

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

June 2015 – version 1.0

Ngaire Hart: *MONITORING NEW ZEALAND'S NATIVE BEES, A Collaborative Approach using Image Analysis*, © June 2015

ABSTRACT

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

However, studies of native bees can be difficult so 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 semi-automated image analysis is proposed and capitalises on unique aspects of their nesting biology. While it is difficult to acquire images of individual bees it is comparatively straightforward to photograph nests. Furthermore, the number of nests in an aggregation can provide a good indicator of community health. For this reason methods centred on counting the number of active nests which are used as a proxy for populations.

Image data were collected over four years from three communities of native bees located in Whangarei (Northland, New Zealand). Fundamental ecological data were collected including manual nest counts. Biomedical imaging platform Fiji was used to process all images. Pixel-level segmentation was implemented via the *Random Forest* algorithm. The performance of Random Forest classifiers were analysed in the workbench Weka. Manual nest estimates taken in the field, and from images were compared to semi-automated nest counts to verify the automated imaging method.

Combined with the ease of model training, construction and application, the Random Forest classifier was well suited to the imaging task. The classifier compared favourably with similar types of machine-learners in analysis performed with Weka. The automated count results correlated well with the manual nest counts taken in the field and manual nest counts taken from images; all methods indicate the number of active nests at the locations monitored have changed over fours years.

The imaging methodology presented in this thesis shows good potential to help increase base-line knowledge and understanding of the population status of native bees in New Zealand. It can also be rapidly adapted for other solitary ground nesting bees worldwide and help to provide much needed information on the health of other 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 Raventhall 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 of the medical imaging package FIJI and would especially like to thank Ignacio Arganda-Carreras, who was ever generous with his time and resources.

...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

1	INTRODUCTION	1
1.1	Objectives	3
1.1.1	Thesis outline	5
1.1.2	Publications and contributions	7
i	REVIEW	8
2	MONITORING NATIVE BEES	9
2.1	Traditional methods	9
2.1.1	Mark-recapture, survey and passive sampling techniques	9
2.1.2	Population methods for solitary bees	10
2.2	Technologies for field research	12
2.2.1	Digital tools	12
2.2.2	Audio data	13
2.2.3	Image data	13
2.3	Relevance of studies	14
2.3.1	Data-intensive science: future field methods?	14
2.3.2	Tool selection: what are the criteria?	15
2.3.3	Image acquisition: are native bees too small?	17
2.3.4	Image verse audio data?	20
2.4	Measuring populations using images	20
2.4.1	Examining closely related problems	21
2.5	Review summary	22
3	IMAGE ANALYSIS	26
3.1	Imaging and scientific discovery	26
3.1.1	Image systems workflow	28
3.1.2	Open-source tools	30
3.2	Image acquisition, format and pre-processing	31
3.3	Image segmentation	33
3.3.0.1	Thresholding by object intensity	34
3.3.0.2	Trainable segmentations	35
3.4	Trainable Weka Segmentations	37
3.5	Verification methods	40
3.6	Data management tools	42
3.7	Review summary	43
ii	METHODS	44
4	FIELD COLLECTION METHODS	45
4.1	Monitoring locations	45
4.1.1	Species diversity	47
4.1.2	Description of monitoring sites	47
4.2	Field monitoring methods	49
4.2.1	Manual nest counts and nest image collections .	50

4.2.2	Control and inactive nest images	51
4.3	Summary of field methods	54
5	IMAGING METHODS	55
5.1	Data management	55
5.2	Computing environment	57
5.3	Image preparation	58
5.3.1	Sorting image collections	59
5.3.2	Pre-processing procedures	61
5.3.3	Overview of file and folder procedures	63
5.4	Trainable segmentation interface	65
5.4.1	Filter descriptions and setting options	65
5.5	Classifier tuning	69
5.5.1	Testing feature importances	69
5.5.2	Testing random forest parameters	71
5.5.3	Classifier benchmarks	72
5.5.4	Segmentation performance of test classifier	72
5.6	Classifier training	72
5.7	Post-processing	75
5.8	Classical segmentation methods	77
5.8.1	Thresholding monitoring images	77
5.9	Manual counts from images	77
5.10	Data preparation and analyses	79
5.10.1	Methods comparisons	80
5.10.2	Statistical tests	81
5.11	Summary of imaging methods	82
iii	OUTCOMES	84
6	RESULTS	85
6.1	Classifier tuning	85
6.1.1	Default features	85
6.1.2	Optimised features	86
6.1.3	Optimised speed	86
6.1.4	Number of trees	86
6.1.5	Number of random features	90
6.2	Classifier benchmarks	93
6.3	Segmentation performance	93
6.4	Training stacks	93
6.5	Monitoring data summary	93
6.5.1	Collections	100
6.5.2	Descriptive statistics	100
6.5.3	Frequency distribution	100
6.6	Number of active nests by method	103
6.6.1	Classical threshold and machine learner	103
6.6.2	Manual-field and image counts	105
6.6.3	Automatic and manual counts	107
6.7	Number of active nests by site and year	109

6.7.1	Mt. Tiger	110
6.7.2	Mt. Parihaka	110
6.7.3	Memorial Drive	110
7	DISCUSSIONS	114
7.1	Research overview	114
7.2	Active nests	117
7.3	Monitoring images	120
7.3.1	Image format	120
7.3.2	Outdoor images	121
7.3.3	Examining nest images	122
7.4	Segmentation methods	123
7.4.1	Thresholding by intensity	124
7.4.2	Canny-Deriche filtering	125
7.4.3	Statistical region merging	125
7.5	Trainable segmentations	129
7.6	Empirical lessons	134
7.6.1	Concepts important for classifier training	134
7.6.2	Feature engineering for nest classifiers	134
7.6.3	Calibrating random forest models	136
7.6.4	Benchmarking classifier performances	137
7.6.5	Segmentation performance	139
7.6.6	Summary of classifier tests	140
7.7	Monitoring classifiers	143
7.7.1	Variability of training images	143
7.7.2	Morphological operators tuned for segmentations	143
7.8	Verification methodology	144
7.8.1	Replica image collections?	145
7.8.2	Manual-field, image and automated nest counts	145
7.9	Performance-cost measures	148
7.9.1	Operating system environment	149
7.10	Imaging pipeline design	149
7.11	Summary of discussions	152
8	CONCLUSIONS	154
iv	APPENDIX	155
A	ACRONYMS	156
B	GLOSSARY	158
C	NOTATIONS	159
D	IMAGE EXIF DATA	160
E	CODE	161
F	RAW FIELD DATA	177
G	REFERENCES	200

LIST OF FIGURES

Figure 2.1	Female native bee grooming dirt from her body.	16
Figure 2.2	Male native bee dislodging load.	16
Figure 2.3	The eight species of native bees in Whangarei.	18
Figure 2.4	Bee size to categorise groups.	18
Figure 2.5	Harmonic radar and I ³ S-Contour.	25
Figure 2.6	Parallel imaging problems.	25
Figure 3.1	Typical image analysis stages and design.	29
Figure 3.2	Simple and difficult segmentation tasks.	36
Figure 3.3	Trainable segmentation procedure.	39
Figure 4.1	Study locations.	46
Figure 4.2	Site species composition data.	48
Figure 4.3	Close-up and wide views of nest sites.	52
Figure 4.4	Signs marking the beginning of the active season.	53
Figure 5.1	Folder structure for image database.	56
Figure 5.2	Image preparation.	58
Figure 5.3	File organising and procedures.	60
Figure 5.4	Sorting image collections.	62
Figure 5.5	Trainable Weka Segmentation (TWS) main training interface.	66
Figure 5.6	TWS features settings interface.	67
Figure 5.7	TWS model options interface.	67
Figure 5.8	Stack of representative images.	69
Figure 5.9	Classifier training and data folders.	73
Figure 5.10	Binary options interface.	78
Figure 5.11	Image data files.	81
Figure 5.12	R-Studio interface for statistical analysis.	82
Figure 6.1	Feature importances test 1.	87
Figure 6.2	Feature importances classifier test 2.	88
Figure 6.3	Feature importances classifier test 3.	89
Figure 6.4	Number of trees.	91
Figure 6.5	Number of random features.	92
Figure 6.6	Test image stack.	96
Figure 6.7	Site 1: training stacks for Mt. Tiger.	97
Figure 6.8	Site 2: training stacks for Mt. Parihaka.	98
Figure 6.9	Training stack Memorial Drive.	99
Figure 6.10	Frequency distribution of nest counts.	101
Figure 6.11	Automatic counts by threshold and model CF.	104
Figure 6.12	Manual counts by two scorers.	105
Figure 6.13	Manual image and field counts.	106
Figure 6.14	Manual image and automatic counts.	107
Figure 6.15	Manual field and automatic counts.	108

Figure 6.16	Yearly nest counts site 1.	111
Figure 6.17	Yearly nest counts site 2.	112
Figure 6.18	Yearly nest counts site 3.	113
Figure 7.1	Image sequences of foraging bees.	118
Figure 7.2	Images of sweep net collections.	119
Figure 7.3	Image quality and preparation for analysis. . .	121
Figure 7.4	Test stack of representative nest images	123
Figure 7.5	Fiji contrast enhancement.	123
Figure 7.6	Images of active nests and thresholding tests. .	124
Figure 7.7	Six thresholding methods.	126
Figure 7.8	Segmentation by edge detection.	127
Figure 7.9	Segmentation by region merging.	128
Figure 7.10	Tests on variable nest images.	130
Figure 7.11	Trainable segmentations on variable nest images.	132
Figure 7.12	Comparison of segmentation methods.	133
Figure 7.13	Image analysis work-flow.	150

LIST OF TABLES

Table 1.1	Development stages	4
Table 2.1	Native bees sampled.	19
Table 3.1	Image analysis scope and topics.	27
Table 3.2	Interactive segmentation method.	38
Table 4.1	Monitoring locations.	45
Table 4.2	Weather measurements and observations. . . .	49
Table 5.1	Computer specifications.	57
Table 5.2	Test classifier models.	70
Table 6.1	Performance results of test models.	85
Table 6.2	Classifier performance and benchmark results.	94
Table 6.3	Final count results from test classifier CF. . . .	95
Table 6.4	Field collection summary of monitoring days.	100
Table 6.5	Descriptive statistics summary.	102
Table 6.6	Summary of comparative analyses.	103
Table 6.7	Summary of monitoring samples.	109
Table 7.1	Trainable Weka Segmentation filters.	133
Table 7.2	Final count results from test classifier CF. . . .	140
Table 7.3	Concordance of correlation guideline	146
Table 7.4	Methods ranked by performance measures. . . .	147
Table 7.5	Decline in the number of active nests.	147
Table 7.6	Image pipeline tasks and tools.	151
Table F.1	Raw manual nest counts	194
Table F.2	Raw monitoring data	199

LISTINGS

Listing 5.1	Bash script	56
Listing 5.2	Operating system configuration.	57
Listing 5.3	Bash snippet to sort collections.	59
Listing 5.4	Collate images into stacks	63
Listing 5.5	Enhance image contrast	63
Listing 5.6	Post-processing snippet.	76
Listing 5.7	R-studio concordance correlation script.	83
Listing 7.1	Fiji output log from test classifier.	141
Listing E.1	Saturated pixels examples.	161
Listing E.2	Stack pre-processing examples.	162
Listing E.3	Thresholding tests.	163
Listing E.4	Thresholding by six common methods.	164
Listing E.5	Otsu thresholding on variable nest images. . .	165
Listing E.6	Segmentation by edge detection.	167
Listing E.7	Segmentation by region merging.	168
Listing E.8	Region merging on variable nest images.. . . .	170
Listing E.9	Trainable segmentation on variable nest images.	172
Listing E.10	Automatic and manual concordance correlation.	174
Listing E.11	Image counts by two observers.	175

INTRODUCTION

Pollination is a natural process pivotal to life on earth. Any changes in services has the potential to affect food security and human welfare [1]. The importance of pollinators was highlighted by the popular press in 2006 when declines in populations of honey bees, *Apis mellifera* (Hymenoptera: Apoidea) were documented throughout the United States (US). The term *Colony Collapse Disorder* was coined to describe the syndrome which rapidly travelled the globe causing alarm. Not surprisingly, as around 80-85% of all cultivated food crops are dependent on pollination from insects; bees are a valuable pollinator group [2]. Studies estimate the economic cost of pollination services could be as high as \$310.9 million US dollars [2]. Others indicate the economic consequences of pollinator declines could result in a reduction of food crops amounting to a total of \$334.1 billion US dollars [3].

The pollination crisis continues to receive attention throughout Europe and the US although the causes of mass honey bee losses remain largely unknown [4]. An added consequence of this phenomenon has been to increase the overall awareness of the role of pollinators. Especially the role of non-*Apis* bees. Many species have been undervalued in the past [5] but in the future may provide a buffer against major colony loss events [6]. In the US, native bees are estimated to contribute about \$3 billion dollars towards fruit production each year [7]. O'Toole [8] explains, "...native bees occupy keystone positions and without them, ecosystems would eventually collapse..." [8, pg.32]

Around 20,000 different bees have been formally described by science [9]. Most of these species are solitary [9]. 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 midlander* (Hymenoptera: Halictidae). It is the most intensively managed solitary bee in the world and is a vital pollinator for alfalfa crops. Alkali bees have a tendency to nest alongside each other so many thousands of bees can form large communities. Some of these communities can persist over decades [10]. One of the most populous and long-lived nesting sites was recorded by Cane [11]. He conducted a monitoring programme on alkali bees over eight years by measuring their populations across 240 km² of agricultural land (Washington, US).

There are at least 40 different species of bees in New Zealand. Around 32 of these are native bees [12]. Most of New Zealand's native bees are solitary ground nesting types. Many have nesting behaviours

which are similar to the alkali bee [10]. They can be described as *gregarious* ground nesting bees. Therefore, at the beginning of the active flight season each year (around September-October), female native bees start to construct their nests in the ground. 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. Baseline knowledge about some species is limited. There are also challenges associated with studies of native bees. Thus long term monitoring programmes are also difficult to manage. 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 [11]. As a result it can be difficult to attract researchers and funding. Most native bees are hard to identify with the naked eye because they look the same. Expert training is required in order to identify them properly and even with the aid of taxonomic keys identification can be difficult [13, 14]. 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 that have been successfully used on larger insects cannot be easily adapted for New Zealand's native bees [15, 16]. Looking towards global research, habitat fragmentation (from urbanisation and agricultural intensification) is an important issue [17]. In the future, communities of native bees may be increasingly 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 [18]. He described the life history of a Māori bee; presenting a range of sketches of larvae, pupae and adult bees. Kelly [19] captured video footage of native bees prising open the flowers of an endangered species of mistletoe. The mistletoe has bird-adapted flowers and was previously thought to be only visited by birds. Donovan's *Apoidea* [12] is the most comprehensive body of research; it is a complete taxonomic treatment of New Zealand's bees. Donovan [20] has also made progress towards relocating native bees. He tested methods for establishing new nests using a species of native bee, *Leioproctus huakiwi* (Hymenoptera: Colletidae). His results were promising. Hart [21, 16] reviewed different methods for studying the foraging ranges of native bees. She collated natural history records from communities in and around Whangarei (Northland, New Zealand) over a three year period. The results from her study showed that at one community up to seven different species of native bees were nesting alongside each other. The diversity of species located within the communities of native bees around Mt. Parihaka (Whangarei) were

the highest that have ever been recorded in New Zealand. Newstrom-Lloyd [6] has provided a comprehensive and current account of pollination services in New Zealand. She suggested more could be done to explore the role of native bees as managed pollinators in New Zealand.

One native bee species (*Leioproctus nunui* (Hymenoptera: Colletidae) is critically endangered, according to a recent report [22]. However, it is difficult to ascertain the conservation status of other species. Native bees might provide essential pollination services. If so, their role New Zealand's ecosystems could be critical. Studies from abroad indicate the consequences of losing pollinator species can be unpredictable and irreversible [23]. Some studies have shown the ecological effects might only be recognised when dependent species of wild and cultivated crops decline [23]. Although the research to date is promising [16, 20, 6] much more is required. Long term monitoring initiatives would help to quantify the population abundance and species diversity of New Zealand's native bees. Image-centric tools could also be used, simplifying ecological methods. If the lessons from abroad are considered relevant, monitoring initiatives might even be community driven. 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 aim of this research was 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 native bees in New Zealand. The techniques for monitoring bees were trialled during the field studies in 2009. The types of digital data were not 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 main 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/quantify *key* indicators?
3. Can image handling, acquisition and analysis be standardised?
4. What pattern recognition/classification techniques best suit the image data?
5. What methods can be used to verify the accuracy/precision of imaging methods?

6. The research questions listed above were used to identify five 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 method for comparative analysis against the image method.
- 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 shown in Table 1.1 below.

myfloatalign

Table 1.1: Development stages

Stage	Tasks
<i>Review Data</i>	Examine past image data and ascertain design constraints; and identify appropriate monitoring sites.
<i>Image Acquisition</i>	Test techniques for standard image acquisition.
<i>Field Methods</i>	Physically 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.
<i>Image Processing</i>	Collate and prepare images. Train and apply classifiers to segment target objects in images. Post-process and count key data.
<i>Verification</i>	Verify the precision and accuracy of image classifiers.
<i>Analysis</i>	Examine and compare manual field and imaging results; review final monitoring data.

1.1.1 Thesis outline

Part I: Literature Review and Theory

There were two main literature reviews. They were drawn from the scientific and technology dimensions of this research. Both were relevant to the development and design of the image-centric monitoring system outlined. The first review was dedicated to examining the field methods used for studying native bees or closely related species. The second review was dedicated to examining the types of imaging methods that are used in biology and life sciences.

Chapter 2 Monitoring Native Bees

Chapter 2 starts with a broad overview of the challenges associated with studies of native bees. The field methods designed for a scientific analysis of native bees or their habitats, are generally very difficult to conduct. There are considerable practical and logistical problems with the methods that are currently used to estimate the population or diversity status of native bees. These issues probably had a cumulative impact; they restricted both the quality and quantity of studies that could be, and were reviewed for this research. Therefore a wider literature search was necessary. Studies with similar types of insects or broadly related research methods were sought. The relevance of literature was constantly referenced back to the specific research tasks outlined.

A broader review of current literature was conducted to show some tools were more practical than others; a wider perspective also revealed there were some studies that were more relevant than others. Especially when a simplified view of the monitoring task was adopted. Regarding the most suitable tools that could aid studies of New Zealand's native bees, the main conclusions that were drawn from a review of current literature settled on digital images and analysis technologies.

Chapter 3 Image Analysis

Biomedical image analysis theory provides the foundation for developing methodologies in a range of other life science disciplines. Software developments within the field of medical image analysis are well represented at the very forefront of research. Significant advances have been achieved, particularly in the areas of open source imaging tools such as ImageJ and Fiji. Both are very powerful but intuitive packages and central to the success of many studies.

Part II: Methods

Chapter 4 Field Collection Methods

Chapter 4 details the design of field monitoring methods. The number of active bees nests were used to estimate the populations of native bees. Three different nesting communities located in and around Whangarei (Northland, New Zealand) were monitored over a five year period (2009–2014). Four areas of representative nests were selected at each location. Areas were marked so the same nests could be monitored throughout and across the active bee seasons. Four plastic grids (245 × 245 mm) were placed over the nest areas. The number of active nests inside the grids were manually counted. Digital images of the grids were collected using a standard Digital single-lens reflex camera (DSLR) camera. Each monitoring day, after all three field collections were completed, the digital images were immediately copied from the camera memory Secure digital device (SD) card, onto an external hard-drive for analysis.

Chapter 5 Imaging Methods

Chapter 5 details the imaging tools and methods used to process the images of active nests. This includes the management of the digital data and the design of the image collections database. Image analysis tools and methods are fully detailed so they can be easily replicated for use in similar applications. Open source biomedical image analysis software Fiji is just Imagej (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 includes 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 Random forest (RF) machine learner, which was the selected classifier used for segmenting monitoring images via TWSs.

Part III: Research Outcomes

Chapter 6 Results

To do...

Chapter 7 Discussions

To do...

Chapter 8 Conclusions

To do...

1.1.2 Publications and contributions

Publications from this research are listed below. Other outcomes and pathways were also important and 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 audience by combining contemporary digital media with Māori design and narratives. The images collected during field monitoring were integrated with modern designs to create a exhibition; this was hosted by the School of Creative Design and Technologies (AUT).

1. Hart N. H. and L. Huang. *Monitoring Populations of Solitary Bees using Image Processing Techniques*. International Journal of Computer Applications in Technology, 50(1): 45–50, 2014.- [24]
2. 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. - [25]
3. 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. - [26]
4. 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 on, pages 353–357. IEEE, 2011. - [27]

Part I
REVIEW

MONITORING NATIVE BEES

OUTLINE

This chapter examines the challenges associated with studies of native bees focusing on the biological field methods typically used to measure species diversity, or populations. A review of conventional techniques is presented by examining two specific approaches for evaluating populations of solitary ground nesting bees using active nest counts. The use of tools to aid field biology research are introduced, highlighting advances in imaging methods. This chapter summarises the progress of technologies for biological research and presents the validation for an image-centric monitoring method for native bees in New Zealand. This chapter also considers a number of perspectives about the synergy of knowledge between applied science and technology, and the benefits that could arise from the cross-fertilisation of ideas.

2.1 TRADITIONAL METHODS

Field research methods which are typically used in biology are often founded on real-world data collections, sampling, measurements or observations. Field data can be messy and inexact particularly when compared to the types of data that are acquired using *laboratory* methods. Natural field data sometimes requires special handling during statistical analyses; especially if it does not conform to assumptions about normality which are required by common statistical techniques. For these reasons considerable care is given to the formation of hypothesis, sampling design and data analysis in studies involving field biology. In the following reviews, methodologies are concerned with multiple species, or groups of insects which are used as indicators of biodiversity. Some studies are specifically designed to measure species diversity while others the population abundance; the methods have or could be used in studies of native bees¹

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

There are several techniques that can be used to study insects. However, mark–recapture techniques are considered the most reliable. Because insects are captured using sweep nets, while foraging or above

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

nest sites, some behavioural information can also be gathered. Mark-recapture methods can be used to quantify species diversity and population abundance but they can be labour intensive. They require individual insects to be captured and uniquely marked, then released and caught again. 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 in repeat observations of over a pre-determined 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 [28] estimated the overall population size of solitary bees by using a small sample of mark-recaptured bees, with an initial observational survey. According to their findings there were good correlations between the methods. They concluded the population size could be reliably established using observational survey walks alone. More importantly they showed their method could save considerable time and effort and thus enable resources to be directed towards long term large scale monitoring. Nonetheless, survey methods are still depend specialist skills and the scientists carrying out observations must be completely familiar with the habitat and bee fauna of survey locations.

Passive sampling methods do not depend heavily on specialist skills so they have some advantages over other techniques. In a comprehensive study, Westphal and Bommarco et al. [29] 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. [30]. By most accounts pan traps are easy to use and reliable [29, 30] but others point out they are subject to taxonomic bias [31, 32, 33]. Traps are also fatal for the insects collected, so the method may not be appropriate for research involving vulnerable species? Moreover, several studies have indicated [31, 34, 33] the data gathered from pan traps do not necessarily provide good abundance measurements. They are more suitable for measuring the diversity of species within a habitat.

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 studies are reviewed in turn; the first is by Bischoff [35] and a second, more recent study by Cane [11].

In a four year study, Bischoff [35] used mark-recapture and nest counts to evaluate populations of a solitary bee, *Andrena vaga* (Hy-

menoptera: Andrenidae). Population estimations using nest count data corresponded well with those estimated using mark–recapture data. Bischoff [35] reported little differences between the cumulative number of nests (164) and the mean population estimations (160) for data collected in 1997. He measured an overall decline in the populations of *Andrena vaga* across four years (1996 to 1999) at the sites he monitored. Attributing some of the population changes to corresponding increases in populations of bee parasitoids. Bischoff [35] also concluded mark–recapture methods provided more reliable data.

In an eight year study, Cane [11] recorded the population changes of the alkali bee, *Nomia melander* (Hymenoptera: Halictidae). Aerial photographs were used to identify fifty-six nest beds across a 240 km² area of agricultural land. Nest densities were surveyed each year using up to twenty, 1 m² grids. The grids were scattered randomly on aggregations and the number of nest holes per grid were counted. Using video recordings, Cane [11] checked the entry holes that were being actively used by nesting bees, against those counted. He found around 66% of those manually counted were nest entrances. Consequently, final nest counts were averaged across the total nest area and multiplied by 2/3. Thus accounting for the differences between the observed numbers of nests, with the actual number of nests. Final analysis showed the population density varied within and between nest aggregations. Over the years 1999 to 2006 the population increased by a factor of nine, to an estimated total of 16.7 million.

Taken together the methods used by Bischoff [35] and Cane [11] support the notion of using nests as a proxy for population abundance. Their studies also highlight the limitations of using nests to estimate absolute population abundances. Manual nests counting methods are likely to produce inflated estimates, at least according to Cane's [11] experiences. There were instances where several females were using the same entrance to multiple nests underground and other cases indicating solitary females were constructing multiple nests. However, despite these limitations, 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 one other study to date [36].

The review also highlighted the paucity of available research. There appears to be a lack of data on the health or populations of many species of solitary bees around the world, not just New Zealand's native bees. In the absence of good data records and with a lack of means to collect any future records, straightforward field methods based on the number of active nests could suffice? At least until alternative field methods have been developed, tested and proved.

2.2 TECHNOLOGIES FOR FIELD RESEARCH

A diverse range of technological tools are available to science and more are developed each year. According to Moore's Law computing power doubles every eighteen months [37, 38] so the rate technological developments and scientific discovery is staggering. Combined with mobile devices, crowd-sourcing projects and increasing connectivity uniting citizen scientists world-wide [39], the volume of data for analysis is changing the scientific landscape. The roadmap is not clearly defined; many studies alter from the traditional knowledge-driven hypothesis testing paradigms to new data-intensive methods [40, 41]. Considering these changes long term meta-analysis of populations of native bees could become a realistic venture within a decade. This review therefore concentrates on 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 [42, 43]. Considerable advances have been made to date, but "the bounds on just what it is possible to achieve remain to be established", as summarised by Gaston and O'Neill [43, p.12]. Automated taxonomic or species identification, refers to computer-based technologies and systems, designed to automatically assign sample specimens into known species using digital data [43]. Several formats can be used based on image, audio, or frequency type collections and all have been used in studies for automated identification of insects. This is a rapidly expanding field of research [44] typified by exploratory design, therefore some novel studies are presented in this review first. These highlight the advances in digital data analysis, notably the use of audio and frequency-based data and classification methods for insect ID.

The main volume of the review is dedicated to the advances in *imaging systems* since these have revolutionised many scientific studies. Specialised tools have been developed for bees [45, 46, 47], water insects [48, 49], live moths [50] and sharks [51, 52]. Combined these studies affirm the *potential* of digital technologies and the benefits that arise from a pooling of resources and knowledge [53]. Many frontier projects share common characteristics: they generally involve a range of specialists, methods are developed using open-source tools – and/or they are designed to be open-access for community science. A selection of representative studies are discussed in the following sections.

2.2.2 Audio data

Since insects are more often heard than seen species identification can be difficult. For rapid identification of invertebrates, techniques using digital audio data and frequency-based classifications are most promising as the following studies demonstrate.

Capitalising on the unique acoustic sounds of crickets, Potamitis et al. [54] used audio data for the recognition of 105 different species. Their system was based on two main steps; acoustic signal parameterisation and classification. During the first stage they determined the features providing the most information. In 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 and a 94% accuracy on the level of species [54].

Raman et al. [55] developed a low cost acoustic insect flight detector to monitor mosquito activity. It was constructed using off the shelf components: a noise cancelling microphone and digital sound recorder. They classified recordings using various harmonic ranges of wing-beat frequencies. Pointing out the research behind mosquito identification from audio recordings dates back to 1949's [56], Raman et al. [55] present a system which they report is relatively inexpensive and could realistically perform well in natural environments.

Similarly, Batista et al [57] developed a low cost optical sensor to count and classify flying insects using wing-beat frequencies. Different species produce distinct wing-beat sounds so they extracted unique acoustic information from recorded sequences. Using Bayesian classification and probabilities they matched wing-beat frequencies with specific species. Large insects produce lower frequencies and smaller insects higher, therefore they suggest classifications are relatively straightforward. They reported an accuracy of 96.04% using a Bayesian classifier on three different species of insects; including one bumblebee, *Bumble impatiens* (Hymenoptera: Apidae) and two mosquitoes, *Culex quinquefasciatus* and *Aedes aegypti* (Diptera: Culicidae).

2.2.3 Image data

Caci et al. [58] used I³S-Contour, Interactive Individual Identification Software² [52], to identify individuals belonging to the threatened beetle species *Rosalia alpina* (Coleoptera: Cerambycidae). Beetles were identified by unique markings on their backs. Selected images were annotated using a semi-automated contour tracing function, matched with unknown images in a reference library and the results given as a ranked picture list. With recognition rates between 94.5–95.2% (using

Cricket ID with digital audio data

Mosquito ID with audio windbeat data

Insect ID with optical wingbeat data

Beetle image ID and matching with I³S-Contour.

² <http://www.reijns.com/i3s/>

290 images) the accuracy of the method was good. Their system could also save time and effort and is perfectly suited for studies concerning vulnerable species.

In a similar study, Towner et al.[51] replace mark–recapture with an imaging method to monitor shark populations around the South African coast. They used digital images of dorsal fins to identify individual white sharks. They collected images over a four year period (2007 to 2011) and used open-source imaging software DARWIN [59] to process 1683 images of dorsal fins; identifying a total of 532 unique individuals. Their results indicated shark populations had not markedly recovered since being nationally protected by the South African Government in 1991 [60].

Mayo and Watson [61] developed an automated species identification system using open-source medical imaging software ImageJ³ [62] and Weka [63]. They compared a range of algorithms to automatically classify 35 different species of moths using images of 774 live individuals. They reported an accuracy of 85% using a support vector machine algorithm, implemented via Weka. Likewise, LoNe et al. [50] developed a real-time image processing system to identify moths in flight in their natural environment. They adapted an algorithm based on randomised trees for training and classification. They achieved an 82% recognition rate with 10 different species of butterflies.

Shark image ID and matching with DARWIN.

Moth ID with ImageJ & Weka.

2.3 RELEVANCE OF STUDIES

There are so many different biological methods that can be used to help gather information about native bees [64]. Only a fraction of these methods have been included in the review. The synthesis was directed towards the methods and tools which could be easily applied or adapted for use, in studies of New Zealand's native bees to increase broad or baseline information about their communities and populations. In the following sections the context of the task is examined closer and in relation to the relevance of the literature reviewed.

2.3.1 Data-intensive science: future field methods?

Multiple disciplines fall within the umbrella of life and natural sciences. Specialists do not always use the same biological methods or field sampling techniques. Consequently, it can be difficult to organise literature into categories for comparative critique. Therefore it was helpful to categorise studies by using the focus of methodologies which could be aimed towards: 1) a understanding of individuals within a species e.g. Caci et al. [58], 2) a selected species group e.g. Towner et al. [51], 3) collections of many different species that combine to form communities e.g. Bischoff et al. [35] 4) larger pop-

³ ImageJ – <http://imagej.nih.gov/ij> and FIJI –<http://fiji.sc/Fiji>

ulations of species on a spatio-temporal scale e.g. Cane [11] 5) or meta-populations or organisms e.g. Murray et. al. [64].

Natural science methodologies could also be broadly classed as *observational* or *experimental*. Observed information is nearly always required before hypotheses are formed, refined and tested experimentally. The scientific method has changed over time; originating from the emphasis on empirical and observational records to a focus on hypothesis testing and experimental design. Modern research methods which are adopting data-intensive techniques are fundamentally observational studies [40]. Some of the technologies that are gaining momentum in biology, lend themselves towards meta-data sampling of large populations or communities of species. Kelling et al.[40] explored some of the difficulties associated with traditional ecological methods. Since many have been reliant on specialist knowledge and skilled collection techniques, there are some practical and logistical problems when a spatial or temporal understanding of species is required. They point out that well designed, data-intensive research methods can draw on a vast network of human *sensors* [65]. These new approaches can thereby increase data capture for the scientific communities to explore and analyse.

2.3.2 Tool selection: what are the criteria?

There are several specialised tools that work well on a range of target insects that might not be so easily adapted for use with New Zealand's native bees or their habitats. Radio telemetry tools [66, 67] were generally excluded from the review even though considerable advances have been achieved in this field over the last decade [15, 68, 69]. This is because it is likely that most species of New Zealand's native bees would not carry the added weight of a radio tag. According to a study by Hart [16] although some species of New Zealand native bees are capable of carrying added loads, they have behavioural characteristics that would precluded the suitability of some tracking technologies. Her observations indicated many species of native bees had behaviours adapted for tunnelling in the soil. They were observed grooming excess dirt from their bodies before attempting flight as shown in Figure 2.1. During load trials she attached added weights to the thorax of bees; in several instances individuals actively manoeuvred into positions in order to dislodge loads [16, pg. 78]. For these reasons, automated and semi-automated taxonomic identification tools were considered more appropriate for studies on native bees.



Figure 2.1: A female native bee has groomed herself to remove clay from her body but some grains remain on her thorax [16, pg.78].



Figure 2.2: A male native bee attempting to dislodge an artificial load attached to his thorax [16, pg.78].

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 technologies available. Most are effectively non-invasive mark–release–recapture *type* tools so they are (or 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 would hinder applications for New Zealand’s native bees. In the first instance the main issues revolve around the types and qualities of images that can be gathered for analysis. According to Donovan [12], many species of New Zealand’s native bees are small compared to other bees. In addition, some closely related species are so morphologically similar they can be difficult to identify properly even using microscopic aids. Previous records collected by Hart [16] were used to determine the species that were most likely to be found at target the monitoring locations central to this study. These were combined with Donovan’s [12] research to examine the biometrics of key species in more detail and in relation to current image analysis methods and tools.

The beetle identification system described by Caci et al. [58] showcases the benefits imaging techniques for studies of vulnerable species. An image-centric sampling method is not fatal for target species and therefore has some advantage over traditional methods. However systems using images of target insects might not be easily adapted for monitoring native bees. Acquisition is a problem because of the size and speed of native bees in flight. Also, distinguishing morphological features are usually only perceptible in close-up views. Collecting good scientific sample images of native bees is therefore the main restriction.

A point and shoot approach, using a typical DSLR camera can result in some useful scientific images but the method cannot be replicated. As shown in Figure 2.4 native bees range in body size from around 3–12 mm so if standard images could be collected *size-features* might be useful to classify bees into broad family groups, or even subfamily groups. Native bees are mostly black in colour unlike the target species in the study by Caci et al. [58]. Compared to the Rosalia longicorn beetle, which is up to 40 mm in body length and has striking physical features (refer to Figure 2.5 – b), the differentiating facial characteristics of native bees can only be captured via microscopic photography as shown in Figure 2.3.

If close-up images could be acquired then shape or colour–features from facial markings could be used to identify bees, using any of the analysis methods used by Caci et al. [58], Mayo and Watson [61] or LoNe et al. [50]. However, image acquisition would most likely be fatal for the bees, unless they were sedated for close-up imaging and then released [46]. For a few species, facial features are not

significantly different and could not be easily used to differentiate between species. For example, there are no obvious differences between sample images [8] and [10] in Figure 2.3. Both are male bees but one belongs to the species *Leioproctus (Leioproctus) huakiwi* (Hymenoptera: Colletidae) and the other to *Leioproctus (Leioproctus) imitatus* (Hymenoptera: Colletidae).



Figure 2.3: The eight species of native bees in Whangarei are represented in close-up sample images [1-16]. See 2.1 for species names. Images from Donovan [12, pp.130–231].

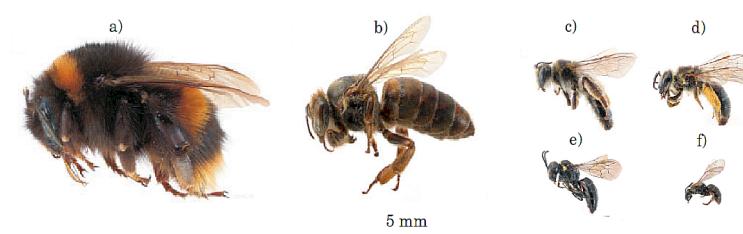


Figure 2.4: Size features could be used to categorise bees into broad family groups: such as a) bumble bees, b) honey bees, and c)-f) four different species of native bees. Images from Donovan [12, pp.130–231].

Table 2.1: Eight species of native bees have previously sampled in Whangarei [16]. Based these records the target monitoring locations central to this thesis were likely to have included some of these species. These bees are shown in Figures 2.3 and 2.4.

Taxonomic tree	Image reference
Superfamily APOIDEA	
Family Colletidae	
Subfamily Colletinae	
Genus <i>Leioproctus</i>	
Subgenus <i>Leioproctus</i>	
<i>boltoni</i>	[3-4] 2.3
<i>huakiwi</i>	[7-8]
<i>imitatus</i>	[9-10] d)♀ 2.4
Subgenus <i>Nesocolletes</i>	
<i>paahaumaa</i>	[11-12] c)♀
Subfamily Hylaeinae	
Genus <i>Hylaeus</i>	
Subgenus <i>Prosopisteron</i>	
<i>relegatus</i>	[11-12] e)♀
Subfamily Hylaeinae	
Family Halictidae	
Subfamily Halictinae	
Tribe Halictini	
Genus <i>Lasioglossum</i>	
Subgenus <i>Austrevylaeus</i>	
<i>sordidum</i>	[15-16] f)♀
Subgenus <i>Chilalictus</i>	
<i>cognatum</i>	[5-6]
Family Apidae	
Subfamily Apinae	
Tribe Bombini	
Genus <i>Bombus</i>	
Subgenus <i>Bombus</i>	
<i>terrestris</i>	a)♀
Tribe Apini	
Genus <i>Apis</i>	
<i>mellifera</i>	b)♀

2.3.4 *Image verse audio data?*

Image acquisition might not be so challenging in the future, especially as the performance of digital cameras 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 and an identification system developed to automatically classify native bees into species? The benefits of an insect identification and counting system using digital audio and wing-beat data would be considerable but there is little evidence the tools developed by Potamitis et al. [54], Raman et al. [55] and Batista et al. [57] and have been used in wider monitoring studies.

This situation might change as the technology becomes more familiar. More so since many of the technologies are being designed for open-access, community-science projects and benefit from crowd-sourced data [41, 65]. Referring to their insect sensor Batista et al. [57] explain, "...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..." [57]. An audio-based identification system for New Zealand's native bees could be worth perusing, particularly for biodiversity studies.

The speed at which new tools and techniques are integrated into ecological methods may be dependent on other intangible factors that may not be easy to overcome? Perhaps more could be achieved towards building greater collaborative relationships across multiple disciplines? Currently, there are many promising technologies have been developed for biological studies (such as those described by [54], Raman et al. [55] and Batista et al. [57]) that do not appear to be utilised? The real benefits of these tools remain largely unknown since they can only be realised when they are incorporated into larger scientific studies.

2.4 MEASURING POPULATIONS USING IMAGES

Broad image data, used on a much larger scale, could also help to describe the population dynamics of species over time and space [40]. However, there are few exemplar studies in this area and no known research around monitoring the populations of native bees using digital images and image analysis. Nevertheless, the progress of imaging technologies for science is well supported by open-source community initiatives; and there are also some parallel studies with closely related problems and possible solutions. These are presented in the preceding sections.

2.4.1 Examining closely related problems

Solis-Sánchez et al. [70] developed a machine vision algorithm to detect whiteflies, *Bemisia tabaci* (Homoptera: Aleyrodidae). By taking advantage of the regular shape of whiteflies, they used geometric image features such as solidity, eccentricity and area, to identify whiteflies from other insects caught in traps. They reported a 97–100% accuracy identifying insects using images of sticky traps and leaves.

Whiteflies with digital images.

They expanded their initial system to identify five different types of insects captured in hunting traps by using their original algorithm, followed by a scale-invariant feature transformation procedure [71]. Their results showed an improvement on their original system; automatic and manual classifications were highly correlated, returning correlations of between 0.96 – 0.99 (R^2). They point out the benefits for pest management in greenhouse crop production environments, since their system was designed to replace time consuming, labour intensive manual counting methods. While they have much to offer to any research involving technologies for biological field data analysis, their system is based on proprietary software so it is most likely limited to a commercial audience.

Checchi et al. [72] presented a method to count the displacement of human populations using high-resolution satellite images. They show that digital images are increasingly used in emergency scenarios; for regional level mapping, site planning, vulnerability or damage assessments and in settings to estimate populations sizes. They calculate there are around 43 million people worldwide who are forcibly displaced, due to armed conflict or other crises. Of those populations, at least 33 million remain internally displaced. Another 10 million people are refugees.

Displaced human populations with satellite images.

