Automated UAV and Satellite Image Analysis For Wildlife Monitoring

by

Ellen Bowler

A thesis submitted in partial fulfilment for the
degree of Doctor of Philosophy

in the
University of East Anglia
School of Computing Sciences

February 2023

Abstract

Very high resolution satellites and unmanned aerial vehicles (UAVs) are revolutionising our ability to monitor wildlife, especially species in remote and inaccessible regions. However, given the rapid increase in data acquisition, computer-automated approaches are urgently needed to count wildlife in the resultant imagery. In this thesis, we investigate the application of convolutional neural networks (CNNs) to the task of detecting vulnerable seabird populations in satellite and UAV imagery. In our first application we train a U-Net CNN to detect wandering albatrosses in 31-cm resolution WorldView-3 satellite imagery. We compare results across four different island colonies using a leave-one-island-out cross validation, achieving a mean average precision (mAP) score of 0.669. By collecting new data on inter-observer variation in albatross counts, we show that our U-Net results fall within the range of human accuracy for two islands, with misclassifications at other sites being simple to filter manually. In our second application we detect Abbott's boobies nesting in forest canopy, using UAV Structure from Motion (SfM) imagery. We focus on overcoming occlusion from branches by implementing a multi-view detection method. We first train a Faster R-CNN model to detect Abbott's booby nest sites (mAP=0.518) and guano (mAP=0.472) in the 2D UAV images. We then project Faster R-CNN detections onto the 3D SfM model, cluster multi-view detections of the same objects using DBSCAN, and use cluster features to classify proposals into true and false positives (comparing logistic regression, support vector machine, and multi-layer perceptron models). Our best-performing multi-view model successfully detects nest sites (mAP=0.604) and guano (mAP=0.574), and can be incorporated with expert review to greatly expedite analysis time. Both methods have immediate real-world application for future surveys of the target species, allowing for more frequent, expansive, and lower-cost monitoring, vital for safeguarding populations in the long-term.

List of Publications

We have published the results of our VHR satellite UNet method for detecting wandering albatrosses in one conference publication and a full journal article, and are currently preparing a manuscript for the research regarding Abbott's booby detection in UAV structure from motion imagery.

Journal Publications

Bowler, E., Fretwell, P. T., French, G. and Mackiewicz, M. Using deep learning to count albatrosses from space: Assessing results in light of ground truth uncertainty. *Remote Sensing 12.12* (2020): 2026.

In preparation for *Remote Sensing in Ecology and Conservation*: Bowler, E., Lipka, C., Green, P. T. and Mackiewicz, M. Multi-view detection of canopy nesting birds using UAV structure from motion imagery.

Conference Publications

Bowler, E., Fretwell, P. T., French G. and Mackiewicz, M. Using deep learning to count albatrosses from space. *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium* (2019), pp. 10099–10102.

Acknowledgements

Many thanks to my supervisor, Dr. Michal Mackiewicz, for his support and guidance throughout the PhD. Also very special thanks to my BAS supervisor, Dr. Peter Fretwell, for his continuing guidance and expertise, not to mention my co-supervisors, Dr. Norman Ratcliffe and Prof. Graham D. Finlayson, for their helpful advice and input. Thanks to the NEXUSS CDT organisers and Natural Environmental Research Council for funding this research project (grant number NE/N012070/1). I gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

I am lucky to have worked with excellent collaborators from the field of ecology and remote sensing, both at the British Antarctic Survey and LaTrobe University. All my thanks to Christina Lipka for sharing her dataset and knowledge of Abbott's boobies, and for being so patient with results during the pandemic. Further thanks to Dr. Peter T. Green at LaTrobe for his input and enthusiasm regarding the project.

I am grateful to Javier Vazquez-Corral for his advice and mentorship throughout my time at UEA, as well as to Geoff French for his invaluable inputs on deep learning. Further thanks to the IT support team at UEA who have always been incredibly helpful, and to Dr. Katharina Huber for her support as PGR director. Also many many thanks to my volunteer albatross counters, who gave up their time to sit in a dark room and click on blurry white dots.

Finally personal thanks to Yuteng Zhu for being such a constant source of joy in the office, and beyond. As well as to Fufu Fang for his friendship and entertaining conversations. I'm thankful to have had Hannah Houlding as my wonderful housemate and friend during my time at UEA. Thanks also to my parents, especially for keeping me company through the lockdown phase of the PhD. And finally to Raquel, Fifi and Pete who have helped me so much through the write up of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Over recent decades advances in technology and sensors such as drones, satellites, camera traps, GPS tags and audio devices have revolutionised our ability to study wildlife populations [144]. This wealth of new data is improving our understanding of species health, behaviour, abundance and distribution; critical to implementing informed conservation actions and safeguarding species into the future. However, as access to these sensors has become more widespread and their cost has decreased, manually reviewing the large amounts of collected data is no longer feasible. As wildlife monitoring moves into the big data realm machine learning methods are needed to automatically extract meaningful information [143]. This will reduce the time and cost required at the data review stage, allowing for important research questions to be answered more efficiently, and for near real-time monitoring and response to wildlife threats. This research will focus specifically on developing ML methods to detect wildlife in image data collected from *very high resolution (VHR) satellites* and *unmanned aerial vehicles (UAV's)*. We will develop methods to survey two species of seabirds: wandering albatross in VHR satellite imagery and Abbott's boobies in UAV imagery. We will compare and contrast the methods and platforms used to census the two populations, and investigate sources of uncertainty as well as approaches for dealing with this uncertainty. This chapter will provide an introduction to the study by first discussing the background and context, followed by the research problem, aims, objectives and questions, the significance and limitations, and concluding with an outline of the thesis structure.

1.2 Research Problem

Conducting regular species abundance surveys is essential for developing informed and optimised conservation plans, yet traditional ground based surveys have many limitations. They can be inaccurate, expensive, time consuming, and logistically challenging, particularly in remote areas. Recently modern RS technologies have been used to address some of these challenges. RS offers an unobtrusive means of conducting surveys, often allowing for wider area coverage, and the ability to conduct counts in otherwise inaccessible regions [80]. For example, researchers at the British Antarctic Survey (BAS) have shown that wandering albatrosses (*Diomedea Exulans*) nesting on remote sub-Antarctic island chains can be surveyed using 31-cm resolution satellite imagery [42]. With an adult wandering albatross having a body length between 107-135cm [2], albatrosses appear as approximately 4 to 5 pixels of white against their green nesting habitats. This enables populations to be monitored more frequently than is currently possible with boat, plane or ground counts [42], so that population declines can be identified and understood.

While VHR satellites are a powerful tool for wildlife monitoring in extreme environments, species must meet key suitability criteria to be amenable to detection. They should be large enough to be observed given the spatial resolution of the sensor, their body colour should contrast against their surroundings, and they should be in open habitat so they can be seen from an aerial perspective [85]. For species which do not meet these criteria, UAVs can provide a more suitable alternative to VHR satellites. For instance Abbott's boobies (*Papasula abotti*), a seabird which nests in the forest canopies of Christmas Island, are too small and their colouration to indistinct to be monitored using current VHR satellites. In addition their habitat is not fully open, and they can be obscured by branches and leaves from an overhead aerial perspective. While this also poses a challenge for aerial UAV surveys, researchers at LaTrobe University are investigating whether UAV imagery used in combination with 3D structure from motion processing can be used to identify birds within the canopy. This method involves collecting multiple images of the canopy from different angles, to compensate for when the birds are obscured in any given single viewpoint. While this has shown to improve detection compared to a single view survey (Lipka et al., unpublished), it increases the number of images to review by 10 fold. This significantly adds to the burden of manual analysis, making it impossible to scale up the approach over large areas and to more frequent time intervals.

In both cases, a limiting factor in the studies is the need for experts to manually analyse the resulting imagery, which is a tedious, time consuming and expensive process. This strongly motivates the development of automated image analysis techniques which can perform this task. Traditional image analysis methods, which often rely on designing

hand crafted features for object detection and recognition, have so far been unable to provide sufficiently reliable and robust results. However, in recent years a branch of machine learning methods called Deep Learning (DL) have shown impressive performance in a range of tasks, including image analysis. Rather than searching for hand crafted features, DL algorithms use Convolutional Neural Network (CNN) structures to automate hierarchical feature learning in image recognition. This has been reported to provide a step-increase in performance. However, these methods have largely been trained and benchmarked using standard RGB camera image datasets such as ImageNet and COCO. When transferring methods to satellite and UAV imagery specific challenges must be addressed. For example satellite imagery generally consists of multiple spectral bands, visibility be affected by atmospheric conditions, and target objects such as wildlife can be small, rare and indistinct. Due to the relatively coarse resolution of wildlife in the imagery ground truth annotations can also be very uncertain, even for domain experts. Considering the implications on network training and assessment, ground truth uncertainty must be factored in when employing supervised classification methods. Similarly annotations of wildlife in UAV imagery can be uncertain, particularly in complex non-open habitats such as forest canopy. Considering novel approaches to account for this in network design can help broaden the application of UAV monitoring to new species.

Our research will investigate how we can apply these modern DL architectures to the task of wildlife detection in satellite and UAV imagery, using the wandering albatross and Abbott's booby datasets as test cases for development.

1.3 Research Aims and Objectives

The aim of this research is to develop automated methods for detecting and counting wildlife in satellite and UAV imagery, to facilitate fast, scalable and reproducible methods which can be used for population censuses. We specifically aim to investigate key sources of uncertainty in our datasets: uncertainty in ground truth labels due to coarse resolution in satellite imagery, and uncertainty in different viewing angles from UAV surveys of canopy nesting species.

Our research objectives are:

RO1: Develop an automated method for counting wandering albatrosses in 31-cm resolution WorldView-3 satellite imagery.

RO2: Assess uncertainty in human annotations of wandering albatrosses in 31-cm resolution satellite imagery, and assess how this impacts the training and assessment of the network developed in R01.

RO3: Develop an automated method for counting Abbott's boobies nesting in forest canopy from imagery collected by a UAV.

RO4: Develop a novel multi-view approach for dealing with viewing angle uncertainty in the Abbott's booby dataset, by merging detections from R03 using 3D structure from motion information.

1.4 Novel contributions

This research contributes to the survey approaches for the specific target species, neither of which have been attempted to be counted automatically before. To summarize, our novel contributions are:

- We conduct experiments to assess inter-observer agreement in counts of wandering albatrosses in 31-cm resolution WorldView-3 imagery. The new dataset, consisting of point annotations for four satellite images from six independent observers, can be used to benchmark manual and automated detection methods for the species and is publicly available to download [16].
- We develop a novel automated method for detecting and enumerating wandering albatrosses in 31-cm WorldView-3 imagery using a VHR U-Net architecture in combination with the focal loss. This is the first automated method for detecting wandering albatrosses in satellite imagery, and has been published in a conference [15] and journal [16] paper.
- We empirically show that the choice of observer ground truth label can impact the accuracy of the VHR U-Net method, highlighting that ground truth uncertainty has a significant impact on the network results. Our experiments prove that the VHR U-Net method falls within the range of human accuracy when benchmarked using the inter-observer annotation dataset.
- We develop an automated method for detecting Abbott's boobies in single-view UAV imagery using a Faster R-CNN architecture. This is the first automated method for detecting Abbott's boobies in UAV imagery, and provides a benchmark for applying this CNN to the detection of the species.
- We propose a novel multi-view approach for combining single-view UAV detections, using 3D structure from motion information. This method can account for uncertainty when objects are obscured from particular viewpoints, helping to monitor wildlife in complex habitats such as forest canopy. This method is being deployed to conduct the first island-wide census of Abbott's boobies using UAV.

The methods developed for both species have direct real-world applications for species management and monitoring for stakeholder organisations. More broadly, specific challenges addressing uncertainty in ground truth annotations and viewing angle have wider applications for other species. Consequently, our findings help to advance the current shortage of research tailoring to these specific challenges in satellite and UAV monitoring of wildlife.

1.5 Limitations

Scope: The datasets examined in this thesis are specifically wandering albatrosses in 31-cm resolution WorldView-3 satellite imagery and Abbotts boobies in UAV imagery (collected with specific flight parameters detailed in Section 5.3.1). Automated methods will therefore be tailored to these species and platforms. For instance, it might be that equivalent imagery collected by a different satellite would have different specifications for spectral bands. However, techniques such as transfer learning and domain adaptation can help to bridge the gap between our system and a new dataset. In addition, both datasets were collected prior to the proposed project, and so collection design, for example flight height, amount of overlap between images and post processing into orthomosaics could not be tailored to improve detection performance. This remains an interesting avenue for further research.

Methodology: Supervised training schemes, particularly deep learning methods, tend to benefit from large annotated datasets, which are rarely available for ecological studies. In this research we leverage transfer learning and data augmentation to artificially increase training size, but we note that adding more data will invariably improve performance and generalisability. In addition, while the CNN architectures selected in this study represent current benchmark approaches, DL is a rapidly evolving field and new network architectures are likely to improve performance in the future.

Resources: Our network training and development was conducted on an NVidia Titan XP graphics card and was restricted by GPU processing speed and memory requirements. Reported timings will increase or decrease depending on the users own processing set up. Time and computational resources meant that certain avenues could not be explored, for instance, it would have been beneficial to manually re-annotate the Abbott's booby dataset with bounding boxes rather than point annotations. Given the need for expert domain knowledge and limited time resources this was not possible within the scope of this PhD, and so approximate bounding boxes were generated. Given the fixed flight height of the UAV and the approximately fixed size of the target species we do not expect results to be significantly affected, however a small improvement would be likely.

1.6 Thesis Structure

In this chapter, the context of the study has been introduced and we have identified the research aims and objectives. We have outlined the value and novelty of our research, and discussed the limitations and scope.

The thesis continues with a literature review chapter (Chapter 2). We present the background of remote sensing for wildlife surveys, focusing particularly on the use of satellite and UAV platforms. We discuss aspects including image acquisition, specifications such as spatial and spectral resolution, and examples of how these emerging technologies have been applied to survey a range of species. Following this we outline the foundations of automated image analysis using CNNs. We establish the core ideas and terminology used throughout the thesis, provide an overview of the specific CNN architectures employed, and summarise key performance metrics used to assess and compare results.

Our four main research chapters present automated methods for detecting wildlife, in particular two species of seabirds, in satellite and UAV imagery. In Chapter 3 we present our output for research objective 1 (RO1); developing an automated method for detecting wandering albatrosses in 31-cm satellite imagery. We use a U-Net architecture, in combination with the focal loss, to detect albatrosses using a satellite image dataset capturing four different colonies. We present the results of this method as a four-fold cross validation across colonies, achieving a mean Average Precision (mAP) score of 0.669. Within the context of these findings, we discuss limitations imposed by the high degree of ground-truth uncertainty in the manually generated annotations.

To extend on this in Chapter 4 we deliver RO2, by investigating the scale of inter-observer variation in human annotations of albatrosses in 31-cm satellite imagery. We present the results of an empirical study where satellite images of albatrosses were annotated under experimental conditions by five volunteers. We analyse observer agreement, and assess whether image quality can influence inter-observer variation in the four different images. Further to this we show how this ground-truth uncertainty can impact the results of the CNN network, both at the training and assessment stage. We find that when accounting for ground truth uncertainty our VHR U-Net results fall within the range of human accuracy for two of the islands, and that misclassifications for the other two islands are simple to filter manually.

For our second application we present methods for surveying birds nesting in forest canopy using UAV imagery. Our study species is the Abbott's booby, a tree nesting seabird endemic to Christmas Island. In Chapter 5 we present our contribution for RO3, and train a Faster R-CNN network to automatically detect Abbott's booby nest sites and guano in images collected by a UAV. We assess the performance of Faster

R-CNN, achieving mAP scores of 0.518 for Abbott's boobies and 0.472 for guano. We end by showing that a multi-view approach (where detections of the same object from different viewpoints are merged) has the potential to significantly improve recall for hard to classify examples.

For our final research objective (RO4), in Chapter 6 we present methods for merging and classifying Faster R-CNN detections using a multi-view detection approach. We describe the process for projecting detections from the 2D UAV images onto the 3D model of the forest canopy using parameters derived from SfM. We use DBSCAN to cluster multi-view results of the same Abbott's boobies into groups, and calculate a range of features for each cluster including the average confidence score and the cluster density. We compare clusters to ground truth labels to assign true positive and false positive classes, and in the final stage train a classifer model to predict and filter out false positives. For our classifier we compare a logistic regression, support vector machine, and multi-layer perceptron model, with the best performing model achieving a mAP of 0.604 for Abbott's boobies and 0.574 for guano. We show that using human-in-the-loop analysis to assess outputs can result in 70% recall of objects with 40% precision, and greatly reduce the manual analysis time. In our discussion we highlight areas for future improvement of the method.

To conclude, in Chapter 7 we summarise thesis contributions and recommendations, and outline suggestions for future work.

Chapter 2

Literature Review

2.1 Overview

In this Chapter we present a literature review of both remote sensing for wildlife detection and deep learning with Convolutional Neural Networks (CNNs). We begin by reviewing the background of wildlife monitoring with satellites and UAVs, and outline key challenges relating to automated detection of wildlife. In Section 2.3, we review how satellites and Unmanned Aerial Vehicles (UAVs) can be used to monitor wildlife, and outline the practicalities and benefits of using them for different survey efforts. In the subsequent sections we discuss methods for analysing the resultant imagery in order to count wildlife. Section 2.4 focuses on manual detection, while in Section 2.5 we review the existing literature on automated detection methods. To conclude, we discuss limitations with current approaches and outline areas for future research, in particular the application of deep learning and Convolutional Neural Networks (CNNs). In Section 2.9 we describe the building blocks of CNN architectures, including pre-processing and data augmentation, feature extraction, classification and training. In Sections 2.10-2.12 we outline the specific CNN and ML architectures that we utilise in the study, and conclude by defining key performance metrics in Section 2.13.

2.2 Remote Sensing for Wildlife Monitoring

2.2.1 Overview

Collecting regular and reliable estimates of wildlife population sizes is essential for successfully monitoring population health and developing conservation plans [69, 115]. Traditionally such surveys have been conducted via ground based counts, where animals are

physically counted by observers in the field. This can be an inexact science, prone to errors due to animals moving or being obscured from view, and additionally hindered by site accessibility, logistical costs and weather constraints, particularly in remote areas [42, 43, 84, 110, 155]. Recently the increased availability of remote sensing technologies, such as satellites and UAVs, has provided a platform for conducting these surveys remotely. This allows data to be collected comparatively cheaply, over wider areas, more frequently and without disturbance to the wildlife or environment. Researchers can also benefit from having a permanent visual record of wildlife distribution, making results verifiable and available for later studies and alternative interpretations [115]. For instance spatial information such as the locations of individuals, preferred habitat type and vegetation quality at the survey time can all be analysed in subsequent studies [118]. As such several papers have suggested the use of remote sensing technology will revolutionise the field of wildlife monitoring [6, 55, 66, 85].

2.3 Vision Data and Platforms

Remote sensing involves detecting physical characteristics of an area remotely, by measuring reflected electromagnetic waves at a distance from the target [69]. Sensors can sample different ranges of the spectrum, with the width of electromagnetic bands recorded being referred to as the *spectral resolution*. This includes single band panchromatic imagery, multispectral imagery which consists of multiple bands (e.g visible light, infrared and thermal), and hyperspectral imagery which can contain hundreds of bands. Sensors also have different *spatial resolution*, referring to the area represented by a single pixel in the image. Higher spatial resolution means more detailed imagery can be collected, helping to distinguish smaller animals and improving classification accuracy. Both the spectral and spatial resolution of sensors have an important impact on the ability to detect wildlife in the resultant images. The properties of images will also depend on the platform used to collect it. While platforms such as camera traps can be used to collect vision data for wildlife monitoring, in this thesis we will be focusing specifically on aerial surveys using VHR satellite and UAV platforms. Satellites and UAVs are more amenable to large area population surveys than in-field camera traps, which tend to be used for presence/absence monitoring of species within a fixed field of view. In the following sections we will outline the background, benefits and limitations of conducting wildlife population surveys using satellites and UAVs.

2.3.1 Satellite Surveys

Satellites were first proposed as a means of surveying wildlife as early as the 1980s, with studies remotely detecting wombat warrens [96] and Adelie penguin rookeries [128, 129] using 15m resolution imagery. Recently the advent of Very High Resolution (VHR, <1m spatial resolution) satellite sensors has lead to an increased interest in their application to directly survey wildlife from space. Multiple satellites now collect VHR imagery, with many including up to eight spectral bands (the specifications of some of the most commonly used VHR satellites are presented in Table 2.1). These satellites benefit from being able to cover very large spatial extents and revisit locations frequently (often every one or two days), which allows for repeat observations to be made. They are also completely passive and cause no disruption to wildlife, which has been reported to be an issue in some aerial surveys [48, 146]. Crucially satellites also facilitate surveys of wildlife in regions which are challenging or impossible for humans to access. This includes species in the open ocean (e.g whales [4, 28, 43]), the Arctic (e.g walruses [10] and polar bears [86, 137]), Antarctica (e.g penguins [8, 40, 44, 82, 100, 130] and seals [5, 81, 84, 104]), and open savannah (e.g elephants [126], and wildebeest and zebra [157, 158]).

There are some constraints to consider when designing VHR satellites surveys. While conducting surveys by this means requires no logistical layout, purchasing imagery can be expensive, especially for large spatial extents [69, 115]. There is also no option to specify the exact time and place that an image should be collected, which can be limiting for some studies. Cloud cover and spatial distortions can also hinder analysis of the imagery [158]. Additionally there are restrictions on the type of species which are amenable to survey by satellite. A feasibility study by LaRue et al. [85] outlines three primary criteria, namely that the animal should live in open habitat, have a colour contrast to the surrounding landscape, and be of detectable size. For this reason many studies to date have focused on wildlife in polar regions, where snow and lack of vegetation provides a good contrast for detection [82]. One of the main challenges in directly counting individuals is the spatial resolution of the imagery [79]. A number of studies have compensated for relatively low spatial resolution by performing indirect counts. In these methods proxies such as colony size and nest area are used to estimate abundance using lower resolution satellites. This has been effectively employed to conduct global estimates of penguin population sizes, by first quantifying colony area or guano staining extent, and then using ground counts and regression to predict population numbers [8, 40, 41, 82]. These studies proved very effective in detecting large, previously undiscovered penguin colonies, which drastically altered the total species population estimates (e.g [40] and [14]). This shows the potential for satellite surveys to complement ground based methods and dramatically improve our understanding of species distribution and trends.

Satellite	Launch Year	MS Band Number	PAN Resolution	MS Resolution	Average Revisit
IKONOS	1999	4	0.82m	3.2m	3 days
Quickbird	2001	4	0.55m	2.16m	2.5 days
WorldView-1	2007	-	0.5m	-	1.7 days
GeoEye	2008	4	0.41m	1.65m	2.6 days
WorldView-2	2009	8	0.46m	1.85m	1.1 days
WorldView-3	2014	8	0.31m	1.24m	1 day
WorldView-4	2016	4	0.31m	1.23m	1 day

TABLE 2.1: Very High Resolution satellite specifications.

2.3.2 Aerial Surveys

Aerial surveys are generally conducted through sensors mounted on UAVs and manned aircraft. A key advantage over satellite acquisition is higher spatial resolution imagery (e.g up to 2.5cm [69]), allowing for greater detection capabilities for a wider range of species. To date UAVs have been used to study three groups of animals [94]; large terrestrial species (e.g elephants [147]), marine mammals (e.g dugongs [101] and whales [65]), and birds (e.g penguins [105, 142], flamingos [30, 58] and geese [19]). Unlike satellite surveys, aerial methods can be designed according to researchers' specifications, and therefore can be adapted to compensate for cloud cover and to fly additional transects when required. UAVs are also increasingly used in situations where real time monitoring of wildlife is a priority, for instance patrolling beaches for sharks [153] and detecting and deterring poachers in African game parks [11]. While aerial imagery is similarly limited by the need for open habitat, a promising avenue of research is the use of thermal sensors to detect species which may be obscured from view, or which need to be detected at night [11, 55, 94]. This thermal information can be used in conjunction with colour imagery to help discriminate wildlife from visually similar background objects such as rocks and shadows. These new developments in technology and improved sensor capabilities can be tested with a faster uptake than satellite platforms [69].

UAVs have some limitations, in particular they require human operation so are not amenable to remote area surveys as satellite platforms are. They also suffer from low flight endurance, meaning that they can often only survey limited areas in a single flight [24, 94]. UAVs are also heavily restricted by weather conditions, particularly wind [118], and require specialist operators and training which can add expense [94]. The cost and quality of imagery depends on the specifications of the platforms and sensors, and purchasing these can be a considerable investment given the techniques are in their relative infancy, which can limit uptake. Further to this the legislation surrounding the use of UAVs can hinder their use [24, 94]. This being said it is expected that with rapid developments and further interest in the technology many of these obstacles will

be overcome [6]. Aerial methods have still been able to prove their effectiveness given these limitations, with studies showing that UAV surveys can be an order of magnitude more precise than traditional ground based counts [66, 67].

2.3.3 Limitations of Current Approaches

To summarise, satellite and UAV surveys each have their limitations and draw backs, and the selection of the technology should depend on the target species and study area. For very remote and hard to access regions, VHR satellite surveys can be a cost effective and safe way of collecting survey data. However the resolution of VHR satellite imagery (currently a maximum of 31-cm ground sample distance) means that it is only suitable for direct detection of large species. Even if an animal is visible it will generally only appear as a few pixels, which can make them challenging to identify and differentiate from other objects in the surrounding landscape. The cost and availability of imagery can also be a limiting factor, particularly in regions with high levels of cloud cover. However as new VHR satellite constellations are launched prices will decrease and the frequency of image acquisition will improve. UAVs provide an alternative for species which are not suitable for survey with VHR satellite. UAVs can generally collect higher resolution imagery, but are limited by flight endurance, weather conditions, and restrictions of being flown within line of sight. Therefore they are not suitable for surveys of very remote and inaccessible areas. Similar to VHR satellites they are also more suited to species in open habitats, where aerial detection is possible without the wildlife being obscured. In more complex terrains post-processing of UAV imagery can introduce distortion from image stitching, which can make detection of wildlife challenging. Accounting for these limitations would allow UAV surveys to be applied to a wider range of species.

2.4 Manual Data Analysis

The recent proliferation of satellite and UAV data presents new challenges, with a key one being the need to identify and count animals in the acquired imagery. Traditionally these counts have been conducted manually by expert observers. Indeed, even if the final goal is to automate detection with machine learning methods, generating labelled training data is often a necessary first step [33, 39]. Here we outline methods for manual detection using both expert analysis and crowd-sourced counts.

2.4.1 Expert Analysis

The most established method for analysing remote sensing imagery is through manual counts [69], where images are hand labelled using image software such as Esri ArcGIS (Redlands, C. E. S. R. I. 2022. ArcGIS Desktop: Release 10.3), GIMP [50] and Agisoft (V 1.5.2, Agisoft LLC, St. Petersburg, Russia). This process is often time-consuming and can require expert interpretation, which is costly and tedious to conduct. This limits the amount of data which can feasibly be annotated, meaning that to date manual analysis has largely been limited to small, proof of concept studies [42, 104]. In addition, manual counts are non-repeatable, and prone to the same subjective analysis and human error as ground based counts [137]. Often the appearance of wildlife can be uncertain, resulting in different experts deriving different counts for the same image - termed *inter-observer variation*. This is particularly prevalent in satellite surveys, where the resolution of animals is generally in the region of only a few pixels [42].

To account for inter-observer variation studies generally recruit multiple observers to annotate the same dataset. The number of observers and their level of experience can differ markedly between different studies, and is dependent on resources (e.g funding and time), the analysts experience with annotation methods and software, and the detectability of the target species [42, 104]. Most studies have selected between between one and five expert analysts [33, 36, 39, 83, 104, 158]. Studies often recruit observers with either general expertise in the review of satellite or UAV imagery [84] or with specific knowledge of the target species (e.g who have conducted field-based ground, boat or aerial counts) [22, 137]. However even within a group of experts observer agreement and accuracy tends to vary depending on the complexity of the environment (e.g it's heterogeneity [33]) and with the quality of the image [22, 42]. Variation in manual counts of colonial animals may also increase with colony size [9].

2.4.2 Crowd-sourced Counts

Recently an increasing number of projects have used crowd-sourcing (also referred to as citizen science) to analyse data. Platforms such as Zooniverse [134] and GeoHIVE (previously Tomnod) have been developed, encouraging members of the public to manually analyse imagery to speed up the annotation process. For many wildlife species, this has the added benefit of raising awareness and public interest, and can generate donations for conservation projects. Crowd-sourcing platforms enable vast quantities of images to be analysed, which would not be feasible with expert analysis alone. For example, recently the "Satellites over Seals" project engaged over 300,000 citizen scientists to count seals over the entire Antarctic Peninsula using VHR satellite imagery [81]. For VHR satellite

imagery in particular this has be additional benefit of being substantially cheaper than purchasing imagery. Maxar's VHR satellite data can be hosted and reviewed remotely by the crowd on the GeoHIVE platform, without the need for researchers to purchase, download and store the imagery. While currently Maxar is the only VHR satellite image provider offering this service, other providers may develop their own platforms in the future. While using citizen science incurs the same limitations and uncertainty as outlined for manual analysis, it can be a useful tool for generating large labelled datasets for use in supervised classification training schemes. In addition, images can be counted multiple times by different observers, which means that uncertainty and inter-observer variation in labels can be assessed and accounted for.

2.4.3 Limitations of Manual Data Analysis

Manually detecting wildlife in satellite and UAV imagery presents a number of challenges. A primary concern is the amount of time and labour it takes to hand-annotate datasets. This creates a bottleneck in analysis when looking to conduct surveys over larger areas, or at more frequent intervals, which motivates the development of automated methods [13]. However, challenges with manual analysis are also exacerbated by the fact that wildlife is small and hard to distinguish. Even large animals appear as only a few pixels [148], and are generally only discernible due to a strong spectral contrast with their habitat, opposed to other recognisable characteristics such as their shape and patterning [85]. If spectral reflectance values of the target animal overlap with non-target objects in the surrounding landscape, the two can be very difficult to differentiate [86]. This makes manual detection not only time-consuming, but also highly subjective, with counts varying between annotators due to uncertainty [42]. This *inter-observer variation* means that errors associated with counts can be difficult to interpret.

In summary, the primary challenges associated with manual detection of wildlife are: i) the time required to systematically scan the imagery [22, 28, 83], ii) the presence of landscape features or challenging terrain that makes detection more complex [137], and iii) inter-observer variation in counts between observers [33]. Some of these limitations can be overcome with crowd-sourcing a large pool of annotators, however their accuracy level needs to be rigorously assessed. In addition promoting and recruiting for new crowd-sourcing campaigns, developing comprehensive training materials, and validating crowd-sourced counts can be a time consuming process. In addition as the collection frequency, affordability, and availability of data continues to increase, automated methods will be required to enable data analysis to keep pace with data collection.

2.5 Automated Data Analysis

2.5.1 Overview

When looking to extend wildlife surveys over larger areas, or to more frequent intervals, then manual analysis is a limiting factor. There is a growing need to address this barrier through automation if the methodology is to advance further [115]. One of the main challenges in automated image analysis is reducing misclassification errors, which can occur as either *false negatives*, (when an animal is present but it is not detected) or *false positives* (when other objects are incorrectly identified as animals). Automated image analysis techniques can be judged by how well they minimise these misclassification errors. To date several alternative methods for detecting and counting wildlife in remote sensing imagery have been proposed, although all are proof of concept studies conducted over relatively small spatial scales. We will summarise some of the key methods and applications below.

2.5.2 Current Approaches

2.5.2.1 Pixel Based Classification

Many studies so far have used pixel based methods to automate counts of wildlife. In these methods spectral reflectance values are established for the target animal, and are then used to segment wildlife from the background at a pixel level. Segmentation has been conducted via three main methods; manual thresholding, supervised classification, and unsupervised classification.

Thresholding is one of the simplest approaches, and involves categorising pixels into features based on their intensity value relative to a manually determined threshold value. These methods were tested on aerial imagery of geese as early as the 1980s [49]. Later Bajzak and Piatt [7] also automated counts of snow geese, which were separated from mud flat backgrounds by using their contrasting white colouration. The output of the automated method was only 2.3% different from the counts from human analysts. More recently Fretwell et al. successfully detected southern right whales by thresholding the water penetrating coastal band in VHR multispectral satellite imagery [43] (with

recall=84.6%, precision=76.2%). In the study, this simple technique outperformed unsupervised IsoData (recall=67.0%, precision=38.6%) and k-means (recall=58.2%, precision=52.0%) clustering (implemented in ENVI5 image processing software (Exelis Visual Information Solutions, Boulder, Colorado)). Despite this, these methods are non-standardised, depend on careful manual analysis of spectral histograms, and rely on a sharp contrast between wildlife and background [157].

Supervised classification also involves manually assessing spectral characteristics of objects in the image. Once a proportion of known objects are labelled, the mean and variance of the spectral signatures are used to classify the remaining pixels in the image, by training an image-processing algorithm (e.g maximum likelihood classifier). These methods have proven very effective for indirect counts, in particular for estimating emperor penguin colony sizes [8, 40]. However supervised classification methods have produced mixed results when applied to direct counts of individual animals. For example LaRue et al. [86] found the technique generated large numbers of false positives (in the order of thousands) when detecting polar bears in VHR satellite imagery. This was due to reflectance values of polar bears overlapping with non-target objects in the surrounding landscape. In general these methods are limited by the user's experience and ability to accurately label training data, by the strength of the target animals spectral signature, and by the amount of spectral overlap between different objects in the imagery [69].

Unsupervised classification schemes use statistical methods to automatically group pixels into clusters based on their spectral properties. Examples of methods used in the literature include IsoData [43, 141] and k-means [24, 141] clustering. An advantage of these methods is that they require minimal user input, aside from specifying the number of output classes, and are therefore a more standardised approach in comparison to supervised methods. Unsupervised techniques have produced reasonable results for satellite surveys of whales [43] (k-means clustering: recall=58.2%, precision=52.0%), as well as aerial counts of cattle and horses [141] (IsoData clustering: recall=82%, precision=69%), however in both cases were outperformed by other methods. In general pixel based methods have produced mixed results. In particular they do not incorporate geometric information into analysis, which could reduce misclassification errors. This has lead to more techniques employing an object based approach.

2.5.2.2 Object Based Image Analysis

Several studies have used object based image analysis (OBIA) to automate detection. OBIA methods build in geometric, contextual and textural details of objects (i.e clusters of neighbouring pixels), rather than solely focusing on spectral information encoded in

single pixels. In some cases this has improved on classification rates obtained in supervised and unsupervised pixel based methods [24]. OBIA has mostly been applied to aerial imagery, which contains more spatial detail in comparison to satellite data. For instance Groom et al. [58] employed an OBIA approach to count flamingos in aerial imagery, by using quadtree image segmentation and sequential object brightness thresholding (achieving >99% accuracy compared to human visual interpretation). Descamps et al. [30] also developed a method for detecting flamingos, by fitting ellipses around targets with specified brightness levels. The method produced counts with precision comparable to manual counts (<5% difference). As with pixel based approaches studies have shown the most promising results when the surrounding landscape is relatively homogeneous, for example Groom et al. achieved better detection rates for flamingos in water rather than on land [58].

Many OBIA methods depend on segmenting the image into classes (representing objects), which can hinder their application to satellite imagery. This is because target animals are small point like objects, which can be challenging to separate using methods such as quadtree segmentation [158]. As such some studies have developed hybrid methods which first use pixel based approaches to highlight potential wildlife, and then filter candidate objects based on factors such as shape and size. This method was employed by Yang et al. [158] to detect wildebeest and zebra in a 41-cm resolution GeoEye-1 satellite image. In the study an artificial neural network was used to classify the image at a pixel level, and then a specific rule set based on expert knowledge was developed to filter misclassifications (with an average count error of 8.2%). These methods performed well but relied on expert input and interpretation as part of the classification procedure.

2.5.2.3 Image Differencing

As discussed, one of the biggest challenges in analysing imagery is correctly distinguishing wildlife from visually similar objects in the landscape, which can cause large numbers of false positives [85, 140]. Some researchers have attempted to address this issue by using image differencing (also referred to as change detection). In these methods two images of an identical study area are taken a short time interval apart. Calculating the difference between the images highlights objects which have moved (i.e animals) and makes evident static objects which have not (i.e rocks and shadows). These methods have been investigated using both satellite [86] and aerial [110, 140] platforms. In the former two VHR satellite images (taken in different years but in the same season) were used to detect polar bears (with recall=87%), with the methods significantly reducing false positive detections in comparison to a single image supervised classification approach [86]. Aerial studies focused on horses [140] and cattle and deer [110], using same

day image differencing. For example Terletzky et al. [140] identified potential livestock by using the difference in the first principal component of two images, and then set manual thresholds to filter highlighted regions based on size. While the methods proved successful in detecting wildlife, large numbers of false positives (e.g 53% of all detections) were reported in the subsequent supervised classification scheme. This was due to mismatches in the alignment of the two images, misidentification of shadows, and the animals congregating in groups [140].

A key challenge with these methods is obtaining a precise registration between the two spatially congruent images, which can be especially difficult when using images collected at different angles, times of day and which are very high resolution [86]. Other changes in the landscape, such as melting snow [86] and moving shadows [140], can also cause errors in analysis. Ideally images should be collected by the same sensor, at the same time of day, and less than a week apart, to ensure that the changes in shadows and vegetation are kept to a minimum. However these requirements can add cost to surveys targeting this approach [86].

2.5.2.4 Thermal Imaging

An increasing number of studies have used the thermal infrared band to aid in detection, although this is currently limited to aerial surveys since VHR satellites do not sample the appropriate range of the EM spectrum. The main benefit of capturing thermal information is that it can be used to distinguish wildlife from other objects with similar spectral signatures (e.g rocks and shadows). A study of white tailed deer in aerial imagery found that applying OBIA approaches to the thermal band could successfully detect deer [24], with an average recall of 50%. The procedure worked equally well on the thermal band alone as when used in combination with the visible (RGB) bands. However the authors note that in practice using only thermal information could be problematic. Challenges are that the thermal signatures of animals change throughout the day, and at times the radiative temperatures of other objects (such as rocks and bare ground) can match those of the target animal, potentially leading to false positive detections. Additionally at some times of day low thermal contrast can decrease the likelihood of detecting animals [24]. To avoid this problem some studies have specifically collected imagery in the early morning when the temperature difference between the landscape and target animal is greatest [55]. The number of studies investigating automated detection from thermal imagery is growing, although until recently sensors have been prohibitively expensive for small scale experiments [99]. This being said promising results have been reported by researchers using machine learning and astronomical source detection software to automate detection [99].

2.5.2.5 Deep Learning

In recent years Deep Learning (DL) methods have shown impressive performance gains in computer vision tasks, in comparison to more traditional machine learning approaches [88]. In particular Convolutional Neural Networks (CNNs), architectures which are specifically designed to process image data, can be trained to learn feature extraction and classification end-to-end. They can therefore provide a more general framework than rule-based pixel and OBIA methods [150]. Different network designs have been applied to the task of wildlife detection in VHR satellite and UAV imagery. Classification networks have been used to conduct presence-absence counts of whales in VHR satellite imagery [13] (rec=0.937, prec=1.0) and turtles in UAV imagery [56] (rec=0.765, prec=0.163). In these approaches larger images are split into small tiles, and the CNN is trained to assign a binary wildlife presence/absence classification. In the applications this was successful in filtering out large areas of empty ocean, however a manual review stage was required to remove excess false positives [56]. This approach is also not suitable for dense animal aggregations, as it only determines presence-absence of wildlife not a count. Object detection CNNs, on the other hand, localise and classify animals simultaneously. The standard form for object detection labels is a bounding box, where an axes orientated box is drawn around each animal in the dataset. A popular architecture is Faster R-CNN [120], which has been used to detect koalas in thermal UAV imagery [25] (probability of detection between 68-100%) and elephants in VHR satellite imagery [33] (F2-score=0.78 in heterogeneous areas and 0.73 in homogenous areas). Segmentation methods act in a similar way to object detection, except that every pixel belonging to the target species is labelled to produce a segmentation mask. A popular choice for segmentation of VHR satellite imagery is the U-Net architecture [124], which has been applied to detect seals [54] (rec=0.253, prec=0.420), as well as other objects such as buildings [111] and trees [77]. Finally, regression networks can be trained using the total count of animals in the image, and have been applied to detect seals in aerial imagery [68] (RMSE=19.03 seals). Regression networks indirectly infer the features of interest needed to obtain the full count and are useful if point or bounding-box annotations are non-existent for each individual animal, or are too time consuming to produce.

2.5.3 Limitations of current approaches

Our literature review has shown that automated detection of wildlife in remote sensing imagery has largely been limited to small scale proof-of-concept studies. Pixel and OBIA based methods have limited success, particularly due to the small size of wildlife in satellite and UAV imagery. Background features can lead to false positives if their

spectral signature closely overlaps with the target species [86]. This has been proven to be a challenge with automated detection of species in VHR satellite imagery, for example whales [27], when using spectral and object based image analysis methods. Particularly in heterogeneous environments supervised classification methods such as maximum likelihood are unable to achieve high accuracy without a distinct spectral signature from the wildlife of interest [8, 86]. In addition, traditional machine learning methods often rely on setting specific threshold values, which can vary between satellite images even of the same species [69]. At this stage, this lack of robust, standardised and reliable automated detection methods is creating a bottleneck in remote sensing based wildlife surveys. Several reviews have highlighted the need to transfer state of the art computer vision methods to address these obstacles [69, 94, 115, 150].

As discussed in Section 2.5.2.5, the number of applications using CNNs to detect wildlife in remotely sensed imagery has expanded in recent years. In fact, a recent review showed that CNN methods have now become the predominant approach for detecting wildlife in UAV imagery [26]. As yet, these state-of-the-art methods have not been applied to surveys of wandering albatross in VHR satellite imagery, or to Abbott's boobies in UAV imagery. Given the success of CNNs in similar applications, they could provide an alternative to time consuming and costly manual review. Creating the first benchmark for automated detection using CNNs will be an important first step in catalysing research for the species. In the remaining literature review, we will outline the fundamentals of DL and CNNs, and detail the specific object detection architectures we adopt in our research application chapters.

2.6 Deep Learning for Computer Vision

2.6.1 Overview

In traditional machine learning, building a model generally requires expert domain knowledge. The data must be examined, and representative features (i.e which successfully encode patterns in the dataset) must be hand-designed and engineered. In recent years Deep Learning methods, which use *Artifical Neural Networks* (ANNs) as their main building blocks, have become increasingly popular. In contrast to traditional methods which require hand-crafted features, DL methods are designed to learn appropriate feature representations directly from the input data. This makes them a powerful and multi-purpose tool, requiring less human interpretation and expert knowledge, which can be quickly adapted and transferred across different datasets. In the following sections we will introduce the fundamentals of ANNs as well as Multi-Layer

Perceptrons (MLPs), describe model training and optimisation and lay out the foundations of CNNs for image data problems. In Section 2.10 we describe the specific CNN architectures used in this research, and detail other ML methods which we employ (Section 2.11 and Section 2.12). We conclude by describing performance metrics used to evaluate results in Section 2.13.

2.7 Introduction to Deep Learning

2.7.1 Biological Inspiration

ANNs are inspired by the network of biological neurons in the brain. While relatively little is understood about how the brain works, the core concept is to produce a simplified computational model of the neuron processes, to replicate their ability to develop knowledge through learning. The building block for ANNs was the *neuron*, a simple linear model which produced a positive or negative output given a set of inputs $(x_1, ..., x_n)$ and weights $(w_1, ..., w_n)$.

$$f(x, w) = x_1 w_1 + ... + x_n w_n \tag{2.1}$$

The neuron received inputs similarly to how neurons in the brain receive electrical signals. When signals were strong enough, they could be passed on to other neurons. The first application of computational neurons were logic gates, where a boolean function is activated as on or off given one or two binary inputs (for example AND and OR), however these were not able to learn weights from the data. That concept was developed later with the introduction of the *perceptron*.

2.7.2 Perceptrons

Perceptrons develop on neuron models by combining inputs into a weighted sum. If the weighted sum exceeds a threshold T then the neuron is triggered and produces an output y (Equation 2.2). The threshold T is referred to as the *activation function*.

$$y = \begin{cases} 1, & \text{if } \sum_i w_i x_i - T > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

In this form perceptrons can be used for binary classification, finding a linear decision boundary.

2.7.2.1 Activation Functions

The *activation function* (sometimes referred to as a transfer function) is the function that determines the output of a node. It maps the resulting values into a range, for example between 0 and 1 or -1 to 1, depending on which activation function is chosen.

Initial perceptron models used non linear functions, such as the sigmoid function. Sigmoid outputs a value between zero and one with an s shaped distribution, according to the equation:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

Since sigmoid outputs values between 0 and 1, it is useful for predicting probabilities.

Tanh outputs a similar s-shaped distribution to sigmoid, but over the range -1 to +1, according to the equation:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.4}$$

The advantage of using Tanh activation is that negative values will be mapped strongly negative, and zero inputs will be mapped near to zero. It is therefore a good choice for classifying between two classes.

Most deep learning papers found that a *Rectified Linear Unit* (ReLU) is effective, and converges better with Stochastic Gradient Descent [53]:

$$f(x) = max(0, x) \tag{2.5}$$

An issue with ReLU is that any negative value will be mapped immediately to zero, which can affect mapping negative values appropriately. To combat this Leaky ReLU function can be used:

$$f(x) = \begin{cases} ax, & \text{if x<0.} \\ x, \end{cases} \tag{2.6}$$

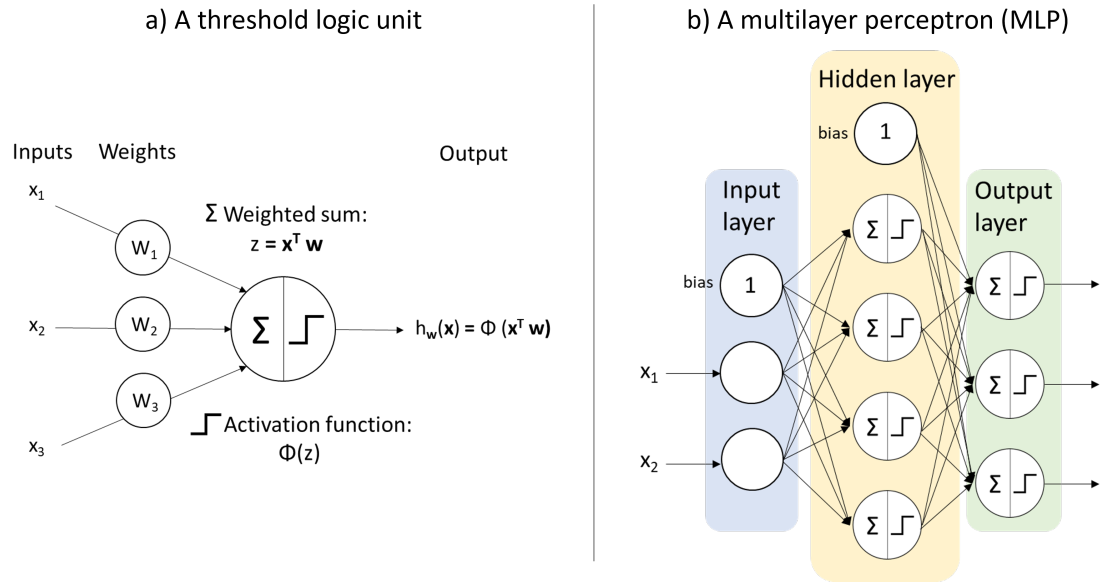

FIGURE 2.1: a) A threshold logic unit, an artificial neuron which computes a weighted sum of inputs and then applies an activation function to get an output. b) An example MLP architecture with two inputs, one hidden layer made up of four neurons, and three output neurons.

2.7.3 Multilayer Perceptrons

In simple perceptron models, the neuron receives inputs and picks and initial set of weights at random. These are combined in a weighted sum, and then ReLU determines the value of the output. Perceptrons use SGD to find the set of weights that minimise the distance between the misclassified points and the decision boundary. However, while perceptrons have the ability to learn weights, having only a single neuron restricts them to only linear data. To deal with cases where the mapping between inputs and output is non-linear, *Multilayer Perceptrons* (MLPs) were introduced. MLPs essentially work by stacking neurons into multiple layers, including an input layer, an output layer, and any number of intermediate hidden layers (an example MLP architecture is presented in Figure 2.1b). These networks are described as *feed forward*, as each linear combination of weighted sums is fed forward onto the subsequent layer of the MLP.

There are several hyperparameters that can be adjusted when designing an MLP architecture, including the number of hidden layers, the number of neurons per layer, and the choice of activation function. Training MLPs, in other words allowing the network to learn the weights which minimise the loss function, is done through *backpropagation*.

