constructed with $N = 50$ and $M = 8$.

7.6.4 *Benchmarking classifier performances*

Determining which were the most important performance measures.

Calibration investigations confirmed the ideal features that should be provided to the monitoring classifiers for training. Also, the ideal tuning parameters for the RF models. However, although the RF model is the default machine learner in TWS, many different types of WEKA machine learners were available [93, 123, 186].In theory, other could have been selected and trained to classify the monitoring images. For example, support vector machines (SVM) are popular classifiers [78, 158, 193]. They also perform well in some imaging applications [78, 193]. Until recently SVMs were considered to be one of the most accurate classifiers available [161].

Therefore, the final test classifier, CF ($N = 50$, $M = 8$) was compared to other well known machine learners, using WEKA Experimenter. The purpose of these tests were to gauge the performance of the RF models, over other types of machine learners. Specifically, when applied to the data-set created from the representative stack of nest images. Four RF models were added to the experiment. The final monitoring classifier was added as the benchmark model. Six other models were included; these are listed below. All other classifiers were left at WEKA or FIJI default settings. The percentage of correctly classified instances were reported (refer to Table 6.2, pg. 90).

Benchmark results showed there were no significant improvements by any other models over CF. Three gave results that were statistically worse. The naive Bayes model (M8) did not perform as well as classifier CF on the test data-set. Correctly classified instances were lower at 90.63%. The ZeroR (M2) was the baseline classifier so was expected to give the lowest benchmark. It returned a result of 90.35%. The VotedPerceptron (M9) and SMO (M10) models returned a lower number of correctly classified instances compared to CF; 95% and 96.72% respectively. The results are listed below. They include a brief description of the model, the percentage correctly classified instances and the WEKA model file-identifier, indicating the number of tuning parameters associated with each classifier.

M1 Final monitoring classifier (CF)-99.42%.
hr.irb.fastRandomForest.FastRandomForest-I | 50 | -K | 8 | -S | 1

M2 Baseline model-94.88%*
rules.ZeroR

M3 WEKA Decision Tree-98.37%
trees.J48-C 0.25 -M | 2

M4 WEKA Random Tree-98.02%
trees.RandomTree-K | 0 | -M | 1.0 | -V | 0.0010 | -S | 1

M5 WEKA Random Forest-99.19%
trees.RandomForest-I | 10 | -K | 0 | -S | 1 | -num-slots | 1

M6 TWS Fast Random Forest-98.95%
hr.irb.fastRandomForest.FastRandomForest-I | 50 | -K | 2 | -S | 1

M7 TWS Fast Random Forest-99.07%
hr.irb.fastRandomForest.FastRandomForest-I | 200 | -K | 2 | -S | 1

M8 WEKA Naive Bayes-90.35%*
bayes.NaiveBayes

M9 WEKA Neural Network-95.71%*
functions.VotedPerceptron-I | 1 | -E 1.0 | -S | 1 | -M | 10000

M10 WEKA Support Vector Machine-96.74%*
functions.SMO-C | 1 | -L | .001 | -P | 1.0E-12 | -N | 0 | -V | -1 | -W | 1 | -K |
functions.supportVector.PolyKernel-E | 1.0 | -C | 250007

7.6.5 *Interpreting results*

The test results could not be used to *preclude* the usefulness of other classifiers for the nest-image segmentations [93, 185, 186]. However, the results did highlight the difficulties associated with calibrating some models [73, 161, 186]. Extensive skills are required to tune specific machine learners for any given task. Most require tuning for each specific task [73, 161]. In the benchmark tests only two of the RF models were tuned; the C4 and CF test classifiers. The other models were left at FIJI or WEKA default settings. This was because most were too complex to attempt tuning. As previously mentioned, WEKA has at least fifty different types of models [93]. Each model requires specific tuning. The number of optimisation parameters varies between classifiers. The tuning options for each algorithm can be identified from the WEKA model names. For example, the VotedPerceptron (M9) has 8 tuning parameters and the SMO (M10) has 16. The complexity of the optimisation task for each model, increases with each added parameter. Therefore, no final conclusions were reached about the performance of one type of classifier, over another type.

The overall objectives of the image-centric monitoring system were equally important to consider. One of the aims of this research was to present a *practical* image-centric monitoring method for native bees. Ideally by selecting the most appropriate tools currently available, during each stage of the imaging pipeline. Also, by using tools which could be easily used, with methods that could be replicated. However there were no major constraints regarding the *types* of tools designated for the imaging system. Therefore, the benchmark tests confirmed two central ideas:

1. It was more productive to concentrate on tuning a single machine learner very well, using fully documented methods, rather than comparing the performance of different algorithms.

2. On the nest-image segmentation task, and in comparison to other WEKA models, the RF classifier was easy to tune, train and apply.

Image classification tools were a key aspect of the image-centric monitoring system, at the *segmentation stage* of the imaging pipeline. Because of this, it was apparent the entire training process would need to be easily replicated by others. It is possible other algorithms could have outperformed the RF. However, without a good knowledge of the affects of tuning parameters, the optimisation of other models was considerably more complex. In comparison to some WEKA algorithms, the RF classifier was straightforward to tune, and easy to train. When it was applied to the stack of representative nest images, the segmentations results were sufficient. For these reasons, the RF classifier was selected for monitoring image segmentations. The remaining discussions therefore concentrate on the RF classifier. As it was applied, to segment images of active nests.

7.6.6 *Segmentation performance*

Manual nest counts and visual image-count agreements.

In biomedical imaging applications, segmentation metrics are generally applied to measure the performance of techniques of methods [174]. This is because microscopic images are normally central in most analyses. In these applications, the most common benchmarking method is to provide manually annotated ground truth labels. These are used in final comparisons. In these cases, there are few other avenues that can be used to determine the validity of automated image segmentations. However, in this research, the number of active nests that were represented in each of the monitoring images, were manually counted in the field. Therefore the manual-field nest counts provided a valid benchmark for comparison [176]. It was not necessary to evaluate the image-centric monitoring system results, in terms of the segmentation metrics. Even though manual-image nest counts were also recorded, the agreements between manual-field and automated counts were more critical in the final analyses. Nevertheless, visual assessments of nest image segmentations were a necessary part of the *development* of the image-centric monitoring system. Image segmentations were constantly checked. They were used to optimise classifiers and refine post processing operations. These methods are discussed in the following paragraphs.

7.6.7 *Preliminary assessments*

The representative image stack of six images were fully processed using the final test classifier CF. The results were visually assessed. The original training for the test classifiers was completed in a single session. This was not enough training to properly construct the test classifier. The segmentations from CF were post-processed. The final nest counts did not correspond well with the manual-image or manual-field counts for the representative stack of images. At least, not over the entire image stack. The number of objects counted in slice 6 was particularly high at 64, as shown in Table 7.2. Therefore, a single trace was added and the model was trained again (CF2). The segmentation results were checked. Another trace was added and the model was trained (CF3). The final counts were visually checked against raw RGB images for verification. They were verified against the manual field counts. This is summarised in Table 7.2 below. The output log from TWS training is documented below, in Listing 7.1 (pg. 138).

The number of pixels representing active nests (class 1) was not altered for classifier CF re-training. The number of background pixels (class 2) for CF1 = 816, CF2 = 864 and CF3 = 996. The output results showed little change in the out of bag error between classifiers: CF1–$oo_b = 0.581\%$, CF2–$oo_b = 0.661\%$ and CF3–$oo_b = 0.577\%$. However, the segmentations changed markedly. The number of objects counted

Table 7.2: The final automatic count results on representative images (slices s1–s6) using CF_{rt} compared to manual-image and manual-field counts.

Method	slice 1	slice 2	slice 3	slice 4	slice 5	slice 6
CF1	21	11	14	22	4	64
CF2	21	2	3	13	3	14
CF3	7	3	8	4	3	8
Manual image	3	3	3	2	2	2
Manual field	2	4	4	3	4	9

```
----------------------------
CF1 Training input:
# of pixels selected as class 1: 44
# of pixels selected as class 2: 816
Creating training data took: 38ms
Training classifier...
FastRandomForest of 50 trees, each constructed while considering 8 random
     features.
Out of bag error: 0.581%
----------------------------
CF2 Training input:
# of pixels selected as class 1: 44
# of pixels selected as class 2: 864
Creating training data took: 34ms
Training classifier...
FastRandomForest of 50 trees, each constructed while considering 8 random
     features.
Out of bag error: 0.661%
----------------------------
CF3 Training input:
# of pixels selected as class 1: 44
# of pixels selected as class 2: 996
Creating training data took: 28ms
Training classifier...
FastRandomForest of 50 trees, each constructed while considering 8 random
     features.
Out of bag error: 0.577%
```

Listing 7.1: FIJI output log from test classifier (CF) re-training.

in slice 1 and 3 were marginally raised. All other slices were reasonably well segmented. These results highlighted several concepts:

1. Nest-classifiers could be retrained using a very minimal number of corrective-traces.
2. For nest images these corrections were always traces of background pixels (e.g. class 2).
3. It was difficult to train the test classifiers to properly segment all the slices in the representative stack (e.g. slice 1 and 3).
4. The out of bag error could not be relied on to reflect the quality of image segmentations.

7.6.8 *Summary of classifier tests*

The lessons gained from empirical analyses and test classifier performance tests were important. They contributed greatly towards the training methods adopted for monitoring images. There are no standard procedures that can be applied. Generally, each machine learning model is tuned to perform well on a specific task [73]. Also, since the advent of digital images, hundreds of automatic and semi-automatic segmentation algorithms have been developed [188]. No single method could be considered appropriate for all types of images. Those designed for a particular type of image, are not always applicable to others. In this research the RF was selected. The tests confirmed they are easy to train, tune and apply.

The variability of active nest images presents a challenge for *any* classifier model. Monitoring photographs were inconsistent and diverse. The same active nest monitored over time could change in colour and texture. This was because the local landscapes naturally altered. The nests of native bees were variable at coarse and fine scales. Nests within the same community could be vastly different. They also varied between geographically separated communities as natural landscapes changed.

Therefore, some general rules for training were adopted. This was in order to gain the best possible performance of the RF classifier for this analyses *and* so training could be *closely replicated* by others. The term closely replicated, comes with a caveat. The semi-supervised training method cannot actually be repeated. This is because, each time a classifier is trained, the final model will always be different. When a new model is applied to the same dataset, it will produce alternative segmentation results. RF models are also constructed with a random seed. The seed is used during training. Because the processes are stochastic, the random seeds, models and trace data-sets would be necessary in order to reproduce the same image segmentations.

The tests highlighted both the strength and weakness of semi–supervised machine learning. User-supervised learning *can improve* image classifications. This is because humans draw on considerably more knowledge when performing manual segmentations. But, a human-interactive approach is dynamic. Therefore, the performance of final classifier models are dependent on a number of subjective choices. Such as the images used for training, the human-selected traces provided during training and the suitability of features supplied to the classifier during construction. As mentioned previously, feature engineering can be ill-defined. Although the importances of single features can be analysed, it is the combined value of all features that contributes to final model performances. The interaction between training features cannot be fully analysed. The process is dynamic and it occurs during classifier training.

Nevertheless, and despite these considerations, on the task of segmenting active nest images, there were few other image analysis options. No other methods could perform appropriately with the monitoring images. As demonstrated, most traditional methods cannot be used to segment the range of nest images gathered during field monitoring. Some could be used to segment horizontal grounds nest images. However, this was because they were much easier to threshold. Most traditional methods, which are based only on the intensity information of pixels, are not suitable for segmenting the range of variable nest images.

The tools and methods to test, train and apply machine algorithms are freely available, in software platforms such as TWS and WEKA. The TWS plug-in was designed to use human knowledge in the segmentation process. This technique improves the accuracy of the labelled regions. The algorithms in TWS have been developed for medical images. However, they can easily be applied to other tasks. Classifiers such as the RF can be adapted for other types of images, in different applications, using the TWS platform. The software is easy to work with and is accessible to experienced and inexperienced users. The methods used to train RF classifiers for segmenting the nest-monitoring images, using TWS, are discussed in the next sections.

7.7 MONITORING CLASSIFIERS

As previously discussed, the more trees and features that are used to train RF classifiers, the better the final image segmentations are. However, the computational costs increase as the number of features and trees grow [191]. This was an important consideration. Computer hardware and performance limited the final design of monitoring classifiers. For this reason a minimalistic approach was adopted. Classifiers were optimised for speed to reduce the overall time required to process the monitoring images. A total of 1896 slices were processed, each image was 32-bit RGB, 500 x 500 pixels in size. Final classifications were completed in a day. This was a substantial improvement on previous analyses [21, 29, 40, 41, 43]. Additionally, any obvious processing errors could be quickly rectified. The batch classifications could be re-run.

7.7.1 *Variability of training images*

Four training stacks were collated for each site. Each site-grid classifier was trained separately using respective stacks. This was to simplify and speed up the training process. This method also improved overall classifications. This was because models could be constructed on a *sub-sample* of variable images. Thus, they could be tuned for the specific image types, associated with each monitoring site. There were twelve stacks in total, four per site.

7.7.2 *Morphological operators tuned for segmentations*

Several morphological operations and pipeline combinations were empirically tested, using the small test stack. This was before the final post-processing operations were applied to the classified stack of monitoring images. When the morphological operations were completed, the counted results were visually checked against the original RGB monitoring images. They were verified against the manual field counts. The post-processing operations were tuned specifically for the segmented monitoring-images produced using the CF classifiers. If new classifiers were constructed, new post-processing operators would most likely be necessary. This is because, even if the same method is used to construct the nest-classifiers, the final models would not be exactly the same. Thus, they would not produce replica image segmentations.

Additionally, the performance of final classifier segmentations were compared to nest-segmentations produced using classical methods. There was no attempt to tune the post-processing operators for image-segmentations produced from the classical intensity-based methods. It is possible, although very unlikely, the final nest count results from the classical methods could have been improved by morphological operators.

7.8 VERIFICATION METHODOLOGY

As discussed previously, classifier performance evaluations provided confirmation of the suitability of the RF, as machine learning tool for nest image segmentation. When combined with the functionality the TWS workbench, they were easy to train, test and apply. Segmentation of nest images using the model, were good enough given the irregularity of image data. They were considerably better than the classical methods tested. When the RF was compared against other machine learners, classifier tests and benchmark investigations did not necessarily preclude the suitability of other models (Table 6.2). But, there were practical benefits associated with the RF classifier that could not be underestimated. These benefits probably account for the popularity of RFs in real-world data applications [169, 194, 195]. They contributed to the adoption of the model for nest image-segmentations.

However, the main verification of the image-centric monitoring method primarily focused on the agreements between manual-field and automatic nest counts. Secondary to this, were the differences between manual-image counts, compared to the automatic and manual-field counts. Inter-observer correlations between manual-image counts were also measured [179]. There were insufficient data for inter-observer correlations between the manual-field counts by two observers (refer to Chapter 4, Section 4.2). This was because field data were collected primarily by a single observer. Before the final comparisons were made, the automatic nest count results were modified. The reasons for this are outlined further in the proceeding paragraphs.

7.8.1 *Replica image collections?*

Each image collection was comprised of near-replica images. They were data collected from the same location, grid and day but separated by minutes and seconds. Separate collections were therefore comprised of different images. Each single one, was acquired under varying natural conditions. During preliminary analyses, there were some individual nest counts that were vastly different between the collections. This was unexpected. The reasons for the variations were investigated further. Where count data was significantly different between the collections, the raw monitoring images were visually checked. A pdf document comprising of raw RGB monitoring images, alongside the count overlay results, was used to visually investigate discrepancies. It was found that in most cases, the exposure and illumination varied considerably between image collections. This therefore produced some very different image segmentations and final count results. To mitigate this issue, the median counts were taken across the three image collections. These were the final data which were used to compare manual-field, manual-image and automatic methods.

7.8.2 *Lin's concordance correlation coefficient measure*

Lin's concordance correlation coefficient is an accepted measure used to determine the level of agreement, accuracy and precision between different methods [176, 179]. There are several points to consider when using the assessment. Firstly, it cannot be used to judge the *correctness* of the separate methods in isolation; only the degree to which the different methods agree with each other [196]. Secondly, descriptive scales for the concordance correlation coefficients have not been formally established. McBride [197] has provided unofficial guidelines for the measures of agreements (of continuous variables); these are shown in Table 7.3 below. Finally, the guidelines shown were designed for *laboratory applications*; they are included here because they are the *only* descriptions documented to date. They provide the ideal standards for continuous variables using laboratory methods. The same level of control over experimental conditions are not feasible when collecting natural, outdoor biological data. For these reasons, it is reasonable to expect more relaxed levels for the nest-count analyses discussed here; as has been reported in similar types of applications [177, 178].

Table 7.3: Guidelines for the strength of agreements using concordance of correlation (ρ_c) for laboratory methods. There are no published guidelines for field biology; a more relaxed scale for natural data is expected.

Range of ρ_c	Description
0.00 - 0.65	Poor
0.65 - 0.80	Moderate
0.80 - 0.90	Substantial
0.99 - 1.00	Almost perfect

7.8.2.1 *Manual-field, image and automated nest counts*

The level of agreement, accuracy and precision between the number of nests counted by different methods was measured using Lin's concordance correlation coefficient [176, 179]. The most important of these comparisons were between the number of nests counted in the field (which were the *gold-standard* measures) and the automatic counts derived from the CF classifier.

In order to better understand the range of results, method comparisons were ranked in order of the *total sum* of performance measures. These are shown in Table 7.4 (pg. 145). Inter-observer counts were ranked the highest. The differences between the manual-image counts from two scorers ($n = 170$) showed there was good agreement ($\rho_c = 0.867$), precision ($r = 0.891$) and accuracy ($C_b = 0.973$). The differences between the automated counts ($n = 1284$) derived by thresholding (a-th) and classification (a-CF), were ranked the lowest. There was no agreement ($\rho_c = 0.04$), precision ($r = 0.284$) or accuracy ($C_b = 0.164$).

These results appeared valid. According to similar types of analyses, inter-observer variability and measurements were within acceptable ranges [177, 178]. The counts derived from the a-th method were not expected to be reasonable. This was because only 1/3 of the image data could be sufficiently segmented using intensity measurements (i.e. only the horizontal ground nests). Therefore, in the absence of more appropriate guidelines, these upper and lower ranks were used to assess the validity of the other comparisons.

The manual-field and a-CF methods were ranked second overall. There was good agreement between manual-field and a-CF counts ($\rho_c = 0.738$), with good precision ($r = 0.828$) and accuracy ($C_b = 0.891$). The manual-image and a-CF methods were ranked third. They showed a similar agreement ($\rho_c = 0.679$) and precision ($r = 0.705$) and slightly higher accuracy ($C_b = 0.963$). The manual-field and manual-image methods were ranked in fourth place. Overall there was much less agreement ($\rho_c = 0.622$) and precision ($r = 0.641$) but still a reasonable accuracy ($C_b = 0.970$).

7.8.3 *Interpreting results*

In summary, these results indicated the automated a-CF counts were in closer agreement with the true nest count values, than those estimated from images. The automated counts showed greater variation. The manual-field counts were generally lower and more discrete. Manual-field nest counts were more likely to be rounded up, or down to the nearest whole numbers; counts above forty were always estimated. Data were highly skewed with a very high dispersal index (refer to Figure 6.10, pg. 99). They were not normally distributed [45, 47] (refer to Figure 6.10, pg. 99). Data were not transformed for analyses. The relative methods performances and yearly trends were sufficiently apparent when raw count data was used. A graphical analysis of the mean nest counts and the standard error of the means (error bars), were used to describe the trends in active nests by three different methods. The standard errors for mean nest counts per site and year were moderate to high in most instances. They were indicative of the wide variation in the numbers of active nests. Similar types of standard error trends have been reported for nest counts in related studies [16, 64, 66].

Table 7.4: Methods ranked by performance measures of precision (r), agreement (ρ_c) and accuracy (C_b).

Rank	Method 1	Method 2	n	r	ρ_c	C_b
1	m-image ob1	m-image ob2	170	0.891	0.867	0.973
2	m-field	a-CF	520	0.828	0.738	0.891
3	m-image	a-CF	170	0.705	0.679	0.963
4	m-field	m-image	170	0.641	0.622	0.970
5	a-ths	a-CF	1284	0.244	0.040	0.164

Table key

Automatic (a-) and manual (m-) methods

Images segmented by monitoring classifier CF	a-CF
Images segmented by default thresholds	a-ths
Nests counted from images by two scorers	ob1-2
Number of (paired) samples	n
Pearson's correlation coefficient (precision)	r
Lin's Concordance of Correlation (agreement)	ρ_c
Bias correction factor (accuracy)	C_b

7.8.3.1 *Nest-counts, population trends and other ecological observations*

Long term, quantitative records would be required before substantial conclusions about ecological trends could be reliably established. Nevertheless, the nest-count data indicated a *general decline* in populations of native bees, at all three monitoring locations, over five years. To examine these trends, nest-count data collected during the first monitoring season at each site (i.e. 2010 for S1 and S2, and 2011 for S3) were used as a benchmark to compare against subsequent years. The trends showed general moderate to marked decreases. The percentage declines varied between 27–70%, at all three locations between 2010 and 2013. These are summarised in Table 7.5 below. There were not enough data collected in 2014 ($n_{days} = 3$) to establish firm conclusions about trends for that year. Some data appeared unstable (e.g. see m-field for S1, Figure 6.16, pg. 108). However, the graphical analysis suggested slight increases in nest numbers.

The underlying ecological processes contributing towards the decreases in the numbers of active nests of native bees (at the locations monitored) are probably multifaceted. The community changes could be a consequence of: 1) natural cyclic successions [198, 199], 2) an alteration of community landscapes creating habitat fragmentation/degradation, 3) the affects of environmental pollutants/toxins, or 4) increasing competition with exotic bees for food resources.

Without long term monitoring, it is difficult to determine if fluctuations in populations (as indicated by the decline in nest counts) are suggestive of a population-crisis or if they are part of a much larger natural cycle?

General observations have indicated macro and micro climatic changes could be a confounding factor impacting communities of native bees in Whangarei. For example, no nest-count data were collected in 2009, the year monitoring was initiated. This was due to a severe drought which impacted Northland. Although surveys indicated native bees were emerging normally, they were only active for a few weeks.

7.9 PERFORMANCE-COST MEASURES

Manual labour, hardware performance and processing requirements.

The time taken to construct the features stacks, train and apply classifiers was an important consideration in this research. This was principally due to the limited capabilities of the desktop computer that was used for imaging tasks (refer to Table 5.1, pg. 57). It would not be considered high performing, by most current standards. There were some adjustments made to the operating system environment that helped to optimise the desktop for image processing tasks; these are discussed further in the sections below.

From initial analysis of classifier settings using the representative stack of six images, a range of feature importances were investigated.

Table 7.5: Decline in the number of active nests (%), *benchmarked* against nest counts in 2010 for S1 and S2; 2011 for S3. The upper (*) and lower (**) percentage decreases are shown for automatic (a-CF) and manual field (m-field) methods. The trends show a decrease in the number of active nests (S1-S3) between the years 2011–2013. Although based on a small sample, a slight rise in 2014 is indicated (▲).