Knowing the size of displaced populations is critically important, especially in the first few days of a disaster. According to Checchi et al. [72] good quantitative data is required to help assess any adverse affects on populations; to allocate aid resources effectively, to plan for or mitigate problems due to the arrival of displaced populations. They use manual identification by counting the number of building structures in images, explaining the results are good enough for estimating human populations. Their method is fit for purpose, especially if there are few alternatives and a critical time frame. For example, a ground census conducted by human personnel may not be possible during a natural emergency or in a crisis. Moreover, broad, reliable quantitative population data is required as quickly as possible.

In a final related study, Johansson [73] also uses satellite imagery for the automatic identification of small marine vessels. He explains there are increasing levels of piracy on the sea, so detecting and monitoring activities of small vessels is becoming important. Larger ships

Marine vessels with satellite images.

are regulated and already monitored by automatic identification systems. However, smaller vessels are more difficult to identify, even with the availability of high-resolution satellite data. This is because until recently imaging methodologies have employed thresholding methods to segment image data into key components.

As with other real-world imaging problems, straightforward segmentation of satellite images into 'vessel' and 'all other objects' is more complex. Satellite images are subjected to natural variations. For example, illumination, brightness and contrast are not necessarily even across an image, so thresholding methods perform poorly. In order to compensate for this, Johansson [73] used a specialised filtering technique to highlight objects in images. He used the combined the filtering with a machine learning algorithm, the *Random Forest* and implemented classifications via WEKA⁴. Johansson [73] described his study as novel within the field of vessel detection. Results of the study were exceptional. He achieved an 85–99% accuracy classifying ships in images obtained from Google Maps. He summarised "as the amount of data increases so does the need for efficient methods of processing such data." [73, pg. 1]. With such a wealth of digital images now available, he raises a good point.

2.5 REVIEW SUMMARY

BEYOND THE DATA DELUGE

Bell et al. [74] explain, "Over the past 40 years or more, Moore's Law has enabled transistors on silicon chips to get smaller and processors to get faster. At the same time, technology improvements for disks for storage cannot keep up with the ever increasing flood of scientific data generated by the faster computers." [74, pg. 1287]

According to Bischoff [35] and Cane [11] 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 [11] 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 easily incorporated into traditional field methods and provide an efficient and reliable method, not heavily dependent on expert manual field protocols.

The future prospects of automated taxonomic identification from images for native bees is promising. But there are practical constraints associated the biology and behaviour of many native species that limits acquisition of good scientific images. At least where images of live

⁴ <http://www.cs.waikato.ac.nz/ml/weka/>

insects are concerned. It may also be possible to sedate or fix insects for image acquisition, as Arcuckle et al. [46] do in their automated bee identification system (ABIS). However, the extra tools⁵, time and effort required may defeat the purpose, at least in terms of the aims of this thesis. This is equally true for audio-based collections, since most are used for species identification and not aimed at collecting rapid broad, spatio temporal population data (Section 2.2.2). The objective of this thesis is to collect reliable base-line information on the health of native bee communities, therefore a *simplified monitoring* method is paramount. The techniques outlined in this thesis could also be viewed as first line tools, highlighting potential population changes and identifying locations where more in depth diversity research is needed.

Once image data has been reliably collected, image collation, management and analysis choices and tasks become central. As Hey [75] explains, " 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." [75]. Data-intensive research brings much opportunities and great challenges [40, 76]. The ways to handle and process vast amounts of information to provide useful results are not always as easily defined as expressed by Johansson [73]; similar concerns are echoed by many others [40, 75, 77]. Certainly there are rapid swings and gains in scientific knowledge [78]. Technologies continually improve in performance and power, doubling every year [38]. Increasing connectivity unites large networks of human sensors [79] as e-science rapidly replaces traditional empirical methods [75, 74].

Many more organisations recognise the benefits of interdisciplinary research; the cross pollination and ideas that emerge from a synergy of different skills and perspectives, that enrich discovery [53]. But as Hampton et al.[80] acknowledge, " The scientists who contribute such information will be at the forefront of socially relevant science – but will they be ecologists? " [80, pg. 156]. Possibly not, since while other scientific fields embrace the era of big data, many branches of ecology are slow to move toward more open models of research [80]. At the other end of this argument, Lindenmayer and Likens[77] 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." [77, pg. 338].

Certainly both perspectives have merit, at least with regards to this thesis, which presents an image-centric system intended for practical applications in biological science – to monitor spatio temporal

⁵ Such as laptop, dissecting microscope, macro lens and flash lighting equipment

changes in populations of New Zealand's native bees using images of active nests and semi-automated image analysis. Without doubt this thesis would not meet the criteria for rigorous scientific analysis. However the image-centric techniques presented do demonstrate the potential use in future monitoring. If the tools were used as part of a wider monitoring programme, the data could contribute to a greater understanding of native bees and their ecosystems. This could be a vast improvement on the current state of knowledge.

Interesting points were raised by Hampton et al. [80], Lindenmayer and Likens [77] as they both elucidate the current issues affecting ecological research. 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. Biomedical imaging research collaborations have culminated in dynamic and useful open source tools, such as Fiji and ImageJ [62]. But there are no such clear paths for many ecological studies and a limited number of examples to draw upon for inspiration. This is somewhat the case in this thesis, as there are no known examples, either in New Zealand or abroad, where images have been used to monitor populations of ground nesting solitary bees.

There is some inspiration found in a wider domain by looking at similar problems and parallel studies. Therefore in the final sections of this review 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 is established. However, in order to implement the many other specific aspects of this method (i.e. image storage, archiving and analysis), biomedical imaging research often provides some of the best methodological examples to follow. The advances in biomedical imaging systems have had a significant impact in the fields of human biology and medicine. They can also be applied to a range of other life-science research and have provided the foundation for other novel image-centric studies. These are introduced in the next chapter with a review of the developments in open-source imaging systems.

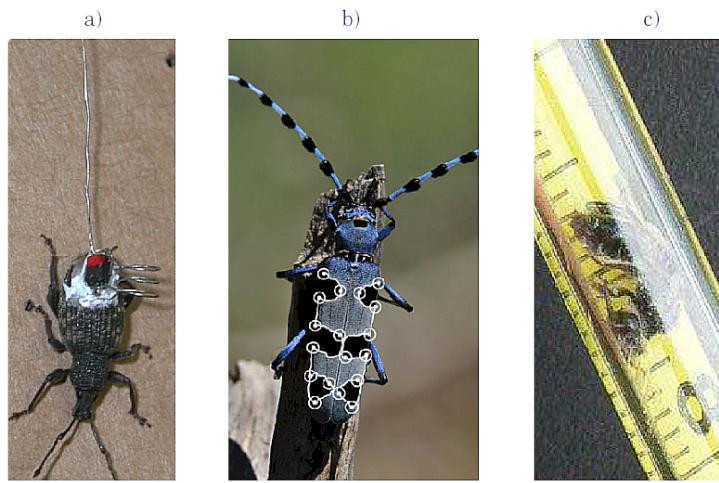


Figure 2.5: From Barzee et al. [81] a) Harmonic radar transponder attached to a weevil, Caci et al. b) I³S-Contour annotations on the longhorn beetle [58, pg.788] and Hart [16, pg.78] d), A native bee *turning around* inside a 5 mm wide tube.

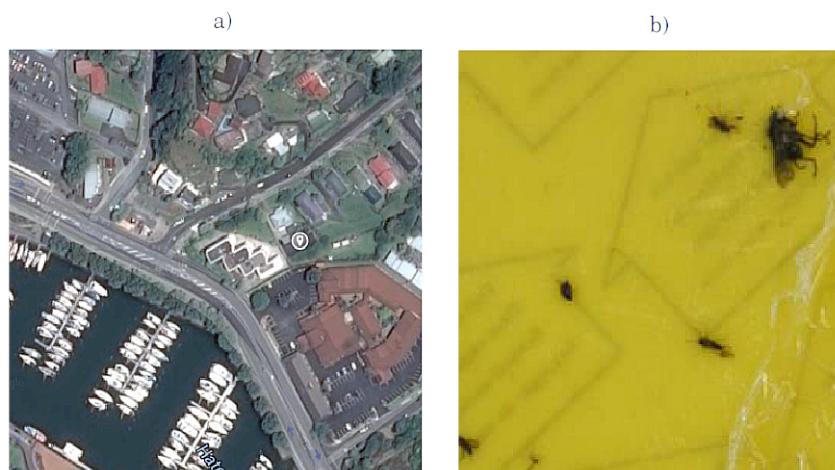


Figure 2.6: Parallel imaging problems. a) Image of Whangarei (©Google Maps) similar to those used by Checchi et al. [72] and Johansson [73] and b) Image of a sticky trap similar to those used by Solis-Sánchez et al. [70]

IMAGE ANALYSIS

OUTLINE

The scope of image analysis is wide and continually developing. Many of the advances in tools and techniques have evolved from the fields of biomedical imaging. Image analysis is central to many scientific methods, particularly where microscopy is concerned, therefore techniques are well established. Imaging methodologies for other life sciences, such as agriculture and ecology have not been as widely reported on although this situation is changing as the hardware tools and software methods are becoming increasingly more powerful. Open source technologies are vital since many of the developments within the field of image analysis has depended on, or contributed towards reproducible scientific methods. Segmentation tools are key to most imaging tasks and methods have traditionally depended on high quality, laboratory acquired photography. This is no longer the case as it is now possible to quantify a range of digital images using new algorithmic machine learning methods, and interactive segmentation techniques. A review of some key aspects of image analysis theory and design are outlined in the following Chapter, with an emphasis on the practical tools and methods used in developing a generic image analysis system. The literature is discussed with the context of an image-centric method to monitor New Zealand's native bees using images of active nests.

3.1 IMAGING AND SCIENTIFIC DISCOVERY

Science and photography have a close interdependent history dating back to the *heliographic* experiments of the 1820's [82]. Imaging technologies are continuing to develop rapidly; according to the observations made by Moore the performance of hardware tools roughly doubles each year [74]. The capabilities of modern imaging systems are remarkable as they are now integrated into everyday technologies such as cellular phones and laptop computers. Within the scientific domain sophisticated digital imaging systems are contributing to all manner of discoveries. From the intricate workings of the human brain in Single Photon Emission Computed Tomography imaging systems, [83] to mapping the universe with the billion-pixel camera on the space probe Gaia [84]. The scope of image analysis is wide and it includes many distinctive strands of knowledge that have coalesced to form the current theory (Table 3.1).

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.

The most comprehensive reviews regarding imaging methods are often directed towards biomedical image analysis. Microscopy is one of the fundamental tools of biology and traditionally scientists relied on visual interpretations of microscopic images. Consequently many advances towards automated imaging and analysis has been developed by and for biologists. Ljosa and Carpenter [85] 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 imaging resources; they also outlined the pitfalls to consider when using or designing a quantitative imaging systems [85]. Across disciplines image-tasks and pipelines are generally similar, therefore Ljosa and Carpenters' [85] manuscript provides some useful theory for other imaging research. Similarly two other reviews, the first by Sharmir et. al. and second by Antony et. al., are equally informative; providing summaries of current software tools and techniques for life science image research [86, 87].

A handful of reviews are written for other areas of life science disciplines. For example, Pennekamp and Schlickzelle [88] presented a hands on guide for imaging techniques in experimental laboratory systems. They introduced past, present and future benefits of technology with an overview of methodologies for experimental laboratory systems, explaining "...despite the advantages of image analysis, the technology has not been fully adopted yet, presumably due to the difficulties of technical implementation." [88, pg.485]. They pointed out the range of benefits that could arise if automated image analysis and experimental laboratory systems were integrated. Imaging tech-

Overview of quantitative analysis of microscopy images.

Pattern recognition software and techniques for biology.

A guide to implementing image analysis for ecology and evolution.

niques can provide a fast reliable and low cost method to increase in the data for analysis in a range of biological parameters.

Gaston and O'Neill [43] reviewed imaging systems for pure and applied biology and questioned the reasons why automated species identification has not been widely adopted. They investigated the 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 that "...vision and enterprise are more limiting...rather than the practical constraints of technologies." [43, pg.1].

Automated species identification systems for biology.

3.1.1 *Image systems workflow*

Image analysis workflows consist of two major steps, image acquisition (capturing digital images) and image analysis (manipulating and measuring digital image data). These are generic stages across all methods, and disciplines. Glasbey and Horgan [89] describe five distinct stages of images analysis as listed below. These are expanded 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 fields great attention is placed on the fidelity of raw image data, particularly with regards to method replication [90]. Guidelines on the appropriate use and manipulation of scientific digital images were presented by Cromey [91]. They are broadly relevant to other scientific imaging applications, although some points are not practically possible when collecting 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 images is usually acceptable.

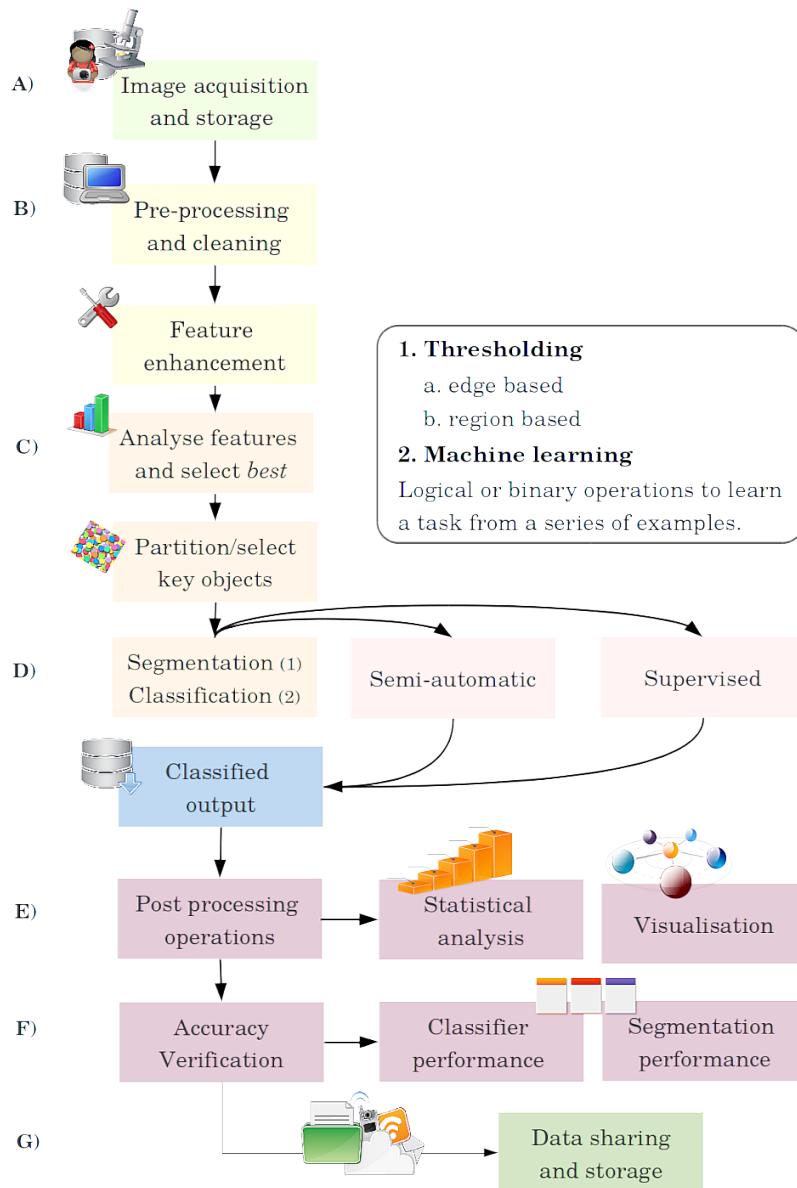


Figure 3.1: Typical image analysis stages and design.

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.

3.1.2 *Open-source tools*

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 and parallel the advances in computers and imaging software tools. Increasing public access to technologies is changing much of the scientific landscapes, since the tools and methods are often readily available, easy to use, open-access and open-source.

For instance, although there are many good proprietary platforms available, open source software solutions are very popular and continue to be at the centre of many scientific advances. For example, Antony et al. [87] reviewed imaging software tools and found open source packages offered substantial benefits, particularly with regards to reproducibility of methods [92, 93]. More generally, the importance of open access was declared by a group of Nobel laureates; they explained that open access "...expands shared knowledge across scientific fields, it is the best path for accelerating multi-disciplinary breakthroughs in research." [94].

Open source tools, methods, and open-access reporting can work in unison to advance the scientific knowledge [95]. The versatility of software tools, access to code, and replication of methods are important factors that are worth considering during research design. As Antony et al. [87] explain, " open-source solutions facilitate community–driven efforts in the development of image analysis " [87, pg.13]. The extendibility, interoperability, scalability of software platforms are key factors that have an impact on overall research methodologies. These are defined as:

1. Expandability: A system design principle taking future growth into consideration.
2. Interpolatable: The capability of different programs to exchange data using a common set of formats (to read and write the same 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.

*Open-source,
community driven
image systems for
biology.*

Fiji [92] is a popular biomedical imaging toolbox that has been carefully designed to consider all three factors listed. However, there are other packages commonly used for image analysis including: ImageJ [62], Icy¹[96], the Matlab plus Image Processing Toolbox [97] and R-EBImage package [98].

3.2 IMAGE ACQUISITION, FORMAT AND PRE-PROCESSING

Digital image acquisition, formats and pre-processing methods are important aspects of image pipeline design. In biomedical imaging fields great attention is placed on the fidelity of raw image data, particularly with regards to method replication [90]. Final decisions can have cumulative impact on the integrity and analysis of image data, the validity of results and the reach of scientific outcomes. The quality and consistency of image capture is dependent on the acquisition methods. Under laboratory conditions important factors can be controlled such as lighting (i.e. brightness and background illumination) and the total capture area. Because there is greater control over image acquisition, the quality and reliability of image data and analysis is more measurable and methods are easier to replicate [99].

Natural images are complex and 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 standardise acquisition. For example, Burks et al. [100] used image analysis for agricultural monitoring to identifying different types of weeds. They designed a specialised image acquisition system to capture field images. They used a wheel mounted self contained imaging device which included four adjustable flood lights with diffuser covers to eliminate shadows [100]. 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" [100, 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, with supplemental lighting to compensate for the effects of cloud cover and angle of sun. A nylon canvas was also used as a wind break at the base of camera system to prevent excessive motion in the plant leaves [100].

Over the last decade image acquisition techniques and analysis for agriculture have advanced considerably. For example Unmanned aerial vehicle s (UAVs) are now used to capture hyper-spectral images of crops [101, 102, 103, 104]. New image capturing tools can help reduce the affects of outdoor variations, however, the challenges described by Burks et al. [100] are still fundamental constraints faced by most image-centric methods. From a practical perspective image

¹ Icy: <http://icy.bioimageanalysis.org/about>

acquisition factors can be fixed, particularly when *field collected* image data is a prerequisite for many environmental studies. Nevertheless, there are new *classification techniques* which will help to mitigate the affects of inconsistent image capture.

For instance Feng et al.[105, 106] employed an off-the-shelf digital camera for UAV imaging 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 DSLR camera typical with a RF classifier and reported good outcomes. RF classifiers offer some advantages over other image segmentation procedures, they are particularly robust to noisy image data and are easy to train [107, 108]. Therefore they are a good choice for the studies by Feng et al.[105, 106] and other closely related ecological imaging problems discussed throughout this manuscript.

Data acquisition, format and storage decisions are interrelated. Compression of image data can be an important consideration when designing imaging systems. Large data volumes create storage problems and also impact the speed of image processing [109]. 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 Joint photographic experts group (JPEG) files and 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 remote sensing applications [110, 111] and continue to be used in a range of image-centric studies.

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 and there are some flexible solutions such as XnView²[112] which has the utilities to open Raw files and export to a range of standard image formats. Biomedical imaging software Fiji and ImageJ[62] also have plug-ins specifically for opening Raw files. 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 and they are easily captured under laboratory conditions. When outdoor image data is required on-camera SD memory card capacity may be limited and 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 [109]. For

² XnView Software: <http://www.xnview.com>

example, Zabala & Pons compared the affects of JPEG and JPEG 2000 (J₂P) compression on remote sensing image classification for mapping crops and forest areas [110, 111]. They found overall that J₂P compression was more reliable than JPEG, at least for the specific categories images tested [110, 111]. Others have shown image compression is not necessarily an issue for some classification tasks. For instance, Paola & Schowengerdt [113] tested three different classification scenarios and found that high quality classifications could still be achieved with a Compression ratio (CR) of 10:1 [113]. Levels of compression below 10:1 are within the boundaries considered acceptable for image classification applications as recommended by Paola and Schowengerdt [113], Lam and Yuan [109] and Zabala and Pons [111]. The CR is defined as the number of bytes of the original image over the number of bytes of the compressed image as follows:

$$\text{CR} = \frac{\text{original image data volume}}{\text{compressed image data volume}} \quad (1)$$

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 Red green and blue (RGB) to grey-scale. Image conversions are frequently required before segmentation operations can be applied. There are generally several options 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.

3.3 IMAGE SEGMENTATION

A *region* in image analysis 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 pixels³. 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.

Thresholding is a region-based, direct method used to turn grey scale images into black and white binarized form. Although it is

³ Segmentation can be defined as "the division of an image into spatially continuous, disjoint and homogeneous regions." [114, pg.215]

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 partly subjective and the performance of different methods are relative. But 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, with some 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. In the proceeding sections three representative segmentation techniques are briefly outlined.

3.3.0.1 *Thresholding by object intensity*

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 Wang et al. [115] 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." [115, 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 [85] 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; when collected under natural environments the processing becomes more difficult. For example, Figures 3.2 (a)–(e), show two imaging pipelines. On the left a simple thresholding task, and on the right an example of a difficult segmentation. The distribution of pixel intensities via the histograms for each image is shown below each image.

A global threshold is set with values between 0-130 pixels. Pixels with values less than 130 are foreground objects, and converted to a binary value of 255 (black). Pixels above 130 are background objects, and assigned a binary value of zero (white). The binary image result can be further analysed using morphological operators. The insects can be categorised into species based on unique characteristics of their size and shape. For example, native bees (*Leioproctus spp.*) have a more rounded body shape compared to native parasitic wasps (*Pseudofoenus spp.*), which have a long slender abdomen. Contrast this segmentation problem to the example image on the right. This image shows a native bee in flight and a honey bee foraging on the flowers of coastal five finger plant, *Pseudopanax lessonii* (Araliaceae). It is not possible to select a global threshold value that can adequately segment the image to show both species of bees and all other background data.

If images are not easily segmented, there are some alternative approaches. Frequently, there are other characteristics that define an image, such as connected structures, outlines, areas or textural qualities. For example, two alternative methods are edge detection, and region merging. *Canny-Deriche filtering* [116] is a popular edge detection method. 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. Another technique is Statistical region merging (SRM). It is a region-based method and it performs well on a range of images [117, 118]. Q – the setting determining the approximate number of regions to segment. The algorithm examines one region per pixel and applies a statistical test on neighbouring regions in ascending order of intensity differences, to test if the mean intensities are sufficiently similar enough to be merged. Segmented regions can be represented by mean grey values as shown, or by index of the regions. Nock and Nielsen [117, 118] explain the SRM is a very fast segmentation algorithm which copes well with noise and occlusions [117, pg.1458].

3.3.0.2 Trainable segmentations

In most analyses *any* aspect of the imaging pipeline can be tweaked to accommodate image characteristics or to highlight the key data required for specific analysis applications. However, 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 this circumstance, machine learning techniques are far more effective. It is also the basis for the growing tendency towards using machine learning in challenging imaging tasks, as proposed by Ljosa and Carpenter [85].



Figure 3.2: Examples of simple and difficult segmentation tasks. The images of (a) insect collections and (b) insects in flight. There is no single threshold values for the images of insects in flight; indicated by the multimodal histogram (h).

Trainable image segmentation techniques work by utilising human visual knowledge to provide a machine learning algorithm with a set of *expertly labelled examples*. For example, in Fiji the TWS plug-in [119] requires a user to provide two sets of labels. Regions of interest (ROI) tools are used to select pixels samples belonging to foreground (class 1) and background (class 2) 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; they can be grouped according to their main filter functions as shown in Table 7.1. During the learning process an machine learning algorithm uses the examples provided, and the features stack, to construct a purpose built classifier. The classifier can be used to segment similar types of images, including the one it was trained on. It is ambitious to ascertain what combination of image features best describes key objects, therefore some aspects of classifier training are subjective. There are also a number of other confounding decisions to consider before applying machine learning for image segmentations. However, providing a user selects appropriate representative pixel samples and chooses filters that will provide a rich features stack for the analysis, machine learning algorithms work very well and can surpass other methods. Especially on challenging imaging problems.

3.4 TRAINABLE WEKA SEGMENTATIONS

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 – a topic which was briefly introduced in Section ???. New 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 [73] novel small vessel detection study, to the UAV flood mapping system proposed by Feng et al. [105, 106].

The TWS plug-in was designed for pixel level segmentation via semi-supervised learning primarily aimed towards biomedical imaging applications. As previously outlined, a user selects a set of features (e.g. such as edge detectors, 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 returning a segmented image, 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, so the process is iterative and interactive. The time taken

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.
-

to train and classify data can range, depending on the image size, the amount of features selected, the chosen classifier and the number of cores of the processing machine. The default classifier is the *FastRandomForest*. The algorithm is a multi-threaded version of random forest which is initialised with 200 trees and 2 random features per node. There are also five default training features (from a possible twenty), automatically set which include: Gaussian blur, Hessian, Membrane projections, Sobel filter and Difference of Gaussians.

The general procedure used for classifier training in TWS is outlined in Table 3.2. Most training consists of at least two runs. The first returns broad segmentations as shown in The second training run is optimised by selecting a few previously miss-classified ROI areas for re-classification. The main concept is to supply the RF classifier with only a very small representative sample of classes. This enables the classifier to *learn* to recognise, rather than *memorise* target pixels⁴. When the results of classifications are satisfactory it is possible to save a fully constructed classifier for use in other similarly related tasks. TWS provides both the option to save as a standard Weka model file or immediately apply a constructed classifier via the Graphical user interface (GUI). As described previously, all the training annotations can also be saved as an arff file. The 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.

⁴ The concept of over-training is

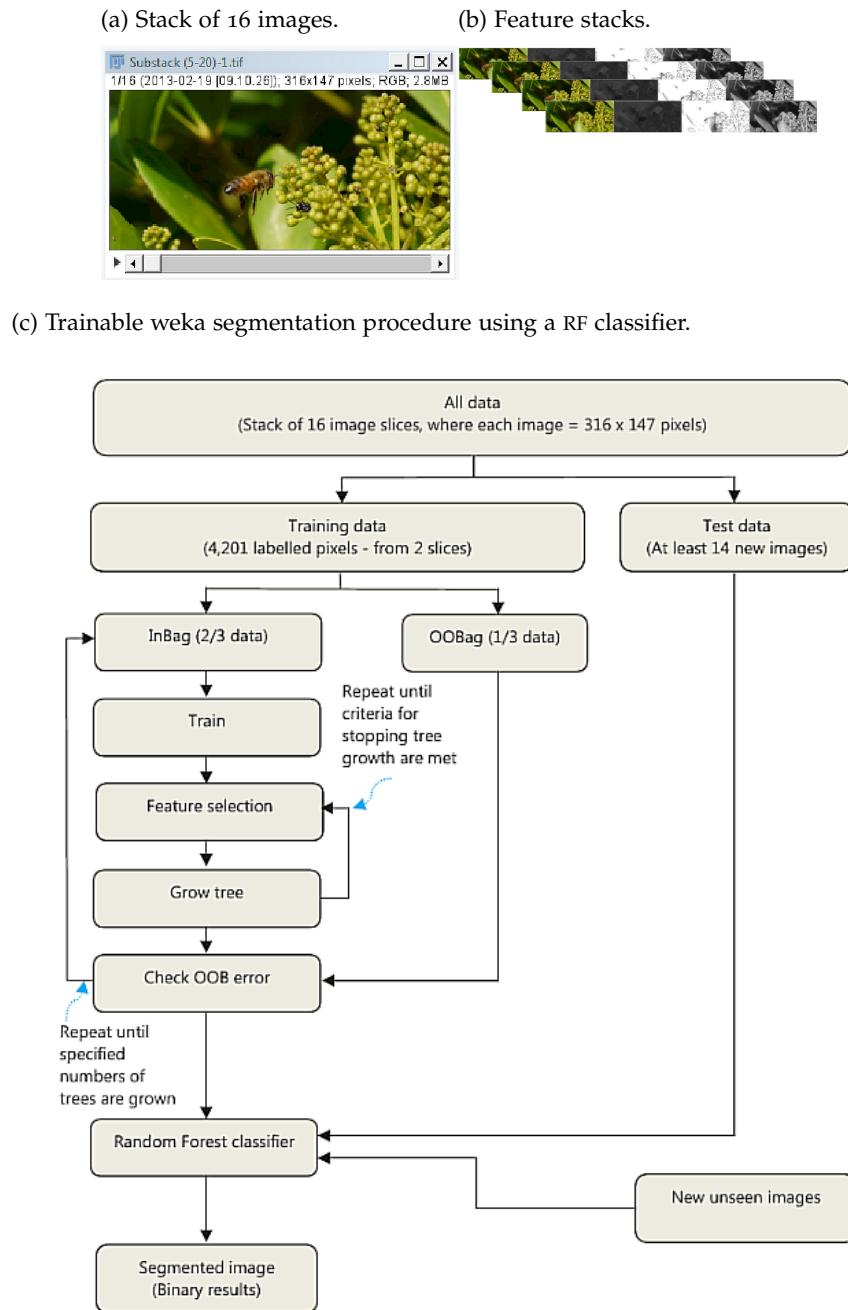


Figure 3.3: Trainable segmentation procedure using an RF classifier. An RGB stack of 16 (a) images are used to build the model; features stacks are constructed for each image slice (b). Sample traces are interactively selected by a user to train the RF classifier using the flowchart outlined in (c).

The default classifier in the TWS workbench is the RF. However, there are at least fifty other Waikato environment for knowledge analysis (Weka) classifiers available. Each machine learning algorithm is equipped with a set of parametrisation tools. 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. Machine learning is a popular area of research is rapidly evolving as more algorithms are developed each year. Traditionally, support vector machines and neural networks were considered state-of-the-art. They have been used very successfully in a range of classification applications using real-world data. Until recently, no other machine learners have surpassed the performances of support vector machines or neural networks.

But according to a recent study by Fernandez et. al. [120] the RFs are most likely to perform the best. Fernandez et. al. [120] 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 data base⁵ (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.

A range of other studies have reported the on benefits of RFs. 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 [121], biomedical imaging [122], economic forecasting [123], genetics [124], pharmacology [125], species identification [48], ecological modelling [126, 127, 128], remote sensing for land-cover classification [129], carbon mapping [130], forest classification [108], UAV flood mapping [105, 106] and invasive species mapping [131, 132]. 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. [120] 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.

3.5 VERIFICATION METHODS

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

⁵ <https://archive.ics.uci.edu/ml/datasets.html>

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 (oo_b) which 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.

In machine learning research, other parameters can be used to help determine the predictive accuracy of a classifiers. These typically 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 so it important to use appropriate performance measurements. Classifiers constructed using natural images can be more difficult to measure and this is more so if training datasets are skewed or unbalanced. When data is unbalanced the Kappa statistic is more informative because it is a *chance-corrected* measure of agreement between classifications and true classes [133]. The statistic is calculated by taking the agreement expected by chance away from the actual observed agreements, and dividing this value by the maximum possible agreement as follows:

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

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 and $\kappa = 1$ shows a very strong agreement. Any value above zero indicates the classifier is, at the very least, performing better than by chance alone.

In many applications segmentation metrics give more insight, since they indicate how well images are partitioned into categories [134, 135]. This is an important consideration for biomedical imaging problems since most performance measures are based on visual-image checks using ground-truth techniques. Until recently, classifier performance has been a subjective measure. As Unnikrishnan et al. [136] 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." [136, pg. 929]. Biomedical imaging method more frequently employ three quantitative segmentation metrics. They include the Pixel, Rand and Warping errors. For image segmentation analysis the ideal metrics for machine-human disagree-

ment should tolerate minor differences in boundary location, penalise topological disagreements, and serve as a cost function for supervised learning [134, 135]. 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*: the measurement of similarity between clusters of pixels. The lower the error the greater the agreement is between images.
3. *Warping Error*: 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.

3.6 DATA MANAGEMENT TOOLS

Database design are becoming vital components for research involving big data analysis. Sharing protocols are well established in some disciplines such as 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 [137]. Similar points have been discussed previously in relation to the use of big data[80] and the slow adoption of automated image analysis [43, p.12]. Data sharing is not always considered virtuous [77] and some authors suggest paradigms are slow to change for some disciplines [138, 80, 130]. However, the progress towards open access, open-source and data sharing well advanced a range of interdisciplinary life science disciplines, which can provide good examples to follow [139, 140].

For example, the Knowledge network for bio-complexity (KNB) provides the facilities to store, share, discover, access and interpret complex ecological data and can be integrated with the open-source statistical package R, or used with a desktop application Morpho which allows users to create metadata for describing data in a standard format, and to search, edit or view data collections. There are a range of generic options which can be used to store data online, for example, cloud storage such as Dropbox⁶ can be used for single users or OpenStack⁷ for large distributed community developments [141]. Github is also a popular repository for collaborative project developments and code design. While it was not specifically designed for sharing data it is used to as centralised hub for smaller research projects [142, 143].

⁶ <https://www.dropbox.com/>

⁷ <http://www.ubuntu.com/cloud/ubuntu-openstack>

3.7 REVIEW SUMMARY

THE IMAGING METHOD

Delaney [82] writes, "from the earliest days of the photograph to the present, its value has been recognised by the scientific community to help them document and understand the natural world." [82, pg.76].

Given the variety and complexity of some machine learning algorithms, and the overall objectives of this thesis, there were few reasons to examine the performances of other classifiers or investigate the internal workings of the RF classifiers. Both lines of these investigations would have quickly led into the areas of machine learning and statistical analysis, which are beyond the scope and expertise of this research. Rather, when taken on the whole, it was difficult to look past the evidence indicating the RF model would be a good tool for the segmentation tasks associated with the image-centric bee monitoring system.

The performance characteristics of RF classifiers can be internally measured during run-time, externally assessed or compared in Weka using statistical tools such as a confusion matrix, or quantified using segmentation metrics.

Part II
METHODS

FIELD COLLECTION METHODS

OUTLINE

Previous records were used in conjunction with one off surveys to identify good monitoring locations. When locations were identified a standard programme was initiated. A range of static and dynamic image data were collected during field work but 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

Table 4.1: Monitoring locations and map co-ordinates.

Site number	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

Several locations around the greater Whangarei district were identified as potential sites for an initial survey and for the development of a multi-seasonal monitoring programme. Selection priority was given to locations that were easy to access, reasonably safe and unlikely to be modified by future public works or developments, or by the wider public. Secondary priority was given to sites known to have supported communities of native bees; or sites with good species diversity, large nest aggregations or high numbers of foraging bees [16]. Communities with different biological structures were chosen in order to collect a range of image data and test the validity of image-centric monitoring system. It was anticipated the range of image data collected would also realistically reflect the natural variations in community structure and simulate real-world conditions. Two geographically separated locations were identified and selected for repeat monitoring on Mt. Tiger and Mt. Parihaka. Another location, Memorial Drive, was included in monitoring from season 2010.

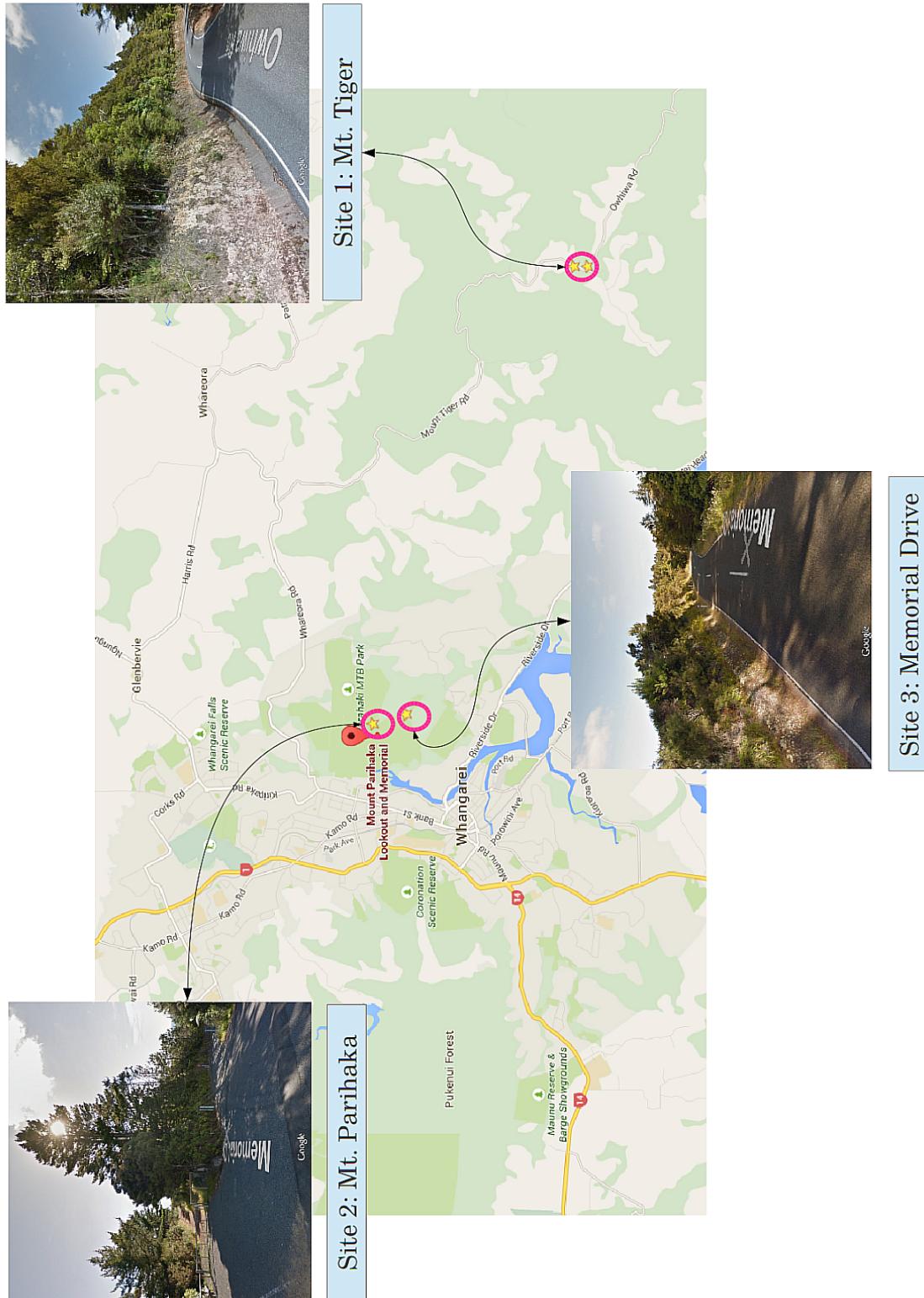


Figure 4.1: Google Map © showing the site locations and photographs of each monitoring area.

These locations are outlined in Figure 4.1 and include general images of the sites and the layout of nest aggregations. The map co-ordinates are listed in 4.1 for reference.

4.1.1 Species diversity

Existing records were used to determine baseline data. This included the known species composition and approximate population densities at locations selected for monitoring [21, 16]. Six species have been identified at these locations previously [21, 16]. Five species from the Colletidae (plaster bees) family as follows: *L.boltoni*, *L.huakiwi*, *L.imitatus*, *L.paahaumaa*, *L.pango* and a single species from the Halictidae family (sweat bees), *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 (Figure 4.2).

4.1.2 Description of monitoring sites

SITE 1 Mt. Tiger is located around 20 minutes east of Whangarei. The monitoring site was located just opposite 510 Owhiwa Road, encompassing a roadside bank around 15 meters long by 10 high, with a bank slope of around 60-80°. The bank consisted of exposed reddish clay soil with areas of sparse ground cover to areas of dense vegetation. The site could be described as a typical rural roadside bank. It was bordered by some native shrubs such as maunuka and kanuka, as well as introduced plants such as ox-eye daisy. The main bank area backed onto agricultural land which appeared to be mainly used for dry life-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. There were no obvious signs of commercial honey bee activities in close proximity to the monitoring site. During most years there were up to three-four hundred commercial honey bee hives distributed along Mt Tiger—Owhiwa roads. These could be seen from a moving car along the main north-south routes.