2.7.3.1 Backpropagation

The backpropagation algorithm is used to train artificial neural networks, by iteratively updating the weights and biases in order to minimise the error between the predicted output and the ground truth. The backpropagation algorithm consists of two main steps: forward propagation and backward propagation.

In the forward propagation step, the inputs are passed through the network and the outputs are predicted. This is done by computing the dot product of the inputs and the weights, followed by the application of an activation function, which is used to introduce non-linearity into the network. This process is repeated for each layer in the network, until the final output is produced.

The backward propagation step is used to adjust the weights and biases of the network in order to reduce the error between the predicted output and the true output. This is done by computing the derivative of the error with respect to each weight and bias in the network, using the chain rule. The weights and biases are then adjusted in the opposite direction of the gradient, which is the direction of steepest descent.

An outline of the backpropagation algorithm goes as follows:

1. Initialize the weights w and biases b of the neural network randomly.
2. For each sample in the training set:
   - Feed the input data x through the network to produce the predicted output y_{pred}.
   - Calculate the error E between the predicted output y_{pred} and the true output y using a loss function.
   - Propagate the error back through the network using the chain rule to compute the derivative of the error with respect to each weight w and bias b:

$$\begin{aligned} \frac{dE}{dw} &= \frac{dE}{dy_{pred}} * \frac{dy_{pred}}{dw} \\ \frac{dE}{db} &= \frac{dE}{dy_{pred}} * \frac{dy_{pred}}{db} \end{aligned} \tag{2.7}$$

   - Update the weights and biases in the opposite direction of the gradient, using the learning rate alpha to control the size of the update:

$$\begin{aligned} w &= w - \alpha * \frac{dE}{dw} \\ b &= b - \alpha * \frac{dE}{db} \end{aligned} \tag{2.8}$$

3. Repeat step 2 for multiple epochs until the error is minimized.

2.8 Neural Network Training

2.8.1 Loss Functions

Training ANNs with the backpropagation algorithm is an optimization process driven by a loss function, which quantifies the difference between the network prediction and the known ground truth. The aim of the optimization is to minimise the error as determined by the loss function. There are different types of loss functions that can be used depending on the task, for example whether the network is being used for classification or regression.

2.8.1.1 Cross Entropy Loss

For classification tasks, where the output is a class label, a common loss function is the cross-entropy loss (also known as the log loss). If N is the number of classes, y_i is the true class label, and $\hat{y_i}$ is the predicted probability of the ith class, then the cross-entropy loss is given by the equation:

$$CE = -\sum_{i=1}^{N} y_i log(\hat{y_i}) \tag{2.9}$$

The cross-entropy loss can be computed for multi-class classification tasks, where the predicted output is a probability distribution over all classes. In this case, the loss is the negative log likelihood of the true class, given the predicted probability distribution.

In the case of binary classification tasks, where there are only two classes, the cross-entropy loss is often referred to as the binary cross-entropy loss. It is given by the equation:

$$BCE = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \tag{2.10}$$

2.8.1.2 Focal Loss

The focal loss is a loss function that was introduced as an alternative to the cross-entropy loss for classification tasks, with the goal of addressing the issue of class imbalance in

the training data [92]. It was originally developed for object detection tasks, where it is common to have a large number of negative examples (background pixels) compared to positive examples (pixels belonging to objects). This can lead to the model learning to classify most examples as negative, leading to poor performance on the positive class. To address this issue the focal loss down-weights the loss contribution for easy examples and up-weights the contribution for rare and hard to classify examples.

The focal loss is given by the equation:

$$FL = -\alpha_t(1 - p_t)^\gamma \log(p_t) \tag{2.11}$$

where p_t is the predicted probability of the true class, α_t is a weighting factor for the loss of each example, and γ is a tunable hyperparameter that controls the rate at which the loss is down-weighted for easy examples.

The weighting factor α_t is defined as follows:

$$\alpha_t = \frac{1}{N}\left(\frac{1 - p_t}{p_t}\right)^\beta \tag{2.12}$$

where N is the number of examples and β is a hyperparameter that controls the rate at which the loss is up-weighted for hard examples.

2.8.1.3 Mean Squared Error

For regression tasks, where the output is a continuous value, common loss functions include the mean squared error (MSE), given by the equation:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.13}$$

The MSE is sensitive to outliers and tends to be affected by large errors. As an alternative the mean absolute error (MAE) can be used, which is affected more equally by all errors. The MAE is given by the equation:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.14}$$

2.8.2 Optimisers

Optimisers are algorithms that find the value of model parameters that minimise an objective function, by performing iterative updates. One round of iterative updates over the entire training dataset is called an *epoch*. Optimisation involves searching for the optimal (generally the minimum value) within the surface created by the objective function. In convex problems the solution space is strictly convex and so there is only one solution which is the global minimum. However in many cases the solution space is non-convex - made up of many local mimima. The challenge of then is to avoid becoming trapped in non-optimal local minimum solutions.

2.8.2.1 Gradient Descent

Gradient descent is one of the most popular choices of optimisation algorithm to train neural networks. Gradient descent is a method for minimising an objective function $J(\theta)$, where θ are the model's parameters. Optimisation is performed by updating the parameters in the opposite direction to the gradient of the objective function $\nabla_\theta J(\theta)$, with respect to the model parameters. The *learning rate* η is a hyperparameter which determines the step size taken to reach a (local) minimum.

There are three variants of gradient descent, with the main difference being the number of training data samples used to compute the gradient of the objective function. The main trade off is between accuracy and the time taken to perform an update. The three variants are summarised below.

Batch Gradient Descent: Computes the gradient of the objective function w.r.t θ for the entire training dataset. The update rule is:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta) \tag{2.15}$$

Since gradients for the entire dataset must be computed before a single update can be performed, batch gradient descent is slow and costly on memory. It is guaranteed to converge on the global minimum for convex error surfaces, and to a local minimum for non-convex surfaces.

Stochastic Gradient Descent (SGD): Computes gradient and performs an update for each training sample $x^{(i)}$ and ground truth label $y^{(i)}$ in the dataset. The update rule is:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)}) \tag{2.16}$$

Since parameters are updated for each new data sample it is much faster than batch gradient descent. The frequent, high variance updates lead to large fluctuations in the objective function, which means that SGD has the ability to jump to new local minima, however this has the cost of complicating convergence to an exact minimum. Gradually decreasing the learning rate η can reduce this issue.

Mini-batch Gradient Descent: Computes the gradient and performs an update for every mini-batch of n training samples:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i+n)}) \tag{2.17}$$

This has the effect of reducing the variance in parameter updates, while also making training more efficient. For these reasons mini-batch gradient descent is the typical choice for neural network training, and tends to be referred to simply as SGD even if mini-batches are used.

Although a popular choice there are there are some challenges in determining the learning rate when training with mini-batch SGD. If the learning rate is too small then convergence is slow, too large and the objective function can fluctuate and not converge to a minimum. Learning rate schedules (where the learning rate is decreased at designated epochs) can alleviate this issue, but must be manually specified in advance and do not adapt based on the characteristics of the training data. Therefore various gradient descent optimisation algorithms have been developed to overcome these challenges. One of the most popular and best performing methods is the Adaptive Moment Estimation (Adam) optimiser [76].

2.8.2.2 Adaptive Moment Estimation (Adam)

While optimisers like SGD maintain a single learning rate throughout the training session, Adaptive moment estimation (Adam) [76] updates the learning rate for each parameter θ. Adam stores an exponentially decaying average of past gradients m_t, as well as past squared gradients v_t, calculated respectively as follows:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \tag{2.18}$$

where g_t is the gradient at time t and β_1 and β_2 are hyperparameters which control the decay rate for the moving averages.

Since m_t and v_t are initialised as vectors of zeros, they bias towards zero, particularly in initial time steps and when decay rates (set by β_1 and β_2) are small. To counteract

this bias-corrected first and second order moment estimates are calculated:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t} \end{aligned} \tag{2.19}$$

These are used to update the parameters using the Adam update rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{2.20}$$

where η is the learning rate and ϵ is a small constant that is added to the denominator to prevent division by zero. The authors propose default values of 0.9 for β_1, 0.999 for β_2, and 10^{-8} for ϵ, and show empirically that Adam compares favourably to other adaptive learning-method algorithms [76].

2.8.3 Regularisation

Regularisation is a method for preventing overfitting a model on the training data samples, generally by lowering the complexity of a neural network during training. Regularisation techniques used in this thesis are outlined below.

2.8.3.1 L1 and L2 Regularisation

In L1 and L2 regularisation the loss function is extended by a regularisation term, which reduces the magnitude of network weights. Smaller weights reduce the impact of the hidden neurons, hence the overall complexity of the neural network is reduced. Choosing the regularisation term is important, as too high will lead to underfitting, and too low will risk overfitting.

In L1 regularisation, also known as Lasso regularisation, the regularisation term is proportional to the absolute value of the weights. The objective function with L1 regularisation is given by the equation:

$$L1 = \sum_{i=1}^{N} L_i + \lambda \sum_{j=1}^{M} |w_j| \tag{2.21}$$

where N is the number of examples, M is the number of weights, L_i is the loss function for the ith example, w_j is the jth weight, and λ is a hyperparameter that controls the strength of the regularisation.

In L2 regularisation, also known as Ridge regularisation, the regularisation term is proportional to the squared value of the weights, and is given by the equation:

$$L2 = \sum_{i=1}^{N} L_i + \frac{\lambda}{2} \sum_{j=1}^{M} w_j^2 \tag{2.22}$$

2.8.3.2 Dropout

In addition to L2 and L1 regularisation *dropout* can be employed to minimise the risk of overfitting within the network design [64]. Dropout schemes work by randomly excluding a percentage of neurons from the network for each training sample. Individual neurons will be dropped with probability p, or kept with probability $1-p$. This leaves a different reduced sub-network for each weight update step. The aim is to prevent co-dependence between neurons, where sets of feature detectors might only work well in combination, reducing their power to perform individually. Empirically this has shown to be a simple and effective method in comparison to other regularisation techniques [136].

2.8.4 Transfer Learning

CNN architectures generally perform best when provided with large amounts of training data. However in many real world applications, including remote sensing of wildlife, it is not possible to collect sufficiently large datasets to train CNNs from scratch. In this case transfer learning can be applied. In this process networks are pre-trained on large image datasets containing millions of annotated samples, commonly ImageNet [29] for classification tasks, and COCO [93] for object detection. Initialising networks weights in this way means that they have already learnt to extract general, low-level features which are common across all images. Using this as a starting point, the network can then be fine tuned to the specific task with the users own dataset.

Benefits of transfer learning are that pre-trained networks can be fine-tuned with smaller amounts of data, which is essential when labelled training data is scarce. Networks can also be trained more quickly, and with a higher learning and accuracy rates. On the other hand, caution should be used to avoid *negative transfer*, when learning from the previous task instead hinders the new model. This can occur when source and target datasets are too dissimilar, meaning the weights transferred from the source task are not able to adapt to the new dataset.

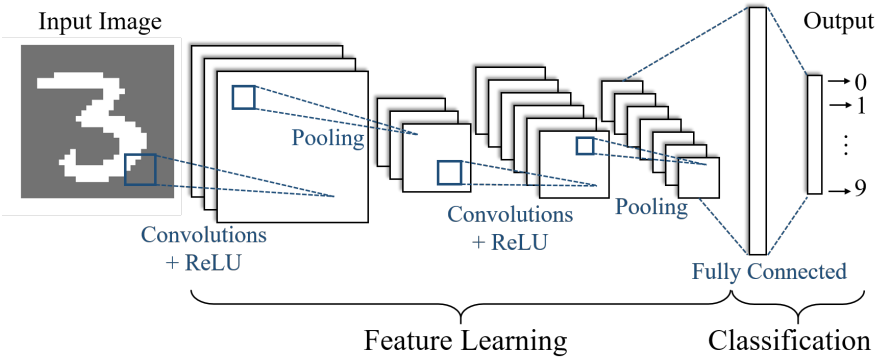

FIGURE 2.2: An example CNN architecture, using the MNIST hand-drawn digit dataset as an example.

In practice transfer learning is performed by adapting the final classification layers of a network suitable for the new task. Model performance can often be improved by *fine-tuning*, where all or parts of the base model are unfrozen and retrained with a low learning rate.

2.9 Convolutional Neural Networks

CNNs are a branch of ordinary Neural Networks which are specifically adapted to deal with array inputs, such as images. CNN architectures are designed to take advantage of the 2D structure of images, through use of shared weights, local connections, pooling and multiple layers [88]. This has the benefit of producing translation invariant features, as well as a reduced number of parameters and quicker training times in comparison to their equivalent fully-connected counterparts.

CNNs are generally structured as a series of layers, with initial layers performing feature learning and final layers performing classification. The feature learning stage typically consists of alternate *convolution* and *pooling* operations, while the final classification stage is carried out by one or more fully connected layers and a classifier such as *softmax*, as shown in Figure 2.2. The most common approach to training CNNs is through supervised training schemes, using *backpropagation* with an optimiser (such as Stochastic Gradient Descent (SGD)). In this method *weights* and *biases* are updated in order to minimise a *loss function*, which quantifies the difference between the output predicted by the network and the desired target output. In the following sections we detail each of these stages individually.

2.9.1 Inputs and Augmentation

As discussed CNNs are primarily used to process image inputs in the form of arrays. For panchromatic images this could be a simple 2D array, where each element (or pixel) records the intensity value at that location. Conventional colour images are stored as 3D arrays, one 2D array for each of the three colour channels (red, blue and green). In some cases images may be made up of more than three channels, for instance many multispectral satellite sensors record numerous multispectral bands. To implement a supervised training scheme, each of these arrays should also have a ground truth label. For instance in the MNIST dataset [87] each hand-drawn digit is labelled with the true value drawn in the image patch. As we will see these labels are needed for backpropagation and training.

In some cases, data availability can be a problem. In order to train a robust model it is important to build a training set containing objects in multiple orientations, scales and light conditions. This allows the model to generalise as best as possible to unseen images. In practice, obtaining all of these combinations from real life data can be difficult, so frequently training schemes include an augmentation step to artificially simulate these variations. For instance images may be flipped on their axes, warped, or have their brightness and saturation adjusted [78]. This helps to build a balanced dataset and reduces the chance of *overfitting*; when the network is tailored to work well on the training set but not able to generalise well when processing new data.

2.9.2 Automatic Feature Extraction

2.9.2.1 Convolution

CNNs are built around the process of convolution. Convolutional layers consist of a set of learnable filters of small *height* and *width*, and a *depth* corresponding to that of the input array. The dimension of the filter is referred to as its *receptive field*. In a convolution step the filter passes over the input array in a sliding window fashion, and performs a matrix dot product, producing an output feature map. An example is presented in Figure 2.3 for demonstration. Intuitively, filters act as feature detectors for the output of the previous layer. All the weights in the filter are learnt, and depending on the values chosen can perform different operations such as edge detection, blurring or sharpening. Since the same filter is used to generate a single feature map, convolution has the important property of being translation invariant. This means that it does not matter where an object is located in an image, only that it exists.

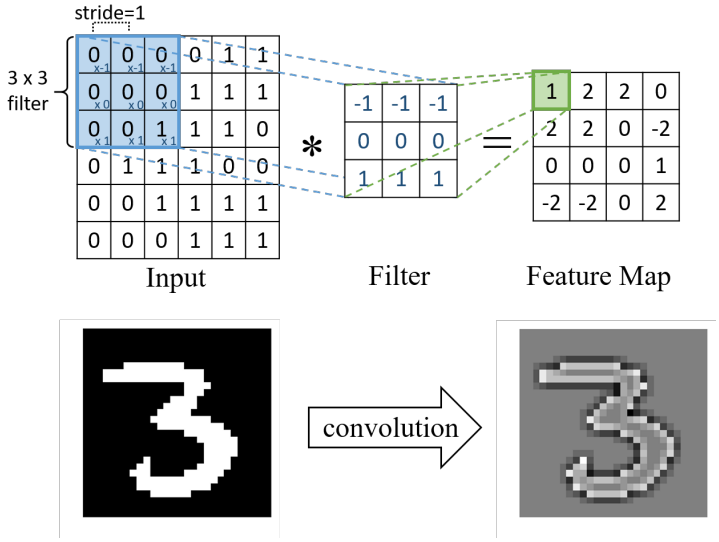

FIGURE 2.3: The convolution operation, using a 3×3 filter with stride 1. The lower images show an example input and output of a convolution operation using an edge detecting filter.

While the weights in the filters are learnt in the training phase, there are four additional hyperparameters which should be specified in the initial design of the network. These are:

- **Filter size**: The height, width and depth of the convolution filters. Often height and width are set to three or five. The depth should match the number of bands in the input array.
- **Filter Number**: The number of different filters in a single layer. This determines how many unique filters should be learnt, and can differ between layers. For example in Figure 2.2 the first convolutional layer uses three filters, the second uses six.
- **Stride**: The number of pixels by which we slide the filter over the input array (for example stride of one means moving the filter one pixel at a time, as in Figure 2.3). The larger the stride the smaller the output feature map.
- **Padding**: Since the convolution operation is unable to generate values at the edges of an input array, output feature maps are of reduced dimension. To combat this the input array can be padded at the edges prior to convolution, in order to control the image size. A common choice is zero-padding, where zeros are are added around all borders of the array. Other options include reflection, mean and constant padding.

2.9.2.2 Pooling

While convolutional layers detect local features in the output of the previous layer, the role of pooling layers is to aggregate these features into lower resolution representations. This is done through pooling operations, where small and typically disjoint neighbourhoods of the image are aggregated into a single value, in a sliding window fashion. The most common choice for CNNs is max-pooling, shown in Figure 2.4, which has shown superior performance compared to other approaches such as average or sum pooling. Two hyperparameters for pooling layers should also be specified in the architecture design; filter size and stride. Most CNNs use a 2×2 filter with stride of 2 (as in Figure 2.4), although in some cases an overlapping pooling operation (e.g filter size 3×3, stride 2) can be employed [78].

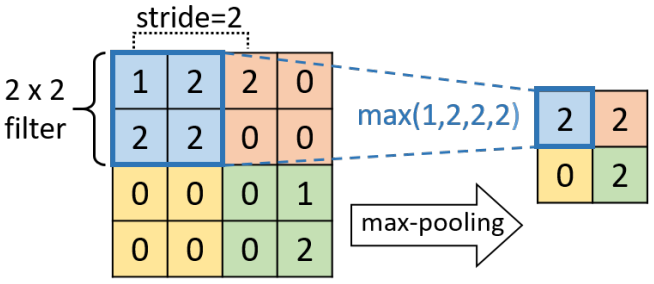

FIGURE 2.4: The max-pooling operation, with a 2×2 filter and stride of 2.

The role of pooling is to reduce the number of parameters and computation required in the network, and also make feature representations invariant to small shifts, distortions and transformations [88]. Together the convolution, ReLU and pooling layers extract features which are robust to changes in scale and translation, and can be modelled non-linearly.

2.9.3 Classification

2.9.3.1 Fully Connected Layers

The classification stage of a CNN generally consists of one or more fully connected layers. These act as traditional multi-layer perceptrons (MLPs), where every neuron in the layer is connected to every neuron in the preceding layer. They take the high level feature representations extracted in the convolution and pooling stages, and use them to classify input images into classes based on the training dataset. For example in the MNIST dataset (Figure 2.2) there are ten possible classes, the integers zero to nine.

In classification problems the outputs of fully connected layers are generally fed into an activation function, usually *softmax*, to convert them into a vector of probabilities for each class. Softmax takes a vector of scores $(x_1, ..., x_K)$ and converts them into probabilities which sum to one, by the following equation:

$$\text{Softmax}(x_j) = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \qquad \text{for } j = 1, ..., K \tag{2.23}$$

Since the dimension of the input feature maps is smaller than the dimension of the original image (i.e there are less neurons to connect), this architecture is more computationally efficient than implementing a MLP directly. Fully connected layers also have the co-benefit of being able to learn the best non-linear combinations of the input feature maps. For example a combination of features may produce better classification results than a single feature alone.

2.9.4 Object Detection

CNNs can be used for a number of tasks in computer vision, not simply for classifying input images into a given class (a summary of four main tasks is presented in Figure 2.5). *Object detection* involves both classifying and localising multiple objects of different classes in an input image. Generally this involves locating objects using a *bounding box*, which specifies the four corner coordinates which precisely outline the object of interest. Each bounding box will have a predicted class label and confidence score for the detection. Until recently this could be simply achieved by passing a CNN trained for classification over an image in a sliding window, however more recently specific object detection architectures have been designed.

Object detection networks fall into two general categories: two-stage detectors and single-stage detectors. Two-stage CNNs are designed to first generate region proposals and then classify the objects in the region proposals. They consist of two main components: a region proposal network (RPN) that generates region proposals, and a classification and regression network that classifies the region proposals and refines the bounding box coordinates. Popular examples of two-stage detectors include Faster R-CNN [120] (see Section 2.10.2) and R-FCN (Region-based Fully Convolutional Network). On the other hand single-stage CNNs are designed to predict the locations and class labels of objects in the input image or video directly, without generating region proposals. Examples of single-stage CNNs for object detection include YOLO (You Only Look Once) [119] and SSD (Single Shot Detector) [95]. Single-stage detectors are typically faster and more efficient than two-stage CNNs, but may have lower accuracy.

Therefore single-stage CNNs may be a good choice for tasks that require fast inference speed, while two-stage CNNs can suit applications which require high accuracy.

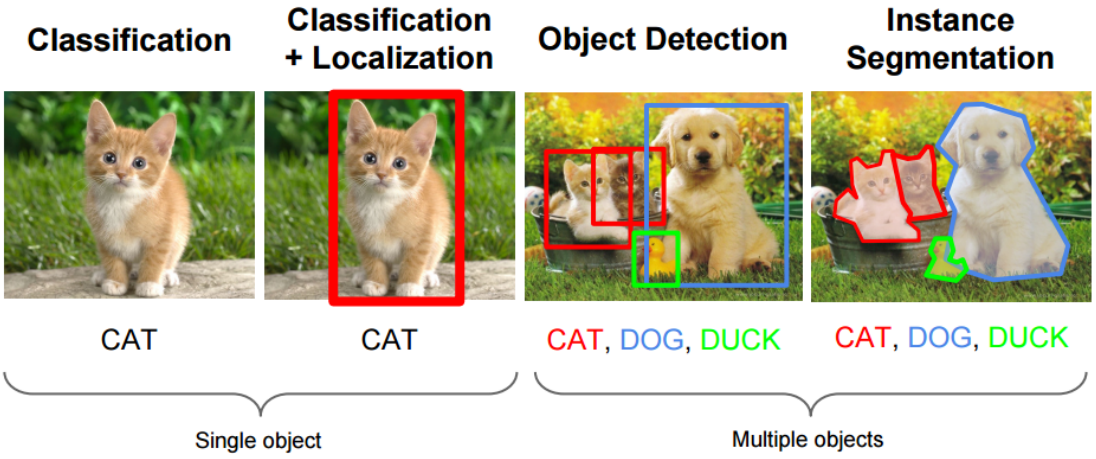

FIGURE 2.5: A comparison of the outputs for image classification, localisation, object detection and instance segmentation. Source: Fei-Fei Li, Andrej Karpathy & Justin Johnson (2016) cs231n, Lecture 8 — Slide 8, Spatial Localization and Detection (01/02/2016). Available: http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf

2.9.5 Image Segmentation

While both single and two-stage detection CNNs predict bounding boxes around objects, in segmentation tasks a class label is inferred for every pixel contained within an object. This allows for precise delineation of object outlines (Figure 2.5). To train segmentation methods ground truth outlines or *masks* should be provided, which show the target class label for each pixel in the image. One of the primary challenges in segmentation tasks is that when images are passed through CNNs they gradually lose their spatial resolution (due to the repeated strides and pooling). To get a precise mask as an output, Long et al. [97] proposed using a fully connected network, where the image is first passed through a CNN and then processed through *upsampling* layers to regain it's original spatial resolution. Instead of using upsampling methods such as bilinear interpolation, these upsampling layers can be learnt from the data using *deconvolution*. These can be thought of as regular convolution layers with a fractional stride (e.g of 1/2), and are commonly employed in image segmentation methods.

Segmentation tasks can be divided into *semantic* and *instance* segmentation. In semantic segmentation every instance of the same class is treated as the same entity. This means if there are two examples of the same object in an image which overlap, they will not be identified as separate objects. Popular semantic segmentation methods included fully-connected CNNs such as U-Net [124], which was originally designed for biomedical

image segmentation. In instance segmentation multiple objects of the same class are each treated as their own distinct instances. One of the most popular instance segmentation methods is Mask R-CNN [62], which is an extension of the Faster R-CNN [120] detection network.

2.10 Deep Learning Architectures Used

2.10.1 U-Net

U-Net is a CNN architecture used for semantic segmentation, which was originally designed for biomedical image analysis [124]. Broadly speaking U-Nets consist of an encoder and a decoder path, connected by a series of skip connections (Figure 2.6). The encoder follows the typical architecture of a CNN, applying repeated convolutions, activation, and max pooling to extract features from input images. The decoder consists of a series of upsampling and convolutional layers that use the feature maps extracted by the encoder to generate a segmentation mask for the input image. The skip connections between the encoder and decoder layers allow the U-Net to incorporate both high-level semantic information from the encoder and low-level spatial information from the decoder. The skip connections are implemented by concatenating the feature maps from the corresponding layers in the encoder and decoder. This allows for precise localization of classified pixels. U-Net has shown impressive performance in a number of tasks [35], in particular segmentation of VHR satellite imagery.

U-Net is typically trained using a supervised learning approach, where the model is provided with a set of labeled images and their corresponding segmentation masks. The model is then optimized to minimize a loss function, such as the binary cross-entropy loss, that measures the discrepancy between the predicted segmentation masks and the true masks. There are a number of hyperparameters which can be adjusted in the U-Net architecture. For example in the convolution blocks the number of filters, kernel size, pooling size and stride length can be adjusted. Similarly the size of the kernel used in the upsampling layers can be altered. Different activation functions, regularisation parameters, loss functions, learning rates and optimisation algorithms can be tested. In addition the size of the network (how many layers it has), can be compared for performance.

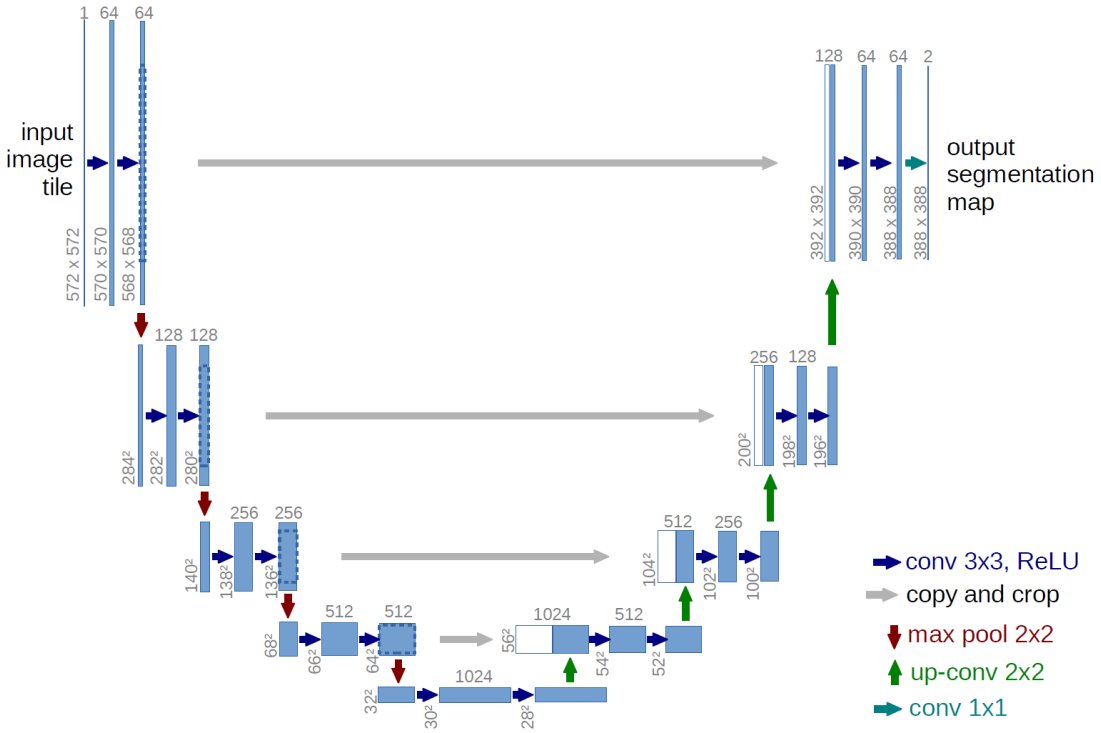

FIGURE 2.6: The original U-Net architecture, designed for biomedical image segmentation. Image taken from original paper [124].

2.10.2 Faster R-CNN

Faster R-CNN is a two-stage object detection network which was developed by Ren et al. in 2015 [120]. Two-stage detection networks have a separate module to generate region proposals as a first stage, which they then classify and localise in the second stage. It essentially solves the object detection task as a classification problem, by being presented with proposals and classifying them into either object or background.

Faster R-CNN has four main components: i) a backbone CNN for classification and feature map generation, ii) a region proposal network (RPN) for generating Regions of Interest (RoI), iii) an RoI pooling layer to make feature vectors from RoIs, and iv) a final classification and regression stage, which is used to find the location of each object and it's class label. The output is a bounding box with predicted class label, bounding box corner coordinates, and a confidence score for the prediction. A general overview of the architecture is presented in Figure 2.7.

The first step of Faster R-CNN is to pass input images through a CNN pretrained for the task of classification (e.g. using ImageNet) to generate a feature map. The original Faster R-CNN used ZF and VGG pretrained on ImageNet [121], however since then lots of different networks have been developed which offer improved performance. Common alternatives include ResNet [63] (a deep residual CNN which can have different numbers

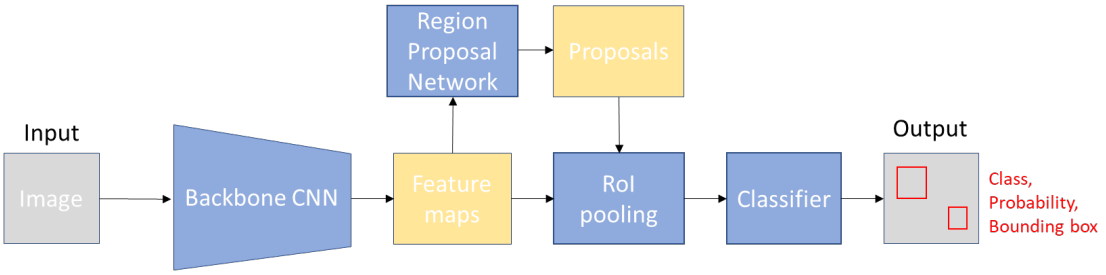

FIGURE 2.7: Overview of the Faster R-CNN architecture.

of layers, e.g 50, 101, 152) and MobileNet [71] (a lightweight CNN which is suitable for mobile devices). The feature map generated in this stage is then passed through to the RPN.

2.10.2.1 Region Proposal Network

The RPN works on the feature map output by the last convolutional layer of the backbone CNN. This developed on previous selective search methods used in the R-CNN [52] and Fast R-CNN [51] networks (where RoI's were input at the pixel level rather than the feature level) making the network much faster. A sliding window is passed over the feature map and for each window k candidate region proposals are generated (Figure 2.8). Each proposal is parameterised by an *anchor box* which has a given *scale* and *aspect ratio*. Generally 3 scales and 3 aspect ratios are used, leaving a total of $k = 9$ region proposals, however other values can be chosen.

Each of the generated region proposals are then converted into a feature vector (the size of the feature vector depends on the backbone CNN, in the original paper it was of length 256 for the ZF net and 512 for the VGG-16 net [121]). This vector is passed into two separate fully connected layers:

- A classification (cls) layer: A binary classifier that generates an *objectness score* for each region proposal. This layer outputs two predictions per anchor: the score for it being background, and the score for it being foreground.
- A regression (reg) layer: Which outputs a 4D vector defining the bounding box of the region (as $[x_{center}, y_{center}, width, height]$).

During training, each anchor is classified as either foreground or background based on the objectness score. The objectness score depends on the Intersection over Union (IoU) of the anchor with a ground truth box (IoU is defined in Section 2.13.1.1). Thresholds are defined as i) foreground: when anchors overlap with the ground-truth object with

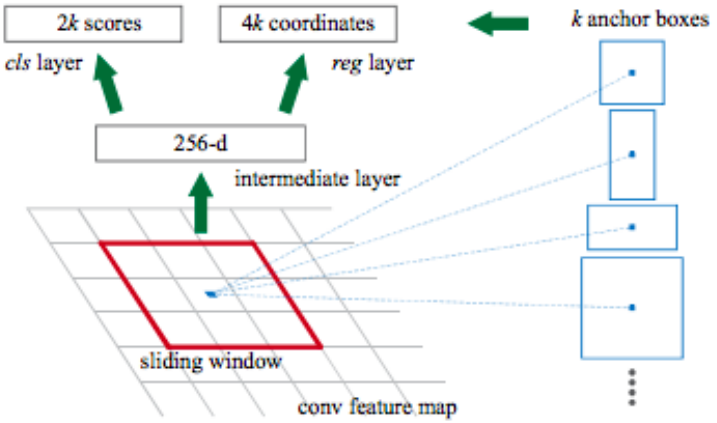

FIGURE 2.8: Overview of the region proposal network (RPN). Figure reproduced from the original paper [120].

an IoU greater than 0.5, and ii) background: when anchors do not overlap any ground truth object or have less than 0.1 IoU with ground truth objects. The anchors are then randomly sampled to form a mini batch size of 256, while trying to maintain a balanced ratio of foreground and background anchors. The RPN uses all anchors in the mini batch to calculate the classification loss (L_{cls}) using binary cross entropy (Equation 2.10), while only the anchors marked as foreground are used to calculate the regression loss.

To calculate the targets for regression the Δ needed to transform the foreground anchor on to the object is computed. If there is no foreground anchor, then the anchor with the biggest IoU with the ground truth box is selected. Since anchors will overlap Non-Maximum Suppression (NMS, outlined in Section 2.10.2.4 below) is performed to delete those with lower IoU. Those with IoU greater than 0.7 are classed as a positive detection and those with IoU less than 0.3 are classed as background. The top N proposals (after being sorted by confidence score) are selected after NMS, and the regression loss is calculated with the smooth L1 loss. If t_i is the bounding box coordinates of the i_{th} anchor and t_i^* is the ground truth coordinates, then the regression loss is defined as:

$$L_{reg}(t_i, t_i^*) = \sum_{i \in \{x,y,w,h\}} smooth_{L1}(t_i - t_i^*) \tag{2.24}$$

where

$$smooth_{L1} = \begin{cases} 0.5x^2 & \text{if } |t_i - t_i^*| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{2.25}$$

2.10.2.2 Region of Interest Pooling

The RPN stage outputs a set of object proposals, but no assigned class labels. In the RoI pooling stage these proposals are processed ready for classification into their predicted class. In Faster R-CNN the existing convolutional feature map is cropped using each proposal, and then resized to $14*14*convdepth$ using interpolation. Finally max pooling with a $2x2$ kernel is used to get a final $7 * 7 * 512$ feature map for each proposal. These dimensions match those used by the final R-CNN classification stage, however can be adapted if a different architecture is employed at the final stage.

2.10.2.3 Region-based Convolutional Neural Network

The final R-CNN stage (which is the Fast R-CNN architecture described in [51]) flattens the feature maps from RoI pooling into one-dimensional vectors, and connects them to two fully connected layers with ReLU activation. The final fully connected layer is used to classify the proposals into one of N classes (with $+1$ being the background class). In parallel, a second fully connected layer with $4N$ units ($\Delta_{center_x}, \Delta_{center_y}, \Delta_{width}, \Delta_{height}$) is used for bounding box regression.

The Fast R-CNN is trained with backpropagation and Stochastic Gradient Descent. The loss function is defined as:

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda.[u \geq 1]L_{reg}(t^u, v) \tag{2.26}$$

where p is the object possibility, u the classification class, t the ground truth label, v the ground truth coordinates for class u, L_{cls} is the loss function for classification and L_{reg} is the loss function for bounding box regression, and λ is a balancing parameter. The L_{cls} is defined to be:

$$L_{cls}(p, u) = -log(\frac{e^{p_u}}{\sum_{j=1}^{K} e^{p_j}}) \tag{2.27}$$

where p is the object possibility, u the classification class, L_{cls} the loss function for classification, and K the number of classes. L_{reg} is given in Equation 2.24 using t^u and v as inputs.

Objects are first classified and then bounding boxes are adjusted by selecting the class with the highest probability for a given proposal. Those assigned as background class are ignored, and class-based NMS is applied to the final set of predicted objects. Finally a probability threshold is set to reduce the final number of returned objects.

In the complete Faster R-CNN model there are two losses for the RPN and two for the R-CNN, these four losses are combined using a weighted sum. Weight adjustments can be made to give classification losses more weight compared to regression, or to give R-CNN losses more influence than the RPN losses. Hyperparameters which can be altered in the Faster R-CNN architecture include the selection of the backbone CNN, the batch size, learning rate, optimiser and weight initialisation. Different IoU thresholds can also be set for the RPN and R-CNN stages and NMS thresholds (e.g the number of proposals to keep after applying NMS).

2.10.2.4 Non-maximum suppression (NMS)

When multiple overlapping bounding boxes are predicted for the same object (which could be treated as false positives when assessing results) then *non-maximum suppression* can be used. In non-max suppression all proposed bounding boxes are sorted by confidence score, then the following selection process is applied:

1. Select the proposal with the highest confidence score and add it to the proposal list.
2. Calculate the IoU between the selected proposal and every other proposal. If the IoU is greater than the threshold N (e.g 0.5), remove the proposal as a candidate.
3. If there are still proposals left, go back to step one and repeat, else return the list of filtered proposals.

2.11 Machine Learning Algorithms Used

2.11.1 Support Vector Machine

Support Vector Machines (SVMs) are supervised machine learning methods which can be used for classification and regression, which we employ for binary classification in Chapter 6 of this thesis. In the classification case SVMs find a hyperplane (or set of hyperplanes for multi-class problems) which maximises the margin between the separating hyperplane and the training data points (Figure 2.9). The margin is the distance between the hyperplane and the nearest data points, with a larger margin indicating a more robust model.

In some cases, it may not be possible to find a hyperplane that perfectly separates the classes. In these cases, the SVM can be modified to allow for some examples to be

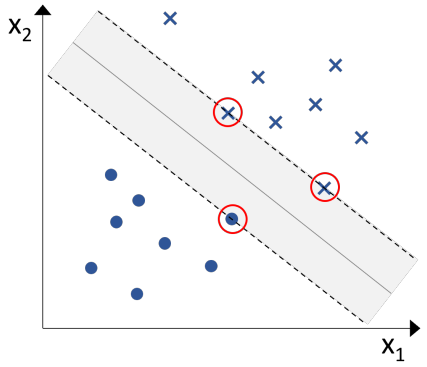

FIGURE 2.9: SVMs find a hyperplane which maximises the margin between classes. Points which fall on the margin are called support vectors(circled in red).

misclassified, using a technique called *soft margin*. The soft margin SVM optimization problem introduces a slack variable ξ_i for each example, which allows the model to penalize misclassified examples. The optimization problem can be written as:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{2.28}$$

subject to the constraints:

$$\begin{aligned} y_i(w^T x_i + b) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \end{aligned} \tag{2.29}$$

where C is a hyperparameter that controls the trade-off between the margin and the number of misclassified examples.

When data is not linearly separable the *kernel trick* can be employed to allow the SVM to learn nonlinear decision boundaries. It works by mapping the input data into a higher-dimensional space (where a linear decision boundary can be learned) using a *kernel function*. A kernel function $K(x, y)$ measures the similarity between two data samples x and y, and returns a scalar value indicating how similar they are. By replacing the dot product $x^T y$ in the optimisation with $K(x, y)$, the model can learn a nonlinear decision boundary by implicitly mapping the input data into a higher dimensional space.

A number of different kernel functions can be used in SVMs, including:

- The Polynomial kernel:

$$K(x, y) = (x^T y + c)^d \tag{2.30}$$

  where c is a hyperparameter that controls the shift of the kernel, and d is a hyperparameter that controls the degree of the polynomial.

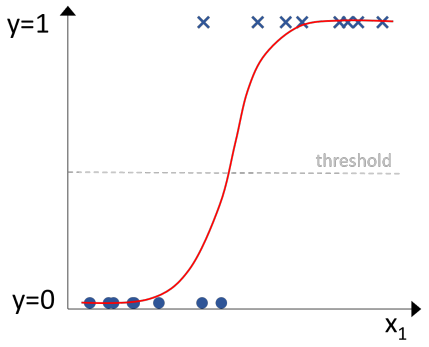

FIGURE 2.10: Logistic regression is used for binary classification, and finds the line which best separates the two classes.

- The Radial basis function (RBF) kernel:

$$K(x, y) = \exp\left(-\frac{|x - y|^2}{2\sigma^2}\right) \tag{2.31}$$

  where σ is a hyperparameter that controls the width of the kernel.

- The Sigmoid kernel:

$$K(x, y) = \tanh\left(\kappa x^T y + c\right) \tag{2.32}$$

  where κ is a hyperparameter that controls the slope of the sigmoid function, and c is a hyperparameter that controls the shift of the kernel.

2.11.2 Logistic Regression

Logistic regression is a special case of linear regression which is used for binary classification problems (used in this thesis in Chapter 6). It is based on the idea of using a logistic function (also referred to as sigmoid function) to model the probability of an example belonging to a particular class. The logistic function is defined as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.33}$$

The logistic regression model is defined by a set of weights w and a bias b, which are learned from the training data. The model makes predictions using the following equation:

$$\hat{y} = f(w^T x + b) \tag{2.34}$$

where x is the input example, w is the weight vector, b is the bias, and $\hat{y}$ is the predicted probability of the example belonging to the positive class.

To train the logistic regression model, a loss function is used to measure the discrepancy between the predicted probabilities and the true labels. A common choice of loss function is the binary cross entropy loss. There are several hyperparameters that can be adjusted in logistic regression, including the regularisation parameter C and the solver used to optimize the model. The regularisation parameter controls the trade-off between the model complexity and the amount of regularisation, and can help to prevent overfitting. The solver specifies the algorithm used to optimize the model, such as gradient descent or Newton's method.

2.12 Clustering Algorithms Used

2.12.1 DBSCAN

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [34] is an unsupervised method for identifying clusters in data. DBSCAN works on the assumption that clusters form dense regions in space, and that these are separated by areas of lower density. DBSCAN is a popular choice for clustering geographic data and unlike other clustering methods (e.g K-means) does not require the number of clusters to be specified in advance. The only required parameters are ε (the minimum distance required for a point to be in a cluster) and N (the minimum number of points required to form a cluster). Points that are not part of any cluster can be classified as noise.

More specifically, The DBSCAN algorithm goes as follows:

1. Initialize an empty list of clusters and a set of unvisited points.
2. Select a point at random from the unvisited points.
3. Find all points within a distance ε of the selected point.
    - If there are at least N points within this distance, the point is considered a core point and a cluster is formed around it
    - If there are fewer than N points within this distance, the point is considered a noise point and is ignored.
4. Add the point and all points within a distance ε of it to the cluster.
5. Mark the point and all points within a distance ε of it as visited.
6. Repeat steps 2-5 until all points have been processed.

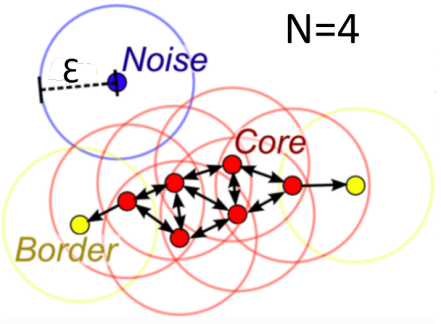

FIGURE 2.11: Diagram of the DBSCAN clustering parameters (adapted from wikipedia). ε is the minimum distance required for a point to be in a cluster, and N is the minimum number of points required to form a cluster.

An example cluster using $N = 4$ is presented in Figure 2.11. DBSCAN uses the Euclidean distance to locate points in space, and so ε can be estimated based on real world distances. It also extends trivially to higher dimensions, in the 3D case forming ε radius spheres instead of 2D circles.

2.13 Performance Metrics

2.13.1 Core Definitions

To quantitatively evaluate detection network results, we must compare network predictions to ground truth labels. There are several important definitions which form the basis of object detection performance metrics.

2.13.1.1 Intersection Over Union

The *Intersection over union* (IoU) is used to calculate the overlap between a ground truth bounding box and a predicted bounding box. To calculate the IoU the area of overlap between the two bounding boxes is divided by the area of union. An IoU threshold can be set to determine if a detection is valid or not (Figure 2.12).

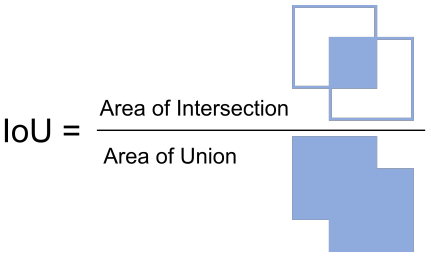

FIGURE 2.12: Intersection over union for two bounding boxes.

2.13.1.2 Confusion Matrix

Correct and incorrect detections can be assessed using a confusion matrix, where there are four possible outcomes (Figure 2.13). A *true positive* (TP) is a correct detection, which intersects with the ground truth bounding box with IoU greater than or equal to the threshold value. A *false positive* (FP) is an incorrect detection, which has IOU value less than the threshold value. A *false negative* (FN) is a ground truth bounding box which is not detected by the network. Finally, a *true negative* (TN) refers to a ground truth object which is correctly not detected. Since in object detection the number of TN's is not well quantified, it is generally excluded from detection performance metrics.

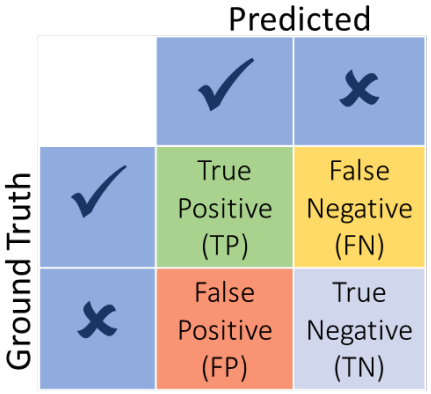

FIGURE 2.13: Confusion matrix for correct and incorrect detections.

2.13.2 Precision and Recall

Detection results are generally reported in terms of recall and precision. *Precision* is defined as the number of true positives (TP) over the number of true positives plus the number of false positives (FP). In other words, the fraction of detections which are correct:

$$precision = \frac{TP}{TP + FP} \tag{2.35}$$

Recall is defined as the number of true positives over the number of true positives plus the number of false negatives. In other words, the fraction of labelled objects which were detected:

$$recall = \frac{TP}{TP + FN} \tag{2.36}$$

2.13.3 Precision-recall Curve

CNNs generally output a confidence for each prediction (i.e a probability in the range 0 to 1), which means the trade off between precision and recall can vary depending on where the prediction threshold is set. To assess the results of a CNN we therefore calculate precision and recall for a range of threshold values and plot the results as a *precision-recall curve* [37].

2.13.4 Average Precision (AP)

The *average precision* (AP) is the area under the precision-recall curve, and is used to evaluate and compare network results. The AP is the precision averaged across all recall values, and varies between 0 and 1 (with 1 being perfect detection).

2.13.4.1 IoU@.50 and @.75

Since the initial IoU threshold for correct detections alters the final precision-recall results, AP can also be calculated for different IoU thresholds. For example AP@0.5 and AP@0.75 are the average precision scores for IoU thresholds of 50% and 75% respectively. Lower IoU thresholds measure the overall detection accuracy, while higher IoU thresholds measure the network's localisation accuracy. In the COCO object detection challenge AP@[0.5:0.95] is calculated, this score is determined by finding the AP score for ten IoU thresholds (ranging from 0.5 to 0.95 with a step of 5) and taking the average.

2.13.4.2 Mean Average Precision (mAP)

In cases where we are predicting multiple classes, or performing a k-fold cross validation, we often report the mean average precision (mAP). This is simply the mean of the AP scores for each individual class.

2.13.5 F1-score

We can also calculate overall accuracy (or rather a figure of merit) at each threshold point by using the *F1-score*, which is the harmonic mean of recall and precision [37]:

$$\text{F1-score} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.37}$$

Depending on the application, we may have different priorities when it comes to valuing recall or precision. For example, in wildlife detection recall may have higher value than precision, as false positives could be quickly filtered out manually whereas finding missed animals would require scanning the entire dataset. The F-beta score (F_β) uses a positive real factor β, such that recall is considered β times more important than precision. This is given by:

$$\mathrm{F}_\beta\text{-score} = (1+\beta^2)\cdot\frac{precision\cdot recall}{(\beta^2\cdot precision)+recall} \tag{2.38}$$

2.13.6 Faster R-CNN metrics

Faster R-CNN training involves joint training of four different loss functions. The final values of these can be reported to give an indication of the success of the different aspects of the detection. The smaller the loss value, the more adept the network is at performing that aspect of detection:

RPN classification loss: Measures the accuracy of the anchor detections generated by the region proposal network stage.