Site	Method	2011	2012	2013	2014
S1	a-CF	41 - 47	45 - 48	55 - 58	5 - 15 ▲
	m-field	32 - 33	36 - 40	48 - 49	40 - 46
S2	a-CF	35	66	42 - 43	26 - 37
	m-field	27	54 - 55	27 - 29	38 - 39 ▲
S3	a-CF		67 - 70	65 - 69**	48 - 59
	m-field		37	36	4* - 13

Table key

Automatic (a-) and manual (m-) methods	
Images segmented by monitoring classifier CF	a-CF
Percentage increase indicated	▲
Lowest percentage decrease	*
Highest percentage decrease	**

Of those tested, Gaussian blur, Mean, Minimum, Median, and Structures filters produced sufficient segmentations (see Figure 6.3). Moreover, they were all shown to have importances that contributed to final classifications. The number of sigma was reduced from the default setting to $\sigma = 2$. This also reduced processing expenditure. Combined with the raw RGB and HSB images, a total of 20 features were used for training for the final monitoring-classifiers. Monitoring images were processed in a single day and classifications could be easily repeated if and when required.

7.9.1 *Operating system environment*

During previous analyses, there were significant memory problems during processing. This problem normally surfaced as out of memory exceptions. They were thrown during TWSs. Some of the early classifications required days to process. A range of solutions were tested such as increasing RAM memory and reducing the size of image stacks. Although most solutions helped a little, the performance of processing considerably improved when the operating system was changed to Linux (early 2014). FIJI and WEKA are cross-platform. However, the performance of both software packages were noticeably enhanced under Linux. The operating system was more versatile. It was easily configured for the specific imaging tasks. Ubuntu swap memory was turned off during image processing jobs. This stopped processes from being swapped out of physical memory. Virtual memory options were passed to the JavaVM from FIJI's main configuration file. This was to increase the memory heap size. Combined these adjustments improved the stability of the operating system, software performance and final classification processing times. The memory leakage problems that occurred during previous analysis, were entirely mitigated during final classifications.

7.10 IMAGING PIPELINE DESIGN

The final image pipeline design is shown in Figure 7.13 below. A list of corresponding software tools and methods that were used to complete each task, are listed for reference in Table 7.6 (pg. 150).

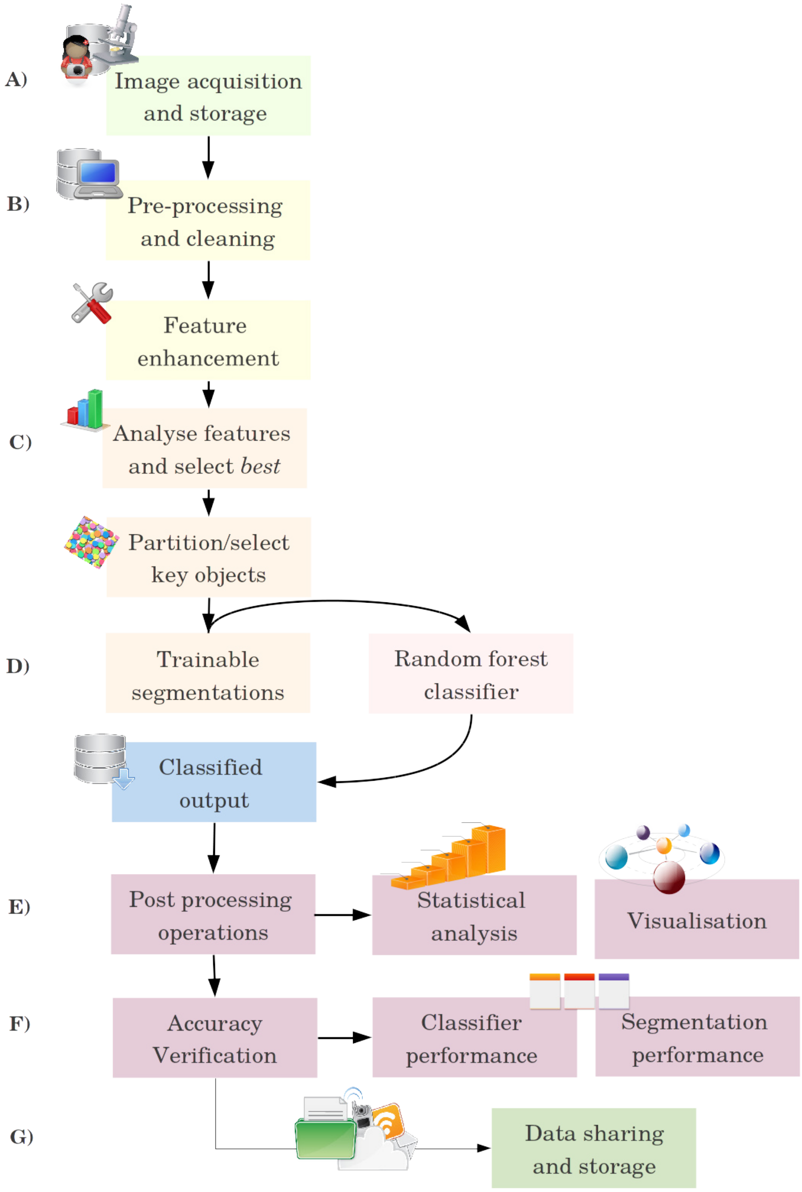


Figure 7.13: Image analysis work-flow.

Table 7.6: Image pipeline tasks and tools.

Stage and task	Tools
A) Acquisition:	
Copying, sorting, storing and sharing data.	Transfer from SD card to PC – Linux terminal bash command [200]. Store on partition hard-drive and external drive [200]. Share data via Git-hub repository [201, 202].
B-C) Preprocessing:	
Rename by EXIF data.	pyRenamer – mass file re-namer for GNOME [203]. Exif tool – read, write and edit meta information [204].
Crop to region of interest.	Xnview – viewing, converting, organising and editing raster images [150].
Collate, adjust parameters and review information.	FIJI – image processing workbench [101].
D) Classify:	
Features and classification.	FIJI TWS plug-in – machine learning algorithms and selected image features for pixel–based segmentations. [101].
E) Post processing:	
Enhance for segmentation.	FIJI – morphological binary operations such as erosion, dilatation and thinning to remove isolated pixels [101].
Count segmented areas.	FIJI Analyze particles plug-in – count number of objects based on size and circularity, save as CSV file.
Prepare CSV results data for analysis.	Apache Open Office – Software suite for word processing, spreadsheets, presentations, graphics, databases. [205].
F-G) Verification:	
Real-time logs.	FIJI TWS plug-in – runtime logs of performance measures [101].
Compare classifiers.	The WEKA workbench – a collection of machine learning algorithms for data mining tasks. [93, 108].
Visualisation.	RStudio–software environment for statistical computing and graphics.[206]
Sharing.	SmartGit – GUI tools for committing (git-gui) and browsing [207].

Three communities of native bees were monitored over five years. The number of active nests were manually counted. Images of the nests were collected for analyses. Monitoring methods were primarily designed to reduce the number of variables influencing image data. Therefore, exactly the same nests were monitored, in nearly the same way, approximately the same time each collection day, every season. Despite this, the images of active nests were highly variable. They proved difficult to segment. Because outdoor images were a requirement, many of the problems were unavoidable. There were also limitations with the hardware tools used for field image collections, processing and analysis. These constraints were not easily overcome. Finally, data management and protection was another factor impacting the outcomes of this research. Lightning strikes were the most likely cause of a fatal hardware failure in 2012; the incident resulted the loss of some key image data. Combined these issues had an impact on the design of the final image-centric monitoring system presented in this thesis.

A complete imaging pipeline was developed, tested and verified using open-source tools and methods. The software tools, code and scripts that were used to process the monitoring images and confirm results, were fully documented for easy replication. Monitoring images were segmented using TWSs and were based on the RF classifier. The RF uses a number of random choices during the learning process. As a result, this aspect of the imaging pipeline would be difficult to replicate. Trainable image segmentation methods are relatively new technologies, as are some of the concepts and theories they are based upon. This includes a synergy of current image analysis theory, applied machine learning, data mining concepts and knowledge discovery. FIJI and the TWS workbench have been invaluable tools in this research. However, FIJI was designed for biomedical applications. Consequently, there were few general procedures to follow. There were no known documented examples to draw upon during the development of the image-centric monitoring system. Therefore, the methods that were used to test, train, construct and apply the monitoring classifiers were fully outlined. There could be alternative methods that are more suitable for the monitoring application described. All the same, the techniques were developed and refined over many hours of empirical tests. They provide a good platform for future developments.

Segmentation metrics are often used to evaluate the performance of imaging systems. However, these metrics were not used to test the performance of the image-centric nest monitoring system. Rather, the level of agreements between counts by different methods were measured [176, 179]. The number of *actual active nests* provided the benchmark. All other data were compared to these counts. Five analyses were conducted; the differences between manual-field, manual-image and automated counts derived from classical threshold methods, or from the final monitoring classifier (CF) were compared. Similar results have been reported for inter-observer variability between visual scorers assessing plant diseases [177].

The automated a-CF counts were in closer agreement with the true nest count values, than those estimated from images. There was moderate agreement between manual-field and a-CF counts ($\rho_c = 0.738$), with good precision ($r = 0.828$) and accuracy ($C_b = 0.891$). The manual-field and manual-image methods indicated there was much less agreement ($\rho_c = 0.622$) and precision ($r = 0.641$); but still good accuracy ($C_b = 0.970$).

Graphical analyses of the mean nest counts and the standard error of the means (error bars), were used to describe the trends in active nests, by three different methods. The standard errors for mean nest counts per site and year, were moderate to high in most instances. They were indicative of the wide variation in the numbers of active nests and similar to levels reported in related studies [16, 64, 66].

The nest count results taken over five years showed moderate to marked decreases. The percentage declines varied between 26–69%, at all three locations. There was not enough data collected in 2014 to form substantial conclusions. However, the graphical analyses suggests a slight upward trend in nest numbers.

8 CONCLUSIONS

Synopsis

There are few tools to aid in the study of New Zealand's native bees. Thus, base-line data on the population health and status of native bees are limited. Traditional field methods can be time consuming and complicated. This situation appears unlikely to change in the near future. Therefore, the main aims of this study were to *design*, *apply* and *verify* the performance of a simplified *image-centric type* monitoring system for New Zealand's native bees. There were no *specific constraints* about the types of imaging methods or technologies used. There were a range of approaches that could have been taken. Looking towards the future, community scientists could play an important role as human sensors. Monitoring initiatives might even be community initiated and driven. Therefore, open source solutions were more viable. For this reason the research design was fundamentally *solutions driven*. This chapter provides an overall summary of the research outcomes and a perspective on future developments.

THE KEY CONCEPTS:

- If the populations of solitary ground nesting bees can be estimated by the number of active nests in a community, then the number of active nests can be manually counted.
- If the number of active nests can be *manually counted* in the field, then they can be estimated from images.
- If they can be counted from images, then it is also feasible to automate the processes.

This research was based on the idea that the number of *active nests* of native bees, could provide a *proxy for populations*. That, it was possible to *design field methods*, to reliably count the number of active nests at communities of native bees over space and time. That digital images could be used *in the place* of manual counts; simplifying methods. Finally, that it was possible to *process* digital images to *reliably count* active nests.

THE OUTCOMES OF THIS RESEARCH ARE:

1. The fully documented designs of the image-centric system.
2. The monitoring data collected over five years.
3. The interpretation of results.
4. The considerations for future developments.

A PRACTICAL MONITORING PROGRAMME was designed to collect manual nest counts and images of active nests. These data were used for comparative analysis and proof of concept. Monitoring was conducted over five years, at three communities of native bees in Whangarei (New Zealand). A total of 1896 images were collected, representing 158 monitoring days. They were processed and used in a comparative analysis against manual field nest counts.

THE PERFORMANCE EVALUATION of the image-centric system was based on a straightforward statistical analyses, using Lin's concordance coefficient of correlation. The agreement between manual-field and automated counts were tested. Results indicated the image-centric monitoring system could be used in the place of manual sampling. There was reasonable agreement, accuracy and precision between manual-field and automated methods.

OPEN SOURCE TOOLS WERE SELECTED for *all* nest imaging procedures. Including data management, pre-processing, classifications, post-processing and analyses. The image processing pipeline and methods were described in detail. Field acquisition tools and equipment were off-the-shelf or easily constructed. Therefore, any of the techniques could be quickly replicated, adapted or modified for similar projects.

MONITORING IMAGES WERE HIGHLY VARIABLE. The pixel areas representing active nests were not easily defined. Therefore, classical segmentation techniques were not appropriate. In the past, there were few alternatives to classical methods. However, there are new techniques based on interactive machine learning. Therefore in this research, nest image segmentations were achieved using a semi-supervised machine learner and the TWS workbench.

THE PROCEDURES DEVELOPED TO OPTIMISE, train and apply machine learning classifiers were described in detail. Although replication of the machine learning methods could be more difficult, the central concepts were detailed. The image processing scripts, raw-field and raw-image data were made available via GitHub[1] [201, 202] and Appendices F and E.

THE FINAL MONITORING RESULTS gave reliable estimations of the number of active nests, at the communities evaluated. Also, the changes in nest numbers over time. Therefore this research has provided sufficient evidence to show the populations of native bees could be reliably measured, by using the image-centric monitoring method outlined.

IN THE FUTURE nest image data could be collected using random sampling methods. Thus, providing data that could be used for a more rigorous scientific analysis. The field sampling methods have little impact on the performance of the image-centric monitoring system.

THE IMAGE-CENTRIC METHOD COULD STAND ALONE. It would not depend on collecting manual nest count data. It could also be possible to modify aspects of the image acquisition methods. For example, imaging methods could incorporate remote imaging/sensing technology. This would reduce the dependence on manual labour and increase data capture for analyses.

THE IMAGE-CENTRIC SYSTEM WAS DESIGNED around four key aspects. These are itemised below; with reference to original research questions. Future developments are included in some of the points discussed.

- What indicators could be used to establish the general health of bee communities?
  1. There was some reasonable evidence supporting the idea that the *active nests of solitary ground nesting bees can provide a good proxy for populations* [16, 64, 65, 66].
  2. This hypothesis was not confirmed. However, in the future, this could be tested in the field. Monitoring would need to include other methods. For example, emergence traps, nest core samples, or mark release recapture techniques [54].
  3. There are good possibilities other types of digital data could be incorporated into future research. For example, mark release recapture techniques could be based on audio-type sampling technologies (e.g. the system by Potamitis et. al [84]).
- Could images be used to capture or quantify *key* indicators?

[1] *NZbees*-http://nzbees.github.io

    1. Several image-types could have been used to quantify aspects of the biology and behaviour of native bees and their communities. For example, Figure 7.1 (b) (Chapter 7) shows images that could be used in survey-type methods. They could be adapted for biodiversity-type sampling.
    2. However, active nests were the basis for the image-centric method presented in this thesis. There were several reasons for this. Firstly, there was reasonably good evidence supporting the methodology [16, 65, 66]. Secondly, compared to other image-types, the images of active nests were much easier to capture, process and analyse. Specifically:
        a) The images of active nests were easy to *capture*.
        b) The image area was easy to *regulate*.
        c) The image analyses required only *two objects* to be segmented.
        d) The results from analyses provided *nest counts*.
        e) *Nest counts* could be directly used to measure populations.
    3. The segmentation tasks for active nests were more straightforward when compared to other types of images. It was only necessary to segment areas of *active nests* from all other *backgrounds*. Furthermore, data from image analysis was based on the *number of active nests* in the images. This data *directly relates* to estimations of populations. Because of this, *interpretation of results* were straightforward. They were achieved with *minimal post-processing* and statistical treatments.

- Can image handling, acquisition and analyses be standardised?
    1. Some of the imaging tasks were standardised - other tasks were not as straightforward.
    2. Since *outdoor images* were required, it was not possible to regulate the variation in images (e.g. brightness and illumination). This aspect could not be standardised. Consequently, some of the methods would be more difficult to replicate (e.g. classification based image segmentations). Nevertheless, *general image acquisition* methods could be easily replicated. Off-the-shelf tools and equipment were used. The grid-based image acquisition technique was straightforward and could be easily adapted.

- What pattern recognition, segmentation or classification techniques best suit the image data?

1. *Classification based* segmentation techniques were the *only valid option* for the images of active nests. This was due to the wide variability of images. Traditional intensity, edge or region-based methods were not appropriate for the segmentation stages of the imaging pipeline.
2. *Feature engineering* concepts are central to many machine learning-based classification techniques. These could be further investigated. Future developments around image-centric monitoring for biological field sampling, could focus solely on feature engineering. For example, in some instances several features may be closely correlated. In these cases, is it really possible to determine which features are important?
3. Also, as a concept feature engineering is often ill-defined; but it is central to the success of machine learning techniques. Is it possible to develop a selection of feature-optimisation criteria for general outdoor imaging applications?
4. The RF was applied as a *classification tool*. It is well suited to applications involving natural data. Consequently, RFs are popular. Current research indicates RF classifiers are in fact, the most powerful and accurate learners available. For these reasons they were a good choice for image segmentations used in this research.
5. However, despite their popularity, the *statistical mechanisms* are still not that well understood. This is partly because they are based on *stochastic processes*. Also, they operate within *algorithmic codes*. Therefore, some of the processes are effectively existing within *black-boxes*. Finally, while other machine learners can be visualised, a RF could have many thousands of branching trees. Therefore, they are nearly *impossible to visualise* on monitors. These are all aspects of RFs that warrant deeper scholastic investigations.

- What methods can be used to verify the accuracy and precision of imaging methods?
    1. Manual-field nest counts were directly compared to counts taken from images of active nests, and those derived from classification-based image segmentations. Thus, verification was straightforward.
    2. Lin's concordance of correlation statistics were used to measure the degree of agreement, precision and accuracy of methods. This was a comparative analyses.
    3. The final analyses showed the number of nests counted from images (visually) or by automated-image counts,

compared favourably with manual-field counts. These data indicated manual-field methods could be substituted with the image-centric methods outlined.

4. There are a range of performance measures that were not tested.

    a) Image-based *segmentation metrics* were not checked. However, they are commonly used measures in biomedical imaging applications [174]. Segmentation metrics are important tools. But, there were no valid reasons to apply them in this research. The proper ground-truth measures were the actual number of active nests counted in the field.

        i. In the future, image segmentations could be applied in *real-time*, in the field. Thereby confirming the number of nests counted via the automated methods, are entirely consistent with the number of active nests counted in the field. This would only be required once (per location). Classifier training could be based on this initial calibration.

        ii. Real-time calibration of the image-centric monitoring system could be *reasonably achieved* with *minimal effort*. This would improve the overall performance of the design and increase the reliability of classifications.

    b) A range of machine learning performance measures were not used. As with segmentation metrics, there were no clear reasons to apply the tests. Furthermore, most measures are important for machine learning outcomes. Especially during the development of algorithms; or comparative tests. However, they are not necessarily relevant to studies involving the application of machine learning algorithms. They were not relevant to the outcomes of this research.

8.1 FINAL CONSIDERATIONS

COMMUNITY-BASED MONITORING Citizen science collaborations might provide benefits *beyond the collection of ecological data* [208]. In the future, citizen scientists could play an important role as *kaitiaki* (stewards) of native bee communities. Lessons taken from established initiatives, such as the Backyard Kiwi [2], might help to springboard community-based, bee-monitoring initiatives in Northland. This re-

[2] http://www.backyardkiwi.org.nz/

search provides a good basis for further development of community-based methods for monitoring native bees, throughout *Aotearoa* (New Zealand) and abroad. But, there are several important aspects to consider:

- Many people are aware of the importance of honey and bumble bees; and can easily identify the insects. This is not *generally* the case for New Zealand's native bees. They are frequently misidentified as wasps or flies. Therefore, raising public awareness is the *first*, critical step towards increasing the conservation and protection of native bees and their habitats.
    1. Knowledge sharing is an important consideration that may impact the conservation of native bees. For instance, there are at least five anecdotal accounts, where home-owners have fly-sprayed entire communities of native bees, mistaking the insects for wasps.
    2. A social network site, nzbees[3], was established to engage with a larger audience and share more information about New Zealand's native bees.
        a) However, management and creation of web-content has proven to be a time-consuming task. This aspect may need further investigation.

        b) Also, there are other avenues for disseminating information, through schools, marae (community houses) and other community-based groups. These have not been widely explored in this research.
    3. A GitHub account, nzbees [4], was created as a repository to share and distribute image-data for analysis. This includes a dedicated website for nzbees[5], with content designed for distribution of information to the public.
        a) As above, the management of the website and repositories can be time-consuming. Furthermore, GitHub was not designed for storage of image based big-data. Therefore, these aspects require further consideration.

        b) Digital image database structures and the management of centralised processing are significant areas that were *not* investigated in this work. In the future however, they could require careful planning.

[3] https://www.facebook.com/NZBees

[4] https://github.com/nzbees

[5] http://nzbees.github.io/

INDIGENOUS PERSPECTIVES A strong oral history in Aotearoa, suggests there is a wealth of valuable indigenous knowledge about the natural world, which is yet to be fully explored [209, 210].

- Alternative ways of *knowing*, can enrich understandings and an increased awareness may even enhance traditional scientific knowledge. The depth of these concepts are explored by Roberts [211, pg.741]:

  > *In common with other oral societies, New Zealand Māori constructed mental maps by means of which they made sense of their phenomenological world. Their cognitive template, called whakapapa, consists of a genealogical framework upon which spiritual, spatial, temporal and biophysical information about a particular place is located.*

  1. Whakapapa and other important oral histories surrounding native bees and their habitats, were not investigated in this research. However, in the future, research into these aspects could provide new and valuable insights.

- Traditional Māori knowledge systems were *founded* on empirical observations of the natural world. These were collated over time and space; and all aspects of nature were viewed as interrelated.

  1. In this research, recorded observations have not been discussed, although they may provide valuable information. For instance, native bees (*Leioproctus spp.*) were observed nesting during the months of April-May in 2015. This phenomenon has not previously been reported.
  2. Therefore, this research could be extended further. To investigate any changes occurring within communities of native bees. For example, to examine whether the nesting behaviours of some species are changing; and, if there are measurable changes, what the reasons might be?

Part IV

APPENDIX

A ACRONYMS

ARFF	Attribute-relation file format
CART	Classification and regression trees
CCC	Concordance correlation coefficient
CR	Compression ratio
CSV	Comma separated values
CV	Cross validation
DataONE	Data Observation Network for Earth
DPI	Dots per inch
DSLR	Digital single-lens reflex
EDM	Euclidean distance map
EXIF	Exchangeable image file format
FIJI	Fiji is Just ImageJ
GB	Gigabyte
GIS	Geographical information systems
GPS	Global positioning system
GUI	Graphical user interface
HD	Hard drive
HSB	Hue saturation and brightness
HSI	Hue saturation and intensity
IJM	ImageJ macro
J2P	JPEG 2000
JPEG	Joint photographic experts group
KNB	Knowledge Network for Bio-complexity
KNIME	Konstanz Information Miner
LCD	Liquid-crystal display

ODG	Open document graphics
ODP	Open document presentation
ODS	Open document spreadsheet
ODT	Open document text
oob	Out of bag error
PC	Personal computer
RAM	Random-access memory
RF	Random forest
RGB	Red green and blue
ROI	Region of interest
SD	Secure digital device
SP	Saturated pixels
SRM	Statistical region merging
SVM	Support vector machine
TIFF	Tagged image file format
TWS	Trainable Weka Segmentation
UAV	Unmanned aerial vehicle
WEKA	Waikato Environment for Knowledge Analysis

B

GLOSSARY

Erode — Shrinks the image, holes became larger and deletes small details.