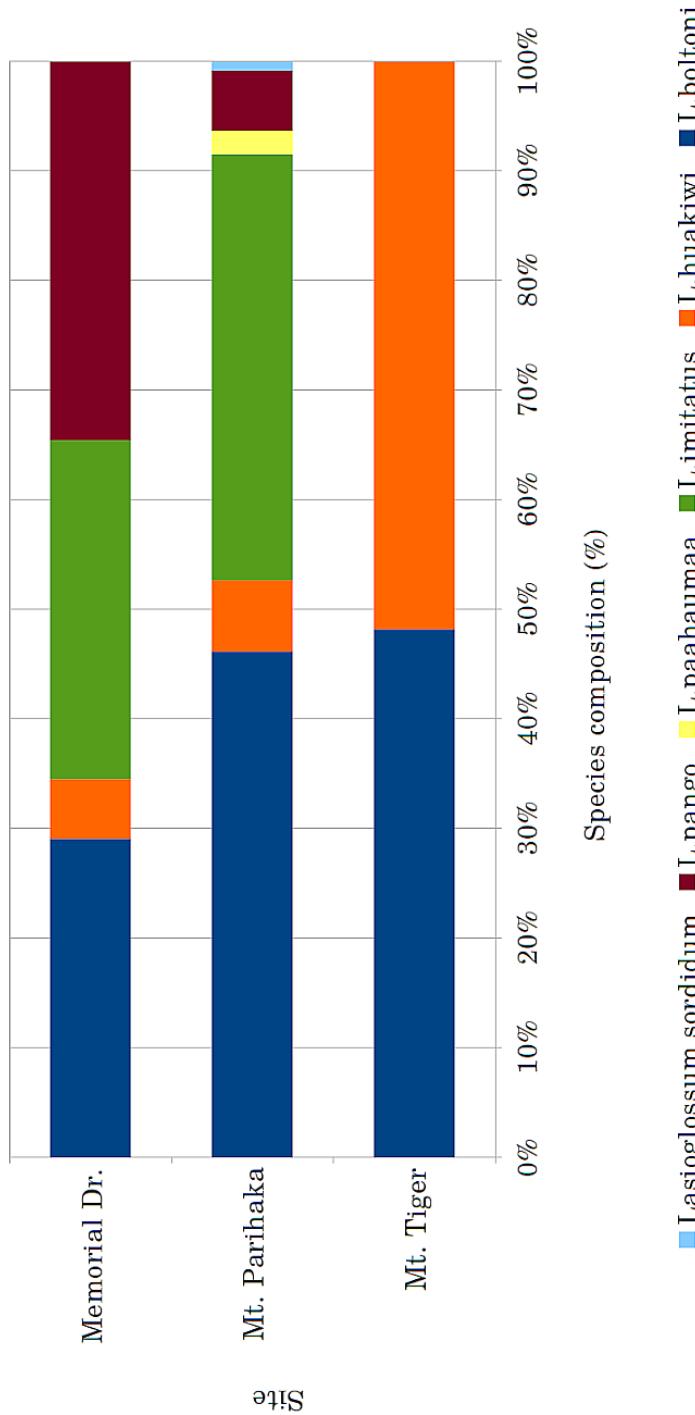


Figure 4.2: Site species composition data from previous studies [21, 16].

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). |
-

SITE 2 Mt. Parihaka¹ is a historical pā located a few minutes from Whangarei central via Memorial Drive. The monitoring site was situated just through a gated forestry area, encompassing a small bank (around 5 meters long by 8 high) with a bank slope of around 60-80°. The bank consisted mainly of white clay soils 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. Pine was the predominant cover but there was gorse and different types of regenerating shrubs such as manuka and kanuka which 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) but there was no evidence of this during monitoring. The area where active nests were monitored gradually 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 roadside bank approximately 5 meters long by 2 high, with a bank 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 in 2009 through to 2014 were conducted weekly in the months of September and October. Once signs of tumuli, or native bees in flight were observed, monitoring was initiated and continued on each

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 although unfortunately some of the confusion about the correct name of this historical pā site still persists (refer to <http://www.beehive.govt.nz/node/23727>).

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) and finished at Memorial Drive (between 1100–1300 hours).

Standard weather and observational data were collected as shown in Table 4.2. At each monitoring location local micro-weather measurements were collected using a Kestrel Meter 1000 Wind Meter. Three measurements were taken and the average results were recorded. Subjective quantities were recorded including the percentage cloud-cover and estimates of the flight-foraging and nesting activity of native bees. The raw data from collections is attached in Appendix for reference.

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 labelled with date-time, site and collection details. Collections were processed immediately after field monitoring was completed. They were placed on standard graph paper, photographed (under natural lighting conditions), re-packaged into containers and couriered to Dr. B. J. Donovan² for professional taxonomic identification and to be included in national entomological records.

In 2009 Northland was affected a drought. Surveys were conducted weekly, but native bees did not emerge as expected. Most sites were only slightly active in late December. Some bees finally emerged in early January but within a week, most had expired. Therefore, 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 and 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. Once constructed, the final internal dimension of each grid was 245 x 245 mm (0.06 m²). When monitoring began, four grid locations were chosen at each site, so exactly the same nests at the same location could be monitored across seasons. Grids were placed over nests that were easy to access and observe, and in areas unlikely to be disturbed. They were spaced out to reflect the distribution of active nests 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

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

wooden skewers. Grids were aligned and removed by using existing holes through the centre of each ruler, through which four skewers were staked (see Figure 4.3). Each 15 mm skewer was pushed into the ground, to a depth of around 8 mm so they would not be easily dislodged. Skewers were nearly undetectable from just a few meters distance, so florescent nail polish was used to mark the very tips so they could be relocated.

The four grids were set up at the start of each monitoring session. They were hooked onto the skewers that marked the nest locations. 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 approximately centred on, and acquired at right angles to the grid to reduce parallax errors. 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; the grids were photographed sequentially. They 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 (or when) the nest entrances were obscured by debris, the number of active nests were roughly estimated.

Each day after monitoring was completed at the three locations, nest-image collections were immediately transferred from the camera's SD memory onto an external hard-drive. A copy of the monitoring images were also backed to a second external hard-drive. The camera batteries were then recharged and the SD memory was cleared in preparation for the next days monitoring collections. 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 and the image file numbering was reset to zero.

4.2.2 *Control and inactive nest images*

A typical clay bank (located on Mt. Tiger) was cleared from vegetation to simulate active nests. The plastic grid was set-up and sixteen holes were bored into the grid-area of the bank. Each hole was the approximate size of a typical nest entrance. Some holes also included small disturbances in soil to mimic the mounds of dirt typically observed around the active nests of native bees. Photographs of the artificial nest were collected so they could be processed alongside monitoring

images to help confirm the accuracy of automated nest count results. In May 2013 inactive nest images were gathered from each site-grid; these were also processed alongside monitoring images and used to help assess the accuracy of automated nest counts.

- (a) Close-up view of Mt. Tiger, grid 3. There are large pockets in the bank that are filling up with the 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.

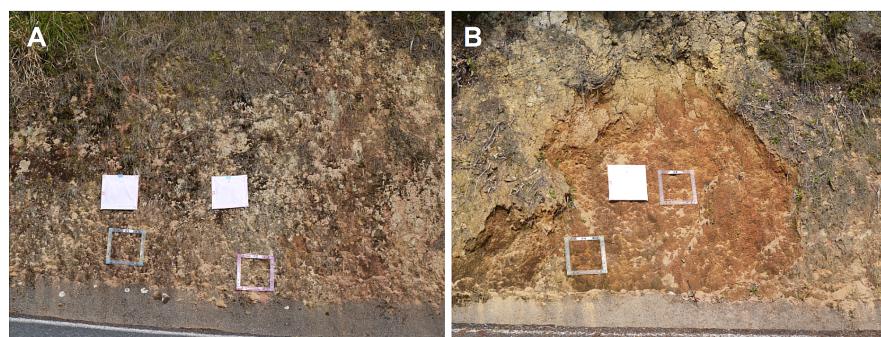


Figure 4.3: Mt. Tiger monitoring images showing (a) the set-up for grid 3 (b) the overall structure of the road-side bank .

(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 clay soil started to accumulate around horizontal ground nest entrances. This could be viably seen from a moving car, especially along (a) Memorial Drive and (b) Mt. Parihaka. The excess reddish clay soils from nest constructions could also bee seen accumulating in the roadside drains along (c) the Mt. Tiger site.

4.3 SUMMARY OF FIELD METHODS

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

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 so they could be processed with monitoring images to help determine the accuracy of automated nest counts.

Manual counting methods have been detailed; only the entrance holes which showed signs of activity could be reliably confirmed as active. 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.

IMAGING METHODS

OUTLINE

5.1 DATA MANAGEMENT

In 2013 monitoring image data were corrupted after hard-drive failures. Images were recovered using open source forensic tools, TestDisk¹ and PhotoRec². When the recovery process was finished the bash scripts shown in Listing 5.1 were run to restore images and a very basic folder structure. A complete folder structure was then created and used for monitoring image data. Folder templates were outlined and they remained consistent over all folders (Figure 5.1). To preserve original data, image operations were performed on duplicate folders (e.g. Folders 071_ sort_ collections and 018_ sorted_ collections). The workflow for each folder (any file or imaging procedures) was documented and saved in readme files. These 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 being used on final folders. The procedures used to restore the monitoring image database are summarised as follows:

1. Invoked TestDisk software to check if the partition table of the old drive could be restored; it was unrecoverable.
2. Invoked PhotoRec to restore raw data files from the failed drive to a new external hard-drive (2T byte). Copied the database onto a separate back-up hard-drive (2T byte) using Grsync³ file synchronization software tool.
3. Restored the folder structure of the raw recovered data by using exiftool⁴ [144] and the bash scripting shown in Listing 5.1 below.
4. Created the final folder structure for monitoring data, shown in Figure 5.1 below.

¹ *TestDisk* <http://www.cgsecurity.org/wiki/TestDisk>

² *TestDisk* <http://www.cgsecurity.org/wiki/PhotoRec>

³ *Grsync* <http://www.opbyte.it/grsync>

⁴ *exiftool* <http://www.sno.phy.queensu.ca/~phil/exiftool>

```

1 #Copy images of a certain type/size into a new folder
#!/bin/bash
recup_dir="${1%/*}" [ -d "$recup_dir" ] ||
echo "Usage: ${0##*/}_recup_dir"; echo "Mirror_files_from_recup_dir_into_"
    recuper_dir_by_ext,_organized_by_extension"; exit 1;
find "$recup_dir" -type f | while read k; do
6 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.
11 $w_dir = '/home/nh/ext_dir';
$r_dir = '/home/nh/photos/';
$jhead_bin = '/usr/bin/jhead';
@rec_dirs = 'ls ${w_dir} | grep recuper_dir';
foreach $recup_dir (@rec_dirs) {print "Scanning_${recup_dir}...";
16 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 $photo_file';
21 print "$jhead_bin -v $photo_file\n";
foreach $line (@exif) {if ($line =~ /Time\ss*\:\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 5.1: Bash script to sort Photorec recovered images by extension; and into date-time based folders.

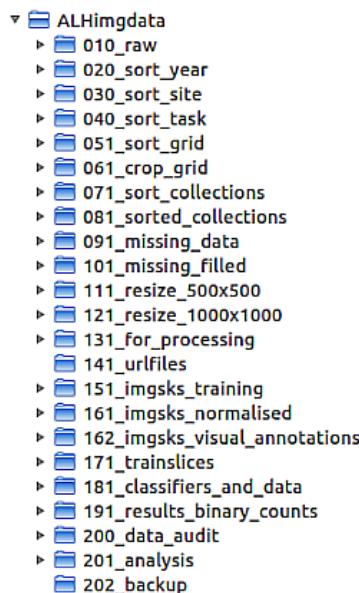


Figure 5.1: Folder structure created for the monitoring image database.

5.2 COMPUTING ENVIRONMENT

A Linux operating system was used. It was configured for the specific image monitoring tasks and to mitigate memory leaks during classifications. Ubuntu (14.04) swap memory was turned off during image processing tasks to prevent processes from swapping out of physical memory. Virtual memory⁵ options were also passed to JavaVM from Fiji's main configuration file to increase the memory heap size. These settings are shown in Listing 5.2 below. Image data were processed on standard computer. There system specifications are listed in Table 5.1 below.

Table 5.1: Computer hardware and operating system specifications.

Operating system	Ubuntu 14.04
Linux kernel	3.13.0-48-generic
PC	CQ1-1240IN Desktop PC
Monitor	46.99cm LCD
Ram	7.4 GiB
Processor	2 x AMD E-350
Graphics card	Gallium 0.4 on AMD PALM
Java(TM)	java.runtime.version 1.6.0_24-b07
Weka	version 3.7.11
Fiji/ImageJ	1.49u

```

1 #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
-----
6 #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
11 # Improve cache management
vm.vfs_cache_pressure=50
-----
```

Listing 5.2: 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.

⁵ refer to: http://docs.oracle.com/cd/E13222_01/wls/docs81/perform/JVMTuning.html

5.3 IMAGE PREPARATION

More than one category of images were collected during monitoring. 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 standard of measure are used in this analysis. These images were sorted by year, site, and grid number (1–4). The preparation of image data for active nests are listed as follows:

- Active nest images were sorted to year, site, grid-folders (Figure 5.2 (a)).
- Images were renamed using Xnview (Figure 5.2 (b)).
- Image exif metadata was combined with the sub-folder grid identifier to create unique file name identifiers.
- Nest images were cropped to a square grid using Xnview (Figure 5.2 (c)).

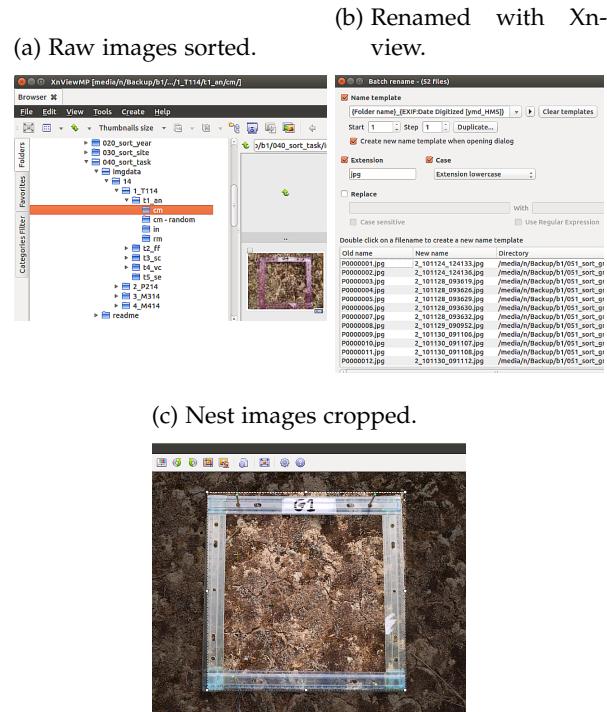


Figure 5.2: Preparation of monitoring data. Nest images (a) acquired using a constructed standard of measure (cm) are used in this analysis. Active nest images were (b) sorted to grids and renamed and (c) cropped to the outside grid using XnView software.

5.3.1 Sorting image collections

The images of active nests were manually cropped to the outside grid edge using XnView as shown in Figure 5.2 (c). Once cropped, images were sorted into separate *collections*. Collections were image sequences of the same nest, separated by minutes-seconds. Collections were sorted into four folders; C₁–C₃ were used in analysis and C₄ contained all/or any other extra images. The following procedure was used to automatically sort images into collections:

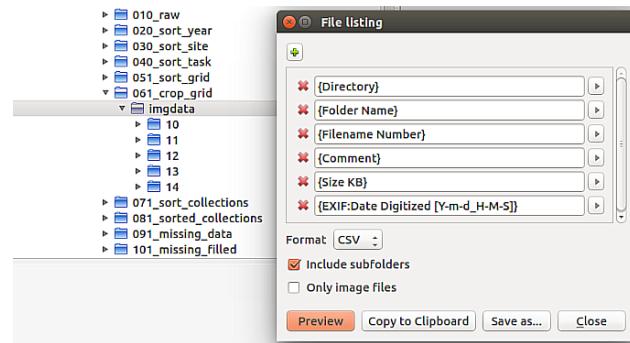
1. Images were cropped (Figure 5.2 (c)) and saved to Folder /061_crop_grid before sorting.
2. XnView catalogue feature *Create > File listing* was used to create a list.csv file (Figure 5.3 (a)).
 - a) The list.csv file was used to organise yearly data.
 - b) Exif metadata data-time information were displayed for each image.
3. The list.csv file was imported into Apache OpenOffice Calc⁶ (Figure 5.3 (b)).
 - a) Date-time attributes of each image were used to sort data into four separate collections.
 - b) The Calc spreadsheet (Figure 5.3 (c) above) was used to write a bash script to:
 - i. Copy images from the source directory.
 - ii. Save images to the predefined destination directory with subfolders: C₁, C₂, C₃ and C₄.
 - a) The exif information for each images was retained during the copying process with exiftool, as shown in Listing snippet 5.3 below.

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

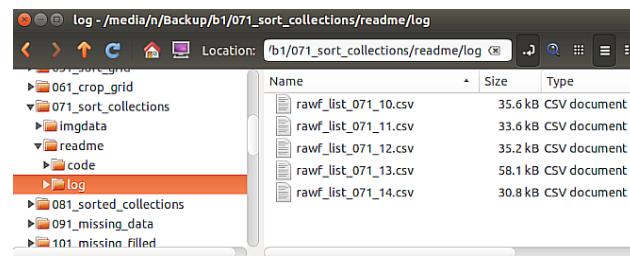
Listing 5.3: Bash snippet to sort collections.

⁶ <https://www.openoffice.org/>

(a) XnView file list.



(b) CSV file listings.



(c) Sorting image collections.

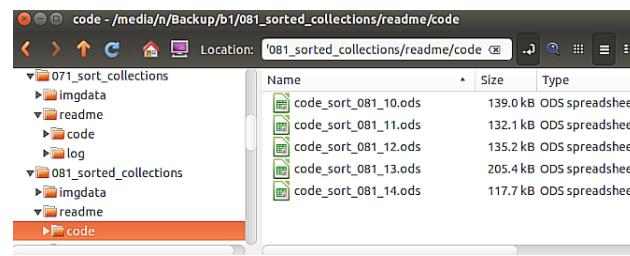


Figure 5.3: Organising data by (a) XnView file listings, (b) CSV file lists and (c) Open Office Cal spreadsheets used to sort image collections.

4. Once the process was completed the file numbers were reviewed via XnView file listings to ensure the correct number of files had been copied.
5. Note: This sorting process was repeated to select training images for training stacks (Figure 5.4 (a)).
 - a) The list.csv was imported into Apache OpenOffice Calc and used to select the third or fourth monitoring day for each year, site and grid.
 - b) The final list was written into a bash script.
 - c) The script automatically copied images from the processing folder (131_for_processing) to the training stacks folder (151_imgsks_training) as shown in Figure 5.4 (b).
6. Filler data (ND) were created monitoring images as shown in Figure 5.4 (c).
 - a) A white image was labelled with ND and copied.
 - b) The exif date-time, and name of each filler image was changed to match the monitoring collections.
 - c) Filler images were selected, copied and pasted into folder (101_missing_filled) using Ubuntu merge.
 - d) Existing images were not overwritten and only images representing *missing* files were copied.

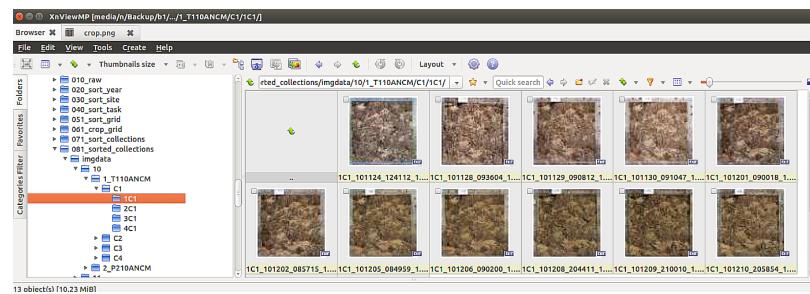
5.3.2 Pre-processing procedures

Pre-processing procedures are summarised briefly below. Classification methods are outlined generally but techniques are examined in more detail in the proceeding Sections.

1. Images were cropped to the outside grid.
2. They were resized to 500 x 500 pixels.
 - a) Bicubic interpolation was selected for resizing.
 - b) It produces smoother images with fewer interpolation artefacts.
 - c) Chosen over other methods since speed was not an issue using bicubic interpolation⁷.
3. Monitoring images were collated into time based stacks.
 - a) Stacks were saved in .tiff format.

⁷ Bicubic interpolation was chosen over other methods since speed was not an issue. Bicubic interpolation produces smoother images with fewer interpolation artefacts

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



(b) Selecting training slices for training stacks.

	D	E	F	G	H	I	K	M	N	O	P	Y	Z	AA	AB
	dir	file	size	type	ext	path	file	ext	path	file	ext	path	file	ext	path
1800	11_PIC2	14	3	1	6		11	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131114.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131114.jpg
1827	11_PIC2	14	3	1	6		12	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131115.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131115.jpg
1828	11_PIC2	14	3	1	6		13	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131116.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131116.jpg
1829	11_PIC2	14	3	1	6		14	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131117.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131117.jpg
1830	11_PIC2	14	3	1	6		15	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131118.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131118.jpg
1831	11_PIC2	14	3	1	6		16	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131119.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131119.jpg
1832	11_PIC2	14	3	1	6		17	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131120.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131120.jpg
1833	11_PIC2	14	3	1	6		18	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131121.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131121.jpg
1834	11_PIC2	14	3	1	6		19	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131122.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131122.jpg
1835	11_PIC2	14	3	1	6		20	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131123.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131123.jpg
1836	11_PIC2	14	3	1	6		21	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131124.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131124.jpg
1837	11_PIC2	14	3	1	6		22	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131125.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131125.jpg
1838	11_PIC2	14	3	1	6		23	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131126.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131126.jpg
1839	11_PIC2	14	3	1	6		24	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131127.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131127.jpg
1840	11_PIC2	14	3	1	6		25	MS	3.mpp				home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131128.jpg		home\VAL\Hengfan171_transfected\transfected\MS\MS13ANCMC2\2\2\1_131128.jpg

(c) Filler image files added to monitoring data.

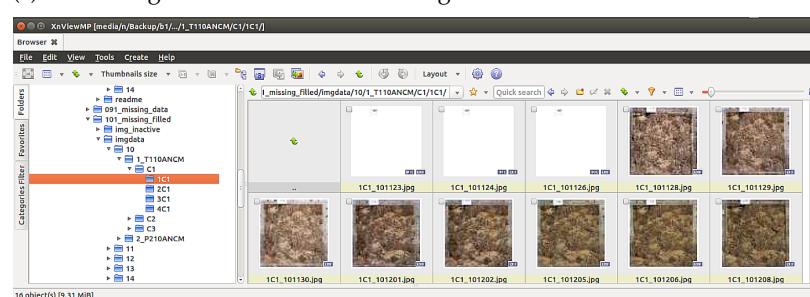


Figure 5.4: File methods for (a) sorting image collections and (b) selecting training slices (c) filling missing image fills with ND data.

- b) Example script snippet in Listing 5.4 below:

```
run("Image_Sequence...", "open=[t110_1c1]convert_to_rgb_sort");
saveAs("TIFF", st110_1c1+"st110_1c1");
```

Listing 5.4: Collate images into stacks

4. Images were enhanced using *contrast enhancement* to improve visual appearance.

- a) Saturated pixels were set to 0.4%.

- b) Each slice was enhanced.

- c) Example script snippet in Listing 5.5 below:

```
run("Enhance_Contrast...", "saturated=0.4_process_all");
saveAs("Tiff", "/161\imgsks\Normalised/imgsks/s10/1\t1\
st110_1c1.tif");
```

Listing 5.5: Enhance image contrast

5.3.3 Overview of file and folder procedures

The following paragraphs provide a general overview of practical procedures used throughout the imaging pipeline. Each important processing *task will be discussed in greater detail in Sections 5.6 and 7.7.2. Several common files types are discussed in this Chapter so are briefly listed first for clarity:

- IJM: ImageJ Macro—a scripting language to automate processing.
- ARFF: Attribute-Relation File Format—an ASCII text file that describes a list of instances sharing a set of attributes; used in TWS and Weka.
- MODEL: Model—Weka classifier model file; once trained and constructed the model can be saved and reloaded for application; used in TWS and Weka.
- R: R file—script written in R for statistical analysis.
- CSV: Comma Separated Value—for statistical analysis in R-studio.
- ODS: Open Document Spreadsheet—for raw data entry in Apache Open Office.
- TXT: Text—plain text output log from TWS and Weka.

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

FOLDERS 071_SORT_COLLECTIONS and Folder 081_sorted_collections: Image data were sorted into separated image collections. These processes are explained further in Section [5.3.1](#)

FOLDER 101_MISSING_FILLED: Filler image data were added to folders to retain the sequential order of image collection monitoring days. This was necessary since some image data were not recovered from the hard-drive failure.

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.

***FOLDER131_FOR_PROCESSING:** Image data for pre-processing were stored in this folder.

FOLDER141_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. All macro (.ijm) and images 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 to this folder including any image overlays.

***FOLDER 171_TRAINSLICES** All training stacks were stored in this folder, including the *control image* and the *inactive nest* images for respective grids. These processes are explained further in Section [7.7.1](#).

***FOLDER 181_CLASSIFIERS_DATA** All training stacks and output files were stored in this folder. This included final annotation data (.arff), classifier models (.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 (1,2,3 and 4) within each of the site folders M3, P2 and T1 as shown below. These processes are explained further in Section [5.6](#).

1. log_final: When respective segmentations were satisfactory then final classifiers were applied to all image stacks and a log file (.txt) of final classifications is saved for records.

2. results: Binary image results were stored in this folder.
3. stacks: Image stacks for processing were stored in this folder.
4. data: Classifiers (.model), data files (.arff), training logs (.txt) and training binary image results (.tiff) were stored in this folder. The final training macro file for each grid was also stored in this folder. The macros used in the first training runs were stored in separate respective subfolders for reference.

***FOLDER 191_RESULTS_BINARY_COUNTS** Post-processing image data (.tif), 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.

5.4 TRAINABLE SEGMENTATION INTERFACE

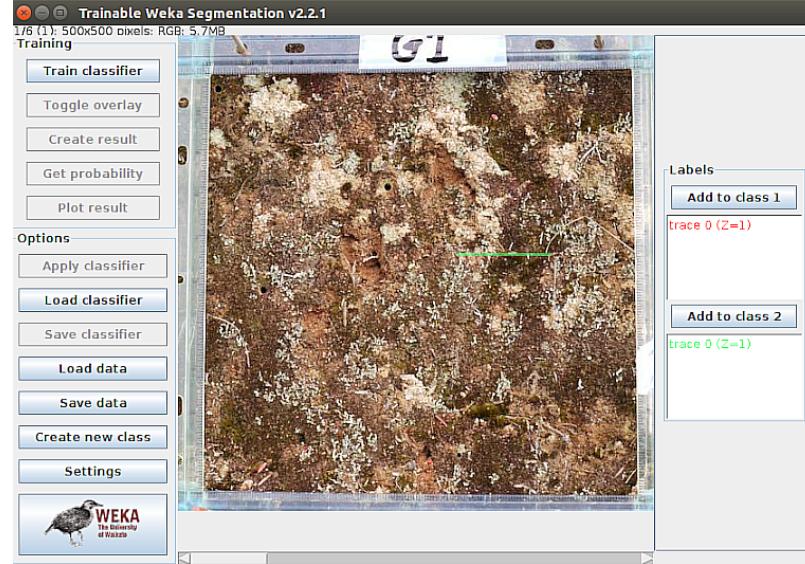
Trainable Weka Segmentation user interface and options are shown in the Figures 5.5, 5.6 and 5.7 below. They are intemperately referred to in later sections regarding classifier testing and training methods, and the models created for nest images. The main user interface and script editor are shown in 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

5.4.1 Filter descriptions and setting options

The filter listed below are used in classifier tuning tests. They were selected from a possible twenty, 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...2n pixels from the target pixel are subjected to the mean, variance, median, minimum and maximum operation; the target pixel is then set to that value.
2. *Structure filter.* For all elements in the input image this filter calculates the eigenvalues (smallest and largest) of the structure tensor 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 [145]. It uses a smoothing scale with $\sigma = 1, 2, 4 \dots 2n$ 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.



(b) Fiji script editor: with the classifier test macro (ijm) loaded.

```

33 makeRectangle(338, 337, 30, 2);
34 call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "4");
35 makeRectangle(299, 279, 79, 2);
36 call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "3");
37 makeRectangle(290, 228, 99, 1);
38 call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "2");
39 makeRectangle(299, 229, 99, 3);
40 call("trainableSegmentation.Weka_Segmentation.addTrace", "1", "1");
41 //select RF training parameters and filters for features stack
42 //optimize for speed and next images
43 call("trainableSegmentation.Weka_Segmentation.setFeature", "Hessian=false");
44 call("trainableSegmentation.Weka_Segmentation.setFeature", "Sobel_filter=false");
45 call("trainableSegmentation.Weka_Segmentation.setFeature", "Difference_of_gaussians=false");
46 call("trainableSegmentation.Weka_Segmentation.setFeature", "Membrane_projections=false");
47 call("trainableSegmentation.Weka_Segmentation.setFeature", "Mean=true");
48 call("trainableSegmentation.Weka_Segmentation.setFeature", "Min=true");
49 call("trainableSegmentation.Weka_Segmentation.setFeature", "Median=true");
50 call("trainableSegmentation.Weka_Segmentation.setFeature", "Structure=true");
51 call("trainableSegmentation.Weka_Segmentation.setMaximumSigma", "4.0");//vary sigma
52 call("trainableSegmentation.Weka_Segmentation.setMembranePatchSize", "1");
53 //train classifier and save data
54 call("trainableSegmentation.Weka_Segmentation.setClassifier", "r.rib.fastRandomForest.FastRandomForest", "-I 50 -K 2 -");
55 call("trainableSegmentation.Weka_Segmentation.trainClassifier");//train classifier
56 call("trainableSegmentation.Weka_Segmentation.saveClassifier", log + "c4-treefinal.model");//save classifier
57 call("trainableSegmentation.Weka_Segmentation.saveData", log + "c4-treefinal.arff");//save data
58 call("trainableSegmentation.Weka_Segmentation getResult");
59 saveAs("Tiff", stack + "c4-treefinal.tif");
60 selectWindow("Log");saveAs("Text", logf +"c4-treefinal.txt");//save log
61 run("Close All");

```

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

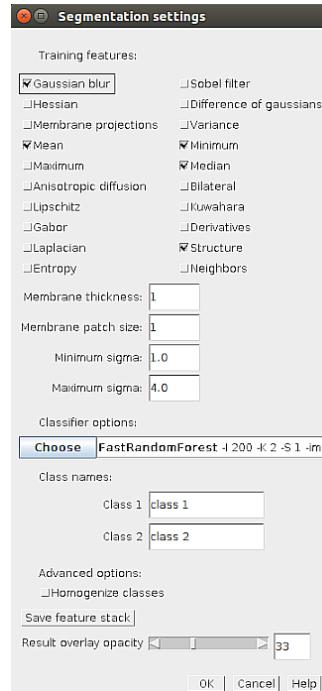


Figure 5.6: TWS features settings interface.

(a) Model options.
(b) Model information

weka.gui.GenericObjectEditor
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

More Capabilities

Open... Save... OK Cancel

Information

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).

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

3. *Difference of gaussians.* Operator performs n individual convolutions with Gaussian kernels with $\sigma = 1, 2, 4...2n$. The larger the radius the more blurred the image becomes until the pixels are homogeneous. By default $n = 4$, therefore = 1, 2, 4, 8 and 16.
4. Membrane projections. Sobel filter calculates the intensity values in a 3×3 region around each image point to approximate the corresponding image gradient. In TWS a Gaussian blur operator with $\sigma = 1, 2, 4...2n$ is performed prior to the Sobel operator.
5. *Anisotropic diffusion.* Filtering with 20 iterations 1, 2, 4, ... $2n$ smoothing per iterations and an edge threshold set to the membrane size [146].
6. *Bilateral filter.* Preserves edges while averaging other parts of the image and is similar to the Mean filter [147]. 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.
7. *Lipschitz filter.* A Lipschitz cover of an image is equivalent to a greyscale 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) [148].
8. *Kuwahara filter.* 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.
9. There are four filter options used in the trainable segmentation set-up and tests. These are briefly described. Only the values of sigma were adjusted for nest segmentations and classifier testing. The other settings mentioned relate to biological image analysis and were not used for nest image processing.
 - Membrane thickness is the expected value of the membrane thickness 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 $N \times N$ 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 (e.g. brain image scans).

- Minimum sigma the minimum radius of the filters used to create the features. By default is 1 pixel. For classifier optimisation testing and nest image analysis $\sigma_{\min} = 1$
- Maximum sigma the maximum radius of the filters used to create the features. By default is 16 pixels. For classifier optimisation testing and nest image analysis $\sigma_{\max} = 2 - 16$.

5.5 CLASSIFIER TUNING

Six images of active nests were selected for classifier optimisation tests. Nest images (Figure 5.8) were chosen to represent the full variation of monitoring data. A macro script was written to automate the testing; this is shown in Figure 5.5 (b). User-traces were saved as rep_nest.arff, so the same feature vectors could be applied to all classifier models during evaluations (see Figure 5.5 (a)). Small repetitive tests were applied to a 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 and are detailed in the proceeding Sections.



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

5.5.1 Testing feature importances

To evaluate the contribution of each filter computeimportances was selected in the TWS segmentation settings dialogue as shown in Figure 5.7 (a). The depth of tree growth maxdepth was not checked and the model was permitted to grow to the maximum depth required during training. 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 $T_n.csv$. Results were imported into an Apache Open Office spreadsheet 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 and manual nest counts taken in the field. This process was repeated until images from each

Table 5.2: Test classifier models (T₁–T₅). The classifier model identifier (C₁–C₅) and the number of features in the stack created from the selected filters (F_n). The number of trees (N) used in tests and the number of random features used (M). The maximum value of sigma (σ_{\max}) selected for tests.

Test	ID	F _n	N	M	σ_{\max}	Comment
T ₁	C ₁	79	200	2	16	Results from TWS tests when default RF settings are used.
T ₂	C ₂	46	200	2	4	Results from TWS tests when filters are optimised for best segmentation of nest images
T ₃	C ₃	20	200	2	2	Results from TWS tests when filters removed to reduce processing time.
T ₄	C ₄	20	10-1000	2	2	Results from Weka tests used to optimise number of trees.
T ₅	C ₅	20	50	0-20	2	Results from Weka tests used to optimise number of random features.

representative slice were sufficiently segmented 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

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 all important and contributed towards final classifications. Based on these tests the ideal number of random features was 5. 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 and 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 $\sigma_{max} = 2$) and 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 analysis.

In test 5, twenty RF models were added to Algorithms for testing and 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 random-feature.arff and analysed in Weka. The out of bag error and overall time to complete processing was evaluated via TWS and the output performance results were saved as tws_random_feature.csv for analysis. The procedures used in random forest parameter optimisation experiments are summarised as follows:

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*

The test classifier CF was compared against other well known machine learners in a final analysis. 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: 7) classifier CF ($N = 50$, $M = 2$) and 6) 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 and 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 5.8 below, 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 for verification, and against the manual field counts.

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 (sub-folder _imagkn) were stored in separate folders for processing. The folder structure is outlined in Figure 5.9 and provides an overview of the procedures used for training classifiers.

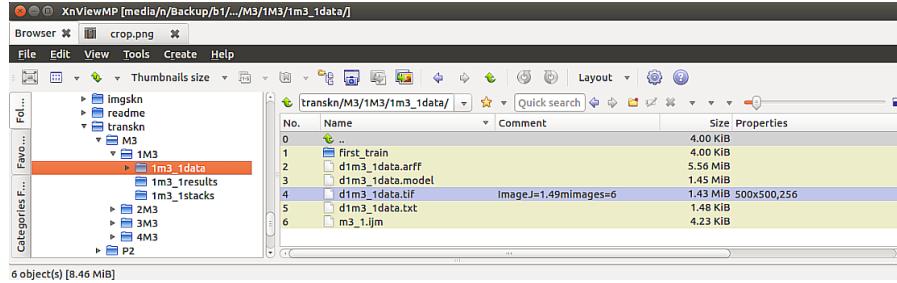


Figure 5.9: Classifier training and data folders

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. For each site, the first grid was used to train initial classifiers. 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_n.arff results were saved. The grid_n.arff files were loaded for the second grid classifier and 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 procedure is summarised as follows:

1. General training set-up procedures:
 - a) Fiji Record Macro was initiated to record all trace samples during training.
 - b) Record Macro files from each training session were saved for records as .ijm files and stored in site classification data folders (i.e. Folder 181_classifiers_data/ transkn/ M3/ 1M3/).
 - c) Only those nests that could be clearly identified as active were used for class_1 traces.
 - d) Only those areas that were clearly backgrounds were used for class_2 traces.

- e) No attempt was made to try and equalise the class-data by providing the classifier with same number of traces for each class.
 - f) Each of the three site-classifiers were trained using exactly the same general procedure outlined below.
2. Grid-training procedure:
- a) Pixels were selected from the control and inactive images slices and these traces were automated with a Fiji macro script.
 - b) The script was applied to train an initial classifier.
 - i. No other traces were added; the initial base classifier was trained using only the automated annotations.
 - c) After the first training-run the segmentation results were checked.
 - i. A *very small* number of inaccurate segmentations, on a single image slice, were assigned to the correct classes with new traces.
 - d) The classifier was retrained and segmentation results were checked.
 - e) If segmentations were still not accurate then a very small number of traces were reassigned, on a single slice and the classifier was retrained.
 - f) If segmentation results deteriorated then the traces added during the previous training-run were removed.
 - i. Another trace was added, at different location, and the classifier was retrained.
 - g) This process was repeated sequentially until all slices were either annotated and/or properly segmented.
3. Site-training sequence:
- a) Two training runs were used.
 - b) First-train data were saved into grid-data subfolders as shown in Figure 5.9 above.
 - c) Start G2 first-training run:
 - i. Load Gf₁.arff and train.
 - ii. Check segmentations and follow procedures 2 above to correct for any inconsistencies.

- iii. Save Gf₂.arff
 - d) The procedure was repeated until all site-grid classifiers were trained.
 - e) Final annotations were saved as final.arff data.
4. Application of final site-classifier to monitoring data:
- a) Data (final-site.arff) and classifier model (final-site.model) from site-training was loaded into TWS.
 - i. Grid monitoring stacks were selected in sequential order.
 - ii. The classifier was applied as a batch process.
 - iii. Binary results and data were saved to grid folders and subfolders.

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, and against the values recorded for manual field counts. The post-processing method was automated in a script (Listing 5.6) and applied in as a batch process. The procedures used in the post-processing pipeline are detailed below. They include the binary options settings, the morphological operators used and object descriptors used for final binary counts.

1. Binary options and final settings. Refer to Figure 5.10 (a) below.
 - a) Iterations:
 - i. Specifies the number of times operations are performed.
 - b) Count:
 - i. The number of adjacent background pixels necessary before they are removed from the edge of objects during the erosions.
 - ii. The number of adjacent foreground pixels necessary before pixels are added to the edge of objects during dilation⁸.
 - c) Pad edges:
 - i. Setting was checked for all post-processing operations.
 - ii. Affects eroding and closing.
 - iii. Edge erosion was not performed during *Process>Binary>Erode* and *Process>Binary>Close*⁸.

⁸ see <http://imagejdocu.tudor.lu/doku.php?id=gui:process:binary>

- d) EDM output was checked to overwrite the 8-bit input images⁸.
- 2. Post-processing pipeline:
 - a) *Fill Holes*
 - i. Filled holes in binary objects.
 - b) *Close*: 10 iterations 5 counts.
 - i. Dilations⁹ followed by erosions¹⁰.
 - ii. Connected any disconnected parts of the binary images
 - iii. Joined breaks, closed holes and smoothed out contours.
 - c) *Fill Holes*
 - i. Filled any remaining holes.
 - d) *Open*: 2 iterations 3 counts.
 - i. Erosion followed by dilation.
 - ii. Separated connected objects in the images.
 - iii. Smoothed contours removing isolated objects.
- 3. Analyze Particles for three binary schemes. Refer to Figure 5.10 (b) below.
 - a) Analyze Particles options:
 - b) *Circularity* was set between 0.10 – 1.00 for all schemes.
 - c) *Pixel sizes (p²)*:
 - i. p² = 10 – ∞
 - ii. p² = 15 – ∞
 - iii. p² = 20 – ∞
 - d) *Image Overlay* was checked to create a separate overlay image stack showing the outlines of counted objects. Refer to Figure 5.10 (c) below.
- 4. *The final binary counts*:
 - a) The average count over the three binary schemes.
 - b) The median values over three image collections.
- 5. Post-processing macro snippet:

```
//open classified image stacks and post-process
run("Make_Binary", "method=Default_background=Default_calculate");
3 run("Options...", "iterations=1_count=1_edm=Overwrite_do=[Fill_
Holes]_stack");
run("Options...", "iterations=10_count=5_pad_edm=Overwrite_do=Close
_stack");
```

⁹ Dilate: enlarges object borders so holes become smaller.

¹⁰ Erode: shrinks the image so holes became larger and small details are deleted.

```

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...
8 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 5.6: Post-processing snippet.