RPN regression loss: Measures how accurately the anchors match the object proposal bounding boxes.

Fast R-CNN classification loss: Measures the classification accuracy of the final bounding box predictions, over all object classes.

Fast R-CNN regression Loss: Measures how accurately the proposed bounding box detections match the ground truth bounding boxes.

2.14 Conclusion

In this Chapter we have reviewed the current literature on wildlife detection in VHR satellite and UAV imagery. We have summarised existing work on manual and automated analysis, and discussed current limitations. Following this we introduced the background of neural networks, outlined the foundations of CNN architectures, and introduced key terms we will use throughout the thesis. We gave an overview of the CNN architectures which are used in the research (U-Net and Faster R-CNN) as well as the ML and clustering methods employed. Finally, we define the performance metrics which will be used to assess model performance. This provides a framework for our own wildlife detection applications discussed in Chapters 3-6. In the following Chapter

we will present the first of these research outputs (RO1); detecting albatrosses in VHR satellite imagery using a U-Net CNN architecture.

Chapter 3

Automated Detection of Albatrosses in VHR Satellite Imagery

3.1 Overview

In this Chapter we apply a CNN architecture to automatically count wandering albatrosses in VHR satellite imagery. We use a dataset of manually labelled imagery provided by the British Antarctic Survey to train and develop our methods. This consists of four 31-cm resolution WorldView-3 (WV-3) satellite images of different island colonies, containing approximately 2000 albatrosses in total. We employ a U-Net architecture, designed for image segmentation, to simultaneously classify and localise potential albatrosses. We aid training with the use of the focal loss criterion, to deal with extreme class imbalance in the dataset. We perform a four-fold cross validation across the islands, and find that the results vary between images. In our misclassification analysis we find that while some false detections are clearly incorrect, the majority appear visually identical to annotated albatrosses. We conclude that an analysis of ground truth uncertainty is needed to place the network results within the context of human performance.

3.2 Introduction

Albatrosses are the world's most threatened seabird family [117], with all six species of the genus *Diomedea* (great albatrosses) classed under some level of threat under the IUCN Red List (IUCN 2021, iucnredlist.org). This is largely attributed to the

impact of incidental mortality from long-line fisheries, disease, and the introduction of pests to their nesting habitats [117]. However, since they nest on remote and often uninhabited island chains, regularly monitoring species is challenging. Accessing islands to conduct ground surveys can be dangerous, expensive and logistically difficult [43]. This means that there is no annual census data in many locations [149], limiting our ability to understand fine-scale population dynamics, monitor population health, and inform conservation actions [69, 115].

Recently, VHR satellites have offered an alternative: to survey albatrosses directly from space. Wandering albatrosses (*Diomedea Exulans*) were first shown to be identifiable in 31-cm resolution WV-3 imagery in 2017 by Fretwell et al. [42]. With an adult wandering albatross having a body length between 107-135cm [2], the albatrosses appear as approximately 4 to 5 pixels of white against their green nesting habitats. Fretwell et al. [42] validated their methods by comparing satellite counts to ground based observations (we refer the reader to this publication for further details on detectability and interpretation of satellite counts in relation to traditional sampling methods). This was the first example of using the WV-3 sensor to count birds directly, adding to previous works which surveyed birds using indirect VHR satellite observations (e.g extrapolating numbers of emperor [40], Adélie [83] and chinstrap [108] penguins based on colony area, and using Google Earth satellite images to detect the nests of masked boobies [72]). The method was later used to survey certain colonies of wandering albatrosses on the French overseas territories of Kerguelen and the Crozet Islands [149].

While the satellite survey method shows promise, manually analysing the resultant imagery is time consuming and subjective [42]. This limits our ability to scale surveys across larger areas and to more frequent time intervals. These factors strongly motivate the development of automated image processing algorithms, to improve speed, reduce cost, and standardise the counting procedure [150]. As discussed in Chapter 2, CNNs have shown state-of-the-art performance at the task of object detection, but as yet have not been applied to the task of detecting albatrosses in VHR satellite imagery.

Despite their proven success, there are challenges when using supervised CNNs to detect albatrosses. Similar to more classical machine learning approaches, the small size of the target wildlife can present challenges for CNNs. While detection networks (such as YOLO and Faster-RCNN) generally locate targets using a bounding box, they have shown to have limited performance for very small objects [116]. However, segmentation approaches, which classify each pixel in the image rather than assigning a bounding box, could provide a suitable architecture for the task. While class imbalance can be a significant barrier in this approach, as the number of background pixels will vastly

outweigh pixels in the target class, new loss functions such as the *focal loss* [92] have recently overcome these issues.

The second challenge with satellite detection of wildlife is that annotated datasets are small and costly to obtain. At present purchasing VHR satellite imagery over large extents can be expensive, and the added time and effort required for manual analysis has restricted many studies to one or two images [69, 148]. However, deep learning algorithms perform best when supplied with a large and diverse set of training images, and without sufficient variation we cannot be confident in the algorithm's ability to generalize to new unseen data. While some researchers have found inventive ways of generating training data from other sources, such as down-sampled aerial imagery [13], many studies consider only a single large satellite image which is divided up into smaller patches to train and test a classifier. Results from these methods are almost inevitably biased due to non-independence of the train and test patches, and we may see a significant drop in performance if an image with different characteristics is presented.

In this Chapter, we will test the application of U-Net in combination with the focal loss to the task of detecting wandering albatrosses in WV-3 imagery. We will use a dataset of four images of different colonies, taken at different locations, times of year, and times of day. We will perform a cross fold validation across the islands, to compare how robustly the model generalises to new imagery and locations. Our main questions are i) whether the U-Net and focal loss can successfully detect albatrosses in WV-3 imagery and ii) whether the model transfers well across the different islands in the dataset.

3.3 Methodology

3.3.1 Data Collection

All satellite imagery was collected by the highest resolution commercially available sensor – Maxar's WorldView-3 satellite. This samples at a spatial resolution of 31cm per pixel in the panchromatic band, and 124cm per pixel in the multispectral bands (in this study we use four multispectral bands; red, green, blue and near-infrared). We collated images of four separate colonies of wandering albatrosses, originally collected as part of previous studies (see [42] and [149]). The colonies are located on Bird Island (BI) and Annenkov Island (AN) in South Georgia, Apotres (AP) in the Crozet islands, and Grande Coulee (GC) on the west coast of Kerguelen Island (Figure 3.1). Images were collected over different months and times of day, and present variation in terms of cloud cover, lighting and vegetation. They also differ in size, with the smallest (BI) covering 16km^2 and the

largest (AN) covering 105km^2. A summary of the satellite image acquisition details is presented in Table 3.1.

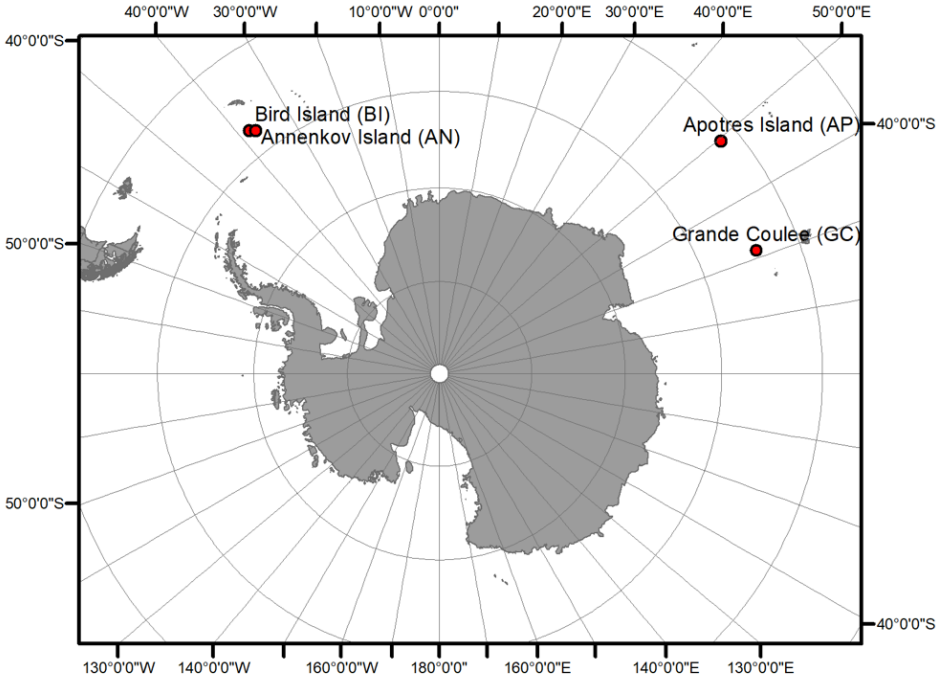

FIGURE 3.1: Locations of the islands imaged in the dataset.

TABLE 3.1: Location and acquisition information for the four images used in the study. Acquired from Maxar's WorldView-3 satellite

Image	Latitude	Longitude	Date/ Time	Area (km^2)
Bird Island (BI)	-54.005408	-38.048144	10 Jan. 2016 / 12:06	16.1
Annenkov Island (AN)	-54.490852	-37.068006	03 Feb. 2017 / 12:44	104.9
Apotres Island (AP)	-45.9666628	50.4499982	03 Mar. 2017 / 07:05	69.2
Grande Coulee (GC)	-49.672945	68.755487	16 Mar. 2017 / 05:39	102.2

3.3.2 Data Pre-processing

3.3.2.1 Expert Annotation

For visual analysis the panchromatic and multispectral bands were pan-sharpened using the ArcMap 10.1 implementation of the Gram-Schmidt algorithm (ArcMap 10.1, Environmental Systems Resource Institute, Redlands, CA, USA), resulting in a high resolution RGB image. Since wandering albatrosses are largely white, with a body length of 107-135cm [2], individuals appear as several pixels of white against their green nesting habitat (Figure 3.2). All four images were annotated by the same expert observer by

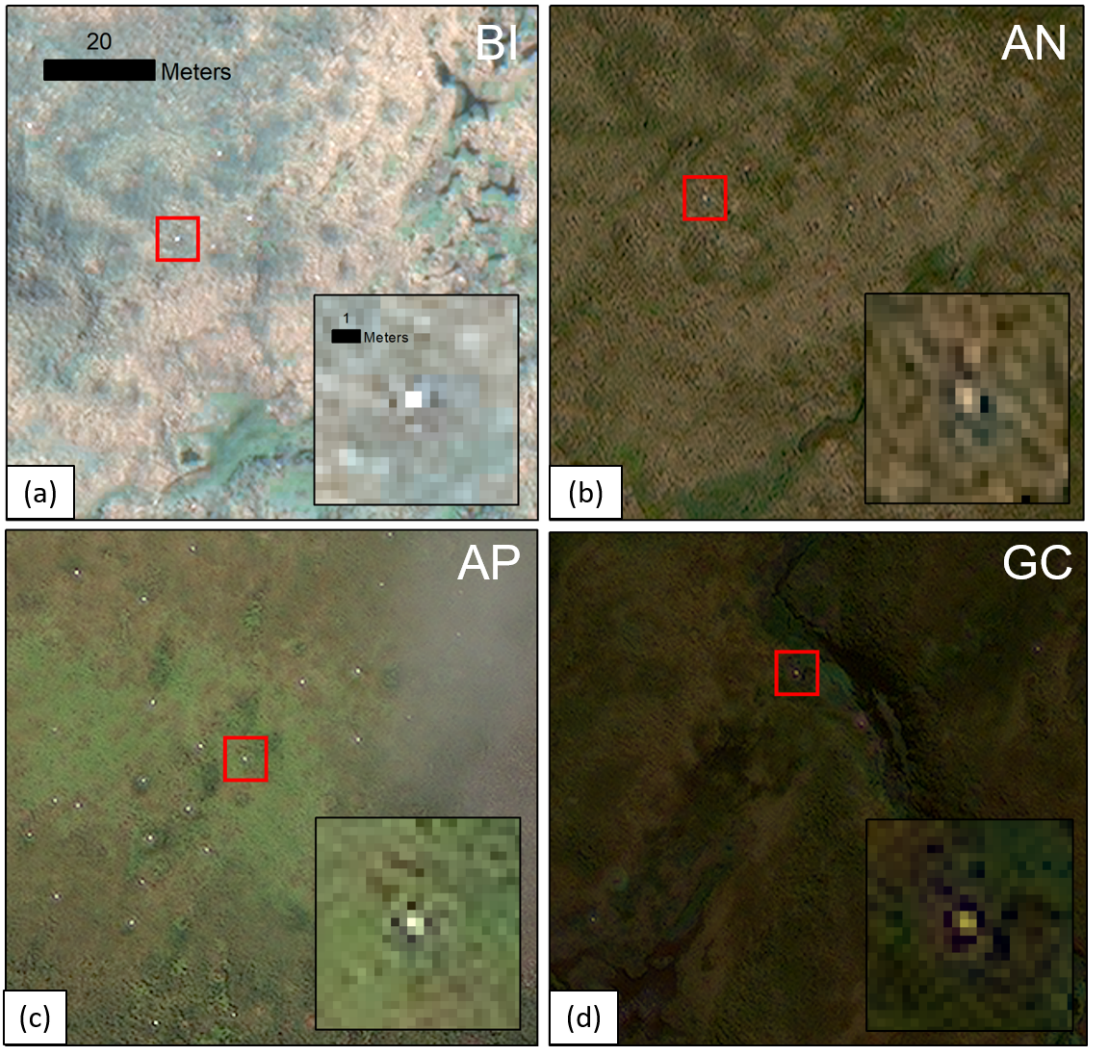

FIGURE 3.2: Examples of albatrosses in the four islands, as viewed in ArcMap 10.5. (**a**) Bird Island. (**b**) Annenkov Island. (**c**) Apotres Island. (**d**) Grande Coulee. Imagery from DigitalGlobe Products. WorldView3 © 2020 DigitalGlobe, Inc., a Maxar company.

drawing a 200×200m grid and scanning through each cell one at a time. When potential albatrosses were identified they were labelled with a single point marker, placed as central to the object as possible [42]. In total the observer identified 1966 albatrosses: 985 on BI, 161 on AN, 171 on AP, and 649 on GC. The locations of the placed markers are shown as red points in Figure 3.3.

3.3.2.2 Tiling and Mask Generation

To prepare our dataset for training we tiled all four satellite images into 500×500 pixel square patches. The four multispectral bands (red, green, blue and near-infrared) were upsampled using bilinear interpolation to match the dimensions of the panchromatic image. ArcMap shapefiles generated by our expert observer were converted into binary

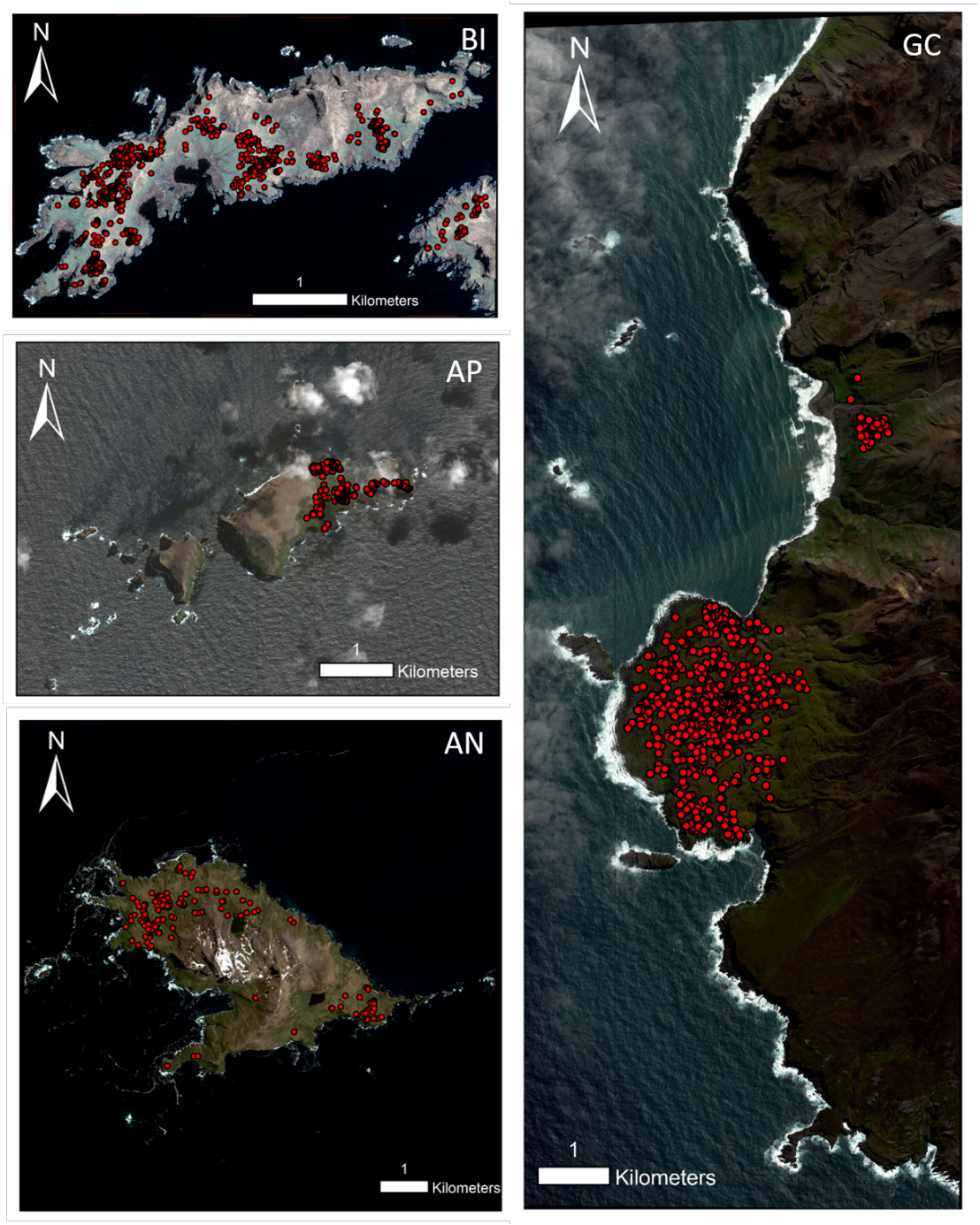

FIGURE 3.3: Examples of the four WV-3 satellite images with albatross annotations (red dots). Shown for BI: Bird island, AP: Apotres, AN: Annenkov and GC: Grande Coulee.

segmentation maps (with background = 0, albatross = 1), which were also tiled into 500× 500 patches exactly overlaying the input images. Since the exact placement of observer labels can differ slightly, we use template matching [37] to shift observer annotations to the centre of each albatross. We segmented albatrosses using a 3×3 pixel square, which was based on visual inspection of the imagery and matched the size of the majority of objects identified in the ground truth.

3.3.2.3 Train, Validation and Test Splits

To keep the dataset proportional we chose an equal number of patches from each island (500 patches from the land and 250 patches from the sea). These numbers approximately represent the maximum number of patches present in the smallest image (BI). For our leave-one-island-out cross validation we trained the network on patches from three islands: resulting in 2250 patches in total, with 20% of these reserved for validation. Input patches of size 412×412, as well as their corresponding target patches of size 340×340, were cropped randomly from the larger tiles to augment the dataset. Tiles were also randomly flipped and reflected to add further variation. At test stage the fourth unseen image was tiled, and the trained network was run over all patches to generate a final prediction.

3.3.3 Network Architecture

For our CNN we use a U-Net architecture (described in detail in Chapter 2 Section 2.10.1), which was originally designed for biomedical image segmentation [124], but has recently shown state-of-the-art performance in a range of tasks [35]. U-Net works by classifying every pixel in the image into a class (in our case albatross and non-albatross). The output probability map can be directly overlaid with the input image, allowing us to classify and localise albatrosses in a single stage.

We present the exact architecture in Figure 3.4. The contracting path (left) follows the typical architecture of a CNN, applying repeated 3×3 convolutions, ReLU activation, and 2×2 max pooling to extract features from input images. The expanding path (right) upsamples feature maps and concatenates them with higher resolution information cropped and copied from the corresponding layer in the contracting path. This allows for precise localization of classified pixels.

Given our small dataset we use transfer learning, a method where convolution filters learned from a larger dataset are copied across to the new network. In principle these represent a generic set of filters, which can be used to extract low-level features common

to all images (e.g edges, patterns and gradients). We initialise our network using filters from a vgg-16 network [133], pre-trained on the ImageNet database [29]. To minimise information loss at the edge of images, we choose not to use padding in convolution operations (aside from those transferred from vgg-16), thus the output predictions are of reduced size (340×340 compared to 412×412 inputs). Experiments also showed no performance gain when using learnt upsampling (through upconvolution), so we favour bilinear upsampling for simplicity.

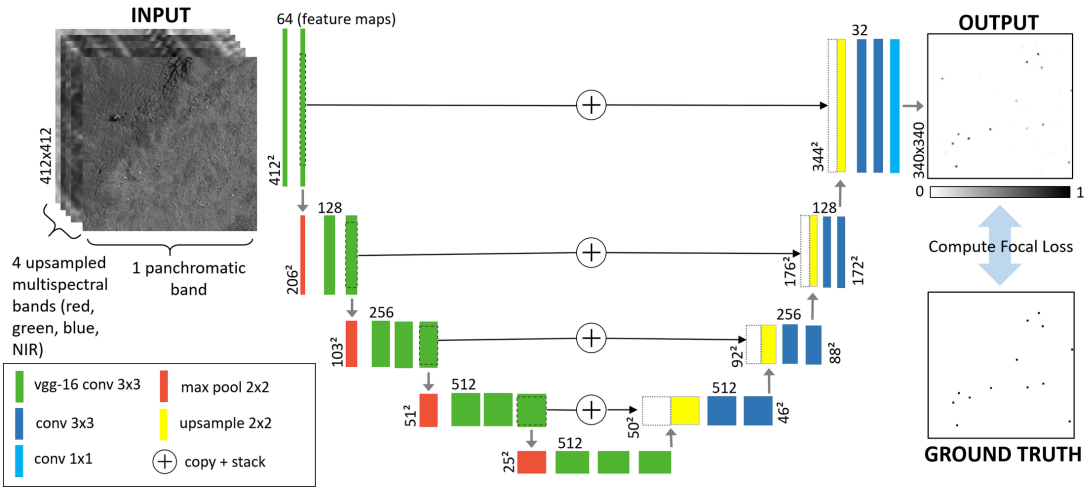

FIGURE 3.4: Diagram of the U-Net architecture and training procedure.

3.3.4 Hyperparameters

Since in our dataset the number of albatross pixels is vastly outweighed by background instances, there is a danger the network would favour ignoring all albatrosses to achieve a high accuracy on the more prevalent class. To account for this we calculate the error between output and ground truth using the Focal Loss, proposed by [92] as a method for addressing extreme class imbalance. It works by adding a modulating factor to the standard cross entropy criterion, which places more focus on hard, misclassified examples. If $y \in \{\pm 1\}$ denotes the ground truth class and $p \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$, then the focal loss can be expressed as:

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma} log(p_t) \quad \text{, where} \quad p_t = \begin{cases} p, & \text{if } y = 1. \\ 1 - p, & \text{otherwise.} \end{cases} \tag{3.1}$$

Increasing the *focusing parameter* $\gamma \geq 0$ reduces the loss contribution from easy to classify background examples. We ran experiments to assess the best choice for γ, and

found that $\gamma = 0.5$ and $\alpha = 0.5$ gave the best results. We trained the model using the Adam optimiser [76], a learning rate of 0.0001 (degrading to 0.00001 after 5 epochs), and a mini-batch size of 4. For each fold, we trained three U-Net models using identical settings, to assess random variation in the output results.

3.3.5 Hardware and Frameworks

Model training is performed on a PC workstation equipped with Intel i7-8700 CPU @ 3.20GHz, 32GB of RAM and NVIDIA Titan Xp graphics card with 12GB of GPU memory. PyTorch 1.12.0, Torchvision 0.13.0 and CUDA 11.6 were used in the training and inference pipeline.

3.3.6 Evaluation Metrics

To evaluate the output of our U-Net method we use the Average Precision (AP) score for per-island results, as well as the mean Average Precision (mAP) to average results across the four test folds. We also report recall, precision and the F1-score for specific confidence thresholds. All metrics are described in detail in Chapter 2 Section 2.13.

3.4 Results

3.4.1 Cross validation results

The results of our four-fold cross validation are presented as precision-recall curves in Figure 3.5, with lines showing the mean results and shaded areas representing the standard deviation of the three U-Net runs. When averaged across the four islands we achieve a mAP score of 0.67, however we can see that the results vary between the four test folds. The AP image scores the highest average precision of 0.78, while AN scores the lowest at 0.51. In terms of F1 accuracy scores, both BI and AP reach 80% accuracy at peak, while GC and AN only reach approximately 70% and 60% respectively. We can see that the precision score for BI and AP remains very high even for recall values up to 80%, suggesting a low number of false positives for the islands. However, recall values fall at higher thresholds, with none of the four islands passing the 90% recall mark, indicating approximately 10% of annotated albatrosses were missed by the U-Net detection method.

For final results, confidence threshold values must be selected to determine the precision-recall trade off. In Figure 3.6 we plot the F1-score accuracy of each model against

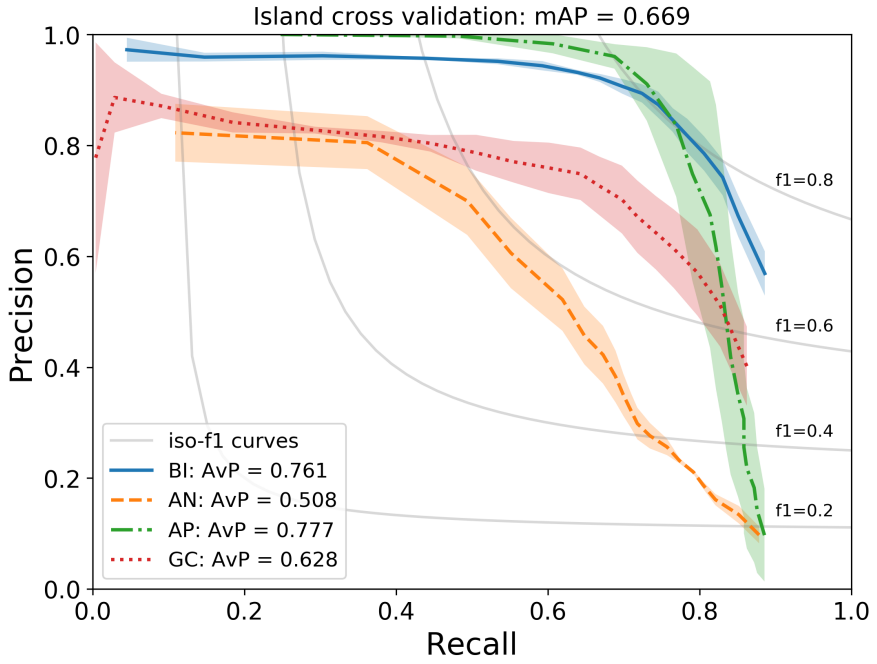

FIGURE 3.5: Average precision-recall curves for each of the four islands. Lines show the mean and shaded area shows standard deviation from three U-Nets.

varying confidence thresholds to investigate this relationship. We find that the optimal thresholds vary between the four images, with BI and GC achieving peak F1 with a threshold of 0.35, while AP and AN benefit from higher thresholds of 0.75 and 0.8 respectively. However, we note that AP and AN both contain much smaller numbers of albatrosses (171 and 161) compared to BI and GC (985 and 649), which makes the results more sensitive to errors. The F1 scores for the more populated islands of BI and GC both remain quite level until confidence threshold 0.5, and selecting the average best threshold of 0.56 would provide reasonable results for all islands. In practical applications this optimal threshold could also be selected manually by an expert observer, based on visual assessment of a small portion of the imagery. Finally, the addition of more data into the training set would inevitably improve and standardise the results, making this selection less variable across different images.

3.4.2 Whole Image Results

In this section we perform a visual assessment of errors (false positives and false negatives) from the U-Net method, to investigate areas for improvement. In Figure 3.7 we plot the U-Net predictions over the whole island images, to check for any regions where obvious misclassifications occur. For these examples we use the optimal F1-score thresholds for each island to generate our true positive, false positive and false negative results. We can see that the U-Net method is successful in filtering out large areas of

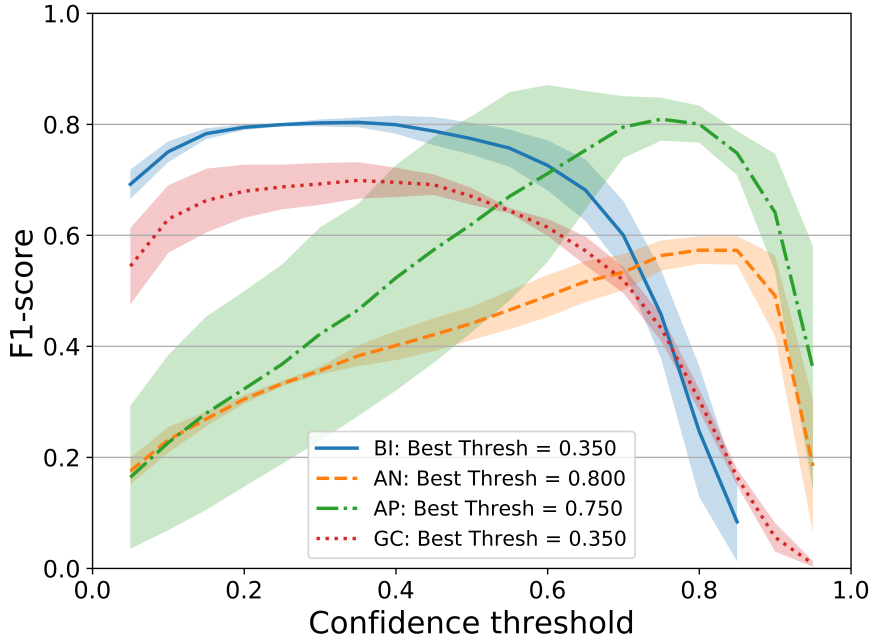

FIGURE 3.6: Average F1 scores for each of the four islands, across a range of confidence threshold values.

background, with very few false positives far from labelled colony areas. Exceptions to this include a small number of detections in the ocean in AP, a cluster of false positives between colonies in GC, and points very close to the coastline in all images.

3.5 Discussion

3.5.1 Misclassification Analysis

We inspect these misclassifications in more detail in Figure 3.8. For GC, we see the cluster of false positives between colonies fall along a steep ridgeline, with the U-Net method picking out white rocks as albatrosses (Figure 3.8a). While the rocks appear similar in size and colour to labelled albatrosses, this terrain is not suitable nesting habitat [42]. For AP, spectral distortion over the ocean also leads to false positive results (Fig 3.8b). It is likely that the combination of bright white waves crests and strong spectral extremes lead to these false detections, as this type of noise is not present in the three training images. AP is the only image where we have false detections over the ocean, with the U-Net method effectively avoiding ocean detections in the three other images. AP is also the only image where there is some hazy cloud cover over the colony, which as can be seen in Figure 3.8c leads to a number of false negative errors. As the network has not been presented with examples of cloud cover in the three training images, it is again not surprising the network is unable to generalise in this case.

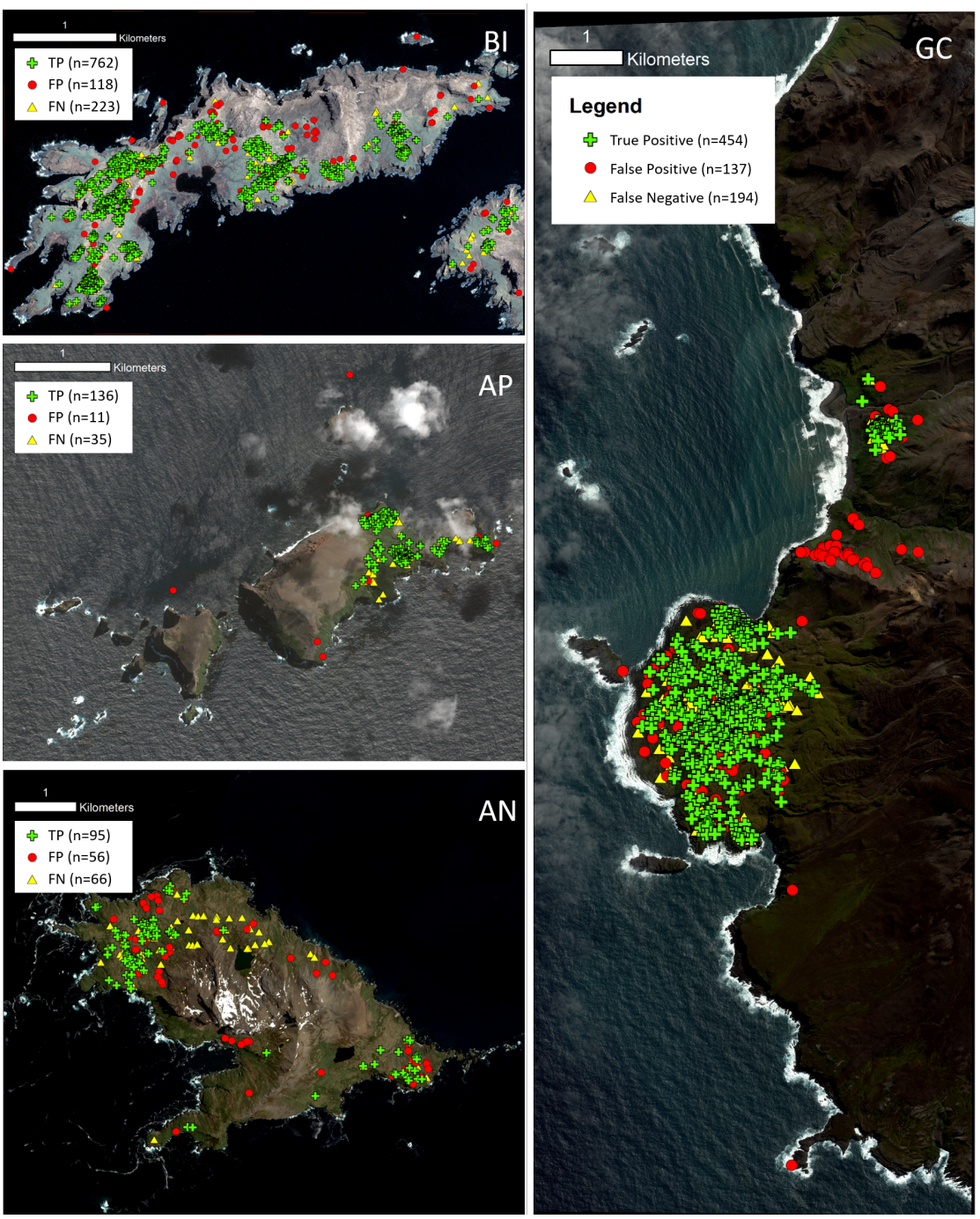

FIGURE 3.7: Examples results for the four islands, showing locations of true positive (TP: green cross), false positive (FP: red dot) and false negative (FN: yellow triangle) U-Net predictions. Shown for BI: Bird island, AP: Apotres, AN: Annenkov and GC: Grande Coulee.

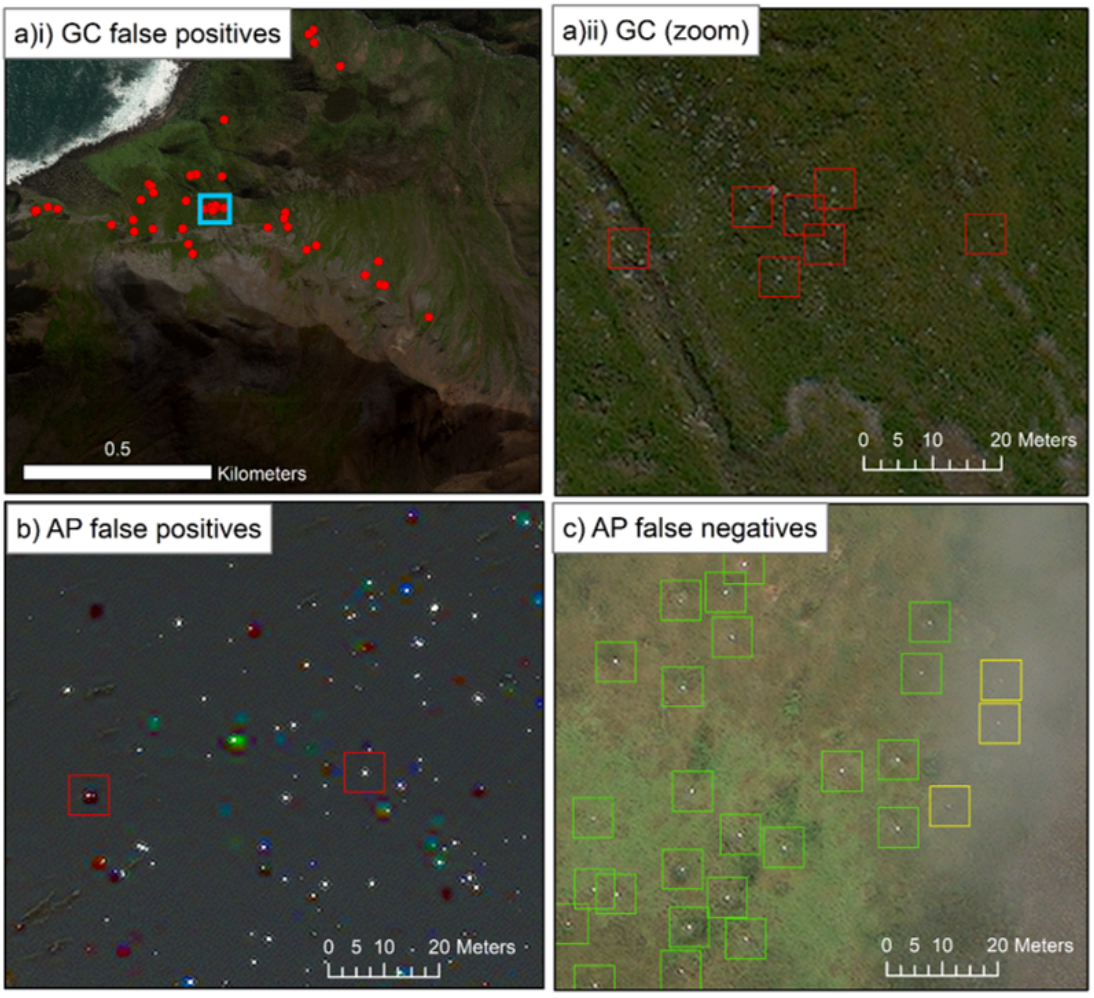

FIGURE 3.8: Examples of clear U-Net errors. a)i) False positive detections along a ridge line in the GC image, which is not a suitable albatross habitat; a)ii) shows that these are rocks appear similar to albatrosses. b) False positives in the ocean in the AP image, caused by spectral distortion and wave crests. c) False negatives under hazy cloud cover in the AP image.

3.5.2 Future Work

In order to improve detection results under cloud we could pre-process images using image dehazing [98]. This could improve clarity and aid both the manual and automated analysis of the imagery. An alternative would be to simulate cloud cover, and add it to a proportion of images at the training stage (for example by adding perlin noise [89]). If the simulated cover is sufficiently accurate, this can be a means of artificially generating the examples needed to train the network. This can be an effective technique when genuine examples are difficult to obtain. This being said we recommend obtaining cloud free imagery wherever possible, as the certainty of our detections in thicker areas of cloud are limited both in manual and automated approaches. Similar augmentation approaches could be employed to deal with the false positives over the ocean in AP.

Making the network more robust to this form of noise is important, as it could present itself differently in future imagery. Another alternative is to mask out areas of ocean before generating network predictions, which could be performed fairly quickly manually, or automatically using the normalised difference water index [91]. Masking out areas of ocean would have the dual benefit of decreasing the potential number of false positives, and reducing the processing time. This is also true for the false positive detections in GC, which fall outside the main colony area. Again, input images could be manually pre-processed by experts to restrict the search to known colony locations. Even without these additions obvious false positives could be easily filtered out manually, with comparatively little effort. It is also important to note that results would almost certainly be improved by the addition of extra imagery, and the application of updated state-of-the-art CNN architectures which are rapidly evolving.

3.5.3 Ground Truth Uncertainty

While the errors discussed in Section 3.4.2 can be clearly identified as incorrect due to their locations, we find that in many cases the distinction is not so obvious. In Figure 3.9 we present regions of the images where there are a mixture of true positive, false positive and false negative results. We can see that in fact the majority of these appear visually identical, making it impossible to definitively rule out errors. Are false positives actually albatrosses which were missed in the manual analysis? And can we confidently say that false negatives are not incorrectly labelled albatrosses? These questions ultimately lead back to the subjectivity in the manual analysis, and the level of uncertainty when annotating such small and indistinct objects.

This ground truth uncertainty poses important questions for our analysis. To understand the success of our U-Net method, it is important that we understand how effectively and consistently human observers perform at the albatross detection task. This will allow us to benchmark the results of our automated method within the context of human performance. For this specific dataset there are additional questions relating to the four images, with our visual analysis of the results indicating that there is more uncertainty in some islands than others. For example for AP we have relatively few errors and albatrosses contrast quite clearly against the bright green nesting habitat (Figure 3.9c), while for lower scoring islands (such as AN) vegetation is darker and albatrosses are less distinct (Figure 3.9b). This suggests that there could be more variability between observer results for some islands than others. This ground truth uncertainty also feeds into the U-Net method, since these subjective annotations are used to train the network. Gaining a better understanding of ground truth confidence will allow us to assess which labels to use in the supervised training scheme. While in this Chapter we have shown

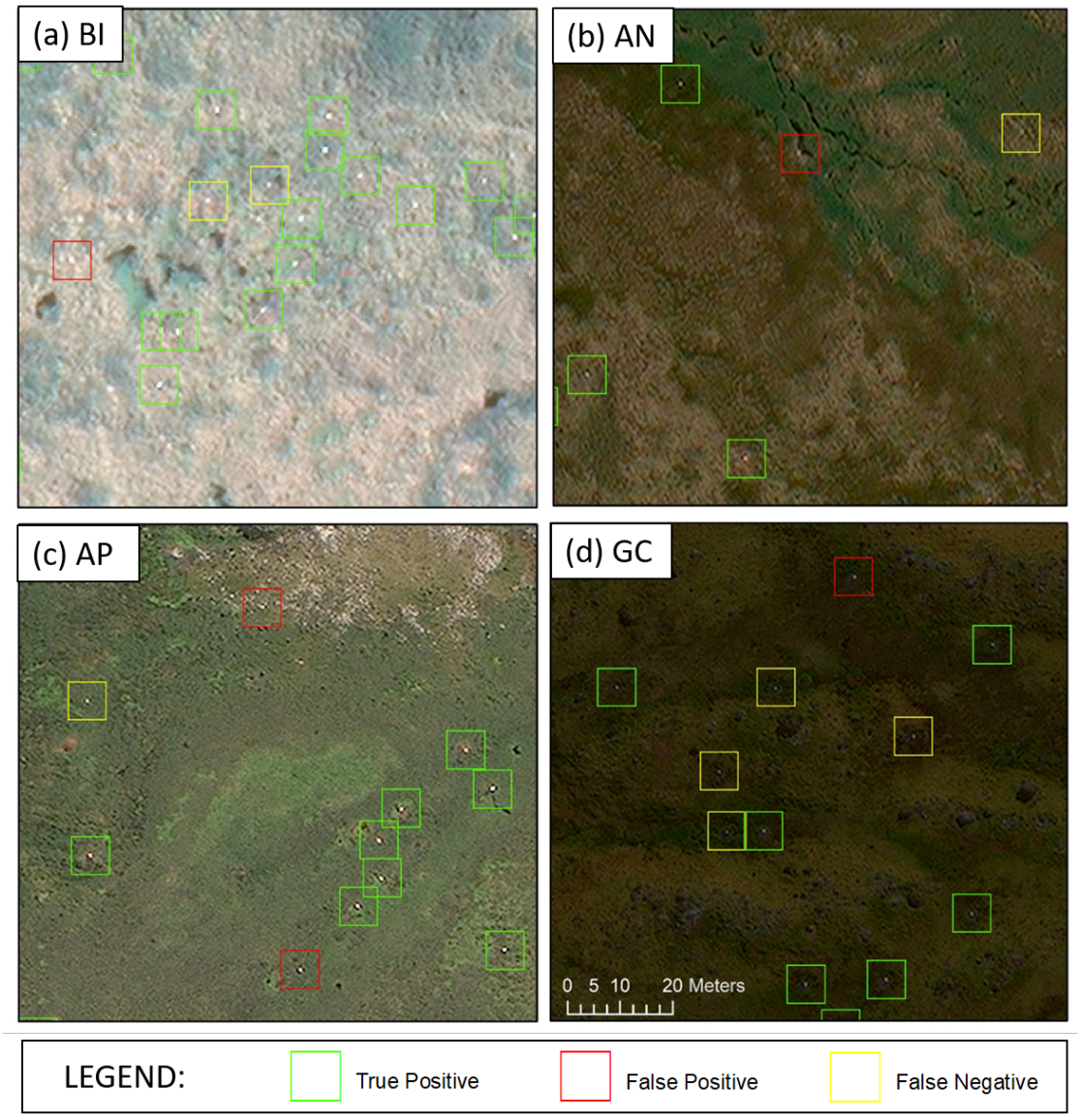

FIGURE 3.9: Example results where the distinction between true positives (green), false positives (red) and false negatives (yellow) is unclear. This raises questions of ground truth uncertainty, and subjectivity in the manual analysis. Presented for (a) Bird Island, (b) Annenkov, (c) Apotres and (d) Grande Coulee.

that i) the U-Net and focal loss can successfully detect albatrosses and ii) the methods transfer across to the different images with the exception of some errors due to noise and different habitat types, we conclude that an assessment of ground truth uncertainty is essential to benchmark the true success of the methods.

3.6 Conclusion

In this Chapter we trained a U-Net architecture to automatically segment and detect albatrosses in 31-cm resolution WV-3 imagery. We performed a four-fold cross validation across islands, and found varying average precision scores. In our analysis of misclassifications we found a small number of locations where clear errors occurred, including false positives over the ocean in AP and in unsuitable habitat in GC, as well as false negatives under fine cloud haze in AP. We suggest future improvements including the addition of habitat variables (e.g digital elevation models) and image pre-processing (e.g dehazing) could enhance the results. However, in the majority of cases misclassifications were visually difficult to discern from true albatross labels. We conclude that using one observer's subjective labels is insufficient, and that more investigation of inter-observer variation and ground truth uncertainty is needed to place the results in context. In Chapter 4 we will conduct this inter-observer analysis, by collecting additional annotations from five observers. We will use these labels to assess the observer agreement for each of the four images, and to provide a final assessment of the U-Net method success.

Chapter 4

Inter-observer Variation in Satellite Counts of Albatrosses

4.1 Overview

In the Chapter 3 we presented a method for detecting albatrosses in VHR satellite imagery. We used a dataset of WorldView-3 satellite images annotated by an expert observer to train a U-Net detection method. The results of the method were promising, however were challenging to accurately assess due to a high level of uncertainty surrounding the ground truth labels. We concluded that the certainty of the annotations would need to be quantified before the results could be placed in context. In this Chapter we perform this uncertainty analysis, by collecting further labels from a number of observers. In addition, we investigate how the choice of ground truth label can impact the U-Net results, both at the training and assessment stage. In Section 4.3 we outline the experimental protocol for collecting our observer counts and describe the network experiments. We present the results of the inter-observer experiments in Section 4.4.1, finding that there was a high degree of uncertainty in annotations, and that the level of uncertainty differed between the four satellite images. In Section 4.4.2 we present the network experiment results, determining that the U-Net detection method comes close to human performance for all four colonies. We conclude by discussing limitation and ideas for future improvements in Section 4.5.

4.2 Introduction

As discussed in Chapter 3, challenges with manual detection of wildlife in VHR satellite imagery leads to ground truth uncertainty, which makes errors difficult to interpret.

Different annotators may generate different sets of labels, and this inter-observer variation has a knock on effect when developing automated detection methods. Since the majority of approaches (including the U-Net method presented in Chapter 3) are based on supervised training schemes, the quality of ground truth labels has a direct impact on the training and assessment of the model. Since the accuracy of a classifier is limited by the accuracy of the provided labels, it is important to establish the level of uncertainty to fully evaluate performance. In the absence of a gold-standard ground truth (i.e the direct linking of satellite data to concurrent ground based surveys), 100% accuracy cannot be the goal, and would in fact indicate over-fitting to a single observer's subjective annotations [60]. Only when we understand how accurately and consistently humans perform at the task, can we benchmark the performance of an automated approach.