Dilate — Enlarges object borders so holes become smaller.

Open — Erode and Dilate, smooths objects contours, removes isolated elements and breaks thin connections.

Close — Dilate and Erode, smooths objects contours, fill small holes and joins breaks.

C NOTATIONS

Compression ratio	$CR = \frac{\textit{original image data volume}}{\textit{compressed image data volume}}$	33
Kappa statistic	$\kappa = \frac{P(A) - P(E)}{1 - P(A)}$	42
Bias correction factor	$C_b = [\frac{(\nu + 1/\nu + u^2)}{2}]^{-}1$	78

D IMAGE EXIF DATA

Information stored in EXIF data from a single JPEG image of tiger beetle nests – refer to Figure 7.3.

File

Filename	P1010979.JPG
Filepath	~/ch3_imgs/test_format/JPEG
File size	4.14 MiB (4,337,982)
Note**	RAW file = 10.51 MiB (11,022,336) therefore CR = 2.54
Creation date/time	26/05/15 - 10:18 AM
Modified date/time	11/05/15 12:05 PM
Accessed date/time	26/05/15 - 10:19 AM
Rating	0
Colour Label	0

Image

Format	JPEG TrueColor (v1.1)
Width	4000
Height	2248
# of bits	24
Color model	*RGB*
Print size	141.11x79.30 cm, 55.56x31.22 inches
Compression	JPEG
Progressive mode	No
Sub-sampling	2x1,1x1,1x1
Estimated quality	*95*
Images/frames count	1
Origin	Top-Left

Camera

Info	*1/640s f/5.6 ISO100*
Model	*DMC-G1*
Date taken	*11/05/15 12:05 PM*

E

CODE

```
#Copy images of a certain type/size into a new folder
#!/bin/bash
recup_dir="${1%/}" [ -d "$recup_dir" ] ||
{echo "Usage:␣$0␣recup_dir";
echo "Mirror␣files␣from␣recup_dir␣into␣recup_dir.by_ext,␣organized␣by␣
    extension";
exit 1};
find "$recup_dir" -type f | while read k; do
ext="${k##*.}";
ext_dir="$recup_dir.by_ext/$ext";
[ -d "$ext_dir" ] || mkdir -p "$ext_dir";
echo "${k%/*}" ln "$k" "$ext_dir"; done

#Sort images into newly created date-time based folders.
$w_dir = '/home/nh/ext_dir';
$r_dir = '/home/nh/photos/'
$jhead_bin = '/usr/bin/jhead';
@rec_dirs = ‘ls ${w_dir} | grep recup_dir‘;
foreach $recup_dir (@rec_dirs) {print "Scanning␣${recup_dir}...";
chomp $recup_dir;
@photos_in_recup = ‘find ${w_dir}${recup_dir}/*jpg -type f -size +800k‘;
foreach $photo_file (@photos_in_recup)
{chomp $photo_file; print "IMG␣$photo_file␣in␣$recup_dir\n";
@exif = ‘$jhead_bin -v $photo_file‘;
print "$jhead_bin␣-v␣$photo_file\n";
foreach $line (@exif) {if ($line =~ /Time\s*:\s*([0-9]{4}):([0-9]{2})
    :([0-9]{2})\s[0-9:]{8}$/) {print "IMG␣$photo_file␣$1-$2-$3\n";
system("mkdir␣${r_dir}$1-$2-$3");
system("mv␣$photo_file␣$r_dir/$1-$2-$3/");last; }}}}
```

Listing E.1: Bash script to sort Photorec recovered images by extension; and into date-time based folders.

```
#JavaVM options passed from Fiji configuration file.
------------------------------------------
jre/bin/java -Xms3000m -Xmx4000m -Xincgc -cp -XX:MaxPermSize=256m
-XX:PermSize=256m
-XX:NewRatio=5
-XX:CMSTriggerRatio=20
-XX:+UseCompressedOops
-XX:+UseParNewGC
-XX:MinHeapFreeRatio=5
-XX:MaxHeapFreeRatio=10
-- ij.jar ij.ImageJ
------------------------------------------
#Bash script to turn off linux memory swap
------------------------------------------
cat /proc/sys/vm/swappiness
gksudo leafpad /etc/sysctl.conf
```

```
# Decrease swap usage to a more reasonable level--or turn off
vm.swappiness=10
# Improve cache management
vm.vfs_cache_pressure=50
------------------------------------------
```

Listing E.2: JavaVM memory options passed from Fiji's configuration files; and Ubuntu swap memory and cache adjustments. These were made to improve the stability of the operating system for image processing tasks.

```
#Use exiftool -tagsFromFile to get tages from files...cp copy from DIR -
    to DIR#
ref_id  //copy exif tags//copy_from Dir //save_to Dir
exiftool -tagsFromFile cp /071_sort_collections/imgdata/10/1_T110ANCM/C1
    /1/1_101124_124112.jpg
/081_sorted_collections/imgdata/10/1_T110ANCM/C1/1C1/1C1_101124_124112_1.
    jpg
exiftool -tagsFromFile cp
/071_sort_collections/imgdata/10/1_T110ANCM/C1/1/1_101124_124113.jpg /081
    _sorted_collections/imgdata/10/1_T110ANCM/C2/1C2/1C2_101124_124113_2
    .jpg
```

Listing E.3: Bash snippet to sort collections.

```
Collate images into stacks and save.
========================================
run("Image␣Sequence...","␣open=[t110_1c1]␣convert␣to␣rgb␣sort");
saveAs("TIFF",st110_1c1+"st110_1c1");

Enhance contrast.
========================================
run("Enhance␣Contrast...", "saturated=0.4␣process␣all");
saveAs("Tiff", \_1c1.tif");
```

Listing E.4: FIJI pre-processing example. Collate images into stacks and enhance contrast.

```
//open classified image stacks and post-process
run("Make␣Binary", "method=Default␣background=Default␣calculate");
run("Options...", "iterations=1␣count=1␣edm=Overwrite␣do=[Fill␣Holes]␣
    stack");
run("Options...", "iterations=10␣count=5␣pad␣edm=Overwrite␣do=Close␣stack
    ");
run("Options...", "iterations=1␣count=1␣edm=Overwrite␣do=[Fill␣Holes]␣
    stack");
run("Options...", "iterations=2␣count=3␣pad␣edm=Overwrite␣do=Open␣stack")
    ;
//count nests using three schemes....
run("Analyze␣Particles...","size=10-Infinity␣circularity=0.10-1.00␣show=[
    Overlay␣Outlines]
run("Analyze Particles...","size=15-Infinity circularity=0.10-1.00 show=[
    Overlay Outlines]
run("Analyze␣Particles...","size=20-Infinity␣circularity=0.10-1.00␣show=[
    Overlay␣Outlines]
```

Listing E.5: FIJI post-processing snippet.

```
//Contrast enhancement by saturated pixels
raw="/enhance_contrast/raw/"; //source and output dir
slice="/enhance_contrast/slice/";
stack="/enhance_contrast/stack/";
pre_process="/enhance_contrast/pre_process/";
montage="/enhance_contrast/montage/";
setBatchMode(true);
//open image and saturate pixels by 0.4, 40 and 90%
open(raw + "sc.tif");//scheme A
        //makeRectangle(154, 85, 30, 30);
        run("Enhance␣Contrast...", "saturated=0.4␣update");
                saveAs("TIFF", slice +"s1.tif");
open(raw + "sc.tif");//scheme B
        //makeRectangle(154, 85, 30, 30);
        run("Enhance␣Contrast...", "saturated=40␣update");
                saveAs("TIFF", slice +"s2.tif");
open(raw + "sc.tif");//scheme C
        //makeRectangle(154, 85, 30, 30);
        run("Enhance␣Contrast...", "saturated=90␣update");
                saveAs("TIFF", slice +"s3.tif");
        run("Collect␣Garbage");
//open adjusted images, measure histogram and save
open(slice +"s1.tif");makeRectangle(0, 0, 400, 400);
        run("Histogram", "stack"); run("Canvas␣Size...", "width=400␣
            height=400␣position=Center␣zero");
                        saveAs("TIFF", slice + "s4␣hist.tif");
open(slice +"s2.tif");makeRectangle(0, 0, 400, 400);
        run("Histogram", "stack"); run("Canvas␣Size...", "width=400␣
            height=400␣position=Center␣zero");
                        saveAs("TIFF", slice + "s5␣hist.tif");
open(slice +"s3.tif");makeRectangle(0, 0, 400, 400);
        run("Histogram", "stack"); run("Canvas␣Size...", "width=400␣
            height=400␣position=Center␣zero");
                        saveAs("TIFF", slice + "s6␣hist.tif");
setBatchMode(true);//do not display images
        run("Image␣Sequence...", "open=/tests/enhance_contrast/slice/0
            _0hist.tif␣sort");
                saveAs("Tiff", stack + "ecs.tif");
open(stack + "ecs.tif");
        run("Make␣Montage...", "columns=3␣rows=2␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label␣use");
                saveAs("Tiff", montage + "1ms.tif"); saveAs("PNG", docf +
                     "mec.png");
setBatchMode(false);run("Close␣All");//end
```

Listing E.6: FIJI saturated pixels example.

```
//Contrast enhancement for image stacks
raw="/pre_processing/raw/"; //source and output dir
slice="/pre_processing/slice/";
stack="/pre_processing/stack/";
pre_process="/pre_processing/pre_process/";
montage="/pre_processing/montage/";
setBatchMode(true);
//open image and apply contrast using 5 schemes
open(stack + "sc.tif");
                saveAs("TIFF", stack + "0_sec.tif"); //raw stack
```

```
open(stack + "sc.tif");//scheme 1
        run("Enhance␣Contrast...", "saturated=35");
                saveAs("TIFF", stack + "1␣sec.tif");
open(stack + "sc.tif");//scheme 2
        run("Enhance␣Contrast...", "saturated=35␣equalize␣process␣all");
                saveAs("TIFF", stack +"2␣sec.tif");
open(stack + "sc.tif");//scheme 3 //***USED
        run("Enhance␣Contrast...", "saturated=0.4␣process␣all");
                saveAs("TIFF", stack +"3␣sec.tif");
open(stack + "sc.tif");//scheme 4
        run("Enhance␣Contrast...", "saturated=35␣process␣all␣use");
                saveAs("TIFF", stack +"4␣sec.tif");
open(stack + "sc.tif");//scheme 5
        run("Enhance␣Contrast...", "saturated=35␣equalize␣process␣all␣use
            ");
                saveAs("TIFF", stack +"5␣sec.tif");
//open processed stacks and make montage
open(stack + "0␣sec.tif"); //original unchanged
        run("Make␣Montage...", "columns=2␣rows=2␣scale=0.50␣first=1␣last
            =7␣increment=2␣border=4␣font=20␣label␣use");
                saveAs("TIFF", slice + "0␣mscec.tif"); //
open(stack + "1␣sec.tif");//scheme 1
        run("Make␣Montage...", "columns=2␣rows=2␣scale=0.50␣first=1␣last
            =7␣increment=2␣border=4␣font=20␣label␣use");
                saveAs("TIFF", slice + "1␣mscec.tif");
open(stack + "2␣sec.tif");//scheme 2
        run("Make␣Montage...", "columns=2␣rows=2␣scale=0.50␣first=1␣last
            =7␣increment=2␣border=4␣font=20␣label␣use");
                saveAs("TIFF", slice +"2␣mscec.tif");
open(stack + "3␣sec.tif");//scheme 3 //***
        run("Make␣Montage...", "columns=2␣rows=2␣scale=0.50␣first=1␣last
            =7␣increment=2␣border=4␣font=20␣label␣use");
                saveAs("TIFF", slice +"3␣mscec.tif");
open(stack + "4␣sec.tif");//scheme 4
        run("Make␣Montage...", "columns=2␣rows=2␣scale=0.50␣first=1␣last
            =7␣increment=2␣border=4␣font=20␣label␣use");
                saveAs("TIFF", slice +"4␣mscec.tif");
open(stack + "5␣sec.tif");//scheme 5
        run("Make␣Montage...", "columns=2␣rows=2␣scale=0.50␣first=1␣last
            =7␣increment=2␣border=4␣font=20␣label");
                saveAs("TIFF", slice +"5␣mscec.tif");
//open processed stacks and measure histograms
open(stack + "0␣sec.tif"); //original unchanged
        run("Histogram", "stack"); run("Canvas␣Size...", "width=402␣
            height=402␣position=Center␣zero");
                saveAs("TIFF", slice + "0␣hist.tif");
open(stack + "1␣sec.tif");
        run("Histogram", "stack"); run("Canvas␣Size...", "width=402␣
            height=402␣␣position=Center␣zero");
                saveAs("TIFF", slice + "1␣hist.tif");
open(stack +"2␣sec.tif");
        run("Histogram", "stack"); run("Canvas␣Size...", "width=402␣
            height=402␣␣position=Center␣zero");
                saveAs("TIFF", slice + "2␣hist.tif");
open(stack +"3␣sec.tif");
        run("Histogram", "stack"); run("Canvas␣Size...", "width=402␣
            height=402␣␣position=Center␣zero");
```

```
                saveAs("TIFF", slice + "3_hist.tif");
open(stack +"4_sec.tif");
        run("Histogram", "stack"); run("Canvas␣Size...", "width=402␣
            height=402␣␣position=Center␣zero");
                saveAs("TIFF", slice + "4_hist.tif");
open(stack+"5_sec.tif");
        run("Histogram", "stack"); run("Canvas␣Size...", "width=402␣
            height=402␣␣position=Center␣zero");
                saveAs("TIFF", slice + "5_hist.tif"); run("Collect␣
                    Garbage");
//collate image stack and stack montage
        run("Image␣Sequence...", "open=/tests/pre_processing/slice/0_sec.
            tif␣sort");
                saveAs("Tiff", stack + "scec.tif");
open(stack + "scec.tif");
        run("Make␣Montage...", "columns=4␣rows=3␣scale=0.80␣first=1␣last
            =12␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "1mscec.tif"); saveAs("PNG",
                    docf + "mscec.png");
setBatchMode(false);run("Close␣All");//end
```

Listing E.7: FIJI stack pre-processing example.

```
//Try all thresholds (16 methods)
raw="/tryall/raw/"; //source and output dir
slice="/tryall/slice/";
stack="/tryall/stack/";
pre_process="/tryall/pre_process/";
logf="/tryall/log/";
montage="/tryall/montage/";
//open single raw image and pre-process
setBatchMode(true);
open(raw + "5.tif");//enhance contrast
        run("Enhance␣Contrast...", "saturated=0.4␣update");
                saveAs("TIFF", slice +"ec5.tif");
open(slice +"ec5.tif");//convert to 8 bit
        run("8-bit");
                saveAs("TIFF", slice +"gec5.tif");
open(slice +"gec5.tif");//Try all thresholds (16 methods)
        run("Auto␣Threshold", "method=[Try␣all]␣white␣show");
                saveAs("TIFF", slice +"mgec5.tif"); saveAs("PNG", docf +"
                    mgec5.tif");
selectWindow("Log");
        saveAs("Text", logf + "Log-tryall.txt");
setBatchMode(false);run("Close␣All");//end
```

Listing E.8: FIJI thresholding tests.

```
/Automatic thresholding on an image of a horizontal ground nest by six
    schemes,
//including some typical post processing operation examples.
//Started t.ijm at Fri Jun 05 16:56:46 NZST 2015
//Started t.ijm at Fri Jun 05 16:56:56 NZST 2015
//Process completed in 10 seconds.
raw="tests/t_six/raw/";
slice="tests/t_six/slice/";
stack="tests/t_six/stack/";
```

```
montage="/tests/t_six/montage/";
post_process="/tests/t_six/post_process/";
logf="/tests/t_six/log/";
docf= "/gfx/chp3/"; //for publishing images
setBatchMode(true);//do not display images
//open and apply automatic threshold method
open(raw + "5-gs.tif");
        run("Auto␣Threshold", "method=Default␣show");
                saveAs("Tiff", slice + "1_Default.tif");
open(raw + "5-gs.tif");
        run("Auto␣Threshold", "method=Huang␣show");
                saveAs("Tiff", slice + "2_Huang.tif");
open(raw + "5-gs.tif");
        run("Auto␣Threshold", "method=Mean␣show");
                saveAs("Tiff", slice + "3_Mean.tif");
open(raw + "5-gs.tif");
        run("Auto␣Threshold", "method=MinError(I)␣show");
                saveAs("Tiff", slice + "4_MinError.tif");
open(raw + "5-gs.tif");
        run("Auto␣Threshold", "method=Minimum␣show");
                saveAs("Tiff", slice + "5_Minimum.tif");
open(raw + "5-gs.tif");
        run("Auto␣Threshold", "method=Otsu␣show");
                saveAs("Tiff", slice + "6_Otsu.tif");
selectWindow("Log");
        saveAs("Text", logf + "Log-thresh-six.txt");
//post-process binary results
run("Image␣Sequence...", "open=/tests/t_six/slice/1-Default.tif␣sort");
        saveAs("Tiff", stack + "t-six.tif");
open(stack + "t-six.tif");
        run("Options...", "iterations=2␣count=2␣do=Open␣stack");//open
            operator
                saveAs("Tiff", post_process + "t-open-22.tif");
open(post_process + "t-open-22.tif");
        run("Options...", "iterations=1␣count=1␣do=[Fill␣Holes]␣stack");
            //fill holes
                saveAs("Tiff", post_process + "t-fh.tif");
open(post_process + "t-fh.tif");
        run("Options...", "iterations=2␣count=4␣pad␣do=Close␣stack");//
            close operator
                saveAs("Tiff", post_process + "t-close-24.tif");
open(post_process + "t-close-24.tif");
        run("Analyze␣Particles...", "size=500-50000␣circularity=0.1-0.9␣
            show=Outlines␣display␣clear␣summarize␣in_situ␣stack");//
            particle count
                saveAs("Tiff", post_process + "t-cb.tif");
selectWindow("Summary␣of␣t-close-24.tif");//save count summary
        saveAs("Text", post_process + "t-cb.txt");
run("Close␣All");
//make stack montage and save results
open(stack + "t-six.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "1mt-six.tif");saveAs("PNG",
                    docf + "1mt.png");
open(post_process + "t-open-22.tif");
```

```
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "2mt-open-22.tif");saveAs("PNG",
                     docf + "2mt.png");
open(post_process + "t-fh.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "3mt-fh.tif");saveAs("PNG", docf
                     + "3mt.png");
open(post_process + "t-close-24.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "4mt-close-24.tif");saveAs("PNG"
                    , docf + "4mt.png");
open(post_process + "t-cb.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "5mt-cb.tif");saveAs("PNG", docf
                     + "5mt.png");
setBatchMode(false);run("Close␣All");//end
```

Listing E.9: FIJI thresholding by six common methods.

```
//Binirization tests on representative sample images of active nests.
raw="/tests/challenge/raw/";
slice="/tests/challenge/slice/";
stack="/tests/challenge/stack/";
montage="/tests/challenge/montage/";
post_process="/tests/challenge/post_process/";
logf="/tests/challenge/log/";
docf= "/gfx/chp3/";
setBatchMode(true);//do not display images
//open and process stack
open(stack + "c.tif"); //convert to 8 bit
        run("8-bit");
                saveAs("Tiff", stack + "8bit.tif");
open(stack + "8bit.tif"); //threshold by Otsu method
        run("Auto␣Threshold", "method=Otsu␣stack");
                saveAs("Tiff", stack + "otsu.tif");
open(stack + "otsu.tif");//open operator
        run("Options...", "iterations=2␣count=2␣do=Open␣stack");
                saveAs("Tiff", stack + "open-11.tif");
open(stack + "open-11.tif"); //fill holes operator
        run("Options...", "iterations=1␣count=1␣do=[Fill␣Holes]␣stack");
                saveAs("Tiff", stack + "fh.tif");
open(stack + "fh.tif");//close operator
        run("Options...", "iterations=2␣count=4␣pad␣do=Close␣stack");
                saveAs("Tiff", stack + "close-43.tif");
open(stack + "close-43.tif");//particle count
        run("Analyze␣Particles...", "size=500-50000␣circularity=0.1-0.9␣
            show=Outlines␣display␣clear␣summarize␣in␣situ␣stack");
                saveAs("Tiff", stack + "cb.tif");
selectWindow("Summary␣of␣close-43.tif");//save count summary
                saveAs("Text", stack + "cp.txt");
//make stack montage and save results
open(stack + "c.tif");
```

```
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mc.tif"); saveAs("PNG", docf +
                     "0mc.png");
open(stack + "8bit.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "m8bit.tif"); saveAs("PNG", docf
                      + "1mc.png");
open(stack + "otsu.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "otsu.tif");saveAs("PNG", docf +
                      "2mc.png");
open(stack + "open-11.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mopen-11.tif");saveAs("PNG",
                     docf + "3mc.png");
open(stack + "fh.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mcfh.tif");saveAs("PNG", docf +
                      "4mc.png");
open(stack + "close-43.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mclose-43.tif");saveAs("PNG",
                     docf + "5mc.png");
open(stack + "cb.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mccb.tif"); saveAs("PNG", docf
                     + "6mc.png");
                setBatchMode(false); run("Close␣All"); //end
```