5.8 CLASSICAL SEGMENTATION METHODS

5.8.1 *Thresholding monitoring images*

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 and 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 randomly selected from image data using the spreadsheet list. One hundred and seventy images were used for manual image counts. They were collated into a single pdf stack. Another stack was created which included counted overlay results; this stack was used for verification (Figure 5.10 (c)). Two observers conducted counts on the raw RGB images. Scorers reviewed which objects could realistically be identified as active nests after a trial run. Final counts were conducted in a complete run with no further discussions. 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. The manual image count procedure is summarised as follows:

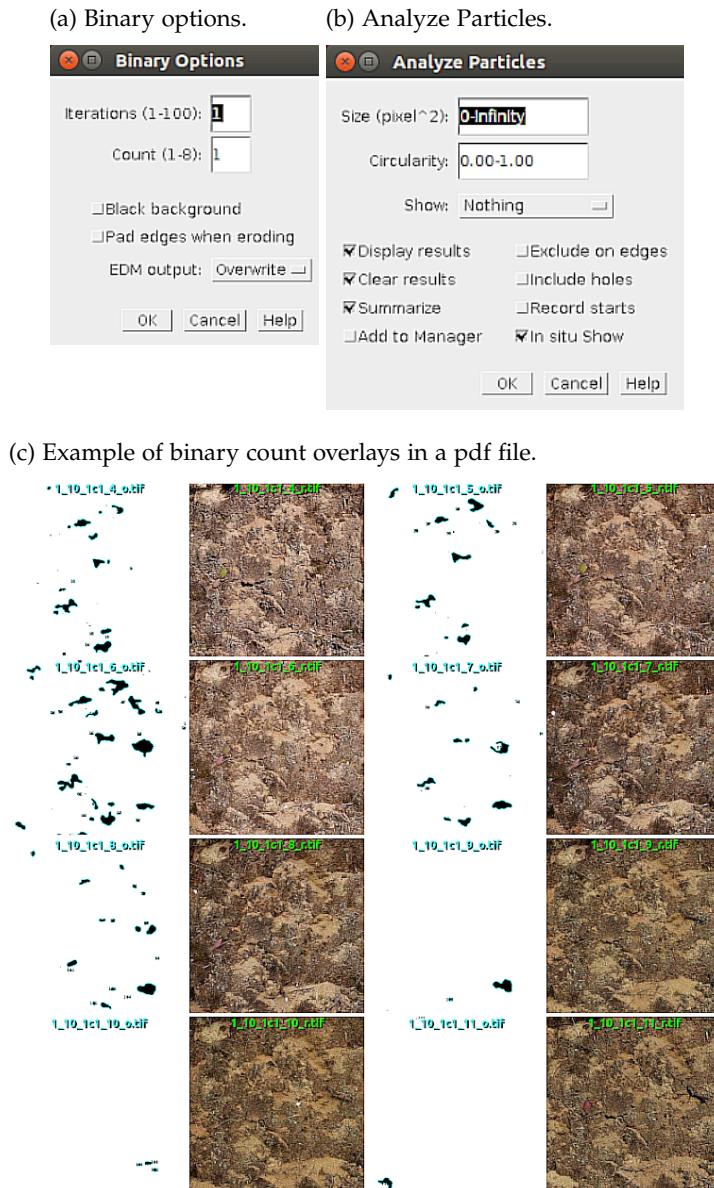


Figure 5.10: Post-processing operators in Fiji. The user interface to select the (a) Binary options and morphological operators applied to segmented images. The settings used to count binary objects in images with the (b) Analyze Particles utility. The binary count overlays (c) which were manually checked against raw RGB images for accuracy.

1. Images were randomly selected but only from paired-monitoring data (i.e ND filler images were not included).
 - a) Raw RGB images were compiled into a single pdf document as shown in Figure 5.10.
 - b) Images were compiled in sequential order and labelled with file reference identifiers as image overlays.
 - c) Trial nests counts from image data:
 - i. A trial nest count from images was made by two observers.
 - ii. Discussions between observers were conducted regarding the images.
 - iii. This was to determine what objects could be reasonably reconsigned as active nests.
 - iv. After reasonable agreement between observers were reached manual nest counts from images were collected.
 - d) Manual nest counts from image data:
 - i. There were no further discussions between observers about nest images when counts were conducted.
 - ii. Counts were made from pdf images in a single complete run.
 - iii. Each scorer conducted three replica counts in turn.
 - iv. While one scorer counted, the other recorded data immediately to a spreadsheet.
2. A corresponding pdf comprising of raw RGB images alongside automatic count overlays were compiled into a corresponding pdf (Figure 5.10 (c)).
3. If required, verification of manual-image counts could be checked alongside raw RBG images and automatic nest count overlays.

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:seconds. Separate collections were therefore comprised of different images; each single one 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:

1. For automatic nest count data.
 - a) Image collections (C₁-C₃).

- i. The mean nest counts from three binary schemes were calculated.
 - ii. Values were rounded up to whole numbers to provide automatic-count subtotals.
 - b) The median counts from three image collections on the were taken on the automatic-count subtotals.
 - c) This provided the final automatic-count data used in comparative method analysis (the CF classifier-*ac* and threshold method-*at*).
2. For manual-image count data.
- a) The mean of three replica counts were taken for each observer and rounded up to whole numbers.
 - i. This provided manual-image count data (*mic_ob1* and *mic_ob2*).
 - ii. Data were used for inter-observational analysis.
 - b) The mean counts from two observers were taken and round to up whole numbers.
 - c) This provided the final manual-image data used for comparative method analysis (*mic_t*).
3. For manual-field count data.
- a) The mean of three replica counts were taken for both observers round up to whole numbers.
 - b) 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 from the images (*mic*).
5. Automated counts from the RF optimised model (ac) and manual counts from the field (*mfc*).

Figure 5.11: The images data sheet shown was one of six used for monitoring data entry: 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 R-Studio. 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 in Figure 5.11. The first column for each sheet contained a sequential base reference identification. All zero count data were input as 0; all *no-data* entries were identified by a text value entered as ND as demonstrated in 5.11 column P–comments.

5.10.2 Statistical tests

Lin's Concordance of Correlation (ρ_c) was used to compare methods. It combines measures of accuracy and precision to test the agreement (or accuracy) between two observations [149]. In the equation $\rho_c = rC_b$ and r is the correlation coefficient and C_b is determined by:

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

Here:

$\nu = \sigma_x / \sigma_y$ where σ is the variance of x and y

μ is the mean value of x and y respectively

$$u = (\mu_x - \mu_y) / \sqrt{\sigma_x \sigma_y}$$

Pearson's correlation coefficient r shows how scattered the data points are around the line of best fit and gives a *measure of precision*. The value of μ defines the scale shift and measures systematic bias in

measured values 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 $\mu = 0$.

The value ρ_c was calculated using the epi.ccc function in the epiR package [?]. The R-studio graphical user interface is shown in Figure 5.12 and the epi.ccc function is demonstrated in Listing 5.7 below. Graphical analyses were saved as results.tiff for records; a notebook was compiled from the R script associated with each analysis and saved as a pdf.

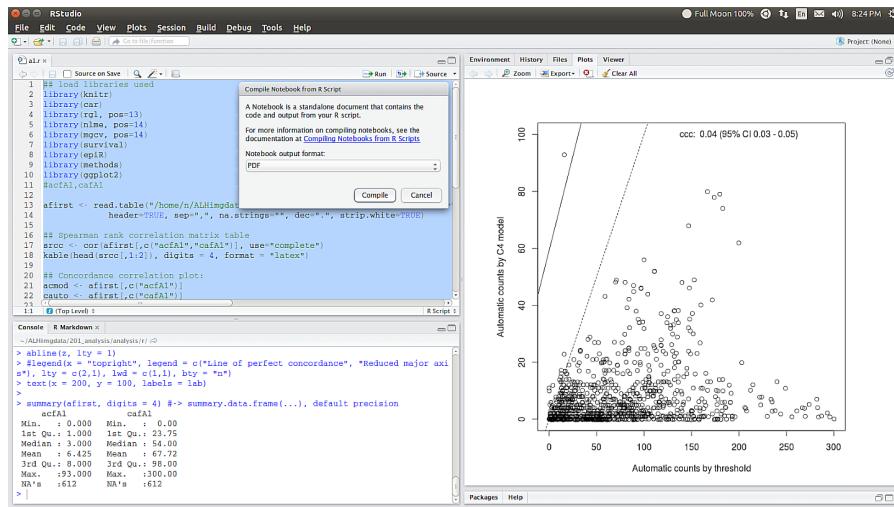


Figure 5.12: R-Studio graphical user interface showing the operation of the script in Listing 5.7 below

5.11 SUMMARY OF IMAGING METHODS

Substantial Chapter. Image data management. Software tools and methods Machine learning Classifier optimisation and tuning Feature engineering Trees and random features Data handling and final preparations Statistical analysis Concentrate on methods - not classifier performance

```

## 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)
5    afirst <- read.table("/300_documents/ch6_imgs/finals/rA1/A1.csv",
                           header=TRUE, sep=",", na.strings="", dec=".",
                           strip.white=TRUE)
## Spearman rank correlation matrix table
srcc <- cor(afirst[,c("acfA1","cafA1")], use="complete")
10   kable(head(srcc[,1:2]), digits = 4, format = "latex")
## Concordance correlation plot:
acmod <- afirst[,c("acfA1")]
cauto <- afirst[,c("cafA1")]
## cc correlation matrix table
15   cccfirst <- epi.ccc(cauto, acmod, ci = "z-transform", conf.level = 0.95)
firstr <- cccfirst$rho.c
kable(head(firstr[,1:3]), digits = 4, format = "latex")
tmpfirst <- epi.ccc(cauto, acmod, ci = "z-transform", conf.level = 0.95)
20   lab <- paste("ccc:", round(tmpfirst$rho.c[,1], digits = 2), "(95%CI",
                  ",",
                  round(tmpfirst$rho.c[,2], digits = 2), ",",
                  round(tmpfirst$rho.c[,3], digits = 2), ")",
                  sep = "")
25   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)
30   text(x = 200, y = 100, labels = lab)

```

Listing 5.7: R-studio concordance correlation script for automatic counts by thresholding and the CF classifier.

Part III
OUTCOMES

6

RESULTS

OUTLINE

6.1 CLASSIFIER TUNING

A summary of classifier tuning tests are shown in Table 6.3 below. Five main tests (T₁–T₅) were performed 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 are compared using the out of bag error (oo_b) and the time taken to construct the features stack (ms).

Test	ID	F	N	M	σ_{max}	$oo_b\%$	ms
T ₁	C ₁	79	200	2	16	1.98	147138
T ₂	C ₂	46	200	2	4	0.93	402974
T ₃	C ₃	20	200	2	2	0.698	23092
T ₄	C ₄	20	50	2	2	1.279	12503
T ₅	C ₅	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: Gaussian 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 σ_{max} = 16.

The TWS default model and features settings produced model with a features stack of 79. The number of pixels selected as class₁ = 44 and class₂ = 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 around 46 (58%) did not provide any additional information. Feature importances (F_i) were $\leq 0\%$ in 58% of the filters. The top twenty most important features for test classifier C₁ are shown in Figure 6.1.

6.1.2 Optimised features

In Test 2 the following filters were selected: Gaussian 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 (48%) did not provide any additional information ($F_i \leq 0\%$). 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.

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 ($F_i \geq 0\%$) as shown in Figure 6.3.

6.1.4 Number of trees

In Test 4, twenty two RF models were loaded into Weka Experimenter Algorithms in a 10 fold cross validation, 10 iteration test. The number of trees were adjusted $N = 10 - 1000$ in each of the 22 RF models (shown along the bottom axis in Figure 6.4). 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 $class_1 = 44$, $class_2 = 816$. The feature stack for six slices with 20 features took 17503 to create; and

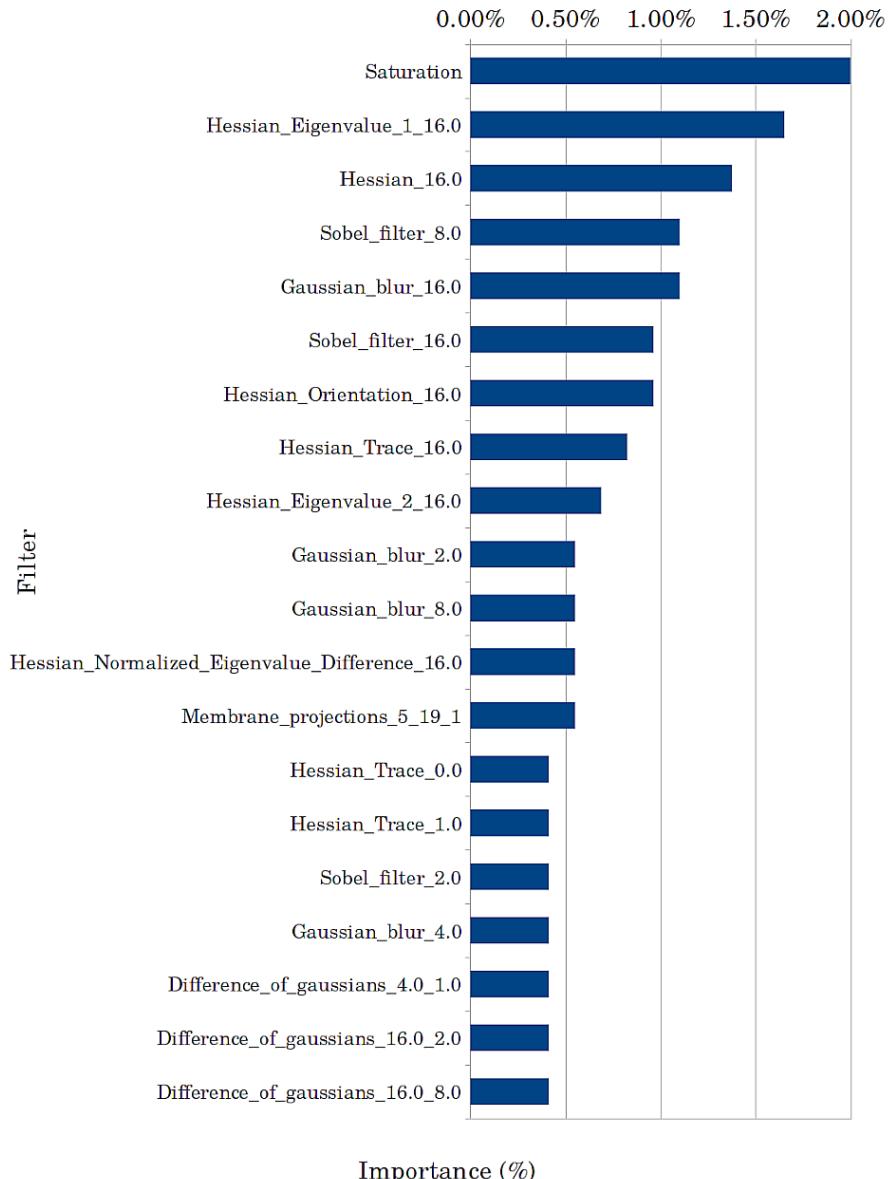


Figure 6.1: Test 1: Feature importances results for classifier C1. This model used default settings. The feature stack for six slices with 79 features took 147138ms to create and the $oo_b = 1.98\%$. The highest ranking filters and respective importances are listed.

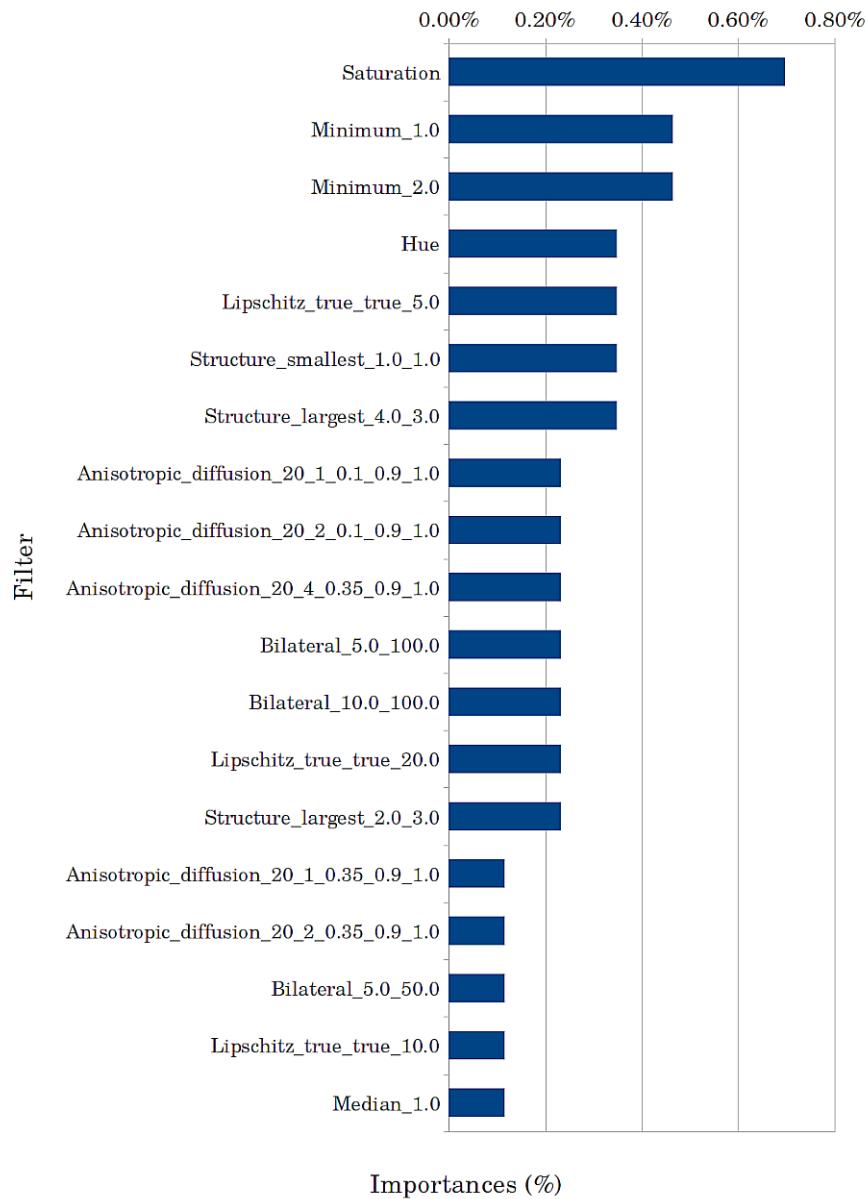


Figure 6.2: Test 2: Feature importance tests on classifier C2 which was optimised to produce the best segmentation of nest images. The feature stack for six slices with 46 features took 402974ms to create and the $oo_b = 0.93\%$. The highest ranking filters and respective importances are listed.

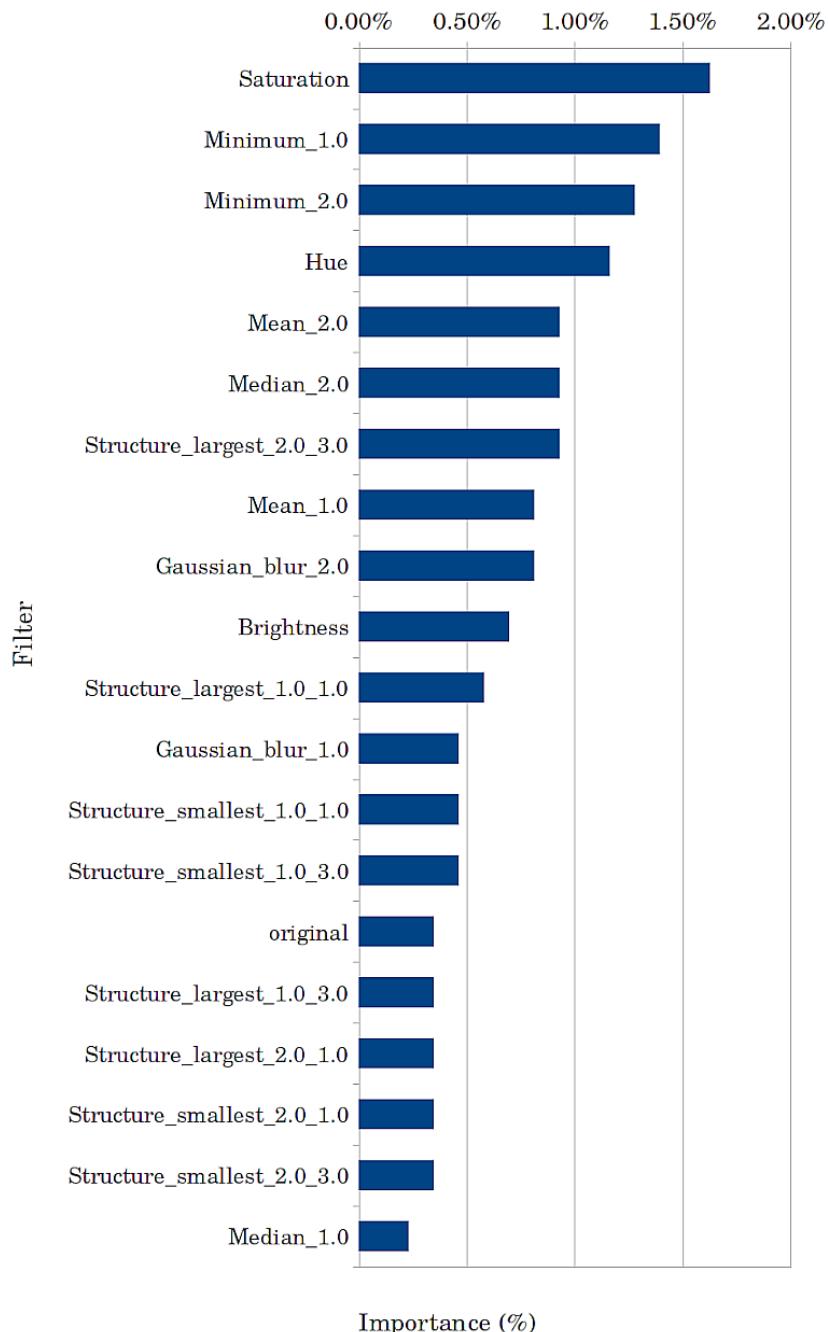


Figure 6.3: Test 3: Feature importance tests on classifier C₃ which was optimised to speed up processing. The feature stack for six slices with 20 features took 23092ms to create and the $oo_b = 0.698\%$. All twenty filters were important and ranked as listed.

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 $oo_b = 1.083\%$ (highlighted in orange on Figure 6.4). Tests were repeated using the rep_nest.arff dataset. An alternative set-up was loaded. 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).

6.1.5 Number of random features

In Test 5, twenty RF models were added to Algorithms for testing using a 10 fold cross validation experiment with 10 iterations. Each classifier model had a varying number of random features which were set between $M = 0 - 20$ (shown along the bottom axis on Figure 6.5).

Classifier C5 was re-run via TWS with $N = 50$, $M = 2$ and $\sigma_{max} = 8$. The TWS settings for classifier C5 produced a features stack of 20. The number of pixels selected as $class_1 = 44$, $class_2 = 816$. 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 $\sigma_{max} = 8$ highlighted in orange on Figure 6.5.

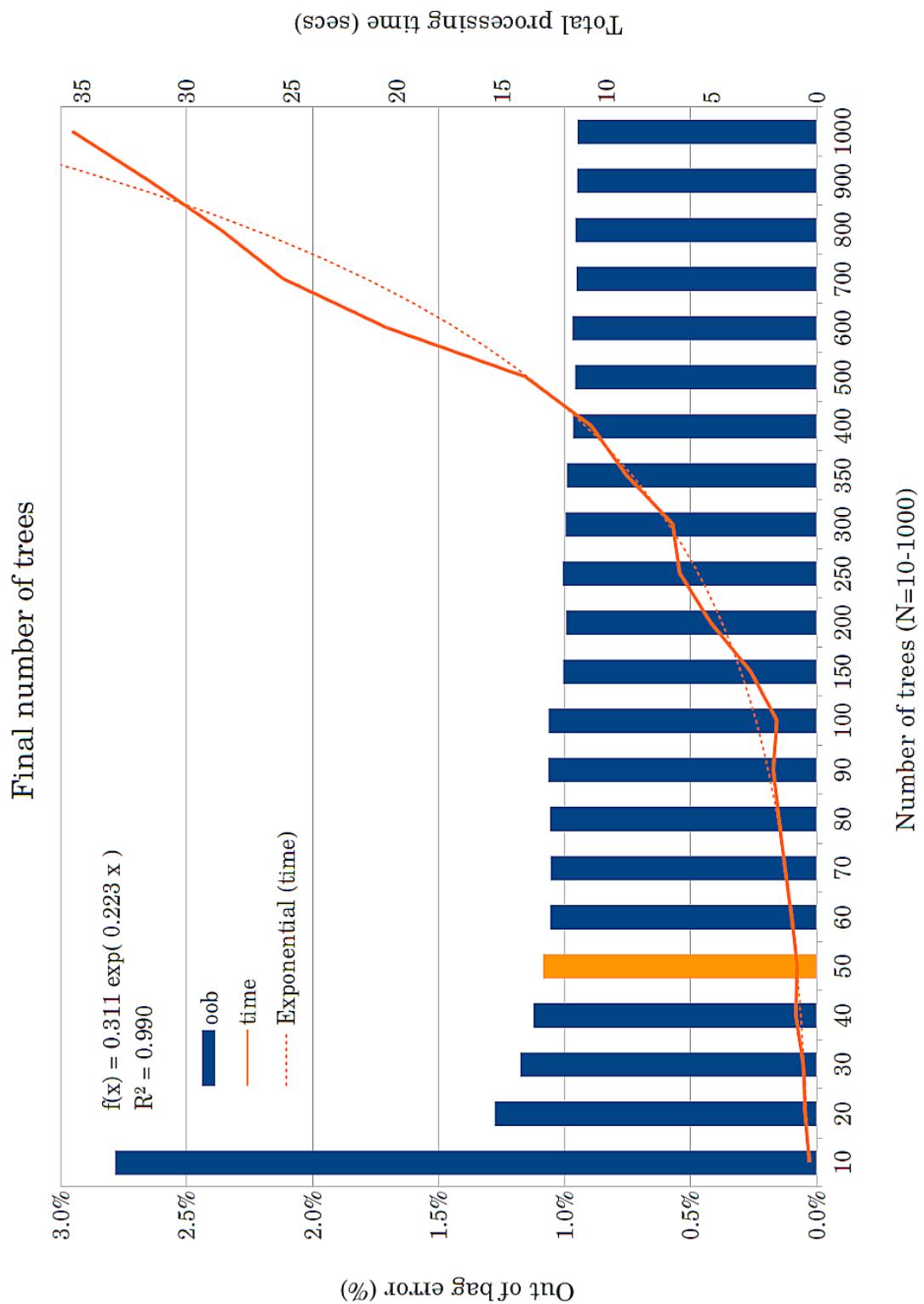


Figure 6.4: Test 4: Fifty trees ($N = 50$) were selected for the random forest model final monitoring classifier (CF). The out of bag error from Weka was $\text{oob} = 1.083\%$ (orange bar) and total processing time $t_{\text{total}} = 0.925(\text{sec})$ using the rep_mest.arff dataset.

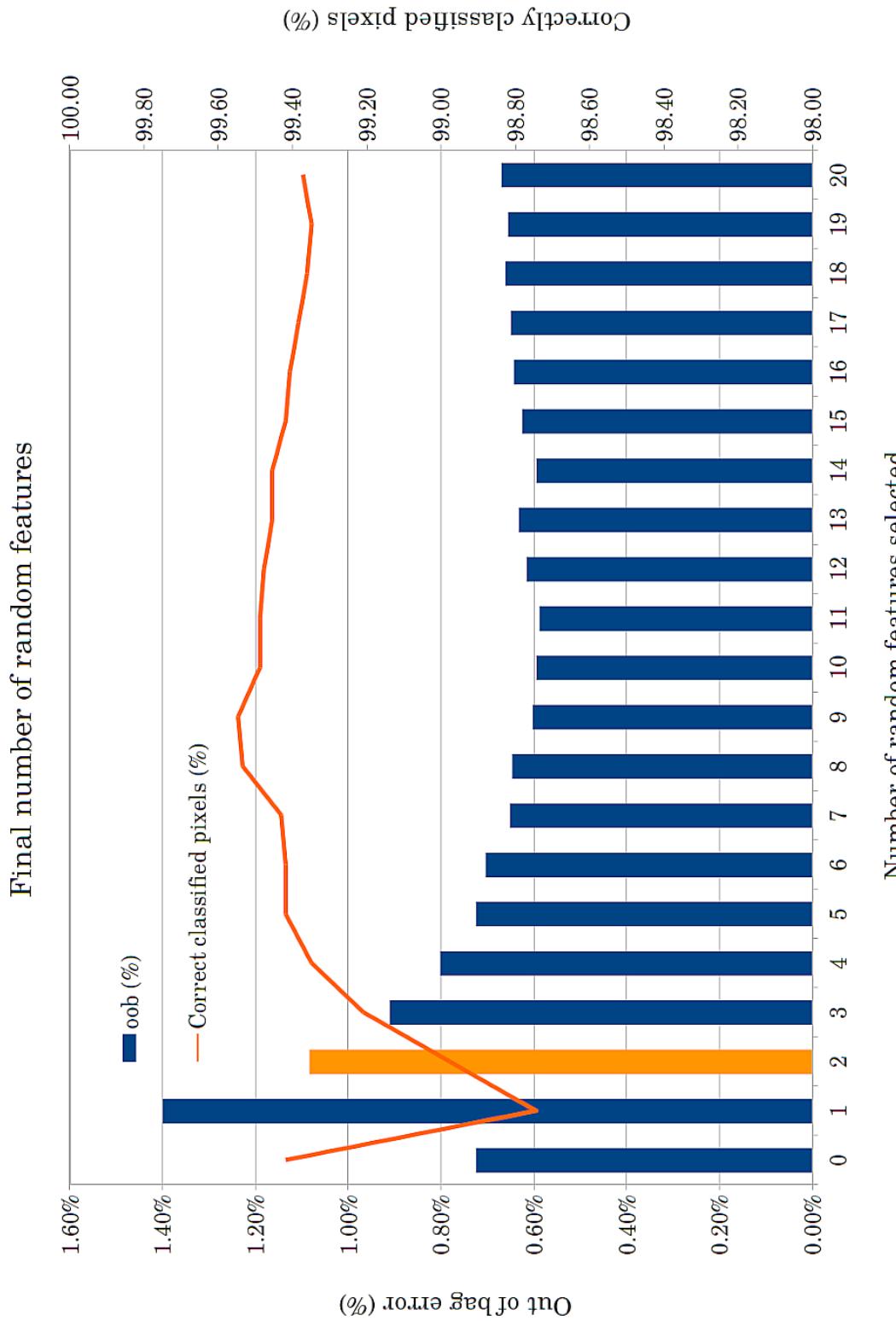


Figure 6.5: Test 5: Two random features ($M = 8$) were selected for the random forest model final monitoring classifier (CF). The out of bag error from Weka was $\text{oob} = 0.628\%$ (orange bar) and the percentage of correctly classified instances (pixels) $cc_i = 99.4\%$ using the rep_nestarff dataset.

6.2 CLASSIFIER BENCHMARKS

The final test classifier CF was compared against nine machine learners. 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, neural network) and SMO (M10, support vector machine) 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).

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.

The out of bag error decreased slightly to $oob = 0.98\%$ when the second trace was included in training (CF₃). The final counts were visually checked against raw RGB images for verification and against the manual field counts as detailed in Table 6.3 below. Classical threshold methods were tested. Intensity, statistical region merging and canny edge detectors were operations were applied to the test stack.

6.4 TRAINING STACKS

The four training stacks collated for each site are displayed in the Figures 6.7, 6.8, and 6.9 below. They demonstrate the wide variation in images between and within monitoring sites. All the trainings 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 (classifica-

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.

Model	Correct (%)	Weka model code
M1	99.42	hr.irb.fastRandomForest.FastRandomForest-I 50 -K 8 -S 1
M2	94.88 •	rules.ZeroR
M3	98.37	trees.J48-C 0.25 -M 2
M4	98.02	trees.RandomTree-K 0 -M 1.0 -V 0.0010 -S 1
M5	99.19	trees.RandomForest-I 10 -K 0 -S 1 -num-slots 1
M6	98.95	hr.irb.fastRandomForest.FastRandomForest-I 50 -K 2 -S 1
M7	99.07	hr.irb.fastRandomForest.FastRandomForest-I 200 -K 2 -S 1
M8	90.35 •	bayes.NaiveBayes
M9	95.71 •	functions.VotedPerceptron-I 1 -E 1.0 -S 1 -M 10000
M10	96.74 •	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

◦, • statistically significant improvement or degradation

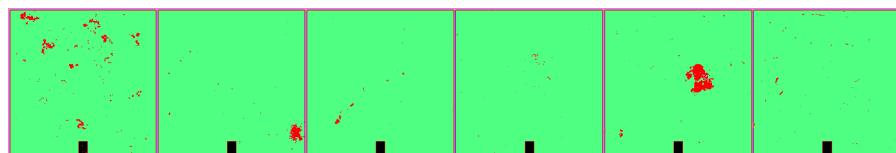
Table 6.3: The final automatic count results on representative images (slices 1–6) using CF_{rt} compared to classical segmentation methods, manual-image and manual-field counts.

Method	s1	s2	s3	s4	s5	s6
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	1	8	4	2	3
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	4	3	2	2	5	5
Srm10	1	2	0	0	2	2
Srm11	1	2	1	1	1	1
Srm12	41	57	10	28	99	14
Srm13	60	72	24	42	114	20
Cedge14	93	62	41	70	95	25
Cedge15	88	92	48	62	39	10
Cedge16	170	202	71	91	133	20
Cedge17	191	230	63	99	149	22
Cedge18	179	225	49	96	169	19
Cedge19	175	225	64	103	143	22
Cedge20	189	226	46	103	179	16

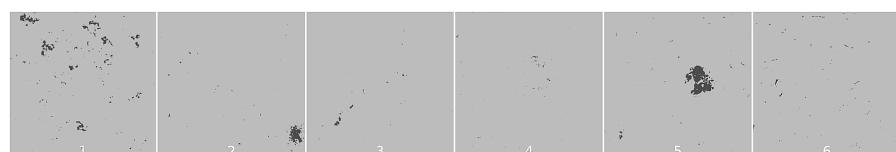
(a) Raw stack images.



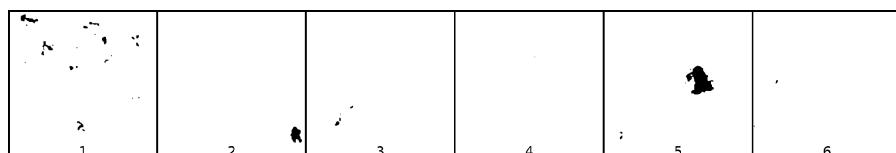
(b) Classified outputs.



(c) Converted to 8-bit binary.



(d) Binary operators applied.



(e) Particle Analysis.