For the specific task of detecting albatrosses in WV-3 imagery, Fretwell et al. [42] noted the effect of inter-observer variation in satellite counts of Bird Island. This particular colony is subject to extensive ground-based surveys, and was used as a test case for validating the manual satellite counts against ground observations. However, this analysis was conducted on a single satellite image of a single site, and the level of variation may differ between islands or images. In this chapter we will investigate the level of inter-observer variation for all colonies, and study the effect of label choice on the U-Net CNN described in Chapter 3.

4.3 Methodology

4.3.1 Data Collection

All four satellite images had been annotated previously by the same experienced observer (we term these our *reference observer* annotations). To extend this and assess inter-observer variation, we conducted additional labelling experiments with five novice volunteers (*observers 1 - 5*). Observers were all colour normal, with an age range of 23-29, and a mixture of two males and three females. They had no previous experience analysing satellite imagery or of the ecology of albatrosses. In our experiments, we restricted changes in viewing conditions by using the same monitor and controlling ambient lighting. Volunteers were given the same information prior to annotating, and identical image patches were used to present examples of albatrosses and potentially confounding objects such as rocks.

We follow the annotation procedure outlined in the original study [42], whereby images were labelled by eye using separate polygons approximately matching the size of the monitor (in our case 160 $\times$ 260m for viewing at a 1:600 scale). For visual analysis

the panchromatic and multispectral bands were pansharpened using the Gram-Schmidt algorithm, resulting in a high resolution RGB image. This processing and the manual analysis was conducted using ArcMap 10.5 (Environmental Systems Resource Institute, Redlands, CA, USA). The time taken to analyse the images varied depending on their size, with each observer taking approximately 30 minutes to complete AP, 1 hour each for BI and AN, and approximately 2 hours for GC.

4.3.2 Network Architecture and Training

Our U-Net architecture remains the same as described in Chapter 3 Section 3.3.3. For our analysis we investigate how closely total observer counts match, the extent of overlap between annotations, and whether there is a best measure for combining them. This includes taking the union of all points (i.e include any point labelled by an observer), the intersection (i.e only include points labelled by all six observers), and each level in between (i.e only include points labelled by at least two observers, at least three, etc).

We run experiments to assess the impact of ground truth selection on our supervised training scheme. This can happen at both the assessment stage (i.e when comparing our network predictions against our chosen ground truth) and at the training stage (i.e when choosing which labels to use when training our network). For our experiments at the assessment stage we use the majority vote labels (i.e only points annotated by at least three observers are included in the ground truth) to train the network. The choice for this is detailed later in Table 4.3. We then assess the results of the network against all individual observers' labels (the reference observer and observers 1 - 5), as well as the intersection, union, and majority vote.

When assessing the impact of label choice at the training stage, we invert this and train on the different options for ground truth and assess using the majority vote. We also experiment with training using a 'mixed' ground truth, where we select a random observer's labels for every patch, at every epoch. We hypothesise that this random shuffling will automatically encode observer uncertainty, as points which are only labelled by one observer will appear on average one sixth of the time, whereas those with complete observer agreement will appear 100% of the time. For all analysis we train each individual model three times, and present the results as the average of all three, to mitigate for variation. All other network parameters are kept the same as described in Chapter 3 Section 3.3.4, only the ground truth labels are altered.

4.3.3 Hardware and Frameworks

Model training is performed on a PC workstation equipped with Intel i7-8700 CPU @ 3.20GHz, 32GB of RAM and NVIDIA Titan Xp graphics card with 12GB of GPU memory. PyTorch 1.12.0, Torchvision 0.13.0 and CUDA 11.6 were used in the training and inference pipeline.

4.3.4 Evaluation Metrics

For our analysis of inter-observer variation we compare agreement using the F1-score (defined in Chapter 2 Section 2.13). In our procedure we take one observer's labels as a "predicted" result, and assess the accuracy of their predictions against every other observer's "ground truth" labels. Using this approach enables us to calculate inter-observer agreement in terms of F1-score accuracy (the same measure used to assess our U-Net performance). We calculate the F1-score accuracy between every pair of observers, and take the mean of all scores to get a measure of average accuracy (Av. F1) for each of the four images. To assess U-Net performance we use Average Precision (AvP) and mean Average Precision (mAP).

4.4 Results

4.4.1 Inter-observer Results

4.4.1.1 Total Counts

We initially assess the total number of albatrosses tagged by each observer in each of the four islands (BI, AN, AP and GC). We find that the variation in total counts was significant in some cases, and also differed between the four images (Table 4.1). For example BI counts range from 612 to 994, with a standard deviation of 17% from the mean. In contrast AP shows significantly higher agreement in total counts, with a deviation of only 3% from the mean. This variation is likely to be due to the differences in both the appearance of albatrosses, and the number of other spectrally similar objects in the background. For example as we saw in Chapter 3 (refer in particular to Figure 3.2) albatrosses are much clearer in AP, with the white points strongly contrasting against the bright green, relatively uniform vegetation. In some cases albatrosses are also encircled by a brown ring, indicating a cleared area of vegetation surrounding the nest. However in other islands the contrast is weaker, with vegetation appearing more yellow and albatrosses not as bright. This said, it is important to note that total counts do not

capture the agreement between labels (two observers could label 100 completely separate points), and to assess observer agreement we must compare how many annotations coincide with the same object.

TABLE 4.1: Total counts with the mean, standard deviation, and percent deviation for each island.

	ref_ob	**ob1**	**ob2**	**ob3**	**ob4**	**ob5**	**mean ± std**	**% dev**
Bird Island	985	994	763	792	846	612	832 ± 145	17
Annenkov	161	155	116	177	174	120	151 ± 26	18
Apotres	171	165	168	177	174	162	170 ± 6	3
Grande Coulee	649	690	656	840	741	638	702 ± 77	11

Figure 4.1 shows the fraction of identified objects which were labelled by at least one observer (obviously 100%), at least two observers, at least three, etc. We see that for AP, over 84% of points in the image are labelled by at least three observers (i.e the majority), suggesting strong agreement for a large fraction of the ground truth. GC shows a similarly high agreement with 73% of annotations having majority agreement. However for AN there is a particularly steep drop, with only 42% of annotations agreed on by the majority, and over half the ground truth made up of low confidence annotations. This level of ground truth uncertainty is likely to have a noticeable impact on our supervised CNN.

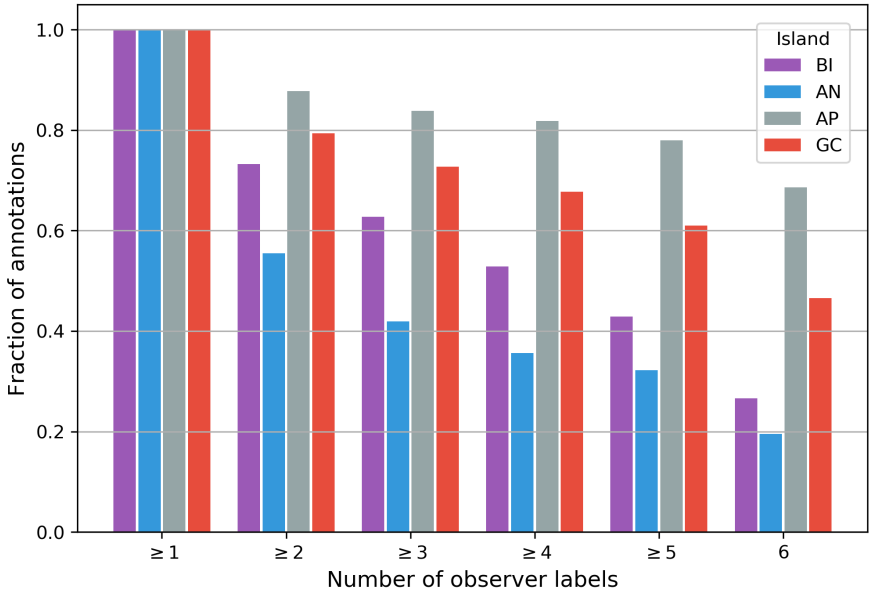

FIGURE 4.1: The distribution of points labelled by multiple observers, compared across the four islands. We see AN has the worst agreement (with only 20% of objects labelled by all six observers), and that AP has the highest (almost 70% of objects labelled by all six observers).

4.4.1.2 Inter-observer Agreement

In Table 4.2 we show the inter-observer agreement for each of the four islands. Once again the overall agreement between observers is highest for AP, in the best case achieving 0.95 between observer 1 and observer 2, and an overall average accuracy of 0.92 (Table 4.2c). In contrast for AN (Table 4.2b) we only reach an average accuracy of 0.67 between observers, with BI (Table 4.2a) and GC (Table 4.2d) falling in between (0.74 and 0.85 respectively). We also see little consistency in agreement between observer pairs, for example observer 2 and 5 achieve the highest F1-score for AN, but the lowest for BI.

TABLE 4.2: Accuracy (as F1-score) between observer labels for **(a)** Bird Island, **(b)** Annenkov Island, **(c)** Apotres Island, **(d)** Grande Coulee. We highlight the worst (red) and best (green) scores, and calculate the mean F1-score per observer, as well as the average F1 score (Av. F1) for each island. Averages exclude the 100% F1-scores achieved when comparing an observer against themselves.

(a) BI: Av. F1 = 0.74

	ref_ob	ob1	ob2	ob3	ob4	ob5
ref_ob	1.00	0.81	0.72	0.81	0.78	0.68
ob1	0.81	1.00	0.70	0.79	0.75	0.69
ob2	0.72	0.70	1.00	0.74	0.70	0.66
ob3	0.81	0.79	0.74	1.00	0.78	0.73
ob4	0.78	0.75	0.70	0.78	1.00	0.70
ob5	0.68	0.69	0.66	0.73	0.70	1.00
mean	0.76	0.75	0.70	**0.77**	0.74	0.69

(b) AN: Av. F1 = 0.67

	ref_ob	ob1	ob2	ob3	ob4	ob5
ref_ob	1.00	0.63	0.62	0.57	0.57	0.60
ob1	0.63	1.00	0.72	0.73	0.70	0.73
ob2	0.62	0.72	1.00	0.68	0.70	0.78
ob3	0.57	0.73	0.68	1.00	0.66	0.69
ob4	0.57	0.70	0.70	0.66	1.00	0.67
ob5	0.60	0.73	0.78	0.69	0.67	1.00
mean	0.60	**0.70**	**0.70**	0.67	0.66	**0.70**

(c) AP: Av. F1 = 0.92

	ref_ob	ob1	ob2	ob3	ob4	ob5
ref_ob	1.00	0.93	0.91	0.93	0.89	0.92
ob1	0.93	1.00	0.95	0.94	0.93	0.95
ob2	0.91	0.95	1.00	0.91	0.91	0.93
ob3	0.93	0.94	0.91	1.00	0.92	0.93
ob4	0.89	0.93	0.91	0.92	1.00	0.91
ob5	0.92	0.95	0.93	0.93	0.91	1.00
mean	0.92	**0.94**	0.92	0.93	0.91	0.93

(d) GC: Av. F1 = 0.85

	ref_ob	ob1	ob2	ob3	ob4	ob5
ref_ob	1.00	0.83	0.82	0.77	0.79	0.78
ob1	0.83	1.00	0.91	0.87	0.89	0.88
ob2	0.82	0.91	1.00	0.85	0.87	0.86
ob3	0.77	0.87	0.85	1.00	0.88	0.82
ob4	0.79	0.89	0.87	0.88	1.00	0.86
ob5	0.78	0.88	0.86	0.82	0.86	1.00
mean	0.80	**0.87**	0.86	0.84	0.86	0.84

We perform a similar analysis to see which combination of observer labels (i.e the union, agreement votes and intersection) offer the best accuracy (Table 4.3). We find that using a ground truth consisting of points labelled by at least three observers (i.e the majority vote) achieves the best mean F1-score when averaged across observers, with the intersection scoring the worst for all four islands. We therefore choose to use the majority vote as the baseline when training and assessing the results of our CNN.

TABLE 4.3: Accuracy (as F1-score) when assessing each observer's predictions (rows) against all options for combined ground truths (ranging from taking the union of all observer annotations, through to the intersection. Results for **(a)** Bird Island, **(b)** Annenkov Island, **(c)** Apotres Island and **(d)** Grande Coulee.

(a) **BI**

	union	**≥ 2**	**≥ 3**	**≥ 4**	**≥ 5**	**inter**
ref_ob	0.83	0.91	0.88	0.83	0.75	0.55
ob1	0.83	0.87	0.87	0.82	0.75	0.54
ob2	0.71	0.77	0.79	0.80	0.78	0.65
ob3	0.72	0.85	0.89	0.89	0.83	0.64
ob4	0.75	0.84	0.85	0.84	0.79	0.61
ob5	0.61	0.73	0.78	0.80	0.82	0.75
mean	0.74	0.83	0.84	0.83	0.78	0.62

(b) **AN**

	union	**≥ 2**	**≥ 3**	**≥ 4**	**≥ 5**	**inter**
ref_ob	0.67	0.66	0.70	0.69	0.70	0.56
ob1	0.66	0.85	0.84	0.81	0.78	0.57
ob2	0.54	0.78	0.85	0.88	0.84	0.70
ob3	0.72	0.82	0.79	0.74	0.70	0.52
ob4	0.71	0.81	0.77	0.74	0.71	0.53
ob5	0.55	0.76	0.84	0.88	0.85	0.68
mean	0.64	0.78	0.80	0.79	0.76	0.59

(c) **AP**

	union	**≥ 2**	**≥ 3**	**≥ 4**	**≥ 5**	**inter**
ref_ob	0.91	0.94	0.94	0.93	0.94	0.90
ob1	0.89	0.96	0.98	0.98	0.97	0.92
ob2	0.90	0.95	0.94	0.96	0.95	0.91
ob3	0.92	0.96	0.96	0.95	0.94	0.88
ob4	0.92	0.95	0.94	0.93	0.92	0.89
ob5	0.88	0.93	0.95	0.96	0.97	0.93
mean	0.91	0.95	0.95	0.95	0.95	0.90

(d) **GC**

	union	**≥ 2**	**≥ 3**	**≥ 4**	**≥ 5**	**inter**
ref_ob	0.79	0.83	0.85	0.85	0.85	0.83
ob1	0.82	0.92	0.95	0.95	0.92	0.80
ob2	0.80	0.91	0.94	0.95	0.92	0.82
ob3	0.92	0.92	0.90	0.88	0.83	0.71
ob4	0.85	0.93	0.93	0.91	0.88	0.76
ob5	0.78	0.87	0.90	0.90	0.91	0.84
mean	0.83	0.90	0.91	0.91	0.89	0.79

4.4.2 Network Results

We present network results for each of our four islands, where each model was trained using our leave-one-island-out cross validation (i.e trained solely on image tiles from the three other islands). To assess the results of the network in the context of inter-observer variation, we take the inter-observer F1-scores from Table 4.2, and plot them as precision-recall points (Figure 4.2, gray points). We also add iso-F1 curves showing the average inter-observer F1-score for reference (Figure 4.2, gray lines), these represent the target for our network precision-recall curves to match human performance.

4.4.2.1 Altering Assessment Labels

The results of assessing the output of the network against different ground truth labels are presented in Figure 4.2. We stress that all four models were trained with using the same ground truth (the majority vote), and all other model parameters were kept the same. We can see from the spread of precision-recall curves that our assessment of model performance can vary significantly depending on our chosen ground truth. At one extreme using the union gives us an overall lower recall, as many high uncertainty points (e.g labelled by only one observer) are not predicted to be albatrosses by the network. On the other hand assessing against the intersection we get a high recall but lower precision, as more of the network predictions are assessed as false positives. We also see a range of results assessing against each individual observer, showing the pitfalls

of using a single set of labels, as many studies currently do. For example, for AN using the reference observer labels would lead us to assess the model performing below human accuracy, whereas simply choosing another observer (e.g observer 5) we would fall within the range (Figure 4.2b). The spread is more evident in the islands with lower observer agreement (BI and AN), opposed to those with higher observer agreement (AP and GC). On average, across all four islands, the best mAP score comes from assessing against the majority vote (mAP = 0.74) and the worst using the intersection (mAP = 0.59; for summary see Appendix Table A.1). We also note that while models for BI and AN fall within the range of human performance (exceeding the average observer F1-scores of 0.74 and 0.67 respectively), for GC and AP we do not hit this target.

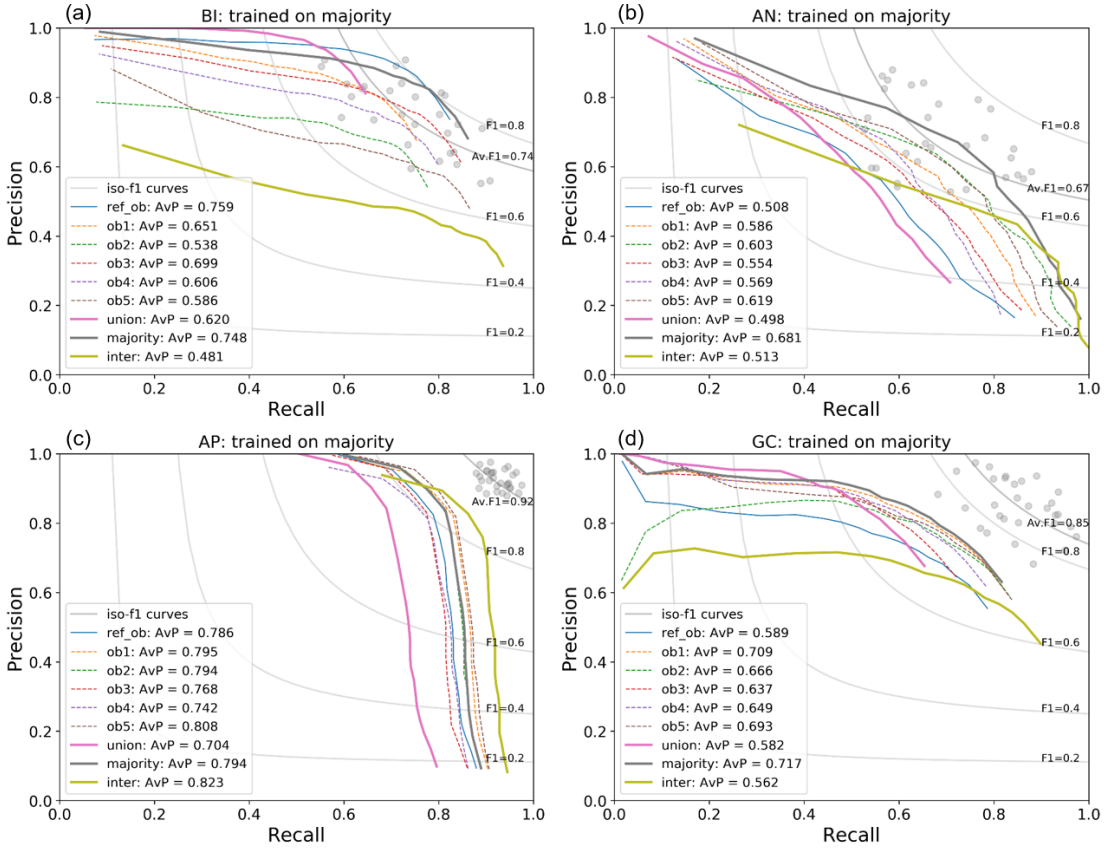

FIGURE 4.2: Precision-recall curves assessed against different sets of ground truth for **(a)** Bird Island, **(b)** Annenkov Island, **(c)** Apotres Island and **(d)** Grande Coulee. We train the models using leave-one-island-out cross validation, and the majority vote labels as training ground truth. Gray points show the individual inter-observer precision-recall points, and grey lines show the corresponding F1-scores. Coloured lines show the precision-recall curves when assessing model output against different ground truth labels. The average precision (AvP) is the area under the precision-recall curve.

4.4.2.2 Altering Training Labels

We see that the network is relatively robust to using different ground truth labels at the training stage (Figure 4.3). For most of the islands there is little deviation in results when trained on either an individual observer, union, majority or intersection ground truth. The most notable exception to this is AN (Figure 4.3b), where we see a lot of variation, and a dramatic drop in recall-precision when training on the intersection labels (AvP = 0.460). This is likely to be because the network has only been trained using high confidence points from the three other islands, which means uncertain points (which Figure 4.1 showed account for over half the dataset in AN) are not predicted by the network at test stage. We also see that using different training labels can improve model performance when compared to training on the majority vote. For example in GC (Figure 4.3d) training with observer three's labels gives a better result (AvP = 0.735) compared to the majority (AvP = 0.717), and brings us nearer the level of inter-observer accuracy. The mAP scores (Appendix Table A.2) show that on average the reference observer's labels give us the best results across all images (mAP = 0.76). This is closely followed by the 'mixed' ground truth (mAP = 0.75).

4.5 Discussion

4.5.1 Manual Counts

Analysis of manual counts show that the level of inter-observer variation is significant, and depends on the properties of the individual images. For example in our dataset we have very high inter-observer agreement for AP (0.92), and much lower inter-observer agreement for AN (0.67). Visual inspection of the images suggests the main reasons are the contrast between albatrosses and background vegetation, and the number of visually similar background objects such as rocks. For demonstration we present examples of albatrosses which have been labelled by all six, exactly three, and only one observer in Figure 4.4. While points labelled by all six observers tend to have a clear contrast and relatively uniform background, for low confidence points there is often a very weak signature, or uneven background. For example many points labelled by one observer are in areas of rocky brown terrain. Even though the white dots themselves have a similar appearance to albatrosses, the context of the background may have lead other observers to discount them. For novice observers it may be beneficial to give further training, such as information on likely nesting habitat, or perform pre-processing steps to reduce the search area to only the most likely habitat regions. It may be possible to gain this information from metrics such as the normalised difference vegetation index [91],

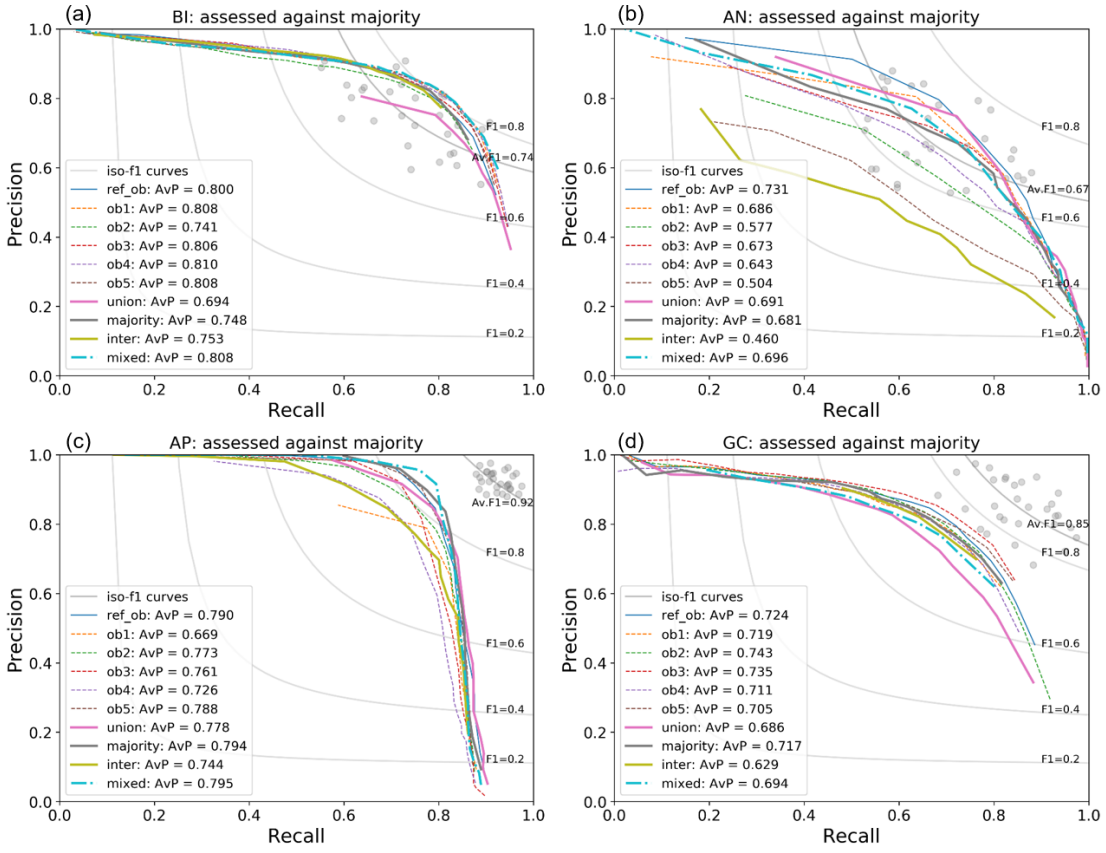

FIGURE 4.3: Precision-recall curves for models trained on different ground truth labels, for **(a)** Bird Island, **(b)** Annenkov Island, **(c)** Apotres Island and **(d)** Grande Coulee. We train each model using leave-one-island-out cross validation, and a different set of ground truth labels (coloured lines). All results are assessed against the majority vote labels. Gray points show the individual inter-observer precision-recall points, and grey lines show the corresponding F1-scores. The average precision (AvP) is the area under the precision-recall curve.

and from digital elevation models which could be used to exclude regions with steep slopes. These steps could also be incorporated into the automated approach, which would perhaps improve detection results and also reduce processing times.

Interestingly in AP observers have labelled points below cloud cover (Figure 4.4), in many cases with majority agreement. Viewed as isolated examples we would perhaps not expect to see such high confidence in these annotations, as the clouds mask out most colour information. This could be a consequence of the labelling procedure, where the image is scanned as a grid, allowing observers to build a wider picture of the distribution of albatrosses. If points below cloud cover are near a cluster of more clearly discernible albatrosses, then observers may be more likely to annotate them. To remove this bias patches could be presented in a randomised way, to make interpretation of the images more standardised. It may also be interesting to investigate whether the cloud cover examples could be enhanced by using dehazing algorithms [59, 74], to remove cloud

cover as a pre-processing step. Incorporating the near-infrared (NIR) band could offer a means of achieving this, as research has shown NIR information can be combined with RGB channels to reduce the appearance of haze in images [38]. Enhancing RGB images by incorporating additional multispectral information could improve detection during manual analysis, and potentially reduce the amount of inter-observer variation and uncertainty. Human observers could then benefit from the information provided in non-RGB bands (for WV-3 this includes NIR-1, NIR-2, red-edge, yellow and coastal), which can be trivially input to the CNN as imagery does not need to be visually interpretable. These approaches could benefit surveys of other species, for example it has been shown that the coastal band can aid detection of whales, since it penetrates further through the water column [43]. Methods for enhancing and adjusting images could also help to reduce variation in manual counts across different images.

4.5.2 Network Performance

Our CNN results highlight the importance of assessing performance within the context of observer accuracy. This is particularly evident in AN, where our best performing networks achieve peak F1-scores of approximately 0.7. While in general we would aim to improve this score towards 100%, when we assess observer accuracy we find that these network results are already in the range of human performance (average F1-score of 0.67). This is also true for our results on BI, with an average observer accuracy of 0.74 and peak network performance of over 0.8. However, even choosing the best scoring networks for AP and GC, we fall below our average observer F1-scores. The primary cause for this are the misclassifications discussed previously in Chapter 3 (Section 3.4.2). For AP this was largely due to the network failing to detect albatrosses through cloud cover, leading to a number of false negatives which lowers the overall recall scores. There were also a small number of false positive detections in the ocean due to spectral distortion and noise. For GC a large cluster of false positive detections of rocks, which were clearly in unsuitable habitat far from the main colony areas, lowered the overall precision score. As we discussed in Chapter 3, these misclassifications could be quickly filtered manually or addressed through further additions to the network, and would bring the results for these islands within the range of human performance.

We find the choice of ground truth labels has an impact on both the training and assessment of our CNN. Choosing a single observer's labels is not recommended, as we can see significant variation in results, particularly at the assessment stage. In terms of combining counts from multiple observers, using the majority vote can improve our assessment, as many low certainty points are removed from the ground truth. However this means discounting a large number of labels which, although low confidence, still

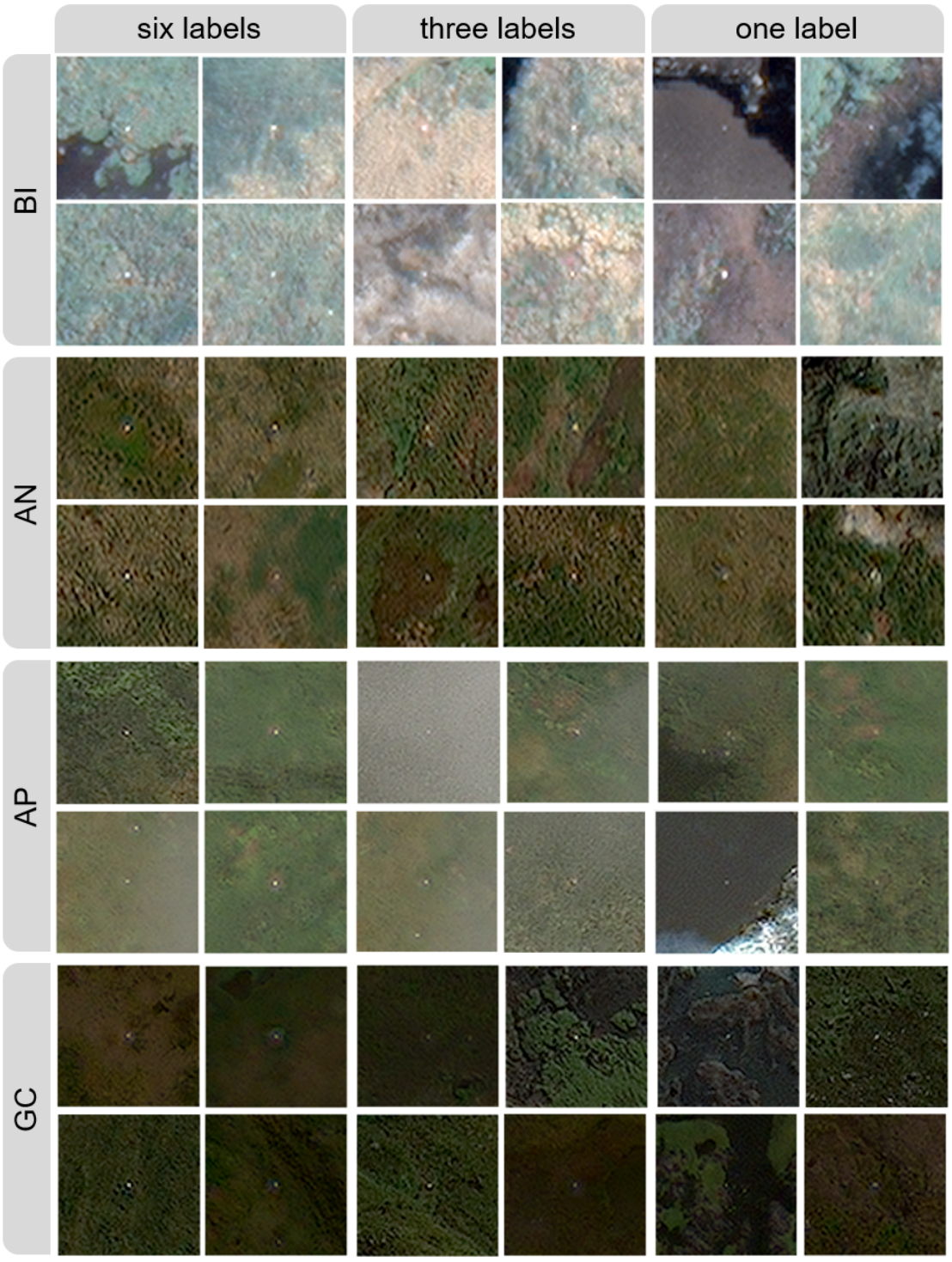

FIGURE 4.4: Examples of albatrosses labelled by all six, exactly three, and only one observer, for each of the four islands. Imagery from Maxar's WorldView-3 satellite © 2021 DigitalGlobe, Inc., a Maxar company.

contain useful information. At the training stage we found that using a mixture of labels can be a good alternative which avoids this problem. In the future other alternatives for combining ground truth could be investigated, including having a probabilistic ground truth (e.g 1 if labelled by all observers, 0.5 if labelled by half etc). Observers could also be asked to assign a confidence class to each of their detections, for example if they think it is certain, likely, or possible. This method has been employed in surveys of larger species such as whales [28, 39]. Giving observers the option to rank their detections in this way, rather than forcing a definite yes/no choice, may be equally informative as checking the agreement between multiple observers. This would potentially reduce the number of observers required to gauge uncertainty, although it may significantly increase the time taken to manually label images. Finally there is potential to incorporate labels from different observers into the architecture of the CNN itself, as shown in recent research by Rodrigues et al. [123]. In this method a crowd-layer is added to the end of the network, allowing each observer to be modelled separately, only combining predictions in the final stages by the use of learned weights. They showed that modelling individual observers in this way can improve classification performance, and is a useful approach for any crowd-source annotated datasets.

4.5.3 Recommendations and Applications

While we have used wandering albatrosses as our study case, it is very likely that topics discussed in this Chapter apply to VHR satellite surveys of other species, as well as UAV and aerial surveys where image resolution is coarse and observer confidence is limited. For example, observer uncertainty has been noted in aerial studies where there is a low contrast between wildlife and background [20, 113], and when flight height and camera resolution results in comparatively low ground sample resolution in relation to the size of the target animal [17]. In particular we show that 1) manual counts of wildlife in satellite imagery can vary significantly between observers, and importantly observer confidence may differ between images, 2) new images may present challenges to CNNs trained on small VHR satellite datasets, and 3) the choice of ground truth can impact supervised schemes at both the assessment and training stage. We also highlight the importance of assessing results of automated approaches within the context of inter-observer variation, for each unique image, to accurately gauge performance.

In general the results of our automated CNN approach are promising; in our leave-one-island-out cross validation we match human performance for two out of four islands, and for the other two misclassifications are mostly obvious and therefore easy to filter. We hope that these methods can facilitate future satellite surveys of wandering albatrosses, with increased frequency and reduced cost. In particular, since the global breeding

population of wandering albatrosses nest in relatively few locations (approximately 18% on South Georgia, 44% on Prince Edward Islands and 38% on Crozet and Kerguelen Islands [2]), the potential to conduct a global population survey using VHR satellite is feasible. The methods could also be adapted for other species, such as the northern royal albatross, which were shown to be visible in WV-3 imagery in previous studies [42]. If repeated at regular intervals this would vastly improve our knowledge of spatial and temporal population dynamics, for a species which is of high conservation concern.

4.6 Conclusion

In this Chapter we investigated inter-observer variation in satellite counts of wandering albatrosses. We found that the agreement between observers differed for the four islands in our dataset, resulting in different accuracy scores for each image. When placing the results of our U-Net detection method (presented in Chapter 3) in the context of human performance, we find that we match human accuracy for two our of the four images, with the other two having obvious misclassifications which would be easy to filter manually. We also conduct experiments into how the choice of ground truth labels can impact the supervised training scheme, both at the assessment and training stage. This proves to be an important factor, which should be considered in future applications. In the next Chapter we will present work on another seabird remote sensing dataset, this time focusing on UAV monitoring of canopy nesting species. We will again examine the application of CNNs to the task, and consider platform and species specific challenges when developing an automated approach.

Chapter 5

Single-view Detection of Abbott's Boobies in UAV Imagery

5.1 Overview

In Chapters 3 and 4, we investigated how CNNs can be used to count nesting seabirds in VHR satellite imagery. We used a U-Net architecture to detect wandering albatrosses, which are amenable to survey by satellite due to their size, colour, and open nesting habitat. In our second application we will focus on automated detection of a different seabird - the Abbott's booby - a canopy nesting species endemic to Christmas Island. Due to their smaller size and the complexity of their nesting habitat, alternative survey methods are required. Recently researchers at LaTrobe University have tested the application of UAVs to complete this task, in particular using the process of Structure from Motion (SfM) to generate 3D models of the survey area from multiple overlapping 2D images. In this Chapter we will outline properties of the Abbott's booby dataset, including how the images were collected, processed and annotated. We use the annotated data to train a Faster R-CNN architecture to the detection task, and investigate the performance on the 2D raw UAV images. To conclude, we outline the potential of a multi-view approach, where projection information gained in the SfM process could be used to map 2D raw image detections on to the 3D model. This would allow multiple views of the same nest site to be mapped together, potentially increasing detection rates for this canopy nesting species.

5.2 Introduction

The Abbott's booby (*Papasula abotti*), although once wide ranging across the Pacific and Indian Ocean, is now classed as Endangered under the IUCN Redlist [1]. It is endemic to Christmas Island, an Australian overseas territory off the south coast of Indonesia, where it nests in the emergent trees of the tropical forest canopy. However, due to the challenging forest terrain and their top-of-canopy position (up to 40m above the forest floor), identifying nests from the ground can be difficult. The last complete survey, still seen as the most extensive and accurate, was conducted on foot using binoculars in 1991 [159]. However since then the composition of vegetation on the canopy floor has become extremely dense [109], which would make the equivalent survey prohibitively time consuming or impossible in many areas in the present day. Alternative survey methods are being examined to perform the population census, which is in urgent need of updating.

In recent years, UAVs have been used to survey a number of different species, including birds. In comparison to satellite imagery (used to monitor albatrosses in Chapters 3 and 4), UAVs offer higher resolution and more flexibility, allowing researchers to tailor surveys to their own requirements. While UAVs have been used to evaluate nesting status of canopy breeding birds on a nest-by-nest basis [151], to date all larger scale UAV surveys of birds have targeted ground-nesting species (e.g penguins [118], albatrosses [103] and waterbirds [32]) which occupy open and flat habitat. For tree-nesting birds, such as the Abbott's booby, the complex surface of the canopy makes detection more challenging. UAV images tend to be collected using nadir photography, where the camera axis is angled directly below the UAV [23]. Even for species that only occupy the top level of the canopy, leaves and branches may partially or completely obscure them when viewed from this perspective alone. While Lipka et al. (unpublished) have shown that Abbott's booby nests can be seen in both UAV and helicopter surveys, the extent to which nests are missed or misidentified is not certain.

For many elusive forest species thermal UAV imagery has been used to overcome this detection challenge [102, 135]. Using this method wildlife can be detected by their thermal contrast with the surrounding vegetation, allowing researchers to effectively "see through" the canopy. However, there can be practical limitations when timing thermal surveys, particularly in tropical regions. Ideally flights will be conducted when the thermal contrast is strongest, for example very early in the morning or during the night [55]. This can add further constraints on survey teams, especially on small islands where there are a limited number of personnel. In addition, thermal imagery can only be used for surveys which target the direct detection of the animal. For other species, including orangutans [106] and Abbott's boobies, nests (or other signatures such as guano

accumulation on leaves) can be more valuable indicators of breeding sites. Without a thermal signature these important breeding site indicators would be difficult to discern.

Recently researchers at LaTrobe University have examined an alternative method for surveying Abbott's boobies in UAV imagery, using Structure from Motion (SfM) photogrammetry [127]. In SfM multiple overlapping 2D *raw images* (here defined as the unprocessed JPEG images captured by the UAV), taken from different locations, are processed together to build a 3D model. The method can be used to render a 3D model of any object or landscape, and is widely used for mapping forest canopies [31, 139]. Often the raw images are stitched together to form a single *orthomosaic* of the surveyed area, using a mosaicing approach guided by the SfM model. This orthomosaic is comprised mostly of the nadir viewpoints, and in complex surfaces such as forest canopy can appear significantly distorted. This can make it difficult to identify small objects, such as wildlife. However, using the underlying raw images collected by the UAV, multiple perspectives can be obtained of the same point in space. This provides more information, and can increase the likelihood of detection when the object is obscured from a certain angles, but visible in others.

Lipka et al. (unpublished) have conducted a manual analysis of the dataset presented in this chapter, and found that detection rates of Abbott's boobies significantly improve when using the raw images compared to the orthomosaic alone. This is for two main reasons: 1) we gain multiple perspectives of nest sites and reduce the issue of occlusion by branches and leaves and 2) the raw images do not go through any processing, so have higher quality and resolution than the final orthomosaic. While the benefits of using the raw images are clear, manually analysing all raw images adds a significant amount of processing time, in this dataset equivalent to covering roughly ten times the area compared to direct analysis of the orthomosaic. To make the approach feasible for island-wide surveys, automated methods are needed to process the data more efficiently.

As discussed in Chapter 2, CNNs have shown state-of-the-art performance in a number of tasks, including object detection. In recent years there have been a number of papers applying CNNs to detect wildlife in UAV imagery, including seabirds [75]. Specifically, Faster R-CNN was recently shown to outperform four other popular detection networks (including the two-stage R-FCN, and single-stage SSD, Retinanet and YOLO) at the task of detecting birds in UAV imagery [70]. CNN results generally benefit from variety in images, which can be artificially increased through different augmentations. In this sense network training will also benefit from using the raw UAV images, which naturally show a variety of perspectives, compared to training on the stitched orthomosaics.

In this chapter we will assess how well Faster R-CNN performs at the task of detecting Abbott's booby nest sites in raw UAV images. Objects of interest include adult birds,

empty and occupied nests, and guano staining on leaves. We will use an annotated dataset of 32 forest plots, collected across Christmas Island by Lipka et al. (unpublished data), to train and test the methods. In this instance, we ignore the multi-view information gained from overlapping images, and treat each raw image as independent. We assess how well the network performs in comparison to an expert observer. To conclude, we use labels provided in the manual analysis to assess the potential of a multi-view approach - where detections from multiple overlapping raw images could be combined to improve detection rates.

5.3 Methodology

5.3.1 Data Collection

In the study, 32 plots each of approximately 4 ha were selected across Christmas Island (105°40'E, 10°25'S). These are each given a unique reference number (N01, N02, etc.). Survey transects were designed and pre-programmed in DroneDeploy, mapping for DJI (V. 4.1.0, DroneDeploy, San Fransico, California, USA). The UAV used to conduct the surveys was a DJI Mavic Pro (MP1, SZ DJI Technology Co., Ltd., Shenzhen, China) with a FC220 camera (sensor ½.3" 1/2.3" (CMOS), FOV 78.8° 26 mm, f/2.2 aperture) in combination with an Apple IPad mini 4 (MK9P2X/A, Apple Pty Ltd, Sydney, New South Wales, Australia) or a Samsung Galaxy S7 (SM-G930F, Samsung Electronics Co., Ltd, Suwon, South Korea). Transects were flown with an average flight height of between 30-50m above the canopy, and a flight speed ranging between 5 and 15m/s (for an example transect see Figure 5.1a). Designs were programmed such that raw images (which here we define as the original JPEG images before SfM processing, each of dimension 4000×3000 pixels) were collected with front overlaps between 80 and 90%, and side overlaps between 80 and 86%. On average there were 204±30 raw images collected per plot, with a range of 136 to 262 (Figure 5.3a).

5.3.2 Data Pre-processing

5.3.2.1 Orthomosaic Generation

After UAV data was collected in the field, raw images were processed into orthomosaics using SfM implemented in Agisoft (V 1.5.2, Agisoft LLC, St. Petersburg, Russia). We present a brief outline of this procedure in Figure 5.1. First all images were aligned using high accuracy, generic preselection, a 40,000 key point limit, a 10,000 tie point limit, and adaptive camera model fitting. Following this high quality dense point clouds were built

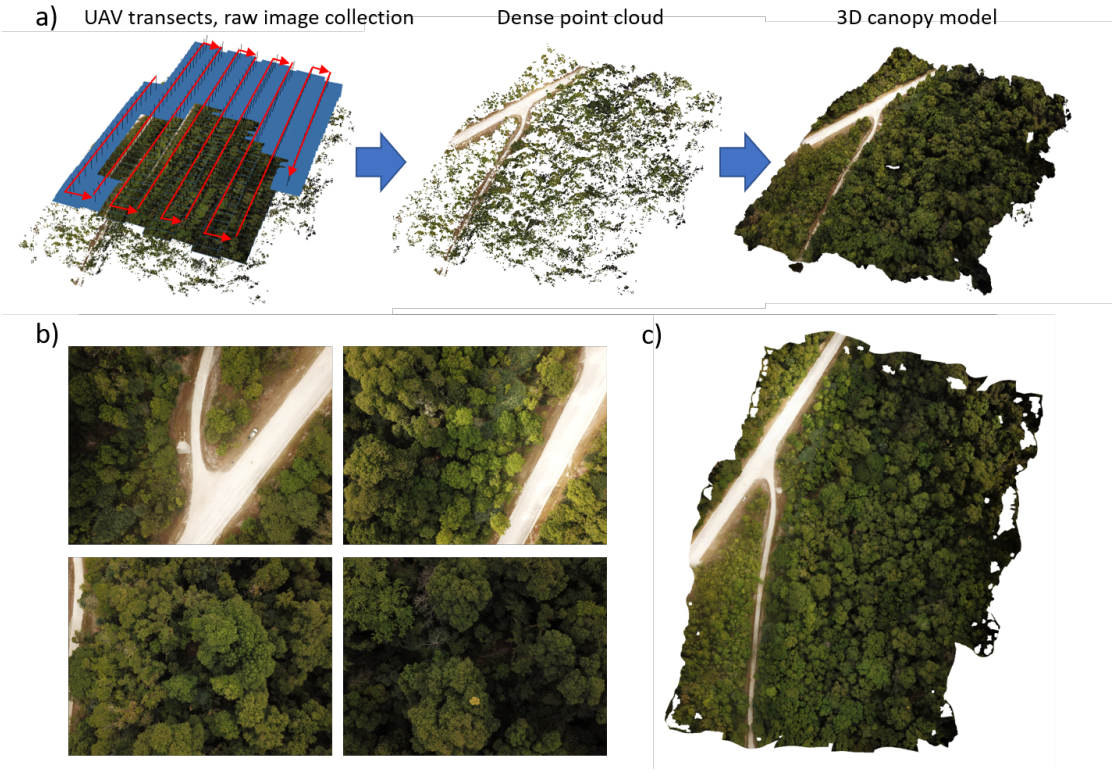

FIGURE 5.1: Example of SfM processing for plot N02. a) Raw images are captured by the UAV, SfM is then used to construct a dense point cloud, and a final 3D model. b) Examples of raw images collected by the UAV (4000 × 3000 pixels). c) The final orthomosaic.

using aggressive depth filtering mode. These dense point clouds were used to generate digital elevation models (DEMs). In the final stage the DEMs and point clouds were used to generate orthorectified orthomosaic images (Figure 5.1c). Final orthomosaics had an average ground sample distance of 2.18±0.30 cm/px.

5.3.2.2 Expert Annotation

All orthomosaics were visually scanned by an expert observer using the same software used for SfM processing (Agisoft V 1.5.2, Agisoft LLC, St. Petersburg, Russia). A grid was overlaid on the orthomosaic, and each cell was systematically scanned for seven main Abbott's booby AB objects (for a summary with descriptions see Table 5.1, for image examples see Figure 5.2a). Five of these classes can be used to determine potential breeding sites, and include guano staining on leaves (*guano*), empty and occupied nests (*empty nest*, *nest*), and adult birds (*adult*, *2ndA*). In the final manual assessment, these classes were used in combination to determine potential breeding sites. For instance, an empty nest next to a large accumulation of guano can be a better indicator of a

permanent nest site (though unoccupied at time of survey), than a single booby perched in a tree with no other classes present.

In addition, there were a number of cases where flying birds were captured at the time of survey. These were also manually identified in the imagery, however are treated as a special case in this analysis for two main reasons: (1) they are not useful for breeding site identification so would ideally be excluded in an automated detection system, and (2) since they are not stationary the same bird often appears at different locations in multiple raw images captured by the UAV. Flying birds were labelled as two separate classes: flying Abbott's boobies (*FAB*) and *other birds*, which included goshawks, frigate birds, red footed boobies, imperial pigeons and golden bosuns. All seven classes were tagged with a single marker placed as central to the object as possible (or in the case of guano which had a less uniform size, at the point of highest intensity).

TABLE 5.1: Summary of classes identified in manual analysis.

Class name	**Description**
Guano	*White faecal staining on leaves*
Adult	*Adult Abbott's booby perched in a tree, but not on nest*
2ndA	*Second adult Abbott's booby in close proximity to another*
Nest	*Adult Abbott's booby on a nest*
Empty nest	*Nest with no Abbott's booby present*
FAB	*A flying Abbott's booby*
Other bird	*A different species of bird (e.g goshawk, frigate bird)*

Following the analysis of the orthomosaics, each of the underlying raw images was scanned using the same annotation procedure. In Agisoft, any placed marker will automatically appear in the corresponding location in all other raw image viewpoints as well as the orthomosaic. This feature was used to guide the annotation process. Objects identified in multiple raw images were assigned a unique reference label, so that they could be assessed as the same point in the orthomosaic. In many cases objects were only visible in the raw images, as image stitching and occlusion by branches made them impossible to discern in the orthomosaic (Figure 5.2b).