Listing E.10: FIJI otsu thresholding on variable nest images.

```
//Edge detection by Canny-Deriche filtering
raw="/tests/edge/raw/";//home dir
slice="/tests/edge/slice/";
stack="/tests/edge/stack/";
montage="/tests/edge/montage/";
post_process="/tests/edge/post_process/";
logf = "/tests/edge/log/";
docf= "/gfx/chp3/"; //for publishing images
//open and apply edge detection with different smoothing values
open(raw + "5-gs.tif");
        run("Deriche...", "alpha=1.0");run("8-bit");
                saveAs("Tiff", slice + "1.tif");
open(raw + "5-gs.tif");
        run("Deriche...", "alpha=0.9");run("8-bit");
                saveAs("Tiff", slice + "2.tif");
open(raw + "5-gs.tif");
        run("Deriche...", "alpha=0.75");run("8-bit");
                saveAs("Tiff", slice + "3.tif");
open(raw + "5-gs.tif");
```

```
        run("Deriche...", "alpha=0.6");run("8-bit");
                saveAs("Tiff", slice + "4.tif");
open(raw + "5-gs.tif");
        run("Deriche...", "alpha=0.45");run("8-bit");
                saveAs("Tiff", slice + "5.tif");
open(raw + "5-gs.tif");
        run("Deriche...", "alpha=0.15");run("8-bit");
                saveAs("Tiff", slice + "6.tif");
run("Close␣All");
//post-process binary results
run("Image␣Sequence...", "open=/tests/edge/slice/1.tif␣sort");
                saveAs("Tiff", stack + "edge.tif");
open(stack + "edge.tif");
        run("Find␣Edges", "stack"); //run global find edges plugin over
            entire image stack
        saveAs("Tiff", stack + "fe.tif");
open(stack + "fe.tif");
        run("Make␣Binary", "method=Minimum␣background=Default␣calculate␣
            list"); //invert binary
        saveAs("Tiff", stack + "min.tif");
open(stack + "min.tif");
        run("Options...", "iterations=3␣count=3␣do=Close␣stack");//close
            operator
                saveAs("Tiff", stack + "close-33.tif");
selectWindow("close-33.tif");
        run("Options...", "iterations=1␣count=1␣do=[Fill␣Holes]␣stack");
            //fill holes
                saveAs("Tiff", stack + "fh.tif");
selectWindow("fh.tif");
        run("Options...", "iterations=4␣count=3␣do=Open␣stack");//open
            operator
                saveAs("Tiff", stack + "open-43.tif");
selectWindow("open-43.tif");
        run("Analyze␣Particles...", "size=1000-50000␣circularity=0.1-0.9␣
            show=Outlines␣display␣clear␣summarize␣in␣situ␣stack"); //
            particle count
                saveAs("Tiff", stack + "cb.tif");
selectWindow("Summary␣of␣open-43.tif");
        saveAs("Text", logf + "cb.txt"); //save count summary
run("Close␣All");
//make stack montage and save results
setBatchMode(true);//do not display images
open(stack + "fe.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.8␣first=1␣last=6
            ␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "fe.tif");saveAs("PNG", docf + "
                    1medge.png");
open(stack + "min.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.8␣first=1␣last=6
            ␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "min.tif");saveAs("PNG", docf +
                    "2medge-b.png");
open(post_process + "close-33.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.8␣first=1␣last=6
            ␣increment=1␣border=4␣font=20␣label␣use");
                saveAs("Tiff", montage + "close-33.tif");saveAs("Tiff",
                    docf + "3medge-close.png");
```

```
open(post_process + "fh.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.8␣first=1␣last=6
            ␣increment=1␣border=4␣font=20␣label␣use");
                saveAs("Tiff", montage + "fh.tif");saveAs("PNG", docf + "
                    4medge-fh.png");
open(post_process + "open-43.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.8␣first=1␣last=6
            ␣increment=1␣border=4␣font=20␣label␣use");
                saveAs("Tiff", montage + "open-43.tif");saveAs("PNG",
                    docf + "5medge-open.png");
open(post_process + "cb.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.8␣first=1␣last=6
            ␣increment=1␣border=4␣font=20␣label␣use");
                saveAs("Tiff", montage + "cb.tif");saveAs("PNG", docf + "
                    6medge-cb.png");
                setBatchMode(false);run("Close␣All");//end
```

Listing E.11: Segmentation by edge detection.

```
//Statistical region merging (srm)
//Started srr.ijm at Sat Jun 06 20:05:38 NZST 2015
//Started srm.ijm at Sat Jun 06 20:05:49 NZST 2015
raw="/tests/srm/raw/";//home dir
slice="/tests/srm/slice/";
stack="/tests/srm/stack/";
montage="/tests/srm/montage/";
logf = "/tests/srm/log/";
docf= "/gfx/chp3/"; //for publishing images
//open and run srm with varying Q
open(raw + "5-gs.tif");
        run("Statistical␣Region␣Merging", "q=1␣showaverages");
                saveAs("Tiff", slice + "1.tif");
open(raw + "5-gs.tif");
        run("Statistical␣Region␣Merging", "q=2␣showaverages");
                saveAs("Tiff", slice + "2.tif");
open(raw + "5-gs.tif");
        run("Statistical␣Region␣Merging", "q=3␣showaverages");
                saveAs("Tiff", slice + "3.tif");
open(raw + "5-gs.tif");
        run("Statistical␣Region␣Merging", "q=6␣showaverages");
                saveAs("Tiff", slice + "4.tif");
open(raw + "5-gs.tif");
        run("Statistical␣Region␣Merging", "q=8␣showaverages");
                saveAs("Tiff", slice + "5.tif");
open(raw + "5-gs.tif");
        run("Statistical␣Region␣Merging", "q=16␣showaverages");
                saveAs("Tiff", slice + "6.tif");
run("Close␣All");
//post-process binary results
run("Image␣Sequence...", "open=/tests/srm/slice/1.tif␣sort");
                saveAs("Tiff", stack + "srm.tif");
open(stack + "srm.tif");
        run("8-bit");
         run("Make␣Binary", "Minimum␣background=Default"); //invert
             binary
        saveAs("Tiff", stack + "min.tif");
open(stack + "min.tif");
```

```
        run("Options...", "iterations=1␣count=1␣do=Open␣stack");//open
            operator
                saveAs("Tiff", stack + "open-11.tif");
selectWindow("open-11.tif");
        run("Options...", "iterations=1␣count=1␣do=[Fill␣Holes]␣stack");
            //fill holes
                saveAs("Tiff", stack+ "fh.tif");
selectWindow("fh.tif");
        run("Options...", "iterations=1␣count=1␣pad␣do=Close␣stack");//
            close operator
                saveAs("Tiff", stack + "close-11.tif");
selectWindow("close-11.tif");
        run("Analyze␣Particles...", "size=1000-50000␣circularity=0.1-0.9␣
            show=Outlines␣display␣clear␣summarize␣in␣situ␣stack"); //
            particle count
                saveAs("Tiff", stack + "cb.tif");
selectWindow("Summary␣of␣close-11.tif");
        saveAs("Text", logf + "srm-cb.txt"); //save count summary
//make stack montage and save results
setBatchMode(true);//do not display images
open(stack + "srm.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=30␣label");
                saveAs("Tiff", montage + "srm.tif");saveAs("PNG", docf +
                    "1msrm.png");
open(stack + "min.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=30␣label");
                saveAs("Tiff", montage + "min.tif");saveAs("PNG", docf +
                    "2msrm.png");
open(stack + "open-11.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=30␣label");
                saveAs("Tiff", montage + "open.tif");saveAs("PNG", docf +
                     "3msrm.png");
open(stack + "fh.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=30␣label");
                saveAs("Tiff", montage + "fh.tif");saveAs("PNG", docf + "
                    4msrm.png");
open(stack + "close-11.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=30␣label");
                saveAs("Tiff", montage + "close.tif");saveAs("PNG", docf
                    + "5msrm.png");
open(stack + "cb.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=30␣label");
                saveAs("Tiff", montage + "cb.tif");saveAs("PNG", docf + "
                    6msrm.png");
setBatchMode(false);run("Close␣All");//end
```

Listing E.12: Segmentation by region merging.

```
//Statistical region merging on represntative sample images
//Started srm_all.ijm at Sun Jun 07 14:56:55 NZST 2015
//Started srm_all.ijm at Sun Jun 07 14:57:02 NZST 2015
```

```
raw="/tests/srm_all/raw/";//home dir
slice="/tests/srm_all/slice/";
stack="/tests/srm_all/stack/";
montage="/tests/srm_all/montage/";
logf = "/tests/srm_all/log/";
setBatchMode(true);//do not display images
//open and run srm on representative images
open(raw + "1.tif");
        run("Statistical␣Region␣Merging", "q=16␣showaverages");
                saveAs("Tiff", slice + "1.tif");
open(raw + "2.tif");
        run("Statistical␣Region␣Merging", "q=16␣showaverages");
                saveAs("Tiff", slice + "2.tif");
open(raw + "3.tif");
        run("Statistical␣Region␣Merging", "q=16␣showaverages");
                saveAs("Tiff", slice + "3.tif");
open(raw + "4.tif");
        run("Statistical␣Region␣Merging", "q=16␣showaverages");
                saveAs("Tiff", slice + "4.tif");
open(raw + "5.tif");
        run("Statistical␣Region␣Merging", "q=16␣showaverages");
                saveAs("Tiff", slice + "5.tif");
open(raw + "6.tif");
        run("Statistical␣Region␣Merging", "q=16␣showaverages");
                saveAs("Tiff", slice + "6.tif");
run("Close␣All");
//post-process binary results
run("Image␣Sequence...", "open=/tests/srm_all/slice/1.tif␣sort");
                saveAs("Tiff", stack + "srm_all.tif");
open(stack + "srm_all.tif");
        run("8-bit");
         run("Make␣Binary", "Minimum␣background=Default"); //invert
             binary
        saveAs("Tiff", stack + "min_all.tif");
open(stack + "min_all.tif");
        run("Options...", "iterations=2␣count=2␣do=Open␣stack");//open
            operator
                saveAs("Tiff", stack + "open-22_all.tif");
open(stack + "open-22_all.tif");
        run("Options...", "iterations=1␣count=1␣do=[Fill␣Holes]␣stack");
            //fill holes
                saveAs("Tiff", stack+ "fh_all.tif");
open(stack + "fh_all.tif");
        run("Options...", "iterations=2␣count=4␣pad␣do=Close␣stack");//
            close operator
                saveAs("Tiff", stack + "close-24_all.tif");
open(stack + "close-24_all.tif");
        run("Analyze␣Particles...", "size=100-50000␣circularity=0.1-0.9␣
            show=Outlines␣display␣clear␣summarize␣in␣situ␣stack"); //
            particle count
                saveAs("Tiff", stack + "cb_all.tif");
//make stack montage and save results
open(stack + "srm_all.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.80␣first=1␣last
            =6␣increment=1␣border=4␣font=30␣label");
                saveAs("Tiff", montage + "srm_all.tif");saveAs("PNG",
                    docf + "1msrmall.png");
```

```
open(stack + "min_all.tif");
        run("Make_Montage...", "columns=6_rows=1_scale=0.80_first=1_last
            =6_increment=1_border=4_font=30_label");
                saveAs("Tiff", montage + "min.tif");saveAs("PNG", docf +
                    "2msrmall.png");
open(stack + "open-22_all.tif");
        run("Make_Montage...", "columns=6_rows=1_scale=0.80_first=1_last
            =6_increment=1_border=4_font=30_label");
                saveAs("Tiff", montage + "open.tif");saveAs("PNG", docf +
                     "3msrmall.png");
open(stack + "fh_all.tif");
        run("Make_Montage...", "columns=6_rows=1_scale=0.80_first=1_last
            =6_increment=1_border=4_font=30_label");
                saveAs("Tiff", montage + "fh.tif");saveAs("PNG", docf + "
                    4msrmall.png");
open(stack + "close-24_all.tif");
        run("Make_Montage...", "columns=6_rows=1_scale=0.80_first=1_last
            =6_increment=1_border=4_font=30_label");
                saveAs("Tiff", montage + "close.tif");saveAs("PNG", docf
                    + "5msrmall.png");
open(stack + "cb_all.tif");
        run("Make_Montage...", "columns=6_rows=1_scale=0.80_first=1_last
            =6_increment=1_border=4_font=30_label");
                saveAs("Tiff", montage + "cb.tif");saveAs("PNG", docf + "
                    6msrmall.png");
setBatchMode(false);run("Close_All");//end
```

Listing E.13: SRM on variable nest images.

```
//RF tests on representative sample images of active nests.
//Started r.ijm at Sat Jun 06 19:13:54 NZST 2015
//Started r.ijm at Sat Jun 06 19:14:28 NZST 2015
raw="/tests/rf/raw/";
slice="/tests/rf/slice/";
stack="/tests/rf/stack/";
montage="/tests/rf/montage/";
post_process="/tests/rf/post_process/";
logf="/tests/rf/log/";
docf= "gfx/chp3/";
//open and process stack
open(stack + "c.tif");
run("Trainable_Weka_Segmentation");wait(2000);
//select foreground active nest pixels Class 1
makeOval(203, 376, 4, 6);call("trainableSegmentation.Weka_Segmentation.
    addTrace", "0", "1");
makeOval(220, 353, 5, 5);call("trainableSegmentation.Weka_Segmentation.
    addTrace", "0", "2");
makeOval(86, 268, 4, 5);call("trainableSegmentation.Weka_Segmentation.
    addTrace", "0", "3");
makeOval(139, 337, 17, 7);call("trainableSegmentation.Weka_Segmentation.
    addTrace", "0", "4");
makeOval(210, 109, 12, 4);call("trainableSegmentation.Weka_Segmentation.
    addTrace", "0", "4");
makeOval(254, 146, 6, 10);call("trainableSegmentation.Weka_Segmentation.
    addTrace", "0", "5");
makeOval(287, 192, 9, 4);call("trainableSegmentation.Weka_Segmentation.
    addTrace", "0", "6");
```

```
//select background pixels Class 2
makeRectangle(15, 267, 47, 2);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "1");
makeRectangle(98, 85, 2, 18);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "1");
makeRectangle(91, 232, 3, 22);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "1");
makeRectangle(44, 354, 4, 15);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "2");
makeRectangle(12, 162, 5, 6);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "3");
makeRectangle(267, 268, 6, 5);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "5");
makeRectangle(99, 105, 2, 17);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "5");
makeRectangle(85, 134, 5, 14);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "6");
makeRectangle(216, 276, 2, 19);call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "6");
//select RF training parameters and filters for features stack.
call("trainableSegmentation.Weka_Segmentation.setFeature", "Hessian=false");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Sobel_filter=false");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Difference_of_gaussians=false");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Membrane_projections=false");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Mean=true");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Minimum=true");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Median=true");
call("trainableSegmentation.Weka_Segmentation.setFeature", "Structure=true");
call("trainableSegmentation.Weka_Segmentation.setMaximumSigma", "2.0");
call("trainableSegmentation.Weka_Segmentation.setMembranePatchSize", "1");
call("trainableSegmentation.Weka_Segmentation.setClassifier", "hr.irb.fastRandomForest.FastRandomForest", "-I␣50␣-K␣2␣-S␣-5571395");
//train classifier and save data
call("trainableSegmentation.Weka_Segmentation.trainClassifier");//train classifier
call("trainableSegmentation.Weka_Segmentation.saveClassifier", logf + "rf.model");//save classifier
call("trainableSegmentation.Weka_Segmentation.saveData",  logf + "rf.arff");//save data
call("trainableSegmentation.Weka_Segmentation.getResult");
        saveAs("Tiff", stack + "rf.tif");
selectWindow("Log");saveAs("Text", logf +"rf.txt");//save log
//post process
run("Collect␣Garbage"); setBatchMode(true);
open(stack + "rf.tif"); //convert to 8 bit
        run("8-bit");
                saveAs("Tiff", stack + "rf8bit.tif");
open(stack + "rf8bit.tif"); //threshold by Otsu method
        run("Make␣Binary", "method=Otsu␣background=Default");
```

```
                saveAs("Tiff", stack + "otsu.tif");
open(stack + "otsu.tif");//open operator
        run("Options...", "iterations=2␣count=2␣do=Open␣stack");
                saveAs("Tiff", stack + "open-22.tif");
open(stack + "open-22.tif"); //fill holes operator
        run("Options...", "iterations=1␣count=1␣do=[Fill␣Holes]␣stack");
                saveAs("Tiff", stack + "fh.tif");
open(stack + "fh.tif");//close operator
        run("Options...", "iterations=2␣count=4␣pad␣do=Close␣stack");
                saveAs("Tiff", stack + "close-24.tif");
open(stack + "close-24.tif");//particle count
        run("Analyze␣Particles...", "size=100-50000␣circularity=0.1-0.9␣
            show=Outlines␣display␣clear␣summarize␣in␣situ␣stack");
                saveAs("Tiff", stack + "cb.tif");
selectWindow("Summary␣of␣close-24.tif");//save count summary
                saveAs("Text", stack + "cp.txt");
//make stack montage and save results
open(stack + "rf.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label␣use");
                saveAs("Tiff", montage + "mrf.tif"); saveAs("PNG", docf +
                     "0mrf.png");
open(stack + "rf8bit.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mrf8bit.tif"); saveAs("PNG",
                    docf + "1mrf.png");
open(stack + "otsu.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mrfotsu.tif");saveAs("PNG",
                    docf + "2mrf.png");
open(stack + "open-22.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mrfopen-22.tif");saveAs("PNG",
                    docf + "3mrf.png");
open(stack + "fh.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mrffh.tif");saveAs("PNG", docf
                    + "4mrf.png");
open(stack + "close-24.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mrfclose-24.tif");saveAs("PNG",
                     docf + "5mrf.png");
open(stack + "cb.tif");
        run("Make␣Montage...", "columns=6␣rows=1␣scale=0.50␣first=1␣last
            =6␣increment=1␣border=4␣font=20␣label");
                saveAs("Tiff", montage + "mrfcb.tif"); saveAs("PNG", docf
                     + "6mrf.png");
        setBatchMode(false); run("Close␣All");//end
```

Listing E.14: TWS on variable nest images.

```
## load libraries used
```

```
library(knitr)
library(survival)
library(epiR)
library(ggplot2)

afirst <-read.table("A1.csv", header=TRUE, sep=",", na.strings="", dec=".
    ", strip.white=TRUE)
## Spearman rank correlation matrix table
srcc <- cor(afirst[,c("acfA1","cafA1")], use="complete")
kable(head(srcc[,1:2]), digits = 4, format = "latex")
## CC correlation plot and correlation matrix table
acmod <- afirst[,c("acfA1")] cauto <- afirst[,c("cafA1")]
cccfirst <- epi.ccc(cauto, acmod, ci = "z-transform", conf.level = 0.95)
firstr <- cccfirst$rho.c
kable(head(firstr[,1:3]), digits = 4, format = "latex")
tmpfirst <- epi.ccc(cauto, acmod, ci = "z-transform",conf.level = 0.95)
lab <- paste("ccc:␣␣", round(tmpfirst$rho.c[,1], digits = 2),"␣(95%␣CI␣",
     round(tmpfirst$rho.c[,2], digits = 2),"-",round(tmpfirst$rho.c[,3],
     digits = 2), ")", sep = "") z <- lm(cauto~acmod) par(pty = "s")
plot(jitter(cauto),jitter(acmod), xlim = c(0, 300), ylim = c(0,100), cex
    =1, xlab = "Automatic␣counts␣by␣threshold", ylab = "Automatic␣counts
    ␣by␣CF␣model", pch = 1)
abline(a = 0, b = 1, lty = 2) abline(z, lty = 1)
text(x = 200, y = 100, labels = lab)
```

Listing E.15: RStudio concordance correlation script for automatic counts by thresholding and the CF classifier.

```
## load libraries used
library(knitr)
library(car)
library(survival)
library(epiR)
library(methods)
library(ggplot2)
mic_mfc_ac_ob <-
  read.table("analysis/main/r/mic_mfc_ac_ob/csv/mic_mfc_ac_ob_4R.csv",
             header=TRUE, sep=",", na.strings="NA", dec=".", strip.white=
                 TRUE)

## Spearman rank correlation matrix table
m_ac <- cor(mic_mfc_ac_ob[,c("acf_t","mfc_t")], use="complete")
kable(head(m_ac[,1:2]), format = "markdown")

## Concordance correlation plot:
ac <- mic_mfc_ac_ob[,c("acf_t")]
mfc <- mic_mfc_ac_ob[,c("mfc_t")]

acmc.ccc <- epi.ccc(mfc, ac, ci = "z-transform",conf.level = 0.95)
rslts <- acmc.ccc$rho.c
rslts1 <- as.data.frame(rslts, row.names = NULL, responseName = "Rho",
    tringsAsFactors = TRUE)
kable(head(rslts1[,1:3]), format = "markdown")

lab <- paste("CCC:␣␣", round(acmc.ccc$rho.c[,1], digits = 2), "␣(95%␣CI␣"
    ,
             round(acmc.ccc$rho.c[,2], digits = 2), "␣-␣",
             round(acmc.ccc$rho.c[,3], digits = 2), ")", sep = "")
z <- lm(mfc ~ ac)
par(pty = "s")
plot(jitter(mfc),jitter(ac), xlim = c(0, 60), ylim = c(0,80), cex=1, xlab
     = "Manual␣field␣counts", ylab = "Automatic␣counts", pch = 1)
abline(a = 0, b = 1, lty = 2)
abline(z, lty = 1)
legend(x = "topleft", legend = c("Line␣of␣perfect␣concordance", "Reduced␣
    major␣axis"), lty = c(2,1), lwd = c(1,1), bty = "n")
text(x = 15, y = 69, labels = lab)
```

Listing E.16: Automatic and manual concordance correlation.

```
##Spearman rank correlation matrix table obs1 obs2
##Lin's Concordance correlation
library(knitr)
library(car)
library(survival)
library(epiR)
library(methods)
library(ggplot2)

mic_mfc_ac_ob <-
  read.table("analysis/r/mic_mfc_ac_ob/csv/mic_mfc_ac_ob_4R.csv",
             header=TRUE, sep=",", na.strings="NA", dec=".", strip.white=
                 TRUE)

## Spearman rank correlation matrix table

obs <- cor(mic_mfc_ac_ob[,c("mic_ob1","mic_ob2")], use="complete")
kable(head(obs[,1:2]), format = "markdown")


## Spearman rank correlation
##scatterplot(mic_ob1~mic_ob2, reg.line=lm, smooth=FALSE, spread=FALSE,
    boxplots=FALSE, span=0.5, jitter=list(x=1, y=1),
##            cex=0.5, cex.axis=1, cex.lab=1, data=mic_mfc_ac_ob)
##par(pty = "s")

## Concordance correlation plot:
mic_ob1t <- mic_mfc_ac_ob[,c("mic_ob1")]
mic_ob2t <- mic_mfc_ac_ob[,c("mic_ob2")]

micob.ccc <- epi.ccc(mic_ob1t, mic_ob2t, ci = "z-transform",
                     conf.level = 0.95)

lab <- paste("CCC:␣", round(micob.ccc$rho.c[,1], digits = 2), "␣(95%␣CI␣"
    ,
             round(micob.ccc$rho.c[,2], digits = 2), "␣-␣",
             round(micob.ccc$rho.c[,3], digits = 2), ")", sep = "")

z <- lm(mic_ob1t~mic_ob2t)
par(pty = "s")
plot(jitter(mic_ob1t),jitter(mic_ob2t), xlim = c(0, 60), ylim = c(0,80),
    cex=1, xlab = "Manual␣image␣counts␣by␣observer␣1", ylab = "Manual␣
    image␣counts␣by␣observer␣2", pch = 1)
abline(a = 0, b = 1, lty = 2)
abline(z, lty = 1)
legend(x = "topleft", legend = c("Line␣of␣perfect␣concordance",
                                 "Reduced␣major␣axis"), lty = c(2,1), lwd
                                      = c(1,1), bty = "n")
text(x = 14, y = 70, labels = lab)
```

Listing E.17: Image counts by two observsers.