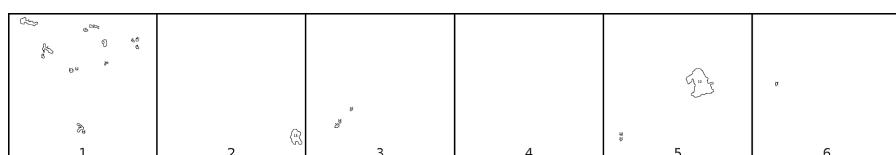


Figure 6.6: Representative images in (a) the test stack used to train initial classifiers (C1–C5). The classified results from CFre-trained (b) were post-processed (c)-(d) to give the (e) final counts. These were compared against classical segmentation methods, visual nest counts from images, and the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

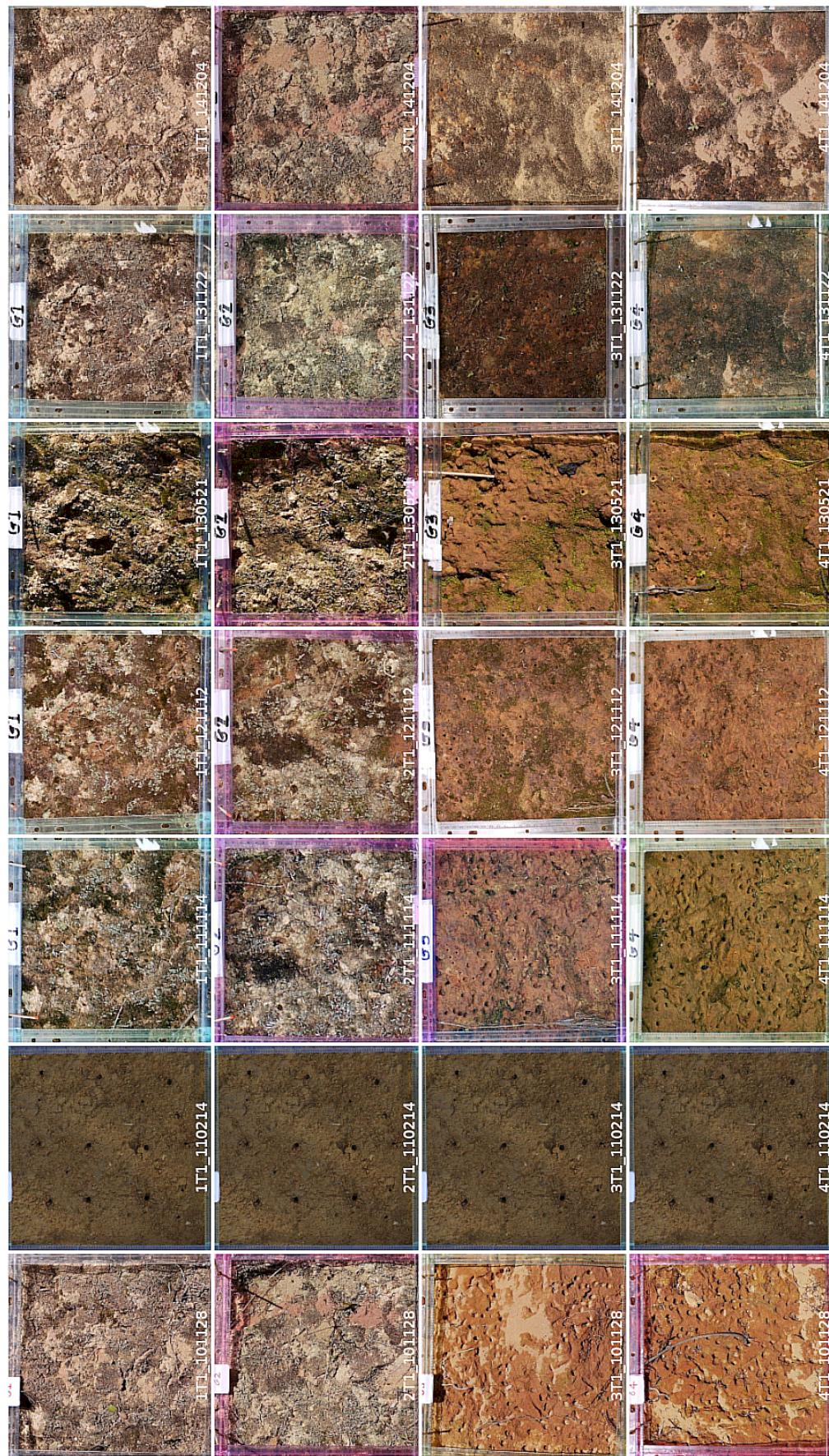


Figure 6.7: Site 1: Training stacks for Mt. Tiger. Slices of collection years from left to right and grid numbers 1–4, from top to bottom. Slice 1 = control and slice 4 = inactive nest. All nests were on a roadside bank with an approximate slope ranging between 60–80°. There were no horizontal ground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

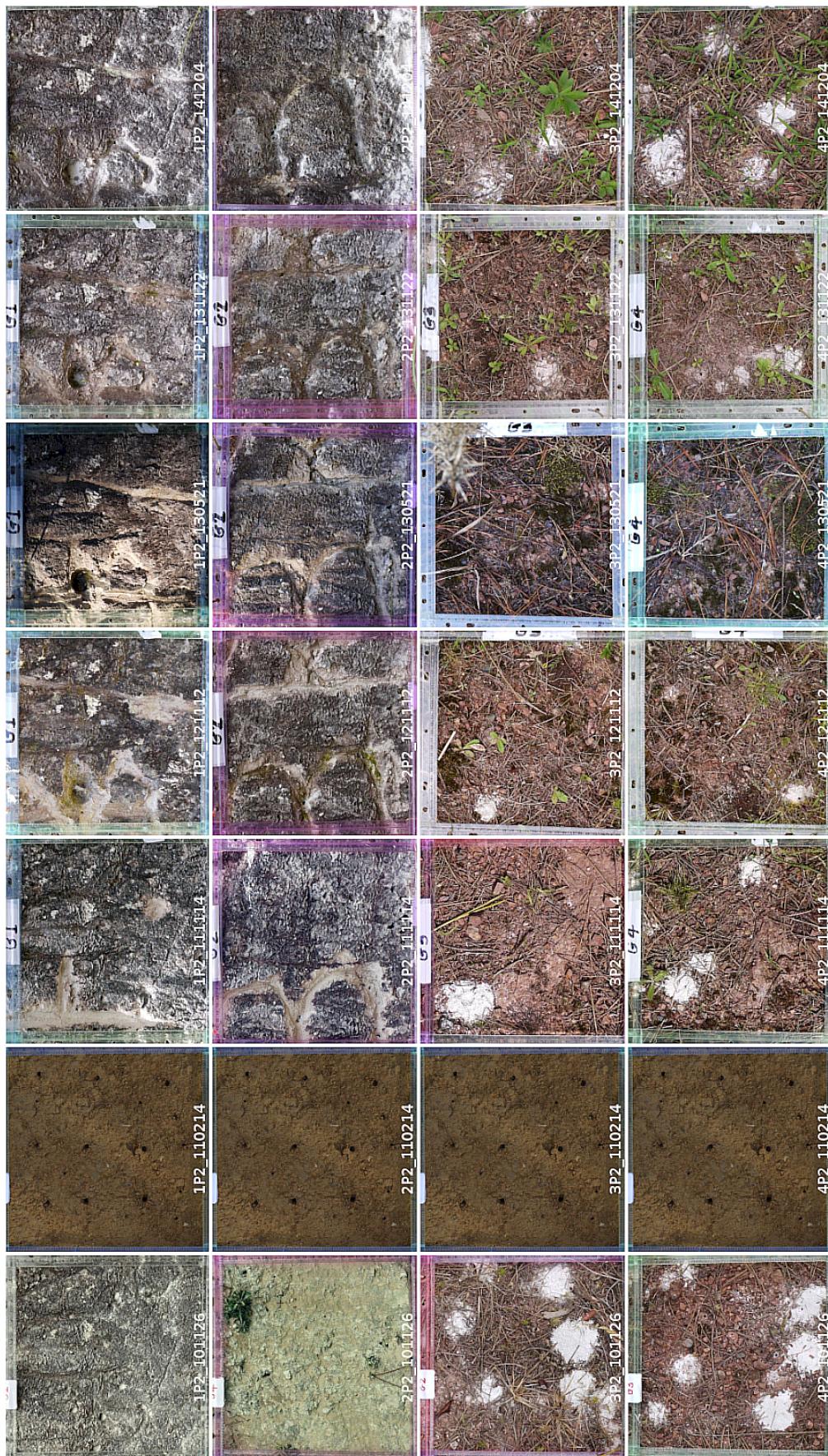


Figure 6.8: Site 2: training stacks for Mt. Parihaka. Slices of collection years from left to right and grid numbers 1–4, from top to bottom. Slice 1 = control and slice 4 = inactive nest. The nests in grids 1–2 were on a large bank (5 m long × 8 m high) with an approximate slope ranging between 60–80°;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

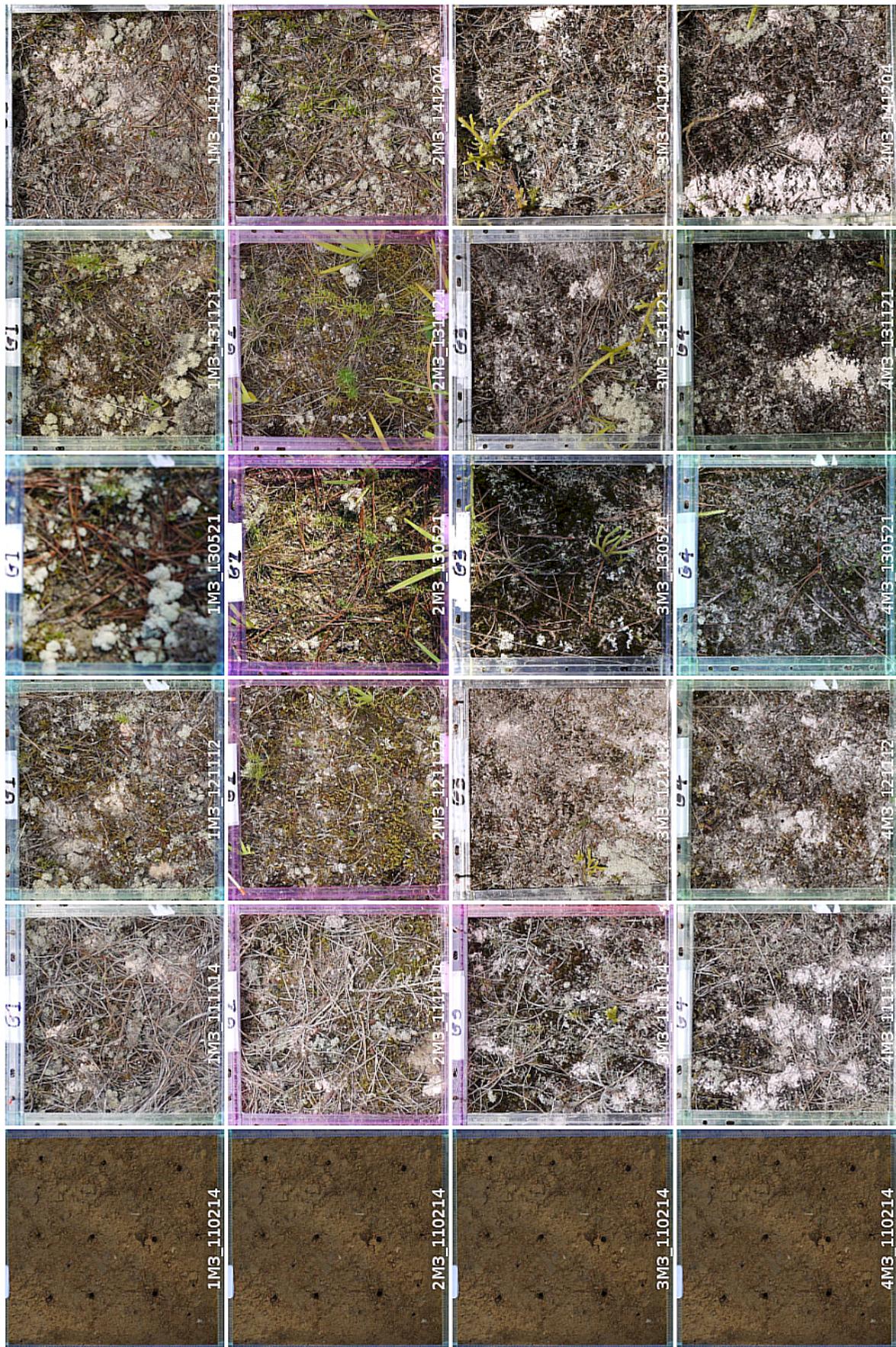


Figure 6.9: Training stack Memorial Drive. Slices of collection years from left to right and grid numbers 1–4, from top to bottom. Slice 2 = control and slice 5 = inactive nest. All nests were on a roadside bank with an approximate slope ranging between 60–80°. There were no horizontal ground nests at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

tions and post-processing) was completed in less than 24 hours. Field monitoring was conducted in a single two and 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 below.

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
S ₁	16	16	13	10	3	59
S ₂	16	16	13	10	3	59
S ₃	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 are shown in Table 6.5 below. The variances were larger than the means for both methods and the dispersal index suggests clustering; data fits a negative binomial model.

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.

Frequency distribution of manual field and automatic counts

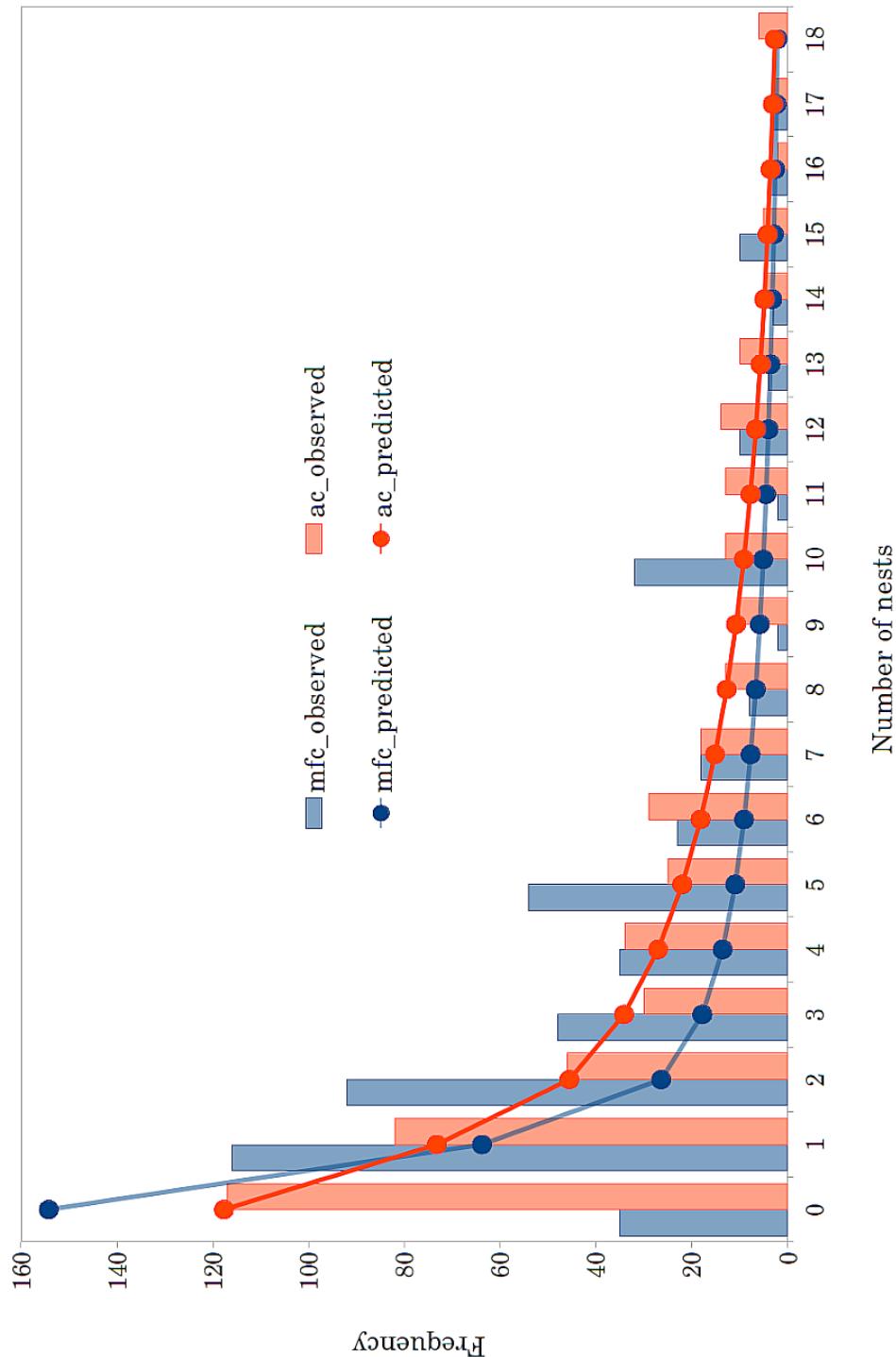


Figure 6.10: Frequency distribution of automatic (ac in red) and manual field (mfc in blue) nest counts. Histograms show the observed frequency distributions for manual field counts and automatic counts. Matching coloured lines, joined with close circles are the predicted frequencies expected from a 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	1896	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 (σ/mean)	DI
Coefficient of variation (SD/mean)	CV

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) which are outlined and summarised in Table 6.6.

Table 6.6: Five comparative analyses and 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) = ρ_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. The methods show poor agreement ($\rho_c = 0.040$) and accuracy ($C_b = 0.164$); with a very weak positive correlation ($r = 0.214$, $P < 0.05$).

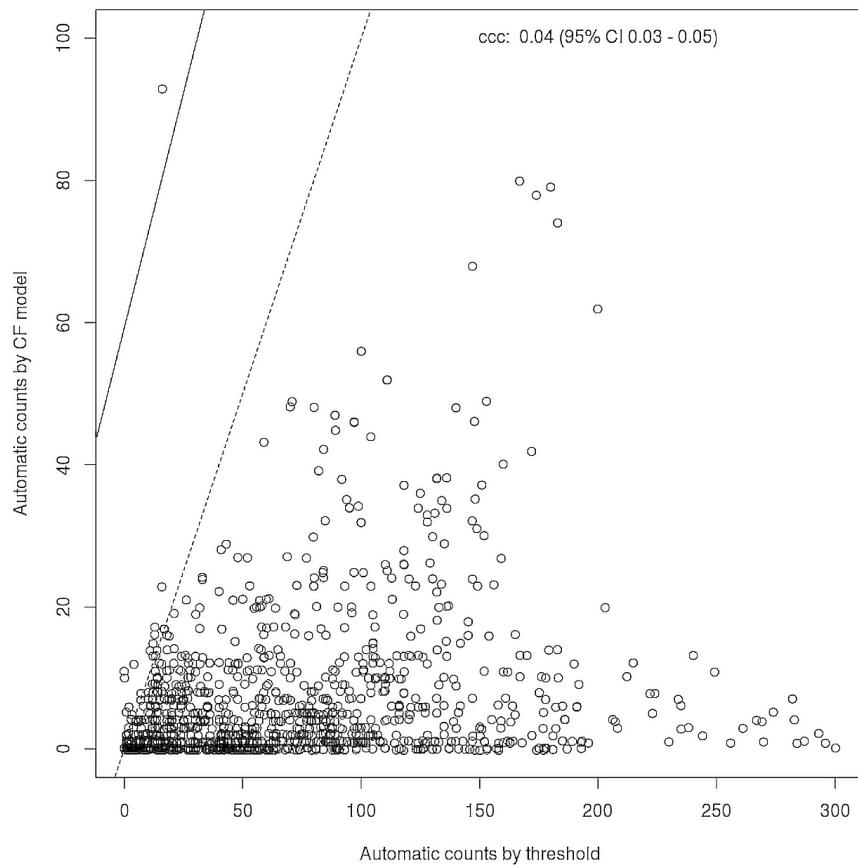


Figure 6.11: Analysis 1: A comparison of nest counts by different image analysis methods. Automatic counts from monitoring image stacks using classical thresholding (default intensity histogram) and the CF model (RF classifier). The dashed line shows perfect concordance; the solid line is the line of 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 image counts

Manual nest counts estimated from images and actual numbers taken in the field were compared. The variability between two image counts by two scorers are outlined first. Nest counts estimated from images by two scorers were compared to analyse the variability between observers. Figure 6.12 shows the output results from epiR. There was close agreement between the estimates from two scorers ($\rho_c = 0.867$) with good precision ($r = 0.891$) and accuracy ($C_b = 0.973$). The mean estimated counts for two scorers ($n = 170$) were compared against manual field counts in Figure 6.13. There was close agreement between the counts from two methods ($\rho_c = 0.622$), good precision ($r = 0.641$) and accuracy ($C_b = 0.97$).

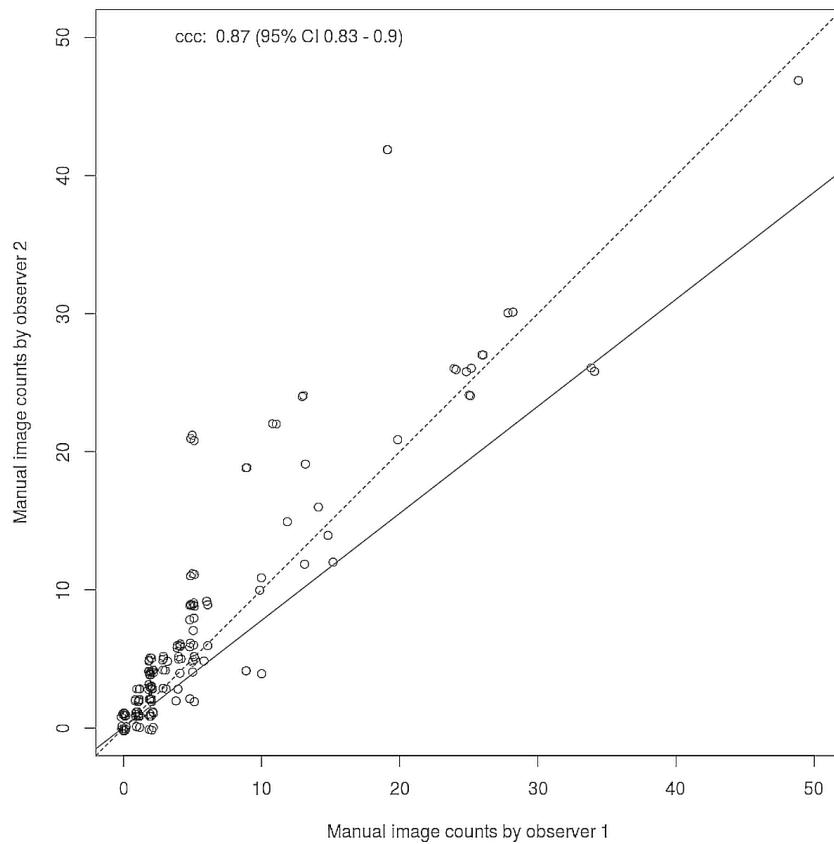


Figure 6.12: Analysis 2: A comparison of nest counts estimated from images by different scorers (observer 1 and 2) image analysis methods. The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.891$, $\rho_c = 0.867$ and $C_b = 0.973$.

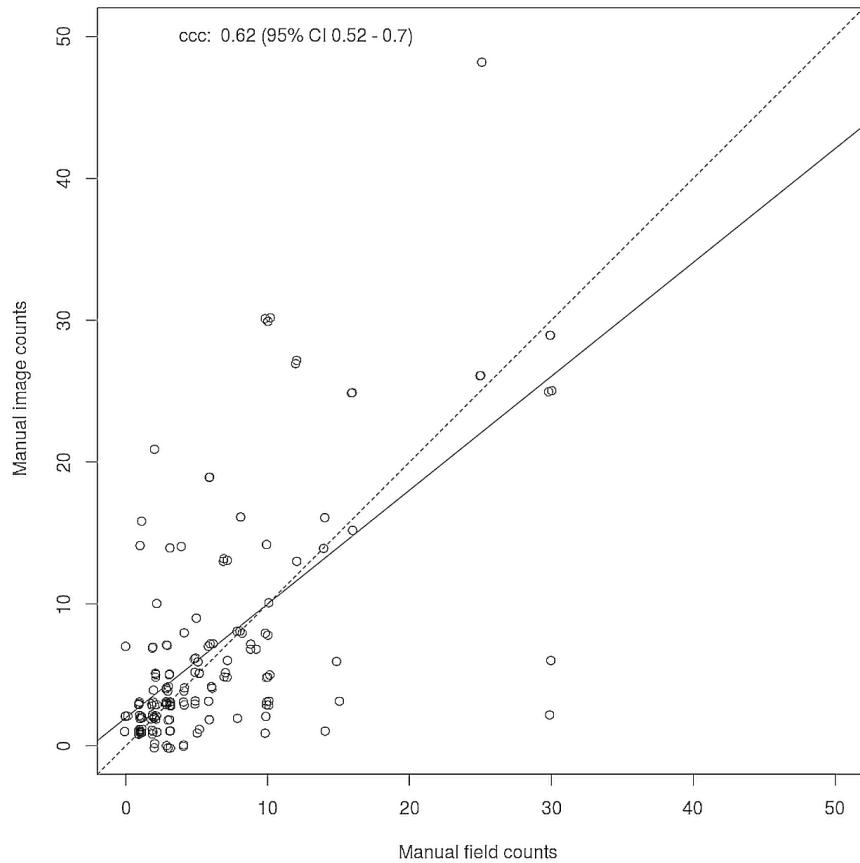


Figure 6.13: Analysis 3: A comparison of manual-field and manual-image (mean counts from two observers $n = 170$) nest counts. The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.641$, $\rho_c = 0.622$ and $C_b = 0.97$.

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 are shown in Figure 6.14. Manual-field counts were compared to automatic counts derived using the CF model and are shown in Figure 6.14. There was closer agreement between manual-field and automated-CF counts ($\rho_c = 0.738$), compared to manual-image counts ($\rho_c = 0.679$). The manual-field and manual-image counts were slightly more accurate ($C_b = 0.963$) than automatic-CF counts ($C_b = 0.891$); both had similar precision (manual-image: $r = 0.705$, manual-field: $\rho_c = 0.738$).

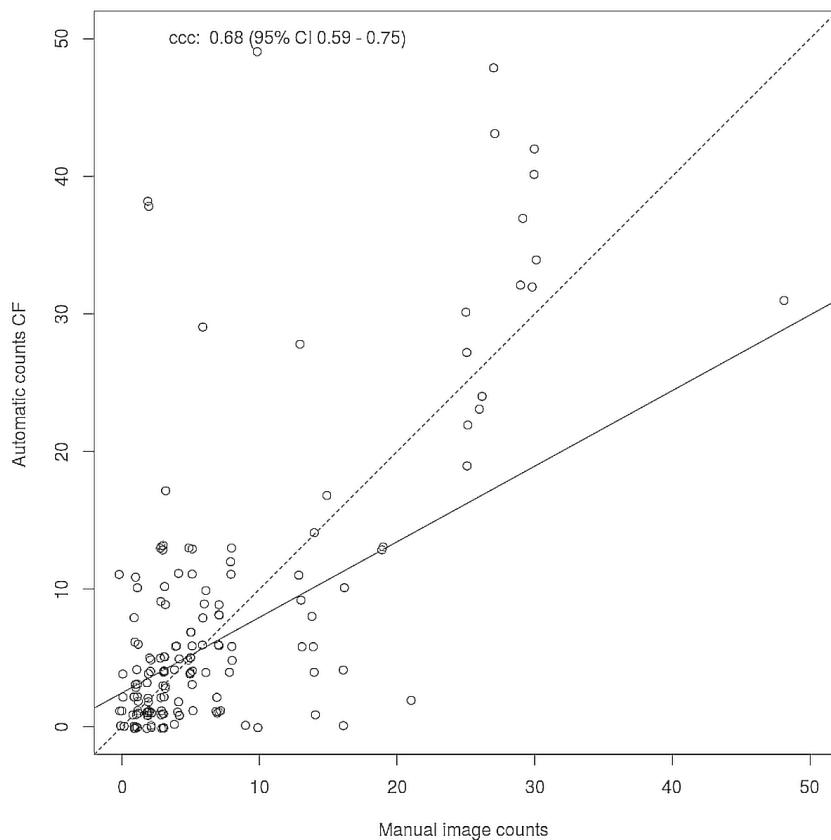


Figure 6.14: Analysis 4: A comparison of manual-image nests counts (the mean from two scorers $n = 170$) and automatic counts derived from the CF model. The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.705$, $\rho_c = 0.679$ and $C_b = 0.963$.

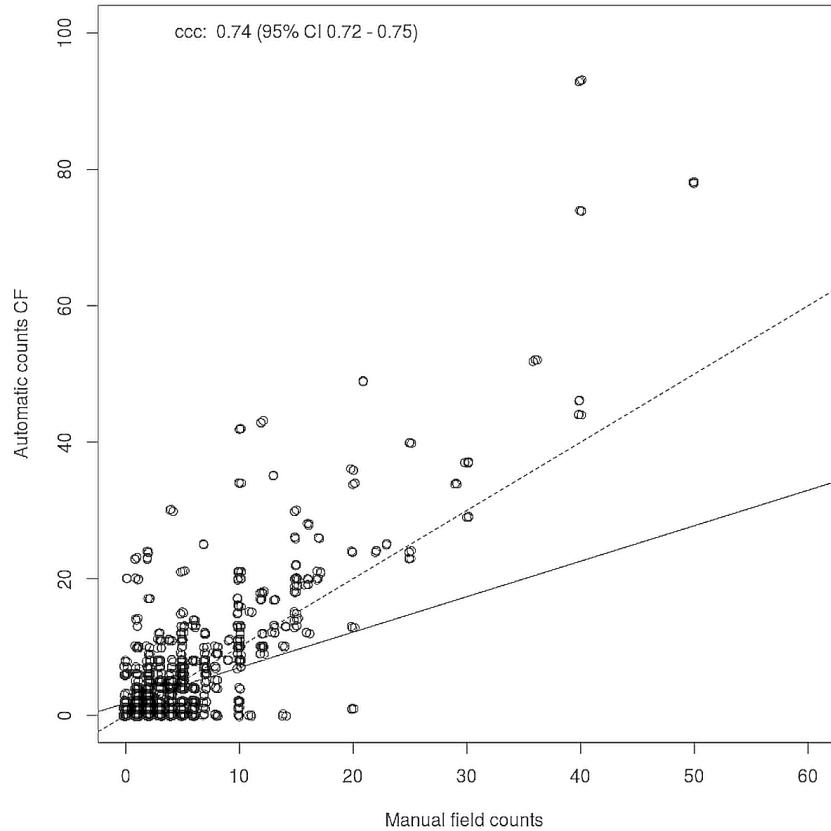


Figure 6.15: Analysis 5: A comparison of manual-field and automatic nest counts derived from segmentations using the CF model (median of three image collections $n = 520$). The dashed line shows perfect concordance; the solid line is the line of best fit. Performance measures: $r = 0.828$, $\rho_c =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

6.7 NUMBER OF ACTIVE NESTS BY SITE AND YEAR

The number of active nests counted by three methods at each location are organised by year in Figures 6.16, 6.17 and 6.18. The changes in the number of active nests over time can be generally visualised for each site. Methods produced similar temporal trends. Yearly mean values include error bars showing the standard error of the mean. Increased errors are associated with the number of monitoring samples collected (n) since they varied between sites and over time. Different methods produced more or less data samples (i.e. collections of near-replica images). For example, year 2014, $n = 36$, from three monitoring days, three sites, four grids; if analysis used three image collections then $n = 108$ (e.g. in the comparison of automatic counting methods).

Table 6.7: Number of yearly monitoring samples taken by the total number of monitoring days (n_m), for three sites (S_1 , S_2 and S_3) and four grids. In 2014 $n_m = 3$ so the total number of samples for analysis used in manual field counts is 12: $n_{S1} = 12$ (site 1), $n_{S2} = 12$ (site 2) and $n_{S3} = 12$ (site 3).

Year	Method	Samples		
		n_{S1}	n_{S2}	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		

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 below. There was a sharp decline in the mean number of active nests between 2010 and 2011; little changes between 2011 to 2013, with an increase in the mean number of active nests in 2014 ($n_{S1} = 12$). Similar trends were displayed by all methods. The automatic counts derived from image segmentation using the CF model (red), on average produced slightly higher mean counts than manual image (green) and manual field counts (blue).

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 below. There was a moderate decline in the mean number of active nests between 2010 and 2012; and a gradual increase the mean number of active nests recorded in 2012 and 2014. Similar trends were displayed by all methods. The automatic counts derived from image segmentation using the CF model (red), on average produced slightly higher mean counts than manual image counts (blue) but lower than manual field counts (green).

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 below. There was a moderate decline in the mean number of active nests between 2011 and 2013; and a gradual increase the mean number of active nests recorded in 2014. Similar trends were displayed by all methods. The automatic counts derived from image segmentation using the CF model (red), on average produced slightly lower mean counts than manual image (blue) and manual field counts (green).

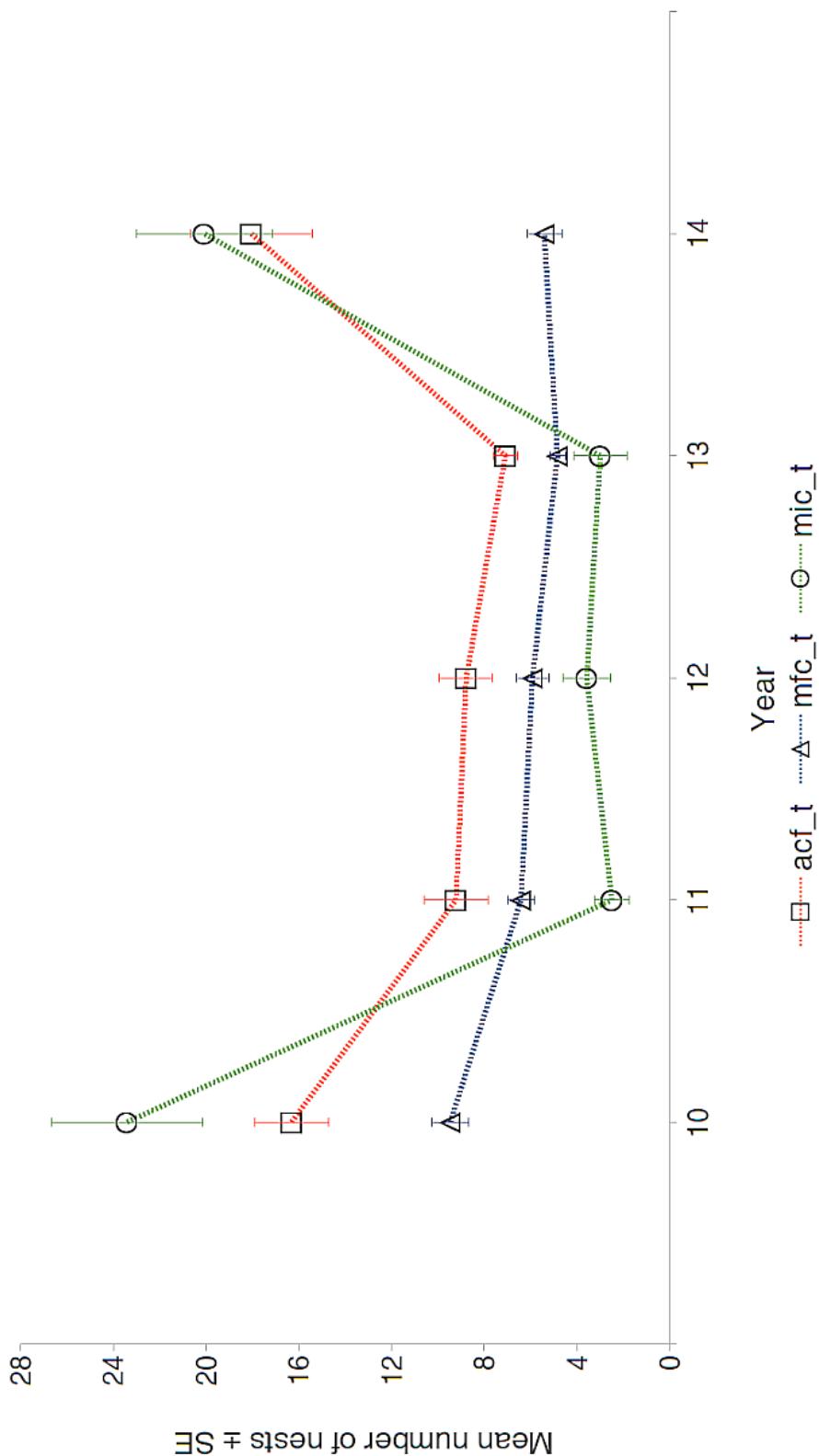


Figure 6.16: Site 1: Yearly mean nest counts \pm SE (error bars) by three methods. Automatically derived segmentations from CF model - *act_t*, $n = 493$, manual field counts (red) - *mfc_t*, $n = 232$ (blue) and manual image counts - *mic_t*, $n = 67$ (green).

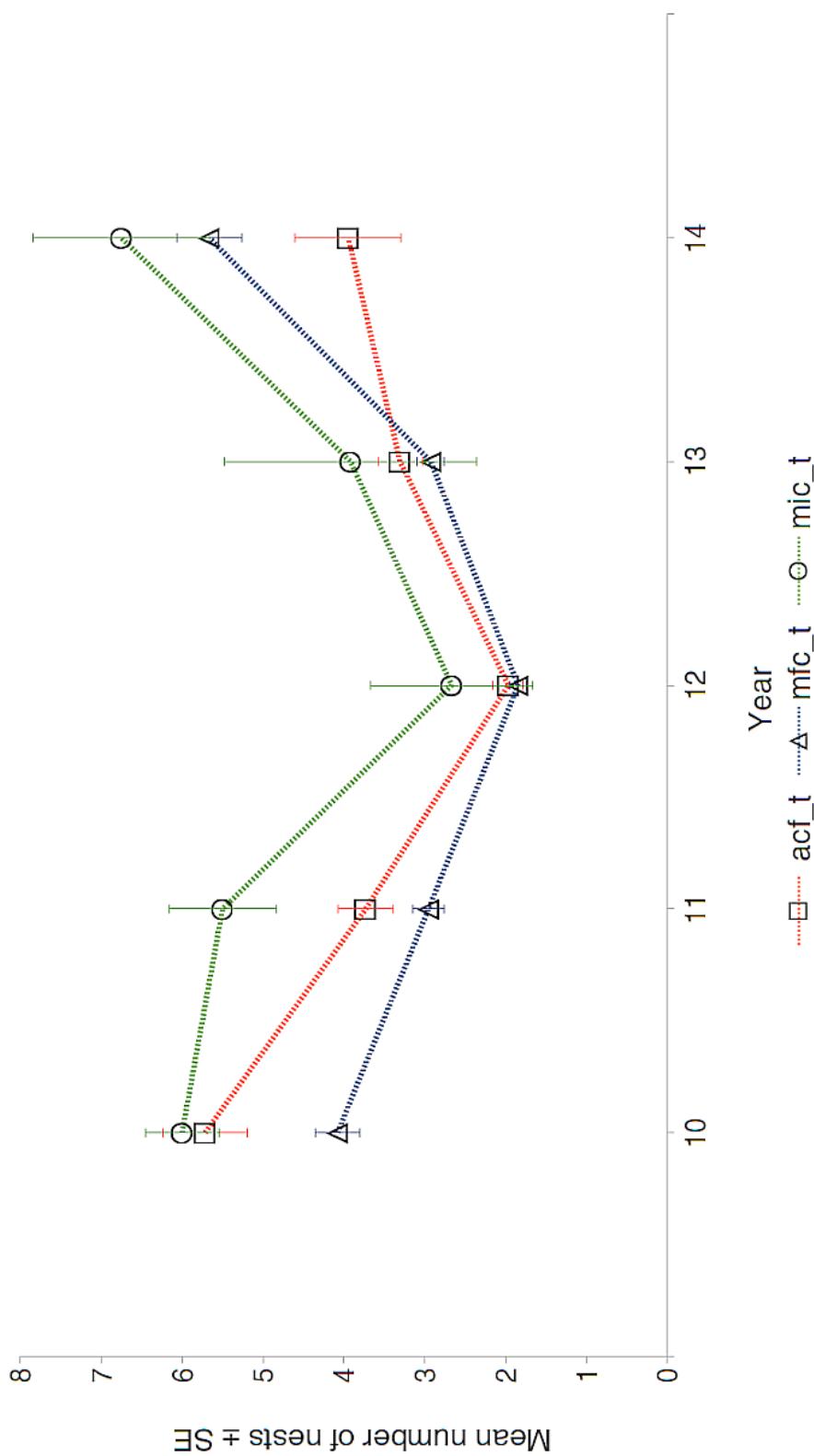


Figure 6.17: Site 2: Yearly mean nest counts \pm SE (error bars) by three methods. Automatically derived segmentations from CF model - $act_t, n = 502$ (red), manual field counts - $mfc_t, n = 232$ (blue) and manual image counts - $mic_t, n = 69$ (green).

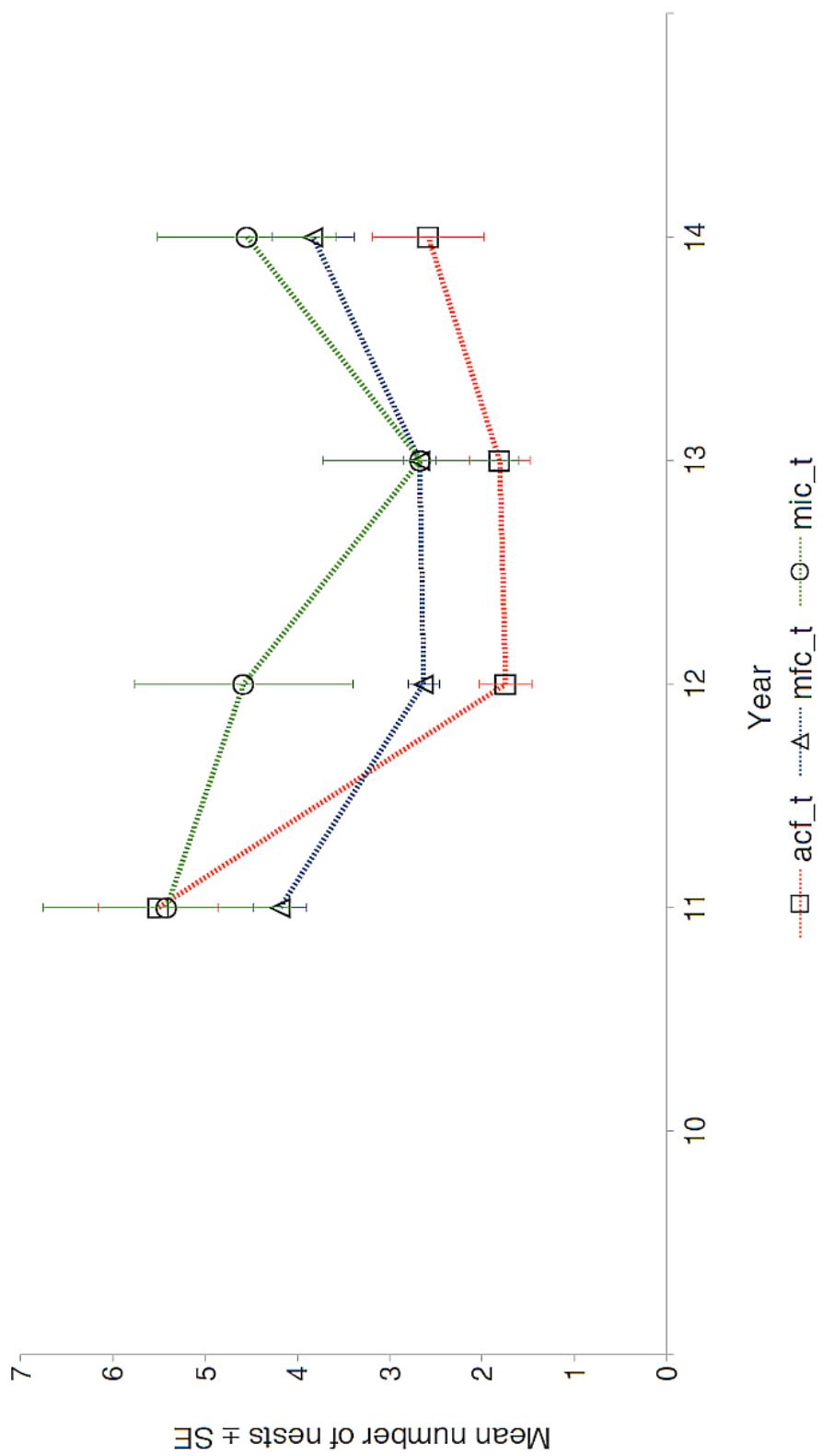


Figure 6.18: Site 3: Yearly mean nest counts \pm SE (error bars) by three methods. Automatically derived segmentations from CF model - act_t , $n = 289$ (red), manual field counts - mfc_t , $n = 168$ (blue) and manual image counts - mic_t , $n = 34$ (green).

DISCUSSIONS

OUTLINE

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 program which was designed to collect manual nest counts and images of active nests for comparative analysis and proof of concept. Monitoring was conducted of 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. This Chapter discusses the design, implementation and final performance evaluation of the image-centric nest monitoring system.

7.1 RESEARCH OVERVIEW

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 bees over space and time, (3) that digital images could be used in the place of manual visual counts, (4) it was possible to process digital images to reliably count active nests. The evidence supporting the central ecological hypothesis that *active nests of solitary ground nesting bees can provide a good proxy measure for populations* was taken from just a few studies [35, 11, 36]. There was no attempt to test this in the field. If time and resources had permitted, emergence traps and mark release recapture methods could have been implemented alongside manual nest counts and digital image acquisition. This could have provided sufficient data to test the *active nest population proxy* hypothesis, but may have also been beyond the scope of this technology-based research. Furthermore, the field methods used in this research were designed towards a comparative

Background context, rationale, design methodology and research outcomes.

analysis between manual and image-centric nest counts. Exactly the same nests were monitored over time. This approach would not meet the criteria required for ecological field research because nest counts were not taken from randomly selected locations within or between native bee communities.

