Improving the accuracy of weed species detection for robotic weed control in complex real-time environments

Alex Olsen
BEng (Electrical and Electronic) & BSc (Mathematics)

A thesis submitted for the degree of

Doctor of Philosophy

at the

College of Science and Engineering
James Cook University

2020

Supervisors:

Dr Peter Ridd

Physicist, author and former professor at James Cook University

Prof Ron White

Dean of the College of Science and Engineering at James Cook University

Dr Mostafa Rahimi Azghadi

Senior lecturer at the College of Science and Engineering at James Cook University

Dr Owen Kenny

Lecturer at the College of Science and Engineering at James Cook University

Acknowledgments

I would like to thank my primary supervisor, Dr. Peter Ridd, who gave me my first job, fostered my academic and professional development, and has been a guiding hand through some of the biggest professional and personal challenges in my life.

I owe deep gratitude to my team of secondary supervisors: Prof. Ron White, Dr. Mostafa Rahimi Azghadi and Dr. Owen Kenny, each of whom have served integral roles in the production of this thesis and the furtherance of my academic and professional career. Dr. Azghadi, in particular, provided unmatched motivation and support with his boundless professional curiosity and mentorship.

I gratefully acknowledge the funding received from the Australian Government Department of Agriculture and Water Resources Control Tools and Technologies for Established Pest Animals and Weeds Program (Grant No. 4-53KULEI) which allowed this work to flourish. I am also grateful for receiving an Australian Postgraduate Award, which made pursuing a PhD possible.

To my former colleagues at James Cook University's Marine Geophysics Laboratory, thank you for sustaining the most vibrant and supportive workplace a person could ever hope to be a part of.

Finally, I dedicate this work to my family; to whom I am indebted for their endless love and support. To my parents, Lance and Tricia. My brothers, Lee and Marcus. My sisters-in-law, Tami and Liz. My nephew, Oscar. My nieces, Lily and Audrey.

Contribution of Others

This research thesis is the foundation of a five-year collaborative project at the College of Science and Engineering at James Cook University with contributions and support from various people and funding bodies.

Supervision

- Dr Peter Ridd is the primary supervisor for this work. He contributed the originating concept for this thesis and helped shape the overall designs of the proof of concept in Chapter 2 and the prototype in Chapter 5. He also provided general academic guidance, reviewed associated publications and reviewed this thesis.

- Prof Ron White also served as the primary supervisor of this work during Peter Ridd's absence. He contributed general academic guidance, reviewed associated publications and reviewed this thesis. He also provided vital project and grant oversight for the project-wide collaboration.

- Dr Mostafa Rahimi Azghadi served as a secondary advisor for this work. He provided deep learning expertise contributing to the methodology of Chapters 3 and 4. He also guided me in the production of the thesis and provided general academic guidance, reviewed associated publications and reviewed this thesis.

- Dr Owen Kenny served as a secondary advisor for this work. He provided general academic guidance, reviewed associated publications and reviewed this thesis. He also supervised collaborating students who assisted with prototype development in Chapter 5.

Intellectual Support

The following people offered important intellectual contributions to this thesis. They have been grouped into tiers representing the extent of their contributions.

Key Contributors

- Jake Wood was a key contributor to this work as a staffed research worker and undergraduate engineering student at the College. He developed the positioning systems and assisted with the overall designs of the proof of concept developed in Chapter 2 and prototype developed in Chapter 5. He also led the team responsible for mechanically mounting electronics onto the prototype of Chapter 5.

- Brendan Calvert was also a key contributor to this work as a staffed research worker and undergraduate engineering student at the College. He contributed to the texture-based classification algorithm in Chapter 2, developed the spraying control unit for the prototype in Chapter 5, and contributed deep learning expertise to the methodology of Chapter 4, including development of the class activation heatmaps. He also assisted with the overall designs of the proof of concept in Chapter 2 and the prototype in Chapter 5.

Dataset Collection and Prototype Development

The following people contributed valued time and effort to the project as staffed research workers at the College.

- Benjamin Girgenti collected and labelled thousands of images in the DeepWeeds dataset of Chapter 3. He also provided field assistance during prototype development and trials, and contributed to the development of the labelling software presented in Chapter 3.

- Jamie Johns collected and labelled thousands of images in the DeepWeeds dataset of Chapter 3. He also assisted with curating the geographical positions of images in the dataset using ArcGIS in order to present the DeepWeeds geographical distribution in Chapter 3.

- Wesley Banks provided in-field assistance during trials of the proof-of-concept in Chapter 2. He also collected and labelled thousands of images in the DeepWeeds dataset of Chapter 3.

- Jakob Otten designed the 3D printed camera housings in the prototype of Chapter 4. He also was part of the team responsible for mechanically mounting electronics onto the prototype of Chapter 5.

- Billy-Ray Rittson was part of the team responsible for mechanically mounting electronics onto the prototype of Chapter 5.

Co-Authors

The following people contributed to the project as co-authors on the associated publications that form parts of this thesis.

- Dr James Whinney provided general academic guidance and consultation regarding the overall design concept for the proof of concept in Chapter 2 and prototype in Chapter 5.
- Dr Bronson Philippa provided general academic guidance and offered expertise in image analysis, dataset collection and deep learning for the work in Chapters 3 and 4.
- Dr Sunghyu Han contributed to the machine learning development of Chapter 2 using traditional artificial intelligence structures.
- Dr Dmitry Konovalov provided expertise contributing to the deep learning methodology presented in chapter 3.
- Corey Lammie contributed deep learning expertise and was the primary author for a journal article applying FPGA-accelerated binarised deep learning models to weed detection. This work is not presented in this thesis. However, the real-time performance benchmarks contributed in this thesis were inspired by Corey's work.
- Tony Carrick was a co-author for the aforementioned journal publication and contributed to the methodology used to measure power and speed of real-time deep learning algorithms.

Financial Support

I received three scholarships that supported my postgraduate tenure, namely, an Australian Postgraduate Award (APA) from the Australian Government for 3.5 years, a 1-year top-up scholarship and a Doctoral Completion Grant from the College of Science and Engineering at James Cook University.

This thesis project was the foundation of a two year $271,772 Australian Government grant under the "Control Tools and Technologies for Established Pest Animals and Weeds Programme" from July 2017 to June 2019. I was a chief investigator of the grant throughout its duration. Grant funding was used to support travel, equipment and research worker salary costs related to dataset collection and prototype development in this thesis.

Yamaha Motor Australia donated a Yamaha Grizzly 700 all-terrain vehicle to the project to serve as the platform for the robotic weed control prototype in Chapter 5. This represented an in-kind contribution valued at $20,000.

North West Local Land Services contributed $20,000 to accelerate the prototype development and support robotic weed control trials on Harrisia Cactus in Boggabilla NSW which are detailed in Chapter 5.

The Queensland Government Department of Agriculture and Fisheries contributed $10,000 to this project to support robotic weed control trials on Navua sedge in Malanda QLD which are detailed in Chapter 5.

The capstone achievement of this thesis, the AutoWeed prototype of Chapter 5, laid the foundation for a $50,000 grant project from July 2019 to June 2020 funded by round two of the Department of Agriculture and Water Resources' Smart Farms Small Grant. The project will deliver a bespoke AutoWeed detection system retro-fitted to a side-by-side utility-terrain-vehicle targeting Harrisia Cactus in Goondiwindi QLD.

Abstract

Weeds invade crops and smother rangeland pastures, consequently reducing the quality and quantity of Australia's agricultural production. It is estimated that weeds impose an average annual cost of $5 billion across Australia, with overall costs increasing by more than 20% from 1982 to 2018.

Due to these significant economic impacts, robotic weed control has seen increased research in the past decade with its potential for boosting the efficiency of weed control activities. This thesis reviews the landscape of robotic weed control, forecasting barriers to its widespread uptake and highlighting important areas of research that will benefit its development in the coming years.

Of the four core technologies at play – detection, mapping, guidance and control – the greatest obstacle to widespread uptake of robotic weed control is the robust classification of weed species in their natural environment. This thesis aims to advance the detection of weed species to be more broad and robust to environmental challenges, bolstering the efficacy of present and future robotic weed control solutions.

First, a case study is conducted to robotically control *Lantana camara* (lantana) with an automatic herbicide spot-spraying vehicle. A new image feature set is developed to detect the weed species from the colour, shape and texture of its leaves. A scale and rotation invariant enhancement of the Histograms of Oriented Gradients (HoG) feature set is proposed for texture based leaf classification. This simple yet powerful classifier achieves high accuracy (86.1%) on a test set of lantana with strong real-time performance. The detection model is then deployed in the field with the development of a towable robotic proof of concept. The proof of concept worked in certain situations, however failed due to small dataset size and imprecise motion sensing equipment.

Lessons learned from the proof of concept development and the recent successes of deep learning led to the next stage of the thesis, large scale dataset collection. The first large, public, multiclass image dataset of weed species from the Australian rangelands, *DeepWeeds*, has been published. A dataset collection and labelling tool, *WeedLogger*, was developed; which allowed for fast and simple dataset collection. The final dataset consists of 17,509 labelled images of eight nationally significant weed species. These images were taken at eight locations across northern Australia under natural conditions to emulate the

vision of a weed control robot. This dataset is of sufficient size to benchmark the latest advances in deep learning.

Next, to demonstrate the superiority of deep learning or Convolutional Neural Networks (CNN) over traditional image feature descriptors, the pioneering AlexNet CNN was deployed on the lantana dataset of our original case study. AlexNet achieves 88.4% classification accuracy compared to 86.1% of the HoG feature set. With better results and a streamlined development pipeline, deep learning is the ideal tool for weed species detection. Next, various state-of-the-art deep CNNs are re-architectured and implemented to benchmark their classification accuracy on the newly collected DeepWeeds dataset. AlexNet, Inception-v3, MobileNetV2, ResNet-50 and VGG16 deliver 86.2%, 95.1%, 91.6%, 95.7% and 92.3% top-1 categorical accuracy, respectively. ResNet-50, a deep residual network, offers the strongest classification performance.

The real-time performance of these architectures are then investigated for the application of robotic weed control. Inference time and power consumption are measured while deploying the five CNNs on four different NVIDIA hardware devices. The devices include a GeForce GTX 1080 GPU card and the three Jetson family GPU devices for mobile computing (Jetson Xavier, Jetson TX2 and Jetson Nano). The first of its kind benchmark comparison of the speed and power performance of the different models and architectures for weed species detection is presented.

Finally, a novel in-field prototype for robotic weed control, *AutoWeed*, is developed as the capstone achievement of this thesis. The robot consists of a Yamaha Grizzly 700 All Terrain Vehicle (ATV) mounted with detection, positioning and spraying subsystems. It employs a ResNet-50 architecture upon an NVIDIA Jetson TX2 module based on its classification accuracy and real time performance. In-field trials were conducted in five trial sites across the eastern coast of Australia where the weed control solution showed up to a 95% reduction in herbicide usage compared to traditional blanket spraying.

Thesis Conventions

The following conventions have been adopted in this thesis:

Typesetting

This document was compiled using the MiKTeX 2.9 free distribution of the LaTeX 2_ε system within the TeXstudio text editing interface. Adobe Illustrator and Adobe Photoshop were used to create and format figures.

Referencing

The IEEE (Institute of Electrical and Electronics Engineers) style has been adopted for referencing.

System of Units

The units stated in this thesis comply with the international system of units recommended in an Australian Standard: AS ISO 1000-1998 [1].

Spelling

Australian English spelling conventions have been followed for this work, as defined in the Macquarie English Dictionary [2].

Awards, Grants and Scholarships

2015

- Australian Postgraduate Award, The Australian Government
- CiSRA Extreme Digital Imaging Award, Canon Information Systems Research (CiSRA)

2016

- TropEco Research Award, James Cook University
- Bev Frangos Graduate Instructor Prize, James Cook University

2017

- Top-Up PhD Scholarship, Marine Geophysics Laboratory
- Bev Frangos Graduate Instructor Prize, James Cook University

2018

- Control Tools and Technologies for Established Pest Animals and Weeds Programme Grant, Department of Agriculture and Water Resources
- Sessional Staff Teaching Award, James Cook University

2019

- Doctoral Completion Grant, James Cook University
- Smart Farms Small Grant, Department of Agriculture and Water Resources

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recognising weeds within crops and pasture has been a challenging research topic for decades. Any method of site-specific robotic weed control, be it physical or chemical, hinges upon successful detection of the weed and rejection of its native background. This chapter introduces the economic and environmental impacts of weeds in Australia, followed by a summary of the existing control methods available. We then review related approaches to robotic weed control and identify improving weed species detection as the motivation for this thesis. This chapter then concludes with an overview of the contributions made by this thesis and outlines the structure to follow.

1.1 Weeds

A weed is considered to be a plant that requires some form of action to reduce its negative effects on the economy, the environment or the community. In Australia, over 3,000 species of plants have naturalised after being introduced from other countries – of those, about 500 species are classified as noxious or are under some form of legislative control as a weed [3]. Weeds negatively impact Australia's agricultural, horticultural and forestry products to the detriment of both industry and consumers; and their invasions change the natural diversity and balance of ecological communities.

1.1.1 Economic Impacts

Weeds invade crops, smother rangeland pastures and occasionally poison livestock. Consequently, they reduce the quality and quantity of Australia's agricultural production and require an ongoing strategic and coordinated response. In a 2012 survey conducted by Landcare Australia, weed and pest control was ranked as the most significant land management problem by nearly half of Australia's primary producers [4]. With overall costs increasing by more than 20% over the 16 years between these major studies, it is clear that, on a national scale, the economic impact of weeds continues to rise.

Three major studies have been undertaken in the literature to assess the economic impact of weeds on the national stage over the last three decades. Combellack's pioneering research in 1987 [5] first attempted to estimate the nationwide impact of weeds in general with data collected between 1981 and 1982. This innovative study accounted for weed control and losses in agriculture, and management of national parks, railway infrastructures, forestry establishments, aquatic areas and industrial buildings. The economic cost of weeds in 1982 was estimated to be $2.096 billion. As we compare studies it is worth noting that new methods of weed control and improved farm management practices have since been developed, and new weed species now occur.

The second such study [6], conducted in 2004, was the most thorough of its kind; including direct financial costs of control (such as herbicide and equipment), losses in production, changes in net revenue, changes in welfare and public expenditure (i.e. taxation levies) with an economic surplus approach. The study concluded the economic loss to the Australian agriculture sector to be at most $4.420 billion in 2002.

The third and most recent study [7] conducted nation-wide loss-expenditure and economic surplus approaches to estimate the weed control and production losses due to residual weeds in Australia. The study concluded that the annual cost of weeds in Australia is at most $5.845 billion.

The outcomes of these three studies can be compared to indicate the trend of increasing

financial costs (i.e. labour, machinery and chemicals) and opportunity costs (i.e. lost agricultural production) over time (Figure 1.1). It is evident that both the financial and opportunity costs are rising from decade to decade. The financial costs have sharply increased from 2002 to 2018, while the opportunity costs have risen steadily.

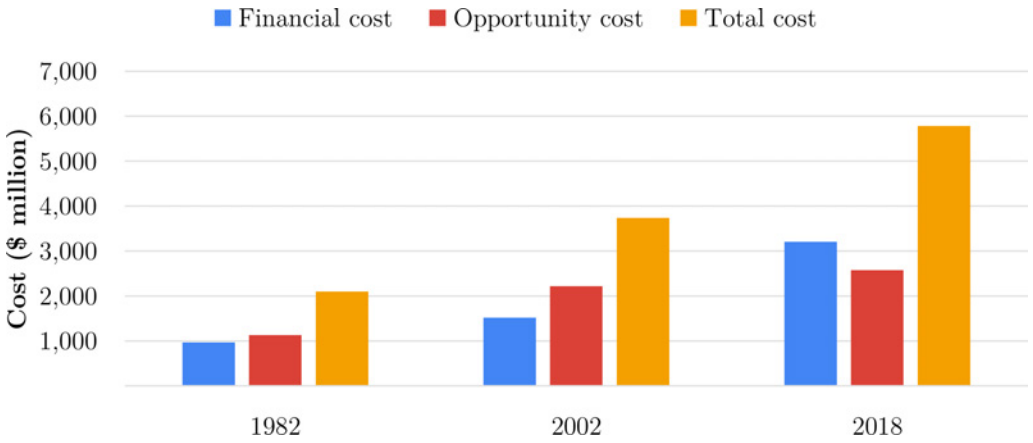


Figure 1.1: A comparison of three major studies estimating the national financial cost, opportunity cost and total cost attributable to Australian weeds over three decades from 1982 [5], 2002 [6] and 2018 [7].

It is also important to understand which industries within the agricultural sector have the highest weed management demand and cost. Here we compare the combined industry financial and opportunity costs associated with crop and pasture over the last three decades (Figure 1.2). In this comparison, the cropping industries include wheat, cotton, sugar, rice, fruit and vegetables, while pasture includes the combined cost of dairy and beef cattle, grain-fed livestock and sheep [5–7]. The overall costs steadily increase year upon year. The combined cost in cropping however has rapidly jumped in the past decade. This jump can be attributed to the large increase in chemical weed control costs for broadacre cropping enterprises [7]. Meanwhile, weed costs in pasture have steadily increased with the major contributing factor being production loss in the beef and wool industries.

With overall costs increasing by more than 20% over the 16 years between these major studies, it is clear that, on a national scale, the economic impact of weeds continues to rise. The Australian Government, recognising the significant burden of weeds on agriculture, has invested markedly to improve and develop new weed control tools, technologies, programs and strategies. As part of the Agricultural Competitiveness White Paper in 2015 [8], the Commonwealth invested $4 billion to agriculture, with a $100 million investment to provide farmers with new and better tools and control methods for managing established pest animals and weeds [9].

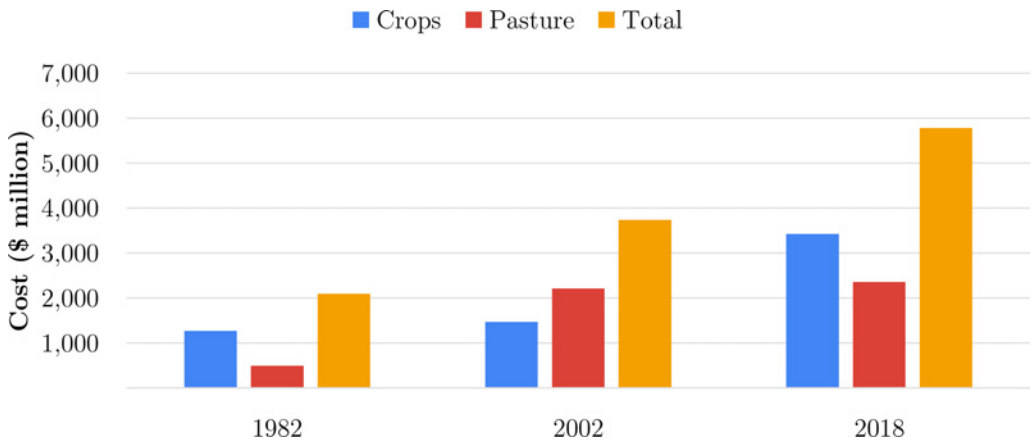


Figure 1.2: A comparison of three major studies estimating the national total cost attributable to weeds in Australian crops and rangelands over three decades from 1987 [5], 2004 [6] and 2018 [7].