F

RAW FIELD DATA

Key:	Count data records Table F.1 below.
rec_id:	Unique record identifier
site:	Monitoring locations
	1-Mt. Tiger
	2-Mt. Parihaka
	3-Memorial Drive
y/m/d:	Year, month, day
dayc:	Collection season
imgc:	Image collection 1 of 4.
grid:	Quadrat 1 of 4
mfc_t:	Manual field count (total)

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
1	1	10	11	23	1	1	1	1
2	1	10	11	24	2	1	1	0
3	1	10	11	26	3	1	1	1
4	1	10	11	28	4	1	1	16
5	1	10	11	29	5	1	1	13
6	1	10	11	30	6	1	1	29
7	1	10	12	1	7	1	1	16
8	1	10	12	2	8	1	1	15
9	1	10	12	5	9	1	1	2
10	1	10	12	6	10	1	1	3
11	1	10	12	8	11	1	1	1
12	1	10	12	9	12	1	1	2
13	1	10	12	10	13	1	1	5
14	1	10	12	11	14	1	1	1
15	1	10	12	12	15	1	1	6
16	1	10	12	13	16	1	1	20
17	1	10	11	23	1	1	2	1
18	1	10	11	24	2	1	2	2

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
19	1	10	11	26	3	1	2	0
20	1	10	11	28	4	1	2	25
21	1	10	11	29	5	1	2	2
22	1	10	11	30	6	1	2	30
23	1	10	12	1	7	1	2	17
24	1	10	12	2	8	1	2	16
25	1	10	12	5	9	1	2	4
26	1	10	12	6	10	1	2	10
27	1	10	12	8	11	1	2	2
28	1	10	12	9	12	1	2	12
29	1	10	12	10	13	1	2	12
30	1	10	12	11	14	1	2	1
31	1	10	12	12	15	1	2	10
32	1	10	12	13	16	1	2	1
33	1	10	11	23	1	1	3	1
34	1	10	11	24	2	1	3	40
35	1	10	11	26	3	1	3	1
36	1	10	11	28	4	1	3	30
37	1	10	11	29	5	1	3	20
38	1	10	11	30	6	1	3	25
39	1	10	12	1	7	1	3	2
40	1	10	12	2	8	1	3	2
41	1	10	12	5	9	1	3	2
42	1	10	12	6	10	1	3	10
43	1	10	12	8	11	1	3	5
44	1	10	12	9	12	1	3	1
45	1	10	12	10	13	1	3	4
46	1	10	12	11	14	1	3	1
47	1	10	12	12	15	1	3	20
48	1	10	12	13	16	1	3	21
49	1	10	11	23	1	1	4	3
50	1	10	11	24	2	1	4	50
51	1	10	11	26	3	1	4	0
52	1	10	11	28	4	1	4	30
53	1	10	11	29	5	1	4	12
54	1	10	11	30	6	1	4	10

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
55	1	10	12	1	7	1	4	0
56	1	10	12	2	8	1	4	2
57	1	10	12	5	9	1	4	2
58	1	10	12	6	10	1	4	5
59	1	10	12	8	11	1	4	5
60	1	10	12	9	12	1	4	14
61	1	10	12	10	13	1	4	4
62	1	10	12	11	14	1	4	5
63	1	10	12	12	15	1	4	1
64	1	10	12	13	16	1	4	2
65	2	10	11	23	1	1	1	1
66	2	10	11	24	2	1	1	0
67	2	10	11	25	3	1	1	1
68	2	10	11	26	4	1	1	2
69	2	10	11	28	5	1	1	1
70	2	10	11	29	6	1	1	3
71	2	10	11	30	7	1	1	2
72	2	10	12	1	8	1	1	2
73	2	10	12	2	9	1	1	2
74	2	10	12	5	10	1	1	1
75	2	10	12	6	11	1	1	0
76	2	10	12	8	12	1	1	1
77	2	10	12	9	13	1	1	1
78	2	10	12	10	14	1	1	1
79	2	10	12	11	15	1	1	0
80	2	10	12	12	16	1	1	1
81	2	10	11	23	1	1	2	1
82	2	10	11	24	2	1	2	15
83	2	10	11	25	3	1	2	5
84	2	10	11	26	4	1	2	2
85	2	10	11	28	5	1	2	3
86	2	10	11	29	6	1	2	1
87	2	10	11	30	7	1	2	2
88	2	10	12	1	8	1	2	7
89	2	10	12	2	9	1	2	6
90	2	10	12	5	10	1	2	5

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
91	2	10	12	6	11	1	2	1
92	2	10	12	8	12	1	2	2
93	2	10	12	9	13	1	2	1
94	2	10	12	10	14	1	2	3
95	2	10	12	11	15	1	2	4
96	2	10	12	12	16	1	2	4
97	2	10	11	23	1	1	3	8
98	2	10	11	24	2	1	3	13
99	2	10	11	25	3	1	3	13
100	2	10	11	26	4	1	3	10
101	2	10	11	28	5	1	3	1
102	2	10	11	29	6	1	3	7
103	2	10	11	30	7	1	3	9
104	2	10	12	1	8	1	3	1
105	2	10	12	2	9	1	3	10
106	2	10	12	5	10	1	3	6
107	2	10	12	6	11	1	3	2
108	2	10	12	8	12	1	3	1
109	2	10	12	9	13	1	3	10
110	2	10	12	10	14	1	3	11
111	2	10	12	11	15	1	3	2
112	2	10	12	12	16	1	3	5
113	2	10	11	23	1	1	4	8
114	2	10	11	24	2	1	4	8
115	2	10	11	25	3	1	4	10
116	2	10	11	26	4	1	4	9
117	2	10	11	28	5	1	4	1
118	2	10	11	29	6	1	4	1
119	2	10	11	30	7	1	4	7
120	2	10	12	1	8	1	4	7
121	2	10	12	2	9	1	4	7
122	2	10	12	5	10	1	4	0
123	2	10	12	6	11	1	4	0
124	2	10	12	8	12	1	4	1
125	2	10	12	9	13	1	4	1
126	2	10	12	10	14	1	4	2

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
127	2	10	12	11	15	1	4	5
128	2	10	12	12	16	1	4	4
129	1	11	11	11	1	1	1	3
130	1	11	11	12	2	1	1	10
131	1	11	11	13	3	1	1	10
132	1	11	11	14	4	1	1	2
133	1	11	11	15	5	1	1	12
134	1	11	11	16	6	1	1	10
135	1	11	11	17	7	1	1	15
136	1	11	11	18	8	1	1	10
137	1	11	11	21	9	1	1	9
138	1	11	11	22	10	1	1	40
139	1	11	11	23	11	1	1	0
140	1	11	11	24	12	1	1	20
141	1	11	11	25	13	1	1	1
142	1	11	11	28	14	1	1	3
143	1	11	12	7	15	1	1	15
144	1	11	12	9	16	1	1	2
145	1	11	11	11	1	1	2	12
146	1	11	11	12	2	1	2	10
147	1	11	11	13	3	1	2	15
148	1	11	11	14	4	1	2	3
149	1	11	11	15	5	1	2	10
150	1	11	11	16	6	1	2	5
151	1	11	11	17	7	1	2	2
152	1	11	11	18	8	1	2	2
153	1	11	11	21	9	1	2	2
154	1	11	11	22	10	1	2	12
155	1	11	11	23	11	1	2	2
156	1	11	11	24	12	1	2	2
157	1	11	11	25	13	1	2	4
158	1	11	11	28	14	1	2	7
159	1	11	12	7	15	1	2	7
160	1	11	12	9	16	1	2	1
161	1	11	11	11	1	1	3	40
162	1	11	11	12	2	1	3	0

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
163	1	11	11	13	3	1	3	2
164	1	11	11	14	4	1	3	3
165	1	11	11	15	5	1	3	3
166	1	11	11	16	6	1	3	2
167	1	11	11	17	7	1	3	3
168	1	11	11	18	8	1	3	1
169	1	11	11	21	9	1	3	1
170	1	11	11	22	10	1	3	1
171	1	11	11	23	11	1	3	1
172	1	11	11	24	12	1	3	1
173	1	11	11	25	13	1	3	14
174	1	11	11	28	14	1	3	25
175	1	11	12	7	15	1	3	5
176	1	11	12	9	16	1	3	1
177	1	11	11	11	1	1	4	16
178	1	11	11	12	2	1	4	2
179	1	11	11	13	3	1	4	1
180	1	11	11	14	4	1	4	4
181	1	11	11	15	5	1	4	4
182	1	11	11	16	6	1	4	3
183	1	11	11	17	7	1	4	3
184	1	11	11	18	8	1	4	2
185	1	11	11	21	9	1	4	2
186	1	11	11	22	10	1	4	2
187	1	11	11	23	11	1	4	2
188	1	11	11	24	12	1	4	2
189	1	11	11	25	13	1	4	2
190	1	11	11	28	14	1	4	3
191	1	11	12	7	15	1	4	1
192	1	11	12	9	16	1	4	1
193	2	11	11	11	1	1	1	3
194	2	11	11	12	2	1	1	1
195	2	11	11	13	3	1	1	8
196	2	11	11	14	4	1	1	8
197	2	11	11	15	5	1	1	1
198	2	11	11	16	6	1	1	6

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
199	2	11	11	17	7	1	1	1
200	2	11	11	18	8	1	1	1
201	2	11	11	21	9	1	1	1
202	2	11	11	22	10	1	1	1
203	2	11	11	23	11	1	1	1
204	2	11	11	24	12	1	1	1
205	2	11	11	25	13	1	1	0
206	2	11	11	28	14	1	1	6
207	2	11	12	7	15	1	1	3
208	2	11	12	9	16	1	1	11
209	2	11	11	11	1	1	2	7
210	2	11	11	12	2	1	2	1
211	2	11	11	13	3	1	2	1
212	2	11	11	14	4	1	2	6
213	2	11	11	15	5	1	2	1
214	2	11	11	16	6	1	2	1
215	2	11	11	17	7	1	2	2
216	2	11	11	18	8	1	2	1
217	2	11	11	21	9	1	2	4
218	2	11	11	22	10	1	2	1
219	2	11	11	23	11	1	2	5
220	2	11	11	24	12	1	2	1
221	2	11	11	25	13	1	2	1
222	2	11	11	28	14	1	2	1
223	2	11	12	7	15	1	2	1
224	2	11	12	9	16	1	2	1
225	2	11	11	11	1	1	3	2
226	2	11	11	12	2	1	3	5
227	2	11	11	13	3	1	3	1
228	2	11	11	14	4	1	3	1
229	2	11	11	15	5	1	3	0
230	2	11	11	16	6	1	3	2
231	2	11	11	17	7	1	3	2
232	2	11	11	18	8	1	3	2
233	2	11	11	21	9	1	3	3
234	2	11	11	22	10	1	3	3

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
235	2	11	11	23	11	1	3	1
236	2	11	11	24	12	1	3	3
237	2	11	11	25	13	1	3	4
238	2	11	11	28	14	1	3	3
239	2	11	12	7	15	1	3	1
240	2	11	12	9	16	1	3	4
241	2	11	11	11	1	1	4	1
242	2	11	11	12	2	1	4	0
243	2	11	11	13	3	1	4	5
244	2	11	11	14	4	1	4	5
245	2	11	11	15	5	1	4	3
246	2	11	11	16	6	1	4	4
247	2	11	11	17	7	1	4	3
248	2	11	11	18	8	1	4	5
249	2	11	11	21	9	1	4	10
250	2	11	11	22	10	1	4	12
251	2	11	11	23	11	1	4	3
252	2	11	11	24	12	1	4	4
253	2	11	11	25	13	1	4	6
254	2	11	11	28	14	1	4	1
255	2	11	12	7	15	1	4	1
256	2	11	12	9	16	1	4	1
257	3	11	11	11	1	1	1	1
258	3	11	11	12	2	1	1	1
259	3	11	11	13	3	1	1	1
260	3	11	11	14	4	1	1	1
261	3	11	11	15	5	1	1	0
262	3	11	11	16	6	1	1	0
263	3	11	11	17	7	1	1	0
264	3	11	11	18	8	1	1	2
265	3	11	11	21	9	1	1	5
266	3	11	11	22	10	1	1	1
267	3	11	11	23	11	1	1	1
268	3	11	11	24	12	1	1	1
269	3	11	11	28	13	1	1	3
270	3	11	12	7	14	1	1	0

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
271	3	11	12	9	15	1	1	2
272	3	11	12	20	16	1	1	2
273	3	11	11	11	1	1	2	3
274	3	11	11	12	2	1	2	5
275	3	11	11	13	3	1	2	5
276	3	11	11	14	4	1	2	6
277	3	11	11	15	5	1	2	4
278	3	11	11	16	6	1	2	5
279	3	11	11	17	7	1	2	6
280	3	11	11	18	8	1	2	5
281	3	11	11	21	9	1	2	15
282	3	11	11	22	10	1	2	7
283	3	11	11	23	11	1	2	1
284	3	11	11	24	12	1	2	4
285	3	11	11	25	13	1	2	8
286	3	11	11	28	14	1	2	5
287	3	11	12	7	15	1	2	3
288	3	11	12	9	16	1	2	7
289	3	11	11	11	1	1	3	2
290	3	11	11	12	2	1	3	1
291	3	11	11	13	3	1	3	7
292	3	11	11	14	4	1	3	3
293	3	11	11	15	5	1	3	0
294	3	11	11	16	6	1	3	2
295	3	11	11	17	7	1	3	1
296	3	11	11	18	8	1	3	5
297	3	11	11	21	9	1	3	5
298	3	11	11	22	10	1	3	2
299	3	11	11	23	11	1	3	2
300	3	11	11	24	12	1	3	2
301	3	11	11	25	13	1	3	1
302	3	11	11	28	14	1	3	10
303	3	11	12	7	15	1	3	7
304	3	11	12	9	16	1	3	10
305	3	11	11	11	1	1	4	8
306	3	11	11	12	2	1	4	0

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
307	3	11	11	13	3	1	4	10
308	3	11	11	14	4	1	4	14
309	3	11	11	15	5	1	4	10
310	3	11	11	16	6	1	4	5
311	3	11	11	17	7	1	4	1
312	3	11	11	18	8	1	4	2
313	3	11	11	21	9	1	4	0
314	3	11	11	22	10	1	4	10
315	3	11	11	23	11	1	4	1
316	3	11	11	24	12	1	4	5
317	3	11	11	25	13	1	4	1
318	3	11	11	28	14	1	4	15
319	3	11	12	7	15	1	4	1
320	3	11	12	9	16	1	4	15
321	1	12	11	9	1	1	1	0
322	1	12	11	10	2	1	1	5
323	1	12	11	11	3	1	1	3
324	1	12	11	12	4	1	1	4
325	1	12	11	13	5	1	1	10
326	1	12	11	14	6	1	1	2
327	1	12	11	15	7	1	1	2
328	1	12	11	16	8	1	1	1
329	1	12	11	18	9	1	1	4
330	1	12	11	19	10	1	1	1
331	1	12	12	4	11	1	1	23
332	1	12	12	11	12	1	1	6
333	1	12	12	12	13	1	1	10
334	1	12	11	9	1	1	2	1
335	1	12	11	10	2	1	2	5
336	1	12	11	11	3	1	2	2
337	1	12	11	12	4	1	2	2
338	1	12	11	13	5	1	2	20
339	1	12	11	14	6	1	2	1
340	1	12	11	15	7	1	2	5
341	1	12	11	16	8	1	2	15
342	1	12	11	18	9	1	2	5

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
343	1	12	11	19	10	1	2	10
344	1	12	12	4	11	1	2	17
345	1	12	12	11	12	1	2	36
346	1	12	12	12	13	1	2	40
347	1	12	11	9	1	1	3	1
348	1	12	11	10	2	1	3	1
349	1	12	11	11	3	1	3	1
350	1	12	11	12	4	1	3	2
351	1	12	11	13	5	1	3	2
352	1	12	11	14	6	1	3	1
353	1	12	11	15	7	1	3	1
354	1	12	11	16	8	1	3	1
355	1	12	11	18	9	1	3	1
356	1	12	11	19	10	1	3	0
357	1	12	12	4	11	1	3	22
358	1	12	12	11	12	1	3	7
359	1	12	12	12	13	1	3	5
360	1	12	11	9	1	1	4	1
361	1	12	11	10	2	1	4	1
362	1	12	11	11	3	1	4	1
363	1	12	11	12	4	1	4	0
364	1	12	11	13	5	1	4	1
365	1	12	11	14	6	1	4	1
366	1	12	11	15	7	1	4	1
367	1	12	11	16	8	1	4	0
368	1	12	11	18	9	1	4	1
369	1	12	11	19	10	1	4	1
370	1	12	12	4	11	1	4	3
371	1	12	12	11	12	1	4	12
372	1	12	12	12	13	1	4	10
373	2	12	11	9	1	1	1	2
374	2	12	11	10	2	1	1	2
375	2	12	11	11	3	1	1	2
376	2	12	11	12	4	1	1	2
377	2	12	11	13	5	1	1	1
378	2	12	11	14	6	1	1	2

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
379	2	12	11	15	7	1	1	2
380	2	12	11	16	8	1	1	2
381	2	12	11	18	9	1	1	2
382	2	12	11	19	10	1	1	0
383	2	12	12	4	11	1	1	3
384	2	12	12	11	12	1	1	4
385	2	12	12	12	13	1	1	4
386	2	12	11	9	1	1	2	1
387	2	12	11	10	2	1	2	1
388	2	12	11	11	3	1	2	1
389	2	12	11	12	4	1	2	1
390	2	12	11	13	5	1	2	2
391	2	12	11	14	6	1	2	2
392	2	12	11	15	7	1	2	2
393	2	12	11	16	8	1	2	2
394	2	12	11	18	9	1	2	2
395	2	12	11	19	10	1	2	0
396	2	12	12	4	11	1	2	5
397	2	12	12	11	12	1	2	5
398	2	12	12	12	13	1	2	5
399	2	12	11	9	1	1	3	2
400	2	12	11	10	2	1	3	2
401	2	12	11	11	3	1	3	1
402	2	12	11	12	4	1	3	1
403	2	12	11	13	5	1	3	1
404	2	12	11	14	6	1	3	1
405	2	12	11	15	7	1	3	0
406	2	12	11	16	8	1	3	1
407	2	12	11	18	9	1	3	0
408	2	12	11	19	10	1	3	0
409	2	12	12	4	11	1	3	2
410	2	12	12	11	12	1	3	2
411	2	12	12	12	13	1	3	2
412	2	12	11	9	1	1	4	2
413	2	12	11	10	2	1	4	2
414	2	12	11	11	3	1	4	2

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
415	2	12	11	12	4	1	4	2
416	2	12	11	13	5	1	4	2
417	2	12	11	14	6	1	4	2
418	2	12	11	15	7	1	4	0
419	2	12	11	16	8	1	4	0
420	2	12	11	18	9	1	4	1
421	2	12	11	19	10	1	4	0
422	2	12	12	4	11	1	4	3
423	2	12	12	11	12	1	4	5
424	2	12	12	12	13	1	4	3
425	3	12	11	9	1	1	1	1
426	3	12	11	10	2	1	1	5
427	3	12	11	11	3	1	1	5
428	3	12	11	12	4	1	1	3
429	3	12	11	13	5	1	1	4
430	3	12	11	14	6	1	1	5
431	3	12	11	15	7	1	1	4
432	3	12	11	16	8	1	1	2
433	3	12	11	18	9	1	1	4
434	3	12	11	19	10	1	1	5
435	3	12	12	4	11	1	1	4
436	3	12	12	11	12	1	1	2
437	3	12	12	12	13	1	1	2
438	3	12	11	9	1	1	2	1
439	3	12	11	10	2	1	2	1
440	3	12	11	11	3	1	2	2
441	3	12	11	12	4	1	2	1
442	3	12	11	13	5	1	2	0
443	3	12	11	14	6	1	2	0
444	3	12	11	15	7	1	2	0
445	3	12	11	16	8	1	2	2
446	3	12	11	18	9	1	2	0
447	3	12	11	19	10	1	2	1
448	3	12	12	4	11	1	2	0
449	3	12	12	11	12	1	2	1
450	3	12	12	12	13	1	2	1

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
451	3	12	11	9	1	1	3	1
452	3	12	11	10	2	1	3	2
453	3	12	11	11	3	1	3	5
454	3	12	11	12	4	1	3	2
455	3	12	11	13	5	1	3	5
456	3	12	11	14	6	1	3	1
457	3	12	11	15	7	1	3	1
458	3	12	11	16	8	1	3	8
459	3	12	11	18	9	1	3	1
460	3	12	11	19	10	1	3	3
461	3	12	12	4	11	1	3	1
462	3	12	12	11	12	1	3	1
463	3	12	12	12	13	1	3	1
464	3	12	11	9	1	1	4	1
465	3	12	11	10	2	1	4	1
466	3	12	11	11	3	1	4	7
467	3	12	11	12	4	1	4	8
468	3	12	11	13	5	1	4	2
469	3	12	11	14	6	1	4	4
470	3	12	11	15	7	1	4	4
471	3	12	11	16	8	1	4	0
472	3	12	11	18	9	1	4	5
473	3	12	11	19	10	1	4	5
474	3	12	12	4	11	1	4	3
475	3	12	12	11	12	1	4	3
476	3	12	12	12	13	1	4	6
477	1	13	11	18	1	1	1	5
478	1	13	11	19	2	1	1	5
479	1	13	11	21	3	1	1	10
480	1	13	11	22	4	1	1	1
481	1	13	11	23	5	1	1	5
482	1	13	11	24	6	1	1	2
483	1	13	11	25	7	1	1	4
484	1	13	11	26	8	1	1	5
485	1	13	11	28	9	1	1	2
486	1	13	11	30	10	1	1	5

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
487	1	13	11	18	1	1	2	4
488	1	13	11	19	2	1	2	5
489	1	13	11	21	3	1	2	5
490	1	13	11	22	4	1	2	3
491	1	13	11	23	5	1	2	3
492	1	13	11	24	6	1	2	4
493	1	13	11	25	7	1	2	3
494	1	13	11	26	8	1	2	5
495	1	13	11	28	9	1	2	3
496	1	13	11	30	10	1	2	6
497	1	13	11	18	1	1	3	1
498	1	13	11	19	2	1	3	1
499	1	13	11	21	3	1	3	1
500	1	13	11	22	4	1	3	1
501	1	13	11	23	5	1	3	2
502	1	13	11	24	6	1	3	3
503	1	13	11	25	7	1	3	10
504	1	13	11	26	8	1	3	4
505	1	13	11	28	9	1	3	15
506	1	13	11	30	10	1	3	17
507	1	13	11	18	1	1	4	4
508	1	13	11	19	2	1	4	10
509	1	13	11	21	3	1	4	1
510	1	13	11	22	4	1	4	1
511	1	13	11	23	5	1	4	5
512	1	13	11	24	6	1	4	4
513	1	13	11	25	7	1	4	5
514	1	13	11	26	8	1	4	12
515	1	13	11	28	9	1	4	7
516	1	13	11	30	10	1	4	5
517	2	13	11	18	1	1	1	2
518	2	13	11	19	2	1	1	2
519	2	13	11	21	3	1	1	2
520	2	13	11	22	4	1	1	2
521	2	13	11	23	5	1	1	2
522	2	13	11	24	6	1	1	1