Since this research was primarily technology based, an image-based active nest counting pipeline could have been developed using pre-existing digital image samples collected during previous studies [21, 16]. The imaging design did not depend on actual long term field monitoring, manual nest counts or new image collections. From a broader perspective, there was little value in designing the image-centric monitoring system if it was not going to be recognised as a potential tool to aid ecological research. From a review of current literature there are many promising tools that have been developed for ecological research but for one reason or another, are underutilised in practical applications [54, 55, 57]. In light of this and contrary to the suggestions by Lindenmayer and Likens [77] there was good motivation to test the image-centric monitoring system by gathering and using real field data. Furthermore, the design of this research was not context-free or based on retrofitted questions [77]. It was founded on natural history observations of communities of native bees in Whangarei, recorded every year for over ten years [21, 16]. Thousands of hours of empirical observations have contributed towards a substantial understanding of native bees and their communities. This knowledge has included the challenges involved with monitoring them and the types of technologies that were best suited to aid in a greater scientific understanding of their communities and populations. Although the field methods used in this thesis are based on untested assumptions about the nest-proxy hypothesis, and they could have been improved to support a more rigorous scientific outcome by randomising sample collections, the image meta-data analysis used in this thesis could not be described as parasitic [77]. However, it is also necessary to point out that if the number of active nests do not provide a good proxy for populations then the image-centric monitoring system outlined in this research is valuable for measuring the changes in active nests only. This was an issue that should have been properly tested when the research methodology was being designed. It is an issue that could have significant implications on the overall outcomes of this study because the wider scientific benefits of the nest monitoring system could be limited. The remaining discussions in this Chapter are therefore primarily concerned with design aspects of the image-centric monitoring system, as it was applied to measure the number of active nests rather than questioning the premise of nest-proxy measurements.

Open source tools were selected for nest image analysis to encourage community engagement in ecological research [80]. The process-

ing pipeline and imaging methods are described in detail so they can be replicated. The core image analysis technique was based on trainable segmentations. Traditional segmentation techniques are *repeatable* and reliable but are they only appropriate if the intensity values delineating target objects are well defined. Monitoring images were highly variable and 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, but there are new techniques based on interactive machine learning. Therefore nest image segmentations were achieved using a semi-supervised machine learner and the TWS workbench. Compared to classical binarization methods it would be difficult to exactly replicate the machine learning image segmentations used in this thesis. Therefore the procedures developed and used to optimise, train and apply machine learning classifiers for segmentations of active nest images in TWS are described in detail. The image processing scripts, raw field and image data are also provided via Github for re-analysis or for use future research.

When the performance of the image-centric monitoring system was benchmarked against manual field counts the results were promising. Analysis showed the number of nests counted from images visually or by image analysis procedures, compared favourably with those from manual counts. The final monitoring results presented provided a reliable estimation of the number of active nests at the communities evaluated and the changes in nest numbers over time. However, as previously stated, because collection data was not randomly sampled the monitoring results from this study cannot be used as a proxy for native bee populations at the selected communities. Nevertheless, this research provided good evidence to show the populations of native bees could be reliably measured by using the image-centric monitoring method outlined in this thesis. In the future nest image data could be collected using proper random sampling methods thus providing data that could be used for a more rigorous scientific analysis. Field sampling methods would have little impact on the performance of the image-centric monitoring system. If the number of active nests can be manually counted in the field, then they can be estimated from images using the image-centric monitoring system with good agreement, accuracy and precision. The imaging method would stand alone and would not depend on collecting manual nest count data. It would also be possible to modify aspects of the image acquisition methods to incorporate remote imaging technology. This would reduce the dependence on manual labour and increase data capture for the scientific community.

7.2 ACTIVE NESTS

There were other *types* of images that could have been used to reflect either the population changes over time or the species diversity of native bees within selected communities. This thesis has focused on the monitoring images of active nests. However, there were a range of image types were collected during field monitoring 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 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 for the following reasons:

1. The images of active nests were easy to *capture*.
2. The image area was easy to *regulate*.
3. The image analysis 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 relatively easy to capture when compared to moving insects (e.g. Figure 7.1 (b)). Image acquisition was easier to govern for active nest data. This was achieved by using a standard of measure placed over an area of active nests. This was used to determine the focal boundaries of digital images. The image analysis 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) six different insects are shown on single flower heads. The images were captured within a set time-frame and equate to a survey method that 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 that were necessary to segment; active nests from other backgrounds. Furthermore, image analysis based on the number of active nests in the images directly relates to an estimate of the populations of bees with minimal post-processing or statistical analysis. 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

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

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



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

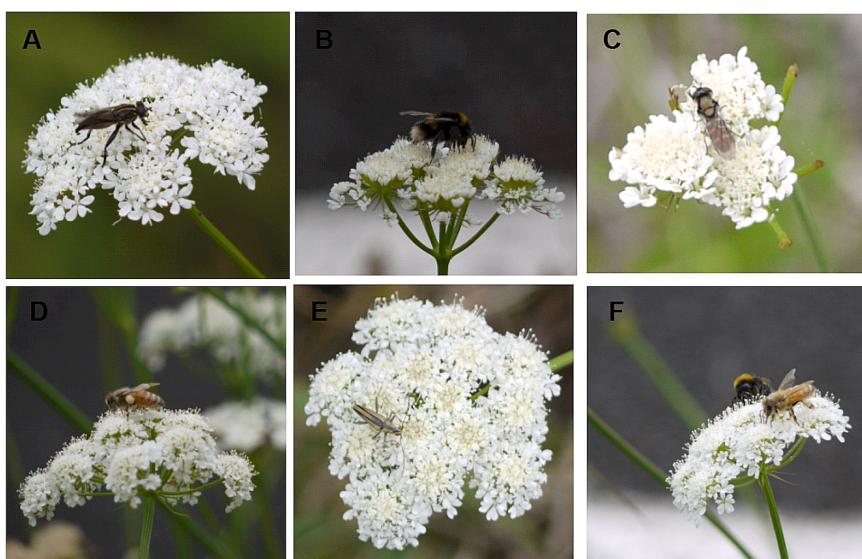


Figure 7.1: Image sequences of foraging bees and other insects focusing (a) on the same plant in a set time-frame and (b) on all plants and insects in a set area.

(a) Image of a sweep net collection.



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

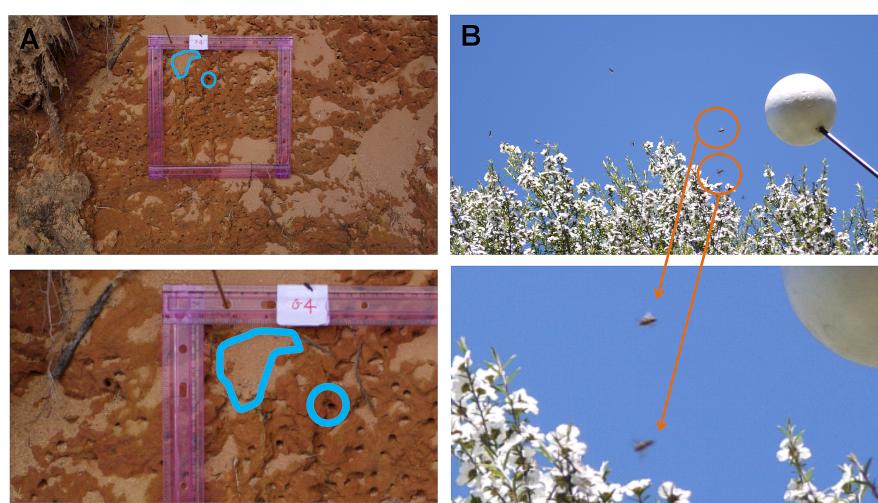


Figure 7.2: Images of (a) sweep net collections and (b) standard of measures for monitoring – grids for active nests (left) and a sphere for bees in flight (right).

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 using evidence of soil excavations or entry holes. The nest counts taken at the start of each season were important. The nest entry holes were more defined and easier to count. Because of this they were the more reliable than counts taken towards the height of the active flight season. Mid season the entry holes to active nests would 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 (Figure 4.3). 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 and completely covered nest entrances. In contrast to Mt. Tiger, there were communities of bees established along a flat area of ground at Mt. Parihaka. These 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 from the ground surface vegetation as shown in Figure 4.4.

7.3 MONITORING IMAGES

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 Panasonic DMC-G1 camera was used for acquisition of nest images which were saved in compressed JPEG format. Although off-the-shelf digital cameras and encrypted image formats are not recommended for biological image analysis they are used in some environmental image applications ref.

*Image formats,
quality and
variability.*

7.3.1 *Image format*

There are no agreed standards for raw files so proprietary platform dependent software is often bundled with off-the-shelf digital cameras 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. Reading and converting the RAW files was an added step in the image processing pipeline. During early tests the added file work was shown to be too resource intensive and time consuming to consider using raw files in the bee monitoring system.

Nonetheless, the quality of image data can impact the reliability of image processing. At least where biological image analysis is con-

cerned high resolution raw images are preferred. However, biological images are captured under laboratory conditions and memory capacity is not usually an issue. On-board camera memory was a consideration for field collections. Because the field camera had a limited amount of memory compressed high quality JPEG formats were more practical and cost effective. Some studies have shown image compression is not always an issue. Paola & Schowengerdt [113] for example, tested three different classification scenarios and found that high quality classifications could still be achieved with a CR of 10:1 [113].

The compression ratio of the field camera was closely examined to determine the qualitative affects on monitoring images. Since active bee nests are ill defined, 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 with a CR of 34.71:1 using XnView software. After the image was compressed the affects were visually checked. The a close-up view highlights some minimal visible affects from the encryption. This can be seen in Figure 7.3 (a)–right as slight blocking. When images were converted on-camera the CR’s were substantially lower at around 2.75:1. This level of compression was less than 10:1 and was well within the boundaries considered acceptable for image classification applications recommended by Paola & Schowengerdt [113] and others [109, 111].

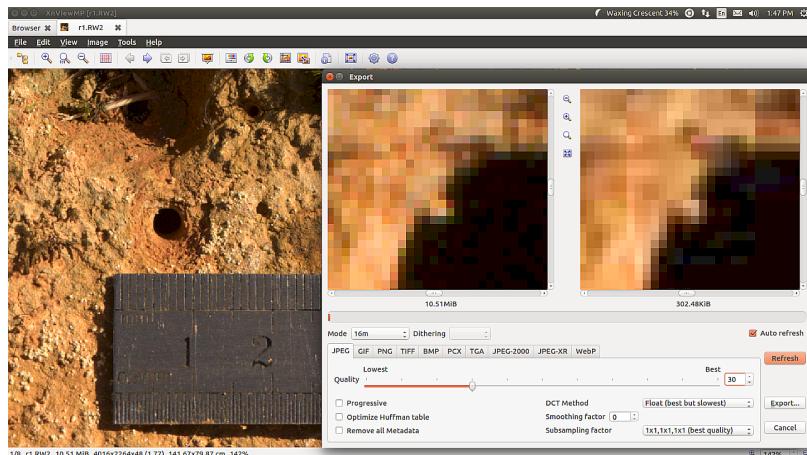


Figure 7.3: A raw image of a tiger beetle nest (background image) taken with the monitoring camera (Panasonic) and a close-up example of file compression using XnView Software.

7.3.2 Outdoor images

A number of techniques can be used to help standardise image acquisition and reduce the affects of outdoor variations [100]. Several

techniques were tested during the prototyping stages of the monitoring system design. For example, flood lights were tested on monitoring nests 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 since field samples were necessary. 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 [100, 105, 106]. There are few practical methods that can be used to improve the quality of natural outdoor images even though there are some novel approaches (e.g. see the design by Burks [100]). A review of the current literature on this topic revealed a number of studies that reported the problems associated with analysis of complex natural images could be mitigated by using RFs for image classifications [105, 106]. The studies showed there were good results from image segmentations using the RF classifier even when compressed, outdoor images were acquired using off-the-shelf digital cameras [105, 106].

The quality of monitoring images was a key issue that had an critical impact on the design of the image-centric monitoring system. A closer examination of the nest images and the decisions that determined the imaging pipeline tools and methods are described in further detail throughout this Chapter.

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 Appendices for reference and there are a range of resources for development of Fiji and ImageJ scripts available online.

Nest images were examined in Fiji. A test stack was compiled using representative nest images selected from each site-grid to reflect the range and variation typically encountered. The test stack, or a single slice from the stack were used to test a range of processing techniques examined over the next few Sections (7.4).

In most analysis workflows images are prepared by using *contrast enhancement* or *histogram equalisation* operations; generally called image normalisations. A single image of a horizontal ground nest was examined in Fiji using different contrast enhancement settings to investigate the most appropriate levels for monitoring images. There



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)

were four different functions in the enhance contrast toolbox that were 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 results from tests are shown in Figure 7.5 (b).

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 saturated pixels was set to 0.4%. This setting was useful for improving the visual qualities of the nest images for displaying on the LCD! (LCD!) monitor used in this research. Histogram equalisation was not used for nest images since this operation fundamentally changes the digital data.

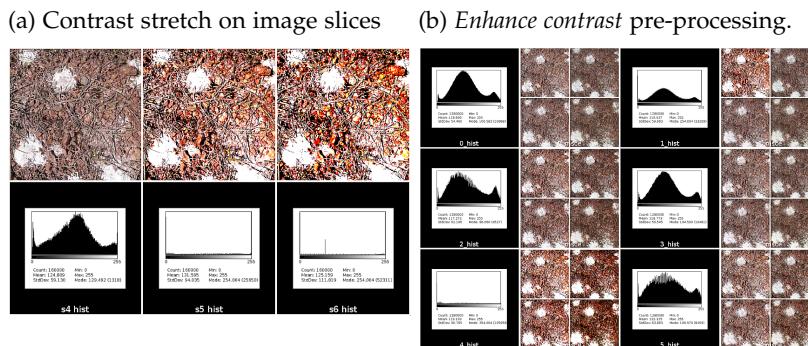


Figure 7.5: Image of a horizontal nest used to test the affects of (a) contrast stretching using 0.4%, 40% and 90% saturated pixels (b) and Enhance contrast toolbox stack functions. Scheme 3 was used for all monitoring images. The number of saturated pixels was set to 0.4% and the *Process all slices*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

7.4 SEGMENTATION METHODS

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

Segmentation challenges

aspects of image data using reproducible and objective techniques, with some capacity to *generalise* over a given range of image data variability. The performance of any segmentation method 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.

To examine this further three segmentation techniques were tested on a sample of representative nest images. These included classical thresholding edge detection by Canny-Deriche filtering and statistical region merging.

7.4.1 Thresholding by intensity

Intensity based thresholding methods produce straightforward segmentations; 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 (slice 5: site 2/horizontal ground nest/grid 4). Fiji Auto threshold Try All¹ [150] function was selected to determine which method best suited the nest image. The active nest in image slice 5, is visually noticeable; the white clay tumulus 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 below.

(a) Test slice 6: Mt. Parihaka ground nest image. (b) Test slice 1: Mt. Tiger roadside bank nest image.

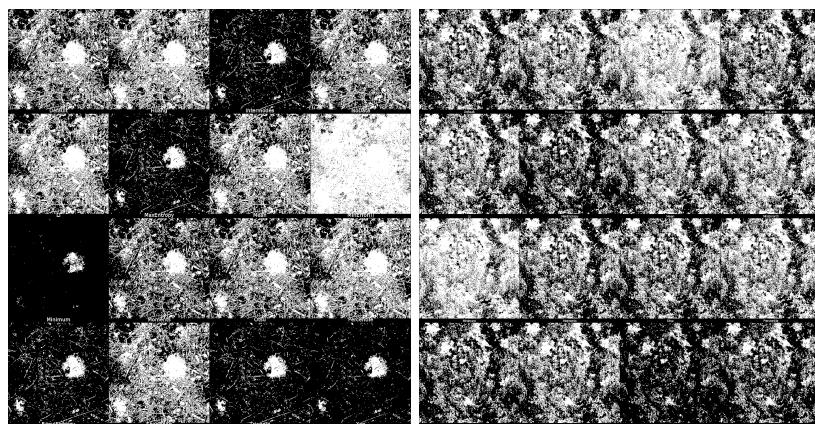


Figure 7.6: Auto thresholds for monitoring images of (a) a horizontal ground nest (b) a roadside bank nest.

The results demonstrate that from 16 possible schemes *Minimum* returned the highest quality segmentation. In this case the image was

¹ Try All http://fiji.sc/Auto_Threshold

adequately segmented *before* any post-processing operations were applied. When post-processing operators were also applied the results confirmed an automatic threshold method using the *Minimum* scheme would produce satisfactory image segmentations for horizontal ground nests. These are shown in Figure 7.7(a)–(e). However, when the auto threshold Try All function was applied to test slice 1 the nest image was not properly segmented (slice 1: site 1/bank nest/grid 1). This is demonstrated in Figure 7.6 (b). These results helped to define the constraints of segmentation methods. They showed the *complete range* of monitoring images could not be segmented easily using intensity-based threshold levels. They also confirmed automatic thresholds could be used to segment images of horizontal grounds nests. Therefore they would be suitable for segmenting 1/6th of the monitoring images collected.

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 such as 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* [116].

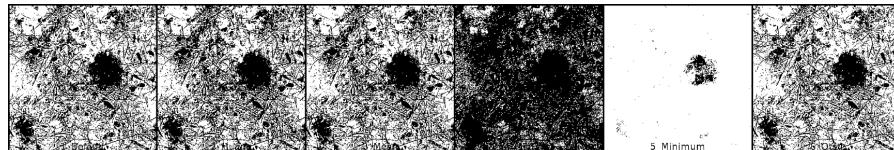
Figure 7.8 demonstrates the affects of the Canny-Deriche filter after different smoothing factors (α) were applied to the horizontal ground nest image. Six smoothing factors were applied to the ground nest image as follows: slices 1–6: $\alpha = 1.0, 0.9, 0.7, 0.6, 0.45$ and 0.15 . The image results from each smoothing were compiled into a stack 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 slices 1–6 on Figure 7.8 (a)–(f).

The lower smoothing values returned slightly better results, which 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.

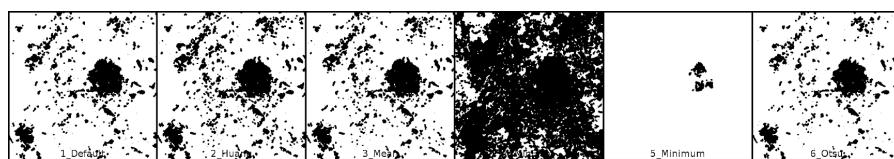
7.4.3 Statistical region merging

In a final examination a region-based method, SRM [117, 118] was tested on the ground nest test image. The results from SRM investigations are shown in Figure 7.9 (a)–(f) using six different values of Q (slices 1–6: $Q = 1, 2, 3, 6, 8$ and 16).

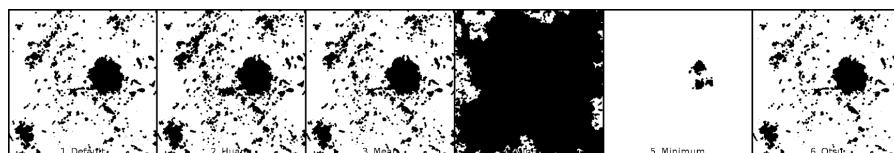
(a) Binary output from six thresholding methods applied to the image of a horizontal ground nest (Mt. Parihaka, grid 4). Output slices 1-6 = Fiji Default threshold, Huang, Mean, MinError, Minimum and Otsu.



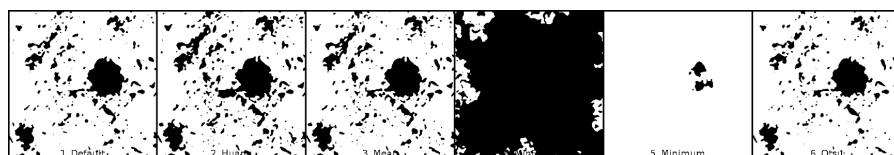
(b) Open binary operation.



(c) Fill holes binary operation.



(d) Close binary operation.



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

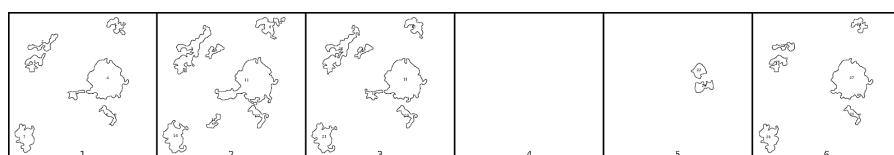
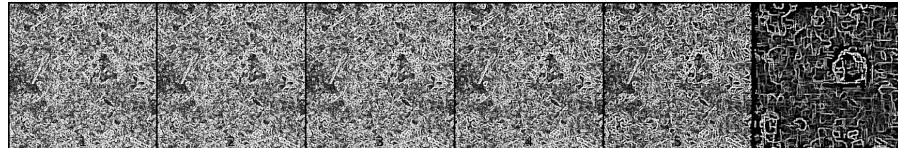
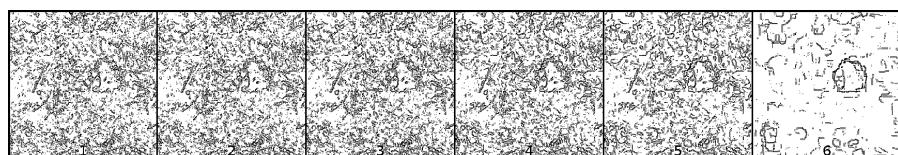


Figure 7.7: Binary results from six thresholding methods (a) applied to a horizontal ground nest image. Common post-processing pipeline morphological operators were applied to the test image including (b) open, (c) fill holes, d) close and (e) count particles.

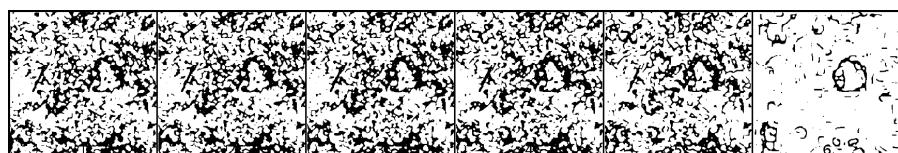
(a) Canny-Deriche filter results from different smoothing values slices 1-6: $\alpha = 1.0, 0.9, 0.7, 0.6, 0.45$ and 0.15 .



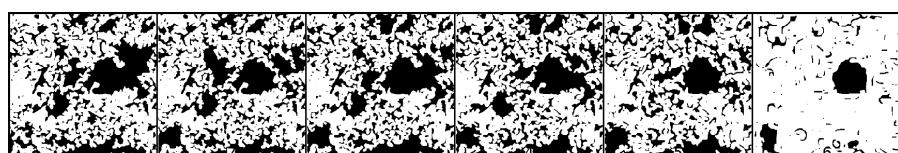
(b) Make *Binary*.



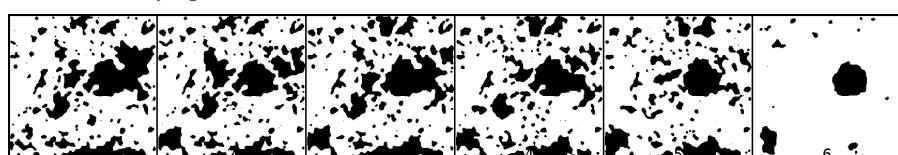
(c) *Open* binary operation.



(d) *Fill holes* binary operation.



(e) *Close* binary operation.



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

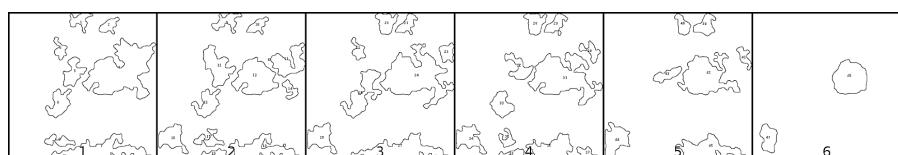
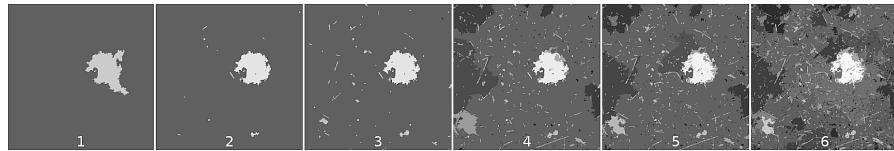
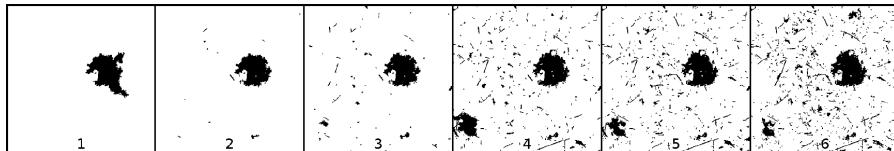


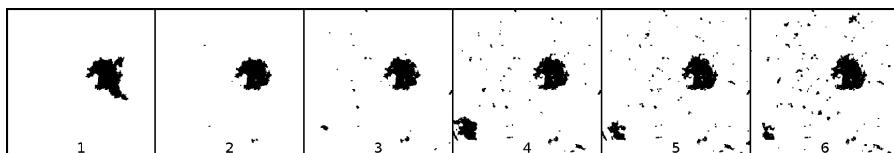
Figure 7.8: Segmentation by edge detection (a) applied to a horizontal ground nest image with a pipeline of common post-processing operations (b)–(f).

(a) SRM with slices 1–6 set to segment regions of $Q = 1, 2, 3, 6, 8$ and 16 .

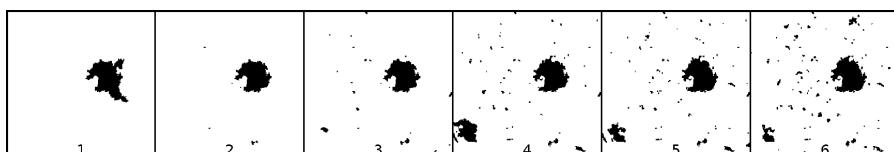
(b) Make Binary.



(c) Open binary operation.



(d) Fill holes binary operation.



(e) Close binary operation.

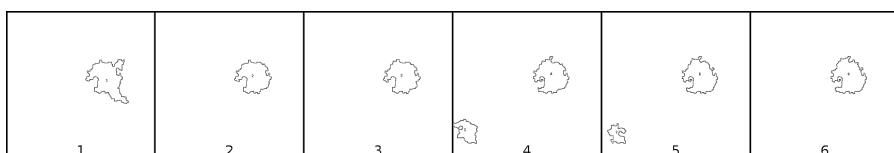
(f) Analyze particles plugin: set to count all image objects between the sizes of $100-\infty$ pixels with a circularity morphology between 0.1-0.9.

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

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 a challenging. Ground truth labels—or manual annotations were added to the representative stack of nest images for comparison with binary results. Figure 7.10 (a) confirms, most of the test images were over-segmented to degrees that could not be easily managed by post-processing operations. For example, there are no active nests in slice 1. Yet the Canny-Deriche edge detection and SRM methods produced binary outputs resulting in multiple segmentations. This is demonstrated in Figures 7.10 (b)–(d) and 7.10 (e)–(g) respectively.

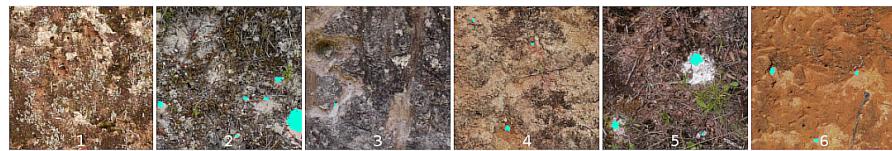
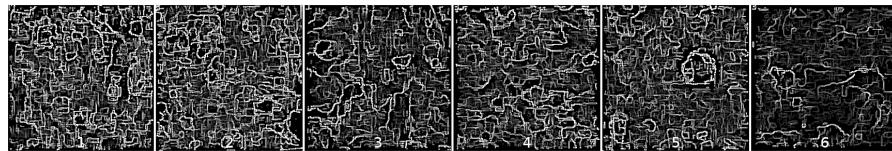
After applying post-processing operations a number of objects were detected that did not correspond to ground truth labels. In most analyses *any* aspect of the imaging pipeline can be tweaked to accommodate image characteristics or to highlight the key data required for specific applications. However, there were *no* techniques that could improve the variability of monitoring images and few image processing techniques that could improve image segmentations. It was concluded that neither of the techniques appropriately segmented the range of nest images. Because there was no single intensity value that could properly define the foreground active nests from backgrounds, none of the traditional methods were suitable for generalising over the range of monitoring images.

7.5 TRAINABLE SEGMENTATIONS

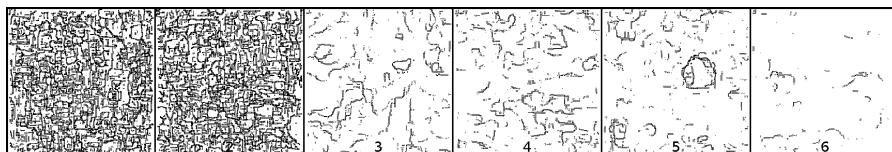
There were other characteristics in the monitoring images that defined active nests; such as colour and texture of tumulii. Overall the monitoring images were highly variable across time and space; therefore image segmentations were challenging. The images of horizontal nests would be adequately segmented 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 results from post-processed binary operations. 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.

A review of current literature showed that when other segmentation methods fail machine learning could provide a plethora of alternative solutions. Some of these were tested on the stack of representative nest images using Fiji and the TWS toolbox. 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).

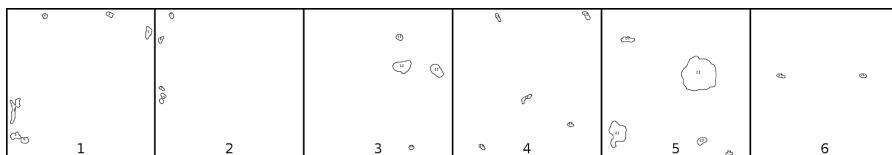
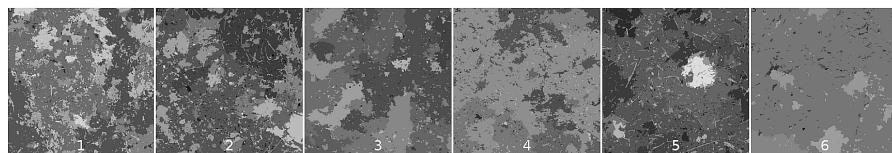
(a) Stack of representative nest images.

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

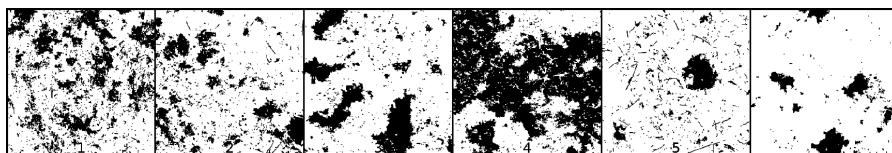
(c) Make Binary.



(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) Canny-Deriche filter edge detection and (e) SRM and final post-processing operations.

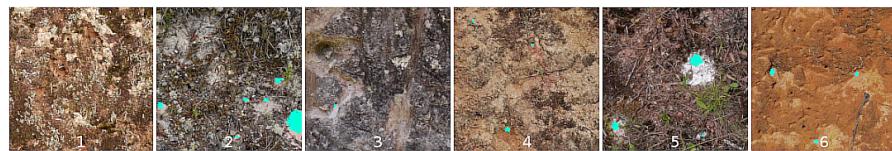
This includes an over-view of training methods and optimisation techniques and a review of the classifiers designed for the image-centric monitoring system.

Trainable image segmentation techniques work by utilising human visual knowledge to provide a machine learning algorithm with a set of expertly labelled examples. 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. These can be grouped according to their main filter functions (Table 7.1). 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 one it was trained on. This is a summary of the overall the process but it is relatively straightforward.

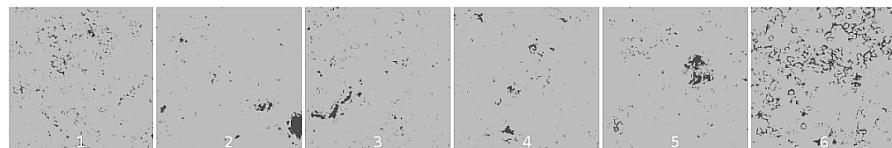
There are some aspects of trainable segmentations that are difficult to quantify. For example, it can be hard to determine what combination of image features best describes key objects. Consequently, some aspects of classifier training are intuitive and subjective. There are also other confounding decisions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

A classifier was constructed using TWS to segment the stack of representative stack of images. The results from the classifications were a stack of binary images which are shown in Figure 7.11. The machine learner classified every pixel in the test images as belonging 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 with the classified output from a machine learner in Figure 7.11. When the variable image stack was used for comparative checks the final number of nest counted from the machine learning method, shown in Figure 7.11 (g), was in much closer agreement with those visually counted and annotated on images (i.e. ground *truth labels*), shown in Figure 7.11 (a). The segmentation results on the variable nest images were *significantly* different to those produced by classical Canny-Deriche edge detection and SRM techniques (Figure 7.10). The difference between the methods was more obvious when the final particle count results from all methods were visualised alongside each other; these are shown in Figure 7.12.

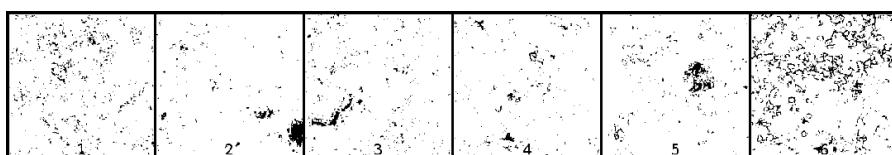
(a) Annotated stack of representative nest images.



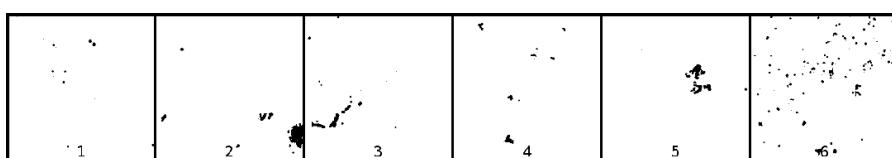
(b) Classified output from TWS.



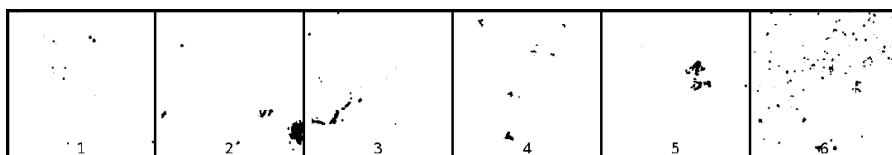
(c) Make Binary



(d) Open binary operation.



(e) Fill holes binary operation.



(f) Close binary operation.

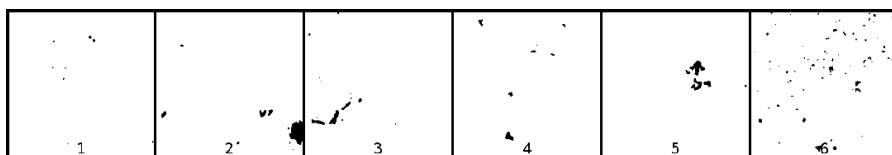
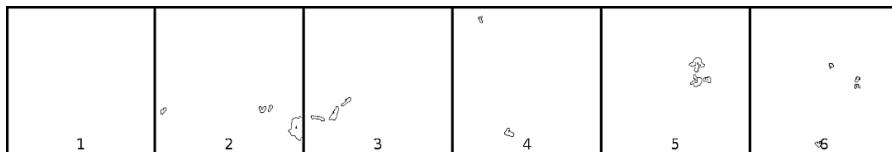
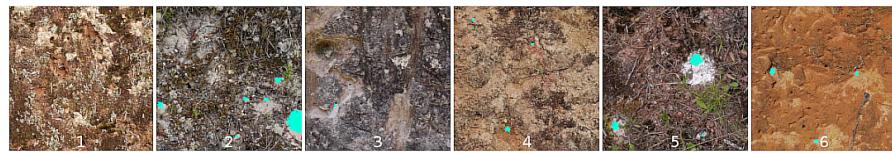
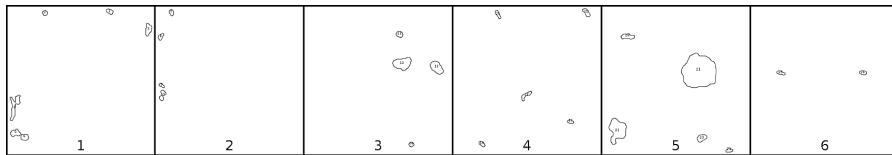
(g) Analyze particles plugin: set to count all image objects between the sizes of 100- ∞ pixels with a circularity morphology between 0.1-0.9.

Figure 7.11: Trainable segmentations on variable nest images (a) with ground truth annotations in blue with a pipeline of common post-processing operations (b)–(g).

(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.

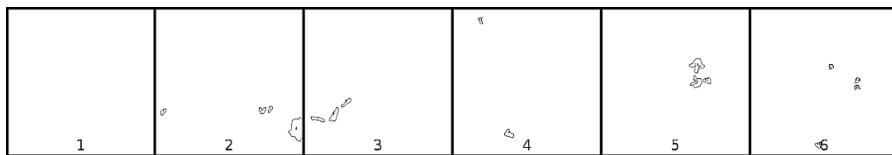


Figure 7.12: The ground truth labels of the (a) stack of variable nest images are compared to the final post-processed outputs from segmentations using the: (b) Canny-Deriche edge detector, (c) SRM and (d) TWSs. Results from TWSs are in closer agreement with the ground truth labels in each slice.

Table 7.1: Filters available in the TWS plugin 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.

7.6 EMPIRICAL LESSONS

Since 2010 the TWS workbench has undergone significant developments. Each year the package has become more versatile. For example, early tests and initial training investigations could only be performed on 8-bit grey-scale images and TWS was not initially integrated with Weka [63]. Also, RF *feature importances* were not easily examined during early tests, however for the final analysis it was possible to compute feature importances and quantitatively examine the impact of filters on classifications. For this reason over the course of this research, each successive classifier testing training and analysis changed [16]. 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.

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

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 the images selected for training, the traces supplied to represent classes during training and 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. The second was to determine the best tuning settings for the RF model; generally referred to as classifier optimisation. 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. This was because it is the *total combination* of features that contributes to the final accuracy of machine learning models; according

to most current literature. Generally, the more features that are provided during training, the better the final classifiers are. However, there were computational costs associated with the construction of the features stacks for monitoring nest-classifiers. Stack creation and processing, was fundamentally limited by the amount of desktop RAM available; this was around 7.2Gb in total.

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. It was hypothesised that these filters would also be the most important for classifying images of active nests.

Initial tests were performed and the classifiers were re-tuned based on:

- 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 and the training was re-run. The segmentation results were checked 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 46 (58%) did not provide any additional information. Therefore the second test classifier, C₂ was tuned for to provide the optimal nest image features. 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 due to the iterative process from the Anisotropic diffusion filter.

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. 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 since 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 since training and classifications would have taken longer. Also, according to Breiman [151] the number of trees in a RF should be set to 200 initially and tuned as required. While the initial number of random features for RF is given by the square root of the maximum number of features [151]. In the final feature-speed optimised test there were 20 features used to construct the test classifier, C₃. Thus following Breiman's [151] rule of thumb the ideal number of random features provided to the model for training should have been around 5. These two parameters were evaluated with Weka Experimenter using the approximate guidelines suggested by Breiman [151].

In the first analysis, twenty one RF models were loaded into Weka Experimenter Algorithms for testing and the experiment was saved for repeat investigations. A 10 fold cross validation with a maximum of 10 iterations were selected for tests and used in the analysis. Each classifier model was tuned with a different number of trees. These were set between N = 10 – 1000. The out of bag error and overall time to complete processing was evaluated. The results showed the processing time exponentially increased as the number of trees increased. The tests also indicated there was little improvement in the out of bag error beyond a certain number of trees. 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 therefore 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 Algorithms for testing using a 10 fold cross validation experiment with 10 iterations. Each classifier model was tuned with a different number of random features. These were set between M = 0 – 20. The test results indicated the classifier performance *improved* when more *random features* were provided to the model for training. These evaluations also confirmed of Breimans [151] rule of thumb for selecting the ideal number of random features. For example at M = 8 the oo_b = 0.62% which was slightly better than 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

The previous investigations confirmed the ideal features that should be provided to the monitoring classifiers for training; and the ideal tuning parameters for the RF models. However, although RF model is the default machine learner in TWS, there are up to fifty different types of Weka machine learners available. In theory, other types could have been selected and trained to classify the monitoring images. For example, support vector machines are popular classifiers. They also perform well in some applications and are generally believed to be one of the most accurate classifiers available to date.

Therefore, in order to verify the performance of the RF models, 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 asses the performance of the RF models over other types of machine learners when applied to the data-set created from the representative stack of nest images. A 10 fold cross validation method, with a maximum of 10 iterations were selected for tests. Four RF models were added to the experiment including the final monitoring classifier which was the benchmark model. Six other models were included and are listed below. All other classifiers were left at Weka or Fiji default settings. The percentage of correctly classified instances were reported for the analysis using $n = 1000$ data results; with a confidence of 0.05 in a paired-corrected two tailed test.

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%. Unexpectedly naive Bayes results were lower than the ZeroR (M2) model which was the baseline classifier and 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, including a brief description of the model, the percentage correctly classified instances and the Weka model ID 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

Determining which were the most important performance measures.