5.3.2.3 Class Balancing

The distribution of classes identified in the raw images using the manual analysis procedure is presented in Figure 5.3b. *Guano* is the most prevalent (7639 examples) followed by *nests* (4150 examples), while other classes such as *other birds* and *adults* are less common (51 and 127 examples respectively). In Figure 5.3c we summarise the underlying number of unique objects (i.e ignoring multiple views), which are present in the plots. On average we have ten views of each object in the raw images, hence there are

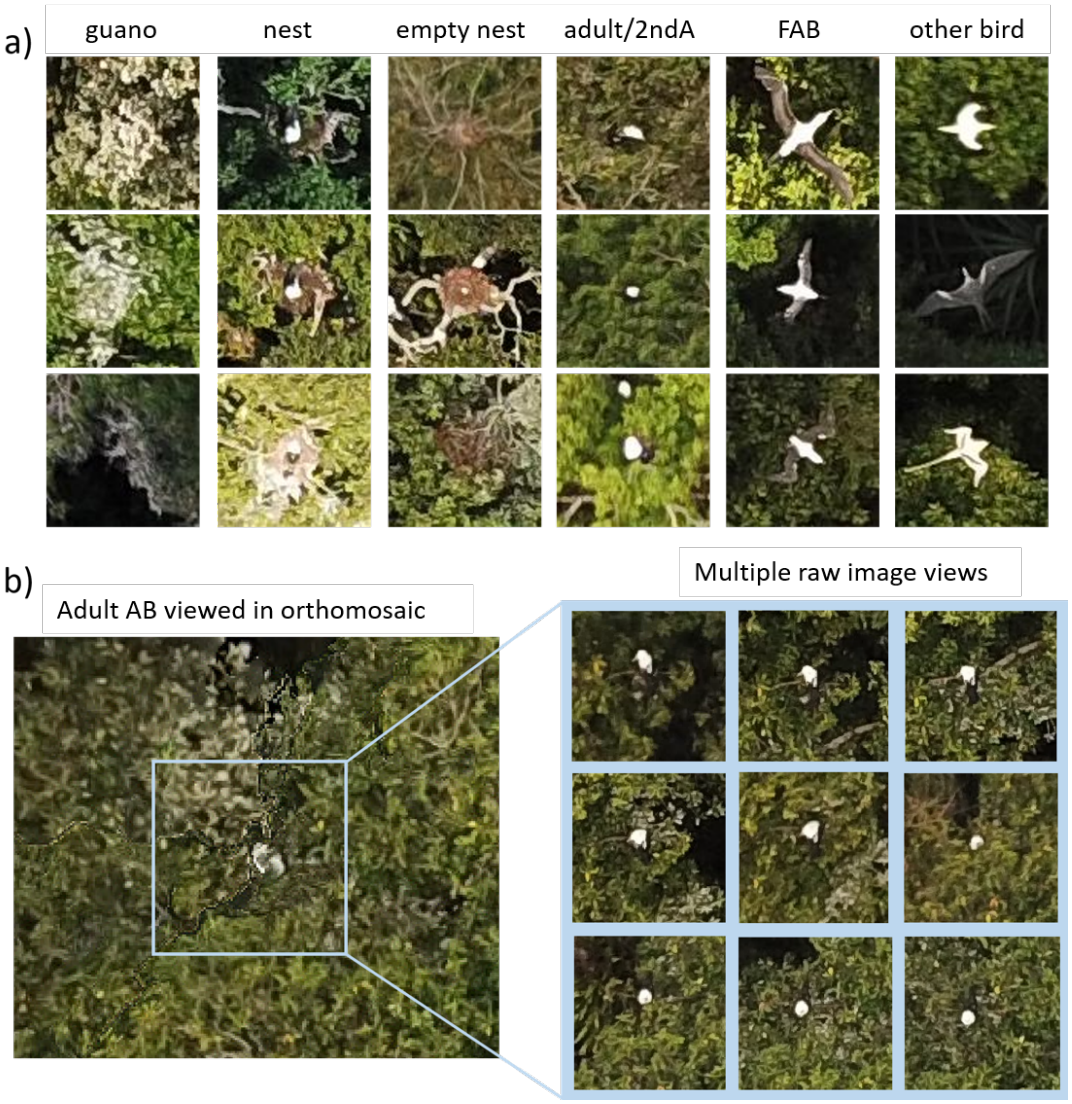

FIGURE 5.2: a) Examples patches showing different classes; b) comparison of an adult Abbott's booby (AB) viewed in the orthomosaic, where there is just a single view and distortion due to image stitching, and the same bird viewed in nine corresponding raw images.

approximately ten times fewer real world objects (e.g 722 examples of *guano*, 484 *nests* and only 11 *adult* birds identified in the plots).

To prepare our data for Faster R-CNN we first simplify our list of seven classes by combining similar classes. We choose to form a two class detection problem of (i) *guano* and (ii) all other *Features of Interest* (*FOI*). This helps to deal with two challenges. Firstly, there are many classes which appear very visually similar, which can make it challenging for the network to differentiate. For instance *2ndA* and *adult* are visually identical when viewed in isolation (without knowledge of their proximity), and a *nest* is defined as a combination of two other classes (*empty nest* + *adult*). Preliminary tests showed that using these classes separately caused the network to label each detection with multiple

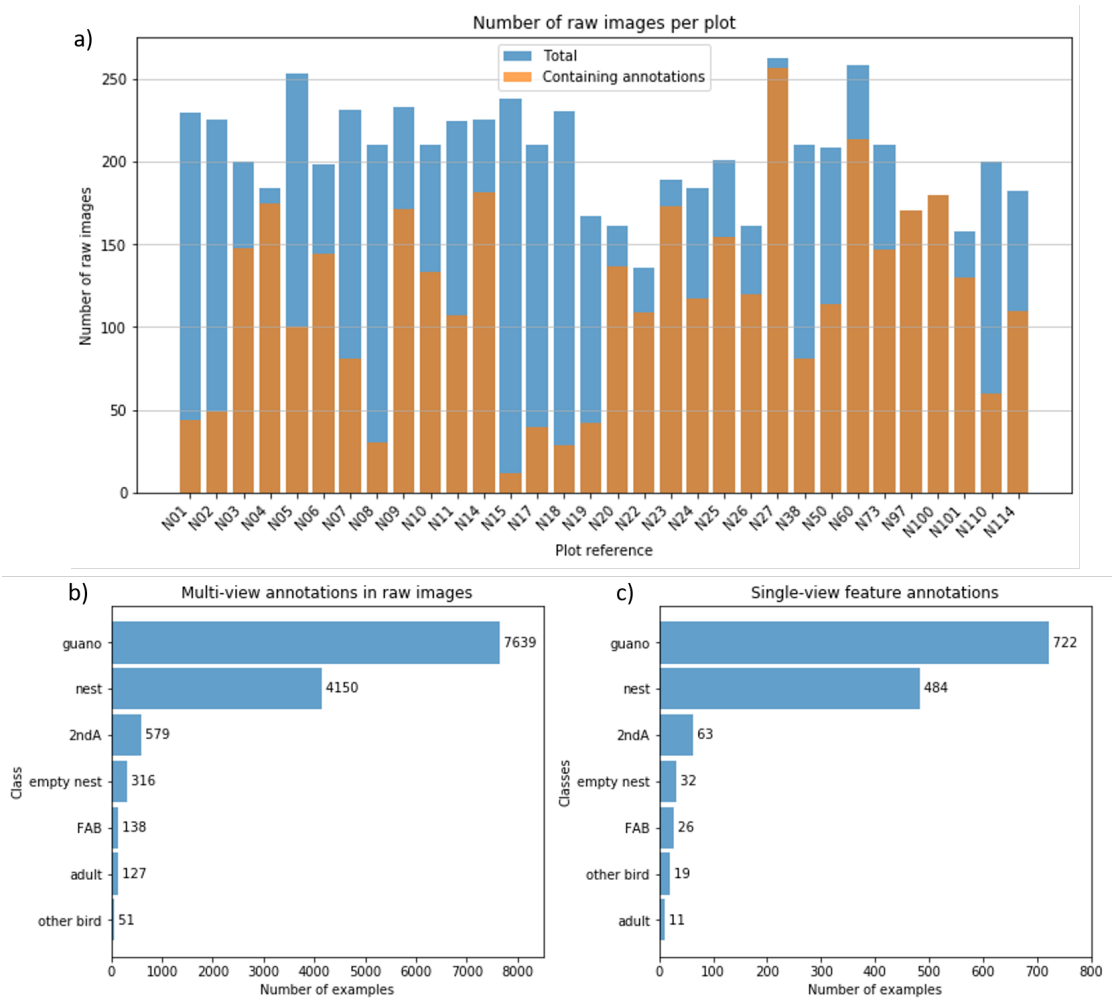

FIGURE 5.3: Summary of the ground truth annotations. a) The number of raw images collected per plot, including the fraction which contain any Abbott's booby objects. b) The number of examples of each class in the raw images. c) The number of unique objects identified, ignoring multiple-views.

class labels. Secondly combining classes helped to deal with class imbalance, where *nests* formed the majority of annotations compared to other rare classes such as *adult* and *other bird* (Figure 5.3b). After combining classes we were left with 5361 FOI labels and 7639 guano labels in the raw images.

Secondly, we reformatted the ground truth annotations from centre point labels to bounding boxes, which are required for Faster R-CNN. We took advantage of the fact that due to the aerial perspective and approximately fixed flight height most objects appear as a roughly fixed size across all images. We choose 50×50 pixel bounding boxes for FOI, which was determined by visual inspection of the imagery. For guano a fixed size bounding-box was harder to determine, since guano can cover different extents. In the initial stage, we define larger bounding boxes of 150×150 pixels for this class, again based on visual inspection of the images.

TABLE 5.2: Description of the different augmentations applied to the training data.

Augmentation	**Description**
None	*No augmentation applied*
Flips	*Flip input either horizontally, vertically or both*
Bbox jitter	*Edges of bounding boxes are randomly shifted by ±3 pixels*
Blur	*Blur the input using a random-sized kernel*
Brightness contrast	*Randomly change the brightness and contrast of the input*
Shift-scale-rotate 0.1	*Randomly apply affine transforms: translate, scale and rotate the input (scale limit=0.1)*
Shift-scale-rotate 0.2	*Randomly apply affine transforms: translate, scale and rotate the input (scale limit=0.2)*
Combination	*All above augmentation methods used in combination*

5.3.2.4 Train, Validation and Test Splits

To assess the robustness of the methods we train Faster R-CNN using a four-fold cross validation. In this dataset, it is likely that there is a high degree of correlation between raw images collected in each of the individual plots. For example, by design there is a large degree of overlap between raw images in the flight transects, they are collected at the same time of day in the same weather conditions, and they capture an area of similar habitat. To avoid bias in the train and test splits, we therefore partition the data by plot. The 32 plots are divided into four groups of eight. In each fold 20 plots are used for training, four for validation, and the remaining eight used as the test data. When assigning data folds we try to ensure that the classes are as balanced as possible. To determine this we generated 10,000 random splits of the data and chose the one which minimised the variation between the number of FOI and guano. The final folds and number of classes are presented in Appendix B.1.

5.3.2.5 Augmentation

We artificially increase the size of our training dataset by using different augmentation methods (summarised in Table 5.2). We choose augmentation suited to aerial imagery - namely flips and shifting, scaling and rotating images - since the orientation of images is not important. We also test randomly adjusting the brightness and contrast, which can emulate the changes in lighting conditions for different plots. Finally, we test adding blur, which is introduced by the motion of the UAV, and random shifts to the bounding box sizes. We test the effect of each of these individually, and compare to a baseline of applying no augmentation. We implement these using the Albumentations python library [18].

5.3.3 Network Architecture

5.3.3.1 Faster R-CNN

For our detection network we use the PyTorch [112] implementation of Faster R-CNN [122] with a ResNet-50-FPN [63] backbone, pre-trained on the COCO train2017 dataset [93]. A detailed overview of the Faster R-CNN architecture is given in Chapter 2 Section 2.10.2. Faster R-CNN was recently shown to outperform four other popular detection networks (including the two-stage R-FCN, and single-stage SSD, Retinanet and YOLO) at the task of detecting birds in UAV imagery [70]. In the pre-trained model the network is designed to classify the 91 classes in the COCO dataset. We replace the network head with our three class problem (FOI, guano and background). We train using a fine-tuning process, whereby all pre-trained weights are updated using our annotated raw image dataset.

5.3.4 Hyperparameters

To fine-tune our network we begin with the hyper-parameters recommended in the PyTorch documentation [112]. We use stochastic gradient descent with momentum of 0.9, weight decay 0.0005 and an initial learning rate of 0.001. We use a learning rate scheduler with a step size of 5 and gamma set to 0.1 (so that the learning rate is divided by 10 every 5 epochs). We used a batch size of 2, and trained the network for 15 epochs, saving the model at the point where mAP on the validation set was the highest. This state model was used to make the final predictions on the test set.

5.3.5 Hardware and Frameworks

Model training is performed on a PC workstation equipped with Intel i7-8700 CPU @ 3.20GHz, 32GB of RAM and NVIDIA Titan Xp graphics card with 12GB of GPU memory. PyTorch 1.12.0, Torchvision 0.13.0, CUDA 11.6, and Albumentations 1.0.3 were used in the training and inference pipeline.

5.3.6 Evaluation Metrics

To evaluate our model we use the Average Precision (AP) score for for each class, as well as the mean Average Precision (mAP) to average results across classes. We also report recall, precision and the F1-score for specific confidence thresholds. All metrics are described in detail in Chapter 2 Section 2.13.

TABLE 5.3: Effect of different augmentation methods on the results, in terms of average precision (AP) for each of the two classes (FOI and guano), and the mean average precision (mAP) averaged over classes. Scores are presented as the mean and standard deviation when averaged across the four test folds.

Augmentation	**FOI AP**	**Guano AP**	**mAP**
None	0.488 ± 0.047	0.414 ± 0.048	0.451 ± 0.060
Flips	0.513 ± 0.048	0.434 ± 0.056	0.473 ± 0.065
Bbox jitter	0.480 ± 0.052	0.429 ± 0.038	0.454 ± 0.052
Blur	0.490 ± 0.054	0.421 ± 0.042	0.455 ± 0.059
Brightness contrast	0.498 ± 0.052	0.422 ± 0.049	0.460 ± 0.063
Shift-scale-rotate 0.1	0.514 ± 0.053	0.472 ± 0.045	0.493 ± 0.053
Shift-scale-rotate 0.2	0.504 ± 0.049	0.465 ± 0.048	0.485 ± 0.052
Combination	$\mathbf{0.518 \pm 0.046}$	$\mathbf{0.472 \pm 0.049}$	$\mathbf{0.495 \pm 0.053}$

5.4 Results

The results of our augmentation experiments are presented in Table 5.3. We find that each proposed augmentation on its own improves the final mAP score, with shift-scale-rotate (with a scaling limit of 0.1) giving the biggest improvement compared to the baseline (improving the mAP from 0.451 to 0.493). We find that using a scaling limit of 0.2 for this was less successful, possibly because this over-exaggerated the variation in flight height naturally present in the dataset. For our final selection we use a combination of all augmentation methods (with scaling on shift-scale-rotate set to 0.1), which gives us a final mAP score of 0.495 when averaged across the folds (0.518 for FOI's and 0.472 for guano).

5.4.1 Faster R-CNN Results

The results of the best performing Faster R-CNN model are presented in Figure 5.4. For the training and validation curves (Figure 5.4a) we see that the training loss continues to decrease marginally before the final epoch, while the mAP on the validation data levels after approximately 10 epochs of training. In Figure 5.4b we present the precision-recall results for each of the four test data folds. For FOI we see that the average precision (AP) scores range across the folds, from 0.44 (fold 4) to 0.57 (fold 1). At peak, we achieve accuracy (in terms of F1-score) of 0.6 for FOI detections, and recall levels do not exceed 70% for any of the folds. In general we have lower scores for guano, with AP scores ranging from 0.41 (fold 3) to 0.52 (fold 1). Precision values for guano also tend to decrease more steeply in comparison to FOI predictions, with peak F1-accuracy of approximately 0.5.

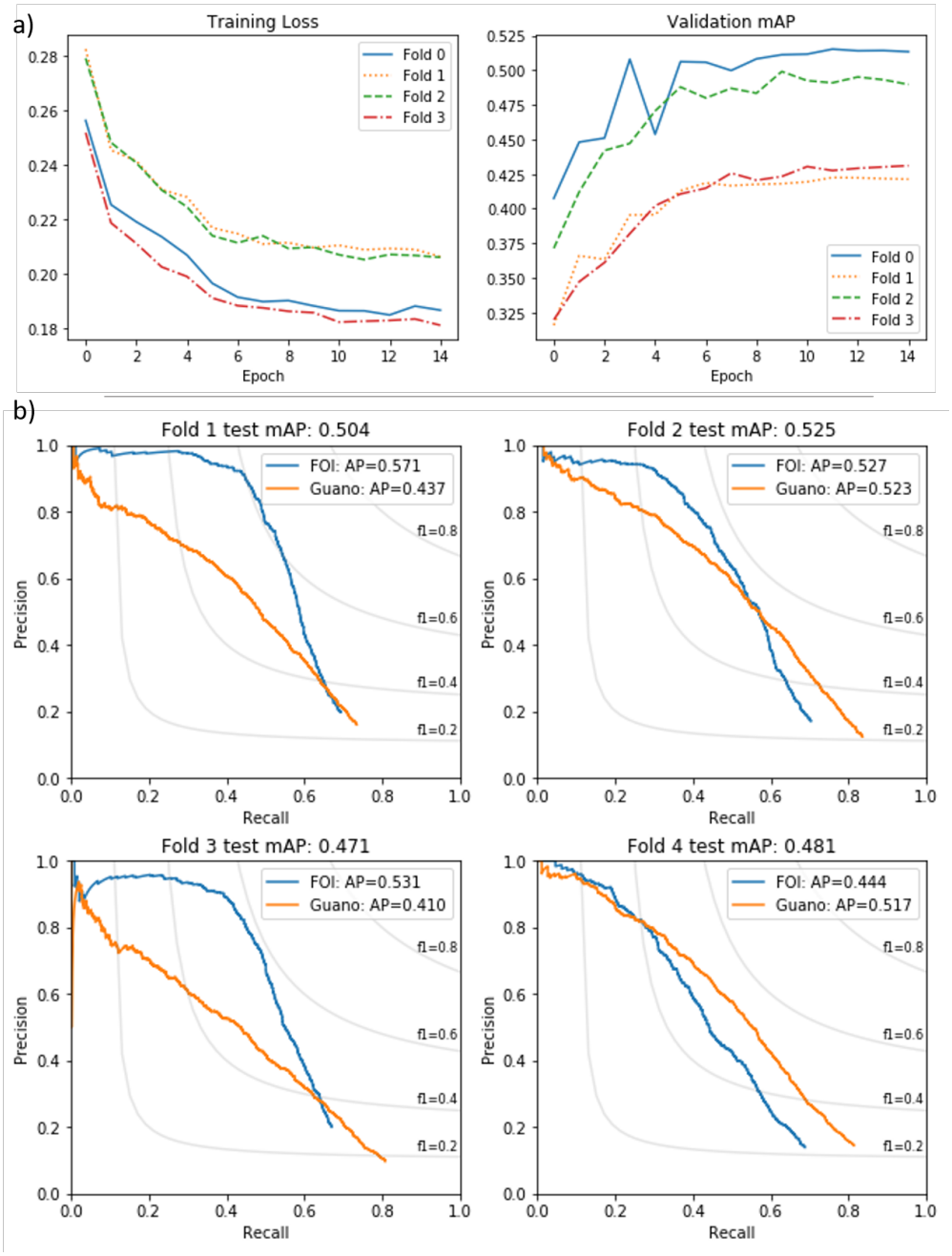

FIGURE 5.4: a) Training loss and validation mAP for the best scoring Faster R-CNN network. b) Precision-recall curves for the best scoring Faster R-CNN network, assessed on the test set for all four data folds.

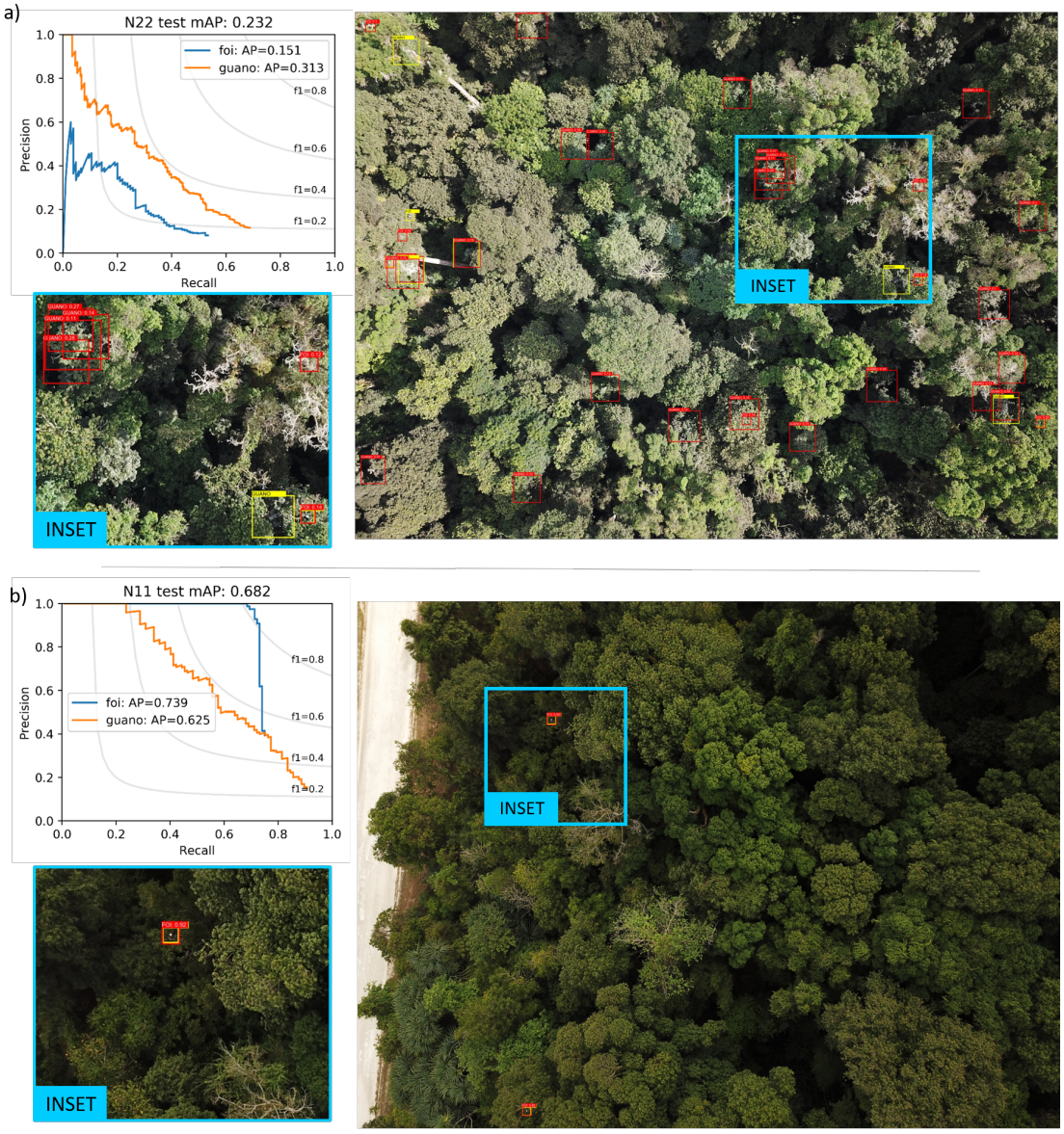

FIGURE 5.5: Example per-plot precision-recall curves and mean average precision (mAP) results for a) plot N22 which scores the lowest mAP at test stage, possibly due to challenging areas of white branches, and b) plot N11 which scores the highest mAP at test stage, with a FOI AP score of 0.739 and very few false positives.

5.5 Discussion

Lower scores for guano are not unexpected, as the class is less well defined than birds and nests. The boundary is unclear, it can vary in spread and intensity, and is more easily confused with other objects such as leaves in bright sunlight or white branches of trees. Guano is also used primarily as an auxiliary class for breeding site identification, adding certainty when other FOI are detected in close proximity. We bear these factors in mind when assessing guano detection results, and will primarily focus on correct identification of FOI in our analysis.

We also see that there is some variation in performance between the different data folds. In Appendix B.1-B.4 we present the precision-recall curves for each of the eight plots included in every data fold, with examples of the lowest and highest scoring individual plots presented in Figure 5.5. In some plots the network performs worse, with plot N22 scoring the lowest mAP overall ($mAP = 0.232$, Figure 5.5a). This may be due to the presence of large amounts of white branches, which are confused for FOI and guano in some cases, as well as the bright sunlight causing areas to look white and washed out. In contrast we achieve good results for other plots, such as N11 which has the highest mAP score of 0.682 (and FOI $AP = 0.739$, Figure 5.5b). In this plot there are few bare branches and the lighting appears more muted, meaning there are very few false positive FOI and guano predictions.

Visual inspection of the network detections gives us some important insights into the Faster R-CNN predictions. In Figure 5.6 we present some examples from four overlapping raw images captured by the UAV, and highlight two main points.

Firstly, the network often predicts guano with a series of overlapping bounding boxes (e.g Figure 5.6a). This is due to the fact that the boundary of guano is poorly defined. During manual annotation single markers were placed at the point where guano intensity was highest (i.e the whitest point), however the exact choice of this location is subjective and does not give an indication of the extent, shape or spread of the guano. While we define fixed bounding box sizes of 150×150 pixels, drawing more precise boundaries would require completely re-annotating the raw images which would be prohibitively time consuming in this instance. However, we can still leverage these overlapping detections, by changing our assessment to accept any guano prediction which intersects with a ground truth bounding box. In this way we can gain more information about the spread and extent of guano than would be possible with a single bounding box.

The second point is that, due to the different viewpoints images are captured at, the same object can appear more clearly in some images than others. This can lead to the network missing objects, or predicting them with lower confidence scores. For example, in Figure 5.6 the FOI in viewpoint (a) is quite unclear and detected with a low confidence of 25%, however in the subsequent views (b)-(d) the same FOI is detected with confidence scores exceeding 90%. This effect can lead to lower recall scores, and makes it more challenging to determine the final threshold to use for network predictions.

5.5.1 False Negative Analysis

We explore the impact of uncertainty in multi-view detections in Figure 5.7, focusing only on FOI opposed to guano. For this we threshold the Faster R-CNN predictions at

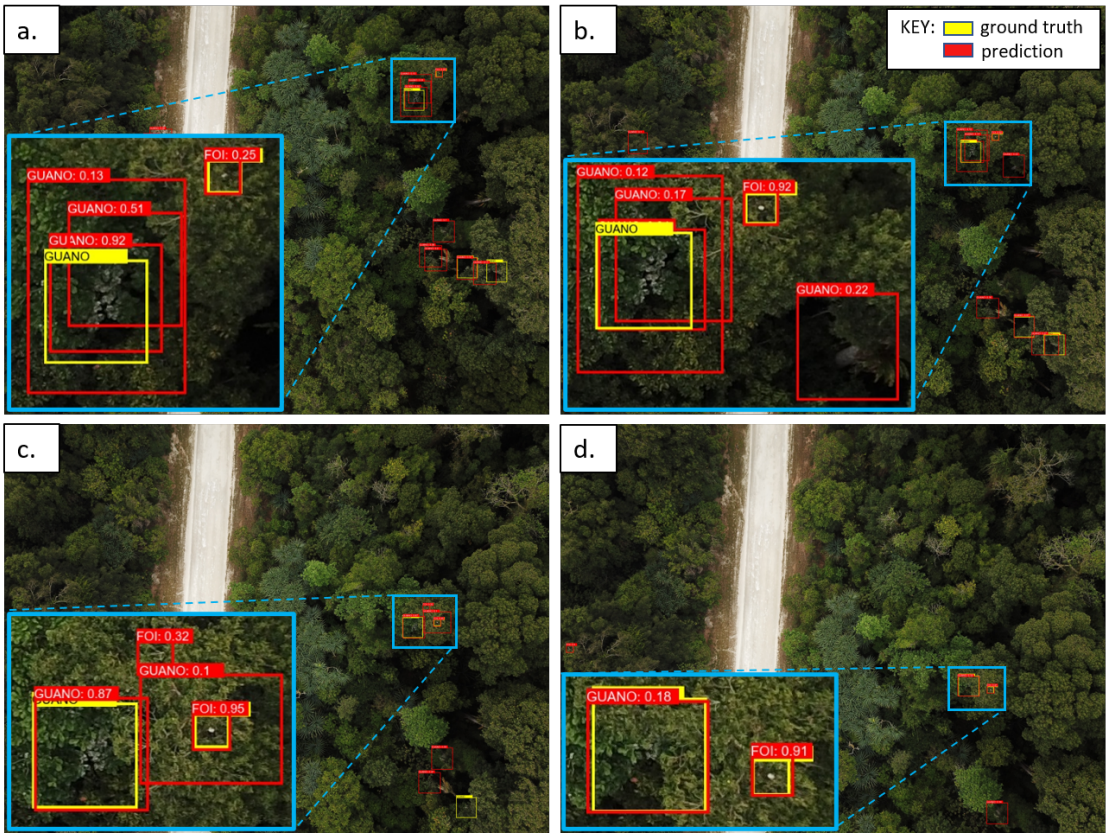

FIGURE 5.6: Examples of raw image detections on four sequential images from the UAV. Yellow boxes show ground truth annotations, red boxes show Faster R-CNN predictions (with class and confidence score). The blue boxes show a zoomed portion of the same nest site (FOI and guano) tracked between images (a)-(d). In image (a) the view of the FOI is less clear - and detected with only 25% confidence - whereas from the views in images (b)-(d) the FOI is detected with over 90% confidence. Guano does not cover a clearly defined area, and so multiple overlapping boxes are predicted, as seen in image insets (a) and (b).

0.1. We arrange multi-view FOI detections from the lowest scoring (according to Faster R-CNN prediction confidence, including false negatives which were completely missed by the network), to the highest. We can see that FOI which are predicted with lower confidence or missed are often not clearly visible. Whereas those taken from a clearer angle are predicted with high confidence (in many cases over 90%). In some cases all views are missed by the network (Figure 5.7 100% FN), while in others only a proportion are missed, or non at all.

The number of false negatives in multi-view detections could partly be attributed to the manual analysis process. In this Agisoft facilitated annotation by displaying any marker placed on a raw image at the same point in every overlapping raw. Where objects were less clear, this information could have increased the observer confidence for particular views, which may not have been assessed as a detection when seen in isolation. In our assessment of the raw image performance, these missed objects reduce the overall

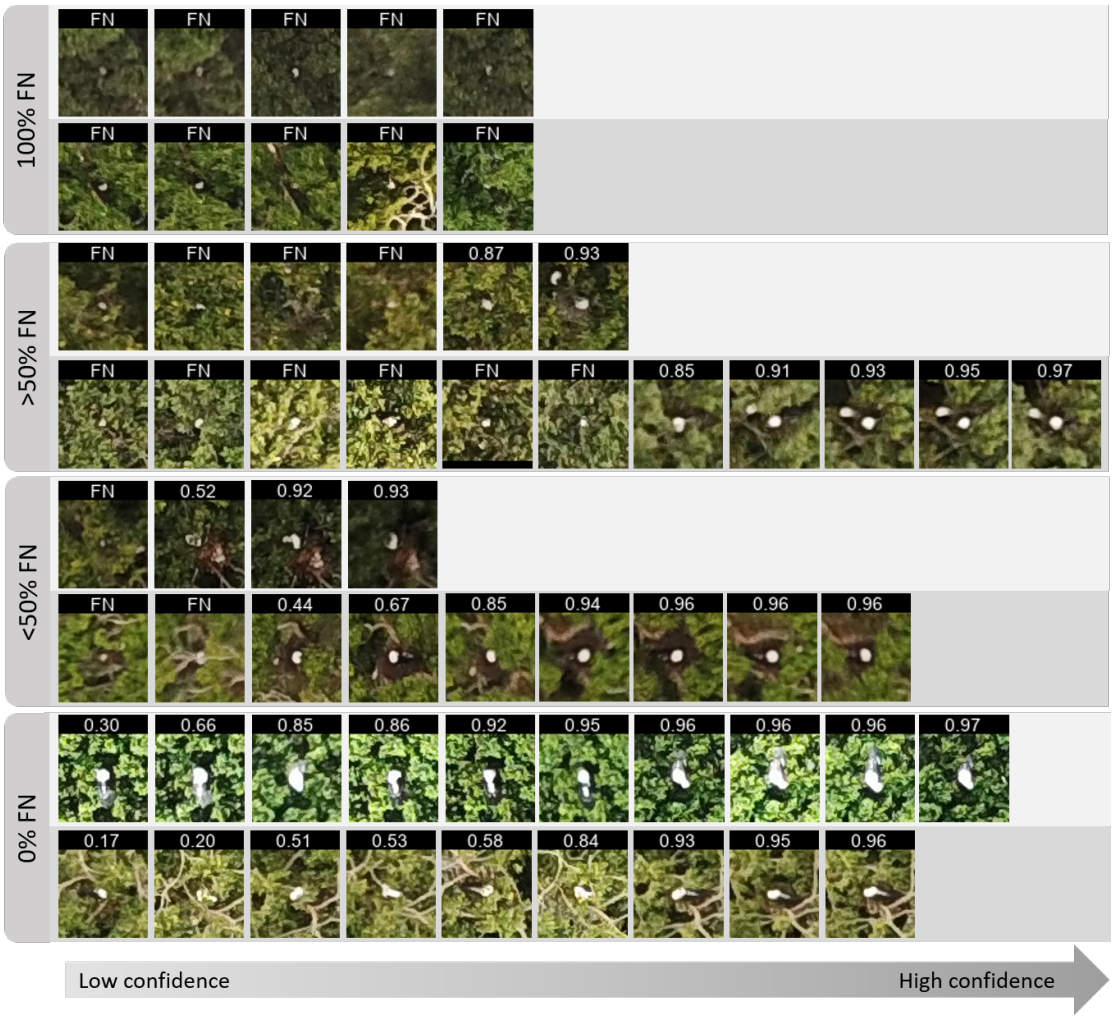

FIGURE 5.7: Multi-view detection examples, ordered by Faster R-CNN confidence score and including false negatives (FN). We show two examples each, divided into those where i) all views were missed by the network (100% FN) ii) over 50% FN, iii) under 50% FN, and iv) no views were missed. In many cases, low confidence or false negative FOI are very unclear or mostly obscured by leaves, while network confidence tends to increase for clearer views.

recall. However, the difficulty of the detection is not taken into consideration in this assessment, and would require further investigation into ground truth uncertainty. Furthermore, the inclusion of these challenging objects in the training datasets may impact the performance of the network, as discussed in the albatross dataset in Chapter 4.

5.5.2 False Positive Analysis

In Figure 5.8 we focus on false positive detections made by the network. For this we have no multi-view information, so instead present a selection of the highest scoring (above 90%) single examples. In many cases light coloured branches cause false detections, these can sometimes appear through the foliage as patches of white roughly the same

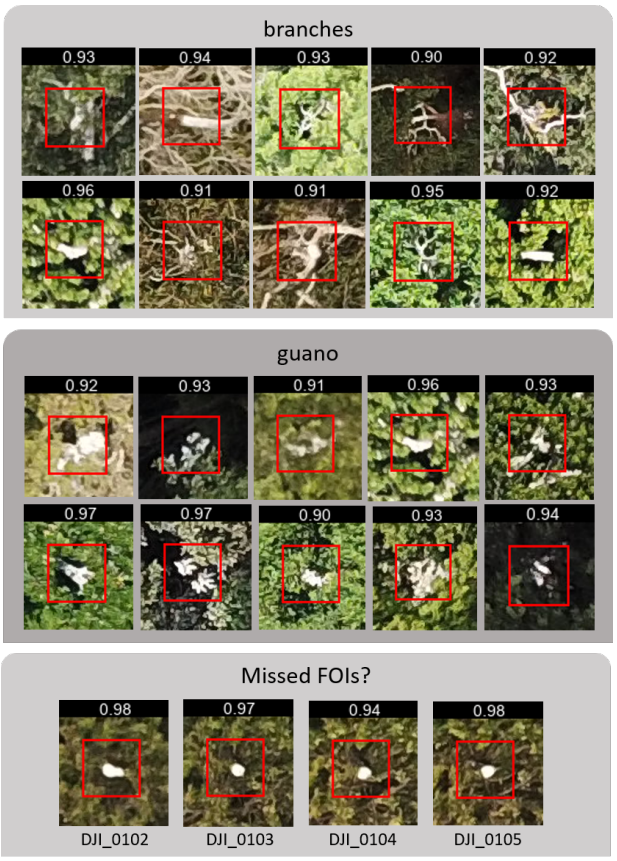

FIGURE 5.8: Examples of high scoring (> 90%) false positives. In many cases branches cause false detections, possibly due to their similarity to empty nests. Small patches of guano can be falsely predicted as FOI, as the bright white resembles adult birds. In a small number of cases there may be FOI which were missed in the manual analysis, in this example a similar object was detected in four sequential raw images (DJI_0102, DJI_0103 etc).

size as adult birds. In other cases it appears that guano is falsely detected with high confidence. Again this is due to the spectral similarity, and possibly also the fact that many FOI in the training data contain or are surrounded by guano staining on the leaves. Finally, a small number of false positives may in fact be birds which were missed in the annotation process in all, or a subset, of the overlapping raw images. We also note that these false detections may only appear in a single raw image, and may not have the support of detections in additional views. Using a multi-view approach outlined in the next section, it may be that many of these false positive detections would be discounted.

5.5.3 Potential of Multi-view Detection

We hypothesise that a multi-view approach - where the single-view Faster R-CNN detections are merged - will allow us to use these detections of varying confidence to our advantage. In this section we explore the potential of this approach by using the labels

provided in the manual analysis. In the multi-view approach, false negatives in the single-view can be compensated for by successful detections from other viewpoints. The case where this would not be possible is when all viewpoints are false negatives in the single-view Faster R-CNN detection stage.

In Figure 5.9a we explore in more detail the objects which are completely missed (i.e all viewpoints are classed as false negatives in our Faster R-CNN assessment). We plot the number of completely missed objects in terms of the total number of views and the underlying FOI classes. We find that the majority of missed objects are nests where there is only one view (accounting for 21 missed objects). This low number of viewpoints can be due to two reasons i) the FOI may be located near the boundary of the surveyed plot, where there are fewer overlapping UAV images, and ii) the FOI is impossible to detect in all other viewpoints. We can see that the number of missed nests decreases as the number of potential views increases.

We present image examples of the missed objects in Figure 5.9b. For the 21 nests with only one view, we see that most are quite unclear and challenging to detect visually, with the majority of the bird obscured by leaves. At the other end of the scale, where nests are missed in multiple (12, 14 and 19) views, nests are in challenging background of bare branches. This can potentially weaken the signature compared to detecting them against green foliage, as in the majority of examples in the dataset. Missed *empty nests* are also visually similar to branches, and have a small representation in the training dataset (a total of 32 in all plots). Missed second adult birds (*2ndA*) are unclear or also in challenging background, and in addition may be missed due to their proximity to another bounding box for the first adult bird. In this sense detection of 2ndA objects is not of as high importance. Finally in the case of missed *other birds* we see that black-bodied frigate birds, quite spectrally different to white Abbott's boobies, are missed in all raw images. In addition one view of a goshawk is missed, however this is in the shadow of the lower canopy. As noted in section 5.3.2.2, flying birds will not map to a fixed point when adopting a multi-view method, and ideally would be excluded from analysis as they do not indicate breeding sites. Hence in the next stage flying and other birds will not contribute to the final false negative count.

In all, 76 of our 635 FOI (12%) are missed in all views at the Faster R-CNN stage. However our assessment shows that these missed objects are largely challenging cases which are difficult to distinguish visually, are only visible in a small number of views, or are flying birds which we later wish to discount. All other objects, which were detected in at least one view in the raw images, can potentially be detected in a multi-view approach. In Figure 5.10 we assess the potential of this method. We consider that a FOI is correctly recalled if it has at least one successful Faster R-CNN detection in any view.

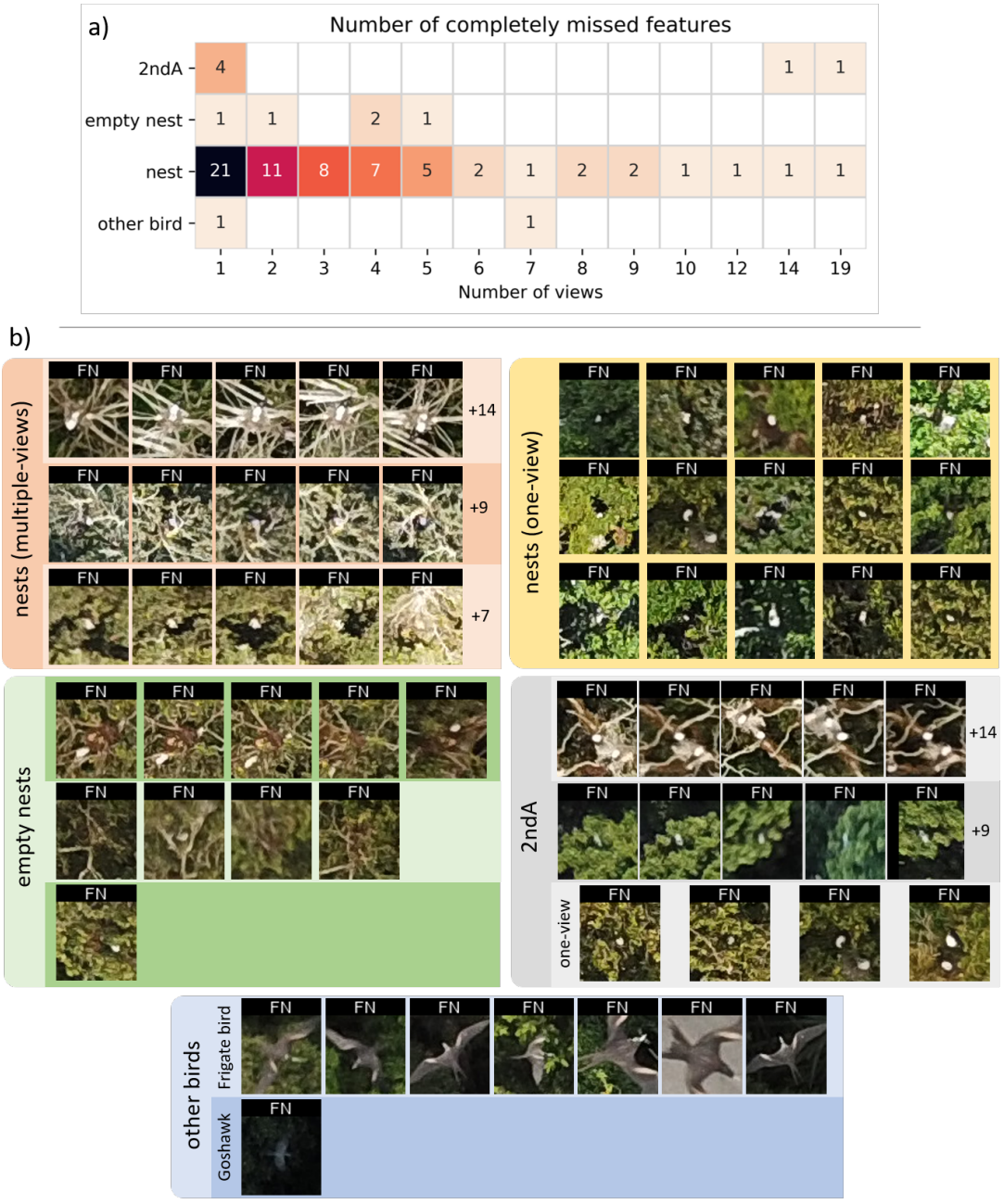

FIGURE 5.9: a) Heatmap with cells showing the number of completely missed objects (i.e false negative in every available view). We compare the total number of views and the detailed class labels. A large number of missed objects (21) are nests which are only annotated in a single raw image b) Gallery of completely missed objects. Many of the 21 *nests* with only one-view are unclear, and those which are missed in multiple-views (12, 14 and 19) are in challenging background with bare branches. Missed *empty nests* are also visually similar to branches and have a small representation in the training dataset. Second adult birds (*2ndA*) are unclear and may be missed due to proximity to another bounding box for the first adult bird. *Other birds* which were not detected are black-bodied frigate birds, and one-view of a goshawk which is in shadow.

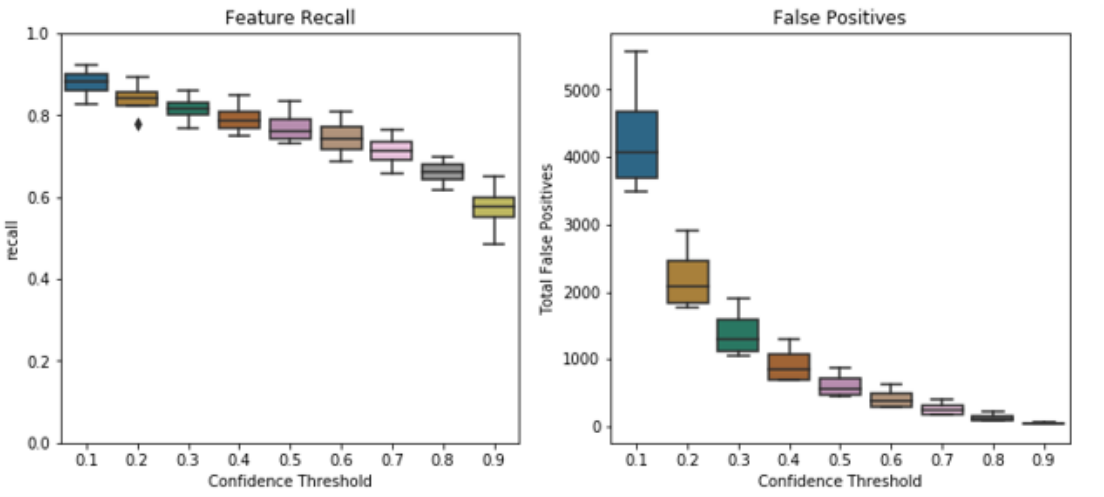

FIGURE 5.10: We assume a FOI is recalled when at least one view is correctly detected by Faster R-CNN. We assess (a) the potential object recall using different confidence thresholds, averaged across the four folds, and (b) potential false positives at the different confidence thresholds.

Since each detection has an associated confidence, the measure of a 'successful Faster R-CNN detection' will be dictated by our acceptance confidence threshold. We calculate the number of objects which are recalled in this way, for a range of confidence threshold values between 0.1 and 0.9. When averaged across the four folds, we see using a network confidence threshold of 0.1, we could potentially recall over 80% of FOI's on average. This recall level remains relatively constant when increasing the confidence threshold, remaining at around 80% until 0.4, and only falling to approximately 70% even with a very high threshold of 0.8. In the second plot, we assess how the number of false positives would be affected by this varying confidence threshold. In this case false positives are all considered as independent single detections, and so we see a high number (on average 4000) at threshold 0.1, which decreases exponentially as the threshold increases (e.g to under 1000 at 0.4). We suggest that in many cases these false positives will be one-off detections in a single raw image, and hence a multi-view approach could filter them out and improve the overall accuracy.

These results outline the possibility of a multi-view approach, which we will develop and assess in the next chapter (Chapter 6). This assessment was completed using labels from the manual analysis, while the final method will depend on how accurately the raw image detections will map to the 3D real-world location, and the subsequent clustering and classification of multi-view detections. However, in this Chapter we have seen that Faster R-CNN can offer promising results on a challenging dataset, and there is scope for the analysis to be improved when combining multiple viewpoints.

5.6 Conclusion

In this Chapter we presented a dataset of UAV images of nesting Abbott's boobies, processed using SfM. We trained a Faster R-CNN detection network to locate FOI and guano in the raw UAV images. We achieve mAP scores of 0.50±0.05, with average precision of 0.52±0.05 for FOI and 0.47±0.05 for guano, when averaged across the four test folds. In our results analysis we find that a proportion of missed detections are attributed to FOI being obscured from clear view by branches and leaves. We propose that a multi-view approach, where detections from multiple viewpoints are merged, will improve results. In Section 5.5.3 we explored the potential of this using labels provided in the manual analysis. We conclude that a multi-view approach has the potential to achieve over 80% recall, with the false negative objects largely being very challenging to detect. We hypothesise that a potentially large number of false positives will be filtered out due to being single outlying detections in the raw images, which will not map to multiple points in the orthomosaic. In the following Chapter we will outline our proposed multi-view approach. We will use projection information derived in the SfM process to cast 2D raw image detections into their corresponding 3D real world location. This will allow us to cluster multi-view detections, potentially allowing for improved detection of elusive canopy species.