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
523	2	13	11	25	7	1	1	2
524	2	13	11	26	8	1	1	1
525	2	13	11	28	9	1	1	4
526	2	13	11	30	10	1	1	4
527	2	13	11	18	1	1	2	3
528	2	13	11	19	2	1	2	2
529	2	13	11	21	3	1	2	2
530	2	13	11	22	4	1	2	2
531	2	13	11	23	5	1	2	4
532	2	13	11	24	6	1	2	3
533	2	13	11	25	7	1	2	4
534	2	13	11	26	8	1	2	2
535	2	13	11	28	9	1	2	6
536	2	13	11	30	10	1	2	4
537	2	13	11	18	1	1	3	2
538	2	13	11	19	2	1	3	2
539	2	13	11	21	3	1	3	1
540	2	13	11	22	4	1	3	1
541	2	13	11	23	5	1	3	1
542	2	13	11	24	6	1	3	5
543	2	13	11	25	7	1	3	5
544	2	13	11	26	8	1	3	2
545	2	13	11	28	9	1	3	1
546	2	13	11	30	10	1	3	0
547	2	13	11	18	1	1	4	3
548	2	13	11	19	2	1	4	6
549	2	13	11	21	3	1	4	7
550	2	13	11	22	4	1	4	7
551	2	13	11	23	5	1	4	5
552	2	13	11	24	6	1	4	4
553	2	13	11	25	7	1	4	4
554	2	13	11	26	8	1	4	2
555	2	13	11	28	9	1	4	5
556	2	13	11	30	10	1	4	0
557	3	13	11	18	1	1	1	1
558	3	13	11	19	2	1	1	1

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
559	3	13	11	21	3	1	1	0
560	3	13	11	22	4	1	1	2
561	3	13	11	23	5	1	1	1
562	3	13	11	24	6	1	1	1
563	3	13	11	25	7	1	1	1
564	3	13	11	26	8	1	1	1
565	3	13	11	28	9	1	1	1
566	3	13	11	29	10	1	1	1
567	3	13	11	18	1	1	2	3
568	3	13	11	19	2	1	2	4
569	3	13	11	21	3	1	2	2
570	3	13	11	22	4	1	2	1
571	3	13	11	23	5	1	2	2
572	3	13	11	24	6	1	2	1
573	3	13	11	25	7	1	2	2
574	3	13	11	26	8	1	2	1
575	3	13	11	28	9	1	2	1
576	3	13	11	29	10	1	2	2
577	3	13	11	18	1	1	3	1
578	3	13	11	19	2	1	3	6
579	3	13	11	21	3	1	3	3
580	3	13	11	22	4	1	3	2
581	3	13	11	23	5	1	3	4
582	3	13	11	24	6	1	3	3
583	3	13	11	25	7	1	3	6
584	3	13	11	26	8	1	3	5
585	3	13	11	28	9	1	3	6
586	3	13	11	29	10	1	3	8
587	3	13	11	18	1	1	4	2
588	3	13	11	19	2	1	4	1
589	3	13	11	21	3	1	4	2
590	3	13	11	22	4	1	4	3
591	3	13	11	23	5	1	4	4
592	3	13	11	24	6	1	4	3
593	3	13	11	25	7	1	4	5
594	3	13	11	26	8	1	4	3

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
595	3	13	11	28	9	1	4	4
596	3	13	11	29	10	1	4	7
597	1	14	11	21	1	1	1	0
598	1	14	11	27	2	1	1	1
599	1	14	12	4	3	1	1	10
600	1	14	11	21	1	1	2	3
601	1	14	11	27	2	1	2	13
602	1	14	12	4	3	1	2	10
603	1	14	11	21	1	1	3	1
604	1	14	11	27	2	1	3	5
605	1	14	12	4	3	1	3	6
606	1	14	11	21	1	1	4	1
607	1	14	11	27	2	1	4	3
608	1	14	12	4	3	1	4	12
609	2	14	11	21	1	1	1	6
610	2	14	11	27	2	1	1	10
611	2	14	12	4	3	1	1	10
612	2	14	11	21	1	1	2	6
613	2	14	11	27	2	1	2	6
614	2	14	12	4	3	1	2	7
615	2	14	11	21	1	1	3	3
616	2	14	11	27	2	1	3	3
617	2	14	12	4	3	1	3	3
618	2	14	11	21	1	1	4	3
619	2	14	11	27	2	1	4	6
620	2	14	12	4	3	1	4	5
621	3	14	11	21	1	1	1	1
622	3	14	11	27	2	1	1	1
623	3	14	12	4	3	1	1	3
624	3	14	11	21	1	1	2	2
625	3	14	11	27	2	1	2	1
626	3	14	12	4	3	1	2	2
627	3	14	11	21	1	1	3	6
628	3	14	11	27	2	1	3	6
629	3	14	12	4	3	1	3	6
630	3	14	11	21	1	1	4	3

continued . . .

rec_id	site	y	m	d	dayc	imgc	grid	mfc_t
631	3	14	11	27	2	1	4	5
632	3	14	12	4	3	1	4	10

Table F.1: Raw manual nest counts

Key: General ecological data records Table F.2 below.

rec_id: Unique record identifier

site: Monitoring locations
1-Mt. Tiger
2-Mt. Parihaka
3-Memorial Drive

y/m/d: Year, month, day

dayc: Collection season

temp: Temperature in degrees Celsius

hum: Percentage relative humility

wind-s: Speed in meters per second

cloud.c: Cloud cover as a percentage (subjective)

flight.act: Flight activity high, medium and low (subjective)

nest.act: Nest activity high, medium and low (subjective)

rec_id	site	y:m:d	dayc	temp	hum	wind-s	cloud-c	flight-act	nest-act
1	1	10:11:23	1						
2	1	10:11:24	2	22	53	1.2		L	L
3	1	10:11:26	3					L	L
4	1	10:11:28	4	21	59	2.5		M	M
5	1	10:11:29	5	23	70	1.2	70	M	M
6	1	10:11:30	6	23	56	1.9	40	M	M
7	1	10:12:1	7	20	76	1.4	60	M	H
8	1	10:11:2	8	20	64	1.1	70	L	L
9	1	10:11:5	9	20	75	1.2	60	M	L
10	1	10:11:6	10	20	62	1.4	5	H	H
11	1	10:11:8	11	20	63	1.1	60	M	L
12	1	10:11:9	12	19	58	2.4	10	H	M
13	1	10:11:10	13	26	48	1.2	5	H	M
14	1	10:12:11	14	22	49	1.4	10		
15	1	10:12:12	15	23	69	3.7		H	H
16	1	10:12:13	16	21	69	4.2		M	L
17	2	10:11:23	1	22	65	3			
18	2	10:11:24	2	24	47	1.1	10	M	M
19	2	10:11:26	3	20	66	1.3	60	M	M
20	2	10:11:28	4						

continued . . .

rec_id	site	y:m:d	dayc	temp	hum	wind-s	cloud-c	flight-act	nest-act
21	2	10:11:29	5	24	66	1.4	40	M	M
22	2	10:11:30	6	24	58	1.8	40	M	H
23	2	10:11:1	7	23	65	0.1	60	M	M
24	2	10:11:2	8	16	72	3.6	20	M	L
25	2	10:11:5	9	26	59	0.1	30	M	M
26	2	10:11:6	10	27	57	1.1	5	H	H
27	2	10:11:8	11	22	62	1.1			
28	2	10:11:9	12	26	47	0.1	0	H	H
29	2	10:11:10	13	24	47	1.3	0	H	H
30	2	10:12:11	14	24	49	3.6	5	H	H
31	2	10:12:12	15	22	65	1.6	10	H	H
32	2	10:12:13	16	23	58	1.3	5	M	M
33	1	11:11:11	1	16	68	0.7		L	L
34	1	11:11:12	2	15	57.9	0.8		L	L
35	1	11:11:13	3	18	62.2	0.1	60	L	L
36	1	11:11:14	4	16	65.8	0.1			
37	1	11:11:15	5	18	66.6	0.8			
38	1	11:11:16	6	18	57.9	0			
39	1	11:11:17	7						
40	1	11:11:18	8	17	68	0.6			
41	1	11:11:21	9	19	64.1	0.7	10		
42	1	11:11:22	10	21	60.1	0	10	M	M
43	1	11:11:23	11	21	59.9	2.05	10	M	M
44	1	11:11:24	12	15	79.4	0.9	60	L	L
45	1	11:11:28	13	25	57.8	0	60	M	L
46	1	11:12:7	14	20	74.2	0.6	20	M	M
47	1	11:12:9	15						
48	1	11:12:20	16						
49	2	11:11:11	1	28	40.7	0.1		L	L
50	2	11:11:12	2	31	48.6	0		L	L
51	2	11:11:13	3	27	42.8	0		L	L
52	2	11:11:14	4	27	50.7	0.7			
53	2	11:11:15	5	23	50	0.6			
54	2	11:11:16	6	23	78	2.2			
55	2	11:11:17	7	31	35.4	0.1			
56	2	11:11:18	8	24	49.4	0		M	M

continued . . .

rec_id	site	y:m:d	dayc	temp	hum	wind-s	cloud-c	flight-act	nest-act
57	2	11:11:21	9	22	61.2	0.7	10	M	H
58	2	11:11:22	10	28	49.3	2.4	10	M	M
59	2	11:11:23	11						
60	2	11:11:24	12	22	67.8	2.9		M	M
61	2	11:11:28	13	22	60.8	1.1		M	L
62	2	11:12:7	14	22	83.8	1.8	10	H	H
63	2	11:12:9	15	21	68.1	1.5			
64	2	11:12:20	16	20	60.9	1.5		H	H
65	3	11:11:11	1					L	L
66	3	11:11:12	2	26	36.5	0.7		L	L
67	3	11:11:13	3	23	53.8	0.7		L	L
68	3	11:11:14	4	26	43.5	1.1			
69	3	11:11:15	5	22	45	0			
70	3	11:11:16	6	23	56.4	2.3			
71	3	11:11:17	7	26	67.3	0.6			
72	3	11:11:18	8	22	50	0.6			
73	3	11:11:21	9	26	51.6	1			
74	3	11:11:22	10	31	44.5	0.7			
75	3	11:11:23	11						
76	3	11:11:24	12	24	58.2	0.6		M	M
77	3	11:11:28	13	24	60.8	0.6	5	H	H
78	3	11:12:7	14	25	56.1	0.7	5	H	H
79	3	11:12:9	15	30	55.6	0.6			
80	3	11:12:20	16						
81	1	12:11:9	1						
82	1	12:11:10	2						
83	1	12:11:11	3	14	69.4	0.3	20	L	L
84	1	12:11:12	4	16	77.4	0.4	50	L	L
85	1	12:11:13	5	17	72.2	0.4	10	M	M
86	1	12:11:14	6	14	69.6	0.3	60	L	L
87	1	12:11:15	7	15	82.1	0.5	80	L	L
88	1	12:11:16	8	19	62.9	0.4	50	L	L
89	1	12:11:18	9	19	53.2	0.6	30	L	L
90	1	12:11:19	10	17	53.9	0.4	60	L	L
91	1	12:12:4	11	20	59	0.6	10	H	H
92	1	12:12:11	12	26	52	0.3	5	H	H

continued . . .

rec_id	site	y:m:d	dayc	temp	hum	wind-s	cloud-c	flight-act	nest-act
93	1	12:12:12	13	20	58	0.2	5	H	H
94	2	12:11:9	1	22	50.3	0.5			
95	2	12:11:10	2	29	42.3	0.3	10	L	L
96	2	12:11:11	3	17	87.3	0	40	L	L
97	2	12:11:12	4	20	63	0.3	70	M	M
98	2	12:11:13	5	18	60.8	0.3	10	L	L
99	2	12:11:14	6	15	62.5	0.4	100	L	L
100	2	12:11:15	7	17	71.1	0.4	80	L	L
101	2	12:11:16	8	22	53.1	0.3		L	L
102	2	12:11:18	9	20	50.4	0.5	50	L	L
103	2	12:11:19	10	20	55.6	0.3	100	L	L
104	2	12:11:4	11	21	67.5	0.4	20	M	L
105	2	12:11:11	12	30	43	0.4	5	M	H
106	2	12:11:12	13	21	61.8	0.3	10	L	M
107	3	12:11:9	1	24	49	1.1			
108	3	12:11:10	2						
109	3	12:11:11	3	19	53.1	0	60	L	L
110	3	12:11:12	4	22	55.1	0.1	60	L	L
111	3	12:11:13	5	23	65.3	0.3	80	L	L
112	3	12:11:14	6	16	64.2	0.4	30	L	L
113	3	12:11:15	7	18	70.3	0.5	20	L	L
114	3	12:11:16	8	22	53.1	0.3		L	L
115	3	12:11:18	9	21	54.9	0.6		L	L
116	3	12:11:19	10	20	56.3	0.1		L	L
117	3	12:11:4	11	23	61.2	0.3	60	M	M
118	3	12:11:11	12	29	51	0.3	20	M	L
119	3	12:11:12	13	25	56.6	0.3	30	M	M
120	1	13:11:18	1	24	60.7	0.4	20	M	M
121	1	13:11:19	2	28	45.6	0.5	70	M	M
122	1	13:11:21	3	20	67	0.5	100	L	L
123	1	13:11:22	4	20	61	0.5	80		
124	1	13:11:23	5	20	69	0.4	50		
125	1	13:11:24	6	21	68	0.4		H	H
126	1	13:11:25	7	23	53	0.1	10	H	H
127	1	13:11:26	8	27	55	0.3	40	H	H
128	1	13:11:28	9	26	51	0	20	M	M

continued . . .

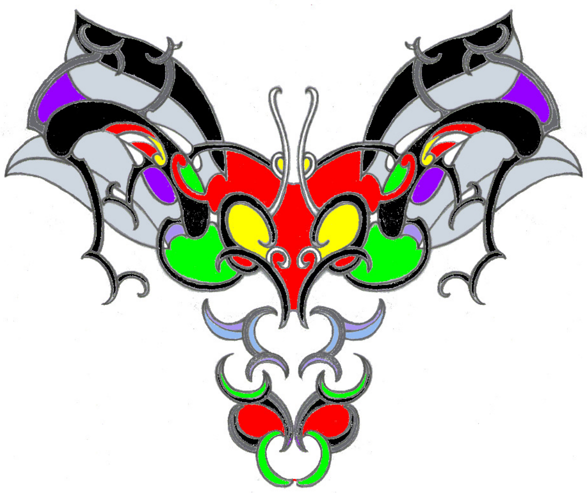
rec_id	site	y:m:d	dayc	temp	hum	wind-s	cloud-c	flight-act	nest-act
129	1	13:11:29	10	26	52	0.3	40	H	H
130	2	13:11:18	1	27	64	0.1	20	L	L
131	2	13:11:19	2	27	50.6	0.8	80	M	M
132	2	13:11:21	3	22	70	0.4	80	M	M
133	2	13:11:22	4	22	61	0.4	90	M	M
134	2	13:11:23	5	22	62	1	30		
135	2	13:11:24	6	26	61	0.1		H	H
136	2	13:11:25	7	22	48	1.1	20	H	H
137	2	13:11:26	8	25	70	0.2	100	M	M
138	2	13:11:28	9	25	55	0.5	20	M	M
139	2	13:11:29	10	29	47	0.2	30	M	M
140	3	13:11:18	1	28	47	0			
141	3	13:11:19	2	29	51	0	90	L	L
142	3	13:11:21	3	23	62	0.7	80		
143	3	13:11:22	4	24	58	0.4	40		
144	3	13:11:23	5	25	56	1	20		
145	3	13:11:24	6	22	57	0.6			
146	3	13:11:25	7	24	45	0.3	5		
147	3	13:11:26	8	27	65	0	50	M	M
148	3	13:11:28	9	32	44	0.3	40	M	M
149	3	13:11:29	10	31	37	0.8	80	M	M

Table F.2: Raw monitoring data

G

EXHIBITION

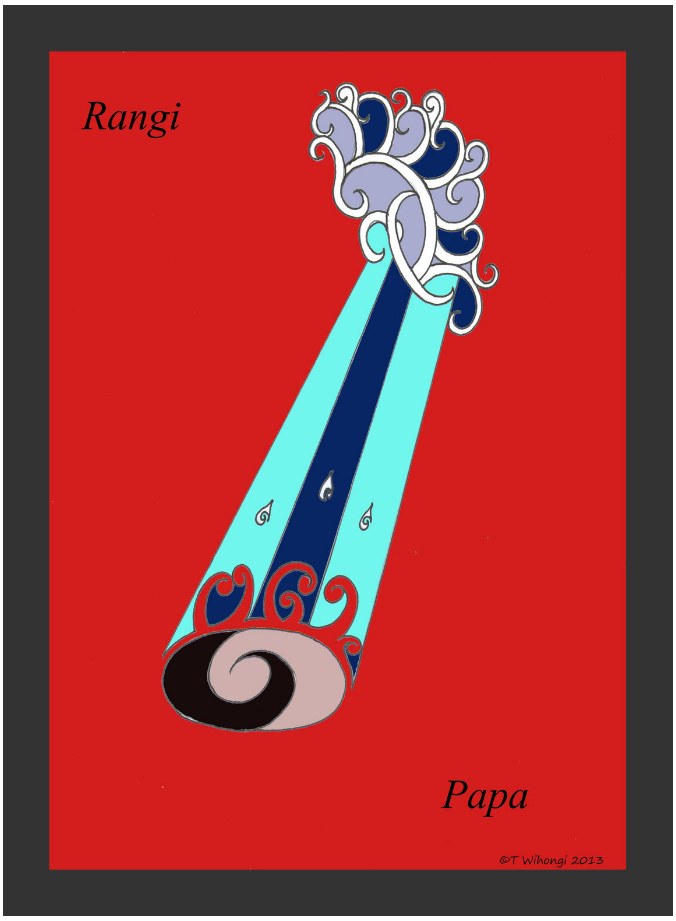
Tradition tells of the time when earth mother, *Papatuanuku*, was clothed in vegetation [1]. After she was adorned, the *Atua* (Gods) turned their attention towards the insects and reptiles of the earth. There were some who viewed the insects as *kutukutu* (vermin); infesting the body of Papa. The *whatukura* (enlightened beings), Ruatau and Rehua intervened. They spoke gently of the creatures to Tāne, the great God of the forests, saying...



Treat kindly the offspring of Torohua and Muhumuhu.
That they may serve as companions for you all.
Some are desirable, others are not.
But they preceded all other things.

Rangi and Papa

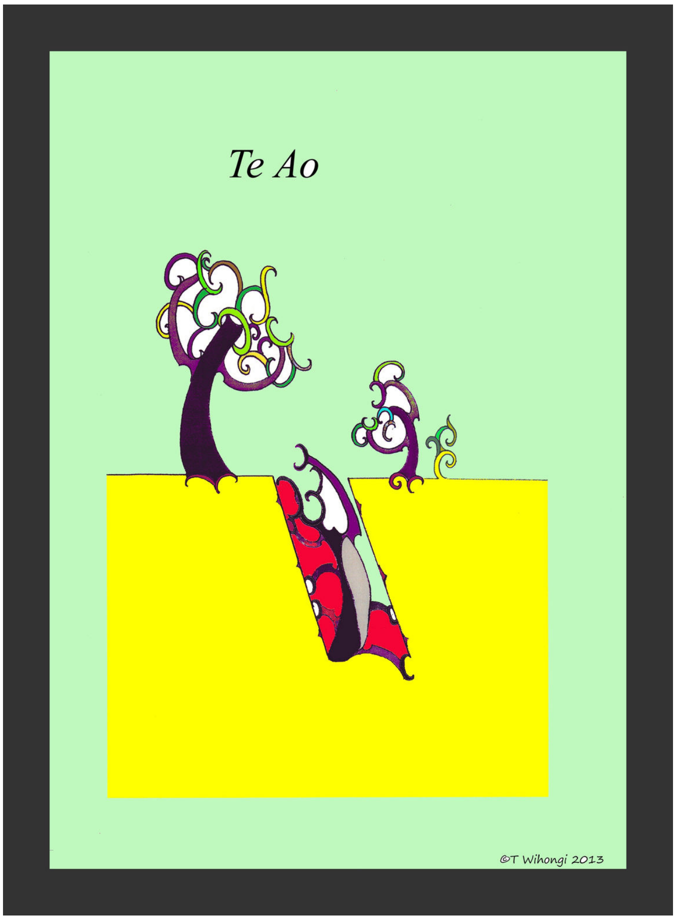
Native bees live underground as grubs throughout the winter months. The soil temperature increases as the rays from above reach the ground. So begins the cue, for a new cycle of life.



Te Waa.

Bees will start to emerge, at exactly the right time. Male bees prepare first. They begin to dig upwards and outwards and are the first to emerge from the ground. Days later, females emerge.

Te Ao.
Once all the bees have emerged from the ground, they enter the outside world. They begin to mate and the seeds of a new generation are created.



Mauri Ora.

When native bees emerge there is a rapid explosion of activity. Male bees form large mating swarms. Female bees start building new nests. Many native plants are in full flower.

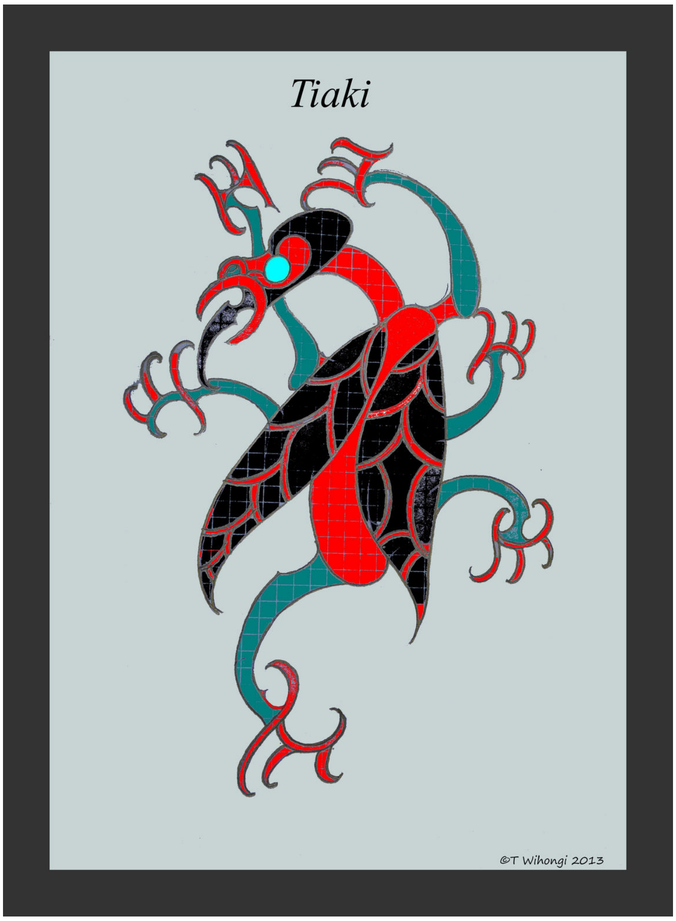
W HANAU.
Native bees are gregarious nester's. They form communities which are described as aggregations and are made up of individual whanau, hapū and iwi (different species and sub-species).

Korero.
Native bees are often attracted to a single, special tree, close by their community. William Doherty [212] explained... "

> *Native trees lived for hundreds of years. In their lifetime they held all their experiences, felt through their branches and leaves, at the base.*

TTiaki.