1.1.2 Environmental Impacts

Weeds affect the structure and function of land and aquatic ecosystems and displace native fauna and flora [3]. Weeds can spread so vigorously that they increase the biomass of ecosystems leading to more intense bushfires, resulting in a drastic change to native vegetation. Many weeds are noxious and present a danger to livestock if eaten. Furthermore, the vast expanse of rangeland environments presents greater opportunity for insidious weed species to penetrate and reduce pasture levels. For example, Prickly acacia, dominates much of the western Queensland landscape strangling several million hectares of pasture in the Mitchell grass plains [10]. Once a landscape is subject to highly dense weed infestations, removal becomes even more costly and labour intensive requiring machine removal and re-vegetation to establish native pasture and trees.

In addition to these environmental impacts, weeds can have significant social impacts including: a loss of ecotourism opportunities, impacts on recreational activities and degradation of water quality.

Recognising these environmental and social impacts, the Australian Government remains committed to an Australian Weeds Strategy [11]; whose major focus is nationally coordinating a response to Weeds of National Significance. A list of twenty weed species was endorsed in 1999 [12] and a further twelve were added in 2012 [13]. Weed species' significance was assessed based on economic, environmental and social impacts, as well as the availability and cost-benefit of potential control measures [12]. Throughout this thesis, priority is given to addressing weed species that present as nationally significant; which are mostly weeds of rangeland pastures and not cropping.

1.1.3 Rangeland Weeds

The beef industry alone was found to have a financial loss of over $1 billion in 2002, equivalent to nearly 18% of the industry's $6.3 billion gross value that same year [6]. As of 2018, it is estimated that weed costs in the beef industry are now 5% of the national industry value [7]. By comparison, the cropping industry spends far more on herbicide control than the beef industry, however, the production losses due to weeds is comparable (Figure 1.3).

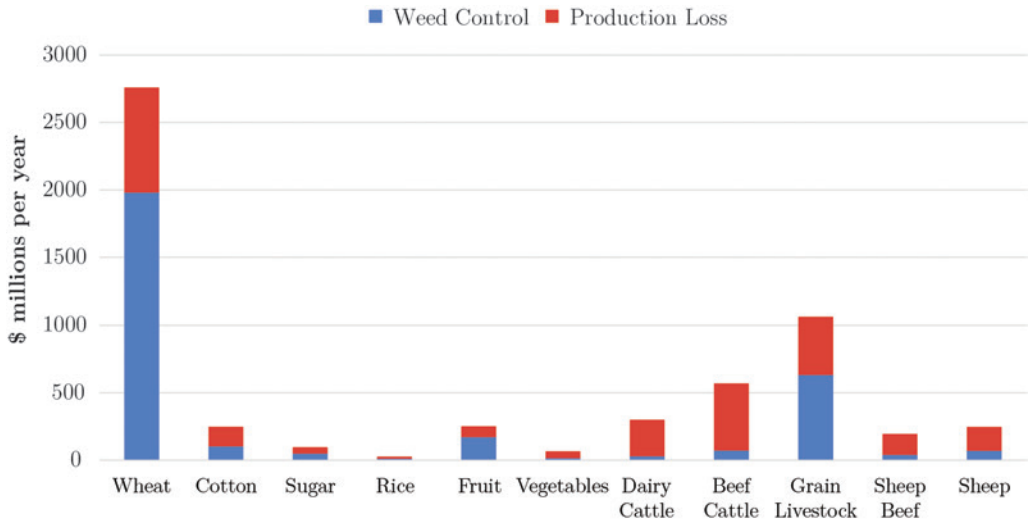


Figure 1.3: The loss versus expenditure estimate for the annual cost of weeds in Australia for 2018 across farming industries [7].

Rangelands occupy almost 70% of the Australian continent [15], as shown in Figure 1.4, and occur in all Australian states except Tasmania. Since 2004 the number of farms in broadacre production has decreased for most industries (Figure 1.5). This trend is significant in Figure 1.5 for cropping and mixed livestock; while beef producers have remained relatively stable. It has been suggested that this trend, particularly for cropping, is owed to farm consolidation and the adoption of new tillage systems [7].

The majority of weed control costs in Australia are spent in the crops. Because of this, past research into robotic weed control and weed species detection has predominantly focused on cropping applications. Meanwhile rangeland weeds and their associated costs – which have production losses on par with the cropping industry – are being ignored in the literature. It is imperative that weed control innovations stem to the Australian rangelands so that the environments and industries with the largest share of invasive weeds can reap the benefit of such technological innovation. Therefore, this thesis will focus on the weed species and environments relevant to Australian rangelands.

Figure 1.4: The extent of the rangelands and major population centres in Australia from 2007 [14].

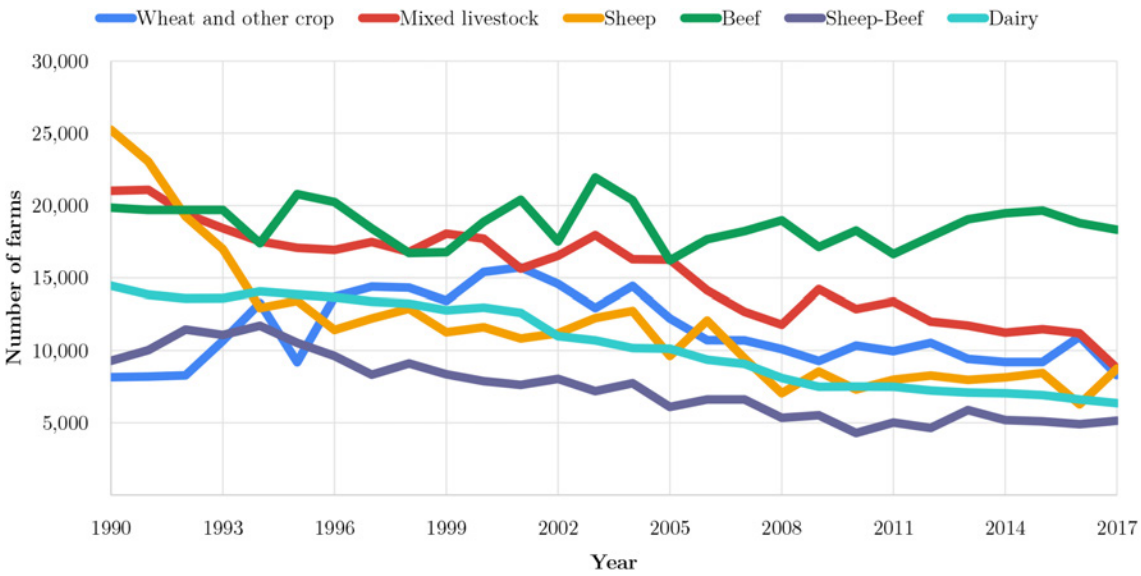


Figure 1.5: Number of Australian broadacre farms from 1990 to 2017 [16].

1.2 Weed Management

1.2.1 Existing Control Methods

The traditional control methods for all weed species can be categorised into three general groups: physical, chemical, and biological. Physical means of removal include: removal by hand (i.e. pulling, trimming, shoveling), removal by machine (i.e. chain pulling, grubbing, tillage) or removal by burning (i.e. fire or other thermal control methods). Heavy machine based control is often required for dense rangeland weed infestations. Mechanical control is incredibly effective but also time consuming and labour intensive.

Chemical control of weed species includes the application of herbicide to target weeds through a variety of approaches, including: foliar application, basal barking, cut stumping, misting or soil application of residual herbicides. Depending on the technique, these applications can be performed by hand, on-board a manned ground vehicle, by a boom spraying vehicle, or by aerial application for large infestations. Chemical control is very effective and relatively cheap compared to the equipment and labour costs of physical control. This has seen chemical control become a dominant approach in Australian agriculture [7].

Biological control of weeds has a long history of success globally [17]. Effective biological control sees the introduction of exotic insects or pathogens (disease) that have been scientifically proven to adversely affect weed health. Due to this introduction, biocontrol is subject to strict legislative control. The current utility of biological control in Australia is to assist in maintaining current populations of weed species at acceptably lower levels. To achieve this, a combination of biocontrol agents are commonly used [17].

Weed management approaches vary greatly depending on the target weed species and the target environment, be it crop or pasture. In rangeland pasture, dense weed infestations can be so large that mechanical means are predominantly used in the first instance, and herbicide control is deployed for follow up treatments.

Meanwhile in broadacre cropping, chemical application alone has proven to be a robust weed suppression strategy. This is due to a range of factors; including the ongoing introduction of new chemical labels that offer selective herbicide treatment of targeted weed species that do not affect neighbouring crops.

1.2.2 Integrated Weed Management

Unfortunately, there is no single control method for all management situations. It is typical to employ an integrated approach combining two or more control methods targeting vulnerable aspects of the weed at different points in its life cycle to achieve effective control [18]. This also helps to combat herbicide resistance which has become a serious

problem in Australia due to the reliance on herbicides alone for weed control.

An integrated weed management approach has proven successful in best combating weeds in both crop and pasture. This approach takes all aspects of the agricultural system into consideration when deploying weed control strategies including: knowledge of the critical period of weed interference, crop and pasture competitiveness, seed bank dynamics and alternative methods of weed control [19].

As such, this approach is greatly benefited by the availability of alternative methods of control. New robotic weed control techniques will present new opportunities to develop more dynamic integrated weed management solutions.

For example, currently there are limited selective herbicides that target a broadleaf weed in a broadleaf crop. With improved weed detection technology, it is possible to detect and apply knockdown herbicide directly to the broadleaf weed and leave the crop untouched. This new tool would therefore allow farmers to more regularly treat the paddock rather than relying on pre-emergent in-fallow herbicide application.

1.2.3 Industry Need

Farmers have access to a variety of weed control methods and expertise from regional landcare groups, Government agencies and agronomists. However, as technology continues to evolve, new avenues are presented to improve these existing control methods to make new weed control tools that are more efficient and less costly to implement

For pasture and rangeland situations, in some cases the burden of weed control is so great that ongoing, timely management is neglected. A robotic weed control technique that allows for faster, lower-cost control will make effective weed management in vast rangeland pastures more practical.

In cropping situations, innovations are continually proposed to achieve better yields in production. As a result, crops have already seen the delivery of robotic weed control techniques (such as the WEEDit [20] and WeedSeeker [21] near infra-red detection and spraying systems). However, these systems are only applicable in fallow crops and broad-acre farmers have desire for new and more accurate green-on-green detection and spray systems.

This thesis aims to improve the detection of site-specific weed species such that any newly developed weed control technique can be viable in both crop and pasture. Such technology would benefit existing weed management strategies, increase the likelihood of eradication, minimise the response cost and limit the impact on trade [8].

1.3 Robotic Weed Control

Agricultural robotics has seen increased development in the last two decades. Innovation in this arena has led to increased efficiency and profitability of agricultural practices [22, 23]. A variety of robotic platforms have been presented for the purpose of controlling weed species [20, 21, 24–28]. A sampling of these are illustrated in Figure 1.6.

(a) WEEDit [20]

(b) SwarmFarm [24]

(c) BoniRob [26]

(d) WeedSeeker2 [21]

(e) Weed Chipper [27]

(f) AgBot II [25]

(g) Lettuce Bot [28]

Figure 1.6: A sampling of some of the prominent robotic weed control approaches from the literature.

The WEEDit [20] and WeedSeeker [21] are two trailed spot-spraying solutions available on the market and have experienced strong uptake in the Australian agricultural industry. These two weed detection units exploit the spectral absorption properties of plants as the basis for detection. They emit a laser light onto a field of view within which any living plant reacts photosynthetically and its response is sensed by the detection unit. As a result, any detected plants are sprayed with a herbicide treatment. These devices provide high weed detection accuracy in real time environments. However, they can only be applied in non-crop situations (i.e. *green-on-brown*), where the weeds are the only plants present. Therefore, there exists a need for *green-on-green* detection in cropping situations. Similarly, these products are not suitable in the pastoral rangeland environments for follow up control of lantana and other rangeland weed species.

The Queensland based SwarmFarm Robotics [24] have focused on the development of a fleet of small cooperative robots to perform a variety of crop-based tasks. Their robotic platforms exhibit state-of-the-art collision avoidance, navigation and swarm technology. However, their focus is on providing the base platform, and not the weed detection technology itself. Similarly, the BoniRob autonomous platform developed by Bosch has served as a base platform for researchers to trial different crop-based weed detection algorithms [26]. The AgBot is another autonomous platform that navigates its terrain requiring knowledge of the crop lines. Displacing this robot from a crop-like environment would render it unusable.

The University of Western Australia and University of Sydney have developed a mechanical control technique for targeted tillage [27]. A rapid response tyne system was developed to cultivate target weeds in a field when weeds are present at densities of up to one plant per 10 m^2 at a travelling speed of 10 km/hr. The study however focused on the significant mechanical component of the system and utilised off-the-shelf WEEDit [20] spectral detection units. As a result, the control technique is only applicable in fallow situations.

One robotic solution that accomplishes green-on-green weed detection is the Lettuce Bot [29] from Blue River Technology (now a subsidiary of John Deere & Company [28]). Its pioneering use of deep learning and big data allows it to differentiate between lettuce crops and invasive weeds using a very large dataset of millions of images from the in situ crop [29]. The lettuce bot however, was able to significantly control the camera field of view in the cropping situation by shrouding the entire detection system in what was essentially a *dark room*. This allowed homogeneous image dataset collection, greatly reducing the variability of the learning problem to increase classification accuracy. However, this luxury is not afforded in applications with unsuitable terrain, foliage and operating conditions – such as the rangeland environment. A large enclosed detection system would not allow the

robotic vehicle to traverse the rangeland terrain. Furthermore, the lettuce bot is restricted greatly in speed of operation due to the size of the vehicle. In a cropping situation, higher speeds are desired from farmers (up to 20 km/hr) for efficient management.

Research in robotic weed control has focused on what many consider to be the four core technologies: detection, mapping, guidance and control [22]. The aforementioned laudable research endeavours have greatly advanced the state of guidance and control. Meanwhile, mapping of weed infestations has exploded with the use of unmanned aerial vehicles, spectral and hyperspectral imaging and Internet of Things technologies. The robust detection of weed species in various complex real-time environments however, remains an obstacle to commercial uptake of robotic weed control technologies.

1.4 Motivation

Effective weed control necessitates continual oversight and action. As such, demands of time, cost, and labor have wrought ceaseless innovation in the automation of this agricultural practice. The literature boasts many approaches for classifying weeds in the visible spectrum [30–38]. Collectively, these approaches discriminate too broadly between species, cannot be applied in the field, and fail to meet a real-time requirement. This thesis resolves to automate real-time and in-field detection of various weed species in the visible light spectrum. To ensure the developed detection models' performance are upheld in the real-world, it is proposed to develop a robotic platform in order to trial the detection systems in the field on select weed species. This work aims to advance the detection of weed species to be more broad and robust to environmental challenges, bolstering the efficacy of present and future robotic weed control solutions.

1.4.1 Research Gaps

The robotic weed control techniques reviewed earlier either fail to provide a robotic vehicle that can be used in non-crop applications; do not have image recognition methods that can be applied in real-time at high speeds, fail to accurately discriminate visually similar weed species with complex floral backgrounds, and have not addressed weed species significant to the Australian national landscape.

Furthermore, contemporary startup agricultural companies have recognised this research gap as of 2019 (including Bilberry [39] and Blue River Technologies [28]) and have begun to develop competing green-on-green weed detection systems targeted for a crop environment using deep learning and machine vision. This thesis separates itself from such contemporary works by proposing to develop detection systems that are applicable to both crop

and pasture weed species and that can be retro-fitted to existing agricultural machinery, providing a lower-cost and more customisable solution for Australian landholders.

1.4.2 Research Questions

The research questions addressed in this thesis include:

1. Can weed species be detected with enough accuracy in the visible spectrum to make real-time robotic weed control a viable solution?

2. What is the best method for classifying weed species based on accuracy and real-time speed between traditional image analysis with handcrafted features, or modern deep learning application with automatically learned features?

3. Can modern deep learning architectures be implemented with fast enough real-time speed for deployment in robotic weed control applications where vehicles operate at up to 20 km/hr?

4. What are the most suited state-of-the-art deep learning architectures and edge computing devices for a robotic weed control prototype?

5. Is robotic weed detection and spot spraying a more efficient tool than traditional blanket spraying in rangeland pasture systems?

1.5 Overview

1.5.1 Contributions

The original contributions of this thesis can be itemised as follows:

- The development of a proof of concept detection and spraying system for foliar spot-spraying of lantana regrowth in a rangeland pasture situation. This included the introduction of a small in situ lantana dataset of 337 images, as well as the development of a novel texture-based classifier utilising the Histograms of Oriented Gradients feature set, achieving 86.1% detection accuracy. The proof of concept positioning system allowed for accurate straight line spraying. This contribution led to a successful bid for a two-year $298,950 Government grant that funded the future development and research in this thesis.

- A custom designed image acquisition and dataset collection instrument, *WeedLogger*, was developed to allow rapid collection of images matching the optical system of a

prototype weed control robot. The system allowed the collection of *DeepWeeds*, the first, large, multiclass weed species image dataset including 17,509 images of eight nationally significant weed species collected from eight rangeland locations across northern Australia. The dataset is shared publicly to foster more research into detection of rangeland Australian weed species.

- State-of-the-art weed species detection in the visible spectrum is presented using the latest deep learning Convolutional Neural Network (CNN) architectures. Benchmark performance on the DeepWeeds dataset is presented for AlexNet, Inception-v3, ResNet-50, MobileNetV2 and VGG16 achieving 86.2%, 95.1%, 91.6%, 95.7% and 92.3% accuracy, respectively. Furthermore, the power of deep learning in comparison to traditional handcrafted feature analysis is demonstrated with a revised AlexNet architecture achieving 88.4% classification accuracy of lantana compared to the earlier HoG based contribution.

- The real-time performance of the aforementioned selection of deep learning architectures is investigated thoroughly on a variety of the latest edge computing devices offered by NVIDIA. This is the first benchmark of inference speed and power consumption on the NVIDIA family of devices for real-time weed species detection. The benchmark reveals AlexNet and MobileNetV2 to be the fastest and least power hungry models during inference. However, ResNet-50 presents the ideal trade-off between detection accuracy and real-time performance.

- Finally, a novel weed detection and spraying prototype, *AutoWeed*, retro-fitted to an all-terrain vehicle is presented here as the capstone achievement of the thesis. The prototype includes off-the-shelf positioning equipment, custom designed software and the aforementioned real-time implementations of state-of-the-art deep learning architectures to deliver fast and accurate weed species detection. The prototype achieves approximately 90% successful weed classification with 5% false detection and 5% non-detections while providing up to 95% herbicide reduction compared to traditional blanket spraying.

1.5.2 Publications

The following is a list of four publications specific to this work, including the chapters to which they pertain. For the two works of which I was primary author, the ideas within the publications are my own as I performed the writing, experiments and content generation of all material within. For the two works I co-authored; I contributed writing, software and

hardware development, dataset collection and experiment development concurrent with the published work.

Conference Papers

Chapter 2: ***A. Olsen***, S. Han, B. Calvert, P. Ridd and O. Kenny, "In situ leaf classification using histograms of oriented gradients," in *Proceedings of the 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, (Adelaide, Australia), pp. 441-448, 2015.

Chapters 4 and 5: B. Calvert, ***A. Olsen***, B. Philippa, M. Rahimi Azghadi, "AutoWeed: Detecting Harrisia cactus in the Goondiwindi region for selective spot-spraying," in *Proceedings of the 1st Queensland Pest Animal and Weed Symposium (PAWS)*, (Gold Coast, Australia), May 2019.

Journal Papers

Chapters 3 and 4: ***A. Olsen***, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, B. Calvert, M. Rahimi Azghadi, and R. D. White, "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning," *Scientific Reports*, vol. 9, pp. 2500-2512, February 2019.