Chapter 6

Multi-view Detection of Abbott's Boobies Using Structure from Motion

6.1 Overview

In the previous Chapter we outlined a method for detecting Abbott's booby *Features of Interest* (FOI; including empty and occupied nests, perched and flying birds) and *guano* in UAV images using Faster R-CNN. In our approach detections of the same object from different raw images were treated independently, and in many cases were missed in certain viewpoints due to partial occlusion. We proposed that a multi-view approach, where detections of the same object are merged, would improve object recall and reduce false positives. In this Chapter we outline this multi-view approach, which consists of three main stages: 2D-to-3D projection, clustering and classification. Our final method successfully recalls 62% of FOI (and 66% of guano) with precision of 67% (50% for guano). We examine misclassifications and outline areas for improvement, and discuss how these methods can be directly applied to Abbott's booby surveys, as well as easily transferred to other species.

6.2 Introduction

Structure from Motion (SfM) is a process which enables a 3D model of a scene or object to be constructed from multiple offset 2D images, using principles of multi-view geometry [61]. It is a technique that has been applied extensively in a range of fields, including

augmented reality [46], archaeology [57] and the geosciences [152]. It is also frequently used for forest canopy mapping, with researchers using the final 3D model to estimate forest height, density and biomass [73]. In these applications the final 3D model is the main focus of the analysis, with the original 2D raw images only used at the model construction phase. However, for detection of small objects such as wildlife (which can often get lost or distorted in the final stitched orthomosaic image), the underlying raw images (i.e the original unprocessed JPEG images collected by the UAV) provide more viewpoints and greater opportunity for detection, particularly for elusive forest canopy species. We propose using these raw images in a multi-view detection approach will provide a solution for monitoring species in these challenging environments.

Objects detected in raw UAV images can be projected into their corresponding real world 3D coordinates using camera parameters estimated during the SfM process. This provides a means of matching multi-view detections of the same object into the same point in space. A key benefit is that all raw image information can be used to guide detection, rather than using the final stitched orthomosaic alone. In the literature we found only a small number of papers employing a similar technique. This includes for sewer inlet detection in urban environments [107], detection of fruit in trees [47] and identifying and classifying objects in x-ray luggage scans [138]. More recently a similar technique was applied to wildlife detection, focusing on counting cattle in open grassland [132]. However in that study the terrain was relatively flat and open, there were fewer issues with occlusion of wildlife from different viewpoints, and the effect of stitching and errors in image projection are likely to be less significant than the forest canopy surface. As yet, the methods have not been tested extensively, and to the best of our knowledge have not been applied to surveying wildlife in forest canopy.

In this Chapter we will outline our multi-view detection method, following a similar pipeline proposed by Moy de Vitry et al. [107] for sewer inlet detection. We first take our Faster R-CNN detections (the results of Chapter 5) and perform a 2D-to-3D projection to get their corresponding real-world coordinates. We then perform clustering using the DBSCAN algorithm, to group multi-view detections. Finally, we perform a classification stage, where we filter false positive detections. While in this application we focus on detecting seabirds nesting in forest canopy, we highlight that the methods are general and could easily transfer to object detection tasks in any imagery processed using SfM.

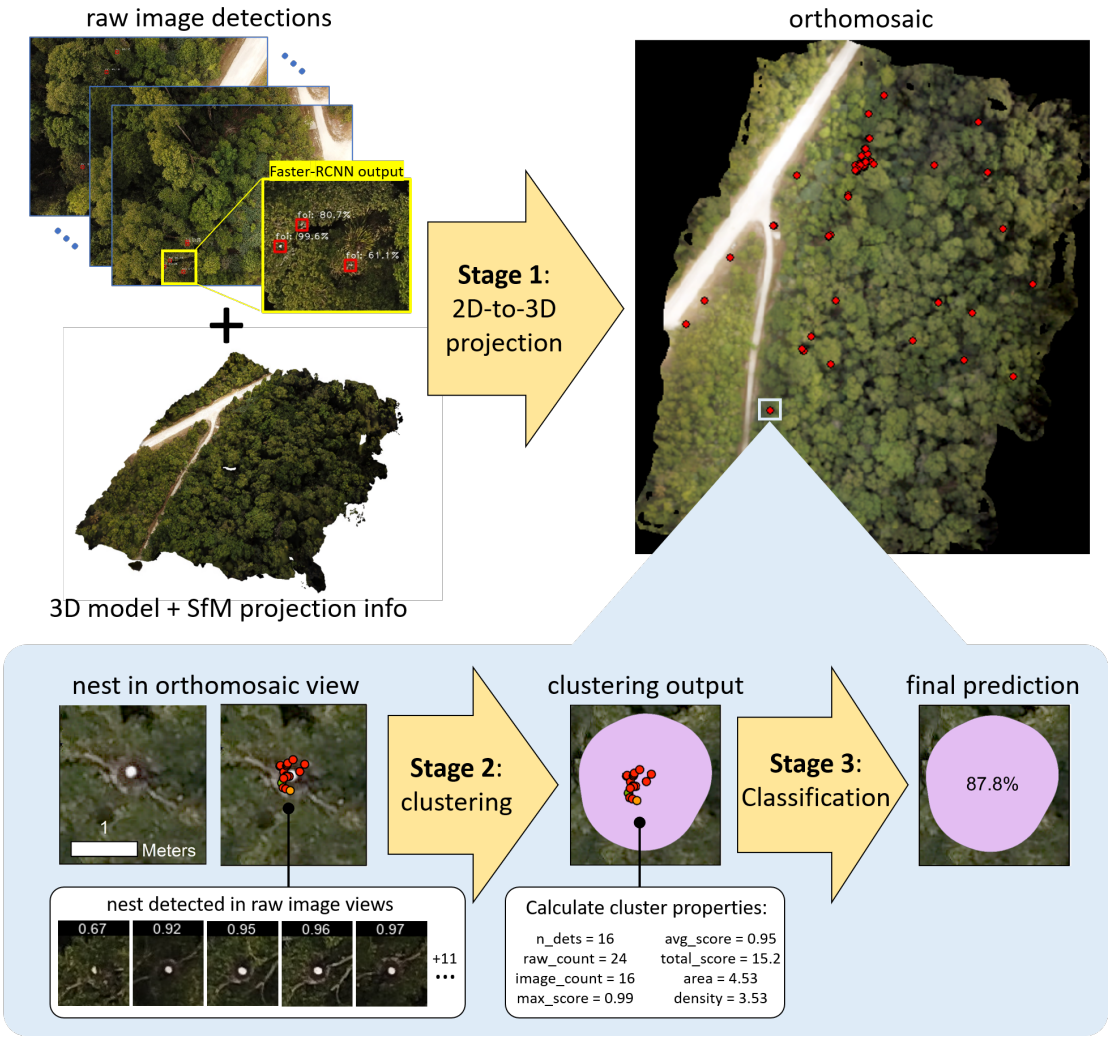

FIGURE 6.1: The multi-view detection pipeline. In Stage 1 Faster R-CNN detections are projected from image pixel coordinates to real world 3D coordinates, allowing multi-view detections of the same object to be mapped together on the orthomosaic. In Stage 2 we cluster these multi-view detections using the DSCAN algorithm, and calculate a range of cluster properties (e.g area, density and average confidence). These clusters are compared to ground truth data to assign true positive (TP), false positive (FP) and false negative (FN) labels. Finally in Stage 3 we train a classifier to predict TP clusters, to filter out excess FPs and give the final prediction result.

6.3 Methodology

6.3.1 2D-to-3D Projection

In order to project the objects detected in the raw 2D UAV images onto the 3D orthomosaic, we first establish our 2D-to-3D projection system. Given that UAV images are collected in motion, every raw UAV image $\{I_1, I_2, .., I_n\}$ is taken from a different viewpoint and camera pose. The mapping between image pixels and their 3D world coordinate is estimated during the SfM process, which predicts and outputs different

camera parameters for each image I_t. This includes the camera's intrinsic matrix $\boldsymbol{K}$, as well as a rotation matrix $\boldsymbol{R}_t$ and translation vector $\boldsymbol{t}_t$ for each unique image.

In any given raw image I_t, with a Faster R-CNN detection centered at pixel coordinate (u, v), our target is to find its location in the 3D world coordinate system $(X, Y, Z)^T$. This can be calculated using pinhole camera model transformations [61]. The equation to transform a world point $(X, Y, Z)^T$ into a camera-relative point $(x, y, z)^T$ is given by the equations:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{K} \left(\boldsymbol{R}_t \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{t}_t \right) \tag{6.1a}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix} \tag{6.1b}$$

To invert this operation, and retrieve the 3D world coordinates from a pixel coordinate (u, v) in image I_t, we solve Equations 6.1 as follows:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \boldsymbol{R}_t^{-1} \left(z\boldsymbol{K}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} - \boldsymbol{t}_t \right) \tag{6.2}$$

This is equivalent to projecting a ray from the camera projection center to the 3D model, and finding the point at which the ray intersects with the surface mesh of the 3D model.

6.3.2 Data Pre-processing

In Equation 6.2, we know $\boldsymbol{K}$, $\boldsymbol{R}_t$ and $\boldsymbol{t}_t$ as they are outputs from the SfM process. The only remaining unknown is z; the distance between the camera and the 3D surface mesh. To find z we construct the surface mesh model in Agisoft (V 1.5.2, Agisoft LLC, St. Petersburg, Russia). We select the option to reuse depth maps constructed in the original SfM processing (detailed in Chapter 5 Section 5.3.2), with high quality and aggressive depth filtering. Then, for every camera image I_t we export the corresponding Z_t; an image showing the distance to the surface mesh from every pixel in I_t. We present an example of one of these pairings in Figure 6.2. Extracting the z value at pixel coordinates (u, v) gives us all values needed to solve Equation 6.2.

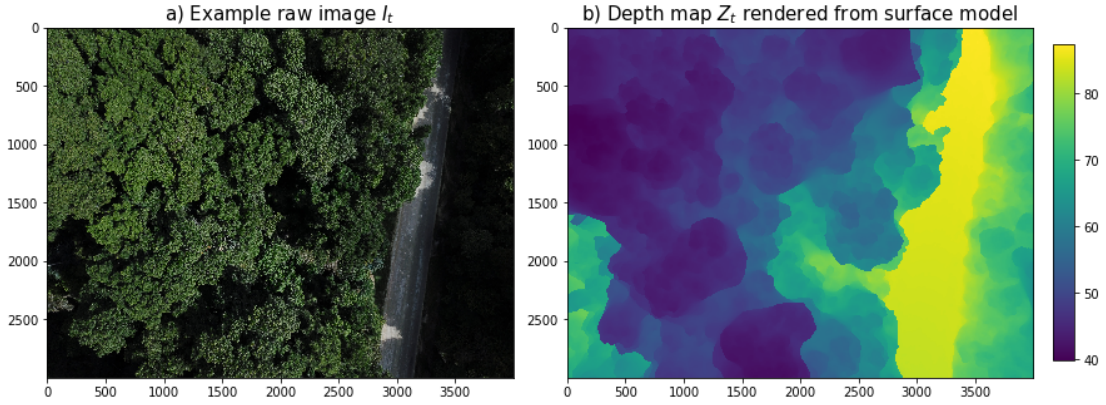

FIGURE 6.2: (a) An example raw image and (b) its corresponding depth map, formed by rendering a surface mesh and calculating the real world distance from the camera's projection centre to every point on the mesh.

6.3.3 Network Architecture

The multi-view detection method consists of three main stages (outlined in Figure 6.1). Firstly, the Faster R-CNN detections from the raw UAV images (the results of Chapter 5) are projected from 2D raw images into 3D real world coordinates. Detections corresponding to the same object form clusters on the orthomosaic. In stage two we cluster these multi-view detections using the DBSCAN algorithm, and compare the proposed clusters to ground truth data to assess whether they are *True Positives* (TPs), *False Positives* (FPs) or *False Negatives* (FNs). Finally, for each cluster proposed by DBSCAN, a range of attributes are calculated. This includes the number of detections in the cluster, the density of detections, and the average detection confidence score. In stage three these attributes are used to classify clusters into TPs and FPs, with the aim of filtering FPs and giving the final result. In the following sections we outline each of these three stages in full.

6.3.3.1 Stage 1: Projection

The first stage is to project the Faster R-CNN detections generated in Chapter 5 onto the 3D orthomosaic model. For each 2D UAV image I_t we extract the center pixel for each Faster R-CNN bounding box prediction (u, v). Using the depth map Z_t for the image I_t, we determine the z needed to solve equation 6.2, and project the 2D point (u, v) on to the 3D model. We initially choose to project all Faster R-CNN detections with a confidence value greater than 0.1. In our clustering stage we experiment with different choices of detection confidence threshold (d_T) as a hyperparameter. Once detections from all 2D UAV images have been projected, those corresponding to the same object form clusters in the real world coordinate system.

6.3.3.2 Stage 2: Clustering

In Stage 2 we cluster projected detections using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [34] implemented in the python scikit-learn library [114]. The DBSCAN algorithm is described in detail in Chapter 2 Section 2.12.1. We use the Euclidean metric calculate the distance between points, and treat min samples N and ε as hyperparameters (detailed in Section 6.3.4). DBSCAN assigns points into clusters based on the density parameters, with any unassigned points classified as noise (unless $N = 1$, in which case noise points form a cluster of size one).

Once points have been assigned to clusters, we form a detection outline by taking the *convex hull*: the smallest convex polygon that encloses all points in the cluster. Since only the centre of the Faster R-CNN bounding box detections are projected, we add a 1m buffer to the convex hull to replicate the bounding box size. For guano, which does not have a clearly defined central point, this approach allows us to estimate the area of guano coverage. This can be useful for determining true Abbott's booby breeding sites, where the spread of guano is likely to be larger compared to non-breeding sites.

Before our final classifier stage, we must assign either a True Positive (TP) or False Positive (FP) label to our DBSCAN clusters by comparing them to the ground truth data. These TP/FP labels will be used to train supervised classification algorithms to filter FPs following the clustering stage. We form the cluster ground truth used for this assessment by projecting the raw image annotations (described in detail in Chapter 5 Section 5.3.2.2) to the 3D model, using the equations outlined in Section 6.3.1. We use the unique object tags assigned during manual analysis to group these multi-view detections into single clusters. We again form a convex hull around these points and add a 1m buffer. Proposed DBSCAN clusters are labelled as TP if their convex hull intersects with the convex hull of a ground truth cluster with an intersection over union (IoU) greater than 0.1. If the IoU is less than 0.1 then the proposed cluster is labelled as FP.

6.3.3.3 Stage 3: Classification

For each cluster identified by DBSCAN we calculate a range of features. These are used to characterise and classify FOI and guano detections into TPs and FPs, with the aim of filtering out FP clusters for the final result. The full list of features calculated is presented in Table 6.1, and includes the maximum, average and total confidence scores of all points in the cluster, cluster area, and the number of detections per cluster.

TABLE 6.1: Summary of cluster features used for classification stage.

Feature Name	**Description**
n_dets	*Number of detections in the cluster*
raw_count	*Number of raw images which map to the cluster center*
image_count	*Number of raw images where the detections came from*
max_score	*Maximum confidence score of any point in the cluster*
avg_score	*Mean of the cluster confidence scores*
total_score	*Sum of the cluster confidence scores*
area	*Area of the cluster convex hull*
density	*Density of points in the cluster (n_dets / area)*

To classify clusters we compare three different models (implemented in scikit-learn [114], with general model overviews presented in Chapter 2): a logistic regression model (LR; detailed in Section 2.11.2), a support vector classifier (SVC; Section 2.11.1), and a multilayer perceptron (MLP; Section 2.7.3) with 100 hidden layers and ReLU activations. For each model we perform a hyperparameter search to find optimal values for other settings, which are summarised in Section 6.3.4. All three of our chosen classifiers output a classification probability score, which we use as the final prediction confidence.

For our classifier train/validation/test splits we use the same folds outlined in Chapter 5 Section 5.3.4 for training the Faster R-CNN network, where the 32 plots were divided into four groups of eight (refer to Appendix B.1 for the complete list of plots per fold). We balance TP and FP classes in the training set by downsampling the more common class to match the number of examples of the rarer class (with FPs generally being more prevalent). We found experimentally that this performed equivalently to upsampling the rarer class, without introducing extra computation time.

6.3.4 Hyperparameters

In our hyperparameter experiments we test the best choice of DBSCAN parameters ε (ranging from 1 to 10 metres using a step size of 1) and N (1, 2 and 3). These ranges were informed by initial coarser scale experiments, as well as visual inspection of the data (estimating realistic sizes of FOI and guano sites, and using knowledge of the number of views per object from our analysis in Chapter 5). We also test different detection thresholds (d_T) for the projected Faster R-CNN detections (ranging from 0.1 to 0.9 using a step size of 0.1). The choice of d_T determines the number of false positive Faster R-CNN detections which are projected, which influences the amount of noise at the clustering stage.

For each of the three classifiers we perform a grid search for a range of different parameter settings, detailed in Table 6.2. For all combinations we also test different data scaling

and transform methods: standard scaling, min-max scaling, normalization, power transformer and quantile transformer. Since the output of the clustering stage influences the success of the classification stage (i.e due to the numbers of FP/TP examples, and the cluster properties) this grid search is performed in combination with our clustering ε, N and d_T experiments.

TABLE 6.2: Hyperparameter combinations tested for each of the three classifiers: logistic regression (LR), support vector classifier (SVC) and a multi-layer perceptron (MLP). Methods implemented in scikit learn [114].

classifier	**parameter**	**grid search values**
LR	penalty	none, l2
	C	0.01, 0.1, 1.0
SVC	kernel	linear, poly, rbf, sigmoid
	C	0.01, 0.1, 1.0
MLP	alpha	0.001, 0.01, 1, 5
	solver	sgd, adam
	learning rate	constant, adaptive

6.3.5 Hardware and Frameworks

Model training is performed on a PC workstation equipped with Intel i7-8700 CPU @ 3.20GHz, 32GB of RAM and NVIDIA Titan Xp graphics card with 12GB of GPU memory. All clustering and classification was performed using scikit-learn 1.0.2. Surface meshes were calculated in Agisoft V 1.5.2, and took approximately 20 minutes per plot to generate using the hardware setup.

6.3.6 Evaluation Metrics

We compare the overall performance of our models using the mean average precision (mAP) score; the area under the precision-recall curve, averaged across the four data folds. When reporting per fold or per class results we use the Average Precision (AP). We also use recall, precision and F_β-scores compare results at specific confidence thresholds.

6.4 Results

6.4.1 Hyperparameter Results

We summarise the results of our hyperparameter search for each class (FOI and guano) and each method (LR, SVC and MLP) in Table 6.3. We find that the mAP scores are

TABLE 6.3: Clustering and classification results for FOI and guano predictions. We present the best mAP scores (mean and standard deviation averaged across four data folds) for the three classifiers, as well as the optimal hyper-parameters used to achieve the results (italics).

class	clf	val mAP	test mAP	N	ε	d_T	trans	C	pen	kernel	alpha	solver	lr
foi	LR	0.59±0.08	0.60±0.07	*1.0*	*5.0*	*0.5*	*quantile*	*1.0*	*l2*	-	-	-	-
	SVC	0.58±0.09	0.61±0.08	*1.0*	*5.0*	*0.5*	*quantile*	*0.01*	-	*poly*	-	-	-
	MLP	0.59±0.08	0.61±0.07	*1.0*	*5.0*	*0.5*	*quantile*	-	-	-	*0.01*	*adam*	*adapt*
guano	LR	0.57±0.11	0.57±0.02	*1.0*	*8.0*	*0.4*	*quantile*	*1.0*	*l2*	-	-	-	-
	SVC	0.57±0.11	0.57±0.02	*1.0*	*8.0*	*0.4*	*quantile*	*0.1*	-	*linear*	-	-	-
	MLP	0.57±0.10	0.58±0.02	*1.0*	*8.0*	*0.4*	*quantile*	-	-	-	*1.0*	*adam*	*adapt*

very similar for the three methods, with an mAP of 0.60-0.61 for FOI's and 0.57-0.58 for guano. For our hyper-parameter tuning experiments, we find that the best data transform for all methods is the quantile transformer. For the LR classifier the optimal test mAP is achieved for $C = 1.0$ and $penalty = l2$, which is the same for both FOI and guano classes. For SVC we find $C = 0.01$ and a polynomial kernel give the best results for FOI, while $C = 0.1$ and a linear kernel give the best score for guano. Finally for our MLP classifier the adam optimizer and an adaptive learning rate are the best choice for both classes, while $\alpha = 0.01$ is optimal for FOI and $\alpha = 1.0$ is optimal for guano. For our DBSCAN clustering parameters we see that a minimum sample of $N = 1$ gives the best result for all classes and classifiers. Between classes the choice for ε is $\varepsilon = 5$ for FOI and $\varepsilon = 8$ for guano for each of the three methods. Having a larger ε follows logically as guano is less clearly defined and has a larger spread than FOI's. The choice of the confidence threshold on Faster R-CNN predictions is also consistent between the classifiers, with $d_T = 0.5$ for FOI and $d_T = 0.4$ for guano.

We examine the relationship between ε and d_T further in Figure 6.3, using $N = 1$ and the optimal classifier hyper-parameters outlined in Table 6.3. Again, the results are very consistent between the three different classifiers. For FOI we find the worst mAP scores at $d_T = 0.1$ and $\varepsilon = 10$. With these settings there are a significant number of false positive Faster R-CNN detections which are all clustered together due to the high ε value. This results in single, large clusters forming on the orthomosaic. Results improve with increasing d_T thresholds, as the boundaries between clusters are clearer when erroneous false positive Faster R-CNN detections are removed. We see a similar pattern for guano clusters, although the optimal values fall at higher ε values due to guano clusters being generally more spread out than FOIs. We see at higher d_T thresholds (e.g > 0.9) the results for guano are significantly worse. This is due to the fact that in general Faster R-CNN predicts guano less confidently than FOI, and so many point detections are excluded completely at this cut off. Since we have agreement between all three models, we can be confident in our selection of ε and d_T (FOI: $\varepsilon = 5$ & $d_T = 0.5$, Guano: $\varepsilon = 8$ & $d_T = 0.4$). When selecting a final choice for the classifier we see very marginal differences, so for the remaining results we select the LR model for its

simplicity in comparison to SVC and MLP. In addition the same LR hyperparameter selection provides optimal results for both FOI and guano, making the model completely consistent for both classes.

6.4.2 Detection Results

We present the final precision-recall curves, using our chosen LR classifier, in Figure 6.4. We see that there is some variation in performance across the different test folds, with greater variation for FOI (ranging from $AP = 0.51$ for fold 3 to $AP = 0.68$ for fold 0) than guano predictions (ranging from $AP = 0.55$ to $AP = 0.60$). In terms of F1-score accuracy, we see a peak of approximately 0.7 for FOI and 0.65 for guano, with an average accuracy of approximately 0.6 for both classes. We note that recall does not exceed the 80% mark for either class, meaning that at least 20% nests are missed by the method, irrespective of the selected operating point.

In Figure 6.5a we plot the F1-score against different threshold values, to determine the point at which maximum overall accuracy is achieved. In our application missing nest sites could be viewed as more costly than having false positives, as finding missed sites would require an exhaustive manual search, whereas FPs could be manually filtered out comparatively quickly. We therefore also plot the F2-score results Figure 6.5b, where recall is weighted with twice the importance of precision. Since the optimal F1-score thresholds give higher precision than recall, we choose to use the F2-score thresholds (0.55 for FOI, 0.44 for guano) as thresholds to present the final results. These give us average recall values of 62% for FOI and 66% for guano, with precision scores of 67% (FOI) and 50% (guano).

6.4.3 Final Detection Output

In Table 6.4 we present a summary of the final detection outputs using the F2-score informed confidence thresholds (FOI : 0.55, guano : 0.44). At the chosen operating point 62±3% of FOI are successfully detected by the method, and 66±5% of guano. Correspondingly we have precision scores of 67±7% for FOI and a slightly poorer result of 50±8% for guano. In terms of practical application, for this dataset we see the method would return a total of 446 FOI detections, 150 of which would be false positives. These 150 FPs remain after the classification stage successfully filtered 1220 (an average 89%) of FP detections made at the clustering stage (classed as *True Negatives* - TN). We see similar success for guano detections, with approximately 80% of FPs filtered out by the classification stage. In terms of false negatives (FNs), 185 of the total 481 FOI ground truth clusters were not detected by the method. We see that a large fraction (on average

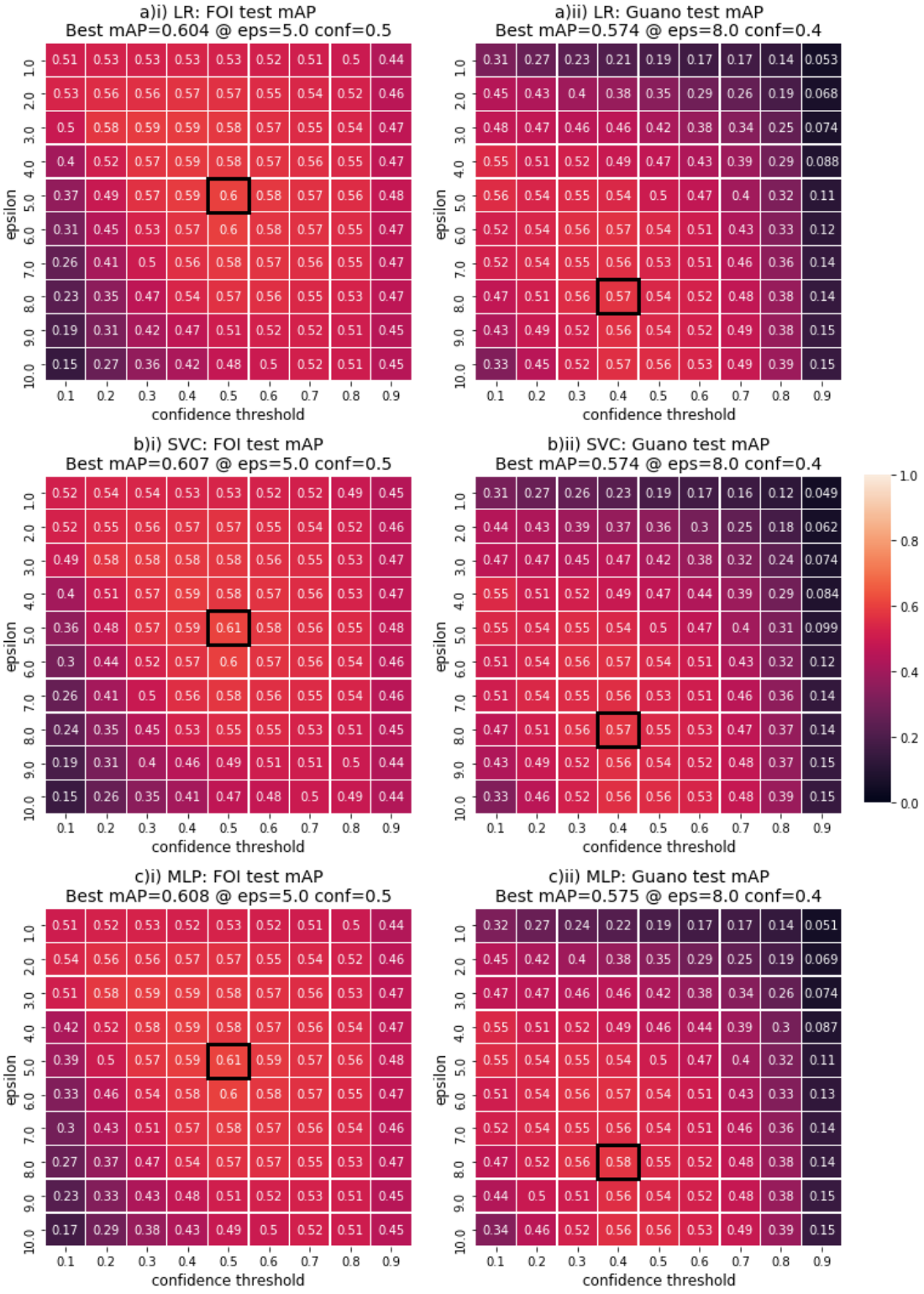

FIGURE 6.3: Sensitivity to DBSCAN clustering parameters for different choices of ε and Faster R-CNN detection threshold d_T (with minimum samples $N = 1$ and optimal classifier hyper-parameters). Scores are the mAP on the test set, averaged across the four data folds. Presented for both FOI and guano, using a) logistic regression (LR), b) support vector classifier (SVC) and c) multi-layer perceptron (MLP).

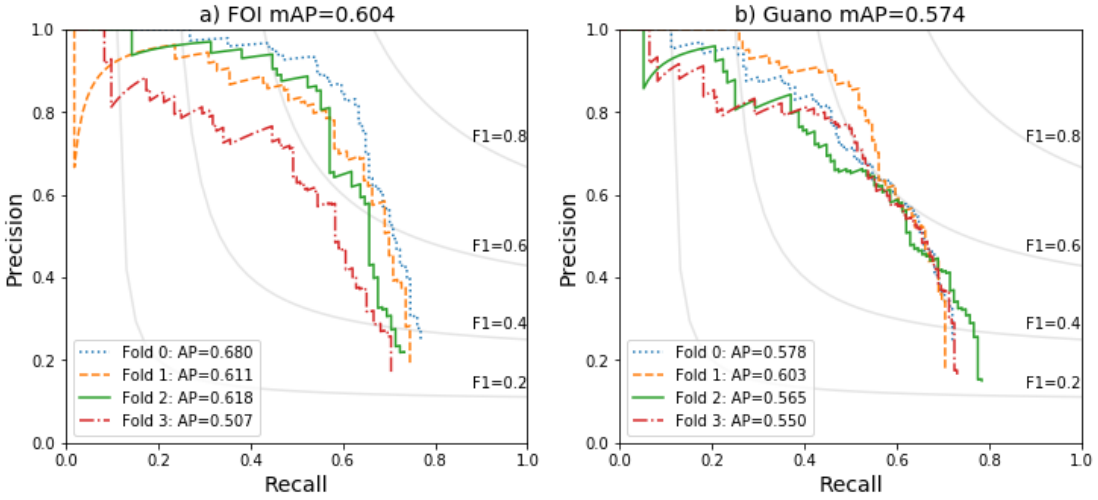

FIGURE 6.4: Precision-recall results for each of the four test folds, using the best scoring logistic regression (LR) classifier. Presented for a) FOI's, with an $mAP = 0.604$ and b) guano, with an $mAP = 0.574$.

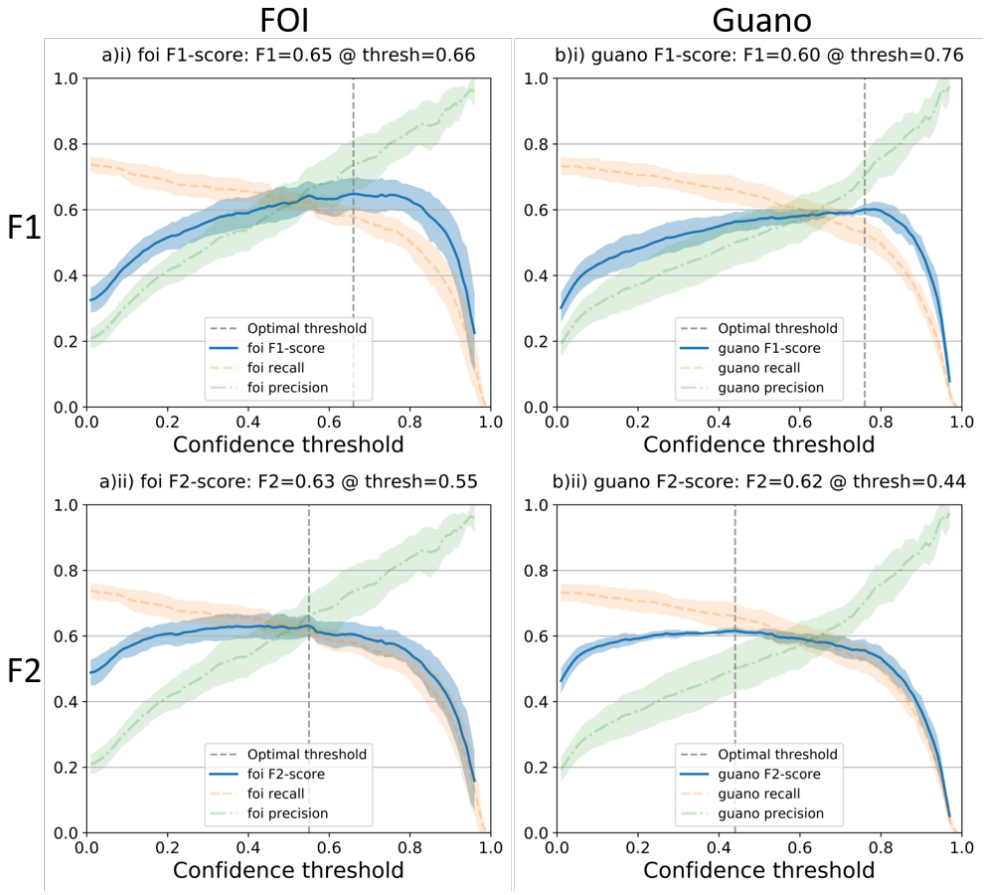

FIGURE 6.5: Fold-averaged F-scores plotted against confidence threshold for a) FOI and b) guano detections. We investigate i) the F1-score, where precision and recall have equal weighting, ii) the F2-score, where recall has two times more weighting than precision. Grey lines show the point where peak F-score is achieved.

TABLE 6.4: Per fold results for the optimal confidence thresholds (FOI : 0.55 and Guano : 0.44). We report the average precision (AP) recall (rec), precision (prec), and the total number of true positives (TP), false negatives (FN), false positive (FP) for the detection results. We also report the number of true negatives (TN), which are FPs correctly filtered out at the classification stage, and the corresponding percentage of correctly filtered FPs (FP_filt). For FNs we report the percentage which were missed at the clustering stage (FN_clus), and the percentage which were TPs incorrectly classified at the classification stage (FN_clf).

Class	Fold	AP	Rec	Prec	TP	FN	FP	TN	FP_filt	FN_clus	FN_clf
FOI	0	0.68	0.64	0.75	86	48	28	279	91%	65%	35%
	1	0.61	0.64	0.69	70	40	32	309	91%	70%	30%
	2	0.62	0.60	0.65	63	42	34	240	88%	67%	33%
	3	0.51	0.58	0.58	77	55	56	392	88%	71%	29%
Total		-	-	-	**296**	**185**	**150**	**1220**	-	-	-
Mean		**0.60**	**0.62**	**0.67**	**-**	**-**	**-**	**-**	**89%**	**68%**	**32%**
Std		0.07	0.03	0.07	-	-	-	-	1.9%	2.9%	2.9%
Guano	0	0.58	0.60	0.60	107	72	70	321	82%	69%	31%
	1	0.60	0.66	0.52	92	47	85	359	81%	87%	13%
	2	0.57	0.71	0.41	82	34	117	397	77%	74%	26%
	3	0.55	0.67	0.47	115	56	131	501	79%	82%	18%
Total		-	-	-	**396**	**209**	**403**	**1578**	-	-	-
Mean		**0.57**	**0.66**	**0.50**	**-**	**-**	**-**	**-**	**80%**	**78%**	**22%**
Std		0.02	0.05	0.08	-	-	-	-	2.1%	8.1%	8.1%

68%) of these were missed at the clustering stage, while fewer (on average 32%) were TPs which were incorrectly classed as FPs at the classification stage. We see a similar pattern for guano detections (78% of FNs missed at clustering, 22% misclassified).

In Appendix Table C.1 we summarise the results from Table 6.4 for each individual plot in the test folds. We find that the success varies, with the best performance achieved for plot N02 (across class mAP of 0.81: FOI $AP = 0.84$, guano $AP = 0.78$) and the worst for plot N38 (across class mAP of 0.28: FOI $AP = 0.40$, guano $AP = 0.19$). In Figure 6.6 we present these as example results on the orthomosaics, using our chosen threshold values. For the best scoring plot (N02; Figure 6.6a) we are able to successfully detect seven out of eight FOI, with one FN and one FP. When visually assessing the raw image annotations for the FN cluster, we find that it is very challenging to discern. In addition, a cluster of guano was correctly identified next to it, so it may be that used in combination an expert analyst would be able to identify the area as a breeding site. When plotting the raw image detections for the FP, we notice that this in fact appears to be a nest site which was missed in the manual annotation. We also note that in both cases the FOIs are not visible in the stitched orthomosaic views. For the worst performing plot (N38; Figure 6.6b) we find that most errors were introduced by missed guano. Of 13 guano ground truth clusters, only three were successfully detected, and a further four were FPs. Inspecting FN guano patches we find that in most examples these are small patches of staining which is challenging to detect in the raw images. Comparatively, the results are better for FOI, with three of seven correctly detected,

and only two FPs. We see one of the FPs is caused by a white branch of similar size to an adult Abbott's booby.

6.5 Discussion

6.5.1 Clustering Stage Assessment

In our final results we find that the majority of FNs (68% of FOI and 78% of guano annotations) are missed at the clustering stage. Either these were not detected in any view by Faster R-CNN (or detected with low confidence), or they were not successfully clustered and matched to ground truth at the DBSCAN stage. After clustering, these missed detections cannot be retrieved at the classification stage (which can only classify and filter out FPs). We summarise the overall success of the clustering stage in Table 6.5 and find that on average 26% of objects are missed at this point (for both FOI and guano classes). This gives us baseline recall values of 0.74 before going through to classification. We also note the large number of false positive clusters (1370 for FOI and 1981 for guano), equating to very low precision scores of approximately 0.2 for both classes. The role of the classification stage is to remove these FPs while retaining as many TPs as possible.

Considering the FNs missed at the clustering stage in more detail, in our initial Faster R-CNN assessment (Chapter 5 Section 5.5.3) we estimated that approximately 12% of FOI were not detected in any view at the Faster R-CNN stage (when using a detection threshold of 0.1). This represented our best case recall score for our multi-view approach. In our visual assessment of those missed objects, we found that the majority were visible in only a single raw image, were in challenging background, or represented rarer examples such as empty nests (refer back to Chapter 5 Figure 5.9 for image examples). In Figure 6.7a we repeat this assessment using our final results. We find again that most missed objects were only identified in small number of view points in the manual analysis. For example of the 109 nest sites which were not detected, 72 were identified in five or less views. Of the nests which were missed in more than five raw images, they again represent challenging cases, which were likely only detected by Faster R-CNN with a low confidence score (below the $d_T = 0.5$ threshold we found gave overall best results). We also find that empty nests present challenges, with the method missing 8 out of a total of 17 in the dataset. A large proportion of second adult birds (2ndA: defined as birds found near another nesting bird) were also missed (8 out of a total of 14 in the dataset). However, we found that in all cases the actual nest site was correctly detected (Figure 6.7b). Since the ultimate goal of the surveys is to identify Abbott's

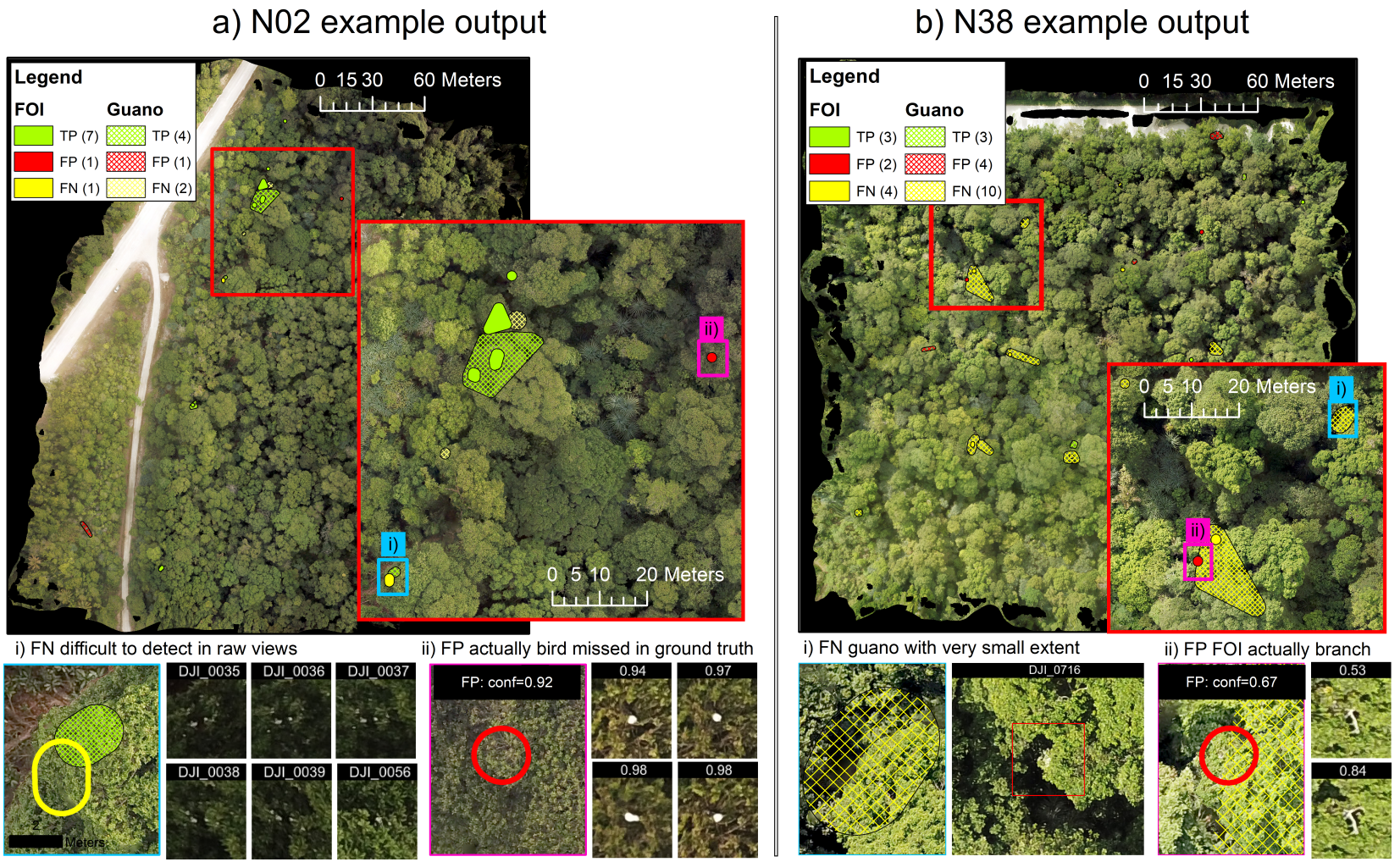

FIGURE 6.6: Examples of the best and worst scoring plots in terms of mAP. We use confidence thresholds of 0.55 for FOI and 0.44 for guano, to get the final true positive (TP), false positive (FP) and false negative (FN) results. **a)** For N02 we successfully detect 7 FOI, with 1 FN which is a challenging case (i) and one FP which was incorrectly missed in the ground truth (ii). **b)** For N38 guano FNs lower the overall score, with 10 out of 13 missed. We note many of these guano patches cover a very small extent, making them difficult to detect (i). FOI FNs are also challenging and white branches can cause FP detections (ii).

TABLE 6.5: Summary of the results after the clustering stage, using the optimal parameters (FOI: $\varepsilon = 5, N = 1, d_T = 0.5$ / Guano: $\varepsilon = 8, N = 1, d_T = 0.4$). We summarise the number of true positive (TP), false negative (FN) and false positive (FP) clusters, as assessed against the ground truth. Final recall (rec) and precision (prec) scores are approximately 0.7 and 0.2 respectively. Of the total objects, approximately 26% are assessed as FN at the clustering stage.

Class	Fold	Rec	Prec	TP	FN	FP	total objects	% FN
FOI	0	0.77	0.25	103	31	307	134	23.1%
	1	0.75	0.19	82	28	341	110	25.5%
	2	0.73	0.22	77	28	274	105	26.7%
	3	0.70	0.17	93	39	448	132	29.5%
Total		**-**	**-**	**355**	**126**	**1370**	**481**	
Mean		**0.74**	**0.21**	**-**	**-**	**-**	**-**	**26.2%**
Std		0.03	0.03	-	-	-	-	2.7%
Guano	0	0.72	0.25	129	50	391	179	27.9%
	1	0.71	0.18	98	41	444	139	29.5%
	2	0.78	0.15	91	25	514	116	21.6%
	3	0.73	0.17	125	46	632	171	26.9%
Total		**-**	**-**	**443**	**162**	**1981**	**605**	**-**
Mean		**0.74**	**0.19**	**-**	**-**	**-**	**-**	**26.5%**
Std		0.03	0.043	-	-	-	-	3.4%

booby breeding sites, these could still be assessed as successful detections of a nest site, despite not locating the second adult.

The problem with detecting rare classes is common in computer vision tasks, as there are fewer examples which can be used in supervised training. In this example empty nests do not have the bright white blob of an adult Abbott's booby, which is present in the nest FOIs which make up the majority of examples in the dataset. Collecting more images of empty nests would likely improve these results, and with enough data could allow us to separate them into a class of their own (rather than grouping into one FOI class). This could help to guide the training, and avoid the network becoming biased towards FOIs which contain an adult bird. Empty nests are also easy to confuse with branches in trees, and determining if it is truly a nest requires expert knowledge. For example other factors such as the proximity to guano staining, or location in the tree, may guide an expert in their interpretation. For this dataset we only have a single observer's annotations, and so evaluating the uncertainty in the ground truth (as we did with our albatross dataset in Chapter 4) is not possible at this stage. However, with this information we could draw more formal conclusions of how the automated method compares against subjective human analysis.

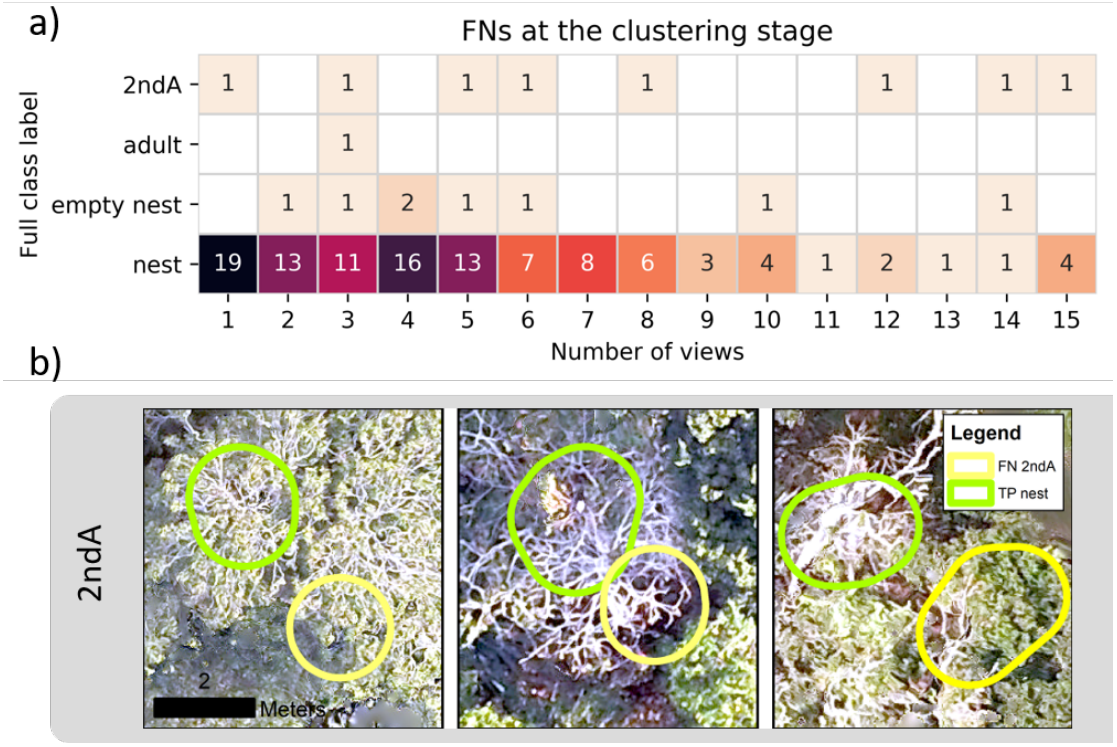

FIGURE 6.7: Heatmap with cells showing the number of FNs according to their full class label and the total number of raw image views. We see that many missed objects are only visible in a low number of views, and classes like second adults (2ndA), and empty nests present challenges. b) Examples of 2ndA clusters which were missed but the corresponding nest was successfully detected.