Native bees preserve the natural structure of native forests and trees, which depend on pollination for their own reproduction. Through this, can been seen the reciprocal act of kaitiakitanga.



Mana. Native bees have a long evolutionary history with native plants. These complex relationships are founded on kaitiakitanga and mana. The mauri of native trees, forests and ecosystems, depends on pollination from native bees, therefore they have enduring mana whenua.

Hine and Tamariki. Female native bees construct their own nests. They provision their brood with enough food to grow and sustain them through the winter months.

Hine
Tamariki
©T Wihongi 2013

KOTAHI. Native bees are important pollinators however, it is through kotahitanga, the unity of all groups combined, that maintains the mauri of the natural world. As described in this whakataukī (saying) by King Tāwhiao:

> *Ki te kotahi te kākaho ka whati, ki te kāpuia, e kore e whati.*
> *If there is but one toetoe stem it will break.*
> *But, if they are together, in a bundle,*
> *they will never break.*

H

REFERENCES

[1] Elsdon Best. *Māori religion and mythology: being an account of the cosmogony, anthropogeny, religious beliefs and rites, magic and folk lore of the Māori folk of New Zealand*. Te Papa Press, 2005. (Cited on pages iii and 209.)

[2] National Research Council (US). Committee on the Status of Pollinators in North America and National Academies Press (US). *Status of pollinators in North America*. Natl Academy Pr, 2007. (Cited on page 1.)

[3] Dias B.S.F., Raw A., and Imperatriz-Fonseca V.L. *International Pollinators Initiative: the São Paulo Declaration on Pollinators: Report on the Recommendations of the Workshop on the Conservation and Sustainable Use of Pollinators in Agriculture with Emphasis on Bees*. Brazilian Ministry of the Environment Brasília, 1999. (Cited on page 1.)

[4] Kearns C.A., Inouye D.W., and Waser N.M. Endangered mutualisms: the conservation of plant-pollinator interactions. *Annual review of ecology and systematics*, pages 83–112, 1998. (Cited on pages 1 and 3.)

[5] Williams G.R., Tarpy D.R., vanEngelsdorp D., Chauzat M., Cox-Foster D.L., Delaplane K.S., Neumann P., Pettis J.S., Rogers R.E., and Shutler D. Colony collapse disorder in context. *BioEssays*, 32(10):845–846, 2010. (Cited on page 1.)

[6] Tylianakis J.M. The global plight of pollinators. *Science*, 339(6127):1532–1533, 2013. (Cited on pages 1 and 2.)

[7] Allsopp M.H., de Lange W.J., and Veldtman R. Valuing insect pollination services with cost of replacement. *PLoS ONE*, 3(9), 2008. (Cited on page 1.)

[8] Bauer D.M. and Wing I.S. Economic Consequences of Pollinator Declines: A Synthesis. *Agricultural and Resource Economics Review*, 39(3):368–383, 2010. (Cited on page 1.)

[9] O'Toole C. Those other bees: changing the funding culture. *Pollinating Bees-The Conservation Link Between Agriculture and Nature*, pages 37–40, 2002. (Cited on page 1.)

[10] Goulson D. Conserving wild bees for crop pollination. *Food, Agriculture & Environment*, 1:142–144, 2003. (Cited on page 1.)

[11] Buchmann S.L., Nabhan G.P., and Mirocha P. *The Forgotten Pollinators*. A Shearwater book. Island Press, 1997. (Cited on page 1.)

[12] Newstrom-Lloyd L.E. *Pollination in New Zealand*, pages 408–431. Manaaki Whenua Press, 2013. (Cited on pages 1, 2, and 3.)

[13] Losey J.E. and Vaughan M. The economic value of ecological services provided by insects. *BioScience*, 56(4):311–323, 2006. (Cited on page 1.)

[14] Michener C.D. *Bees of the World*. Johns Hopkins University Press, 2000. (Cited on page 1.)

[15] Cane J. Annual displacement of soil in nest tumuli of alkali bees (*Nomia melanderi*) (Hymenoptera: Apiformies: Halictidae) across an agricultural landscape. *The Journal of the Kansas Entomological Society*, 76:172–176, 2003. (Cited on pages 1 and 2.)

[16] Cane J.H. A native ground-nesting bee (*Nomia melanderi*) sustainably managed to pollinate alfalfa across an intensively agricultural landscape. *The Journal of the Kansas Entomological Society*, 39:315–323, 2008. (Cited on pages 1, 2, 11, 15, 24, 112, 144, 152, 155, and 156.)

[17] Donovan B.J. *Apoidea (Insecta: Hymenoptera), Fauna of New Zealand*, volume 57. Manaaki Whenua Press, Landcare Research, 2007. (Cited on pages 1, 2, 3, 18, and 19.)

[18] Wheeler Q.D. Taxonomic triage and the poverty of phylogeny. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 359(1444):571–583, 2004. (Cited on page 2.)

[19] de Carvalho M.R., Bockmann F.A., Amorim D.S., Brandão C.R., de Vivo M., de Figueiredo J.L., Britski H.A., de Pinna M., Menezes N.A., Marques F.P., et al. Taxonomic impediment or impediment to taxonomy? a commentary on systematics and the cybertaxonomic-automation paradigm. *Evolutionary Biology*, 34(3-4):140–143, 2007. (Cited on page 2.)

[20] Osborne J.L., Clark S.J., Morris R.J., Williams I.H., Riley J.R., Smith A.D., Reynolds D.R., and Edwards A.S. A landscape-scale study of bumble bee foraging range and constancy, using harmonic radar. *Journal of Applied Ecology*, 36(4):519–533, 1999. (Cited on pages 2 and 16.)

[21] Hart N. *Methods for studying potential flight distances and foraging behaviours of native pollinating insects in New Zealand*. Pgdip, University of Auckland, School of Environmental Science, 2004. (Cited on pages 2, 3, 16, 49, 113, 131, and 141.)

[22] Steffan-Dewenter I. Importance of habitat area and landscape context for species richness of bees and wasps in fragmented orchard meadows. *Conservation Biology*, 17(4):1036–1044, 2003. (Cited on page 2.)

[23] Tarlton R. *A cluster of bees*. Endeavour press, 1935. (Cited on page 2.)

[24] Quinn R. Survey of native bees (Hymenoptera: Colletidae and Halictidae) in the Mackenzie Basin. *New Zealand Entomologist*, 8(1):41–44, 1984. (Cited on page 2.)

[25] Godley E.J. Flower biology in New Zealand. *New Zealand journal of botany*, 17(4):441–466, 1979. (Cited on page 2.)

[26] Kelly D., Ladley J.J., Robertson A.W., and Edwards J.and Smith D.C. The birds and the bees. *Nature*, page 384, 1996. (Cited on page 2.)

[27] Robertson A.W., Ladley J.J., and Kelly D. Effectiveness of short-tongued bees as pollinators of apparently ornithophilous new zealand mistletoes. *Austral Ecology*, 30(3):298–309, 2005. (Cited on pages 2 and 3.)

[28] Donovan B.J., Howlett B., and Walker M. Relocation and establishment of nesting populations of the native bee *Leioproctus huakiwi* Donovan (Hymenoptera: Colletidae). *New Zealand Entomologist*, 33(1):109–113, 2010. (Cited on page 3.)

[29] Hart N. *New Zealand Native Bees: A case Study in Whangarei*. Msc, University of Auckland, School of Enviromental Science, 2007. (Cited on pages 3, 16, 17, 18, 19, 20, 46, 48, 49, 113, 131, and 141.)

[30] Ward D.F., Early J.W., Schnitzler F.R., Hitchmough R.A., and Stringer I.A. The conservation status of New Zealand Hymenoptera. *New Zealand Entomologist*, 35(2):116–119, 2012. (Cited on page 3.)

[31] Kevan P.G. and Viana B.F. The global decline of pollination services. *Biodiversity*, 4(4):3–8, 2003. (Cited on page 3.)

[32] Field D., Tiwari B., Booth T., Houten S., Swan D., Bertrand N., and Thurston M. Open software for biologists: from famine to feast. *Nature biotechnology*, 24(7):801–803, 2006. (Cited on page 3.)

[33] Nikolaou N., Sampaziotis P., Aplikioti M., Drakos A., Kirmitzoglou I., Papamarkos N., and Promponas V.J. VeSTIS: A Versatile Semi - Automatic Taxon Identification System from Digital Images, 2010. (Cited on page 3.)

[34] Steiniger S.and Hay G.J. Free and open source geographic information tools for landscape ecology. *Ecological Informatics*, 4(4):183–195, 2009. (Cited on pages 3 and 31.)

[35] Irwin A. Constructing the scientific citizen: Science and democracy in the biosciences. *Public Understanding of Science*, 10(1):1–18, 2001. (Cited on page 3.)

[36] Newman H.B., Ellisman M.H., and Orcutt J.A. Data-intensive e-science frontier research. *Communications of the ACM*, 46(11):68–77, 2003. (Cited on page 3.)

[37] Goodchild M.F. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007. (Cited on pages 3 and 24.)

[38] Goodchild M.F. and Glennon J.A. Crowdsourcing geographic information for disaster response: a research frontier. *International Journal of Digital Earth*, 3(3):231–241, 2010. (Cited on pages 3 and 24.)

[39] Catlin-Groves C.L. The citizen science landscape: from volunteers to citizen sensors and beyond. *International Journal of Zoology*, 2012, 2012. (Cited on pages 3, 11, and 21.)

[40] Hart N.H. and Huang L. Monitoring populations of solitary bees using image processing techniques. *International Journal of Computer Applications in Technology*, 50(1):45–50, 2014. (Cited on pages 6, 131, and 141.)

[41] Hart N.H. and Huang L. Counting insects in flight using image processing techniques. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, pages 274–278. ACM, 2012. (Cited on pages 6, 131, and 141.)

[42] Hart N.H. and Huang L. Monitoring nests of solitary bees using image processing techniques. In *Mechatronics and Machine Vision in Practice (M2VIP), 2012 19th International Conference*, pages 1–4. IEEE, 2012. (Cited on page 7.)

[43] Hart N.H. and Huang L. An image based approach to monitor New Zealand native bees. In *Conference on Robotics, Automation and Mechatronics (RAM), 2011 IEEE Conference*, pages 353–357. IEEE, 2011. (Cited on pages 7, 131, and 141.)

[44] Fowler J., Cohen L., and Jarvis P. *Practical statistics for field biology*. John Wiley & Sons, 2013. (Cited on page 9.)

[45] Ridout M., Demétrio C., and Hinde J. Models for count data with many zeros. In *Proceedings of the XIXth international biometric conference*, volume 19, pages 179–192, 1998. (Cited on pages 9 and 144.)

[46] Eberhardt L.L. and Thomas J.M. Designing environmental field studies. *Ecological Monographs*, 61(1):53–73, 1991. (Cited on page 9.)

[47] Fletcher D., MacKenzie D., and Villouta E. Modelling skewed data with many zeros: a simple approach combining ordinary and logistic regression. *Environmental and ecological statistics*, 12(1):45–54, 2005. (Cited on pages 9 and 144.)

[48] Southwood T.R.E. and Henderson P.A. *Ecological methods*. John Wiley & Sons, 2009. (Cited on page 9.)

[49] Ludwig J.A. and Reynolds J.F. *Statistical ecology: a primer in methods and computing*, volume 1. John Wiley & Sons, 1988. (Cited on page 9.)

[50] Magurran A.E. *Measuring biological diversity*. John Wiley & Sons, 2013. (Cited on page 9.)

[51] Southwood T.R.E. Ecological methods, with particular reference to the study of insect populations. *London: The English Language Book Society and Chapman and Hall*, 1978. (Cited on page 9.)

[52] Kuno E. Sampling and analysis of insect populations. *Annual review of entomology*, 36(1):285–304, 1991. (Cited on page 9.)

[53] Krebs C.J. *Ecological methodology*, volume 620. Benjamin/Cummings Menlo Park, California, 1999. (Cited on page 9.)

[54] Morris R.F. Sampling insect populations. *Annual Review of Entomology*, 5(1):243–264, 1960. (Cited on pages 9, 112, and 155.)

[55] Seber G.A.F. The estimation of animal abundance. 1982. (Cited on page 9.)

[56] Hagler J.R. and Jackson C.G. Methods for marking insects: current techniques and future prospects. *Annual review of entomology*, 46(1):511–543, 2001. (Cited on pages 9 and 10.)

[57] Larsson M. and Franzé n M. Estimating the population size of specialised solitary bees. *Ecological Entomology*, 33(2):232–238, 2008. (Cited on page 10.)

[58] Westphal C., Bommarco R., Carré G., Lamborn E., Morison N., Petanidou T., Potts S.G., Roberts S.P., Szentgyörgyi H., Tscheulin T., et al. Measuring bee diversity in different european habitats and biogeographical regions. *Ecological Monographs*, 78(4):653–671, 2008. (Cited on page 10.)

[59] Nielsen A., Steffan-Dewenter I., Westphal C., Messinger O., Potts S.G., Roberts S P., Settele J., Szentgyörgyi H., Vaissiere B.E., Vaitis M., et al. Assessing bee species richness in two mediterranean communities: importance of habitat type and sampling techniques. *Ecological research*, 26(5):969–983, 2011. (Cited on page 10.)

[60] Cane J.H., Minckley R.L., and Kervin L.J. Sampling bees (Hymenoptera: Apiformes) for pollinator community studies: pitfalls of pan-trapping. *Journal of the Kansas Entomological Society*, pages 225–231, 2000. (Cited on page 10.)

[61] Roulston T.H., Smith S.A., and Brewster A.L. A comparison of pan trap and intensive net sampling techniques for documenting a bee (Hymenoptera: Apiformes) fauna. *Journal of the Kansas Entomological Society*, 80(2):179–181, 2007. (Cited on page 10.)

[62] Tuell J.K. and Isaacs R. Elevated pan traps to monitor bees in flowering crop canopies. *Entomologia experimentalis et applicata*, 131(1):93–98, 2009. (Cited on page 10.)

[63] Wilson J.S., Griswold T., and Messinger O.J. Sampling bee communities (Hymenoptera: Apiformes) in a desert landscape: are pan traps sufficient? *Journal of the Kansas Entomological Society*, 81(3):288–300, 2008. (Cited on page 10.)

[64] Bischoff I. Population dynamics of the solitary digger bee *Andrena vaga Panzer* (Hymenoptera, Andrenidae) studied using mark-recapture and nest counts. *Population ecology*, 45(3):197–204, 2003. (Cited on pages 10, 11, 15, 24, 112, 144, 152, and 155.)

[65] Fellendorf M., Mohra C., and Paxton R.J. Devasting effects of river flooding to the ground-nesting bee,*Andrena vaga* (Hymenoptera: Andrenidae), and its associated fauna. *Journal of Insect Conservation*, 8(4):311–312, 2004. (Cited on pages 11, 112, 155, and 156.)

[66] Vinchesi A.C. and Walsh D.B. Quadrat method for assessing the population abundance of a commercially managed native soil-nesting bee, *Nomia melanderi* (Hymenoptera: Halictidae), in proximity to alfalfa seed production in the Western United States. *Journal of economic entomology*, 107(4):1695–1699, 2014. (Cited on pages 11, 112, 144, 152, 155, and 156.)

[67] Mack C. The multiple lives of Moore's law. *Spectrum, IEEE*, 52(4):31–31, April 2015. (Cited on pages 11 and 24.)

[68] Schaller R.R. Moore's law: past, present and future. *Spectrum, IEEE*, 34(6):52–59, 1997. (Cited on page 11.)

[69] Kelling S., Hochachka W.M., Fink D., Riedewald M.and Caruana R., Ballard G., and Hooker G. Data-intensive science: A new paradigm for biodiversity studies. *BioScience*, 59(7):613–620, 2009. (Cited on pages 11, 15, 21, 24, and 25.)

[70] Irwin A. *Citizen science: a study of people, expertise, and sustainable development*. Psychology Press, 1995. (Cited on pages 11, 12, and 24.)

[71] Mortensen E.N., Delgado E. L., Deng H., Lytle D., Moldenke A., Paasch R., Shapiro L., Wu P., Zhang W., and Dietterich T.G. Pattern recognition for ecological science and environmental monitoring: An initial report. *Algorithmic Approaches to the Identification Problem in Systematics*, 2007. (Cited on page 12.)

[72] Gaston K.J. and O'Neill M.A. Automated species identification-why not? *Philosophical Transactions of the Royal Society of London*, B(359):655–667, 2004. (Cited on pages 12, 27, and 31.)

[73] Domingos P. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012. (Cited on pages 12, 129, 132, 136, and 139.)

[74] MacLeod N. *Automated taxon identification in systematics: theory, approaches and applications*. CRC Press, 2007. (Cited on page 12.)

[75] Roth V., Pogoda A., Steinhage V., and Schröder S. Pattern recognition combining feature-and pixel-based classification within a real world application. In *Mustererkennung 1999*, pages 120–129. Springer, 1999. (Cited on page 12.)

[76] Arbuckle T., Schroder S, Steinhage V., and Wittmann D. Biodiversity informatics in action: identification and monitoring of bee species using ABIS. In *Proc. 15th Int. Symp. Informatics for Environmental Protection*, volume 1, pages 425–430. Citeseer, 2001. (Cited on pages 12, 18, and 24.)

[77] Steinhage V., Arbuckle T., Schröder S., Cremers A.B., and Wittmann D. Abis: automated identification of bee species. In *BIOLOG workshop, German programme on biodiversity and global change, status report*, pages 194–195, 2001. (Cited on page 12.)

[78] Larios N., Soran B., Shapiro L.G., Martínez-Muñoz G., Lin J., and Dietterich T.G. Haar random forest features and svm spatial matching kernel for stonefly species identification. In *ICPR*, volume 1, page 7, 2010. (Cited on pages 12, 40, and 134.)

[79] Lytle D. A., Martínez-Muñoz G., Zhang W., Larios N., Shapiro L., Paasch R., Moldenke A., Mortensen E. N., and Todorovic S.and Dietterich T. G. Automated processing and identification

of benthic invertebrate samples. *Journal of the North American Benthological Society*, 29(3):867–874, 2010. (Cited on page 12.)

[80] Lone K.S., Egerton S., Cristofaro D., and Clementson S. Automated real time dynamic identification of flying and resting butterfly species in the natural environment. In *2011 International Conference on Environment Science and Engineering (ICESE 2011)*, pages 179–183. Institute of Electrical and Electronics Engineers, 2011. (Cited on pages 12, 13, and 18.)

[81] Towner A.V., Wcisel M.A., Reisinger R.R., Edwards D., and Jewell O.J. Gauging the threat: the first population estimate for white sharks in South Africa using photo identification and automated software. *PloS one*, 8(6):e66035, 2013. (Cited on pages 12, 13, and 15.)

[82] Van Tienhoven A. M., Den Hartog J. E., Reijns R. A., and Peddemors V. M. A computer-aided program for pattern-matching of natural marks on the spotted raggedtooth shark *carcharias taurus*. *Journal of Applied Ecology*, 44(2):273–280, 2007. (Cited on pages 12 and 13.)

[83] Kaku M. *Visions: how science will revolutionize the 21st century*. Oxford University Press, 1999. (Cited on pages 12 and 24.)

[84] Potamitis I., Ganchev T., and Fakotakis N. Automatic acoustic identification of insects inspired by the speaker recognition paradigm. In *INTERSPEECH*, 2006. (Cited on pages 12, 21, 113, and 155.)

[85] Raman D.R., Gerhardt R.R., and Wilkerson J.B. Detecting insect flight sounds in the field: Implications for acoustical counting of mosquitoes. *Transactions of the ASABE*, 50(4):1481, 2007. (Cited on pages 12, 21, and 113.)

[86] Offenhauser J.and Kahn W. and Morton C. The sounds of disease-carrying mosquitoes. *The Journal of the Acoustical Society of America*, 21(3):259–263, 1949. (Cited on page 12.)

[87] Batista G.E., Keogh E.J., Mafra-Neto A., and Rowton E. SIGKDD demo: sensors and software to allow computational entomology, an emerging application of data mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 761–764. ACM, 2011. (Cited on pages 13, 21, and 113.)

[88] Caci G., Biscaccianti A. B., Cistrone L., Bosso L., Garonna A. P., and Russo D. Spotting the right spot: computer-aided individual identification of the threatened cerambycid beetle *rosalia alpina*. *Journal of insect conservation*, 17(4):787–795, 2013. (Cited on pages 13, 14, 15, and 18.)

[89] Pierrer X. and Jacquemoud-Collet J.P. DARWIN software, version 5.0. *Available from Internet: http://darwin. cirad. fr/Download. php.[cited January 5, 2010]*, 2006. (Cited on page 13.)

[90] Compagno L.J. Government protection for the great white shark *(Carcharodon carcharias)* in South Africa. *South Afrian Journal of Science*, 87(7):284–285, 1991. (Cited on page 13.)

[91] Mayo M. and Watson A.T. Automatic species identification of live moths. *Knowledge-Based Systems*, 20(2):195–202, 2007. (Cited on pages 13 and 18.)

[92] Schneider C.A, Rasband W.S., and Eliceiri K. W. NIH Image to ImageJ: 25 years of image analysis. *Nature methods*, 9(7):671–675, 2012. (Cited on pages 13, 21, 25, 30, and 33.)

[93] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I.H. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009. (Cited on pages 13, 39, 129, 131, 134, 135, 136, and 150.)

[94] Murray T.E., Kuhlmann M., and Potts S.G. Conservation ecology of bees: populations, species and communities. *Apidologie*, 40(3):211–236, 2009. (Cited on page 15.)

[95] Vitolo C., Elkhatib Y., Reusser D., Macleod C.J., and Buytaert W. Web technologies for environmental big data. *Environmental Modelling & Software*, 63:185–198, 2015. (Cited on pages 15 and 21.)

[96] O'Neal M.E., Landis D.A., Rothwell E., Kempel L., and Reinhard D. Tracking insects with harmonic radar: a case study. *American Entomologist*, 50(4):212–218, 2004. (Cited on page 16.)

[97] Mascanzoni D. and Wallin H. The harmonic radar: a new method of tracing insects in the field. *Ecological Entomology*, 11(4):387–390, 1986. (Cited on page 16.)

[98] Lövei G.L, Stringer I., Devine C.D, and Cartellieri M. Harmonic radar - a method using inexpensive tags to study invertebrate movement on land. *New Zealand Journal of Ecology*, 21(2):187–193, 1997. (Cited on page 16.)

[99] Colpitts B.G and Boiteau G. Harmonic radar transceiver design: miniature tags for insect tracking. *Antennas and Propagation, IEEE Transactions on*, 52(11):2825–2832, 2004. (Cited on page 16.)

[100] Brazee R.D., Miller E.S., Reding M.E., Klein M.G., Nudd B., and Zhu H. A transponder for harmonic radar tracking of the black vine weevil in behavioral research. *Transactions of the ASAE*, 48(2):831–838, 2005. (Cited on page 16.)

[101] Schindelin J., Arganda-Carreras I., Frise E., Kaynig V., Longair M., Pietzsch T., Preibisch S., Rueden C., Saalfeld S., Schmid B., et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012. (Cited on pages 21, 25, 26, 30, 128, 131, 133, and 150.)