Chapter 4: C. Lammie, ***A. Olsen***, T. Carrick and M. Rahimi Azghadi, "Low-Power and High-Speed Deep FPGA Inference Engines for Weed Classification at the Edge," *IEEE Access*, vol. 7, pp. 51171-51184, 2019.

1.5.3 Thesis Outline

This thesis encompasses six chapters, as illustrated in the outline of Figure 1.7. In the current chapter, some introductory remarks on existing weed management techniques and related work in robotic weed control research were addressed. Furthermore, we described the motivations for this thesis for improving weed detection systems in complex crop and pasture environments to develop viable robotic weed control prototypes.

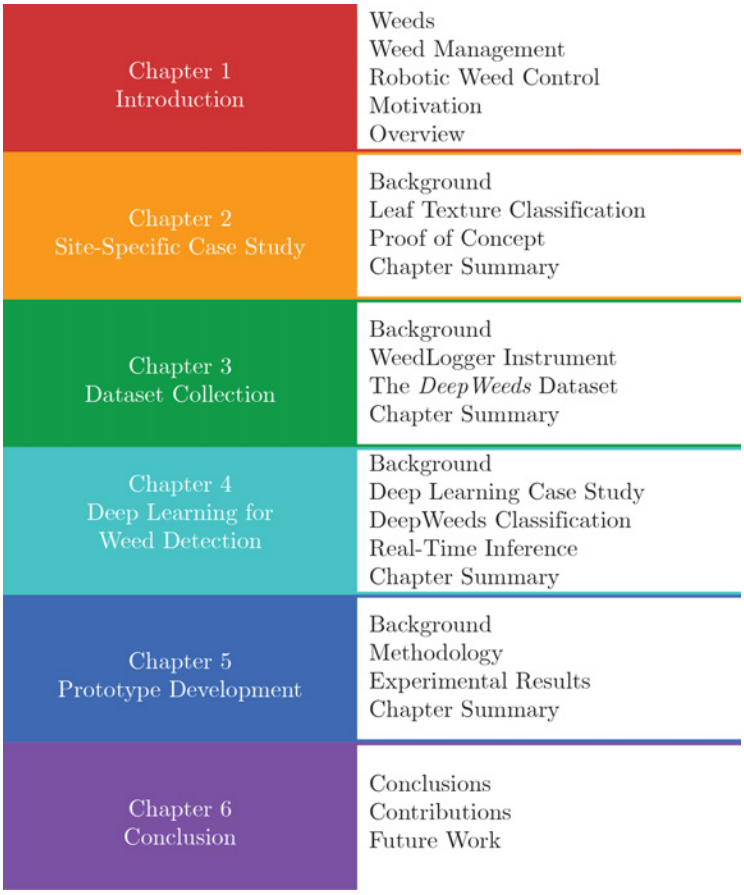


Figure 1.7: A graphical overview of the structure of this thesis.

Chapter 2 begins with a case study focusing on controlling *Lantana camara* with robotic foliar spot spraying, which would be more efficient than traditional hand gunning by a manual operator. This chapter provides a review of weed species detection methods in the literature and chooses to focus on handcrafted traditional image features, specifically a novel texture-based leaf classifier. The chapter delivers high accuracy lantana detection on an in situ dataset then proceeds to develop a proof of concept robotic system to field test the detection algorithms. A simple proof of concept is developed with some limitations in its positioning system that restrict it to only be proficient in straight line motion. Chapter 2 identifies a variety of improvements to be made to the prototype which lay the foundation for subsequent chapters. These improvements include, collecting a larger image dataset for better in-field detection performance, purchasing more sophisticated motion sensing equipment and using a self-propelled base platform for a prototype. Furthermore, the proof of concept developed in Chapter 2 was the focus of a successful bid for a two-year $298,950 Government grant, which funded the work in subsequent chapters to pursue the

aforementioned improvements.

Chapter 3 focuses on improving in-field weed species detection by collecting a large, multiclass weed species image dataset of 17,509 images of eight weed species in eight different locations in northern Australia. The DeepWeeds dataset is the first of its kind and was made publicly available to facilitate collaborative research into rangeland weed species detection. This chapter details how the dataset was collected to mitigate sources of error and match the target application for robotic weed control as much as possible. A dataset collection tool and labelling software was developed for this chapter to enable fast and consistent dataset collection. The key result of this chapter is the DeepWeeds dataset, which then allows for the implementation of deep learning architectures in Chapter 4.

Chapter 4 is dedicated to deploying the dominant tool of modern computer vision, deep learning with convolutional neural networks, to the task of weed species detection. The chapter begins with a case study deploying the first deep CNN, AlexNet, on the lantana dataset from Chapter 2. AlexNet beats the texture-based HoG classifier of Chapter 2 with 88.4% classification of lantana. The chapter then benchmarks performance of several modern deep CNN architectures on the DeepWeeds dataset. ResNet-50 offers the strongest classification accuracy with 95.7%. The chapter then examines the real-time performance of these deep learning architectures when deployed on various edge computing devices. The chapter presents an extensive benchmark of real-time inference speeds and power consumption statistics for a variety of deep CNN architectures on the Jetson family of edge computing devices with various batch sizes. This benchmark may serve as a valuable look up table for researchers interested in deploying weed species detection systems in the field. The chapter concludes with recommendations for which software and hardware architectures are most suited for the full-scale prototype development in the following chapter.

Chapter 5 takes on the recommendations from Chapter 4 for the purpose of developing a robust weed detection and spot-spraying system for both crop and pasture retro-fitted to a manned all-terrain vehicle. The chapter begins by reviewing related works in robotic weed control and establishing the aims and performance targets for the proposed system. A detailed look at the methodology behind the detection, positioning and spraying subsystems are proposed. The chapter then evaluates the completed *AutoWeed* prototype with several field trials across the east coast of Australia. The chapter concludes by discussing some of the limitations of the prototype, how it can be improved and commercialised for uptake by Australian land managers.

Chapter 6 provides concluding remarks of this thesis. It also discusses future research directions in deep learning and dataset collection that may see significant commercial uptake of this robotic weed control technology in the near future.

Chapter 2

A Site-Specific Case Study

The jumping-off point for this research thesis considers a site-specific case study on automatically detecting and controlling *Lantana camara* in its natural environment. This chapter first introduces lantana and the threat it poses to Australian rangeland environments. A detection algorithm is then developed to classify lantana from the texture of its leaves. We present a rotation and scale invariant innovation of the Histograms of Oriented Gradients feature set, combined with machine learning, to detect the weed from an in situ dataset with 86.1% accuracy. This lab-based accuracy is then implemented in real-time on a proof of concept robotic weed control platform. The platform consists of a towable trailer mounted with cameras, sprayers and low-cost position sensors. The proof of concept is moderately successful and highlights the need for improvements which are addressed in subsequent chapters.

2.1 Background

2.1.1 Lantana Camara

Description

There are a variety of common names used for *Lantana camara L.* (lantana) in Australia, such as Lantana, Common lantana, Kamara lantana and Wild sage [40]. *Lantana camara* (lantana) plants are multi-branched shrubs that generally grow from 2-4 m in height [41]. This woody weed has invaded more than five million ha throughout most coastal and subcoastal areas of eastern Australia, from the Torres Strait islands to southern New South Wales [42]. It grows in a wide variety of habitats, from exposed dry hillsides to wet, heavily shaded gullies. It forms dense thickets that smother and kill native vegetation and are impenetrable to animals, people and vehicles. Lantana leaves (as shown in Figure 2.1) are $2-10$ cm long, with toothed contours and a bright green and coarsely textured surface. The leaves grow opposite one another along the stems, and the plant size and shape depends on the availability of light and moisture [18].

Figure 2.1: Pink flower type lantana exhibiting large green leaves with a velvety texture that is visibly discernible amongst neighbouring flora.

The aggregate species known as *L. camara* contains a wide diversity of varieties arising from horticultural and natural hybridisation, selection and somatic mutation [41]. In 1982, there were at least 29 different varieties of *L. camara* that had naturalised in Australia, and it is highly likely that this number has been far exceeded today [43]. Varieties can be partially differentiated by flower colour, but complete differentiation relies on further detailed analysis of flower size, leaf shape and colour and stem characteristics. Four major groups of *L. camara* are distinguished as red, pink, white, orange flowered varieties. It is therefore important to categorise any developed recognition algorithms within the subspecies of lantana being targeted. This case study is primarily focused on pink flower type lantana that has naturalised in North Queensland.

Environmental Impact

Lantana is classed as a weed in more than 60 countries and is considered to be one of the ten worst weeds worldwide [44, 45]. The seeds of lantana are spread by fruit eating birds and mammals after ingestion. A large number of native and exotic birds have been recorded as feeding on lantana fruits in Australia [41]. The weed infests millions of hectares of grazing land globally and is of serious concern in 14 major crops including cotton and sugarcane. The invasion of natural ecosystems by lantana puts at risk populations of more than 1,400 native species, including 279 plant and 93 animal species listed as rare and/or threatened under state and federal legislation [46]. In Queensland alone, approximately 107,000 ha of endangered regional ecosystems contain lantana [47]. Where infestations are dense, lantana kills native species via smothering and allelopathic effects [48]. It dominates understoreys, prolongs succession and reduces biodiversity [44, 49].

Economic Impact

Lantana has been classified as a Weed of National Significance in Australia since 2000 [12] partly due to its significant impact on pastoral production. Lantana costs the Australian grazing industry more than $104 million in lost production, and $17 million in management efforts each year, as of 2006 [50]. The majority of these costs are associated with reduced livestock-carrying capacity, increased property maintenance expenditure and stock poisoning [51]. These significant costs and its proximal impact in Queensland make it a focus of this work.

Control

Lantana is so well established along the east coast of Australia that the highest priority and most cost-effective approach for lantana management is to prevent its spread into uninfested areas [51]. Part of the national plan for prevention includes strategically controlling infestations that threaten areas where lantana is not yet a significant weed [52]. As with all weed management plans, an integrated approach is the best practice, using the full range of methods such as: herbicides, mechanical removal, fire, biological control and revegetation. There are numerous existing lantana control methods using herbicide and mechanical removal (Figure 2.2).

For plants less than 2 m high, foliar herbicide spraying is very effective. For larger plants, more intrusive herbicide application is required by cutting the stump or covering the bark. For large, dense infestations, mechanical or physical control is required with stickraking, bulldozing, ploughing or grubbing. Fire is also recommended prior to introducing mechanical or herbicide controls [18].

Weed recognition technology has the potential to significantly benefit management strategies for lantana in Australia. Specifically, this technology can provide a new robotic spot-spraying technique to target lantana regrowth following the reduction of dense woody weed infestations using mechanical or physical control.

Figure 2.2: Left: Foliar spot-spraying of a small lantana infestation. Right: Mechanical removal of dense lantana infestation using an excavator.

2.1.2 Proof of Concept

Robotic Control

The research presented in this chapter proposes to automate the herbicidal spot-spraying of lantana as part of an integrated management approach. Three follow up sprays are recommended after the initial control effort in a dense lantana infestation [52]. By making follow up control more efficient and effective, the effort to control a lantana infestation is greatly reduced and the probability of success is improved. This spot-spraying approach is also applicable to a variety of other weeds in other stages of weed management; making it a more worthwhile endeavour. The proposed robotic solution will therefore need to operate in complex rangeland environments with terrain that is difficult to navigate in comparison to a crop situation. This environment and the real-time considerations will impose certain requirements on the proof of concept solution.

Scope

The development of weed recognition algorithms for herbicidal spot-spraying of weed species has seen commercial and research interest for over a decade [22, 23]. A need exists for algorithms that can detect weeds among a variety of backgrounds to be applicable for both crop and pasture environments. This type of recognition technology is also beneficial to the development of new alternate control methods. But by focusing on the herbicide spot-spraying control method, we can drastically simplify the mechanical

and robotic requirements of the proof of concept. For example, control will only require electronic activation of a solenoid with knowledge of the vehicle's position in order to accurately spray a target. Automating spray activation is far simpler than automating more complex mechanical control methods. This allows us to precisely target our scope to demonstrating the efficacy of the weed detection algorithms when operating in the field.

2.1.3 Image Analysis

Early applications of leaf image classification predominantly used traditional hand-crafted features to describe leaf shape, colour and texture; while later approaches have utilised deep learning or deep convolutional neural networks for automatic feature extraction [53]. For this chapter, we focus on the traditional approach to classification which has the benefit of being less computationally expensive and thus does not require high-performance GPU implementation for our proof of concept.

Leaf Image Features

Many methods have been proposed to discriminate plant species using their leaf image features. Most of these methods have involved the analysis of leaf shape [31,54,55], texture [56–58], venation patterns [59,60], or a combination of the three [30,61,62]. A review of this literature shows that leaf image feature selection is dependent upon the target application – specifically the trade-off between classification accuracy and computational performance, and the characteristics of the target image dataset.

For example, Kumar et al. [31] uses the distinctive shapes of leaves as the sole visual cue by extracting histograms of curvature from a binarised image at multiple scales. The fruit of their labour is *LeafSnap*, a mobile phone application that helps users identify trees from photographs of their leaves. Choosing to discriminate by shape stemmed from the limited imaging capabilities in mobile phone cameras. And yet for reliable extraction of leaf image shape they require users to photograph leaves against a light, non-textured background – which is not sufficient for in situ classification.

After leaf shape, the next most studied leaf feature is their vein structure (or venation). Fu and Chi [60] introduced an accurate approach that combined a thresholding method and an artificial neural network classifier to extract leaf vein data. However, their experiments required venation enhancement of leaf images through high resolution photography using a fluorescent light bank. Obtaining such images is not practical for in situ classification.

Perhaps the most neglected feature in the literature has been leaf texture. Bruno et al. [56] and Backes et al. [57] have applied fractal dimensions to plant classification using leaf texture with promising degrees of success. However, such methods lag in computational

performance and are not realisable for real time application. Cope et al. [58] achieved an 85% identification rate on 32 species of *Quercus* using the co-ocurrences of different scale Gabor filters. However, their datasets required high resolution leaf images taken on non-textured backgrounds.

Many studies have discriminated plant species using a combination of leaf image features. [30] employed a probabilistic neural network with 12 geometric and morphological leaf features to implement an automated leaf recognition system for plant classification. This study introduced the *Flavia* dataset, consisting of 1,907 leaves from 32 different species, upon which 90% inter-species classification was achieved. However, the dataset is photographed against a white laboratory background in high resolution which is not suitable for real time, in situ classification. Nevertheless, the dataset does provide a means of validation for many new approaches in plant leaf classification – as will be done here.

The focus of this chapter is the real-time detection of in situ weed species based on leaf recognition. As such, a reliably attainable leaf image feature set is required that can discriminate accurately and with low computational complexity. With almost all leaf specimen possessing well-preserved and relatively easily imaged leaf material, we propose a texture based leaf recognition algorithm. However, prior to texture extraction we require a method of segmenting leaf regions from a complex in situ background.

Image Segmentation

Image segmentation is a staple of all traditional object recognition algorithms. It requires the partitioning of a digital image into multiple segments, and obtaining a binary image separating the target objects from the background. Leaf area segmentation is particularly challenging owing to shadows, blur, reflections, and the foreign obstructions and textured backgrounds present among in situ leaf images. The majority of the studies reviewed here [30, 31, 55, 60–62] segment leaves from simple, non-textured backgrounds using RGB and HSV thresholding. This however is not applicable to in situ images due to complicated floral backgrounds, overlapping leaves and other interferences.

This chapter demonstrates a robust image segmentation approach, founded in first principle image analysis techniques, combining selective RGB thresholding and binary morphological processing to segment lantana from a complicated in situ background.

Feature Extraction

Much of the texture present on a leaf is due to its venation, with other sources of texture including hairs, stomata and glands. Presuming leaf texture is dominated by the venation, certain species of leaves should have a similarly discernible pattern for discrimination.

With this in mind, we propose the recognition of repeatable leaf textures using Histograms of Oriented Gradients (HoG). HoG descriptors have been used for a variety of applications, most notably in human detection [63, 64] and face recognition [65]. The method is based on the notion that local object appearance and shape can often be characterised rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions [63]. The computation of the HoG feature set requires parametric tuning. To find the optimal parameter values for discriminating leaf texture, a validation study must be conducted similar to [63].

Unfortunately the HoG feature set is not invariant to the rotation and scale changes associated with in situ leaf image datasets. In order to achieve invariance to rotation we propose a dominant rotation alignment of the histogram descriptors, similar to that featured in Lowe's Scale-invariant Feature Transform (SIFT) [66].

To account for scale invariance Déniz et al. [65] proposed the fusion of HoG descriptors at different scales to capture important scale-space structure for face recognition. Their fusion strategy involves the product combination of the classifiers at different patch sizes. However this approach assumes statistical independence of the classifiers and knowledge of the prior probabilities of each decision class. In lieu of these conditions, we propose an alternative solution that ignores prior probability by averaging the classifiers across the different patch sizes.

Machine Learning

The machine learning approach utilises an algorithm to optimise a quantifiable representation of the extracted image features to discriminate between labelled classes. There are a tremendous number of different machine learning algorithms – from logistic regression (LR) to convolutional neural networks (CNNs), as illustrated in Figure 2.3.

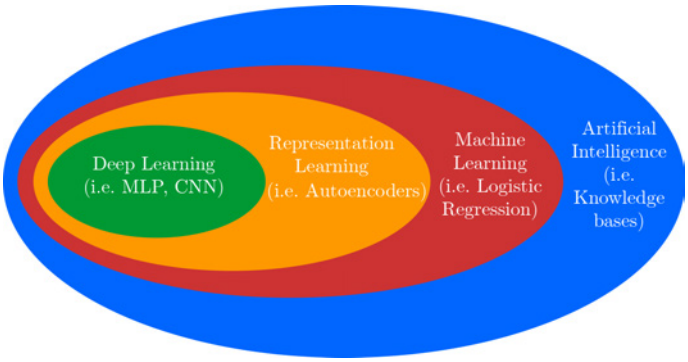


Figure 2.3: Venn diagram illustrating where traditional machine learning methods like LR and ANNs fit within the wider artificial intelligence realm compared to deep learning and CNNs [67].

CNNs are the latest and most advanced form of machine learning, which automatically extract the salient image features by learning the coefficients of a series of convolutional filters applied to images in the training dataset. This is not to be confused with the earlier and simpler neural network implementation of artificial neural networks (ANNs), which take a set of input features and learn the weight coefficients of the input features to fit to the given dataset. This difference in algorithmic approach is illustrated in Figure 2.4.