6.5.2 Classification Stage Assessment

In our results we find that our LR classifier successfully filters high percentages of false positives (on average 89% of FOI and 80% of guano; Table 6.4) using our final confidence thresholds. Investigating the performance of the classifier alone (ignoring FNs missed at the clustering stage), we achieve mAP scores of 0.81 and 0.78 for FOI's and guano respectively (Figure 6.8a). Measured in terms of the Receiver Operating Characteristic (ROC) curve, which takes into account the true negative rate (i.e correctly filtered FPs), we have mean Area Under the Curve (AUC) scores of 0.92 for FOI and 0.93 for guano (Figure 6.8b). This shows that the classifier is very effective at filtering out excess FP clusters. Despite this, there is not perfect separability between TP and FP clusters, with a trade off between recall and precision depending on the final threshold value. We found in our classifier comparison experiments that the LR, SVC and MLP methods all gave very similar results, suggesting that alternative classifiers are unlikely to offer significant gains. This is likely because we have a relatively simple feature space (with only eight variables), and that TP and FP clusters cannot be categorically separated based on these features alone.

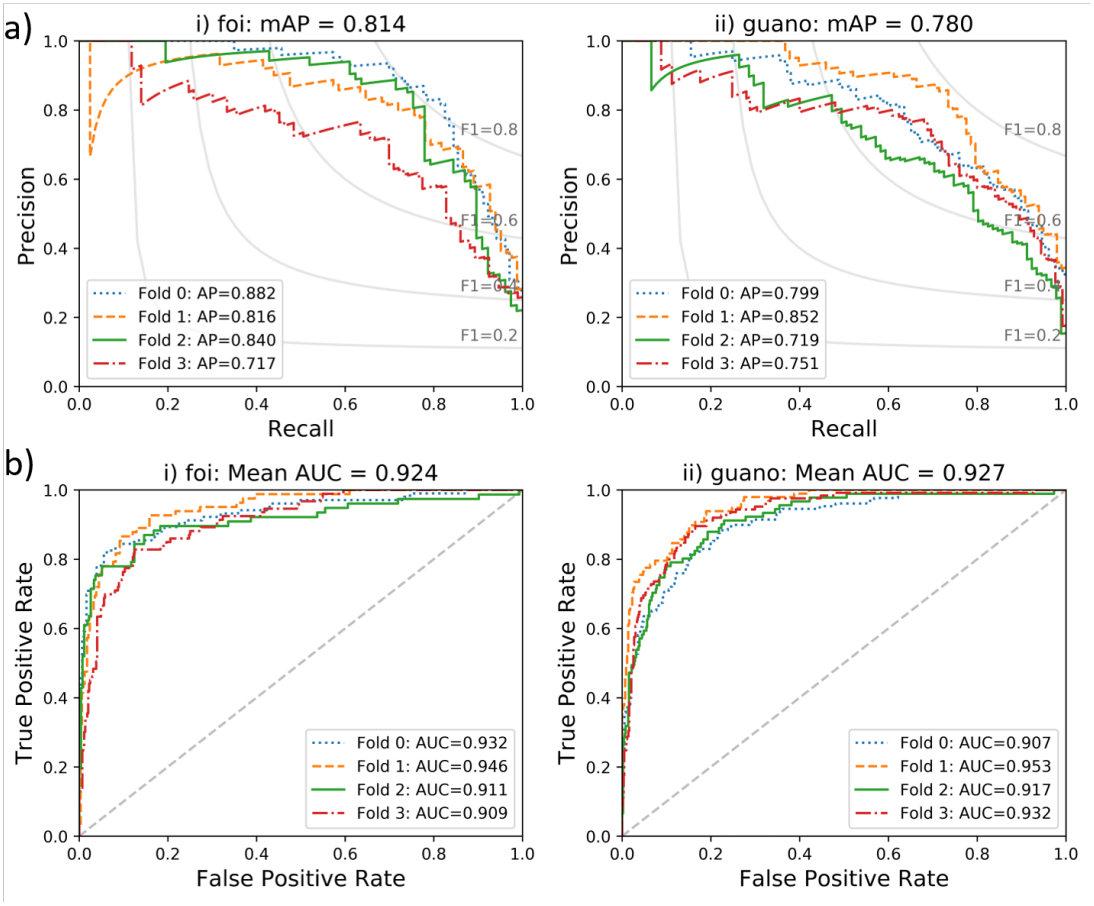

FIGURE 6.8: Results of the logistic regression (LR) classifier (ignoring FNs missed at the clustering stage). We present a) precision-recall curves and b) the ROC curves for each test fold.

In Figure 6.9 we plot the probability density functions (PDFs) of the eight variables used for classification, and in Table 6.6 assess their relative importance based on both the F-score and mutual information criteria. In our feature ranking we find under both measures the top three features are image_count (the number of raw images where a detection was made), n_dets (the number of detections in the cluster - almost synonymous with image_count except for cases when there are two detections in the same raw image), and the total_score (the sum of the confidence scores for every detection in the cluster). We can see from the PDFs that the majority of FPs fall in the lower end of these scales, with clusters made up of only a few low confidence detections. These are likely to be false positives which were detected by Faster R-CNN in a single raw image, but which were not repeatedly detected in other views. On the other hand, for TPs there is a trend towards larger numbers of detections in each cluster, with higher total, average and maximum confidence scores. However we also note that in all cases there is some overlap in the TP/FP distributions, with some TPs also having low numbers of detections and total scores. This makes drawing a definitive boundary between the classes challenging.

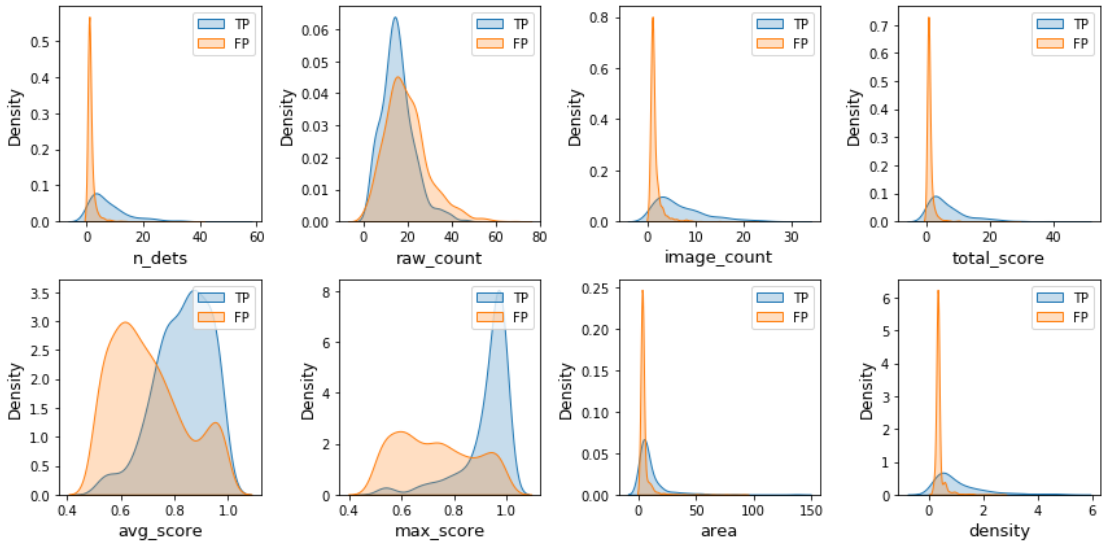

FIGURE 6.9: Probability Density Functions (PDFs) for each of the variables used in the classification stage, showing the distribution of True Positives (TP) and False Positives (FP) from our clustering stage assessment. We present the results for the FOI class only.

TABLE 6.6: Feature ranking of FOI variables according to a) the F-score and b) the mutual information score.

a) F-score ranking

Feature	**F-scores**
image_count	943.5
total_score	779.0
n_dets	702.2
density	634.0
max_score	568.4
avg_score	270.1
area	142.1
raw_count	52.4

b) Mutual info ranking

Feature	**Mutual info**
total_score	0.224
image_count	0.207
n_dets	0.194
density	0.191
max_score	0.161
area	0.156
avg_score	0.106
raw_count	0.029

To improve the classification stage we could consider adding extra features, for instance elevation. The elevation could be used to filter out clusters which are outside the suitable habitat range for Abbott's boobies, which tend to nest at the very top of the canopy. For this particular dataset there were ground truthing issues during data collection which made the elevation models very variable between plots, so it was not factored into this analysis. Further habitat variables could also be considered, such as the clusters' bearing in the tree, since Abbott's boobies tend to nest on the north west side to shelter from the prevailing wind. This would require some further processing to automatically determine the relative location of clusters within individual trees, however the addition of such habitat variables is a topic which could be discussed with expert ecologists.

It is also important to note that the final aim of the surveys is to count active breeding sites, rather than individual FOI and guano sites as we have in this analysis. Breeding

site identification requires significant expertise, for example taking into consideration the presence and extent of guano build up near FOIs, the structure of a nest in case it is left over from a previous breeding season, and possibly detecting chicks or eggs within the nest. Combinations such as a low confidence nest detection along with a high confidence guano detection could lead to a positive assessment as a breeding site, compared to say a very confident detection of an adult bird which is not near any nest or guano. By formalising these measures we could potentially train a second stage classifier, to combine the detected FOIs and guano and categorise them as either breeding or non-breeding sites. When viewed in this light misclassifications associated with the individual classes may not be as important, or could be compensated for by the nearby detection of another object. This would be the case, as discussed previously with missed second adult birds, which could be compensated for by detecting the associated first adult bird.

6.5.3 Misclassification Analysis

We present a gallery showing our highest scoring FP detections after the final classification stage in Figure 6.10. We find that the majority of FP FOI predictions are caused by white branches which appear similar to adult birds and nests. If the same branch is detected in multiple raw image views, then the cluster is predicted to be a TP by our LR classifier. This can be a problem in plots which have a lot of bare branches, and could be affected by the time of year the survey is conducted in. For example in the dry season certain trees drop their leaves, which may lead to more FPs for surveys conducted in this period. One of our examples appears to be a tree stump which is detected next to the road, which shows the potential of using altitude to rule out detections which are not in the canopy level. In some cases branches also appear similar to empty nests, which can be visually hard to distinguish without expert knowledge.

Other FPs identified were a small number of flying Abbott's boobies (Figure 6.10b). In this multi-view method our aim is to avoid detecting flying birds in the final result, as they are not associated with a nest site. Also, since they are in motion during the UAV survey, detections of the same flying bird appear at different points in the orthomosaic. We find that the classification stage filters many of these out, as they form a cluster of only one detection, which is more associated with the FP class. However, in some cases this single detection can have very high confidence (e.g 0.96 and 0.99 for the two examples presented), so overall the cluster can be assigned a TP label with relatively high confidence (e.g of 0.57 and 0.63 respectively for our examples). For guano, we find the presence of roads in some plots introduces FP detections (Figure 6.10c). When bright road surface is viewed through sparse leaves, it can be interpreted as bright staining on the surface of the leaves by the network. Since there are relatively few examples of roads

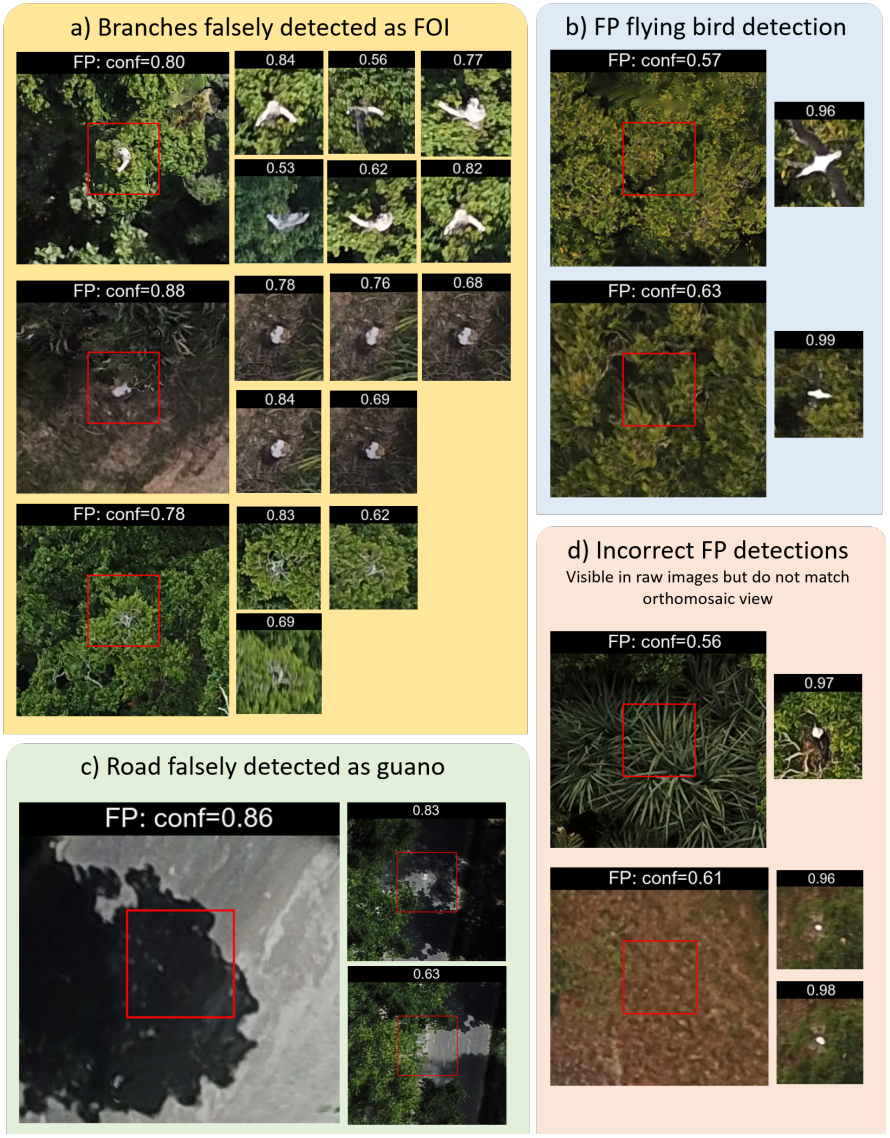

FIGURE 6.10: Examples of common false positive (FP) results. a) The most common cause for high scoring FP detections are white branches which appear similar to birds or nests. b) In some cases flying birds (which we wish to exclude from final detections) are still included. c) For guano road showing through the canopy can cause FPs. d) In a small number of cases clusters are assessed as FP although the detections in the raw images appear correct.

in each training fold, these errors would likely be reduced by the addition of more data. Furthermore, these misclassifications can be observed and manually removed relatively quickly by directly looking at the orthomosaic, without the need to assess the underlying raw images.

Finally, we find that a small number of FPs appear to contain correct detections in the raw images (Figure 6.10d). When assessing these FP clusters on the orthomosaic, we find that this can be caused by uncertainty in the 2d-to-3D projection stage. Since FOIs are relatively small objects we would expect most points to map together very

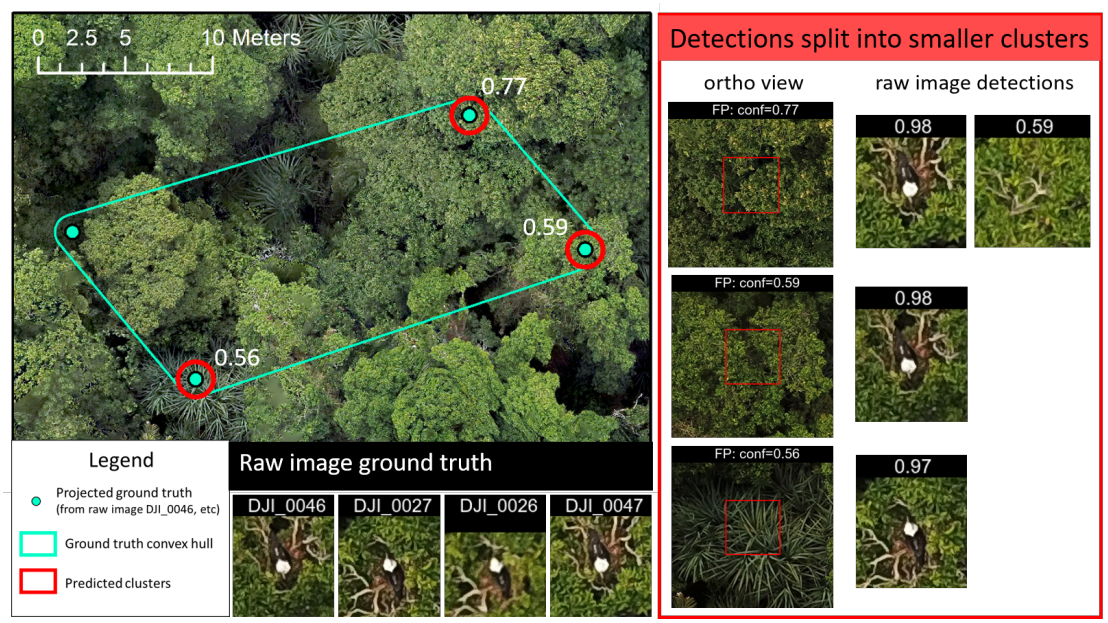

FIGURE 6.11: Example of false positives due to projection error. The ground truth points are projected to different areas of the canopy, forming a large convex hull (blue outline). While three of the four ground truth points are successfully detected, they are not merged at the clustering stage due to their large separation. They are therefore assessed as false positives (FPs).

closely. In a perfect model, where all projected points mapped together exactly to a single location, we would have a cluster covering an area of approximately $3m^2$ (a circle with radius equal to our convex hull buffer of 1m). However, in some cases projection errors mean that multi-view detections are cast over large extents, which impacts the success of our clustering stage. In Figure 6.11 we present an example of this, where the projected ground truth detections form a cluster covering nearly $300m^2$. While three of the four views were successfully detected by the network, due to their spread they were not successfully clustered using our DBSCAN parameters. Instead the three detections form individual clusters. Since none of the points have IoU greater than 0.1 with the ground truth cluster, none are assessed as TP, and the ground truth cluster itself is assessed as FN. In the final result our classifier predicts the three clusters to be true FOI detections, with confidence scores of 0.56, 0.59 and 0.77. Clearly this can introduce some errors in our analysis. In the following section we discuss the cause and extent of these projection errors in more detail.

6.5.4 Projection Error

As noted in our misclassification analysis, some of our detection errors are caused by uncertainty in the 2D-to-3D projection stage. A level of uncertainty is anticipated in the SfM process, influenced by factors during image collection (e.g the consistency of

the UAV flight height during transects), as well as the accuracy parameters selected in the processing stage (e.g choosing the number of tie points). However, we find further projection inaccuracies for Abbott's booby nest sites in certain locations. Recalling the method for projecting raw image pixel coordinates to 3D world coordinates: we calculate a ray from the camera projection centre to the 3D model, and find the point at which it intersects the 3D surface mesh model (represented by the depth map Z_t for each image I_t). However, the 3D surface mesh is modelled as a continuous smooth surface, while the forest canopy is naturally complex with gaps, discontinuities and protruding branches. When FOIs are detected on branches at the edge or very top of the canopy, or are only visible through a gap in the canopy, then projection errors can occur.

Figure 6.12 we show an example of this projection error using the ground truth annotations. We can see that in Figure 6.12b projected points form a tight cluster in the orthomosaic, while for a nearby nest the projected points are spread over a large area (Figure 6.12c). Assessing the well clustered case, we see that the depth maps form a relatively accurate model of the canopy at this location (when compared against the raw images). The detected nest sits towards the centre of the tree, and so in all three examples our projected ray intersects with the depth map at the correct location. However, in the poorly clustered example the nest sits at the top of a branch protruding from the main canopy. Since the surface mesh model does not accurately render these fine scale details, we see that the depth maps do not capture what is observed in the raw images. For example, in raw image DJI_0376 the nest is accurately annotated, but in the corresponding depth map we predict our projected ray to intersect with the road clearing behind. This is why points in this cluster are spread across locations on the road and surrounding canopy (Figure 6.12a). Plotting the histogram of all ground truth cluster areas in the dataset (for FOI points only, excluding guano), we find a long tailed distribution (Figure 6.13). While the majority of projected annotations form relatively tight clusters between 3 - 10 m^2, a small percent cover larger extents, up to a maximum of 597m^2. Clearly, individual nests cannot cover this area in real life, and so we can safely assume that this is caused by inaccurate projection when using the smooth 3D surface mesh. These outlying cases cause errors at the DBSCAN clustering stage, as the ε which produces the best results for the majority of the dataset does not capture these edge cases.

There are many options which could be considered to deal with projection error. Firstly the data collection and processing stage could be refined. For example during UAV transects images could be collected with higher degrees of overlap, to allow for clearer reconstruction of the canopy surface, and different SfM accuracy settings could be experimented with (for instance increasing the number of tie points). At the 2D-to-3D projection stage, errors could be accounted for to some extent by projecting all corners

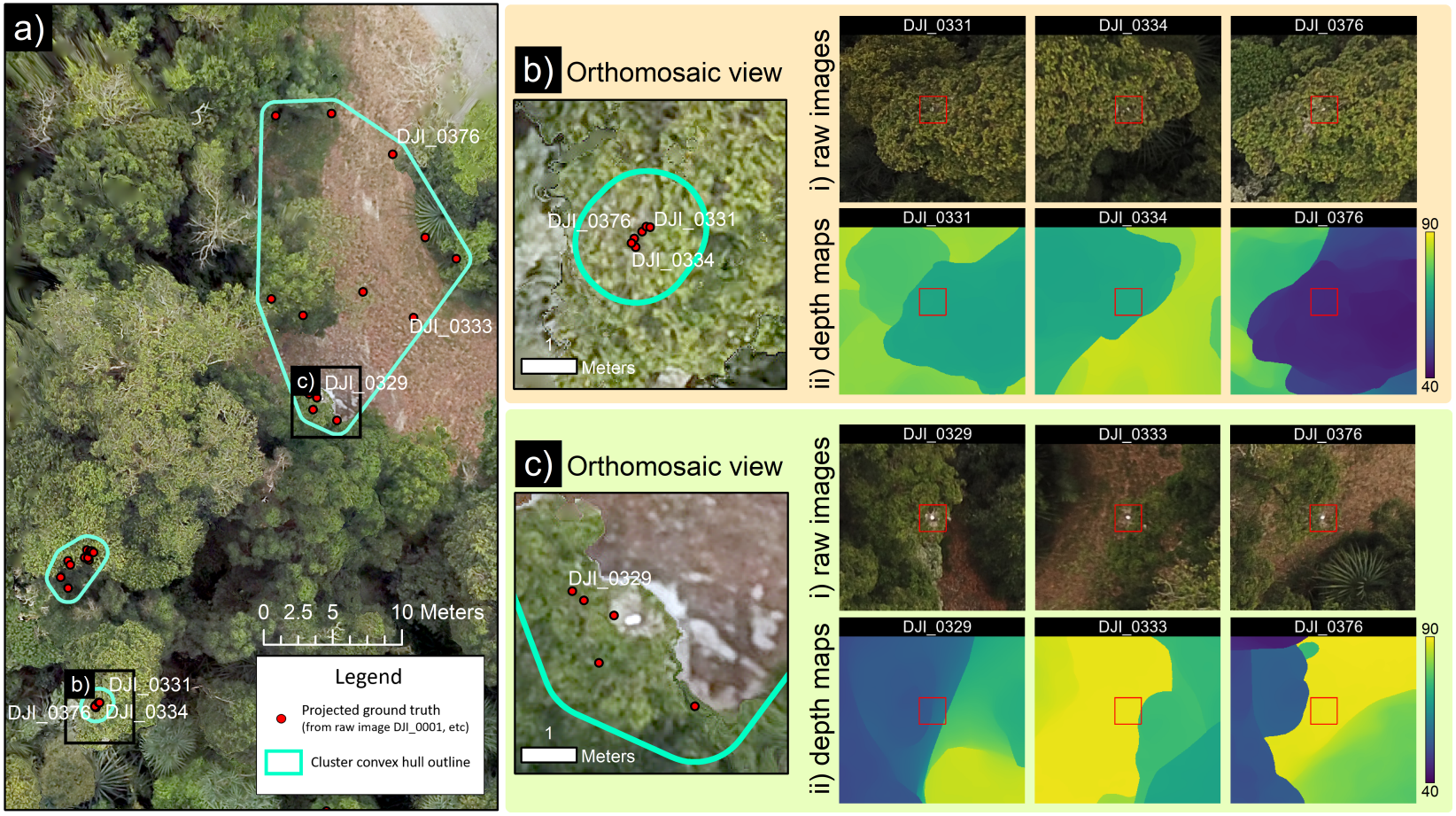

FIGURE 6.12: Example of projection error using the raw image ground truth annotations. a) An area of plot N04, with projected raw image detections (red dots) and each object's convex hull outline (blue lines). We compare b) a nest where points are projected accurately, forming a tight cluster and c) a nest where projected points are spread far apart. Plotting i) raw image detections (from images DJI_0333 etc) and ii) their corresponding depth map used for projection, we see this is due to the smoothness of the depth map. The branch with nest c) in images DJI_0333 and DJI_0376 is not resolved in the surface mesh, and so the ray cast from these points is estimated to intersect with the road behind.

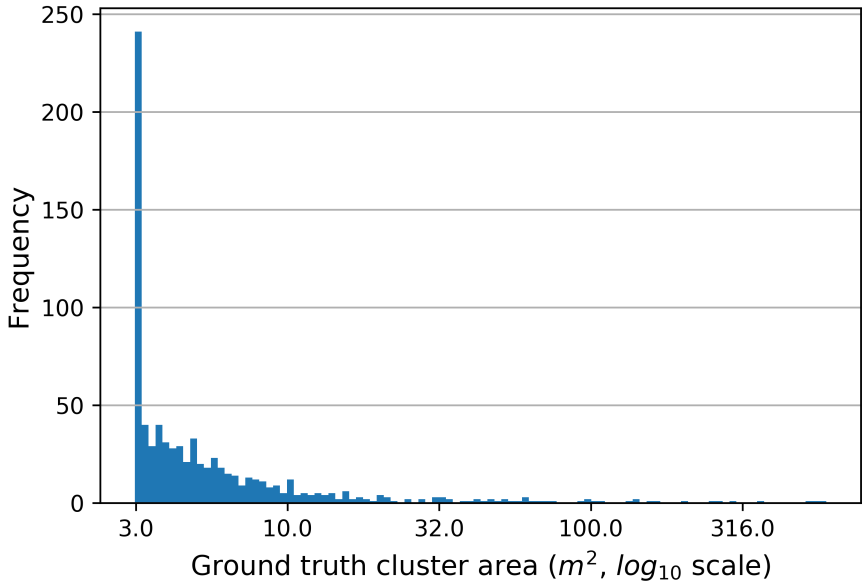

FIGURE 6.13: Histogram showing the distribution of FOI ground truth cluster areas, on a log_{10} scale. Many points are accurately projected and so form tight clusters, with areas between 3 - $10m^2$. However, there are outlying cases where clusters have areas of up to $597m^2$, caused by projection errors.

of the Faster R-CNN bounding boxes, as opposed to only the center point. Instead of single points this would produce polygons on the orthomosaic, which would appear more stretched where there is large uncertainty (Figure 6.14). This information could possibly be factored in at the clustering stage, for example by using different weightings for each point in the DBSCAN algorithm. Alternatives such as spectral clustering could also be investigated and compared to the success of DBSCAN. Finally, alternative measures could be used at the cluster assessment stage. Currently we use a cluster IoU of 0.1 to label predictions as either TP, FP, or FN, however we could change this criterion. For example we could compute the distances between every point in the predicted cluster and every point in the ground truth cluster, to factor in how well individual raw image detections are matched. Currently the penalty for cases with projection error is high (e.g for the example in Figure 6.11 we gain three FPs and one FN), so finding an alternative assessment would improve results for these outlying cases.

6.5.5 Applications and Future Work

In our discussion we have highlighted several potential areas for future work. This includes i) improving the results of our Faster R-CNN detector (with further discussion in Chapter 5), ii) reducing projection error at the 2D-to-3D projection stage, iii) testing alternative clustering algorithms or comparison measures at the clustering stage and iv) improving classification accuracy by adding additional features, such altitude. Many

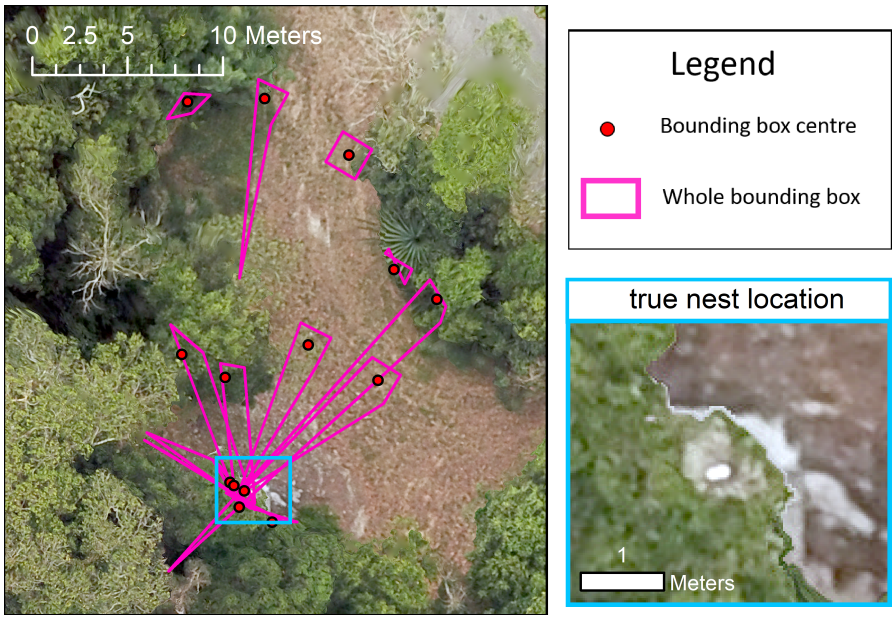

FIGURE 6.14: Comparison of projecting the centre of Faster R-CNN bounding boxes (red dots) compared to projecting the corners and forming polygons (pink lines). Projecting the corners captures some of the projection error, as polygons stretch towards the true nest location.

of these challenges are introduced by having a multi-stage system, where each stage is dependant on the output of the previous one. Ultimately, combining these separate stages into an end-to-end system would provide a more elegant and likely more successful method. For instance, in this case we train our CNN architecture as a first step, and later stages are all performed as post-processing. However if this projection information could be built into the CNN stage, either through training or making adjustments to the architecture itself, then the process could be optimised as one.

Despite these areas for improvements, the methods currently provide very promising results on a challenging dataset. We achieve final average recall values above 60% (FOI=0.62; guano=0.66) and precision scores of 67% for FOI and 50% for guano. Noting the exceptions and quick improvements outlined in the discussion, this could be directly applied to Abbott's booby surveys to reduce the manual analysis time. We also note that these values were calculated using our optimal F2-score thresholds (FOI=0.56; guano=0.44), however these could easily be adjusted to reduce the number of false negatives. Additional false positives could then be manually filtered by an expert observer, for example using automatically generated image galleries. While this manual analysis would take time to conduct, it would be significantly reduced compared to an exhaustive search of every raw image. Reducing thresholds to those determined by the F3-score for example (Figure 6.15), would increase average recall to approximately 70% for both FOI and guano, with precision dropping to 40% for both. In real terms for this dataset, this

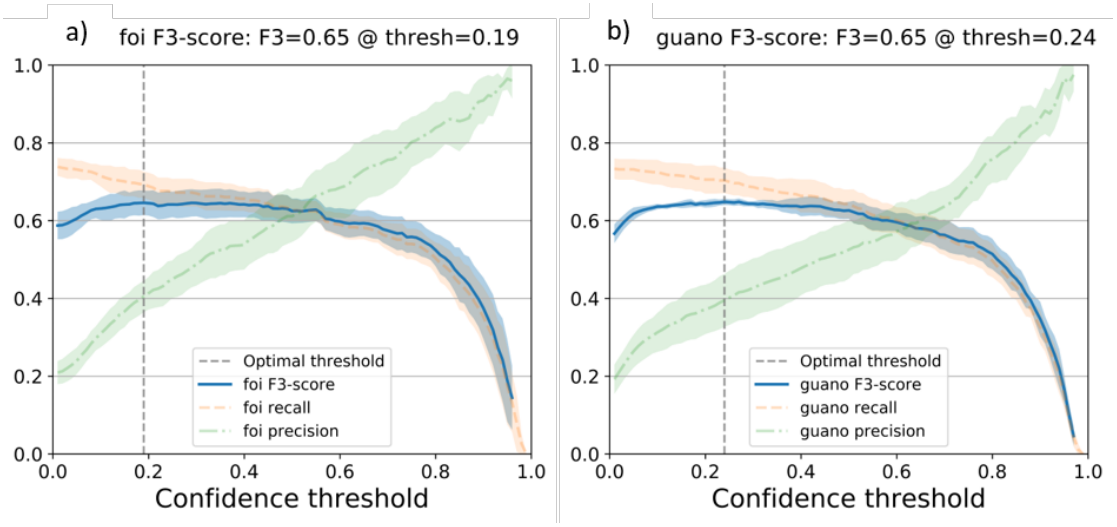

FIGURE 6.15: Fold-averaged F3-scores plotted against confidence threshold for a) FOI and b) guano detections.

would require an observer to manually filter through 827 galleries of proposed FOI clusters, with 333 true positives and 494 false positives (missing 148). It is likely a number of false positives could be removed in advance based on direct inspection of the orthomosaic, for example using expert knowledge of their preferred nesting habitat or tree species, or discounting false detections over the road. The cost benefit of this approach should be discussed with the end user, to assess the feasibility for future surveys.

6.6 Conclusion

In this Chapter we have outlined a method for detecting Abbott's boobies nesting in forest canopy using UAV SfM imagery. To the best of our knowledge, this is the first attempt to apply a multi-view detection method to monitor canopy species. Since we use RGB imagery it has particular benefit for surveys targeting objects such as nests (opposed to the direct detection of the animal itself), where thermal UAV imagery is not suitable. The methods have direct applicability to other canopy species, including other species of booby as well as surveys of chimpanzee [12, 145] and orangutan [154] nests. In fact, since all that is required is a SfM generated model, the techniques are directly applicable to any dataset constructed using this method. Similar approaches have already been used to improve detection of cattle in open grasslands [132], and would be of particular benefit in habitats here partial occlusion can occur at certain viewpoints. This could include species on cliffs, in semi-wooded grasslands, and mangroves. These also represent environments which are challenging to access for traditional ground-based counts, and so having an efficient UAV survey as an alternative is of particular benefit. In line with our albatross detection method (Chapters 3 and 4) we find that CNNs

can provide a general purpose framework for monitoring wildlife in remotely sensed datasets.

Chapter 7

Conclusions and Future Work

7.1 Contributions

In this thesis we have developed two approaches for detecting wildlife in challenging remote sensing datasets, using the latest CNN architectures.

Our first application, for research objective 1 (RO1), centered on detecting wandering albatrosses in 31-cm satellite imagery. This was the first application of deep learning to directly survey individual birds from space. In Chapter 3 we presented our U-Net detection method, and showed that the segmentation approach could successfully delineate the small point objects from the complex background (mAP=0.669). We showed the generalisability of the methods by performing a cross validation across islands, to assess how well the method transfers to new locations. We achieved mixed results for the four locations (average precision scores: BI=0.761, AN=0.508, AP=0.777, GC=0.628). Assessing the results on a per-island basis allowed us to identify causes for misclassifications, including false negatives under cloud haze and false positives due to image noise, which can be used to inform satellite surveys of other wildlife. We also highlighted the challenges with dealing with ground truth uncertainty and the effect of inter-observer variation, which motivated our experiments for RO2.

For RO2, in Chapter 4 we collected empirical data on inter-observer variation in satellite counts of albatrosses. The new (publicly available [16]) dataset of annotations can be used to benchmark manual and automated detection methods for the species, and consists of point annotations for four different colonies from 6 different observers. We developed a measure of inter-observer accuracy (based on the F1-score) which was directly comparable to our CNN results, which we suggest is a novel and useful method for similar analyses. Importantly, we found that our inter-observer F1-score accuracies

differed significantly between the four images in the dataset (BI: 0.74, AN: 0.67, BI: 0.92, GC: 0.85). These findings have direct implications for the counting of other species from satellite images, and emphasises the need to consider uncertainty on a case-by-case basis. In our network experiments we showed that choosing different labels at the training and assessment stage can have a significant impact on supervised training schemes. At assessment stage we found using the majority vote labels gave the overall best performance (mAP=0.74), while using a "mixed" ground (where a different set of observer labels are presented at each training epoch) provided promising results at the training stage (mAP=0.75). These insights help to inform the best practice for choosing ground truth labels for similar datasets, for instance those annotated by citizen scientists (e.g [81]).

For our second application we focused on detecting Abbott's boobies in forest canopy using UAV imagery (RO3). In Chapter 5 we showed that a fine-tuned Faster R-CNN architecture is able to detect nest site features of interest and guano with reasonable success mAP=0.5, FOI AP=0.52, Guano AP=0.47). False positives were largely caused by branches which had similar spectral properties and shape to nesting Abbott's boobies. Faster R-CNN performed well when nest sites and guano were clearly visible, with prediction confidence decreasing when objects were partially occluded by the forest canopy. In our misclassification analysis we found that in many cases false negatives could be compensated for by clear detections from another angle, which lead us to recommend and propose a multi-view approach.

In Chapter 6 our main contributions were implementing a multi-view method for detecting Abbott's boobies in UAV SfM imagery, to overcome issues with occlusion and uncertainty based on viewing angle (RO4). To the best of our knowledge this is the first attempt to apply this technique to canopy nesting species in UAV SfM imagery. We found the multi-view method gave reasonable results when using the F2-score threshold (recall: FOI=0.62, guano=0.66; precision: FOI=0.67, guano=0.5). However recall could be improved to 0.7 for both classes by adjusting to F3-score thresholds, with precision falling to 0.4. We suggest that the additional false positives could be easily removed in a manual review stage, and could be directly applied to Abbott's booby surveys in the future. We also investigated different causes of errors, including at the projection, clustering and classification stage, and proposed methods to overcome them. Highlighting these is an important step in developing these novel techniques in the future.

We highlight that while both methods we have developed in this thesis have direct applications to the target species, they also adopt general frameworks which could be easily transferred to other wildlife, as well as more general satellite and UAV object detection tasks.

7.2 Research Applications

Following on from our albatross research, we were successful in gaining funding to continue further work from Darwin Plus[1]. In the project we will be conducting an archipelago-wide survey of all wandering albatross colonies on South Georgia, leveraging crowd-sourced annotations from citizen scientists. This data will be collected and used to update the U-Net method, by adding more data to the training set. It will also allow for further research into inter-observer variation, following on from the findings of Chapter 4. This will be the first step in expanding the satellite survey method, with the intention of extending to a global survey of wandering albatrosses in the near future. The satellite survey technique will also be trialled on a new species, the Critically Endangered Tristan albatross (*Diomedea dabbenena*), to i) validate if they are detectable in WV-3 imagery and if so ii) test if the same U-Net method can transfer to different albatross species.

There are also direct applications of our U-Net method to other great albatross species, such as the northern royal albatross (*Diomedea sanfordi*), which have already been proven to be visible in WV-3 imagery [42]. In applications for wildlife apart from albatrosses, the U-Net approach presented in our publications [15, 16] has subsequently been used to detect wildebeest in 41cm GeoEye-1 satellite imagery with good success (F1-score of 0.87) [156]. This shows that the method transfers well to wildlife in other environments, and which have a lower contrast to background than the albatrosses in this study. It also shows promise for transferring to images collected by other VHR satellites, aside from WV-3. Given the wide range of animals that have been the subject of satellite survey - for example whales [28], seals [81, 104], elephants [33], polar bears [137] and penguins [40] - there is a great potential to transfer this general CNN framework to detect other wildlife.

Our multi-view Abbott's booby detection method will be directly applied to perform counts over an additional 70 forest plots across Christmas Island (Lipka et al., unpublished data). These plots have already been imaged and processed using SfM, but have not been manually annotated due to time constraints. The multi-view method will enable these to be analysed and added to the final census results for this year, and could be used to aid in future Abbott's booby surveys on the island. As discussed in Chapter 6, the multi-view approach could also be trialled on other canopy species, for example to detect chimpanzee and orangutan nests [12, 145, 154], and to other detection tasks where multi-view information would be beneficial.

[1]Project code DPLUS132: Monitoring albatrosses using very high resolution satellites and citizen science. URL: https://www.darwininitiative.org.uk/project/DPLUS132/

7.3 Discussion and Future Work

In Section 7.1 we outlined key successes and contributions from our research objectives. However we have noted limitations and areas for improvement throughout the thesis. Here we draw together common themes that apply across both applications, and identify areas for future work.

In our applications we noted that the inclusion of extra habitat information could help to guide automated detection methods. For example, in both cases digital elevation models (DEMs) could be used in conjunction with the spectral bands to encode habitat preference into the network decision. For the wandering albatross dataset this would help to reduce the chance of false positive detections of rocks in high altitude regions, as well as false detections in the ocean. Slope could also be calculated directly from the DEM, and used to highlight flatter regions of the islands where the wandering albatrosses prefer to nest [42]. For the Abbott's booby dataset incorporating the 3D canopy height model could help to exclude false positive detections on the road, and in trees which are lower than their preferred nesting height. By enforcing these constraints during training we can be more confident that our detections are in realistic habitat for the species of interest. The best method for encoding this extra information along with the spectral channels would have to be examined. It could be that DEMs could be stacked directly with the RGB inputs, although weight initialisation for these channels would have to be considered. It may also be achievable through multi-task learning [125], for example if the CNN were to learn to detect the target species as well as predict elevation as an auxiliary task. Advances in multi-task learning for 3D object detection, developed in the field of self-driving cars [90], could also provide an end-to-end solution for the Abbott's booby SfM detection task.

A second theme is that of ground-truth uncertainty. Although we did not collect empirical evidence of inter-observer variation for the Abbott's booby dataset, we noted that in many examples ground truth annotations made from certain viewpoints were unclear to an untrained observer. As we saw from our inter-observer experiments in Chapter 4, these uncertain annotations can have an effect on our supervised training scheme. This highlights the limitations of relying on ground truth data when training supervised CNN detection methods. We emphasise that instead of just assessing the level of ground truth uncertainty between human observers, more investigation is needed to assess the impact of ground truth label choice when training CNNs. Methods for dealing with uncertain ground truth, and in particular combining multiple observer labels, are particularly relevant given the recent trend of conducting large scale satellite surveys using crowd-sourcing. This includes LaRue et al.'s *Satellites Over Seals* project [81], the jointly run British Antarctic Survey/ World Wildlife Fund *Walrus From Space*

campaign [3], as well as the Darwin Plus albatross project described above in Section 7.2. These campaigns have provided a platform to process huge amounts of satellite imagery, for instance in the case of Weddell seals surveying over 250,000km^2 of Antarctic fast ice [81]. In these projects each image is assessed by multiple volunteers, and so when developing automated approaches the best choice of training label should be considered. As discussed in Chapter 4, this could include taking a majority vote census, forming a probabilistic ground truth, or testing new CNN architectures which build crowd layers into their design (for instance [123]). As more crowd-sourced data is collected, this is likely to be an active and interesting research area to pursue.

Finally, while we have developed applications for UAV and satellite images separately, multi-model surveys are an interesting avenue for future research. In these methods information from UAV and satellite images could be combined or transferred to aid detection. This has been employed to detect whales in satellite imagery, by training a CNN using downsampled aerial images [13]. More recently even web-searched photos have been used to train a CNN to detect polar bears in aerial images [21]. These techniques can be especially beneficial for sparsely distributed wildlife where it is challenging to collect real world examples for CNN training. For certain species which congregate in large groups and so can be monitored using non-VHR satellites (for instance emperor penguin colonies [45] and walrus haul-outs [36], which have both been identified in 10m resolution Sentinel-2 imagery), different resolution sensors could be used in combination. For example free Sentinel-2 imagery could be used to detect and monitor species congregations over large areas and fine temporal scales, and costly VHR images could be tasked to count individuals within the group. This would enable global monitoring of species which have extremely large ranges. For more elusive species such as the canopy nesting Abbott's booby, information from other sensors such as low-cost audio devices [131] could be used to inform nest locations. Ultimately data collected from multi-sensor networks, and processed using state-of-the-art machine learning, could pave the way for autonomous monitoring of species in challenging-to-access environments.

Appendix A

Supplementary material: Chapter 4

TABLE A.1: Average precision scores from the U-Net trained on the majority vote, and assessed against different ground truth labels.

	BI	AN	AP	GC	mAP
ref_ob	0.76	0.51	0.79	0.59	**0.66**
ob1	0.65	0.59	0.80	0.71	**0.69**
ob2	0.54	0.60	0.79	0.67	**0.65**
ob3	0.70	0.55	0.77	0.64	**0.67**
ob4	0.61	0.57	0.74	0.65	**0.64**
ob5	0.59	0.62	0.81	0.69	**0.68**
union	0.62	0.50	0.70	0.58	**0.60**
majority	0.75	0.68	0.79	0.72	**0.74**
inter	0.48	0.51	0.82	0.56	**0.59**

TABLE A.2: Average precision scores from the U-Net trained on different ground truth labels, and assessed against the majority vote.

	BI	**AN**	**AP**	**GC**	**mAP**
ref_ob	0.8	0.73	0.79	0.72	**0.76**
ob1	0.81	0.69	0.67	0.72	**0.72**
ob2	0.74	0.58	0.77	0.74	**0.71**
ob3	0.81	0.67	0.76	0.74	**0.74**
ob4	0.81	0.64	0.73	0.71	**0.72**
ob5	0.81	0.50	0.79	0.71	**0.70**
union	0.69	0.69	0.78	0.69	**0.71**
majority	0.75	0.68	0.79	0.72	**0.74**
inter	0.48	0.51	0.82	0.56	**0.59**
mixed	0.81	0.70	0.79	0.69	**0.75**

Appendix B

Supplementary material: Chapter 5

TABLE B.1: Data splits used for four-fold cross validation, including the number of features in each plot, and the total per fold.

Test Fold	Plot Ref	Date	FOI examples	Guano examples
1	N04	2019-09-26	442	504
	N05	2019-07-23	54	87
	N06	2019-07-23	115	223
	N07	2019-07-24	66	95
	N10	2019-08-14	258	171
	N22	2019-07-02	105	266
	N26	2019-10-01	145	144
	N60	2019-10-08	185	423
		Total:	**1370**	**1913**
2	N02	2019-09-18	121	52
	N03	2019-09-16	148	242
	N08	2019-10-16	21	25
	N11	2019-09-18	108	97
	N18	2019-09-24	26	20
	N25	2019-10-03	173	262
	N27	2019-07-29	579	989
	N114	2019-09-09	129	247
		Total:	**1305**	**1934**
3	N09	2019-08-15	353	254
	N17	2019-10-09	47	22
	N19	2019-07-04	10	40
	N38	2019-07-03	25	86
	N50	2019-09-18	99	143
	N73	2019-10-03	122	299
	N100	2019-10-22	651	608
	N110	2019-07-01	56	27
		Total:	**1363**	**1479**
4	N01	2019-07-22	32	81
	N14	2019-07-19	102	355
	N15	2019-07-17	0	13
	N20	2019-07-05	202	234
	N23	2019-10-10	342	281
	N24	2019-07-19	186	164
	N97	2019-09-10	283	975
	N101	2019-10-24	176	210
		Total:	**1323**	**2313**

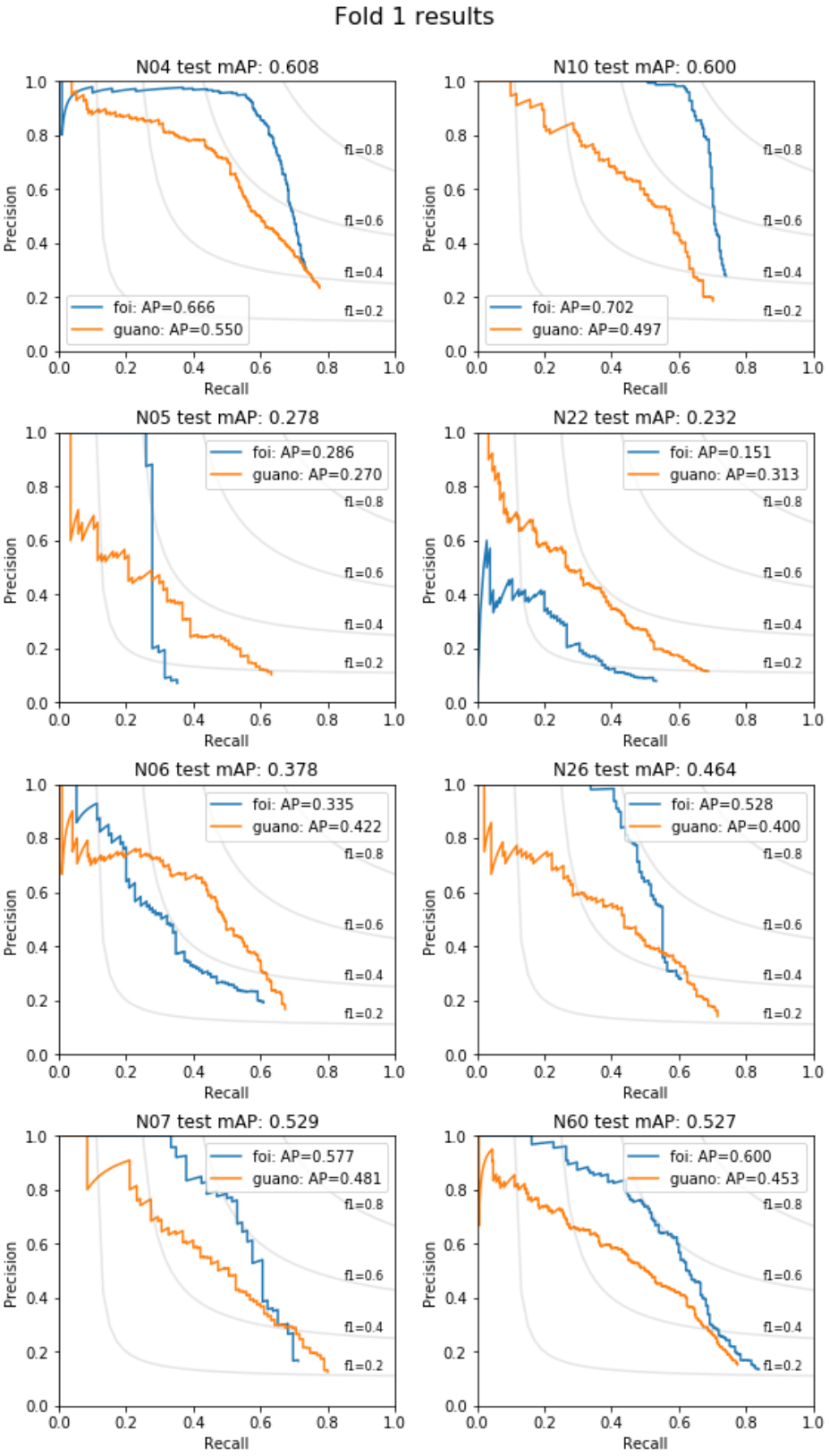

FIGURE B.1: Per-plot precision-recall curves for test data fold 1. Including average precision (AP) for FOI's and guano, as well as the mean average precision (mAP) averaged across classes.