[102] Solis-Sánchez L.O., García-Escalante J.J., Castañeda-Miranda R., Torres-Pacheco I., and Guevara-González R. Machine vision algorithm for whiteflies *(bemisia tabaci genn.)* scouting under greenhouse environment. *Journal of Applied Entomology*, 133(7):546–552, 2009. (Cited on pages 21, 22, and 23.)

[103] Solis-Sánchez L.O., Castañeda-Miranda R., García-Escalante J.J., Torres-Pacheco I., Guevara-González R.G., Castañeda-Miranda C.L., and Alaniz-Lumbreras P.D. Scale invariant feature approach for insect monitoring. *Computers and Electronics in Agriculture*, 75(1):92–99, 2011. (Cited on page 22.)

[104] Checchi F., Stewart B.T., Palmer J.J., and Grundy C. Validity and feasibility of a satellite imagery-based method for rapid estimation of displaced populations. *Int J Health Geogr*, 12(4), 2013. (Cited on pages 22 and 23.)

[105] Johansson P. Small vessel detection in high quality optical satellite imagery. Technical report, Chalmers University of Technology Sweden, 2011. (Cited on pages 22, 23, 24, and 44.)

[106] Gray J. Jim Gray on eScience. A Transformed Scientific Method, pp. xvii-xxxi. *The Fourth Paradigm. Data-Intensive Scientific Discovery. Edited by T. Hey, S. Tansley and K. Tolle. Redmond, WA: Microsoft Research*, 2009. (Cited on pages 24 and 25.)

[107] Lindenmayer D. and Likens G.E. Benchmarking open access science against good science. *Bulletin of the Ecological Society of America*, 94(4):338–340, 2013. (Cited on pages 24, 25, and 31.)

[108] Sanchez C. et al. Volunteer clouds and citizen cyberscience for LHC physics. In *Journal of Physics: Conference Series*, volume 331, page 062022. IOP Publishing, 2011. (Cited on pages 24 and 150.)

[109] Bell G., Hey T., and Szalay A. Beyond the data deluge. *Science*, 323(5919):1297–1298, 2009. (Cited on pages 24 and 26.)

[110] Hampton S.E., Strasser C.A., Tewksbury J.J., Gram W.K., Budden A.E., Batcheller A.L., Duke C.S., and Porter J.H. Big data and the future of ecology. *Frontiers in Ecology and the Environment*, 11(3):156–162, 2013. (Cited on pages 25 and 31.)

[111] Poisot T.E., Mounce R., and Gravel D. Moving toward a sustainable ecological science: don't let data go to waste! *Ideas in Ecology and Evolution*, 6(2), 2013. (Cited on page 25.)

[112] Glasbey C.A. and Horgan G.W. *Image analysis for the biological sciences*, volume 1. Wiley Chichester, 1995. (Cited on pages 25, 26, 28, 117, and 120.)

[113] Pearson H. The good, the bad and the ugly. *Nature*, 447(7141):138–140, 2007. (Cited on pages 25, 28, and 31.)

[114] Delaney J. An inconvenient truth? scientific photography and archival ambivalence. *Archivaria*, 65:75–95, 2008. (Cited on page 26.)

[115] Amen D.G., Trujillo M., Newberg A., Willeumier K., Tarzwell R., Wu J.C., and Chaitin B. Brain SPECT imaging in complex psychiatric cases: an evidence-based, underutilized tool. *The open neuroimaging journal*, 5:40, 2011. (Cited on page 26.)

[116] Mignard F., Coryn Bailer-Jones, Bastian U, Drimmel R., Eyer L., Katz D., Van Leeuwen F., Luri X., O'Mullane W., Passot X., et al. Gaia: organisation and challenges for the data processing. *Proceedings of the International Astronomical Union*, 3(S248):224–230, 2007. (Cited on page 26.)

[117] Ljosa V. and Carpenter A.E. Introduction to the quantitative analysis of two-dimensional fluorescence microscopy images for cell-based screening. *PLoS computational biology*, 5(12):e1000603, 2009. (Cited on pages 26, 27, 35, and 38.)

[118] Shamir L., Delaney J.D., Orlov N., Eckley D.M., and Goldberg I.G. Pattern recognition software and techniques for biological image analysis. *PLoS computational biology*, 6(11):e1000974, 2010. (Cited on page 27.)

[119] Antony P.A, Trefois C., Stojanovic A., Baumuratov A. Sagatovich, and Kozak K. Light microscopy applications in systems biology: opportunities and challenges. *Cell Communication and Signaling*, 11(1):1–19, 2013. (Cited on pages 27 and 30.)

[120] Pennekamp F. and Schtickzelle N. Implementing image analysis in laboratory-based experimental systems for ecology and evolution: a hands-on guide. *Methods in Ecology and Evolution*, 4(5):483–492, 2013. (Cited on page 27.)

[121] Silverstein D.A. and Farrell J.E. The relationship between image fidelity and image quality. In *International Conference on Image Processing, 1996.*, volume 1, pages 881–884. IEEE, 1996. (Cited on page 28.)

[122] Cromey D.W. Avoiding twisted pixels: ethical guidelines for the appropriate use and manipulation of scientific digital images. *Science and engineering ethics*, 16(4):639–667, 2010. (Cited on page 28.)

[123] Bouckaert R.R, Frank E., Hall M.A, Holmes G., Pfahringer B., Reutemann P., and Witten I.H. WEKA—Experiences with a Java open-source project. *The Journal of Machine Learning Research*, 11:2533–2541, 2010. (Cited on pages 30, 40, and 134.)

[124] Signed by 25 Nobel Prize Winners. An Open Letter to the U.S. Congress. *Document available at http://fas.org/sgp/news/2004/08/nobel082604.pdf*, 2004. (Cited on page 30.)

[125] Sonnenburg S., Braun M.L, Ong C.S, Bengio S., Bottou L., Holmes G., LeCunn Y., Muller K.R., Pereira F., Rasmussen C.E., et al. The need for open source software in learning. *Journal of Machine Learning Research*, 8:2443–2466, 2007. (Cited on page 30.)

[126] De Chaumont F., Dallongeville S., Chenouard N., Hervé N., Pop S., Provoost T., Meas-Yedid V., Pankajakshan P., Lecomte T., Le Montagner Y., et al. Icy: an open bioimage informatics platform for extended reproducible research. *Nature methods*, 9(7):690–696, 2012. (Cited on page 30.)

[127] Qidwai U. and Chen C. *Digital image processing: an algorithmic approach with MATLAB*. CRC press, 2011. (Cited on page 30.)

[128] Pau G., Fuchs F., Sklyar O., Boutros M., and Huber W. EBImage–an R package for image processing with applications to cellular phenotypes. *Bioinformatics*, 26(7):979–981, 2010. (Cited on page 30.)

[129] Soranno P.A., Cheruvelil K.S., Elliott K.C., and Montgomery G.M. It's good to share: Why environmental scientists' ethics are out of date. *BioScience*, 65(1):69–73, 2015. (Cited on page 31.)

[130] Evans J..S, Murphy M.A., Holden Z.A., and Cushman S.A. Modeling species distribution and change using random forest. In *Predictive Species and Habitat Modeling in Landscape Ecology*, pages 139–159. Springer, 2011. (Cited on page 31.)

[131] Mascaro J., Asner G.P., Knapp D.E., Kennedy-Bowdoin T., Martin R.E., Anderson C., Higgins M., and Chadwick K.D. A tale of two "forests": Random forest machine learning aids tropical forest carbon mapping. *PloS one*, 9(1):e85993, 2014. (Cited on pages 31 and 41.)

[132] Reichman O.J., Jones M.B., and Schildhauer M.P. Challenges and opportunities of open data in ecology. *Science*, 331(6018), 2011. (Cited on page 31.)

[133] European Union. Advancing technologies and federating communities. 2012. (Cited on page 31.)

[134] Easlon H.M. and Bloom A.J. Easy leaf area: Automated digital image analysis for rapid and accurate measurement of leaf area. *Applications in plant sciences*, 2(7), 2014. (Cited on page 31.)

[135] Altermatt F., Fronhofer E.A, Garnier A., Giometto A., Hammes F., Klecka J., Legrand D., Mächler E., Massie T.M, Pennekamp F., et al. Big answers from small worlds: a user's guide for protist microcosms as a model system in ecology and evolution. *Methods in Ecology and Evolution*, 2014. (Cited on page 31.)

[136] Mike Rossner and Kenneth M Yamada. What's in a picture? the temptation of image manipulation. *The Journal of Cell Biology*, 166(1):11–15, 2004. (Cited on page 31.)

[137] North A.J. Seeing is believing? a beginners' guide to practical pitfalls in image acquisition. *The Journal of cell biology*, 172(1):9–18, 2006. (Cited on page 31.)

[138] Burks T.F., Shearer S.A., Payne F.A., et al. Classification of weed species using color texture features and discriminant analysis. *Transactions of the ASAE*, 43(2):441–448, 2000. (Cited on pages 32 and 118.)

[139] Jay S., Lawrence R., Repasky K., and Keith C. Invasive species mapping using low cost hyper spectral imagery. In *ASPRS 2009 Annual Conference, March*, pages 9–13, 2009. (Cited on page 32.)

[140] Peña B., José M., Kelly M., Castro A.I de, Francisca López G., et al. Object-based approach for crop row characterization in uav images for site-specific weed management. 2012. (Cited on page 32.)

[141] Peña J.M, Torres-Sánchez J., Serrano-Pérez A., de Castro A.I, and López-Granados F. Quantifying efficacy and limits of unmanned aerial vehicle (uav) technology for weed seedling detection as affected by sensor resolution. *Sensors*, 15(3):5609–5626, 2015. (Cited on page 32.)

[142] Pajares G. Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs). *Photogrammetric Engineering & Remote Sensing*, 81(4):281–330, 2015. (Cited on pages 32 and 117.)

[143] Feng Q., Liu J., and Gong J. Uav remote sensing for urban vegetation mapping using random forest and texture analysis. *Remote Sensing*, 7(1):1074–1094, 2015. (Cited on pages 32, 41, 44, 117, 118, and 119.)

[144] Feng Q., Liu J., and Gong J. Urban flood mapping based on unmanned aerial vehicle remote sensing and random forest classi-

fier - a case of yuyao, china. *Water*, 7(4):1437–1455, 2015. (Cited on pages 32, 41, 44, 117, 118, and 119.)

[145] Waske B., Heinzel V., Braun M., and Menz G. Random forests for classifying multi–temporal sar data. In *Proc. Envisat Symposium*, pages 23–27, 2007. (Cited on page 32.)

[146] Mellor A., Haywood A., Stone C., and Jones S. The performance of random forests in an operational setting for large area sclerophyll forest classification. *Remote Sensing*, 5(6):2838–2856, 2013. (Cited on pages 32 and 41.)

[147] Lam K.W, Li Z., and Yuan X. Effects of jpeg compression on the accuracy of digital terrain models automatically derived from digital aerial images. *The Photogrammetric Record*, 17(98):331–342, 2001. (Cited on pages 32, 33, and 118.)

[148] Zabala A, Pons X, Díaz-Delgado R, García F, Aulí-Llinàs F, and Serra-Sagristà J. Effects of jpeg and jpeg2000 lossy compression on remote sensing image classification for mapping crops and forest areas. In *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on*, pages 790–793. IEEE, 2006. (Cited on pages 33 and 117.)

[149] Zabala A and Pons X. Effects of lossy compression on remote sensing image classification of forest areas. *International Journal of Applied Earth Observation and Geoinformation*, 13(1):43–51, 2011. (Cited on pages 33, 117, and 118.)

[150] Pierre emmanuel G. XnView: Software for viewing, converting, organising and editing raster images. *Software available at http://www. xnview. com*, 2014. (Cited on pages 33 and 150.)

[151] Paola J.D and Schowengerdt R.A. The effect of lossy image compression on image classification. In *Geoscience and Remote Sensing Symposium, 1995. IGARSS 95.Quantitative Remote Sensing for Science and Applications, International*, volume 1, pages 118–120. IEEE, 1995. (Cited on pages 33, 117, and 118.)

[152] Blaschke T., Burnett C., and Pekkarinen A. Image segmentation methods for object-based analysis and classification. In *Remote sensing image analysis: Including the spatial domain*, pages 211–236. Springer, 2004. (Cited on page 34.)

[153] Wang S., Chung F., and Xiong F. A novel image thresholding method based on parzen window estimate. *Pattern Recognition*, 41(1):117–129, 2008. (Cited on page 35.)

[154] Deriche R. Using canny's criteria to derive a recursively implemented optimal edge detector. *International journal of computer vision*, 1(2):167–187, 1987. (Cited on pages 37 and 124.)

[155] Nock R. and Nielsen F. Statistical region merging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(11):1452–1458, 2004. (Cited on pages 37, 38, and 124.)

[156] Nock R. and Nielsen F. Semi-supervised statistical region refinement for color image segmentation. *Pattern Recognition*, 38(6):835–846, 2005. (Cited on pages 37 and 124.)

[157] Kaynig V., Arganda-Carreras I., and Cardona A. Trainable weka segmentation plugin. *Kaynig, V and Arganda-Carreras, I, Editors*, 2010. (Cited on page 38.)

[158] Platt J. and et al. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods–support vector learning*, 3, 1999. (Cited on pages 40 and 134.)

[159] Kumar M. and Thenmozhi M. Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian Institute of Capital Markets 9th Capital Markets Conference Paper*, 2006. (Cited on page 40.)

[160] Kampichler C., Wieland R., S. Calmé, Weissenberger H., and Arriaga-Weiss S. Classification in conservation biology: a comparison of five machine-learning methods. *Ecological Informatics*, 5(6):441–450, 2010. (Cited on page 40.)

[161] Fernández-Delgado M., Cernadas E., Barro S., and Amorim D. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014. (Cited on pages 40, 41, 134, and 136.)

[162] Albert J., Aliu E., Anderhub H., Antoranz P., Armada A., Asensio M., Baixeras C., Barrio J.A., Bartko H., Bastieri D., et al. Implementation of the random forest method for the imaging atmospheric cherenkov telescope magic. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 588(3):424–432, 2008. (Cited on page 40.)

[163] Genuer R., Michel V., Eger E., and Thirion B. Random forests based feature selection for decoding fmri data. In *Proceedings Compstat*, number 267, pages 1–8, 2010. (Cited on page 40.)

[164] Díaz-Uriarte R. and De Andres S. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006. (Cited on page 40.)

[165] Svetnik V., Liaw A., Tong C., and Wang T. Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. In *Multiple Classifier Systems*, pages 334–343. Springer, 2004. (Cited on page 40.)

[166] Prasad A.M., Iverson L.R., and Liaw A. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, 9(2):181–199, 2006. (Cited on page 40.)

[167] Cutler D.R., Edwards J. Thomas C, Beard K.H., Cutler A., Hess K.T., Gibson J., and Lawler J.J. Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007. (Cited on page 40.)

[168] Schwartz M.W., Iverson L.R., Prasad A.M., Matthews S.N., and O'Connor R.J. Predicting extinctions as a result of climate change. *Ecology*, 87(7):1611–1615, 2006. (Cited on page 40.)

[169] Rodriguez-Galiano V.F., Ghimire B., Rogan J., Chica-Olmo M., and Rigol-Sanchez J.P. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67:93–104, 2012. (Cited on pages 41 and 142.)

[170] Cutler R., Brown L., Powell J., Bentz B., and Cutler A. Identifying redtops: Classification of satellite imagery for tracking mountain pine beetle progression through a pine forest. In *Proceedings of the 35th Symposium on the Interface, March*, pages 12–15, 2003. (Cited on page 41.)

[171] Lawrence R.L., Wood S.D., and Sheley R.L. Mapping invasive plants using hyperspectral imagery and breiman cutler classifications (randomforest). *Remote Sensing of Environment*, 100(3):356–362, 2006. (Cited on page 41.)

[172] Jain V., Seung H.S., and Turaga S.C. Machines that learn to segment images: a crucial technology for connectomics. *Current opinion in neurobiology*, 20(5):653–666, 2010. (Cited on pages 41 and 128.)

[173] Jain V., Bollmann B.and Richardson M., Briggman K.L. Berger D.R.and Helmstaedter M.N., Denk W., Bowden J.B., Mendenhall J.M., C. Abraham W., et al. Boundary learning by optimization with topological constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2488–2495. IEEE, 2010. (Cited on page 41.)

[174] Unnikrishnan R., Pantofaru C., and Hebert M. Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):929–944, 2007. (Cited on pages 41, 137, and 158.)

[175] Di E.B. and Glass M. The kappa statistic: A second look. *Computational linguistics*, 30(1):95–101, 2004. (Cited on page 42.)

[176] Lin L., Hedayat A.S., Sinha B., and Yang M. Statistical methods in assessing agreement: Models, issues, and tools. *Journal of the American Statistical Association*, 97(457):257–270, 2002. (Cited on pages 42, 137, 143, and 151.)

[177] Stewart E.L. and McDonald B.A. Measuring quantitative virulence in the wheat pathogen zymoseptoria tritici using high-throughput automated image analysis. *Phytopathology*, 104(9):985–992, 2014. (Cited on pages 42, 43, 143, 144, and 152.)

[178] Bock C.H., Parker P.E., Cook A.Z., and Gottwald T.R. Visual rating and the use of image analysis for assessing different symptoms of citrus canker on grapefruit leaves. *Plant Disease*, 92(4):530–541, 2008. (Cited on pages 43, 143, and 144.)

[179] Lawrence I. and Lin K. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, pages 255–268, 1989. (Cited on pages 43, 78, 142, 143, and 151.)

[180] Meijering E. Featurej: A java package for image feature extraction, 2007. (Cited on page 63.)

[181] Tschumperle D. and Deriche R. Vector-valued image regularization with PDEs: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):506–517, 2005. (Cited on page 66.)

[182] Chaudhury K.N., Sage D., and Unser M. Fast bilateral filtering using trigonometric range kernels. *IEEE Transactions on Image Processing*, 20(12):3376–3382, 2011. (Cited on page 66.)

[183] Štencel M. and Janáček J. On calculation of chamfer distance and lipschitz covers in digital images. *Proceedings S4G. Prague: Union of Czech Mathematicians and Physicists*, pages 517–522, 2006. (Cited on page 66.)

[184] Stevenson M., Nunes T., Sanchez J., Thornton R., Reiczigel J., Robison-Cox J., and Sebastiani P. epiR: An R package for the analysis of epidemiological data. *R package version 0.9–43*, 2012. (Cited on page 78.)

[185] Forman G. and Scholz M. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newsletter*, 12(1):49–57, 2010. (Cited on pages 86 and 135.)

[186] Thornton C., Hutter F., Hoos H.H., and Leyton-Brown K. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data*

mining, pages 847–855. ACM, 2013. (Cited on pages 86, 134, 135, and 136.)

[187] Gonzalez R.C. *Digital image processing*. Pearson Education India, 2009. (Cited on page 120.)

[188] Pham D.L., Xu C., and Prince J.L. Current methods in medical image segmentation. *Annual review of biomedical engineering*, 2(1):315–337, 2000. (Cited on pages 120, 128, and 139.)

[189] Abo-Eleneen Z.A. Thresholding based on fisher linear discriminant. *Journal of pattern recognition research*, 2:326–334, 2011. (Cited on page 120.)

[190] Gabriel L. and others. Auto threshold. *Software package available at http://fiji.sc/AutoThreshold*, 2015. (Cited on page 120.)

[191] Latinne P., Debeir O., and Decaestecker C. Limiting the number of trees in random forests. In *Multiple Classifier Systems*, pages 178–187. Springer, 2001. (Cited on pages 133 and 140.)

[192] Breiman L. Random forests. *Machine learning*, 45(1):5–32, 2001. (Cited on pages 133 and 134.)

[193] Bottou L. and Lin C. Support vector machine solvers. *Large scale kernel machines*, pages 301–320, 2007. (Cited on page 134.)

[194] Pal M. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005. (Cited on page 142.)

[195] Gislason P.O., Benediktsson J.A., and Sveinsson J.R. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300, 2006. (Cited on page 142.)

[196] Kwiecien R., Kopp-Schneider A., and Blettner M. Concordance analysis: part 16 of a series on evaluation of scientific publications. *Deutsches Ärzteblatt International*, 108(30):515, 2011. (Cited on page 143.)

[197] McBride G.B. A proposal for strength-of-agreement criteria for Lin's concordance correlation coefficient. *NIWA client report: HAM*, 62:2005, 2005. (Cited on page 143.)

[198] K Thompson and J Pi Grime. Seasonal variation in the seed banks of herbaceous species in ten contrasting habitats. *The Journal of Ecology*, pages 893–921, 1979. (Cited on page 146.)

[199] Michael Huston and Thomas Smith. Plant succession: life history and competition. *American Naturalist*, pages 168–198, 1987. (Cited on page 146.)

[200] Matthew N. and Stones R. *Beginning Linux Programming*. John Wiley & Sons, 2011. (Cited on page 150.)

[201] Dabbish L., Stuart C., Tsay J., and Herbsleb J. Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1277–1286. ACM, 2012. (Cited on pages 150 and 155.)

[202] Hart N. Native bees: Digital image collections used for monitoring nz bees. *Information and data repository – http://nzbees.github.io*, 2013. (Cited on pages 150 and 155.)

[203] Adolfo G.B. Pyrenamer. *Software available at https://launchpad.net/pyrenamer*, 2015. (Cited on page 150.)

[204] Harvey P. Exiftool: Read, write and edit meta information. *Software package available at http://www.sno.phy.queensu.ca/ phil/exiftool*, 2013. (Cited on page 150.)

[205] The Apache Software Foundation. Apache OpenOffice 4.1.1: The Free and Open Productivity Suite . *Software available at https://www.openoffice.org/*, 2015. (Cited on page 150.)

[206] Hothorn T. and Everitt B.S. *A handbook of statistical analyses using R*. CRC Press, 2014. (Cited on page 150.)

[207] Syntevo. SmartGit 6.5.7: A Git client with Mercurial and Subversion support. *Software package available at http://www.syntevo.com/smartgit/download*, 2013. (Cited on page 150.)

[208] Rebecca M Jarvis, Barbara Bollard Breen, Christian U Krägeloh, and D Rex Billington. Citizen science and the power of public participation in marine spatial planning. *Marine Policy*, 57:21–26, 2015. (Cited on page 158.)

[209] Maori Marsden and TA Henare. *Kaitiakitanga: A definitive introduction to the holistic world view of the Maori*. Ministry for the Environment, 1992. (Cited on page 160.)

[210] Mere Roberts, Waerete Norman, Nganeko Minhinnick, Del Wihongi, and Carmen Kirkwood. Kaitiakitanga: Maori perspectives on conservation. *Pacific Conservation Biology*, 2(1):7–20, 1995. (Cited on page 160.)

[211] Mere Roberts. Mind maps of the maori. *GeoJournal*, 77(6):741–751, 2012. (Cited on page 160.)

[212] William Doherty. *Mātauranga Tūhoe: the centrality of Mātauranga-a-iwi to Māori education.* PhD thesis, ResearchSpace@ Auckland, 2010. (Cited on page 215.)