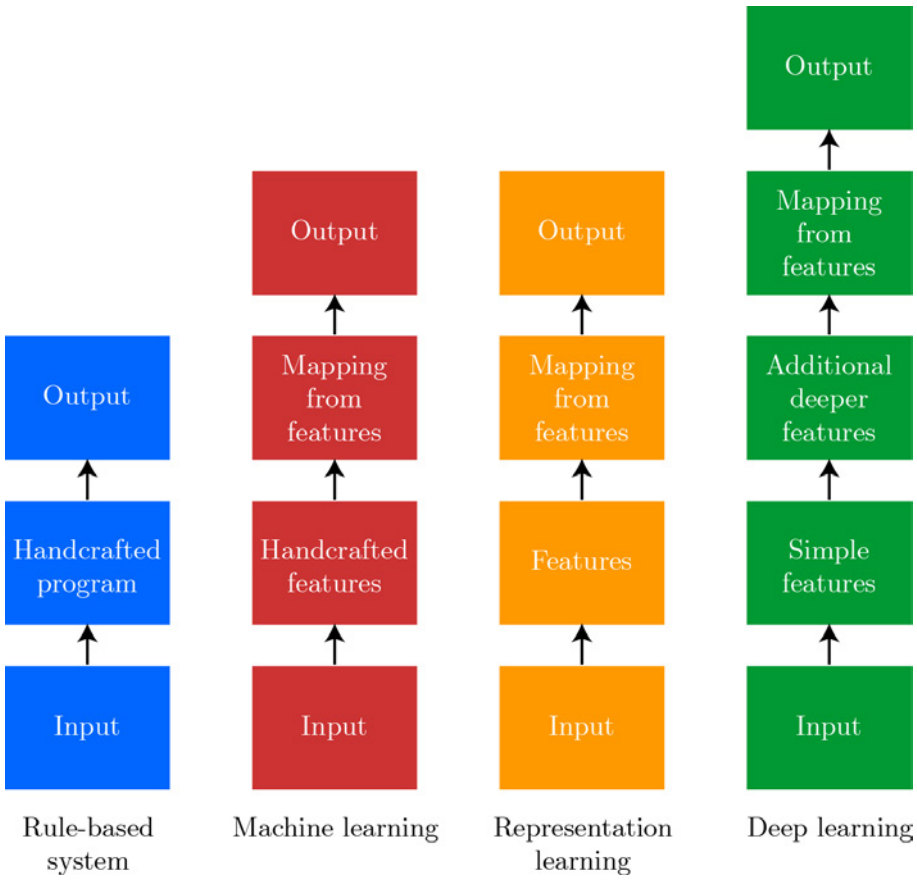


Figure 2.4: A comparison of the detection frameworks for different artificial intelligence algorithms including traditional machine learning versus deep learning [67].

A variety of machine learning algorithms have proven successful for the classification of plant species using image features. The choice of technique depends upon the nature of the classification task. A majority of cases consider multi-class classification of plant species via their leaves [30, 31, 57, 61]. For this chapter, we wish to detect the presence of one plant species by its leaf type amongst neighbouring flora in its natural state. As a result we require methods suitable for learning binary classifiers. We have selected four such machine learning approaches – logistic regression (LR) [68], linear support vector

machines (LSVM) [69], Gaussian support vector machines (GSVM) [69] and artificial neural networks (ANN) [70].

ANNs have been used extensively in the classification of plants using their leaf image features [30,60,62]. While LSVM and GSVM have been used in various pattern recognition applications [63, 71]. Here we will ignore CNNs since we are using hand-crafted image features and require machine learning algorithms that map the feature vectors to their class labels.

2.2 Leaf Texture Classification

An overview of the proposed methodology is provided in Figure 2.5. Variants of this three step framework are commonly used in object recognition.

Figure 2.5: An overview of our leaf recognition algorithm.

We propose the use of leaf texture and venation pattern as the sole visual cues for in situ recognition of specific plant species. Other leaf features are not suitable in this application – contour and shape features are not considered reliable due to highly textured backgrounds, overlapping leaves and obstructing neighbouring flora, while colour features are inconsistent due to variable scene illuminance. Leaf texture however is reliably extractable and highly variable among species – especially for lantana.

A simple but powerful approach is presented for recognising leaf texture using the HoG feature set. These features are histograms of image gradient orientations computed on a dense grid of uniformly spaced cells and normalised for invariance to illumination [63]. In order for the HoG descriptors to be robust to variations in rotation and scale, we propose an enhancement of the HoG feature set via dominant rotation alignment [66] and multiple scale histogram fusion [65].

HoG features are extracted from pertinent leaf image regions segmented from a textured background. These features serve as the input to a two-stage binary classification algorithm which predicts the presence of the target leaf within an image. Several machine learning methods are investigated along with varied implementations of the HoG descriptors in order to find the ideal discriminators for leaf texture.

For this chapter, unique image segmentation and feature extraction software was developed in the C# programming language using Microsoft Visual Studio and Emgu CV (a

C# *wrapper* to OpenCV [72] – an open source C++ image processing library). MATLAB's machine learning implementations were also used to classify our target datasets.

2.2.1 Leaf Area Segmentation

We propose to segment leaf areas from a textured background following the framework shown in Figure 2.6. A step-by-step implementation of the algorithm is provided in Figure 2.7 – displaying the output at various stages as follows:

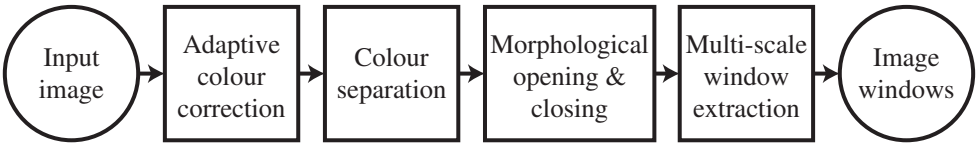


Figure 2.6: An overview of our image segmentation algorithm.

Figure 2.7: An example of our leaf area segmentation algorithm. Left to right: Input image, contrast normalised output, RGB thresholding output, morphological opening and closing output, final result superimposed with extracted windows.

1. Colour Correction and Separation

Firstly, local colour correction is applied to the input image to reduce within-image illuminance variation. The image is then colour separated by thresholding the green colour channel. This produces a binary image that masks the green shaded subjects of interest from the arbitrary background. Ideal values for colour correction and thresholding were identified through experimentation with the target dataset.

2. Image Morphology

Image morphology is performed on the binary image to refine the leaf area mask. A morphological opening with a large square element is favoured in order to segment broad shaped leaves and ignore narrow shaped ones (such as grass). A morphological closing follows in order to remove small holes in the leaf area regions that were eroded as a result of the opening. This leaves us with a refined leaf area binary mask.

3. Window Extraction

An algorithm was developed to extract rectangular regions of interest (or *feature windows*) from the segmented leaf areas. The algorithm decimates the image horizontally and vertically in order to rapidly find windows of a desired size whose four corner points correspond to leaf areas in the segmented mask. For invariance to scale, three different window sizes are extracted. Optimal window sizes are selected through experimentation with the target datasets.

2.2.2 Scale and Rotation Invariant HoG Feature Extraction

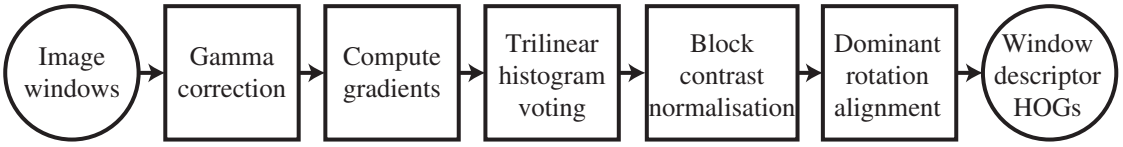


Figure 2.8: An overview of our feature extraction algorithm.

Our proposed algorithm for extracting scale and rotation invariant HoG descriptors from the segmented feature windows is summarised in Figure 2.8. It closely follows the algorithm introduced by Dalal and Triggs [63]. The main steps of this algorithm are as follows:

1. Gamma Correction

Firstly, power law gamma correction is applied to the feature windows to provide normalisation to within-image illuminance variation. A power of $\frac{1}{2}$ (or square root correction) provided the best results in experimentation.

2. Gradient Computation

Image gradient magnitudes and orientations are computed by convolving the feature window with a discrete derivative kernel and its transpose to approximate image derivatives in the horizontal and vertical directions.

If we define $\mathbf{A}$ as the feature window image, and $\mathbf{K}$ and $\mathbf{K}^T$ as the horizontal and vertical derivative kernels, then the respective directional derivative approximations of $\mathbf{A}$ are given by:

$$\mathbf{G}_x = \mathbf{K} * \mathbf{A} \text{ and } \mathbf{G}_y = \mathbf{K}^T * \mathbf{A}, \tag{2.1}$$

where $*$ denotes the 2-dimensional convolution operation. Approximations of the gradient magnitudes ($\mathbf{G}$) and orientations ($\mathbf{\Theta}$) of the image $\mathbf{A}$ are then given by:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \text{ and } \mathbf{\Theta} = \text{atan2}(\frac{\mathbf{G}_y}{\mathbf{G}_x}). \tag{2.2}$$

In this study we investigate three derivative kernels and their transposes – the 3×3 Sobel kernel $\mathbf{K} = \left[\begin{smallmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{smallmatrix}\right]$, the 3×3 Prewitt kernel $\mathbf{K} = \left[\begin{smallmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{smallmatrix}\right]$ and the 1-D cubic-corrected kernel $\mathbf{K} = \left[\begin{smallmatrix} 1 & -8 & 0 & 8 & -1 \end{smallmatrix}\right]$.

3. Trilinear Histogram Voting

For each pixel in the feature window, weighted votes are distributed to a set of orientation histograms based on the gradient orientation centred at that pixel. Votes are accumulated into orientation bins that are equally spaced over $[0° - 180°]$ (for *unsigned* gradients) or $[0° - 360°]$ (for *signed* gradients). The votes are also distributed over local spatial regions, known as *cells*. To reduce aliasing, we linearly interpolate orientation votes between neighbouring orientation bin centres; and bilinearly interpolate spatial votes between neighbouring horizontal and vertical cells. The result is a trilinear approach, which is further detailed in [64]. For this study, we let the unweighted vote value be equal to the gradient magnitude for each pixel.

4. Block Contrast Normalisation

Further invariance to illumination is achieved by grouping cells into larger spatial *blocks* and normalising the histogram responses within each block. For this methodology, we investigate two methods of block contrast normalisation.

Let us define $\mathbf{v}$ as the non-normalised descriptor vector for one block of cell responses and $\|\mathbf{v}\|_k$ as its k-norm for $k = 1, 2$. The *L1-Norm* and *L2-Norm* methods are then given by:

$$\frac{\mathbf{v}}{\|\mathbf{v}\|_1 + \epsilon} \text{ and } \frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}}, \tag{2.3}$$

where ϵ is an arbitrarily small constant to avoid dividing by zero.

5. Dominant Rotation Alignment

Invariance to rotation is achieved via a dominant rotation alignment of the histogram responses in each cell. This involves computing the dominant orientation bin in the cell – or the local maximum. The cell histogram bins are then re-aligned such that the dominant

bin is first and the cyclic-order of the histogram is maintained. This approach is similar to that utilised by Lowe for the scale-invariant feature transform (SIFT) [66].

6. Window HoG Descriptors

The output of this process is a set of HoG descriptor vectors for each set of multi-scale feature windows. The descriptors then serve as the input to the machine learning phase of the methodology. Scale-invariance is achieved in the combined classification of the multi-scale window descriptors.

The visualisation of the HoG features computed from four sample lantana leaf texture windows represents the magnitude of all present gradients in a 360 degree spider plot (Figure 2.9). The dominant gradients in each neighbourhood have a larger magnitude. For example, the dominant 45 degree leaf vein skews the HoG at 45 degrees in the bottom-right graphic of Figure 2.9.

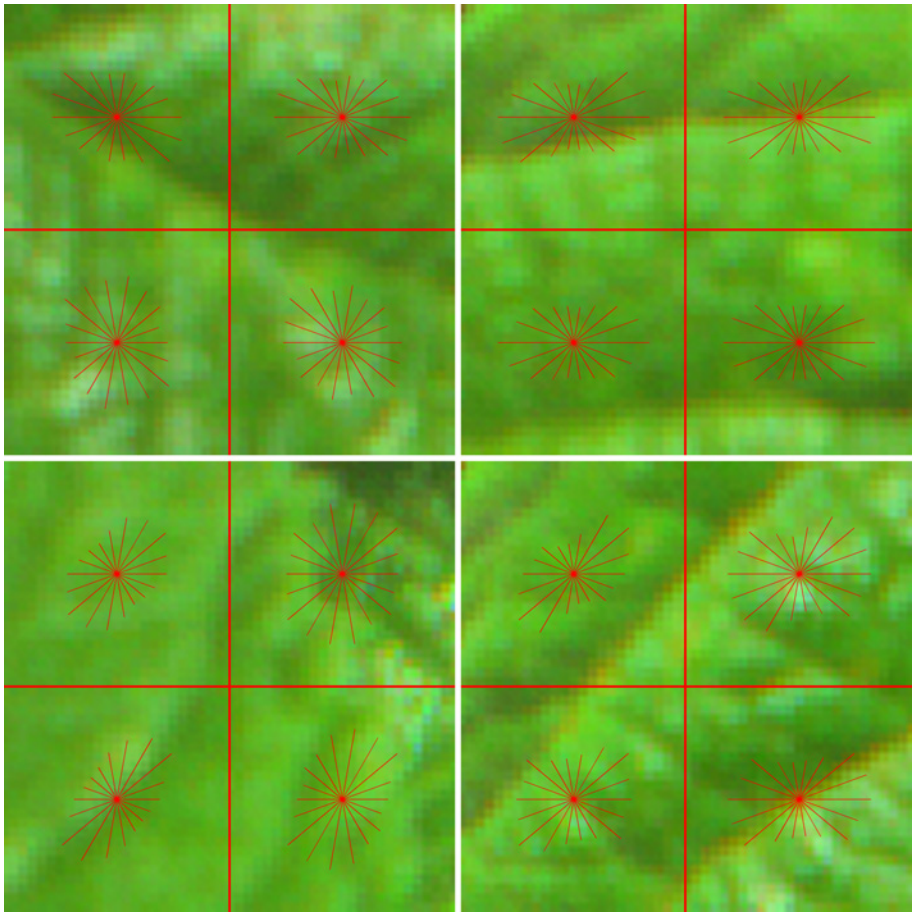


Figure 2.9: A visualisation of the HoG feature descriptors overlaid in red against four 96 × 96 pixel extracted lantana leaf image windows.

2.2.3 Two-Stage Binary Classification

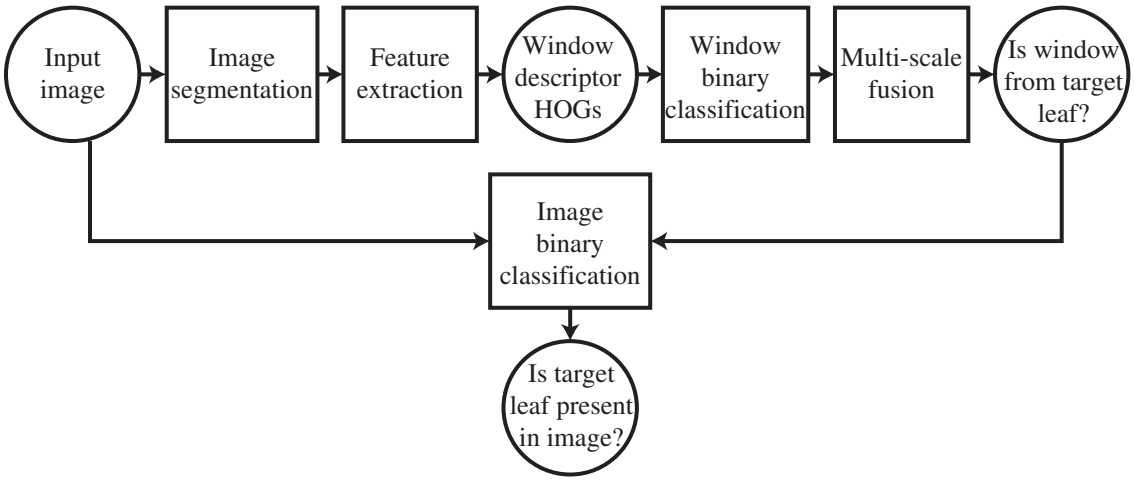


Figure 2.10: An illustration of our proposed two-stage binary classification algorithm for detecting if a target leaf is present in an image.

The machine learning problem here is binary in nature. We wish to recognise a single plant species within an image based on a texture description of small regions taken from the plant's leaf. To accomplish this we propose a two-stage binary classification algorithm for recognising a specific plant species by its leaf within an in situ floral image (Figure 2.10). What follows is a detailed explanation of each step in the two-stage binary classification process:

1. Window Classification

The stage one (or *window*) classification takes place after image segmentation and feature extraction, when the HoG descriptors have been computed for the extracted feature windows. We train this classifier by manually labelling whether or not the HoG descriptors for each feature window are from a target leaf. The output of this first stage is a set of predicted labels for each feature window, describing whether or not they are from the target plant species.

2. Multi-Scale Fusion

In order to achieve scale invariance we make use of our multi-scale feature window HoG descriptors. Suppose n individual binary classifiers c_k for $k = 1, 2, \ldots, n$ are trained using HoG descriptors extracted from the same region of interest in an image using n different window sizes. Each classifier assigns new regions of interest x a label representing one of

two classes $y = 0, 1$. These decisions are based on whether the confidence score for each classifier (z_k) is above or below zero. We propose a fusion of the classifier scores such that the final label decision is:

$$y = \begin{cases} 1, & \text{if } \frac{\sum_{k=1}^{n} z_k}{n} \geq 0 \\ 0, & \text{otherwise} \end{cases}. \tag{2.4}$$

This method is similar to [65] but varies in the method used to calculate the fusion output. Deniz et al fuses classifiers via a product of the posterior probability vectors, whereas we take an average of the confidence scores. Both achieve a similar result however the latter approach is less computationally complex.

3. Image Classification

The set of predicted labels resulting from multi-scale fusion serves as the input to the second stage (or *image*) classification. In order to train at this stage, the set of feature window labels are grouped by image. The decision strategy here is based on the ordered pair (two-dimensional vector) (a, b) where a is the number of predicted target windows in the image, and b is the number of predicted non-target windows. The final output is a prediction of whether the target plant species is present in the image based on the number of target texture windows found within the entire image.

2.2.4 Validation with the *Flavia* Dataset

Now that a full detection framework has been put forth, we can optimise and evaluate its performance. Two experiments are proposed: firstly, an evaluation of the novel HoG framework to optimise its parameters against the Flavia leaf image dataset [30] and secondly, in situ evaluation of the performance on lantana. The former experiment allows us to validate the performance of the model and tune available parameters using an existing leaf image dataset, prior to an evaluation of in situ performance.

For both experiments, four interchangeable machine learning methods (see 2.1.3) were tested to determine which provides the best results. To estimate how the predictive model performs in practice, we utilise k-fold cross-validation (with $k = 10$) for in situ lantana classification. While for the Flavia dataset, 70% of the data is partitioned for training and the remaining 30% for testing.

Dataset

This dataset contains 1,907 leaf images of 32 different types of plants (not including lantana). The images are photographed against a white background at a very high resolution. They are not taken in situ, and do not exhibit the pitfalls of illumination variance, overlapping leaves and other occlusions. The dataset provides a high variance of leaf texture among the 32 species, which are illustrated in Figure 2.11.

Figure 2.11: A sample image from each of the 32 species of the Flavia leaf image dataset [30]. Note that two species had narrow shaped leaves and they have been removed for this broadleaf specific analysis.

Experimental Method

Leaves of two species (*Cedrus deodara* and *Podocarpus macrophyllus*) were removed from the experiment because their narrow, grass-like shape was not fit for feature window extraction. This highlights a central problem for this detection algorithm, as it cannot be adopted for all weed species. The methodology outlined earlier was implemented here to conduct 30 individual binary classifications on the Flavia dataset – attempting to classify each individual species amongst the remaining 29. Because each image contains just one leaf from a plant species, the latter part of the two-stage binary classification discussed in Section 2.2.3 is not required.

The optimal values for the HoG parameters discussed in Section 2.2.2 (cell size, block size, number of orientation bins, etc.) were found through experimentation on the target dataset. This required the definition of a default HoG implementation against which parameter adjustments were compared.

Results

The results of the HoG parameter optimisation are documented below in Table 2.1. Several inferences can be made about the optimal HoG feature set for leaf classification.

Table 2.1: Summary of the parameter optimisation strategy to validate the performance of the HoG feature set using the Flavia leaf image dataset.