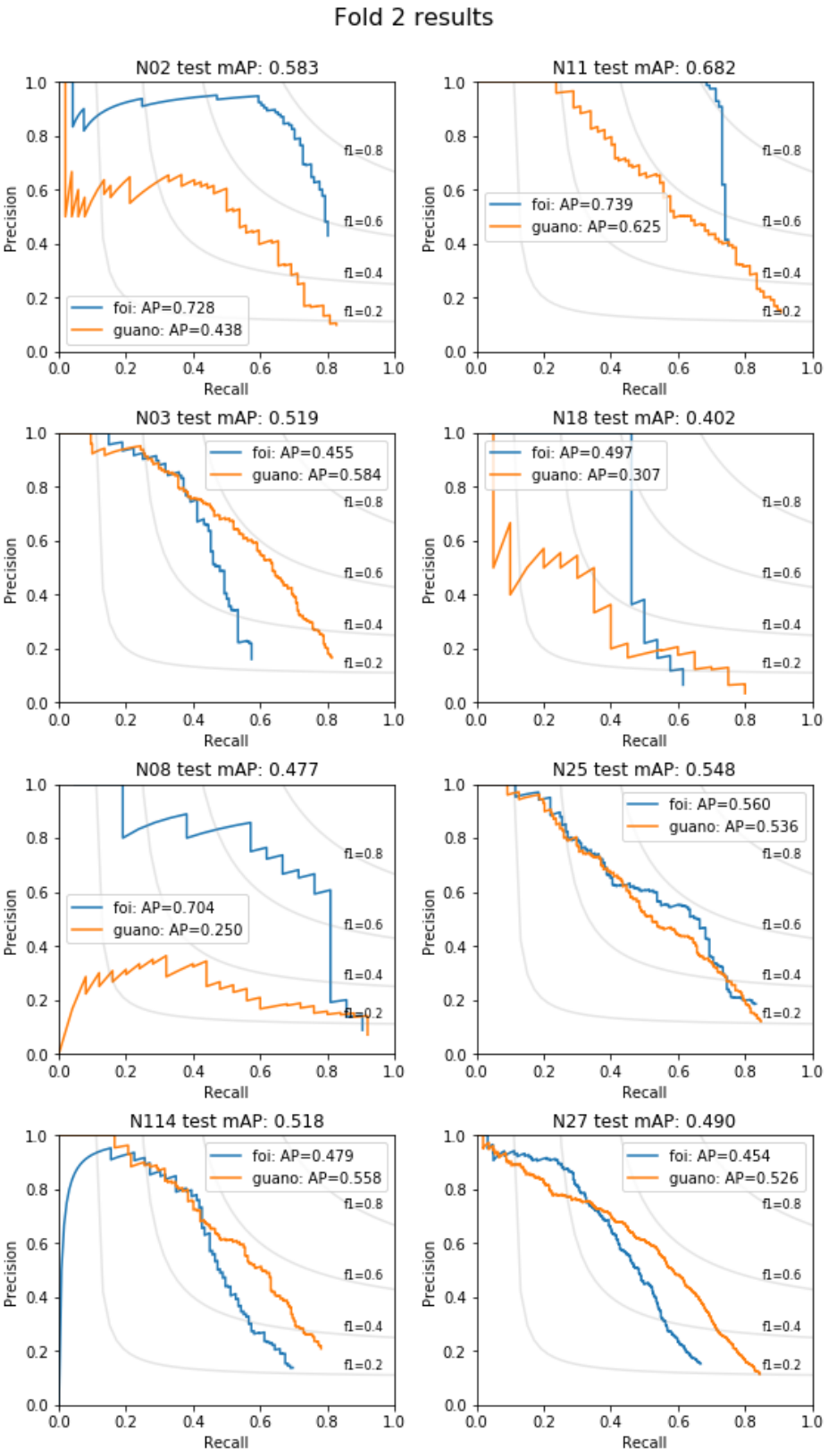

FIGURE B.2: Per-plot precision-recall curves for test data fold 2. Including average precision (AP) for FOI's and guano, as well as the mean average precision (mAP) averaged across classes.

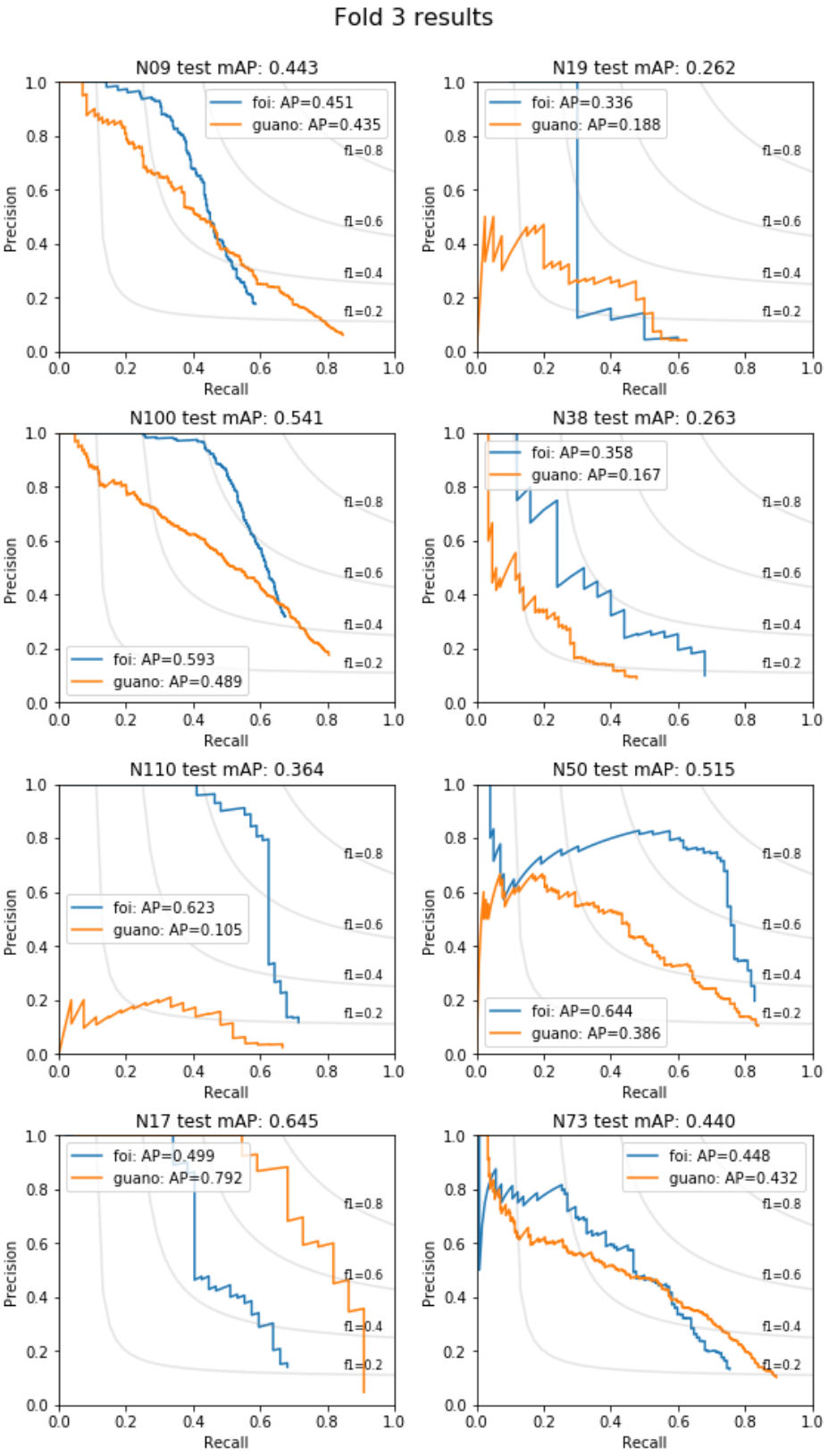

FIGURE B.3: Per-plot precision-recall curves for test data fold 3. Including average precision (AP) for FOI's and guano, as well as the mean average precision (mAP) averaged across classes.

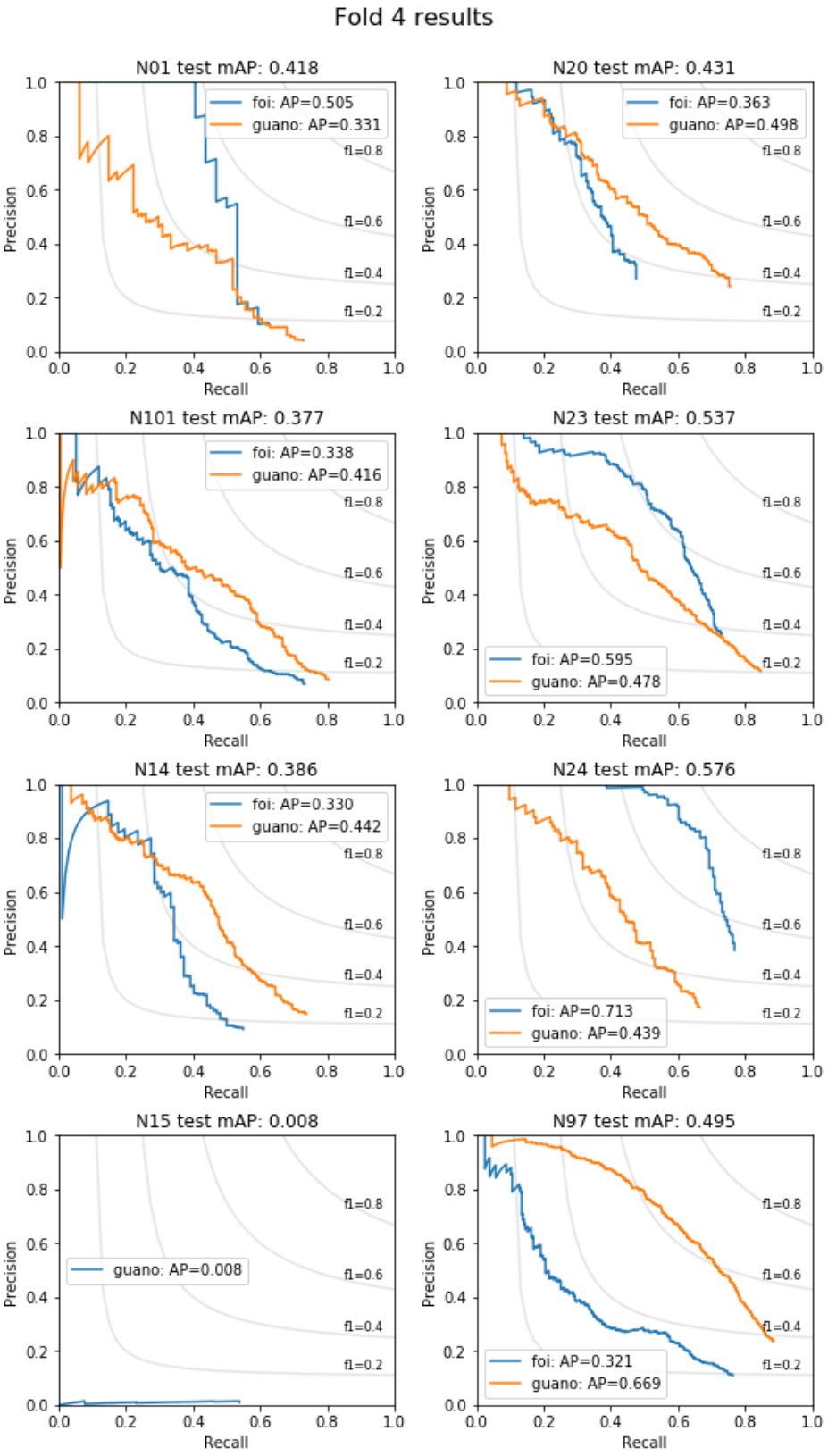

FIGURE B.4: Per-plot precision-recall curves for test data fold 4. Including average precision (AP) for FOI's and guano, as well as the mean average precision (mAP) averaged across classes.

Appendix C

Supplementary material: Chapter 6

TABLE C.1: Final results for every plot in the four test folds. mAP is the mean of the average precision scores across classes (FOI and guano). We report the average precision (AP) per class, the recall, precision, and actual number of true positives and negatives, and false positives and negatives.

Fold	Plot	mAP	Class	AP	Rec	Prec	TPs	FNs	FPs	TNs
1	N04	**0.665**	foi	0.73	0.68	0.71	25	12	10	76
			guano	0.6	0.67	0.59	22	11	15	55
	N05	**0.325**	foi	0.28	0.29	0.67	2	5	1	20
			guano	0.37	0.33	0.38	3	6	5	23
	N06	**0.585**	foi	0.51	0.38	0.71	5	8	2	23
			guano	0.66	0.7	0.76	16	7	5	24
	N07	**0.66**	foi	0.7	0.67	0.67	4	2	2	19
			guano	0.62	0.64	0.54	7	4	6	18
	N10	**0.715**	foi	0.77	0.71	0.85	17	7	3	34
			guano	0.66	0.5	1	10	10	0	19
	N22	**0.515**	foi	0.64	0.64	0.9	9	5	1	40
			guano	0.39	0.41	0.46	13	19	15	77
	N26	**0.745**	foi	0.78	0.58	1	7	5	0	21
			guano	0.71	0.83	0.71	15	3	6	43
	N60	**0.685**	foi	0.75	0.81	0.65	17	4	9	46
			guano	0.62	0.64	0.54	21	12	18	62
2	N02	**0.81**	foi	0.84	0.88	0.88	7	1	1	9
			guano	0.78	0.67	0.8	4	2	1	21
	N03	**0.695**	foi	0.6	0.59	0.77	10	7	3	26
			guano	0.79	0.78	0.64	18	5	10	40
	N08	**0.71**	foi	0.67	1	0.5	3	0	3	10
			guano	0.75	1	0.5	4	0	4	8
	N11	**0.79**	foi	0.88	0.88	0.88	7	1	1	14
			guano	0.7	0.89	0.53	8	1	7	23
	N18	**0.365**	foi	0.5	1	0.4	2	0	3	10
			guano	0.23	0.75	0.23	3	1	10	25
	N25	**0.74**	foi	0.71	0.8	0.67	8	2	4	55
			guano	0.77	0.88	0.52	14	2	13	64
	N27	**0.51**	foi	0.54	0.5	0.69	24	24	11	146
			guano	0.48	0.53	0.46	31	28	36	137
	N114	**0.49**	foi	0.43	0.64	0.6	9	5	6	39
			guano	0.55	0.56	0.71	10	8	4	41
3	N09	**0.6**	foi	0.69	0.65	0.68	17	9	8	55
			guano	0.51	0.86	0.3	12	2	28	81
	N17	**0.585**	foi	0.5	0.5	1	3	3	0	19
			guano	0.67	1	0.5	3	0	3	20
	N19	**0.345**	foi	0.5	0.5	1	1	1	0	11
			guano	0.19	0.14	0.1	1	6	9	26
	N38	**0.275**	foi	0.4	0.29	0.5	2	5	2	11
			guano	0.15	0.23	0.43	3	10	4	26
	N50	**0.625**	foi	0.62	0.67	0.44	8	4	10	15
			guano	0.63	0.67	0.52	14	7	13	34
	N73	**0.65**	foi	0.62	0.54	0.7	7	6	3	29
			guano	0.68	0.82	0.41	14	3	20	74
	N100	**0.705**	foi	0.66	0.66	0.72	21	11	8	78
			guano	0.75	0.89	0.52	33	4	30	85
	N110	**0.44**	foi	0.54	0.57	0.57	4	3	3	22
			guano	0.34	0.5	0.17	2	2	10	51
4	N01	**0.38**	foi	0.42	0.5	0.75	3	3	1	13
			guano	0.34	0.44	0.16	4	5	21	69
	N14	**0.39**	foi	0.25	0.24	0.5	4	13	4	47
			guano	0.53	0.67	0.47	16	8	18	59
	N15	**0**	foi	0		0	0	0	2	14
			guano	0	0	0	0	3	8	47
	N20	**0.58**	foi	0.52	0.36	0.83	10	18	2	24
			guano	0.64	0.76	0.68	19	6	9	34
	N23	**0.67**	foi	0.74	0.75	0.75	18	6	6	53
			guano	0.6	0.69	0.51	18	8	17	79
	N24	**0.625**	foi	0.75	0.81	0.76	13	3	4	17
			guano	0.5	0.47	0.7	7	8	3	39
	N97	**0.515**	foi	0.35	0.77	0.44	20	6	25	131
			guano	0.68	0.76	0.57	37	12	28	95
	N101	**0.5**	foi	0.54	0.6	0.43	9	6	12	93
			guano	0.46	0.7	0.34	14	6	27	79

Bibliography

[1] Birdlife international. 2019. *Papasula abbotti. The IUCN Red List of Threatened Species 2019*: e.t22696649a152726109. Downloaded on 22 November 2021.

[2] Birdlife international. species factsheet: Diomedea exulans. Downloaded from http://www.birdlife.org on 08/06/2020.

[3] Walrus from space website, https://www.wwf.org.uk/learn/walrus-from-space. Accessed on 20/01/2022.

[4] Abileah, R. Marine mammal census using space satellite imagery. *U.S. Navy Journal of Underwater Acoustics 52*, 3 (2002), 709–724.

[5] Ainley, D. G., Larue, M. A., Stirling, I., Stammerjohn, S., and Siniff, D. B. An apparent population decrease, or change in distribution, of Weddell seals along the Victoria Land coast. *Marine Mammal Science 31*, 4 (oct 2015), 1338–1361.

[6] Anderson, K., and Gaston, K. J. Lightweight unmanned aerial vehicles will revolutionize spatial ecology. *Frontiers in Ecology and the Environment 11*, 3 (2013), 138–146.

[7] Bajzak, D., and Piatt, J. F. Computer-aided procedure for counting waterfowl on aerial photographs. *Wildlife Society Bulletin 18*, 2 (1990), 125–129.

[8] Barber-Meyer, S. M., Kooyman, G. L., and Ponganis, P. J. Estimating the relative abundance of emperor penguins at inaccessible colonies using satellite imagery. *Polar Biology 30*, 12 (oct 2007), 1565–1570.

[9] Bird, C. N., Dawn, A. H., Dale, J., and Johnston, D. W. A semi-automated method for estimating adélie penguin colony abundance from a fusion of multi-spectral and thermal imagery collected with unoccupied aircraft systems. *Remote Sensing 12*, 22 (2020), 3692.

[10] Boltunov, A., Evtushenko, N., Knijnikov, A., Puhova, M., and Semenova, V. Space technology for the marine mammal research and conservation in the Arctic.

[11] Bondi, E., Fang, F., Hamilton, M., Kar, D., Dmello, D., Choi, J., Hannaford, R., Iyer, A., Joppa, L., Tambe, M., and Nevatia, R. SPOT Poachers in Action : Augmenting Conservation Drones with Automatic Detection in Near Real Time.

[12] Bonnin, N., Van Andel, A. C., Kerby, J. T., Piel, A. K., Pintea, L., and Wich, S. A. Assessment of chimpanzee nest detectability in drone-acquired images. *Drones 2*, 2 (2018), 17.

[13] Borowicz, A., Le, H., Humphries, G., Nehls, G., Höschle, C., Kosarev, V., and Lynch, H. J. Aerial-trained deep learning networks for surveying cetaceans from satellite imagery. *PLoS ONE 14*, 10 (2019).

[14] Borowicz, A., McDowall, P., Youngflesh, C., Sayre-McCord, T., Clucas, G., Herman, R., Forrest, S., Rider, M., Schwaller, M., Hart, T., Jenouvrier, S., Polito, M. J., Singh, H., and Lynch, H. J. Multi-modal survey of Adélie penguin mega-colonies reveals the Danger Islands as a seabird hotspot. *Scientific Reports 8*, 1 (dec 2018), 3926.

[15] Bowler, E., Fretwell, P. T., French, G., and Mackiewicz, M. Using deep learning to count albatrosses from space. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium* (2019), IEEE, pp. 10099–10102.

[16] Bowler, E., Fretwell, P. T., French, G., and Mackiewicz, M. Using deep learning to count albatrosses from space: Assessing results in light of ground truth uncertainty. *Remote Sensing 12*, 12 (2020), 2026.

[17] Brack, I. V., Kindel, A., and Oliveira, L. F. B. Detection errors in wildlife abundance estimates from unmanned aerial systems (uas) surveys: Synthesis, solutions, and challenges. *Methods in Ecology and Evolution 9*, 8 (2018), 1864–1873.

[18] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. Albumentations: Fast and flexible image augmentations. *Information 11*, 2 (2020).

[19] Chabot, D., and Bird, D. M. Evaluation of an off-the-shelf Unmanned Aircraft System for Surveying Flocks of Geese. *Waterbirds 35*, 1 (mar 2012), 170–174.

[20] Chabot, D., and Bird, D. M. Evaluation of an off-the-shelf unmanned aircraft system for surveying flocks of geese. *Waterbirds 35*, 1 (2012), 170–174.

[21] Chabot, D., Stapleton, S., and Francis, C. M. Using web images to train a deep neural network to detect sparsely distributed wildlife in large volumes of remotely sensed imagery: A case study of polar bears on sea ice. *Ecological Informatics* (2022), 101547.

[22] Charry, B., Tissier, E., Iacozza, J., Marcoux, M., and Watt, C. A. Mapping arctic cetaceans from space: A case study for beluga and narwhal. *PloS one 16*, 8 (2021), e0254380.

[23] Che, Y., Wang, Q., Xie, Z., Zhou, L., Li, S., Hui, F., Wang, X., Li, B., and Ma, Y. Estimation of maize plant height and leaf area index dynamics using an unmanned aerial vehicle with oblique and nadir photography. *Annals of botany 126*, 4 (2020), 765–773.

[24] Chrétien, L.-P., Théau, J., and Ménard, P. Visible and thermal infrared remote sensing for the detection of white-tailed deer using an unmanned aerial system. *Wildlife Society Bulletin 40*, 1 (mar 2016), 181–191.

[25] Corcoran, E., Denman, S., Hanger, J., Wilson, B., and Hamilton, G. Automated detection of koalas using low-level aerial surveillance and machine learning. *Scientific reports 9*, 1 (2019), 1–9.

[26] Corcoran, E., Winsen, M., Sudholz, A., and Hamilton, G. Automated detection of wildlife using drones: Synthesis, opportunities and constraints. *Methods in Ecology and Evolution 12*, 6 (2021), 1103–1114.

[27] Cubaynes, H. C. *Whales from space: Assessing the feasibility of using satellite imagery to monitor whales*. PhD thesis, University of Cambridge, 2020.

[28] Cubaynes, H. C., Fretwell, P. T., Bamford, C., Gerrish, L., and Jackson, J. A. Whales from space: Four mysticete species described using new VHR satellite imagery. *Marine Mammal Science 35*, 2 (apr 2019), 466–491.

[29] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09* (2009).

[30] Descamps, S., Bechet, A., Descombes, X., Arnaud, A., and Zerubia, J. An automatic counter for aerial images of aggregations of large birds. *Bird Study 58*, 3 (aug 2011), 302–308.

[31] Dugdale, S. J., Malcolm, I. A., and Hannah, D. M. Drone-based structure-from-motion provides accurate forest canopy data to assess shading effects in river temperature models. *Science of The Total Environment 678* (2019), 326–340.

[32] Dulava, S., Bean, W. T., and Richmond, O. M. Environmental reviews and case studies: applications of unmanned aircraft systems (uas) for waterbird surveys. *Environmental Practice 17*, 3 (2015), 201–210.

[33] Duporge, I., Isupova, O., Reece, S., Macdonald, D. W., and Wang, T. Using very-high-resolution satellite imagery and deep learning to detect and count african elephants in heterogeneous landscapes. *Remote Sensing in Ecology and Conservation 7*, 3 (2021), 369–381.

[34] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (1996), vol. 96, pp. 226–231.

[35] Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., et al. U-net: deep learning for cell counting, detection, and morphometry. *Nature methods 16*, 1 (2019), 67–70.

[36] Fischbach, A. S., and Douglas, D. C. Evaluation of satellite imagery for monitoring pacific walruses at a large coastal haulout. *Remote Sensing 13*, 21 (2021), 4266.

[37] Forsyth, D. A., and Ponce, J. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[38] Fredembach, C., and Süsstrunk, S. Colouring the near-infrared. In *Color and Imaging Conference* (2008), vol. 2008, Society for Imaging Science and Technology, pp. 176–182.

[39] Fretwell, P. T., Jackson, J. A., Encina, M. J. U., Häussermann, V., Alvarez, M. J. P., Olavarría, C., and Gutstein, C. S. Using remote sensing to detect whale strandings in remote areas: The case of sei whales mass mortality in chilean patagonia. *PloS one 14*, 10 (2019).

[40] Fretwell, P. T., LaRue, M. A., Morin, P., Kooyman, G. L., Wienecke, B., Ratcliffe, N., Fox, A. J., Fleming, A. H., Porter, C., and Trathan, P. N. An emperor penguin population estimate: The first global, synoptic survey of a species from space. *PLoS ONE 7*, 4 (apr 2012), e33751.

[41] Fretwell, P. T., Phillips, R. A., Brooke, M. d. L., Fleming, A. H., and McArthur, A. Using the unique spectral signature of guano to identify unknown seabird colonies. *Remote Sensing of Environment 156* (2015), 448–456.

[42] Fretwell, P. T., Scofield, P., and Phillips, R. A. Using super-high resolution satellite imagery to census threatened albatrosses. *Ibis 159*, 3 (jul 2017), 481–490.

[43] Fretwell, P. T., Staniland, I. J., and Forcada, J. Whales from space: Counting southern right whales by satellite. *PLoS ONE 9*, 2 (feb 2014), e88655.

[44] Fretwell, P. T., and Trathan, P. N. Penguins from space: Faecal stains reveal the location of emperor penguin colonies. *Global Ecology and Biogeography 18*, 5 (sep 2009), 543–552.

[45] Fretwell, P. T., and Trathan, P. N. Discovery of new colonies by sentinel2 reveals good and bad news for emperor penguins. *Remote Sensing in Ecology and Conservation 7*, 2 (2021), 139–153.

[46] Fukuda, T., Nada, H., Adachi, H., Shimizu, S., Takei, C., Sato, Y., Yabuki, N., and Motamedi, A. Integration of a structure from motion into virtual and augmented reality for architectural and urban simulation. In *International Conference on Computer-Aided Architectural Design Futures* (2017), Springer, pp. 60–77.

[47] Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J. R., Morros, J.-R., Ruiz-Hidalgo, J., Vilaplana, V., and Gregorio, E. Fruit detection and 3d location using instance segmentation neural networks and structure-from-motion photogrammetry. *Computers and Electronics in Agriculture 169* (2020), 105165.

[48] Giese, M., and Riddle, M. Disturbance of emperor penguin Aptenodytes forsteri chicks by helicopters. *Polar Biology 22*, 6 (nov 1999), 366–371.

[49] Gilmer, D. S., Brass, J. A., Strong, L. L., and Card, D. H. Goose counts from aerial photographs using an optical digitizer. *The wildlife society Bulletin 16*, 2 (1988), 204–206.

[50] GIMP. GNU Image Manipulation Programme for Windows Version 2.8.14., 2018.

[51] Girshick, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1440–1448.

[52] Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of*

the IEEE conference on computer vision and pattern recognition (2014), pp. 580–587.

[53] Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. *AISTATS '11: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics 15* (2011), 315–323.

[54] Gonçalves, B. C., Spitzbart, B., and Lynch, H. J. Sealnet: A fully-automated pack-ice seal detection pipeline for sub-meter satellite imagery. *Remote Sensing of Environment 239* (2020), 111617.

[55] Gonzalez, L. F., Montes, G. A., Puig, E., Johnson, S., Mengersen, K., and Gaston, K. J. Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation. *Sensors (Switzerland) 16*, 1 (jan 2016), 97.

[56] Gray, P. C., Fleishman, A. B., Klein, D. J., McKown, M. W., Bezy, V. S., Lohmann, K. J., and Johnston, D. W. A convolutional neural network for detecting sea turtles in drone imagery. *Methods in Ecology and Evolution 10*, 3 (2019), 345–355.

[57] Green, S., Bevan, A., and Shapland, M. A comparative assessment of structure from motion methods for archaeological research. *Journal of Archaeological Science 46* (2014), 173–181.

[58] Groom, G., Petersen, I. K., Anderson, M. D., and Fox, A. D. Using object-based analysis of image data to count birds: Mapping of Lesser Flamingos at Kamfers Dam, Northern Cape, South Africa. *International Journal of Remote Sensing 32*, 16 (2011), 4611–4639.

[59] Guo, F., Cai, Z.-x., Xie, B., and Tang, J. Review and prospect of image dehazing techniques. *Jisuanji Yingyong/ Journal of Computer Applications 30*, 9 (2010), 2417–2421.

[60] Hänsch, R., and Hellwich, O. The truth about ground truth: Label noise in human-generated reference data. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium* (2019), IEEE, pp. 5594–5597.

[61] Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision*, 2 ed. Cambridge University Press, 2004.

[62] He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2961–2969.

[63] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[64] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors.

[65] Hodgson, A., Peel, D., and Kelly, N. Unmanned aerial vehicles for surveying marine fauna: Assessing detection probability. *Ecological Applications 27*, 4 (jun 2017), 1253–1267.

[66] Hodgson, J. C., Baylis, S. M., Mott, R., Herrod, A., and Clarke, R. H. Precision wildlife monitoring using unmanned aerial vehicles. *Scientific Reports 6*, 1 (2016), 22574.

[67] Hodgson, J. C., Mott, R., Baylis, S. M., Pham, T. T., Wotherspoon, S., Kilpatrick, A. D., Raja Segaran, R., Reid, I., Terauds, A., and Koh, L. P. Drones count wildlife more accurately and precisely than humans, feb 2018.

[68] Hoekendijk, J., Kellenberger, B., Aarts, G., Brasseur, S., Poiesz, S. S., and Tuia, D. Counting using deep learning regression gives value to ecological surveys. *Scientific reports 11*, 1 (2021), 1–12.

[69] Hollings, T., Burgman, M., van Andel, M., Gilbert, M., Robinson, T., and Robinson, A. How do you find the green sheep? A critical review of the use of remotely sensed imagery to detect and count animals, feb 2018.

[70] Hong, S.-J., Han, Y., Kim, S.-Y., Lee, A.-Y., and Kim, G. Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery. *Sensors 19*, 7 (2019), 1651.

[71] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[72] Hughes, B. J., Martin, G. R., and Reynolds, S. J. The use of google earthtm satellite imagery to detect the nests of masked boobies sula dactylatra. *Wildlife Biology 17*, 2 (2011), 210–216.

[73] Iglhaut, J., Cabo, C., Puliti, S., Piermattei, L., O'Connor, J., and Rosette, J. Structure from motion photogrammetry in forestry: A review. *Current Forestry Reports 5*, 3 (2019), 155–168.

[74] Jiang, H., and Lu, N. Multi-scale residual convolutional neural network for haze removal of remote sensing images. *Remote Sensing 10*, 6 (2018), 945.

[75] Kellenberger, B., Veen, T., Folmer, E., and Tuia, D. 21 000 birds in 4.5 h: efficient large-scale seabird detection with machine learning. *Remote Sensing in Ecology and Conservation* (2021).

[76] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[77] Korznikov, K. A., Kislov, D. E., Altman, J., Doležal, J., Vozmishcheva, A. S., and Krestov, P. V. Using u-net-like deep convolutional neural networks for precise tree recognition in very high resolution rgb (red, green, blue) satellite images. *Forests 12*, 1 (2021), 66.

[78] Krizhevsky, A., Sutskever, I., and Geoffrey E., H. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS2012)* (2012), 1–9.

[79] Laliberte, A. S., and Ripple, W. J. Automated wildlife counts from remotely sensed imagery. *Wildlife Society Bulletin 31*, 2 (2003), 362–371.

[80] LaRue, M., Brooks, C., Wege, M., Salas, L., and Gardiner, N. High-resolution satellite imagery meets the challenge of monitoring remote marine protected areas in the antarctic and beyond. *Conservation Letters* (2022), e12884.

[81] LaRue, M. A., Ainley, D. G., Pennycook, J., Stamatiou, K., Salas, L., Nur, N., Stammerjohn, S., and Barrington, L. Engaging 'the crowd'in remote sensing to learn about habitat affinity of the weddell seal in antarctica. *Remote Sensing in Ecology and Conservation 6*, 1 (2020), 70–78.

[82] Larue, M. A., and Knight, J. Applications of Very High-Resolution Imagery in the Study and Conservation of Large Predators in the Southern Ocean. *Conservation Biology 28*, 6 (dec 2014), 1731–1735.

[83] LaRue, M. A., Lynch, H., Lyver, P., Barton, K., Ainley, D., Pollard, A., Fraser, W., and Ballard, G. A method for estimating colony sizes of adélie penguins using remote sensing imagery. *Polar Biology 37*, 4 (2014), 507–517.

[84] LaRue, M. A., Rotella, J. J., Garrott, R. A., Siniff, D. B., Ainley, D. G., Stauffer, G. E., Porter, C. C., and Morin, P. J. Satellite imagery can be used to detect variation in abundance of Weddell seals (Leptonychotes weddellii) in Erebus Bay, Antarctica. *Polar Biology 34*, 11 (2011), 1727–1737.

[85] LaRue, M. A., Stapleton, S., and Anderson, M. Feasibility of using high-resolution satellite imagery to assess vertebrate wildlife populations. *Conservation Biology 31*, 1 (feb 2017), 213–220.

[86] Larue, M. A., Stapleton, S., Porter, C., Atkinson, S., Atwood, T., Dyck, M., and Lecomte, N. Testing methods for using high-resolution satellite imagery to monitor polar bear abundance and distribution. *Wildlife Society Bulletin 39*, 4 (dec 2015), 772–779.

[87] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2323.

[88] LeCun, Y. A., Bengio, Y., and Hinton, G. E. Deep learning. *Nature 521*, 7553 (may 2015), 436–444.

[89] Lee, K.-Y., and Sim, J.-Y. Cloud removal of satellite images using convolutional neural network with reliable cloudy image synthesis model. In *2019 IEEE International Conference on Image Processing (ICIP)* (2019), IEEE, pp. 3581–3585.

[90] Liang, M., Yang, B., Chen, Y., Hu, R., and Urtasun, R. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 7345–7353.

[91] Liang, S., and Wang, J. *Advanced remote sensing: terrestrial information extraction and applications*. Academic Press, 2019.

[92] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2980–2988.

[93] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision* (2014), Springer, pp. 740–755.

[94] Linchant, J., Lisein, J., Semeki, J., Lejeune, P., and Vermeulen, C. Are unmanned aircraft systems (UASs) the future of wildlife monitoring? A review of accomplishments and challenges, oct 2015.

[95] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. Ssd: Single shot multibox detector. In *European conference on computer vision* (2016), Springer, pp. 21–37.

[96] Löffler, E., and Margules, C. Wombats detected from space. *Remote Sensing of Environment 9*, 1 (1980), 47–56.

[97] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3431–3440.

[98] Long, J., Shi, Z., Tang, W., and Zhang, C. Single remote sensing image dehazing. *IEEE Geoscience and Remote Sensing Letters 11*, 1 (2013), 59–63.

[99] Longmore, S. N., Collins, R. P., Pfeifer, S., Fox, S. E., Mulero-Pázmány, M., Bezombes, F., Goodwin, A., De Juan Ovelar, M., Knapen, J. H., and Wich, S. A. Adapting astronomical source detection software to help detect animals in thermal images obtained by unmanned aerial systems. *International Journal of Remote Sensing 38*, 8-10 (feb 2017), 2623–2638.

[100] Lynch, H. J., and LaRue, M. A. First global census of the Adélie Penguin. *The Auk 131*, 4 (oct 2014), 457–466.

[101] Maire, F., Mejias, L., Hodgson, A., and Duclos, G. Detection of dugongs from unmanned aerial vehicles. In *IEEE International Conference on Intelligent Robots and Systems* (nov 2013), IEEE, pp. 2750–2756.

[102] McCarthy, E. D., Martin, J. M., Boer, M. M., and Welbergen, J. A. Drone-based thermal remote sensing provides an effective new tool for monitoring the abundance of roosting fruit bats. *Remote Sensing in Ecology and Conservation* (2021).

[103] McCLELLAND, G. T., Bond, A. L., Sardana, A., and Glass, T. Rapid population estimate of a surface-nesting seabird on a remote island using a low-cost unmanned aerial vehicle. *Marine Ornithology 44* (2016), 215–220.

[104] McMahon, C. R., Howe, H., Van Den Hoff, J., Alderman, R., Brolsma, H., and Hindell, M. A. Satellites, the all-seeing eyes in the sky: Counting elephant seals from space. *PLoS ONE 9*, 3 (mar 2014), e92613.

[105] McNeill, S., Barton, K., Lyver, P., and Pairman, D. Semi-automated penguin counting from digital aerial photographs. In *International Geoscience and Remote Sensing Symposium (IGARSS)* (jul 2011), IEEE, pp. 4312–4315.

[106] Milne, S., Martin, J. G., Reynolds, G., Vairappan, C. S., Slade, E. M., Brodie, J. F., Wich, S. A., Williamson, N., and Burslem, D. F. Drivers of bornean orangutan distribution across a multiple-use tropical landscape. *Remote Sensing 13*, 3 (2021), 458.

[107] Moy de Vitry, M., Schindler, K., Rieckermann, J., and Leitão, J. P. Sewer inlet localization in uav image clouds: Improving performance with multi-view detection. *Remote Sensing 10*, 5 (2018), 706.

[108] Naveen, R., Lynch, H. J., Forrest, S., Mueller, T., and Polito, M. First direct, site-wide penguin survey at deception island, antarctica, suggests significant declines in breeding chinstrap penguins. *Polar Biology 35*, 12 (2012), 1879–1888.

[109] O'Dowd, D. J., Green, P. T., and Lake, P. S. Invasional 'meltdown'on an oceanic island. *Ecology letters 6*, 9 (2003), 812–817.

[110] Oishi, Y., and Matsunaga, T. Support system for surveying moving wild animals in the snow using aerial remote-sensing images. *International Journal of Remote Sensing 35*, 4 (feb 2014), 1374–1394.

[111] Pan, Z., Xu, J., Guo, Y., Hu, Y., and Wang, G. Deep learning segmentation and classification for urban village using a worldview satellite image based on u-net. *Remote Sensing 12*, 10 (2020), 1574.

[112] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems 32* (2019), 8026–8037.

[113] Patterson, C., Koski, W., Pace, P., McLuckie, B., and Bird, D. M. Evaluation of an unmanned aircraft system for detecting surrogate caribou targets in labrador. *Journal of Unmanned Vehicle Systems 4*, 1 (2015), 53–69.

[114] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research 12* (2011), 2825–2830.

[115] Pettorelli, N., Laurance, W. F., O'Brien, T. G., Wegmann, M., Nagendra, H., and Turner, W. Satellite remote sensing for applied ecologists: Opportunities and challenges, aug 2014.

[116] Pham, M.-T., Courtrai, L., Friguet, C., Lefèvre, S., and Baussard, A. Yolo-fine: one-stage detector of small objects under various backgrounds in remote sensing images. *Remote Sensing 12*, 15 (2020), 2501.

[117] Phillips, R., Gales, R., Baker, G., Double, M., Favero, M., Quintana, F., Tasker, M., Weimerskirch, H., Uhart, M., and Wolfaardt, A. The conservation status and priorities for albatrosses and large petrels. *Biological Conservation 201* (sep 2016), 169–183.

[118] Ratcliffe, N., Guihen, D., Robst, J., Crofts, S., Stanworth, A., and Enderlein, P. A protocol for the aerial survey of penguin colonies using UAVs 1. *Journal of Unmanned Vehicle Systems 3*, 3 (sep 2015), 95–101.

[119] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 779–788.

[120] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Nips* (2015), 91–99.

[121] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497* (2015).

[122] Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 6 (jun 2017), 1137–1149.

[123] Rodrigues, F., and Pereira, F. C. Deep learning from crowds. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[124] Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Miccai* (2015), 234–241.

[125] Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[126] Schlossberg, S., Chase, M. J., and Griffin, C. R. Testing the accuracy of aerial surveys for large mammals: An experiment with African savanna elephants (Loxodonta Africana). *PLoS ONE 11*, 10 (oct 2016), e0164904.

[127] Schonberger, J. L., and Frahm, J.-M. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 4104–4113.

[128] Schwaller, M., Olson Jr, C., Ma, Z., Zhu, Z., and Dahmer, P. A remote sensing analysis of Adelie penguin rookeries. *Remote sensing of environment 28*, February (1989), 199–206.

[129] Schwaller, M. R., Benntnghoff, W. S., and Olson, C. E. Prospects for satellite remote sensing of adelie penguin rookeries. *International Journal of Remote Sensing 5*, 5 (sep 1984), 849–853a.

[130] Schwaller, M. R., Southwell, C. J., and Emmerson, L. M. Continental-scale mapping of Adélie penguin colonies from Landsat imagery. *Remote Sensing of Environment 139* (dec 2013), 353–364.

[131] Sethi, S. S., Ewers, R. M., Jones, N. S., Orme, C. D. L., and Picinali, L. Robust, real-time and autonomous monitoring of ecosystems with an open, low-cost, networked device. *Methods in Ecology and Evolution 9*, 12 (2018), 2383–2387.

[132] Shao, W., Kawakami, R., Yoshihashi, R., You, S., Kawase, H., and Naemura, T. Cattle detection and counting in uav images based on convolutional neural networks. *International Journal of Remote Sensing 41*, 1 (2020), 31–52.

[133] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations* (2015).

[134] Simpson, R., Page, K. R., and De Roure, D. Zooniverse: observing the world's largest citizen science platform. In *Proceedings of the 23rd international conference on world wide web* (2014), pp. 1049–1054.

[135] Spaan, D., Burke, C., McAree, O., Aureli, F., Rangel-Rivera, C. E., Hutschenreiter, A., Longmore, S. N., McWhirter, P. R., and Wich, S. A. Thermal infrared imaging from drones offers a major advance for spider monkey surveys. *Drones 3*, 2 (2019), 34.

[136] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research 15* (2014), 1929–1958.

[137] Stapleton, S., LaRue, M., Lecomte, N., Atkinson, S., Garshelis, D., Porter, C., and Atwood, T. Polar bears from space: Assessing satellite imagery as a tool to track arctic wildlife. *PLoS ONE 9*, 7 (jul 2014), e101513.

[138] Steitz, J.-M. O., Saeedan, F., and Roth, S. Multi-view x-ray r-cnn. In *German Conference on Pattern Recognition* (2018), Springer, pp. 153–168.

[139] Swinfield, T., Lindsell, J. A., Williams, J. V., Harrison, R. D., Gemita, E., Schönlieb, C. B., Coomes, D. A., et al. Accurate measurement of tropical forest canopy heights and aboveground carbon using structure from motion. *Remote Sensing 11*, 8 (2019), 928.

[140] Terletzky, P., and Ramsey, R. D. A semi-automated single day image differencing technique to identify animals in aerial imagery. *PLoS ONE 9*, 1 (jan 2014), e85239.

[141] Terletzky, P. A., and Ramsey, R. D. Comparison of Three Techniques to Identify and Count Individual Animals in Aerial Imagery. *Journal of Signal and Information Processing 7*, 7 (2016), 123–135.

[142] Trathan, P. N. Image analysis of color aerial photography to estimate penguin population size. *Wildlife Society Bulletin 32*, 2 (jun 2004), 332–343.

[143] Tuia, D., Kellenberger, B., Beery, S., Costelloe, B. R., Zuffi, S., Risse, B., Mathis, A., Mathis, M. W., van Langevelde, F., Burghardt, T., et al. Perspectives in machine learning for wildlife conservation. *Nature communications 13*, 1 (2022), 1–15.

[144] Turner, W. Sensing biodiversity. *Science 346*, 6207 (2014), 301–302.

[145] Van Andel, A. C., Wich, S. A., Boesch, C., Koh, L. P., Robbins, M. M., Kelly, J., and Kuehl, H. S. Locating chimpanzee nests and identifying fruiting trees with an unmanned aerial vehicle. *American journal of primatology 77*, 10 (2015), 1122–1134.

[146] Vas, E., Lescroel, A., Duriez, O., Boguszewski, G., and Gremillet, D. Approaching birds with drones: first experiments and ethical guidelines. *Biology Letters 11*, 2 (feb 2015), 20140754–20140754.

[147] Vermeulen, C., Lejeune, P., Lisein, J., Sawadogo, P., and Bouché, P. Unmanned Aerial Survey of Elephants. *PLoS ONE 8*, 2 (feb 2013), e54700.

[148] Wang, D., Shao, Q., and Yue, H. Surveying wild animals from satellites, manned aircraft and unmanned aerial systems (uass): A review. *Remote Sensing 11*, 11 (2019), 1308.

[149] Weimerskirch, H., Delord, K., Barbraud, C., Le Bouard, F., Ryan, P. G., Fretwell, P., and Marteau, C. Status and trends of albatrosses in the French Southern Territories, Western Indian Ocean. *Polar Biology 41*, 10 (oct 2018), 1963–1972.

[150] Weinstein, B. G. A computer vision for animal ecology. *Journal of Animal Ecology* (nov 2017).

[151] Weissensteiner, M. H., Poelstra, J. W., and Wolf, J. B. Low-budget ready-to-fly unmanned aerial vehicles: An effective tool for evaluating the nesting status of canopy-breeding bird species. *Journal of Avian Biology 46*, 4 (2015), 425–430.

[152] Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., and Reynolds, J. M. 'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology 179* (2012), 300–314.

[153] Westpac. The Little Ripper Life Saver | Search and Rescue Operations.

[154] Wich, S., Dellatore, D., Houghton, M., Ardi, R., and Koh, L. P. A preliminary assessment of using conservation drones for sumatran orang-utan (pongo abelii) distribution and density. *Journal of Unmanned Vehicle Systems 4*, 1 (2015), 45–52.

[155] Witmer, G. W. Wildlife population monitoring: Some practical considerations. *Wildlife Research 32*, 3 (jul 2005), 259–263.

[156] Wu, Z. Counting wildebeest from space using deep learning. Master's thesis, University of Twente, 2021.

[157] Xue, Y., Wang, T., and Skidmore, A. K. Automatic counting of large mammals from very high resolution panchromatic satellite imagery. *Remote Sensing 9*, 9 (aug 2017), 878.

[158] Yang, Z., Wang, T., Skidmore, A. K., De Leeuw, J., Said, M. Y., and Freer, J. Spotting East African mammals in open savannah from space. *PLoS ONE 9*, 12 (dec 2014), e115989.

[159] Yorkston, H. D., and Green, P. T. The breeding distribution and status of abbott's booby (sulidae: Papasula abbotti) on christmas island, indian ocean. *Biological Conservation 79*, 2-3 (1997), 293–301.