- 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|1.0|-L|0.0010|-P|1.0E-12|-N|0|-V|-1|-W|1|-K|
 functions.supportVector.PolyKernel-E|1.0|-C|250007

The test results probably could not be used to *preclude* the usefulness of other classifiers for the nest-image segmentations. However, the results did highlight the difficulties associated with calibrating some models. Extensive skills and time are required to tune specific machine learners for any given task; and most require tuning for each specific task. 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. Each model requires specific tuning and 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 systems were equally important to consider. At least 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) so they could be used in the most efficient way possible and easily 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.

Because image classification tools were a key aspect of the image-centric monitoring system at the segmentation stage of the imaging pipeline, 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.

7.6.5 Segmentation performance

The importance of manual and visual agreements for active nest counts. In biomedical imaging applications, segmentation metrics are generally applied to measure the performance of techniques of methods. 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 for final comparisons. There are few other avenues that can be used to determine the validity of automated image segmentations in these cases. However, in this research, the number of active nests that are represented in each of the monitoring images was manually counted in the field. Therefore the manual-field nest counts provided a valid benchmark for comparison. 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 analyses of nest image segmentations was a necessary part of the development of the image-centric monitoring system. Image segmentations were constantly checked and used to optimise classifiers and refine post-processing operations. These methods are discussed in the following paragraphs.

The representative image stack of six images were fully processed using the final test classifier CF and the results were visually assessed. Original training for the test classifiers was completed in a single ses-

*Manual nest counts
and visual
image-count
agreements.*

sion. This was not sufficient to properly train the classifier. When the segmentations from CF were post-processed, the final nest counts did not correspond well with either the manual image or field counts for the representative stack of images. The number of objects counted in slice 6 was particularly high at 64 (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 a final time (CF3).

The final counts were visually checked against raw RGB images for verification and against the manual field counts. This is summarised in 7.2 below. The output log from TWS training is shown in Listing 7.1.

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	s1	s2	s3	s4	s5	s6
CF1	21	11	14	22	4	64
CF2	21	2	3	13	3	14
CF3	7	1	8	4	2	3
Manual image	3	3	3	2	2	2
Manual field	2	4	4	3	4	9

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 CF-class 2 = 816, CF2-class 2 = 864 and C3-class 2 = 996. Output results also showed there was little change in the out of bag error between classifiers: $CF_1-oo_b = 0.581\%$, $CF_2-oo_b = 0.661\%$ and $CF_3-oo_b = 0.577\%$. However, the segmentations changed markedly.

The number of objects counted in slice 1 and 8 were marginally raised. All other slices were reasonably well segmented. These results highlighted several concepts. Firstly, the classifiers could be retrained using a very minimal number of corrective-traces. Secondly, for nest images these corrections were always traces of background pixels (e.g. class 2). Thirdly, it was difficult to train the test classifiers to properly segment all the slices in the representative stack (e.g. slice 1 and 8). Finally, the out of bag error could not be relied on to reflect the quality of image segmentations.

7.6.6 Summary of classifier tests

The lessons gained from empirical analysis and test classifier performance tests contributed greatly to the training methods adopted for monitoring images. There are no standard procedures that can be ap-

```
-----  
5 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%  
-----  
10 CF2 Training input:  
# of pixels selected as class 1: 44  
# of pixels selected as class 2: 864  
Creating training data took: 34ms  
Training classifier...  
15 FastRandomForest of 50 trees, each constructed while considering 8 random  
features.  
Out of bag error: 0.661%  
-----  
20 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.

plied. Each machine learning model is tuned to perform well on a specific task. Also, there are hundreds of automatic and semi-automatic segmentation algorithms that have been developed since the advent of digital images. 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 and tests confirmed they are easy to train, tune and apply.

The variability of active nest images presents a challenge for *any* classification model. Monitoring photographs were inconsistent and diverse. The same active nest monitored over time could change in colour and texture as 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 in order to gain the best possible performance of the RF classifier for this analysis *and* so training could be *closely replicated* by others, some general rules for training were adopted. The term closely replicated comes with a caveat. The semi-supervised training method cannot actually be repeated, since 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 which 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 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 contribute to final model performances. The interaction between training features cannot be fully analysed since 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 that 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 since they

are much easier threshold. Statistical region merging performed well segmenting images of horizontal nests. However 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 TWS tests also demonstrated the tools and methods to test, train and apply machine algorithms are freely available in software platforms such as TWS and Weka. TWS is designed to use human knowledge in the segmentation process to improve the accuracy of the labelled regions. The algorithms in TWS have been developed for natural and medical images but 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 Section.

7.7 MONITORING CLASSIFIERS

Typically 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 [152]. 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, substantially improving on previous analyses [21, 16, 27, 25, 24]. Additionally, any obvious processing errors could be quickly rectified and the batch classifications re-run.

7.7.1 *Variability of training images*

Four training stacks were collated for each site. To simplify and speed up the training process each site-grid classifier was trained separately using respective stacks. This method also improved overall classifications since models could be constructed on a sub-sample of variable images and 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*

Before the final post-processing operations were applied to the classified stack of monitoring images, 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 monitoring images and against the manual field counts. It is important to point out, the post-processing operations were tuned specifically for the segmented monitoring-images that were 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. But, there was no attempt to tune the post-processing operators for image-segmentations that were 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 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; and they were considerably better than the classical methods tested. When the RF was compared against other machine-learners, classifier tests and benchmark analysis did not necessarily preclude the suitability of other models. 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 and 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. There were insufficient data for inter-observer correlations between the manual-field counts by two observers since 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:seconds. Separate collections were therefore comprised of different images; each single one was acquired under varying natural conditions. During preliminary analysis, 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, were used to visually investigate discrepancies. It was found that in most cases the exposure and illumination varied considerably between image collections - i.e. the minute:second sequences. 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 Manual-field, image and automated nest counts

Lin's concordance correlation coefficient was used to determine the level of agreement, accuracy and precision between the number of nests counted by different methods [149]. The most important of these comparisons were between the number of nests counted in the field, which were the *gold-standard* measures, with the automatic counts derived from the CF classifier. It is important to highlight the measurement cannot be used to judge the *correctness* of the separate methods in isolation, only the degree to which the different methods agreed with each other [?]. In addition to this, a descriptive scale for values of the concordance correlation coefficients has not been universally accepted [?]. McBride provided some guidelines for the measures of agreements (of continuous variables) for laboratory applications. He stresses the criteria have not been widely tested but suggests in order to assess the level of agreements in water testing applications, the lower one-sided 95% confidence limit for the calculated correlation of concordance should be compared to the scales outlined in 7.3.

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 below. 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 ($C_b = 0.973$). Automated counts ($n = 1284$) derived by thresholding (a-th) and clas-

Table 7.3: Guidelines for the strength of agreements

0.00 - 0.65	Poor
0.65 - 0.80	Moderate
0.80 - 0.90	Substantial
0.99 - 1.00	Almost perfect

sification (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 [? ?]. The counts derived from the a-th method were not expected to be reasonable since 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 ($C_b = 0.970$).

More research would be required before substantial claims could be made about the reliability and reproducibility of the automated image-centric monitoring method over the manual-field method. For example, although nest counts were collected by two observers most years, this was only on the first day of monitoring. As a result, there was insufficient data to properly determine the inter-observer variability for manual-field nest counts ($n = 60$). Also, there are few analyses concerning the performance of automated image-analysis over manual-visual methods reported in literature. Most analyses originate from biomedical research, and although broadly relevant, the methods do not compare well with those outlined in this research (e.g. microscopic imaging methods). Nevertheless, some plant sciences research methods were closely related to those used in this thesis. Similar results have been reported for inter-observer variability between visual scorers assessing plant diseases [?]. In summary however, these results indicated the automated a-CF counts were in closer agreement with the true nest count values, than those estimated from images.

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

Analysis	Method 1	Method 2	n	r	ρ_c	C_b	Rank
A2	m-image ob1	m-image ob2	170	0.891	0.867	0.973	1
A5	m-field	a-CF	520	0.828	0.738	0.891	2
A4	m-image	a-CF	170	0.705	0.679	0.963	3
A3	m-field	m-image	170	0.641	0.622	0.970	4
A1	a-ths	a-CF	1284	0.244	0.040	0.164	5

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

Table 7.5: Decline in the number of active nests (%). The upper and lower percentages are shown for automatic (a-CF) and manual field (m-field) methods, benchmarked against 2010 nest counts. The trends show a decrease in the number of active nests (S1-S3) between the years 2010–2013 and a slight rise in 2014 (marked *)

Site	Method	2011	2012	2013	2014
S1	a-CF	40 - 46	44 - 48	55 - 57	*4 - 15
	m-field	31 - 32	35 - 39	49 - 48	39 - 46
S2	a-CF	34	65 - 65	41 - 42	26 - 36
	m-field	27	54 - 55	27 - 29	*38 - 39
S3	a-CF		67 - 69	65 - 69	48 - 59
	m-field		37 - 36	35 - 36	4 - 13

Table key

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

Images segmented by monitoring classifier CF a-CF

Percentage increase *

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, and were not normally distributed [? ?]. They were not transformed for statistical analysis. 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 and 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 [11, 35, 153].

When the nest count results taken over five years were examined the trends showed a moderate to marked decreases. The percentage change in nest counts for each location was benchmarked against the first years monitoring data, for sites 1-2 and the second years data for site 3. The percentage declines varied between 26–79% at all three locations as shown in Table 7.5. There was not enough data collected in 2014 ($n_{days} = 4$), to reach any substantial conclusions about an upward trend in the number of active nests, although the graphical analysis does suggest a slight increase in nest numbers.

7.9 PERFORMANCE-COST MEASURES

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 comparatively limited capabilities of the desktop computer that was used for image analysis tasks. 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 below.

However, classifier optimisations also had a marked impact on processing resources and times. This included the working memory that was required during classifier training sessions, as well as resources that were necessary when the final monitoring-classifiers were applied. The processing capabilities of the desktop computer, also had an equally reciprocal impact on classifier optimisations. In part, tuning was a requirement. It was impossible to process a large features stack with the desktop without causing out of memory exceptions.

From initial analysis of classifier settings using the representative stack of six images, a range of feature importances were investigated. Of those tested, Gaussian blur, Mean, Minimum, Median, and Struc-

*Manual labour,
hardware
performance and
processing
requirements.*

tures filters produced sufficient segmentations. Moreover, they were all shown to have importances that contributed to final classifications. The number of σ was reduced from the default setting $\sigma = 1 - 16$ to $\sigma = 2$ which also reduced processing expenditure. Combined with the raw RGB, Hue, Saturation and Brightness images, a total of 20 features were used for training the final monitoring-classifiers. Monitoring images were processed in a single day and classifications could be easily repeated if/when required.

7.9.1 *Operating system environment*

During previous analysis there were significant memory problems during processing. This problem normally surfaced as out of memory exceptions that were thrown during TWSs. Some of the early classifications required days to process. A range of solutions were tried, such as increasing RAM memory and reducing the size of image stacks. Although most solutions helped a little, the performance of processing noticeably improved when the operating system was changed over to Linux (early 2013). Fiji and Weka are cross-platform, however the performance of both software packages were noticeably enhanced under Linux. The operating system was more versatile and was easily configured for the specific imaging tasks. The memory leak problems that occurred during previous processing and analyses were entirely mitigated during final classifications. Ubuntu swap memory was turned off during image processing tasks to avoid swapping processes out of physical memory. Virtual memory options were also passed to JavaVM from Fiji's main configuration file to increase the memory heap size. Combined these adjustments improved the stability of the operating system, software performance and final classification processing times.

7.10 IMAGING PIPELINE DESIGN

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

Figure 7.13: Image analysis work-flow.

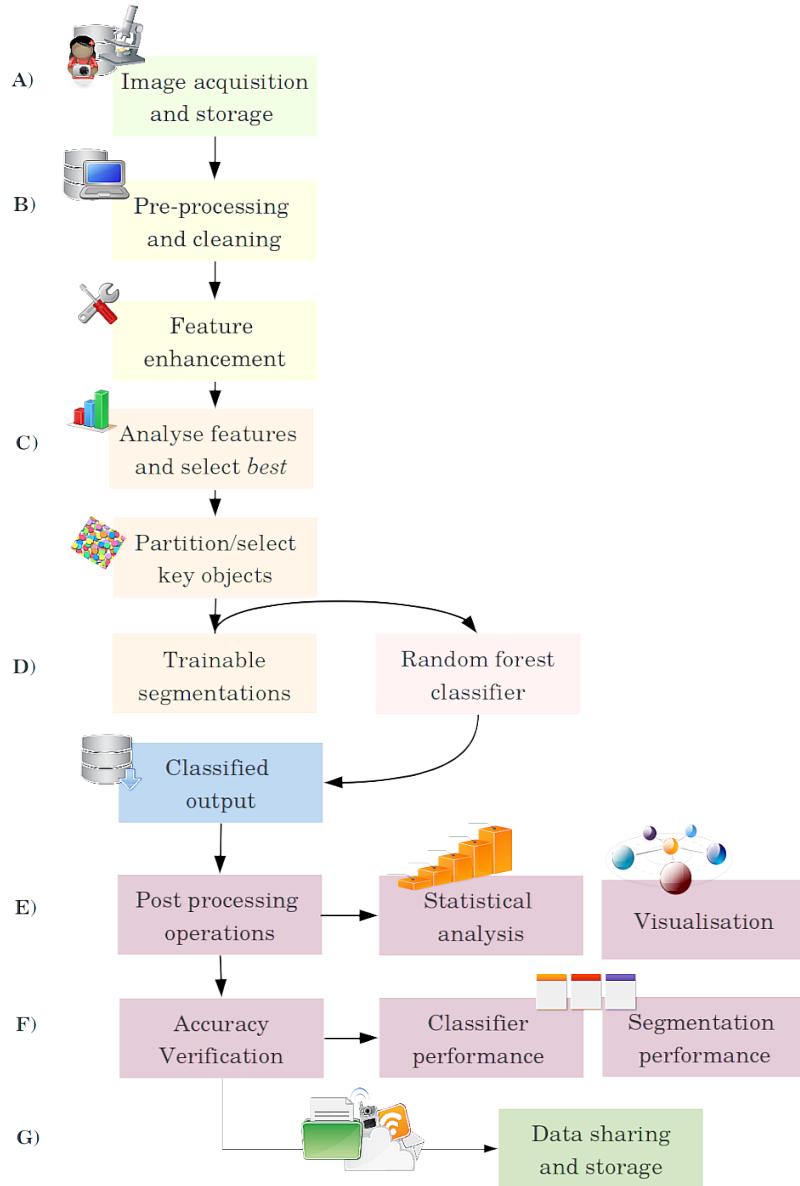


Table 7.6: Image pipeline tasks and tools.

Stage and task	Tools
<i>A) Acquisition:</i>	
Copying, sorting, storing and sharing data.	Transfer from SD card to PC – Linux terminal bash command [154]. Store on partition hard-drive and external drive [154]. Share Github data repository – see stage G.) [155, 156].
<i>B-C) Preprocessing:</i>	
Rename by EXIF data.	pyRenamer – mass file re-namer for GNOME [157]. Exif tool – read, write and edit meta information [144].
Crop to region of interest.	Xnview – viewing, converting, organising and editing raster images [112].
Collate, adjust parameters and review information.	Fiji – image processing workbench [92].
<i>D) Classify:</i>	
Features and classification.	Fiji TWS plugin – machine learning algorithms and selected image features for pixel-based segmentations. [92].
<i>E) Post processing:</i>	
Enhance for segmentation.	Fiji – morphological binary operations such as erosion, dilatation and thinning to remove isolated pixels [92].
Count segmented areas.	Fiji AP plugin – 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. [158].
<i>F-G) Verification:</i>	
Real-time logs.	Fiji TWS plugin – runtime logs of performance measures [92].
Compare classifiers.	The Weka workbench – a collection of machine learning algorithms for data mining tasks. [63, 78].
Visualisation.	R-software environment for statistical computing and graphics.[159]
Sharing.	GitEye – GUI tools for committing (git-gui) and browsing (gitk) [160].

7.11 SUMMARY OF DISCUSSIONS

The methods used for biological data collections are central to many studies but the validity of a monitoring method for native bees, which uses the number of active nests in a community to estimate their populations was not tested. Additionally, the field methods that were used in this research would not meet the requirements necessary for rigorous scientific analysis. Nevertheless, three communities of native bees were monitored over five years. The number of active nests were manually counted and images of the nests were collected for analysis. 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 and 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 which were not easily overcome. Finally, data management and protection was another factor that had an impact on the outcomes of this research. Lightning strikes were the most likely cause of a fatal hardware failure in 2012; the incident resulted in the loss of some key image data. Combined these issues have had an impact on the design of the final image-centric monitoring system presented in this thesis.

Despite the challenges, 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 learning process and 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. Furthermore, while Fiji and the TWS workbench have been invaluable tools in this research they were designed for biomedical applications. Consequently, there were few general procedures to follow and 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. However the techniques outlined were developed and refined over many hours of empirical tests to provide a good platform for future developments.

Segmentation metrics are frequently used to benchmark the final performance of automated image analysis systems. However, these metrics were not needed to test the ultimate performance of the image-centric nest monitoring system. The number of actual active nests provided the benchmark and all other data were compared to these counts. Five analyses were conducted to compare the differences between manual-field, manual-image and automated counts derived from classical threshold methods, or from the final monitoring classifier, CF. More research would be required before substantial claims could be made about the reliability and reproducibility of the automated image-centric monitoring method over the manual-field method. For example, although nest counts were collected by two observers most years, this was only on the first day of monitoring. As a result, there was insufficient data to properly determine the inter-observer variability for manual-field nest counts ($n = 60$). Also, there are few analyses concerning the performance of automated image-analysis over manual-visual methods reported in literature. Most analyses originate from biomedical research, and although broadly relevant, the methods do not compare well with those outlined in this research (e.g. microscopic imaging methods). Nevertheless, some plant sciences research methods were closely related to those used in this thesis. Similar results have been reported for inter-observer variability between visual scorers assessing plant diseases [?]. In summary, the results indicated 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 showed good accuracy ($C_b = 0.970$).

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 and 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 [11, 35, 153]. When the nest count results taken over five years were examined the trends showed a moderate to marked decreases. The percentage change in nest counts for each location was benchmarked against the first years monitoring data, for sites 1-2 and the second years data for site 3. The percentage declines varied between 26–79% at all three locations. There was not enough data collected in 2014 to form substantial conclusions, however the graphical analysis did suggest there is a slight upward trend in nest numbers.

8

CONCLUSIONS

OUTLINE

Part IV
APPENDIX

A

ACRONYMS

AP	Analyse particles
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 camera
EC	Enhance contrast
EH	Equalise histogram
EXIF	Exchangeable image file format
Fiji	Fiji is just Imagej
GIS	Geographical information systems
GPS	Global positioning system
GUI	Graphical user interface
HD	Hard drive
HSI	Hue saturation and intensity
J2P	JPEG 2000
JPEG	Joint photographic experts group
KNB	Knowledge network for bio-complexity
KNIME	Konstanz information miner
ODG	Open document graphics
ODP	Open document presentation

ODS	Open document spreadsheet
ODT	Open document text
OOB	Out of bag
PA	Process all
PC	Personal computer
RF	Random forest
RGB	Red green and blue
ROI	Regions of interest
SD	Secure digital device
SEH	Stack equalise histogram
SLR	Single lens reflex
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 + Dilate, smooths objects contours, removes isolated elements and breaks thin connections.
Close	Dilate + Erode, smooths objects contours, fill small holes and joins breaks.

C

NOTATIONS

CR	Compression ratio = $\frac{\text{original image data volume}}{\text{compressed image data volume}}$	33
\mathcal{A}	A supervised learning algorithm	??
$\mathcal{A}(\theta, \mathcal{L})$	The model $\varphi_{\mathcal{L}}$ produced by algorithm \mathcal{A} over \mathcal{L} and hyper-parameters θ	??
α_s	The proportion of samples in a random patch	??
α_f	The proportion of features in a random patch	??
b_l	The l-th value of a categorical variable	??
B	A subset $B \subseteq V$ of variables	??
c_k	The k-th class	??
C_p^k	The number of k-combinations from a set of p elements	??
$C(N)$	The time complexity for splitting N samples	??
\mathbb{E}	Expectation	
$\bar{E}(\varphi_{\mathcal{L}}, \mathcal{L}')$	The average prediction error of $\varphi_{\mathcal{L}}$ over \mathcal{L}'	??
$\text{Err}(\varphi_{\mathcal{L}})$	The generalization error of $\varphi_{\mathcal{L}}$??, ??

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	o
Colour Label	o

Image

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

Camera

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

CODE

```

1 //Contrast enhancement by saturated pixels
raw="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/enhance_contrast/raw
    /"; //source and output dir
slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/enhance_contrast/
    slice/";
stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/enhance_contrast/
    stack/";
pre_process="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/
    enhance_contrast/pre_process/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/enhance_contrast/
    /montage/";
docf= "/home/n/ALHimgdata/301_thesis/thesis/gfx/chp3/"; //for publishing
    images
setBatchMode(true);
//open image and saturate pixels by 0.4, 40 and 90%
open(raw + "sc.tif");//scheme A
11     //makeRectangle(154, 85, 30, 30);
        run("Enhance_Contrast...", "saturated=0.4_update");
            saveAs("TIFF", slice +"s1.tif");
open(raw + "sc.tif");//scheme B
16     //makeRectangle(154, 85, 30, 30);
        run("Enhance_Contrast...", "saturated=40_update");
            saveAs("TIFF", slice +"s2.tif");
open(raw + "sc.tif");//scheme C
21     //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);
26     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);
31     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=/home/n/ALHimgdata/300_documents/
            ch3_imgs/tests/enhance_contrast/slice/0_0hist.tif_sort");
            saveAs("Tiff", stack + "ecs.tif");
open(stack + "ecs.tif");
36     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.1: Saturated pixels examples.

```

1 //Contrast enhancement for image stacks
raw="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/pre_processing/raw/"
; //source and output dir
slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/pre_processing/
slice/";
stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/pre_processing/
stack/";
pre_process="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/pre_processing/
pre_processing/pre_process/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/pre_processing/
montage/";
docf= "/home/n/ALHimgdata/301_thesis/thesis/gfx/chp3/"; //for publishing
images
setBatchMode(true);
//open image and apply contrast using 5 schemes
open(stack + "sc.tif");
11           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
16           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");
21 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
");
26           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_mssec.tif"); //
31 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_mssec.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_mssec.tif");
36 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_mssec.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_mssec.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_mssec.tif");
46 //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");
51 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");
56 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
61     run("Image_Sequence...", "open=/home/n/ALHimgdata/300_documents/
        ch3_imgs/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 + "1mssec.tif"); saveAs("PNG",
            docf + "mssec.png");
66 setBatchMode(false);run("Close_All");//end
71

```

Listing E.2: Stack pre-processing examples.

```

//Try all thresholds (16 methods)
raw="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/tryall/raw/"; //
    source and output dir
slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/tryall/slice/";
4 stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/tryall/stack/";
pre_process="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/tryall/
    pre_process/";
logf="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/tryall/log/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/tryall/montage/"
    ;
docf= "/home/n/ALHimgdata/301_thesis/thesis/gfx/chp3/"; //for publishing
    images
9 //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");
14 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");
19 selectWindow("Log");
    saveAs("Text", logf + "Log-tryall.txt");
setBatchMode(false);run("Close_All");//end

```

Listing E.3: Thresholding tests.

```

/Automatic thresholding on an image of a horizontal ground nest by six
    schemes,
//including some typical post processing operation examples.
3 //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="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/t_six/raw/";
slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/t_six/slice/";
8 stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/t_six/stack/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/t_six/montage/";
post_process="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/t_six/
    post_process/";
logf="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/t_six/log/";
docf= "/home/n/ALHimgdata/301_thesis/thesis/gfx/chp3/"; //for publishing
    images
13 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");
18 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");
23     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");
28     saveAs("Tiff", slice + "5_Minimum.tif");
open(raw + "5-gs.tif");
    run("Auto_Threshold", "method=Otsu_show");
        saveAs("Tiff", slice + "6_Otsu.tif");
33 selectWindow("Log");
    saveAs("Text", logf + "Log-thresh-six.txt");
//post-process binary results
run("Image_Sequence...", "open=/home/n/ALHimgdata/300_documents/ch3_imgs/
    tests/t_six/slice/1-Default.tif_sort");

```

```

    saveAs("Tiff", stack + "t-six.tif");
38 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");
48 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");
53 //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");
58 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");
63 setBatchMode(false);run("Close_All");//end
68

```

Listing E.4: Thresholding by six common methods.

```

1 //Binirization tests on representative sample images of active nests.
raw="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/challenge/raw/";
slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/challenge/slice/";
stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/challenge/stack/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/challenge/
    montage/";

```

```

6 | post_process="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/challenge/
|   post_process";
| logf="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/challenge/log/";
| docf= "/home/n/ALHimgdata/301_thesis/thesis/gfx/chp3/";
| setBatchMode(true); //do not display images
| //open and process stack
11 | 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");
16 | 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");
21 | open(stack + "fh.tif");//close operator
|   run("Options...", "iterations=2_count=4_pad_do=Close_stack");
|     saveAs("Tiff", stack + "close-43.tif");
26 | 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");
31 | //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");
36 | 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");
41 | 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");
46 |     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");
51   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.5: Otsu thresholding on variable nest images.

```

//Edge detection by Canny-Deriche filtering
2 raw="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/edge/raw/";//home
      dir
slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/edge/slice/";
stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/edge/stack/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/edge/montage/";
post_process="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/edge/
      post_process/";
7 logf = "/home/n/ALHimgdata/300_documents/ch3_imgs/tests/edge/log/";
docf= "/home/n/ALHimgdata/301_thesis/thesis/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");
12      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");
17 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");
22 open(raw + "5-gs.tif");
      run("Deriche...", "alpha=0.15"); run("8-bit");
      saveAs("Tiff", slice + "6.tif");
open("Close_All");
//post-process binary results
run("Image_Sequence...", "open=/home/n/ALHimgdata/300_documents/ch3_imgs/
      tests/edge/slice/1.tif_sort");
      saveAs("Tiff", stack + "edge.tif");
32 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");
37 open(stack + "min.tif");
      run("Options...", "iterations=3_count=3_do=Close_stack");//close
      operator

```

```

        saveAs("Tiff", stack + "close-33.tif");
42    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");
47    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
52    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");
67    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");
72    setBatchMode(false);run("Close_All");//end

```

Listing E.6: Segmentation by edge detection.

```

2 //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="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm/raw/";//home dir

```

```

7      slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm/slice/";
stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm/stack/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm/montage/";
logf = "/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm/log/";
docf= "/home/n/ALHimgdata/301_thesis/thesis/gfx/chp3/"; //for publishing
    images
//open and run srm with varying Q
open(raw + "5-gs.tif");
12     run("Statistical_Region_Merging", "q=1_showaverages");
        saveAs("Tiff", slice + "1.tif");
open(raw + "5-gs.tif");
16     run("Statistical_Region_Merging", "q=2_showaverages");
        saveAs("Tiff", slice + "2.tif");
open(raw + "5-gs.tif");
17     run("Statistical_Region_Merging", "q=3_showaverages");
        saveAs("Tiff", slice + "3.tif");
open(raw + "5-gs.tif");
21     run("Statistical_Region_Merging", "q=6_showaverages");
        saveAs("Tiff", slice + "4.tif");
open(raw + "5-gs.tif");
25     run("Statistical_Region_Merging", "q=8_showaverages");
        saveAs("Tiff", slice + "5.tif");
open(raw + "5-gs.tif");
27     run("Statistical_Region_Merging", "q=16_showaverages");
        saveAs("Tiff", slice + "6.tif");
run("Close_All");
//post-process binary results
run("Image_Sequence...", "open=/home/n/ALHimgdata/300_documents/ch3_imgs/
    tests/srm/slice/1.tif_sort");
32     saveAs("Tiff", stack + "srm.tif");
open(stack + "srm.tif");
36     run("8-bit");
        run("Make_Binary", "Minimum_background=Default"); //invert
            binary
        saveAs("Tiff", stack + "min.tif");
open(stack + "min.tif");
37     run("Options...", "iterations=1_count=1_do=Open_stack");//open
            operator
        saveAs("Tiff", stack + "open-11.tif");
selectWindow("open-11.tif");
38     run("Options...", "iterations=1_count=1_do=[Fill_Holes]_stack");
            //fill holes
        saveAs("Tiff", stack+ "fh.tif");
selectWindow("fh.tif");
42     run("Options...", "iterations=1_count=1_pad_do=Close_stack");//
            close operator
        saveAs("Tiff", stack + "close-11.tif");
selectWindow("close-11.tif");
47     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");
48     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 + "srn.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");
62 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");
67 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.7: Segmentation by region merging.

```

//Statistical region merging on representative 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
4 raw="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm_all/raw/";//home
    dir
slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm_all/slice/";
stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm_all/stack/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm_all/montage/
    ";
logf = "/home/n/ALHimgdata/300_documents/ch3_imgs/tests/srm_all/log/";
9 docf= "/home/n/ALHimgdata/301_thesis/thesis/gfx/chp3/"; //for publishing
    images
setBatchMode(true);//do not display images
//open and run srm on representative images
open(raw + "1.tif");
    run("Statistical_Region_Merging", "q=16_showaverages");
14     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");
19     saveAs("Tiff", slice + "3.tif");
open(raw + "4.tif");

```

```

    run("Statistical_Region_Merging", "q=16_showaverages");
    saveAs("Tiff", slice + "4.tif");
24 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");
29 run("Close_All");
//post-process binary results
run("Image_Sequence...", "open=/home/n/ALHimgdata/300_documents/ch3_imgs/
    tests/srm_all/slice/1.tif_sort");
    saveAs("Tiff", stack + "srm_all.tif");
34 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");
44 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");
49 //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 + "1msrsmall.png");
54 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 +
        "2msrsmall.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");
59     saveAs("Tiff", montage + "open.tif");saveAs("PNG", docf +
        "3msrsmall.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 +
        "4msrsmall.png");
open(stack + "close-24_all.tif");

```

```

64     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");
69 setBatchMode(false);run("Close_All");//end

```

Listing E.8: Region merging on variable nest images..

```

1 //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="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/rf/raw/";
slice="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/rf/slice/";
5 stack="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/rf/stack/";
montage="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/rf/montage/";
post_process="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/rf/
      post_process";
logf="/home/n/ALHimgdata/300_documents/ch3_imgs/tests/rf/log/";
docf= "/home/n/ALHimgdata/301_thesis/thesis/gfx/chp3/";
11 //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");
16 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");
21 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");
26 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");

```

```

31 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");
36 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");
41 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
46 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
51 //post process
run("Collect_Garbage"); setBatchMode(true);
open(stack + "rf.tif"); //convert to 8 bit
    run("8-bit");
        saveAs("Tiff", stack + "rf8bit.tif");
56 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");
61 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");
66 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");
71 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 + "mrffopen-22.tif"); saveAs("PNG",
  docf + "3mrf.png");
86 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.9: Trainable segmentation on variable nest images.

```

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

mic_mfc_ac_ob <-
  read.table("/home/n/ALHimgdata/201_analysis/analysis/main/r/mic_mfc_ac_
  ob/csv/mic_mfc_ac_ob_4R.csv",
            header=TRUE, sep=",", na.strings="NA", dec=".",
            strip.white=
            TRUE)

```

```

15 ## 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")

20 ## 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)
25 rslts <- acmc.ccc$rho.c
rslts1 <- as.data.frame(rslts, row.names = NULL, responseName = "Rho",
  stringsAsFactors = TRUE)
kable(head(rslts1[,1:3]), format = "markdown")