Description of HoG parameter variation	**Accuracy (%)**	**Accuracy Comparison (%)**
Default implementation	83.75	=
With Gaussian smoothing	86.29	+2.54
With gamma correction	84.12	+0.37
With colour feature windows	84.61	+0.86
With unsigned gradient orientations	84.35	+0.60
No block normalisation	83.75	=
L1-Norm block normalisation	85.07	+1.32
L2-Norm block normalisation	86.05	+2.30
0% block overlap	84.21	=
50% block overlap	83.48	-0.73
Sobel derivative operators	83.75	=
Cubic-corrected derivative operators	84.44	+0.69
Prewitt derivative operators	84.40	+0.65
180 orientation bins	83.75	=
90 orientation bins	83.72	-0.03
30 orientation bins	83.61	-0.14
10 orientation bins	82.88	-0.87
4 orientation bins	79.15	-4.60

Gaussian smoothing prior to gradient computation improves the classification accuracy. We infer that smoothing ensures the leaves venation pattern dominates texture from other sources (hair, glands and imperfections). The use of gamma correction and RGB feature windows significantly increases classification accuracy. This shows that the illumination invariance is achieved through the use of gamma correction and that the gradient information in each colour channel is of importance.

Removing signed gradients increases the performance. We assume that this is because the dominant rotation alignment is more effective when only using unsigned gradients. Local contrast normalisation is shown to be essential for classification performance. With both L1-Norm and L2-Norm block normalisation methods outperforming the non-normalised descriptor classification, this again highlights the importance of illumination invariance for leaf classification.

The overlapping of spatial blocks proved to be an important component in using HoGs for human detection [63]. For leaf classification however, 0% block overlap has provided the best results. We infer that the redundancy of texture information for the overlapping case may lead to over fitting of the classifiers. The 1-D cubic-corrected derivative operator outperforms the other 3-D derivative kernels for classification of the Flavia dataset. We infer that the cubic-corrected derivative kernel provides a sharper and stronger intensity gradient magnitude than the other kernels. As a result, the discrimination of leaf texture is more apparent. The last item of Table 2.1 illustrates that classification accuracy increases as the number of orientation bins increase. This tells us that fine orientation sampling is required for discriminating leaf texture.

By comparison, Figure 2.12 reveals that rather coarse spatial sampling is more effective for classifying leaf texture. With large 160×160 pixel cells of 1×1 cell blocks providing the best accuracy at 84.85%. Similar results are achieved for the 4×4 block sizes. However, the size of the descriptor increases quadratically with the block size. Hence, the speed of the extraction process is optimal for block sizes of 1×1.

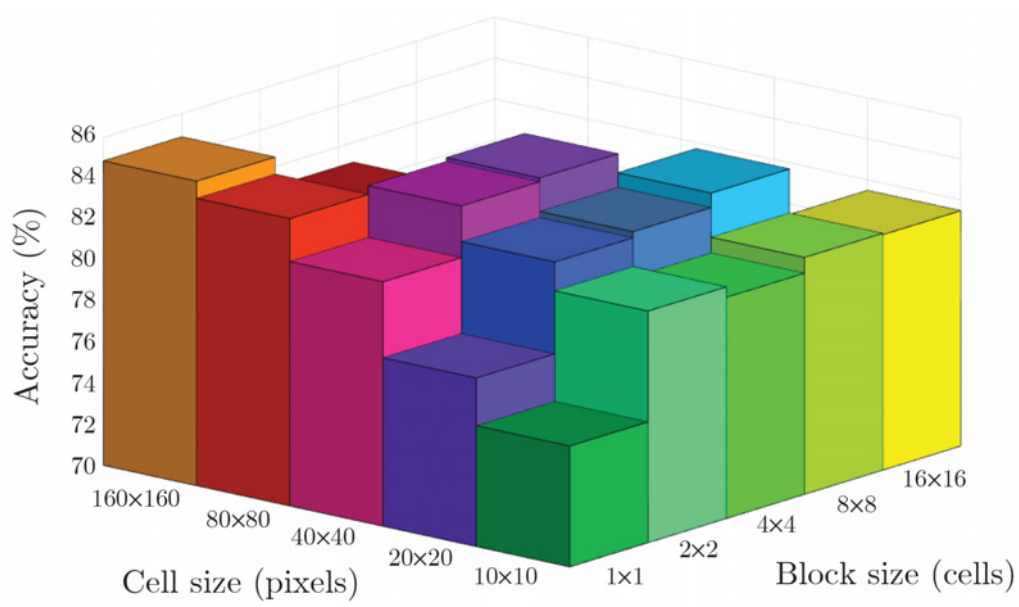


Figure 2.12: The average Flavia species classification accuracy using logistic regression as the cell and block sizes are varied (with the block overlap fixed at 0%).

The results from Table 2.1 and Figure 2.12 help us define the optimal HoG implementation for describing leaf texture. Using the optimal feature window descriptors we carry out a final simulation varying the four machine learning methods of interest against the three selected window sizes and the multi-scale window fusion method. As shown in Figure 2.13.

This figure illustrates that the window fusion algorithm in tandem with Gaussian SVM provides the best performance – with an average accuracy of 94.72%.

This result compares very well with the 90% multi-class classification results obtained using multiple feature Probabilistic Neural Networks by Wu et al on the exact same Flavia dataset [30]. In addition this result validates the use of texture as a single discriminator and shows that our proposed HoG implementation is viable for this dataset.

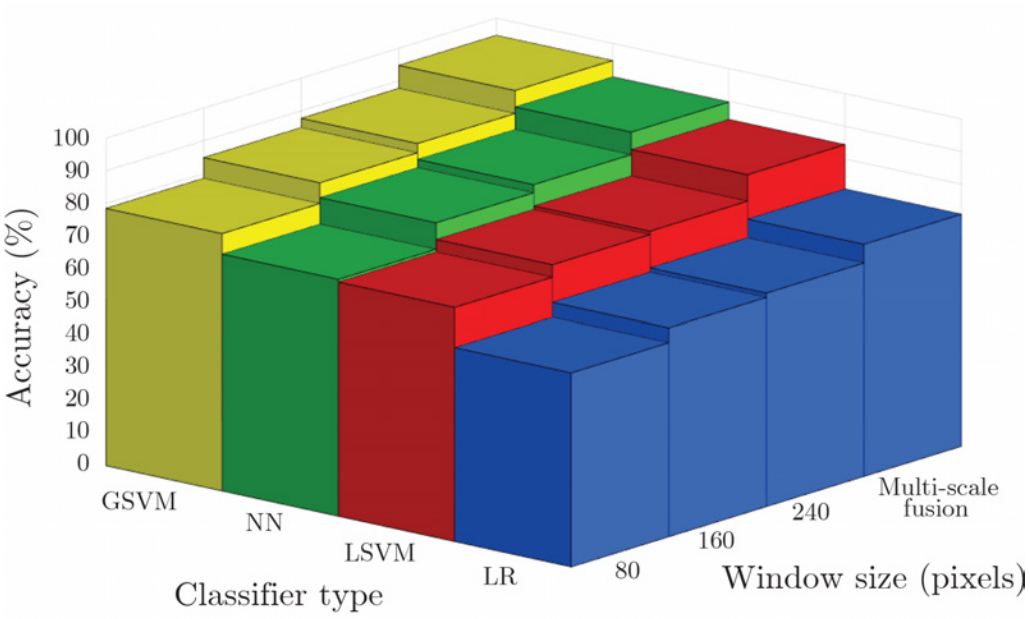


Figure 2.13: The average Flavia species classification accuracy as the binary classifier and window size are varied. Multi-scale window fusion with Gaussian SVM performs best with 94.72 % accuracy.

2.2.5 Lantana Classification

Following the successful validation of our HoG feature set in Section 2.2.4, we then tested this approach on a field collected dataset of lantana images.

Dataset

A collection of images were acquired of lantana in its native environment amongst neighbouring flora. 337 images were acquired exhibiting variance in illumination, rotation and scale as well as overlapping leaves and other interferents. The number of images is small in comparison to the previous dataset. However, the number of leaves photographed is comparable.

Experimental Method

All images were labeled prior to the experiment as to whether or not a lantana leaf is present within its frame. The full methodology discussed in Section 2.2 was carried out. The latter stage of the two-stage binary classification discussed in section 2.2.3 allows us to classify the images in the dataset based on the number of lantana and non-lantana windows classified within.

The optimal HoG implementation for leaf texture discrimination derived in Section 2.2.4 was utilised for this experiment. Different window sizes (48 pixels, 96 pixels and 144 pixels) were selected due to the change in resolution of the target dataset. As a result the cell sizes were scaled down to match the multi-scale window sizes.

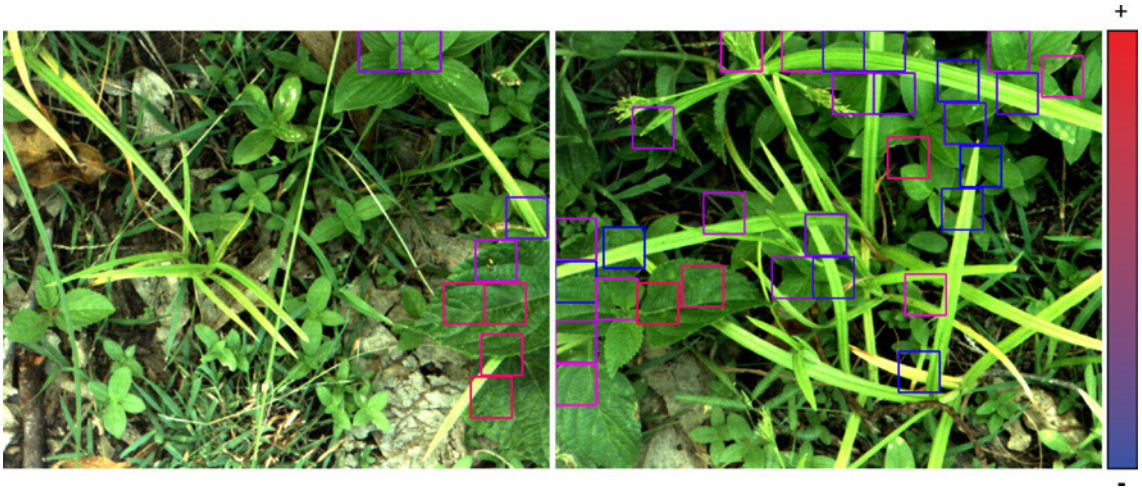


Figure 2.14: A demonstration of the classification output of our proposed methodology on two random images from the lantana dataset. The border colour of the leaf image feature windows correspond to the classifier's likelihood estimation that the feature window is a positive (red) or negative (blue) target – in this case lantana.

Results

The leaf texture-based classifier performed well, correctly classifying lantana texture regions regardless of illumination variance, overlapping leaves and other interferents in the in situ image, as shown in Figure 2.14. While neighbouring flora is correctly classified as non-target plants despite similar shape and colour features.

The results from stage one of the binary classification algorithm (Figure 2.15) show the window classification accuracy varying with the four machine learning methods of interest against the three selected window sizes and the multi-scale window fusion method. This figure illustrates that the window fusion algorithm in tandem with the ANN classifier

provides the best performance – with an average accuracy of 75.42%.

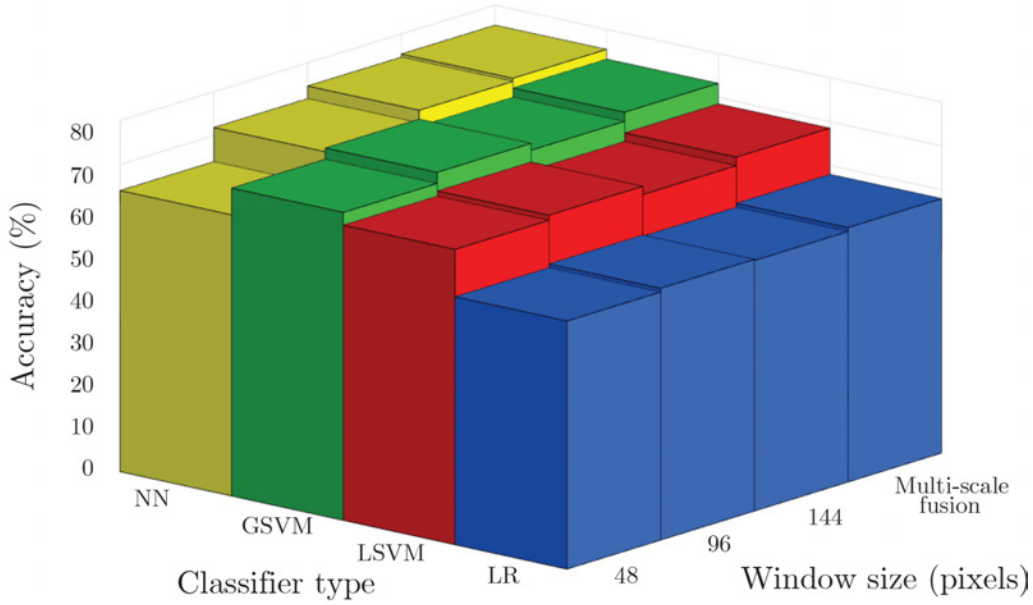


Figure 2.15: The window classification accuracy for the lantana dataset using four different binary classifiers on three different window sizes and using multi-scale window fusion.

This result compares poorly to the 94.72% window classification accuracy of the Flavia dataset because of the in situ variations of the images. Taken at face value, this result might be too inefficient for field implementation. However, field implementation depends upon whether any lantana is identified within the frame (i.e. the latter part of our classification algorithm).

The results from stage two of the binary classification algorithm (Figure 2.16) illustrate the accuracy of our method at correctly predicting the presence of lantana in an in situ image. Note that these simulations were conducted after using an ANN classifier for the stage one window classification, which was shown to offer the best performance from Figure 2.15.

It can be seen that our multi-scale window fusion approach in tandem with a logistic regression classifier offers the best results – with an image classification accuracy of 86.07%. We infer that logistic regression outperforms other machine learning methods for image classification due to the small size of the input vector at this stage.

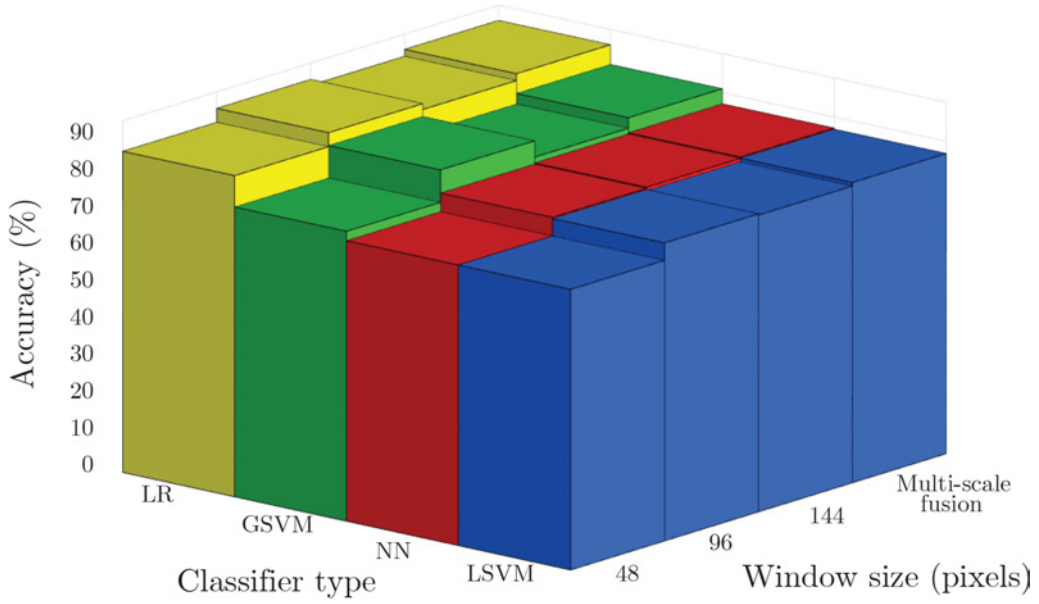


Figure 2.16: The relationship between classifier type and window size on the classification accuracy for the lantana dataset using four different binary classifiers, three different window sizes and multi-scale window fusion.

Real-Time Speed

For the classification technique to be applicable in a real-time environment, the computation time of the algorithms need to be considered. Using the optimal HoG parameters, the computation time of the image segmentation and feature extraction processes averaged less than 100 milliseconds in processing 1280×1024 frames on a 5th Generation Intel® Core™ i7 Processor and without a high-end GPU. If a CPU-only hardware implementation can perform classification in less than 100 ms, this means an optimised parallel implementation of the algorithm can far outperform the real-time requirement for in-field detection. This is a strong result that ensures the real-time implementation of this approach is viable. Next, we take this lab-verified weed detection system and deploy it on a proof of concept weed control system.

2.3 Proof of Concept

A proof of concept for the robotic spot-spraying of lantana regrowth during follow up treatment is presented here. Spot-spraying was chosen as the control method because of its simplicity to automate. The central goal of the proof of concept is to field-test the

weed species detection algorithms developed above. Therefore, budget priority was given to the machine vision and computer processing subsystems of the design. Design choices were henceforth made to simplify remaining subsystems (i.e. positioning and spraying) as much as possible by purchasing off-the-shelf products rather than developing them from the ground up.

The vehicle base was chosen to be a small-sized towable car trailer, upon which the detection and spray system could be retro-fitted. The trailer could then be towed by a car over target weeds to simulate how an autonomous, or self-propelled vehicle would perform. The trailer had enough room to house a chemical spray tank and pump for herbicide application, as well as several mounting points for affixing the cameras and a line of herbicide sprayers.

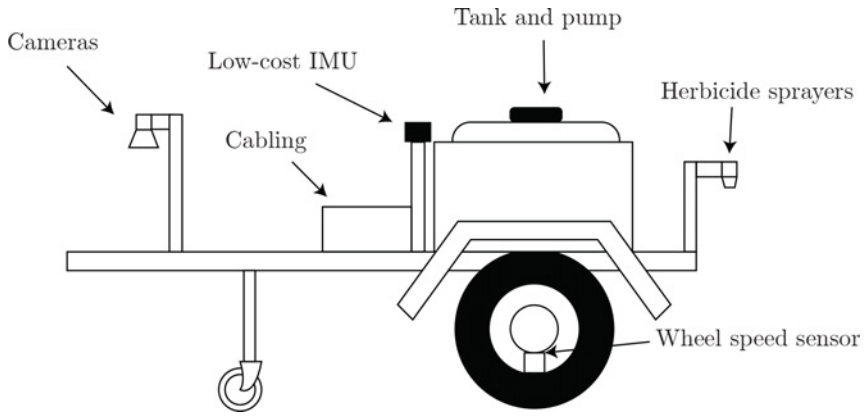


Figure 2.17: Design illustration for the proof of concept weed detection and spot spraying system mounted to a towable trailer.

2.3.1 Design Overview

The initial design of the system consists of a row of cameras at the front of the towable vehicle and a row of sprayers at the back (Figure 2.17). Images are collected from the cameras, and processed by an on-board laptop computer using the leaf classifying detection algorithm outlined above. The position of the vehicle is continuously monitored using a low-cost Inertial Measurement Unit (IMU) and Wheel Speed Sensor (WSS). Any detected weed targets have their position saved and tracked in reference to the position of the sprayers as the vehicle moves forward. When a target passes under one of the sprayers at the back of the trailer, the sprayer solenoid is activated to douse the target weed with herbicide (or water during experimental trials).

Figure 2.18: Completed design of the proof of concept weed detection and spot spraying system mounted to a towable trailer.

The design includes an array of eight cameras aligned with an array of eight electrically activated solenoid sprayers. The camera field of view and spray swath covers a total width of 2 m. The fully realised proof of concept is photographed in Figure 2.18.

2.3.2 Hardware Design

Mount Design

A number of modifications were made to the base towable trailer to retro fit the detection and spray system. The trailer hitch was extended by more than 1 m to accommodate a jockey wheel and an unobstructed camera field of view. The central hitch bar presented as a potential obstruction of the view of the camera array. The optical system was therefore specifically design such that the angular field of view was not obstructed by the hitch bar. An additional rail bar at the rear of the vehicle was welded into place to allow mounting of the spray line. The spray line is mounted along a 40 $\times$ 40 mm steel square hollow section.

Inertial Measurement Unit

Dead reckoning was the chosen method of positioning for the proof of concept. Dead reckoning refers to the use of on-board positioning sensors to determine the future position of the vehicle based on past measurements of its direction and velocity [73]. A Sparkfun Razor IMU (SEN-10736) sensor was chosen due to its relatively low-cost. The sensor

provides nine degrees of freedom with an on-board gyroscope, digital accelerometer and magnetometer. The IMU was implemented to provide estimations of the vehicle's yaw over time. The IMU was positioned along the central access of the trailer, as shown in Figure 2.19. It was also mounted on a wooden frame above the hitch bar of the trailer to limit the interference of the magnetometer relative to the steel frame. The position of the IMU served as the local positioning origin of the trailer and a reference point for the camera and sprayer positions in software.

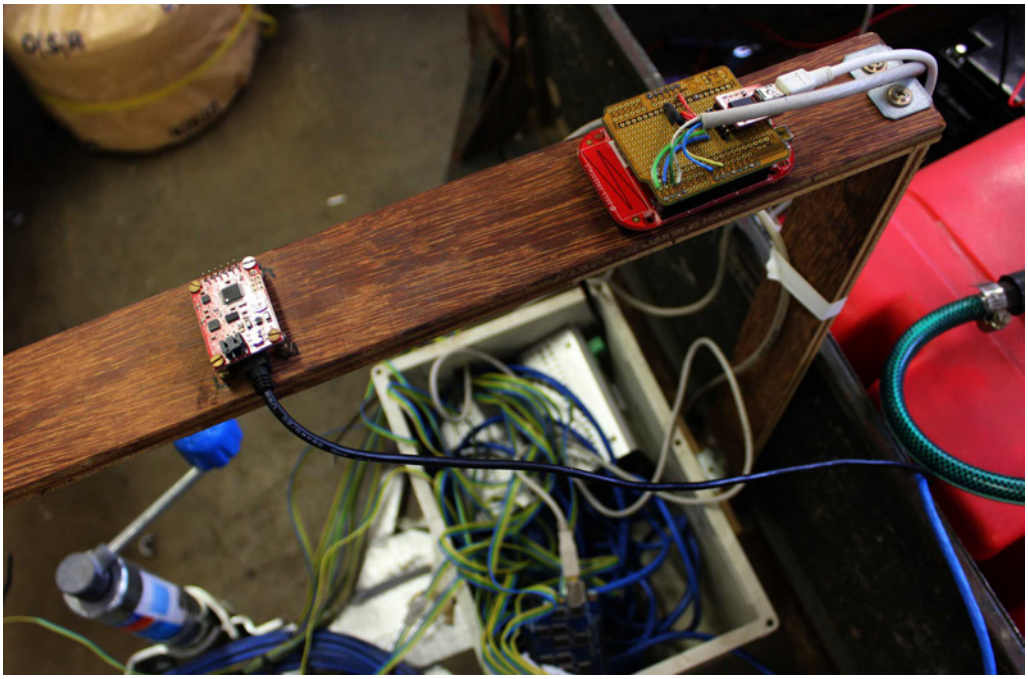
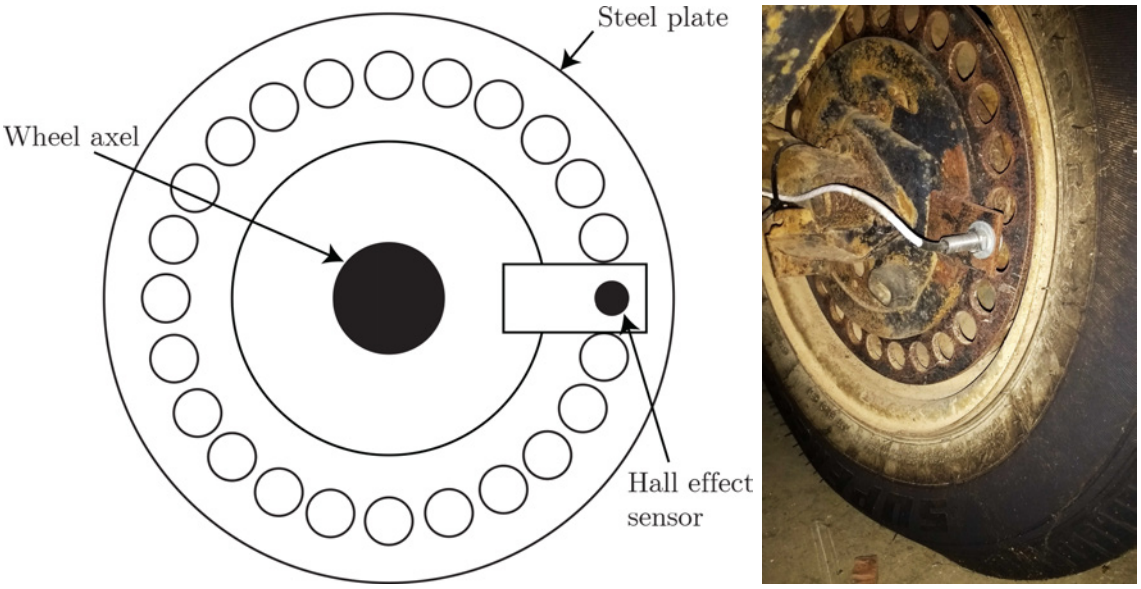


Figure 2.19: The IMU mounted in-place on the proof of concept vehicle.

Wheel Speed Sensor

The speed of the vehicle is to be estimated by the wheel speed, which can be measured by an appropriate sensor. A hall effect magnetic sensor was installed next to the right wheel of the vehicle along with a custom fabricated metal disk with a series of concentrically cut holes, as shown in Figure 2.20. A hall effect sensor detects and counts the presence of a ferrous material. Here, the absence of a hole on the metal disk acts as a magnetic pickup point for the hall effect sensor. Therefore, when over metal, the hall effect sensor reads high and when over a hole it reads low. As the wheel rotates, a square wave is generated. The period of the square wave can then be measured to determine the rotational speed of the wheel. The vehicle speed can then be estimated from the rotational speed of the wheel using the apparent diameter of the wheel when inflated optimally.

A Kinetis FRDM-KL25Z development board was programmed to interface with the hall effect wheel speed sensors and performed the processing discussed above providing the vehicle speed to the on-board laptop. The Kinetis development library had a variety of signal processing functions to simplify calculation of the time period between magnetic pickups. The Kinetis library also provided a real-time operating system such that the wheel speed could be processed efficiently.



(a) (b)

Figure 2.20: Left: Illustrative design of the hall effect sensor and steel pickup disk. Right: Photograph of the disk and sensor mounted to the inner right wheel axel.

With instantaneous estimates of vehicle heading and speed, the vehicle's velocity is known over time. An estimate of vehicle position can then be made by integrating velocity with respect to time. This is a crude dead-reckoning algorithm for estimating the vehicles position. It is subject to a number of sources of error including but not limited to: changes in wheel diameter, friction, drag, magnetic interference and sensor manufacturer error. These errors will also accumulate with time when approximating vehicle position through integration. We therefore expect the positioning system to be limited in capability. However, wheel speed sensors are known to provide realistic estimates of vehicle position for straight line motion. And at the very least, straight line spray testing should be achievable.

Optical System

A target resolution of at least 5 px/mm was sought for deploying the aforementioned leaf texture classification algorithm. The cameras required mounting at least 1 m above the ground in order to pass over the regrowth lantana weed targets. To achieve the 2 m field of view, the cameras were spaced 25 cm apart allowing some overlap of camera field of views so that duplicate target detections are minimised. The Point Grey (now FLIR) BlackFly 13E4C-CS Gigabit Ethernet cameras were chosen based on their sensor resolution, easy-to-use software API, small form-factor and suitability to mobile application. With the camera sensor selected, the aforementioned optical constraints led to the selection of the 25 mm Fujinon HF25HA-1B c-mount lens. The 19°58′ angular horizontal field of view of the lens allowed the centre-most cameras to see below the centre hitch bar of the trailer and have an unobstructed field of view. The cameras were mounted to a 40 mm steel square hollow section and connected to the on-board laptop via an 8-port Power-over-Ethernet (PoE) network switch (Figure 2.21). This allowed the cameras to be powered by the Ethernet port.

Figure 2.21: Photograph of the chosen camera and lens optical system mounted to the proof of concept.

Spraying System

The sprayer bar (as shown in Figure 2.22) positioned eight solenoid sprayers at 1 m off the ground spaced 25 cm apart, aligning parallel to the cameras. An overlap of 5 cm in the 30 cm diameter circular spray areas ensured adequate coverage of weed targets, even if a

weed were to pass between the middle of two sprayers. Each sprayer consists of a TeeJet tee-junction nozzle body connected to a wet boom made up of hose sections. A TeeJet e-chemSaver electric solenoid shutoff valve is attached to each nozzle body allowing precise electronic control of the spray. A standard selection of TeeJet nozzles are then applicable to the spraying unit, and can be adjusted for each weed species under consideration. The solenoids are each electrically connected to a relay control board which can be directly controlled by the on-board laptop. A standard 50L Silvan Smoothflo Spotpak Sprayer was purchased for the central pump and tank. The 120 psi maximum pressure system provided adequate line pressure for fast activation of the sprayers.

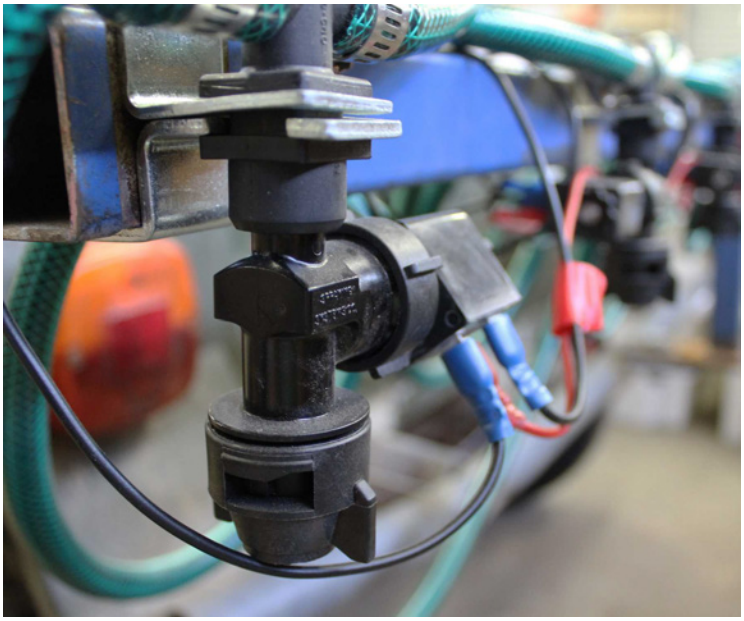


Figure 2.22: Photograph of the spraying system positioned at the rear of the vehicle.

Processing and Power

A Dell Inspiron 15 3000 with a 1.8 GHz AMD A6-6310 CPU was utilised to perform the main on-board processing. Where possible, task-specific computation was performed off the on-board laptop. For example, wheel speed values were processed into vehicle speed with a Kinetis FRDM-KL25Z interface board. IMU readings were also collected and formatted using an Arduino Uno and provided to the on-board laptop. The laptop was therefore responsible for collecting images from the cameras, running the detection algorithm, integrating vehicle speed and yaw into position, tracking position over time and activating sprayers when necessary.

The entire system is powered with a single 12V lead-acid car battery. The spraying

system and camera network switch operate with a nominal 12 V, while the lower voltage electronics are powered through their connection to the laptop.

2.3.3 Software Design

The software outlined here covers the tasks performed by the on-board laptop (Figure 2.23). This software was written in C# to comply with API libraries for various peripherals, including the IMU, cameras and solenoid-relay driver board.

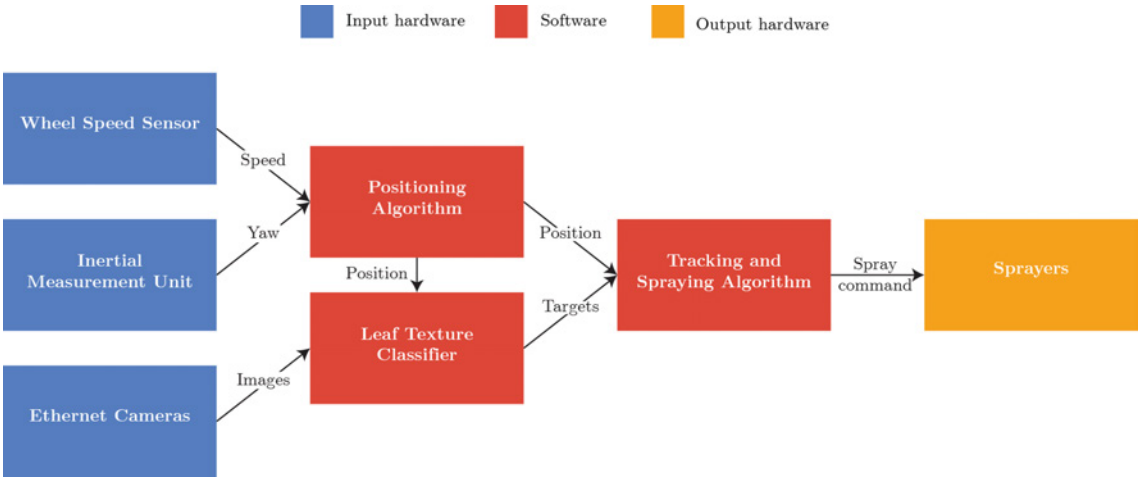


Figure 2.23: Flowchart describing the software required to detect, track and spray weed targets on the proof of concept system.

Leaf Texture Classifier

The program interfaces with the input and output hardware as defined above. The leaf texture classifier for detecting lantana was written in C#. It was slightly modified to include input from a stream of camera images, as opposed to a static dataset for the previous classification task. The algorithm was also modified to take a current position input from the simultaneously running positioning algorithm. The leaf texture classifier would then output position and time-stamped weed targets to the tracking and spraying algorithm in order to be sprayed.

Positioning Algorithm

The positioning software implemented a very primitive dead reckoning system consisting of yaw readings from an IMU and a hall effect based wheel speed sensor for velocity. At start-up, the yaw value from the IMU was taken as the origin. When the trailer began to move the yaw and velocity readings were used to calculate trajectory and absolute position

from the starting point. The positioning algorithm is subject to error, which would worsen over time. To limit this error, integration of velocity to find position was performed for each target with the initial time being set to zero when the target is found. Therefore, the total time of integration is the amount of time it takes for the target to pass from underneath the camera to the sprayer as the vehicle moves.

Tracking and Spraying Algorithm

After a target had been identified by the recognition system, its position would be entered into an array storing all currently not-sprayed targets. To ensure the target position was accurate, each image was stamped with the time and position of when the image was acquired. A software thread would then loop through this array to determine if any of the targets were about to pass under any of the sprayers at the back of the trailer. If so, the sprayer or sprayers (if the target passed between the overlap of two) would be set to fire for a period of time based on the velocity of the trailer, ensuring sufficient coverage of the target weed.

Performance Testing

The proof of concept weed detection and spot spraying system was then taken through a series of experimental trials to determine its detection, positioning and spraying accuracy. The first series of tests investigated the position accuracy of the dead-reckoning system for straight line and path motion.

2.3.4 Experimental Results

Straight Line Positioning

This test involved measuring out an exact path for the vehicle to follow in a straight line while recording positioning information from the system's software. The ground truth measured distance can then be compared to the system's estimated distance to determine the accuracy of the dead-reckoning positioning algorithm. A 5 m straight line path was measured and the vehicle was manually pushed to traverse the path. The resultant plot of the measured and estimated paths are shown in Figure 2.24. The final position error after traversing 5 m was approximately $\pm$ 50 mm in absolute position. If this positioning system was to be used for autonomous navigation this error would be unacceptable. However, for our given application, position only needs to be integrated for a maximum of 2 m – the distance between the camera and spray (i.e. the distance between a target being seen and sprayed). Therefore, assuming linear accumulation of the error, the absolute error in the

positioning system for hitting a target weed is ± 20 mm. With target weeds varying in size from 200 − 500 mm, this was deemed to be an acceptable level of error. Therefore, this crude dead-reckoning positioning system is sufficient for tracking and spraying weed targets if travelling in a straight line.

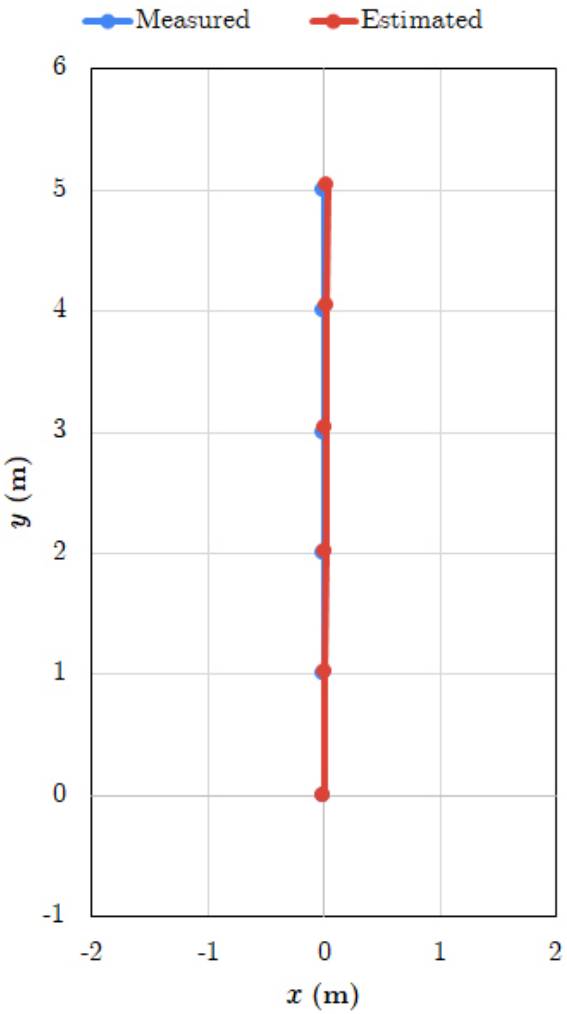
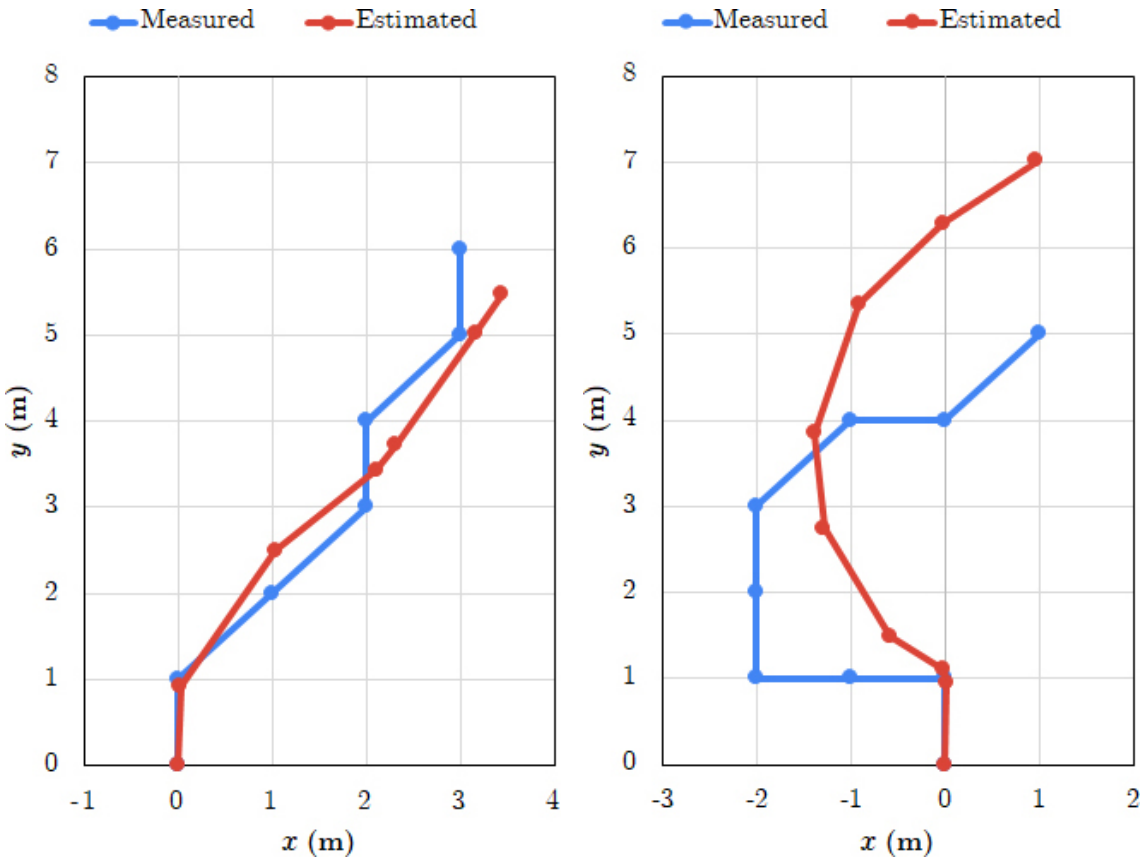


Figure 2.24: Plot of the measured and estimated paths for the proof of concept during the straight line experiment.