lab <- paste("CCC: ", round(acmc.ccc$rho.c[,1], digits = 2), "(95% CI"
30 ,
  round(acmc.ccc$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.10: Automatic and manual concordance correlation.

```

##Spearman rank correlation matrix table obs1 obs2
2 ##Lin's Concordance correlation
library(knitr)
library(car)
library(rgl, pos=13)
library(nlme, pos=14)
7 library(mgcv, pos=14)
library(survival)
library(epiR)
library(methods)
library(ggplot2)

12 mic_mfc_ac_ob <-
  read.table("/home/n/ALHimgdata/201_analysis/analysis/r/mic_mfc_ac_ob/
    csv/mic_mfc_ac_ob_4R.csv",
    header=TRUE, sep=",", na.strings="NA", dec=".",
    strip.white=TRUE)

17 ## 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")

22 ## 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")

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

32 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 = "")

37 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.11: Image counts by two observers.

RAW FIELD DATA

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
1	1	1	10	11	23	1	1	1	1
2	2	1	10	11	24	2	1	1	0
3	3	1	10	11	26	3	1	1	1
4	4	1	10	11	28	4	1	1	16
5	5	1	10	11	29	5	1	1	13
6	6	1	10	11	30	6	1	1	29
7	7	1	10	12	1	7	1	1	16
8	8	1	10	12	2	8	1	1	15
9	9	1	10	12	5	9	1	1	2
10	10	1	10	12	6	10	1	1	3
11	11	1	10	12	8	11	1	1	1
12	12	1	10	12	9	12	1	1	2
13	13	1	10	12	10	13	1	1	5
14	14	1	10	12	11	14	1	1	1
15	15	1	10	12	12	15	1	1	6
16	16	1	10	12	13	16	1	1	20
17	17	1	10	11	23	1	1	2	1
18	18	1	10	11	24	2	1	2	2
19	19	1	10	11	26	3	1	2	0
20	20	1	10	11	28	4	1	2	25
21	21	1	10	11	29	5	1	2	2
22	22	1	10	11	30	6	1	2	30
23	23	1	10	12	1	7	1	2	17
24	24	1	10	12	2	8	1	2	16
25	25	1	10	12	5	9	1	2	4
26	26	1	10	12	6	10	1	2	10
27	27	1	10	12	8	11	1	2	2
28	28	1	10	12	9	12	1	2	12
29	29	1	10	12	10	13	1	2	12
30	30	1	10	12	11	14	1	2	1

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
31	31	1	10	12	12	15	1	2	10
32	32	1	10	12	13	16	1	2	1
33	33	1	10	11	23	1	1	3	1
34	34	1	10	11	24	2	1	3	40
35	35	1	10	11	26	3	1	3	1
36	36	1	10	11	28	4	1	3	30
37	37	1	10	11	29	5	1	3	20
38	38	1	10	11	30	6	1	3	25
39	39	1	10	12	1	7	1	3	2
40	40	1	10	12	2	8	1	3	2
41	41	1	10	12	5	9	1	3	2
42	42	1	10	12	6	10	1	3	10
43	43	1	10	12	8	11	1	3	5
44	44	1	10	12	9	12	1	3	1
45	45	1	10	12	10	13	1	3	4
46	46	1	10	12	11	14	1	3	1
47	47	1	10	12	12	15	1	3	20
48	48	1	10	12	13	16	1	3	21
49	49	1	10	11	23	1	1	4	3
50	50	1	10	11	24	2	1	4	50
51	51	1	10	11	26	3	1	4	0
52	52	1	10	11	28	4	1	4	30
53	53	1	10	11	29	5	1	4	12
54	54	1	10	11	30	6	1	4	10
55	55	1	10	12	1	7	1	4	0
56	56	1	10	12	2	8	1	4	2
57	57	1	10	12	5	9	1	4	2
58	58	1	10	12	6	10	1	4	5
59	59	1	10	12	8	11	1	4	5
60	60	1	10	12	9	12	1	4	14
61	61	1	10	12	10	13	1	4	4
62	62	1	10	12	11	14	1	4	5
63	63	1	10	12	12	15	1	4	1
64	64	1	10	12	13	16	1	4	2
65	65	2	10	11	23	1	1	1	1
66	66	2	10	11	24	2	1	1	0

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
67	67	2	10	11	25	3	1	1	1
68	68	2	10	11	26	4	1	1	2
69	69	2	10	11	28	5	1	1	1
70	70	2	10	11	29	6	1	1	3
71	71	2	10	11	30	7	1	1	2
72	72	2	10	12	1	8	1	1	2
73	73	2	10	12	2	9	1	1	2
74	74	2	10	12	5	10	1	1	1
75	75	2	10	12	6	11	1	1	0
76	76	2	10	12	8	12	1	1	1
77	77	2	10	12	9	13	1	1	1
78	78	2	10	12	10	14	1	1	1
79	79	2	10	12	11	15	1	1	0
80	80	2	10	12	12	16	1	1	1
81	81	2	10	11	23	1	1	2	1
82	82	2	10	11	24	2	1	2	15
83	83	2	10	11	25	3	1	2	5
84	84	2	10	11	26	4	1	2	2
85	85	2	10	11	28	5	1	2	3
86	86	2	10	11	29	6	1	2	1
87	87	2	10	11	30	7	1	2	2
88	88	2	10	12	1	8	1	2	7
89	89	2	10	12	2	9	1	2	6
90	90	2	10	12	5	10	1	2	5
91	91	2	10	12	6	11	1	2	1
92	92	2	10	12	8	12	1	2	2
93	93	2	10	12	9	13	1	2	1
94	94	2	10	12	10	14	1	2	3
95	95	2	10	12	11	15	1	2	4
96	96	2	10	12	12	16	1	2	4
97	97	2	10	11	23	1	1	3	8
98	98	2	10	11	24	2	1	3	13
99	99	2	10	11	25	3	1	3	13
100	100	2	10	11	26	4	1	3	10
101	101	2	10	11	28	5	1	3	1
102	102	2	10	11	29	6	1	3	7

continued ...

mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
103	103	2	10	11	30	7	1	3
104	104	2	10	12	1	8	1	3
105	105	2	10	12	2	9	1	3
106	106	2	10	12	5	10	1	3
107	107	2	10	12	6	11	1	3
108	108	2	10	12	8	12	1	3
109	109	2	10	12	9	13	1	3
110	110	2	10	12	10	14	1	3
111	111	2	10	12	11	15	1	3
112	112	2	10	12	12	16	1	3
113	113	2	10	11	23	1	1	4
114	114	2	10	11	24	2	1	4
115	115	2	10	11	25	3	1	4
116	116	2	10	11	26	4	1	4
117	117	2	10	11	28	5	1	4
118	118	2	10	11	29	6	1	4
119	119	2	10	11	30	7	1	4
120	120	2	10	12	1	8	1	4
121	121	2	10	12	2	9	1	4
122	122	2	10	12	5	10	1	4
123	123	2	10	12	6	11	1	4
124	124	2	10	12	8	12	1	4
125	125	2	10	12	9	13	1	4
126	126	2	10	12	10	14	1	4
127	127	2	10	12	11	15	1	4
128	128	2	10	12	12	16	1	4
129	129	1	11	11	11	1	1	1
130	130	1	11	11	12	2	1	1
131	131	1	11	11	13	3	1	1
132	132	1	11	11	14	4	1	1
133	133	1	11	11	15	5	1	1
134	134	1	11	11	16	6	1	1
135	135	1	11	11	17	7	1	1
136	136	1	11	11	18	8	1	1
137	137	1	11	11	21	9	1	1
138	138	1	11	11	22	10	1	40

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
139	139	1	11	11	23	11	1	1	0
140	140	1	11	11	24	12	1	1	20
141	141	1	11	11	25	13	1	1	1
142	142	1	11	11	28	14	1	1	3
143	143	1	11	12	7	15	1	1	15
144	144	1	11	12	9	16	1	1	2
145	145	1	11	11	11	1	1	2	12
146	146	1	11	11	12	2	1	2	10
147	147	1	11	11	13	3	1	2	15
148	148	1	11	11	14	4	1	2	3
149	149	1	11	11	15	5	1	2	10
150	150	1	11	11	16	6	1	2	5
151	151	1	11	11	17	7	1	2	2
152	152	1	11	11	18	8	1	2	2
153	153	1	11	11	21	9	1	2	2
154	154	1	11	11	22	10	1	2	12
155	155	1	11	11	23	11	1	2	2
156	156	1	11	11	24	12	1	2	2
157	157	1	11	11	25	13	1	2	4
158	158	1	11	11	28	14	1	2	7
159	159	1	11	12	7	15	1	2	7
160	160	1	11	12	9	16	1	2	1
161	161	1	11	11	11	1	1	3	40
162	162	1	11	11	12	2	1	3	0
163	163	1	11	11	13	3	1	3	2
164	164	1	11	11	14	4	1	3	3
165	165	1	11	11	15	5	1	3	3
166	166	1	11	11	16	6	1	3	2
167	167	1	11	11	17	7	1	3	3
168	168	1	11	11	18	8	1	3	1
169	169	1	11	11	21	9	1	3	1
170	170	1	11	11	22	10	1	3	1
171	171	1	11	11	23	11	1	3	1
172	172	1	11	11	24	12	1	3	1
173	173	1	11	11	25	13	1	3	14
174	174	1	11	11	28	14	1	3	25

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
175	175	1	11	12	7	15	1	3	5
176	176	1	11	12	9	16	1	3	1
177	177	1	11	11	11	1	1	4	16
178	178	1	11	11	12	2	1	4	2
179	179	1	11	11	13	3	1	4	1
180	180	1	11	11	14	4	1	4	4
181	181	1	11	11	15	5	1	4	4
182	182	1	11	11	16	6	1	4	3
183	183	1	11	11	17	7	1	4	3
184	184	1	11	11	18	8	1	4	2
185	185	1	11	11	21	9	1	4	2
186	186	1	11	11	22	10	1	4	2
187	187	1	11	11	23	11	1	4	2
188	188	1	11	11	24	12	1	4	2
189	189	1	11	11	25	13	1	4	2
190	190	1	11	11	28	14	1	4	3
191	191	1	11	12	7	15	1	4	1
192	192	1	11	12	9	16	1	4	1
193	193	2	11	11	11	1	1	1	3
194	194	2	11	11	12	2	1	1	1
195	195	2	11	11	13	3	1	1	8
196	196	2	11	11	14	4	1	1	8
197	197	2	11	11	15	5	1	1	1
198	198	2	11	11	16	6	1	1	6
199	199	2	11	11	17	7	1	1	1
200	200	2	11	11	18	8	1	1	1
201	201	2	11	11	21	9	1	1	1
202	202	2	11	11	22	10	1	1	1
203	203	2	11	11	23	11	1	1	1
204	204	2	11	11	24	12	1	1	1
205	205	2	11	11	25	13	1	1	0
206	206	2	11	11	28	14	1	1	6
207	207	2	11	12	7	15	1	1	3
208	208	2	11	12	9	16	1	1	11
209	209	2	11	11	11	1	1	2	7
210	210	2	11	11	12	2	1	2	1

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
211	211	2	11	11	13	3	1	2	1
212	212	2	11	11	14	4	1	2	6
213	213	2	11	11	15	5	1	2	1
214	214	2	11	11	16	6	1	2	1
215	215	2	11	11	17	7	1	2	2
216	216	2	11	11	18	8	1	2	1
217	217	2	11	11	21	9	1	2	4
218	218	2	11	11	22	10	1	2	1
219	219	2	11	11	23	11	1	2	5
220	220	2	11	11	24	12	1	2	1
221	221	2	11	11	25	13	1	2	1
222	222	2	11	11	28	14	1	2	1
223	223	2	11	12	7	15	1	2	1
224	224	2	11	12	9	16	1	2	1
225	225	2	11	11	11	1	1	3	2
226	226	2	11	11	12	2	1	3	5
227	227	2	11	11	13	3	1	3	1
228	228	2	11	11	14	4	1	3	1
229	229	2	11	11	15	5	1	3	0
230	230	2	11	11	16	6	1	3	2
231	231	2	11	11	17	7	1	3	2
232	232	2	11	11	18	8	1	3	2
233	233	2	11	11	21	9	1	3	3
234	234	2	11	11	22	10	1	3	3
235	235	2	11	11	23	11	1	3	1
236	236	2	11	11	24	12	1	3	3
237	237	2	11	11	25	13	1	3	4
238	238	2	11	11	28	14	1	3	3
239	239	2	11	12	7	15	1	3	1
240	240	2	11	12	9	16	1	3	4
241	241	2	11	11	11	1	1	4	1
242	242	2	11	11	12	2	1	4	0
243	243	2	11	11	13	3	1	4	5
244	244	2	11	11	14	4	1	4	5
245	245	2	11	11	15	5	1	4	3
246	246	2	11	11	16	6	1	4	4

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
247	247	2	11	11	17	7	1	4	3
248	248	2	11	11	18	8	1	4	5
249	249	2	11	11	21	9	1	4	10
250	250	2	11	11	22	10	1	4	12
251	251	2	11	11	23	11	1	4	3
252	252	2	11	11	24	12	1	4	4
253	253	2	11	11	25	13	1	4	6
254	254	2	11	11	28	14	1	4	1
255	255	2	11	12	7	15	1	4	1
256	256	2	11	12	9	16	1	4	1
257	257	3	11	11	11	1	1	1	1
258	258	3	11	11	12	2	1	1	1
259	259	3	11	11	13	3	1	1	1
260	260	3	11	11	14	4	1	1	1
261	261	3	11	11	15	5	1	1	0
262	262	3	11	11	16	6	1	1	0
263	263	3	11	11	17	7	1	1	0
264	264	3	11	11	18	8	1	1	2
265	265	3	11	11	21	9	1	1	5
266	266	3	11	11	22	10	1	1	1
267	267	3	11	11	23	11	1	1	1
268	268	3	11	11	24	12	1	1	1
269	269	3	11	11	28	13	1	1	3
270	270	3	11	12	7	14	1	1	0
271	271	3	11	12	9	15	1	1	2
272	272	3	11	12	20	16	1	1	2
273	273	3	11	11	11	1	1	2	3
274	274	3	11	11	12	2	1	2	5
275	275	3	11	11	13	3	1	2	5
276	276	3	11	11	14	4	1	2	6
277	277	3	11	11	15	5	1	2	4
278	278	3	11	11	16	6	1	2	5
279	279	3	11	11	17	7	1	2	6
280	280	3	11	11	18	8	1	2	5
281	281	3	11	11	21	9	1	2	15
282	282	3	11	11	22	10	1	2	7

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
283	283	3	11	11	23	11	1	2	1
284	284	3	11	11	24	12	1	2	4
285	285	3	11	11	25	13	1	2	8
286	286	3	11	11	28	14	1	2	5
287	287	3	11	12	7	15	1	2	3
288	288	3	11	12	9	16	1	2	7
289	289	3	11	11	11	1	1	3	2
290	290	3	11	11	12	2	1	3	1
291	291	3	11	11	13	3	1	3	7
292	292	3	11	11	14	4	1	3	3
293	293	3	11	11	15	5	1	3	0
294	294	3	11	11	16	6	1	3	2
295	295	3	11	11	17	7	1	3	1
296	296	3	11	11	18	8	1	3	5
297	297	3	11	11	21	9	1	3	5
298	298	3	11	11	22	10	1	3	2
299	299	3	11	11	23	11	1	3	2
300	300	3	11	11	24	12	1	3	2
301	301	3	11	11	25	13	1	3	1
302	302	3	11	11	28	14	1	3	10
303	303	3	11	12	7	15	1	3	7
304	304	3	11	12	9	16	1	3	10
305	305	3	11	11	11	1	1	4	8
306	306	3	11	11	12	2	1	4	0
307	307	3	11	11	13	3	1	4	10
308	308	3	11	11	14	4	1	4	14
309	309	3	11	11	15	5	1	4	10
310	310	3	11	11	16	6	1	4	5
311	311	3	11	11	17	7	1	4	1
312	312	3	11	11	18	8	1	4	2
313	313	3	11	11	21	9	1	4	0
314	314	3	11	11	22	10	1	4	10
315	315	3	11	11	23	11	1	4	1
316	316	3	11	11	24	12	1	4	5
317	317	3	11	11	25	13	1	4	1
318	318	3	11	11	28	14	1	4	15

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
319	319	3	11	12	7	15	1	4	1
320	320	3	11	12	9	16	1	4	15
321	321	1	12	11	9	1	1	1	0
322	322	1	12	11	10	2	1	1	5
323	323	1	12	11	11	3	1	1	3
324	324	1	12	11	12	4	1	1	4
325	325	1	12	11	13	5	1	1	10
326	326	1	12	11	14	6	1	1	2
327	327	1	12	11	15	7	1	1	2
328	328	1	12	11	16	8	1	1	1
329	329	1	12	11	18	9	1	1	4
330	330	1	12	11	19	10	1	1	1
331	331	1	12	12	4	11	1	1	23
332	332	1	12	12	11	12	1	1	6
333	333	1	12	12	12	13	1	1	10
334	334	1	12	11	9	1	1	2	1
335	335	1	12	11	10	2	1	2	5
336	336	1	12	11	11	3	1	2	2
337	337	1	12	11	12	4	1	2	2
338	338	1	12	11	13	5	1	2	20
339	339	1	12	11	14	6	1	2	1
340	340	1	12	11	15	7	1	2	5
341	341	1	12	11	16	8	1	2	15
342	342	1	12	11	18	9	1	2	5
343	343	1	12	11	19	10	1	2	10
344	344	1	12	12	4	11	1	2	17
345	345	1	12	12	11	12	1	2	36
346	346	1	12	12	12	13	1	2	40
347	347	1	12	11	9	1	1	3	1
348	348	1	12	11	10	2	1	3	1
349	349	1	12	11	11	3	1	3	1
350	350	1	12	11	12	4	1	3	2
351	351	1	12	11	13	5	1	3	2
352	352	1	12	11	14	6	1	3	1
353	353	1	12	11	15	7	1	3	1
354	354	1	12	11	16	8	1	3	1

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
355	355	1	12	11	18	9	1	3	1
356	356	1	12	11	19	10	1	3	0
357	357	1	12	12	4	11	1	3	22
358	358	1	12	12	11	12	1	3	7
359	359	1	12	12	12	13	1	3	5
360	360	1	12	11	9	1	1	4	1
361	361	1	12	11	10	2	1	4	1
362	362	1	12	11	11	3	1	4	1
363	363	1	12	11	12	4	1	4	0
364	364	1	12	11	13	5	1	4	1
365	365	1	12	11	14	6	1	4	1
366	366	1	12	11	15	7	1	4	1
367	367	1	12	11	16	8	1	4	0
368	368	1	12	11	18	9	1	4	1
369	369	1	12	11	19	10	1	4	1
370	370	1	12	12	4	11	1	4	3
371	371	1	12	12	11	12	1	4	12
372	372	1	12	12	12	13	1	4	10
373	373	2	12	11	9	1	1	1	2
374	374	2	12	11	10	2	1	1	2
375	375	2	12	11	11	3	1	1	2
376	376	2	12	11	12	4	1	1	2
377	377	2	12	11	13	5	1	1	1
378	378	2	12	11	14	6	1	1	2
379	379	2	12	11	15	7	1	1	2
380	380	2	12	11	16	8	1	1	2
381	381	2	12	11	18	9	1	1	2
382	382	2	12	11	19	10	1	1	0
383	383	2	12	12	4	11	1	1	3
384	384	2	12	12	11	12	1	1	4
385	385	2	12	12	12	13	1	1	4
386	386	2	12	11	9	1	1	2	1
387	387	2	12	11	10	2	1	2	1
388	388	2	12	11	11	3	1	2	1
389	389	2	12	11	12	4	1	2	1
390	390	2	12	11	13	5	1	2	2

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
391	391	2	12	11	14	6	1	2	2
392	392	2	12	11	15	7	1	2	2
393	393	2	12	11	16	8	1	2	2
394	394	2	12	11	18	9	1	2	2
395	395	2	12	11	19	10	1	2	0
396	396	2	12	12	4	11	1	2	5
397	397	2	12	12	11	12	1	2	5
398	398	2	12	12	12	13	1	2	5
399	399	2	12	11	9	1	1	3	2
400	400	2	12	11	10	2	1	3	2
401	401	2	12	11	11	3	1	3	1
402	402	2	12	11	12	4	1	3	1
403	403	2	12	11	13	5	1	3	1
404	404	2	12	11	14	6	1	3	1
405	405	2	12	11	15	7	1	3	0
406	406	2	12	11	16	8	1	3	1
407	407	2	12	11	18	9	1	3	0
408	408	2	12	11	19	10	1	3	0
409	409	2	12	12	4	11	1	3	2
410	410	2	12	12	11	12	1	3	2
411	411	2	12	12	12	13	1	3	2
412	412	2	12	11	9	1	1	4	2
413	413	2	12	11	10	2	1	4	2
414	414	2	12	11	11	3	1	4	2
415	415	2	12	11	12	4	1	4	2
416	416	2	12	11	13	5	1	4	2
417	417	2	12	11	14	6	1	4	2
418	418	2	12	11	15	7	1	4	0
419	419	2	12	11	16	8	1	4	0
420	420	2	12	11	18	9	1	4	1
421	421	2	12	11	19	10	1	4	0
422	422	2	12	12	4	11	1	4	3
423	423	2	12	12	11	12	1	4	5
424	424	2	12	12	12	13	1	4	3
425	425	3	12	11	9	1	1	1	1
426	426	3	12	11	10	2	1	1	5

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
427	427	3	12	11	11	3	1	1	5
428	428	3	12	11	12	4	1	1	3
429	429	3	12	11	13	5	1	1	4
430	430	3	12	11	14	6	1	1	5
431	431	3	12	11	15	7	1	1	4
432	432	3	12	11	16	8	1	1	2
433	433	3	12	11	18	9	1	1	4
434	434	3	12	11	19	10	1	1	5
435	435	3	12	12	4	11	1	1	4
436	436	3	12	12	11	12	1	1	2
437	437	3	12	12	12	13	1	1	2
438	438	3	12	11	9	1	1	2	1
439	439	3	12	11	10	2	1	2	1
440	440	3	12	11	11	3	1	2	2
441	441	3	12	11	12	4	1	2	1
442	442	3	12	11	13	5	1	2	0
443	443	3	12	11	14	6	1	2	0
444	444	3	12	11	15	7	1	2	0
445	445	3	12	11	16	8	1	2	2
446	446	3	12	11	18	9	1	2	0
447	447	3	12	11	19	10	1	2	1
448	448	3	12	12	4	11	1	2	0
449	449	3	12	12	11	12	1	2	1
450	450	3	12	12	12	13	1	2	1
451	451	3	12	11	9	1	1	3	1
452	452	3	12	11	10	2	1	3	2
453	453	3	12	11	11	3	1	3	5
454	454	3	12	11	12	4	1	3	2
455	455	3	12	11	13	5	1	3	5
456	456	3	12	11	14	6	1	3	1
457	457	3	12	11	15	7	1	3	1
458	458	3	12	11	16	8	1	3	8
459	459	3	12	11	18	9	1	3	1
460	460	3	12	11	19	10	1	3	3
461	461	3	12	12	4	11	1	3	1
462	462	3	12	12	11	12	1	3	1

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
463	463	3	12	12	12	13	1	3	1
464	464	3	12	11	9	1	1	4	1
465	465	3	12	11	10	2	1	4	1
466	466	3	12	11	11	3	1	4	7
467	467	3	12	11	12	4	1	4	8
468	468	3	12	11	13	5	1	4	2
469	469	3	12	11	14	6	1	4	4
470	470	3	12	11	15	7	1	4	4
471	471	3	12	11	16	8	1	4	0
472	472	3	12	11	18	9	1	4	5
473	473	3	12	11	19	10	1	4	5
474	474	3	12	12	4	11	1	4	3
475	475	3	12	12	11	12	1	4	3
476	476	3	12	12	12	13	1	4	6
477	477	1	13	11	18	1	1	1	5
478	478	1	13	11	19	2	1	1	5
479	479	1	13	11	21	3	1	1	10
480	480	1	13	11	22	4	1	1	1
481	481	1	13	11	23	5	1	1	5
482	482	1	13	11	24	6	1	1	2
483	483	1	13	11	25	7	1	1	4
484	484	1	13	11	26	8	1	1	5
485	485	1	13	11	28	9	1	1	2
486	486	1	13	11	30	10	1	1	5
487	487	1	13	11	18	1	1	2	4
488	488	1	13	11	19	2	1	2	5
489	489	1	13	11	21	3	1	2	5
490	490	1	13	11	22	4	1	2	3
491	491	1	13	11	23	5	1	2	3
492	492	1	13	11	24	6	1	2	4
493	493	1	13	11	25	7	1	2	3
494	494	1	13	11	26	8	1	2	5
495	495	1	13	11	28	9	1	2	3
496	496	1	13	11	30	10	1	2	6
497	497	1	13	11	18	1	1	3	1
498	498	1	13	11	19	2	1	3	1

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
499	499	1	13	11	21	3	1	3	1
500	500	1	13	11	22	4	1	3	1
501	501	1	13	11	23	5	1	3	2
502	502	1	13	11	24	6	1	3	3
503	503	1	13	11	25	7	1	3	10
504	504	1	13	11	26	8	1	3	4
505	505	1	13	11	28	9	1	3	15
506	506	1	13	11	30	10	1	3	17
507	507	1	13	11	18	1	1	4	4
508	508	1	13	11	19	2	1	4	10
509	509	1	13	11	21	3	1	4	1
510	510	1	13	11	22	4	1	4	1
511	511	1	13	11	23	5	1	4	5
512	512	1	13	11	24	6	1	4	4
513	513	1	13	11	25	7	1	4	5
514	514	1	13	11	26	8	1	4	12
515	515	1	13	11	28	9	1	4	7
516	516	1	13	11	30	10	1	4	5
517	517	2	13	11	18	1	1	1	2
518	518	2	13	11	19	2	1	1	2
519	519	2	13	11	21	3	1	1	2
520	520	2	13	11	22	4	1	1	2
521	521	2	13	11	23	5	1	1	2
522	522	2	13	11	24	6	1	1	1
523	523	2	13	11	25	7	1	1	2
524	524	2	13	11	26	8	1	1	1
525	525	2	13	11	28	9	1	1	4
526	526	2	13	11	30	10	1	1	4
527	527	2	13	11	18	1	1	2	3
528	528	2	13	11	19	2	1	2	2
529	529	2	13	11	21	3	1	2	2
530	530	2	13	11	22	4	1	2	2
531	531	2	13	11	23	5	1	2	4
532	532	2	13	11	24	6	1	2	3
533	533	2	13	11	25	7	1	2	4
534	534	2	13	11	26	8	1	2	2

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
535	535	2	13	11	28	9	1	2	6
536	536	2	13	11	30	10	1	2	4
537	537	2	13	11	18	1	1	3	2
538	538	2	13	11	19	2	1	3	2
539	539	2	13	11	21	3	1	3	1
540	540	2	13	11	22	4	1	3	1
541	541	2	13	11	23	5	1	3	1
542	542	2	13	11	24	6	1	3	5
543	543	2	13	11	25	7	1	3	5
544	544	2	13	11	26	8	1	3	2
545	545	2	13	11	28	9	1	3	1
546	546	2	13	11	30	10	1	3	0
547	547	2	13	11	18	1	1	4	3
548	548	2	13	11	19	2	1	4	6
549	549	2	13	11	21	3	1	4	7
550	550	2	13	11	22	4	1	4	7
551	551	2	13	11	23	5	1	4	5
552	552	2	13	11	24	6	1	4	4
553	553	2	13	11	25	7	1	4	4
554	554	2	13	11	26	8	1	4	2
555	555	2	13	11	28	9	1	4	5
556	556	2	13	11	30	10	1	4	0
557	557	3	13	11	18	1	1	1	1
558	558	3	13	11	19	2	1	1	1
559	559	3	13	11	21	3	1	1	0
560	560	3	13	11	22	4	1	1	2
561	561	3	13	11	23	5	1	1	1
562	562	3	13	11	24	6	1	1	1
563	563	3	13	11	25	7	1	1	1
564	564	3	13	11	26	8	1	1	1
565	565	3	13	11	28	9	1	1	1
566	566	3	13	11	29	10	1	1	1
567	567	3	13	11	18	1	1	2	3
568	568	3	13	11	19	2	1	2	4
569	569	3	13	11	21	3	1	2	2
570	570	3	13	11	22	4	1	2	1

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
571	571	3	13	11	23	5	1	2	2
572	572	3	13	11	24	6	1	2	1
573	573	3	13	11	25	7	1	2	2
574	574	3	13	11	26	8	1	2	1
575	575	3	13	11	28	9	1	2	1
576	576	3	13	11	29	10	1	2	2
577	577	3	13	11	18	1	1	3	1
578	578	3	13	11	19	2	1	3	6
579	579	3	13	11	21	3	1	3	3
580	580	3	13	11	22	4	1	3	2
581	581	3	13	11	23	5	1	3	4
582	582	3	13	11	24	6	1	3	3
583	583	3	13	11	25	7	1	3	6
584	584	3	13	11	26	8	1	3	5
585	585	3	13	11	28	9	1	3	6
586	586	3	13	11	29	10	1	3	8
587	587	3	13	11	18	1	1	4	2
588	588	3	13	11	19	2	1	4	1
589	589	3	13	11	21	3	1	4	2
590	590	3	13	11	22	4	1	4	3
591	591	3	13	11	23	5	1	4	4
592	592	3	13	11	24	6	1	4	3
593	593	3	13	11	25	7	1	4	5
594	594	3	13	11	26	8	1	4	3
595	595	3	13	11	28	9	1	4	4
596	596	3	13	11	29	10	1	4	7
597	597	1	14	11	21	1	1	1	0
598	598	1	14	11	27	2	1	1	1
599	599	1	14	12	4	3	1	1	10
600	600	1	14	11	21	1	1	2	3
601	601	1	14	11	27	2	1	2	13
602	602	1	14	12	4	3	1	2	10
603	603	1	14	11	21	1	1	3	1
604	604	1	14	11	27	2	1	3	5
605	605	1	14	12	4	3	1	3	6
606	606	1	14	11	21	1	1	4	1

continued ...

	mcf_recid	site	year	month	date	day	imgc	grid	mfc_t
607	607	1	14	11	27	2	1	4	3
608	608	1	14	12	4	3	1	4	12
609	609	2	14	11	21	1	1	1	6
610	610	2	14	11	27	2	1	1	10
611	611	2	14	12	4	3	1	1	10
612	612	2	14	11	21	1	1	2	6
613	613	2	14	11	27	2	1	2	6
614	614	2	14	12	4	3	1	2	7
615	615	2	14	11	21	1	1	3	3
616	616	2	14	11	27	2	1	3	3
617	617	2	14	12	4	3	1	3	3
618	618	2	14	11	21	1	1	4	3
619	619	2	14	11	27	2	1	4	6
620	620	2	14	12	4	3	1	4	5
621	621	3	14	11	21	1	1	1	1
622	622	3	14	11	27	2	1	1	1
623	623	3	14	12	4	3	1	1	3
624	624	3	14	11	21	1	1	2	2
625	625	3	14	11	27	2	1	2	1
626	626	3	14	12	4	3	1	2	2
627	627	3	14	11	21	1	1	3	6
628	628	3	14	11	27	2	1	3	6
629	629	3	14	12	4	3	1	3	6
630	630	3	14	11	21	1	1	4	3
631	631	3	14	11	27	2	1	4	5
632	632	3	14	12	4	3	1	4	10

Table F.1: Raw manual nest counts

	rec_id	day_c	wind.speed	cloud.cover	flight.activity	nest.activity	notes
1	1	1					Start
2	2	2	1.00		L	L	
3	3	3			L	L	
4	4	4	3.00		M	M	
5	5	5	1.00	70%	M	M	
6	6	6	2.00	40%	M	M	

continued ...

	rec_id	day_c	wind.speed	cloud.cover	flight.activity	nest.activity	notes
7	7	7	1.00	60%	M	H	
8	8	8	1.00	70%	L	L	
9	9	9	1.00	60%	M	L	
10	10	10	1.00	5%	H	H	
11	11	11	1.00	60%	M	L	
12	12	12	2.00	10%	H	M	
13	13	13	1.00	5%	H	M	
14	14	14	1.00	10%			
15	15	15	4.00		H	H	
16	16	16	4.00		M	L	End
17	17	1	3.00				Start
18	18	2	1.00	10%	M	M	
19	19	3	1.00	60%	M	M	
20	20	4					
21	21	5	1.00	40%	M	M	
22	22	6	2.00	40%	M	H	
23	23	7	0.00	60%	M	M	
24	24	8	4.00	20%	M	L	
25	25	9	0.00	30%	M	M	
26	26	10	1.00	5%	H	H	
27	27	11	1.00				
28	28	12	0.00	0%	H	H	
29	29	13	1.00	0%	H	H	
30	30	14	4.00	5%	H	H	
31	31	15	2.00	10%	H	H	
32	32	16	1.00	5%	M	M	End
33	33	1	1.00		L	L	Start
34	34	2	1.00		L	L	
35	35	3	0.00	60%	L	L	
36	36	4	0.00				
37	37	5	1.00				
38	38	6	0.00				
39	39	7					
40	40	8	1.00				
41	41	9	1.00	10%			
42	42	10	0.00	10%	M	M	

continued ...

	rec_id	day_c	wind.speed	cloud.cover	flight.activity	nest.activity	notes
43	43	11	2.00	10%	M	M	
44	44	12	1.00	60%	L	L	
45	45	13	0.00	60%	M	L	
46	46	14	1.00	20%	M	M	
47	47	15					
48	48	16					End
49	49	1	0.00		L	L	Start
50	50	2	0.00		L	L	
51	51	3	0.00		L	L	
52	52	4	1.00				
53	53	5	1.00				
54	54	6	2.00				
55	55	7	0.00				
56	56	8	0.00		M	M	
57	57	9	1.00	10%	M	H	
58	58	10	2.00	10%	M	M	
59	59	11					
60	60	12	3.00		M	M	
61	61	13	1.00		M	L	
62	62	14	2.00	10%	H	H	
63	63	15	2.00				
64	64	16	2.00		H	H	End
65	65	1			L	L	Start
66	66	2	1.00		L	L	
67	67	3	1.00		L	L	
68	68	4	1.00				
69	69	5	0.00				
70	70	6	2.00				
71	71	7	1.00				
72	72	8	1.00				
73	73	9	1.00				
74	74	10	1.00				
75	75	11					
76	76	12	1.00		M	M	
77	77	13	1.00	5%	H	H	
78	78	14	1.00	5%	H	H	

continued ...

	rec_id	day_c	wind.speed	cloud.cover	flight.activity	nest.activity	notes
79	79	15	1.00				
80	80	16					End
81	81	1					Start
82	82	2					
83	83	3	0.00	20%	L	L	
84	84	4	0.00	50%	L	L	
85	85	5	0.00	10%	M	M	
86	86	6	0.00	60%	L	L	
87	87	7	1.00	80%	L	L	
88	88	8	0.00	50%	L	L	
89	89	9	1.00	30%	L	L	
90	90	10	0.00	60%	L	L	
91	91	11	1.00	10%	H	H	HUM
92	92	12	0.00	5%	H	H	
93	93	13	0.00	5%	H	H	End
94	94	1	1.00				Start
95	95	2	0.00	10%	L	L	
96	96	3	0.00	40%	L	L	
97	97	4	0.00	70%	M	M	
98	98	5	0.00	10%	L	L	
99	99	6	0.00	100%	L	L	
100	100	7	0.00	80%	L	L	
101	101	8	0.00		L	L	
102	102	9	1.00	50%	L	L	
103	103	10	0.00	100%	L	L	
104	104	11	0.00	20%	M	L	
105	105	12	0.00	5%	M	H	
106	106	13	0.00	10%	L	M	End
107	107	1	1.00				Start
108	108	2					
109	109	3	0.00	60%	L	L	
110	110	4	0.00	60%	L	L	
111	111	5	0.00	80%	L	L	
112	112	6	0.00	30%	L	L	
113	113	7	1.00	20%	L	L	
114	114	8	0.00		L	L	

continued ...

	rec_id	day_c	wind.speed	cloud.cover	flight.activity	nest.activity	notes
115	115	9	1.00		L	L	
116	116	10	0.00		L	L	
117	117	11	0.00	60	M	M	
118	118	12	0.00	20	M	L	
119	119	13	0.00	30	M	M	End
120	120	1	0.00	20%	M	M	Start
121	121	2	1.00	70%	M	M	
122	122	3	1.00	100%	L	L	
123	123	4	1.00	80%			
124	124	5	0.00	50%			
125	125	6	0.00		H	H	
126	126	7	0.00	10%	H	H	
127	127	8	0.00	40%	H	H	
128	128	9	0.00	20%	M	M	
129	129	10	0.00	40%	H	H	End
130	130	1	0.00	20%	L	L	Start
131	131	2	1.00	80%	M	M	
132	132	3	0.00	80%	M	M	
133	133	4	0.00	90%	M	M	
134	134	5	1.00	30%			cell_ph
135	135	6	0.00		H	H	
136	136	7	1.00	20%	H	H	
137	137	8	0.00	100%	M	M	
138	138	9	1.00	20%	M	M	
139	139	10	0.00	30%	M	M	
140	140	1	0.00				
141	141	2	0.00	90%	L	L	
142	142	3	1.00	80%			cell_ph
143	143	4	0.00	40%			
144	144	5	1.00	20%			
145	145	6	1.00				
146	146	7	0.00	5%			
147	147	8	0.00	50%	M	M	
148	148	9	0.00	40%	M	M	
149	149	10	1.00	80%	M	M	End
150	150	1	0.00	90%	M	M	Start

continued ...

	rec_id	day_c	wind.speed	cloud.cover	flight.activity	nest.activity	notes
151	151	2	1.00	10%	M	M	
152	152	3	1.00	30%	L	L	
153	153	4	1.00	10%	L	L	
154	154	1	0.00	40%	L	L	
155	155	2	1.00	10%	M	M	
156	156	3	1.00	20%	L	M	
157	157	4					
158	158	1	2.00	50%	L	L	
159	159	2	3.00	10%	L	L	
160	160	3	2.00	20%	M	M	
161	161	4					End

Table F.2: Raw monitoring data

G

REFERENCES

- [1] 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.
- [2] Allsopp M.H., de Lange W.J., and Veldtman R. Valuing Insect Pollination Services with Cost of Replacement. *PLoS ONE*, 3(9), 2008.
- [3] Bauer D.M. and Wing I.S. Economic Consequences of Pollinator Declines: A Synthesis. *Agricultural and Resource Economics Review*, 39(3):368–383, 2010.
- [4] 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.
- [5] Buchmann S.L., Nabhan G.P., and Mirocha P. *The Forgotten Pollinators*. A Shearwater book. Island Press, 1997.
- [6] Newstrom-Lloyd L.E. *Pollination in New Zealand*, pages 408–431. Manaaki Whenua Press, 2013.
- [7] Losey J.E. and Vaughan M. The economic value of ecological services provided by insects. *BioScience*, 56(4):311–323, 2006.
- [8] O'Toole C. Those other bees: changing the funding culture. *Pollinating Bees-The Conservation Link Between Agriculture and Nature*, pages 37–40, 2002.
- [9] Michener C.D. *Bees of the World*. Johns Hopkins University Press, 2000.
- [10] Cane J. Annual displacement of soil in nest tumuli of alkali bees (*Nomia melanderi*) (Hymenoptera: Apiformes: Halictidae) across an agricultural landscape. *The Journal of the Kansas Entomological Society*, 76:172–176, 2003.
- [11] 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.

- [12] Donovan B.J. *Apoidea (Insecta: Hymenoptera), Fauna of New Zealand*, volume 57. Manaaki Whenua Press, Landcare Research, 2007.
- [13] Wheeler Q.D. Taxonomic triage and the poverty of phylogeny. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 359(1444):571–583, 2004.
- [14] 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.
- [15] 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.
- [16] Hart N. *New Zealand Native Bees: A case Study in Whangarei*. Msc, University of Auckland, School of Environmental Science, 2007.
- [17] Ingolf S. 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.
- [18] Tarlton R. *A cluster of bees*. Endeavour press, 1935.
- [19] Kelly D., Ladley J.J., Robertson A.W., and Edwards J. and Smith D.C. The birds and the bees. *Nature*, page 384, 1996.
- [20] 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.
- [21] Hart N. *Methods for studying potential flight distances and foraging behaviours of native pollinating insects in New Zealand*. Pg dip, University of Auckland, School of Environmental Science, 2004.
- [22] 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.
- [23] Kevan P.G. and Viana B.F. The global decline of pollination services. *Biodiversity*, 4(4):3–8, 2003.
- [24] 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.

- [25] 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.
- [26] 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.
- [27] 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 on*, pages 353–357. IEEE, 2011.
- [28] Larsson M. and Franzé n M. Estimating the population size of specialised solitary bees. *Ecological Entomology*, 33(2):232–238, 2008.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] Fellendorf M., Mohra C., and Paxton R.J. Devasting effects of river flooding to the ground-nesting bee, *andrena vaga* (hy-menoptera: Andrenidae), and its associated fauna. *Journal of Insect Conservation*, 8(4):311–312, 2004.
 - [37] Schaller R.R. Moore’s law: past, present and future. *Spectrum, IEEE*, 34(6):52–59, 1997.
 - [38] Mack C. The multiple lives of moore’s law. *Spectrum, IEEE*, 52(4):31–31, April 2015.
 - [39] Irwin A. *Citizen science: a study of people, expertise, and sustainable development*. Psychology Press, 1995.
 - [40] Steve Kelling, Wesley M. Hochachka, Daniel Fink, Mirek Riedewald, Rich Caruana, Grant Ballard, and Giles Hooker. Data-intensive science: A new paradigm for biodiversity studies. *Bio-Science*, 59(7):613–620, 2009.
 - [41] Christina L Catlin-Groves. The citizen science landscape: from volunteers to citizen sensors and beyond. *International Journal of Zoology*, 2012, 2012.
 - [42] Eric N Mortensen, Enrique L Delgado, Hongli Deng, David Lytle, Andrew Moldenke, Robert Paasch, Linda Shapiro, Pengcheng Wu, Wei Zhang, and Thomas G Dietterich. Pattern recognition for ecological science and environmental monitoring: An initial report. *Algorithmic Approaches to the Identification Problem in Systematics*, 2007.
 - [43] 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.
 - [44] MacLeod N. *Automated taxon identification in systematics: theory, approaches and applications*. CRC Press, 2007.
 - [45] 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.
 - [46] 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.

- [47] 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.
- [48] 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.
- [49] 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.
- [50] 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.
- [51] Towner A.V., Wcislo 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.
- [52] 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.
- [53] Kaku M. *Visions: how science will revolutionize the 21st century*. Oxford University Press, 1999.
- [54] Potamitis I., Ganchev T., and Fakotakis N. Automatic acoustic identification of insects inspired by the speaker recognition paradigm. In *INTERSPEECH*, 2006.
- [55] 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.
- [56] 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.
- [57] 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.
- [58] 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.
 - [59] 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.
 - [60] Compagno L.J. Government protection for the great white shark(*carcharodon carcharias*) in south africa. *S. Afr. J. SCI./S.-Afr. TYDSKR. WET.*, 87(7):284–285, 1991.
 - [61] Mayo M. and Watson A.T. Automatic species identification of live moths. *Knowledge-Based Systems*, 20(2):195–202, 2007.
 - [62] 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.
 - [63] 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.
 - [64] Murray T.E., Kuhlmann M., and Potts S.G. Conservation ecology of bees: populations, species and communities. *Apidologie*, 40(3):211–236, 2009.
 - [65] 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.
 - [66] Mascanzoni D. and Wallin H. The harmonic radar: a new method of tracing insects in the field. *Ecological Entomology*, 11(4):387–390, 1986.
 - [67] 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.
 - [68] 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.
 - [69] 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.

- [70] 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.
- [71] 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.
- [72] 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.
- [73] Johansson P. Small vessel detection in high quality optical satellite imagery. Technical report, Tech. Report Chalmers University of Technology Sweden, 2011.
- [74] Bell G., Hey T., and Szalay A. Beyond the data deluge. *Science*, 323(5919):1297–1298, 2009.
- [75] Hey T., Tansley S., and Tolle K.M. Jim gray on escience: a transformed scientific method., 2009.
- [76] 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.
- [77] 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.
- [78] 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.
- [79] Goodchild M.F. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.
- [80] Hampton S.E., Strasser C.A., Tewksbury J.J., Gram W.K., Buden 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.
- [81] RD Brazee, ES Miller, ME Reding, MG Klein, B Nudd, and H Zhu. A transponder for harmonic radar tracking of the black vine weevil in behavioral research. *Transactions of the ASAE*, 48(2):831–838, 2005.

- [82] Delaney J. An inconvenient truth? scientific photography and archival ambivalence. *Archivaria*, 65:75–95, 2008.
- [83] 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.
- [84] 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.
- [85] 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.
- [86] 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.
- [87] 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.
- [88] Frank Pennekamp and Nicolas Schtickzelle. 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.
- [89] Glasbey C.A and Horgan G.W. *Image analysis for the biological sciences*, volume 1. Wiley Chichester, 1995.
- [90] Pearson H. The good, the bad and the ugly. *Nature*, 447(7141):138–140, 2007.
- [91] Crome 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.
- [92] 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.
- [93] 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.
- [94] 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.
 - [95] 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 machine learning. *Journal of Machine Learning Research*, 8:2443–2466, 2007.
 - [96] 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.
 - [97] Qidwai U. and Chen C. *Digital image processing: an algorithmic approach with MATLAB*. CRC press, 2011.
 - [98] 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.
 - [99] North A.J. Seeing is believing? a beginners’ guide to dctical pitfalls in image acquisition. *The Journal of cell biology*, 172(1):9–18, 2006.
 - [100] 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.
 - [101] 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.
 - [102] 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.
 - [103] Peña J.M, Torres-Sánchez J., Serrano-Pérez A., de Castro A.I, and López-Granados F. Quantifying efficacy and limits of un-manned aerial vehicle (uav) technology for weed seedling detection as affected by sensor resolution. *Sensors*, 15(3):5609–5626, 2015.
 - [104] 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.

- [105] 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.
- [106] Feng Q., Liu J., and Gong J. Urban flood mapping based on unmanned aerial vehicle remote sensing and random forest classifier - a case of yuyao, china. *Water*, 7(4):1437–1455, 2015.
- [107] 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.
- [108] 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.
- [109] 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.
- [110] 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.
- [111] 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.
- [112] Pierre emmanuel G. Xnview: Software for viewing, converting, organising and editing raster images. *Software available at <http://www.xnview.com>*, 2014.
- [113] 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.
- [114] 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.
- [115] Wang S., Chung F., and Xiong F. A novel image thresholding method based on parzen window estimate. *Pattern Recognition*, 41(1):117–129, 2008.

- [116] Deriche R. Using canny's criteria to derive a recursively implemented optimal edge detector. *International journal of computer vision*, 1(2):167–187, 1987.
- [117] Nock R. and Nielsen F. Statistical region merging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(11):1452–1458, 2004.
- [118] Nock R. and Nielsen F. Semi-supervised statistical region refinement for color image segmentation. *Pattern Recognition*, 38(6):835–846, 2005.
- [119] Kaynig V., Arganda-Carreras I., and Cardona A. Trainable weka segmentation plugin. *Kaynig, V and Arganda-Carreras, I, Editors*, 2010.
- [120] 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.
- [121] 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.
- [122] 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.
- [123] 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.
- [124] Díaz-Uriarte R. and De Andres S. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.
- [125] 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.
- [126] 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.

- [127] 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.
- [128] 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.
- [129] 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.
- [130] 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.
- [131] 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.
- [132] 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.
- [133] Di E.B. and Glass M. The kappa statistic: A second look. *Computational linguistics*, 30(1):95–101, 2004.
- [134] 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.
- [135] 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.
- [136] 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.
- [137] 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.

- [138] 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.
- [139] Reichman O.J, Jones M.B, and Schildhauer M.P. Challenges and opportunities of open data in ecology. *Science*, 331(6018), 2011.
- [140] Steiniger S. and Hay G.J. Free and open source geographic information tools for landscape ecology. *Ecological Informatics*, 4(4):183–195, 2009.
- [141] European Union. Advancing technologies and federating communities. 2012.
- [142] Hsien Ming Easlon and Arnold J Bloom. Easy leaf area: Automated digital image analysis for rapid and accurate measurement of leaf area. *Applications in plant sciences*, 2(7), 2014.
- [143] 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.
- [144] Harvey P. Exiftool: Read, write and edit meta information. Software package available at <http://www.sno.phy.queensu.ca/~phil/exiftool>, 2013.
- [145] Meijering E. Featurej: A java package for image feature extraction, 2007.
- [146] Tschumperle D. and Deriche R. Vector-valued image regularization with pdes: A common framework for different applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(4):506–517, 2005.
- [147] Chaudhury K.N., Sage D., and Unser M. Fast bilateral filtering using trigonometric range kernels. *Image Processing, IEEE Transactions on*, 20(12):3376–3382, 2011.
- [148] 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.
- [149] Lawrence I. and Lin K. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, pages 255–268, 1989.
- [150] Gabriel L. and others. Auto threshold. Software package available at <http://fiji.sc/AutoThreshold>, 2015.

- [151] Breiman L. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [152] Latinne P., Debeir O., and Decaestecker C. Limiting the number of trees in random forests. In *Multiple Classifier Systems*, pages 178–187. Springer, 2001.
- [153] 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.
- [154] Matthew N. and Stones R. *Beginning Linux Programming*. John Wiley & Sons, 2011.
- [155] 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.
- [156] Nart N. Native bees: Digital image collections used for monitoring nz bees. *Information and data repository – <http://nzbees.github.io>*, 2013.
- [157] Adolfo G.B. Pyrenamer. Software available at <https://launchpad.net/pyrenamer>, 2015.
- [158] The Apache Software Foundation. Apache openoffice 4.1.1: The free and open productivity suite. Software available at <https://www.openoffice.org/>, 2015.
- [159] Hothorn T. and Everitt B.S. *A handbook of statistical analyses using R*. CRC Press, 2014.
- [160] Synteko. Smartgit 6.5.7: A git client with mercurial and subversion support. Software package available at <http://www.synteko.com/smartgit/download>, 2013.