Complex Path Positioning

The second round of positioning tests involved mapping out slightly more complex paths that incorporated a series of left and right turns. The results for two paths are illustrated in Figure 2.25 below. It was found that anything but straight line motion was highly inaccurate. The absolute position error for motion trial 1 is ± 0.69 m over 7.24 m and the

absolute position error for motion trial 2 is ± 2.01 m over 8.83 m. This significant error level would make it impossible to accurately spray target weeds on complex paths using this dead-reckoning positioning system. The position of the vehicle becomes less accurate as a result of the vehicle turning (Figure 2.25).



(a) (b)

Figure 2.25: Plots of the measured and estimated paths for the proof of concept during (a) motion trial 1 and (b) motion trial 2.

Detection and Spraying

To investigate the performance of weed detection and spraying, a custom path was mapped out for the vehicle to traverse through a small lantana infestation. However, during testing of the detection algorithm, it was found that there was very low detection accuracy (approximately 60%) of in situ lantana when using the prototype spot-spraying vehicle. This was apparently due to the small size of the lantana dataset only capturing a small sample size of the population of lantana targets encountered in the field.

Consequently, to complete the detection and spraying trial, photographs of lantana sam-

ples from within the testing dataset were printed and positioned as ground truth lantana targets along a custom vehicle path. As the vehicle traverses the path, the detections made by the system are recorded along with the estimated path of the vehicle and the locations where the vehicle activated its sprayers.

In this trial on lantana it can be seen that the initial straight line motion allows for successful and accurate detection and spraying of the targets (Figure 2.26). However, as the system begins to move laterally the error in yaw estimation causes the position estimation to be inaccurate, and the resulting target tracking and spraying algorithm to miss all subsequent targets. Despite the spray accuracy suffering due to inaccurate dead reckoning, the detection accuracy of the system was upheld with all five ground truth targets detected.

During the spraying testing it was found that the sprayer solenoid response was too slow at 250 ms, and its timing not consistent. This resulted in targets being missed entirely as this timing inconsistency could not be accounted for. It was also found that the solenoids caused electromagnetic interference with the hall effect sensor, resulting in severe noise and incorrect velocity determination. However, when the solenoids did fire in a suitable time and were not causing electromagnetic interference the system performed adequately for straight line motion.

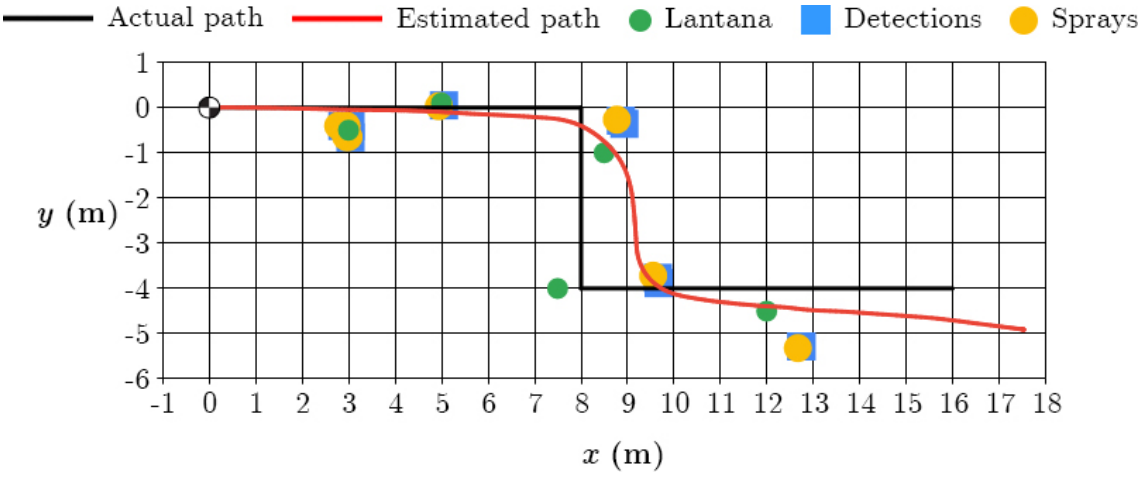


Figure 2.26: Detection and spray trial results for the proof of concept, illustrating strong straight line performance and positional accuracy that deteriorates as it turns.

2.3.5 Limitations

The proof of concept showed that real-time and in situ detection and spraying of weed species was possible. However, the design was subject to a variety of limitations that

needed to be overcome if developing a prototype for commercial use. These limitations include:

- The positioning system of the proof of concept is too crude. Dead reckoning with a limited number of on-board sensors only allowed adequate positioning accuracy for straight line motion. For a successful prototype to track and spray weed targets regardless of vehicle path, a more sophisticated positioning system would be required. This would most likely involve the use of Kalman filtering to integrate global positioning accuracy of a GPS system with local positioning of a highly-accurate on-board IMU.

- Relying solely on yaw estimation with a magnetometer caused a variety of issues if the vehicle were to pass by large ferrous objects. During the trials, if the vehicle passed by larger steel objects the yaw reading would vary wildly due to disturbance of the magnetic field affecting the magnetometer.

- Large, intermittent positioning error was also apparent due to aberrant skipping of the wheel speed sensor. Investigation revealed that these large skips of the hall effect sensor were due to electromagnetic interference caused by firing of the solenoids which were coupled together through power wires and the chassis.

- The single CPU (on-board laptop) limited the vehicle operation speed to 5 km/hr. Much higher speeds would be required for a prototype. Furthermore, better housing and protection of the CPU from the elements would be required. This limitation could be fixed by operating fewer cameras or utilising parallel GPU processing.

- The insufficient dataset size of 337 images did not capture enough variability of the lantana weed targets for the detection system to be successful in the target environment for testing. In order to ensure that the detection model can be applied to unseen targets in real-time, much larger datasets would need to be collected.

- The system had poor maneuverability throughout the rangeland environment. This was mainly due to the short trailer hitch making it all too easy to jack-knife the direction of the trailer when towing it with a car. A self-propelled and manned vehicle, like an All-Terrain Vehicle (ATV) or quad bike would make for a much more navigable solution.

- The response time of the solenoids was far too long, with a maximum of 250 ms. This is likely due to the use of a relay-board between the CPU and solenoids. It is recommended to instead use a custom printed circuit board to electronically activate

the solenoid with a transistor circuit design. Alternate solenoids with faster response times should also be investigated.

2.4 Chapter Summary

2.4.1 Contributions

In summary, this chapter covered the site-specific case study of detecting and controlling lantana with a texture based classifier and a novel proof of concept robotic solution. The contributions of this chapter include:

- Introducing a successful novel leaf area segmentation algorithm for complex in situ backgrounds.
- Presenting a rotation and scale invariant HoG feature set, which was used to accurately classify lantana from neighbouring flora by discriminating solely on regions of texture within their leaves.
- The introduction of a challenging in situ dataset of lantana, upon which our HoG feature set performed accurately with real-time speed.
- Development of a novel dead-reckoning positioning algorithm using only an IMU for yaw and WSS for speed to achieve $\pm$ 50 mm accuracy over 5 m of straight line motion.
- Development of a novel weed detection and spraying proof of concept that shows the efficacy of robotic weed control in complex environments.

2.4.2 Lessons Learned

Based on the aforementioned limitations encountered throughout the design of the detection system and proof of concept, several recommendations can be made for the development of future spot-spraying prototype systems. A more reliable and accurate positioning system is needed, ideally one that combines an IMU, velocity sensing, and RTK-GPS using sensor fusion and Kalman filtering. To ensure accurate herbicide application, the sprayers need to have a fast activation time (<50 ms and consistent timing). More powerful computing devices should be sought that are also more suited to edge computing in harsh environments. A self-propelled or manned vehicle would provide better maneuverability of the target system. Larger image datasets of the targeted weed species should be collected to allow more robust in field classification.

2.4.3 Future Funding

The work in this chapter laid the foundation for a funding application under the "Control Tools and Technologies for Established Pest Animals and Weeds Grant Programme" administered by the Australian Government's Department of Agriculture and Water Resources [9]. The bid, which was successful in 2017, proposed to extend the proof of concept and weed detection system developed in this chapter on a variety of fronts. Firstly, it aimed for robust classification of eight important weed species with the collection of a large, multiclass image dataset which is the basis of chapter 3. The next goal was to investigate using state-of-the-art deep learning detection algorithms to classify the weeds in situ, which is detailed in chapter 4. Then finally, the grant project funded further development of the proof of concept into an all-terrain spot-spraying prototype for rangeland and crop environments, which is the basis of chapter 5.

Chapter 3

Dataset Collection

Often the most important step in the machine vision framework, dataset collection, is not paid the attention it deserves. Care taken to mitigate sources of error during dataset collection reaps future benefits by maximising the performance of learning models that are developed to classify the dataset. This chapter reviews the current landscape of weed species image datasets that are available and examines their triumphs and pitfalls. A methodology for fast and consistent dataset collection and labelling is proposed with the development of custom hardware and software. This leads to the introduction of DeepWeeds, the first large, public, multiclass, weed species image dataset of weed species from the Australian rangelands.

3.1 Background

3.1.1 Poor Data, Poor Results

This thesis is primarily concerned with developing detection models for weed species that are applicable to the real world scenario of robotic weed control. When deploying a decision model in the field – be it a convolutional neural network, or simple logistic regression – the performance of the model is dependent upon the quality of the dataset on which it was trained. As such, there are a variety of ways dataset collection can fail an otherwise well conceived learning algorithm. To avoid falling prey to such error, this chapter will highlight the pitfalls to be avoided in the construction of sound datasets.

A common pitfall of weed species classification and detection models in the literature is the insufficient size of datasets [74–76]. For instance, Ahmed et al. developed a Support Vector Machine (SVM) that classified a dataset of 224 images with six classes of weeds and crops based on their shape and colour features [74]. The model achieved over 97% classification accuracy on the test dataset however no field implementation is presented to explain how this model performs in the real world. Models trained to this dataset would only offer suitable performance on real world scenarios matching the dataset imagery; and with only 37 images for each of the six species, there are minimal real world scenarios that are applicable. Similarly, Herrera et al. presents a novel approach to discriminate between grasses and broadleaf weed species with a set of shape descriptors and a Fuzzy decision-making methodology [75]. They achieve 92.9% accuracy on a 66 image dataset. Obviously for this detection model to be evaluated for real world use, more images would need to be trained and tested against.

This issue is no more apparent than is shown in the results of Chapter 2 of this thesis when deploying the HoG leaf texture model in a variety of in situ locations. The model, which performed well on the 337 image lantana dataset (and in specific real world locations that matched the dataset) subsequently failed to detect lantana in the majority of in situ locations. This is considered to be due to the inability of the model to adapt to image backgrounds and target conditions which it has not seen previously.

In contrast, works that leverage the ability of deep CNNs have maximised their success with larger datasets [34, 77]. The former work [34] collected a 15,336 image dataset from aerial image acquisition of a soybean field. The resulting dataset included 3,249 images of soil, 7,376 images of soybean, 3,520 images of grass and 1,191 of broadleaf weeds. An implementation of the AlexNet deep CNN architecture achieved 99% average classification of all species with 98% recall accuracy of the weed class. The size and alignment of the dataset to its target application, aerial mapping on a soybean field, provides a firm foundation for strong real world performance.

Similarly, Yu et al. attached a machine vision camera to an ATV to collect 118,000 images in winter cereal and spring cereal fields over a 770 hectare total area [77] . Of these, only 1,368 images were selected for training and validation with a Single Shot Detector [78] implementation of VGG16 [79] for object detection. This comparatively low number of images collected is likely due to the time constraints of labelling the images with bounding boxes. After the labelling process, the original 1,368 image dataset became a 13,177 bounding box image dataset. The deep CNN object detector achieved 60% detection accuracy on the weed object class. This accuracy is relatively low compared to past studies, however this is due to this work focusing on object detection rather than image classification. Object detection is a more difficult task and an accuracy of 60% in such a challenging dataset is a fair result.

Another source of dataset collection error occurs when the dataset fails to match the target application in either the method of image acquisition, or the representation of the classes present. An example of the mismatched image acquisition in a dataset would be taking a very broad public dataset with images acquired from a variety of devices (smartphones, cameras, etc.) such as ImageNet [80] or the plant-specific Pl@ntNet [81] and Leafsnap [31], and deploying it for a specific object or plant detection task. If image acquisition in the target application is known to be fixed, a disservice to model performance is done if training on a broad dataset. Such a dataset will be subject to variations in image quality and features that are outside the domain of the target detection problem.

The second common error in collection is when the distribution of classes in the target application does not match the distribution of classes in the training dataset. In most cases of weed management, the percentage of weed density is quite low with more non-target plant life than target weeds. Therefore, for maximal performance this distribution should be reflected in the sampled dataset. Dos Santos Ferreira et al. [34] accomplishes this and presents a 7.8% infestation of broadleaf weeds in the aerial image dataset of a soybean field.

The design of the optical system for image acquisition also plays a significant role in model performance and should be tailored to the application. This can range from the obvious – whereby the optical system should be designed such that the target objects are always in frame, in focus and optimally lit – and it can also include more novel decisions, such as which spectrum of light should be utilised to maximise the target object's features. For example, for the application of non-destructive agricultural crop grading, the choice of utilising visible image or spectral information will depend on whether the crop product has a husk, and if the spectral signature has more descriptive feature information than a visual image of the target.

3.1.2 Weed Species Detection

Three primary methods of detection exist that focus on different representations of the light spectrum. Varied success has been achieved using image-based [30–37, 82, 83], spectrum-based [84, 85] and spectral image (or hyperspectral) based [86, 87] methods to identify weeds from both ground and aerial photography. These three methods and the types of devices used in dataset acquisition are illustrated in Figure 3.1.

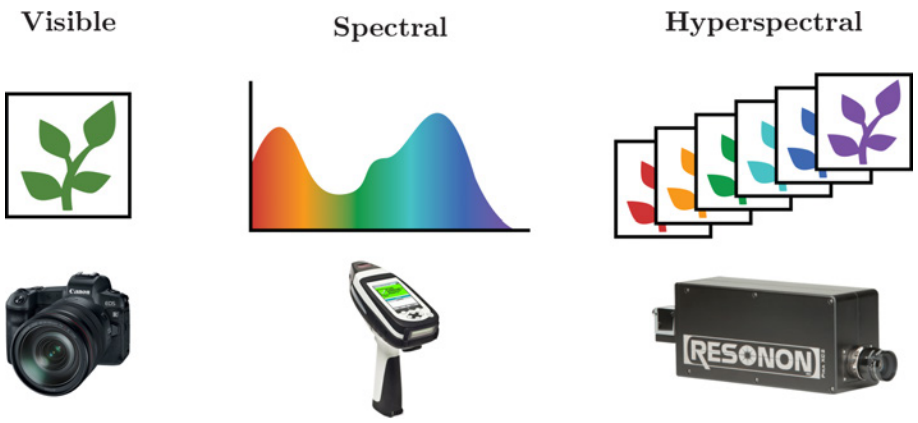


Figure 3.1: An illustration of the differences between visible image analysis, spectral analysis and hyperspectral analysis, including examples of their associated data acquisition devices.

Visible image analysis is concerned with representation of the target object in the visible spectrum. Colour, shape, texture and other visible patterns are the features being discriminated against. This is a powerful feature space, as evidenced by the ability of the human eye and brain to perform classification. Image acquisition in this domain is relatively simple compared to spectral data acquisition. Image sensors sensitive to the narrow visible band of the light spectrum capture reflectance from the target field of view and quantify their intensity. These images can be manipulated in a variety of image and colour formats. They embody limited spectral information with 2D spatial information.

The visible spectrum is however quantitatively limited. Extending to the non-visible spectrum of information permits more characteristics of the target objects for discrimination. Spectral analysis concerns the quantification and discrimination of the recorded spectral signature of a target object. This spectral signature is limited however to sampling from a single spatial dimension. Therefore the average spectral intensity of an entire field of view is quantised for each spectra available in the spectrometer, the device used for data acquisition.

Hyperspectral (or spectral-image) based methods utilise the best of both worlds – the 2D spatial information of visible images and the breadth of spectral information from

spectrometers. The result is a hypercube of spectral images, with each image capturing the spectral response of a specific wave band in the light spectrum for each of its pixels. This results in computationally expensive manipulation of hyperspectral data, which can be reduced by focusing the instrumentation on specific bandwidths of the light spectrum.

Spectrum and spectral image-based methods are most suitable for highly controlled, site-specific environments, such as arable croplands where spectrometers can be tailored to their environment for consistent acquisition and detection. The homogeneity of the cropping environment makes it simpler to calibrate spectral responses to distinguish soil from crop. However, the more varied and dense plant life in pastoral rangeland environments make spectral-based methods challenging to implement. Meanwhile image-based methods benefit from cheaper and simpler image acquisition in varying light conditions, especially when deployed on a moving vehicle in real time [88]. The primary focus of this thesis is to deliver an applicable detection method for weed control in both crop and pasture environments. We therefore focus our efforts on the visible light spectrum.

3.1.3 Existing Datasets

A plethora of plant species image datasets are present in the literature with varying size, purpose and utility. The first and most prominent dataset to consider is the annual LifeCLEF [89–91] plant identification challenge. Their 2017 plant identification challenge presented roughly 1.1 million plant image samples from 10,000 different plant species. The size of this dataset was achieved by querying images submitted by the users of the mobile application *Pl@ntNet*. This crowd-sourced dataset collection and labelling tool has amassed an incredibly large and useful dataset for training extremely generalised plant species classifiers. The 2017 competition winner achieved an 88.5% top-1 classification accuracy on the dataset by averaging the output of ImageNet pretrained GoogLeNet [92], ResNet [93] and ResNeXt [94] deep CNNs [95]. This result speaks to the power of transfer learning and indicates that the discriminating features of the general ImageNet dataset are not only transferable to a plant specific dataset – but they may be intrinsic.

Pl@ntNet currently boasts a total of 1.9 million images of 18,067 different species [81]. It also includes a weeds-specific subset with 711,408 images of 1,321 different species.

Another crowd-sourced dataset of plant image features is Leafsnap [31] which now constitutes 30,866 plant leaf images of 185 different species. Leafsnap allows users to photograph and add to the dataset with a phone app (as shown in Figure 3.2) and also utilises a developed learning model to help users predict the species in their taken photograph. Kumar et al. [31] used the distinctive shapes of leaves as the sole visual cue to differentiate between species by extracting histograms of curvature from a binarised image at multiple scales.

This dataset is not useful for real-time robotic weed control primarily because it requires users to photograph leaves of plant species against a light, non-textured background which is not suitable for in situ application. This was however, desirable for Leafsnap to make up for the immense difference in image quality with a multitude of users having different mobile phone cameras to collect the dataset.

Figure 3.2: Left: A screenshot of the iPhone Leafsnap app. Right: Sample images from all 184 tree species in the Leafsnap dataset.

Both Pl@ntNet [81] and Leafsnap [31] also contain an inherent hurdle due to the crowd-sourced nature of their dataset collection method. They exhibit a massive variability in image acquisition, with different cameras, field of view, image quality, and other sources of variability related to the optical system. These are all unnecessary obstacles for the challenge of robotic weed control and should be mitigated with the design of a consistent optical system for dataset collection and real-time classification. We therefore look beyond Pl@ntNet and Leafsnap to site-specific weed species datasets that have been tailored for a specific robotic application.

An abundance of site-specific weed control studies exist focusing on different weed species, with varying detection algorithms in multiple application environments [30–37, 74–77, 82, 83]. The majority of studies concentrate on cropping situations, with few targeting pasture weed species and rangeland environments. Two major shortcomings in many of these works include: insufficient dataset size and a lack of in field scrutiny. A common reason for insufficient dataset size can be attributed to excessive collection and labelling time. Therefore, efforts made to streamline a consistent collection and labelling process are worthwhile. The best studies, such as [34] and [77], exhibit very large, field-collected datasets. To bolster the efficacy of robotic weed control, large datasets capturing the variability of complex environments in situ need to be collected so that robust classification models can be trained that exhibit both strong lab and field performance.

3.1.4 Goals

The aim of this chapter is to support the development of robust weed detection models with the collection of a large, in-field image dataset of various weed species in complex rangeland environments. Careful consideration was taken for the key factors in dataset collection that can aid the learning process. These factors include: the optical system, scene variability, dataset size, weed targets, weed locations, negative samples, image metadata and labelling. Several goals were established to support this aim.

1. Develop a dataset collection instrument to facilitate consistent and efficient image acquisition matching the optical system of the targeted robotic weed control application.
2. Identify several weed species and locations of interest to local landholders in the northern Queensland region of Australia.
3. Collect at least 1,000 images of each target species to ensure sufficient dataset size.
4. Attain a 50:50 split of positive to negative class images from each location to prevent any location bias, where image classifiers could predict weed species based on location image features rather than weed image features.
5. Publish the dataset to foster research engagement in weed species detection.

3.2 WeedLogger

The dataset collection process outlined here aims to support the development of a robotic weed control prototype, which is detailed further in Chapter 5. Unfortunately, the robotic platform of the system presented later in Chapter 5 was not available at the time of dataset collection. Therefore, it could not be utilised to collect images in the dataset. Instead, a dataset collection instrument, *WeedLogger*, was developed to expedite the image acquisition and labelling process. This instrument was tailored to match the target application for robotic weed control with an identical optical system.

3.2.1 Target Application

Oftentimes image processing frameworks fail in real world application because they are hamstrung by unforeseen errors during the first and most important step in the framework: image acquisition [96]. The images acquired must match the target application as closely as possible for real world success. Our goal is to use the collected dataset to train a ground-based weed control robot; therefore, we must tailor the dataset to match this application.

Our prototype ground-based weed control robot, *AutoWeed* (illustrated in Figure 3.3 and discussed more extensively in Chapter 5), will incorporate high-resolution cameras and fast acting solenoid sprayers to perform selective spot spraying of identified weed targets. Boundary conditions in the prototype which affect the design of the optical system and subsequent classification models include:

- The height from the camera lens to the ground was set to 1 m in order to allow the system to target weed regrowth for a variety of weed species. Similarly, the ground clearance underneath the robotic vehicle is 288 mm. Weed targets will rarely exceed this height, therefore the optical system should have a depth of field of approximately 288 mm from the ground up.
- No external shading or lighting is to be used for the optical system, so as not to limit the vehicle's maneuverability. The camera and lens must be chosen and utilised to adequately capture dynamic lighting in the scene.
- The ideal vehicle speed while spraying is $10 - 20$ km/hr. The field of view of the optical system is 450×280 mm. This gives the system approximately $100 - 50$ ms per image (or $10 - 20$ fps) to detect a weed target before a new image is captured and ready for processing. This vehicle speed also requires a fast shutter speed to resolve images without motion blur.
- The system will operate under harsh environmental conditions. Therefore we look exclusively at machine vision cameras and lenses to provide the robust mechanical specification required here. Similarly, it is beneficial for external camera and lens parameters (such as focus, iris, zoom) to be fixed.

3.2.2 Instrument Design

A data logging instrument was developed to: (1) photograph images with the same optical system as the target robotic platform, (2) ensure consistent image acquisition and (3) accelerate the image collection process.

The *WeedLogger* (pictured in Figure 3.4) consists of a Raspberry Pi, high resolution camera, machine vision lens and a GPS receiver. The FLIR Blackfly 23S6C Gigabit Ethernet high-resolution colour camera was chosen for this design. Its large (1920 x 1200 px) and high dynamic range (73.90 dB) image sensor affords robust imaging of a wide field of view in a highly contrasted scene. The 25mm fixed focal length Fujinon CF25HA-1 machine vision lens was paired with the $1/1.2''$ image sensor to provide a 254 mm depth of field focused to a working distance just above the ground with an aperture of $f/8$. At a working distance of 1 m, this optical system provides a 450 mm $\times$ 280 mm field of view

Figure 3.3: *AutoWeed*: A prototype weed control robot for selective foliar spot spraying in the Australian rangelands.

for one image. This translates to just over 4 px per mm resolution, which was shown to be sufficient for the leaf texture recognition in Chapter 2 [76].

3.2.3 Optical System

The lens' mode of operation was designed to resolve detail in the shadows and highlights of high dynamic range scenes without motion blur while moving at high speeds. This was achieved by selecting an aperture size of $f/8$ to allow some sunlight in; while simultaneously restricting the shutter speed to less than 0.05 ms. The automatic exposure and automatic white balance algorithms within FLIR's FlyCapture Software Development Kit were utilised to achieve acceptable imaging without the need for manual tuning between different sites. Nevertheless, colour variations will occur in the images due to changing light conditions in the natural environment throughout the day. Rather than accounting for this directly, our preference is to capture this variability in the training set for associated machine learning algorithms. Finally, the touchscreen interface allowed for in-field labelling of geo-mapped images. GPS data was collected automatically using a SkyTraq Venus638FLPx GPS receiver, V.Torch VTGPSIA-3 GPS antenna, an Arduino Uno and custom electronics shield. The GPS data was used exclusively to track progress during the dataset collection process. GPS data has not been used in the development of our classification models.

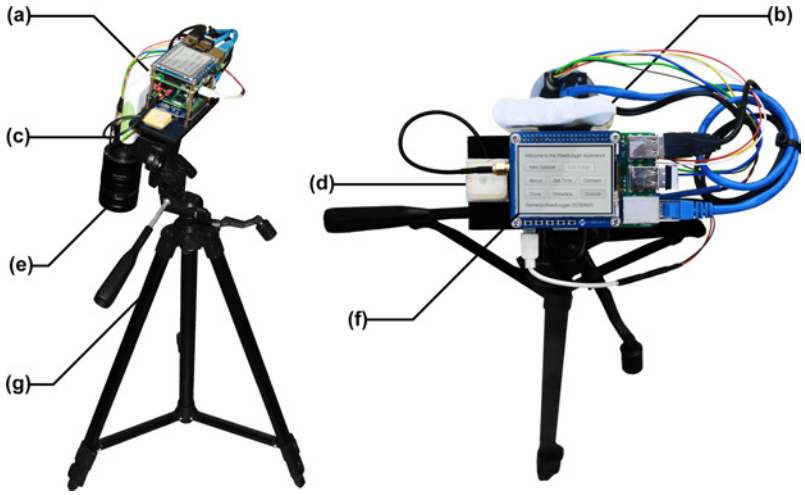


Figure 3.4: The *WeedLogger* field instrument consisting of: (a) a Raspberry Pi 3, Arduino Uno and custom electronics shield, (b) a rechargeable lithium-ion battery pack, (c) a FLIR Blackfly 23S6C Gigabit Ethernet high-resolution colour camera, (d) a SkyTraq Venus638FLPx GPS receiver and V.Torch VTGPSIA-3 GPS antenna, (e) a Fujinon CF25HA-1 machine vision lens, (f) a 4D Systems Raspberry Pi touchscreen display module and (g) an Inca i330G light-weight tripod.

3.2.4 Labelling Software

The most time consuming step in the dataset collection process is labelling each image by the eye of a human expert. There are various forms of image dataset labelling depending on the learning problem being addressed. For image classification, the image label indicates whether or not the target being classified is present in the image. More complicated labels include drawing bounding boxes around targets for object detection and localisation.

The nature of the learning task for the robotic weed control methodology investigated in this thesis is classification of images as to whether they contain weed species or not. A software application was developed using C# and Visual Studio 2017. The image format collected by the WeedLogger instrument is accepted exclusively by the labelling software. Images are then presented to the end user four or more at a time for labelling. The user is then able to preselect a weed species label from a configurable list and click (or touch) each image if that weed species is present. The human selected labels are then automatically saved and formatted into a comma-separated file. The WeedLogger instrument allowed on-board labelling to be performed, therefore the software program provided a means of verifying the WeedLogger labels or labelling unlabeled WeedLogger subsets.

A screenshot of the program during labelling of the DeepWeeds dataset is provided below in Figure 3.5. It was found that this software allowed a user to label over 1,000

images per hour for the DeepWeeds dataset.

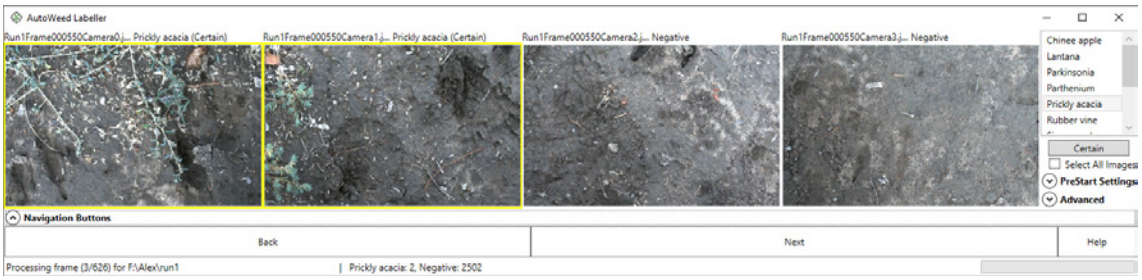


Figure 3.5: A screenshot of the AutoWeed Labelling software developed by Alex Olsen and Benjamin Girgenti being used here to label Prickly acacia in the DeepWeeds dataset.

3.3 The *DeepWeeds* Dataset

3.3.1 Performance Targets

Variability

When designing models or algorithms for learning features, our goal is to separate the factors of variation that explain the observed data [67]. The *depth* of a deep learning model conceptually refers to said model's layer count and parameter complexity. Typically, the more confounding factors of variability in the dataset, the deeper and more complex the model required to achieve acceptable performance [67]. Despite our efforts to mitigate inter-scene variance of photographed images in the design of the optical system; scene and target variability will persist in our target application. Thus, a major design consideration in the construction of this dataset is to capture images that reflect the full range of scene and target variability in our target application. Hence we have chosen to abide several factors of variation, namely: illumination, rotation, scale, focus, occlusion, dynamic backgrounds; as well as geographical and seasonal variation in plant life.

Illumination will vary throughout the day with changing sunlight and canopy cover creating highly dynamic range scenes with bright reflectance and dark shadows. Rotation and scale of the target weed species will vary as they are being photographed in situ with unknown size and orientation. The distance of photographed weed species to the camera are also variable. Therefore, the fixed focal region of the camera will cause some targets to be blurred and out of focus. Fortunately, motion blur is mitigated by operating with an extremely fast shutter speed. Perhaps the largest variability in the dataset is due to complex and dynamic target backgrounds. The locations subject to dense weed infestations are also inhabited by immeasurable counts of other native species. As we are unable to

curate a dataset of all plant life, we must resort to labelling all other non-target plant life as *negative* samples; along with all non-target background imagery. Unfortunately, this creates a highly variable class in the dataset that will be difficult to consistently classify.

In addition to complex backgrounds, target foreground objects can be unexpectedly occluded from view by interfering objects; most often being other neighbouring flora. This is yet another unavoidable factor of variation. Finally, the dataset must account for seasonal variation in our target weed species. This means that a single class of weed species will include photographs of the weed with and without flowers and fruits and in varying health condition; which can affect foliage colour, strength of features and other visible anomalies.

Collection Targets

Two quantifiable targets were established to achieve the required variability and generality of the dataset. First, collect at least 1,000 images of each target species. Second, attain a 50 : 50 split of positive to negative class images from each location. The first goal is a necessity when training high-complexity CNNs which require large labelled datasets. The second target helps to prevent over-fitting of developed models to scene level image features by ensuring targets are identified from their native backgrounds. Finally, the dataset required expert analysis to label each image as to whether it contains a target weed species or not. The rigidity of this collection process will ensure that the accuracy and robustness of all learning models developed to classify from it, will be upheld when applied in the field. The complexity of the learning problem is apparent when viewing sample images from each class due to the inherent variation within classes of the dataset (Figure 3.6).

3.3.2 Weeds

Liaison with land care groups and property owners across northern Australia led to the selection of eight target weed species for the collection of a large weed species image dataset; (1) Chinee apple (*Ziziphus mauritiana*), (2) lantana (*Lantana camara*), (3) parkinsonia (*Parkinsonia aculeata*), (4) parthenium (*Parthenium hysterophorus*), (5) prickly acacia (*Vachellia nilotica*), (6) rubber vine (*Cryptostegia grandiflora*), (7) Siam weed (*Chromolaena odorata*) and (8) snake weed (*Stachytarpheta spp.*). These species were selected because of their suitability for foliar herbicide spraying, and their notoriety for invasiveness and damaging impact to rural Australia. Five of the eight species have been targeted by the Australian Government as *Weeds of National Significance* in a bid to limit their potential spread and socio-economic impacts [13].

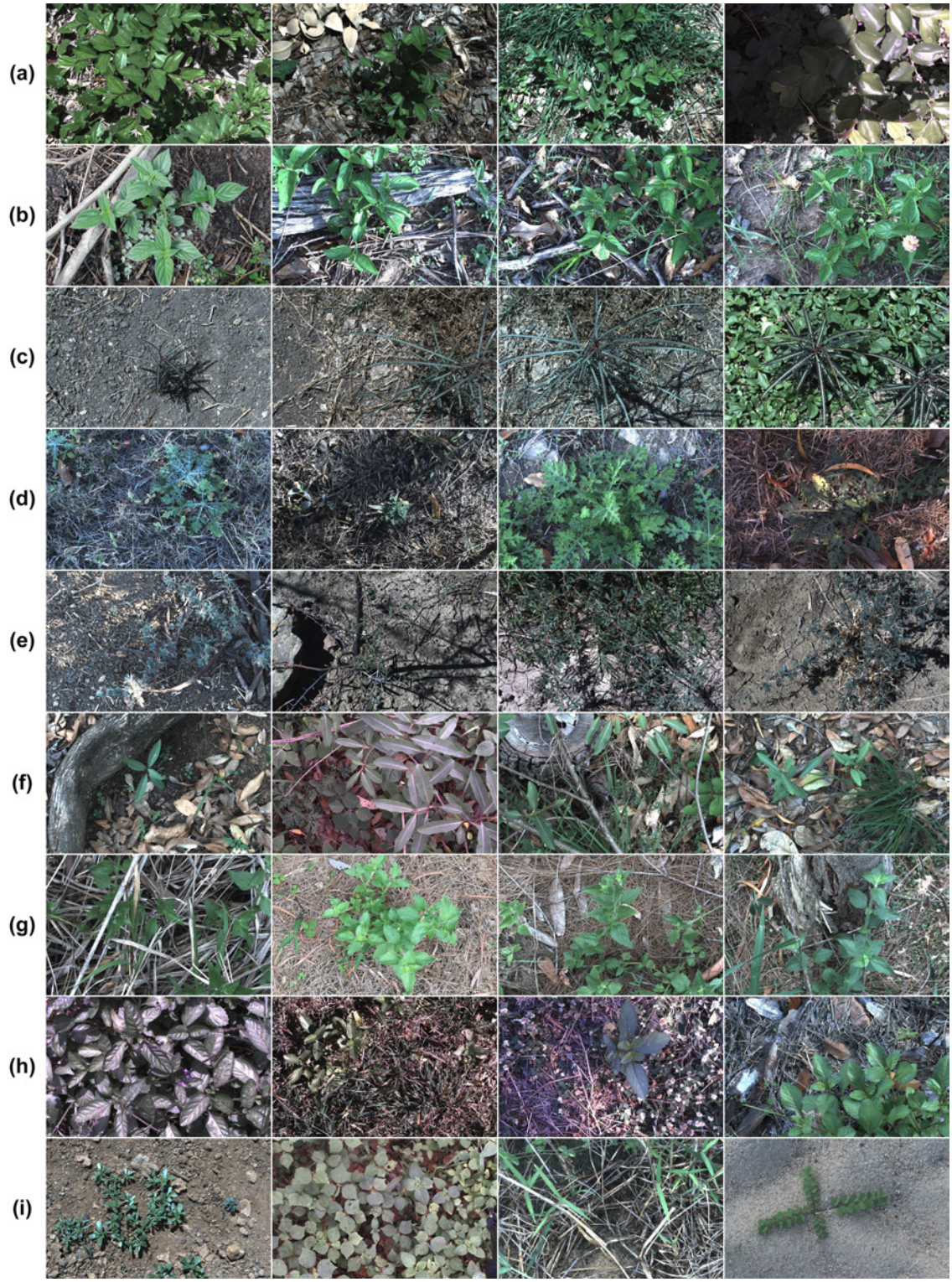


Figure 3.6: Sample images from each class of the *DeepWeeds* dataset, namely: (a) Chinee apple, (b) lantana, (c) parkinsonia, (d) parthenium, (e) prickly acacia, (f) rubber vine, (g) Siam weed, (h) snake weed and (i) negatives.

3.3.3 Locations

The geographical distribution of the collected images can be observed in Figure 3.7. The breadth of the dataset is apparent from Figure 3.7, which spans several collection sites across northern Australia. Each collection site was identified to contain large infestations of specific target species, including: Siam weed from *Black River* (19°13′44″ S, 146°37′45″ E), rubber vine from *Charters Towers* (20°04′44″ S, 146°10′55″ E), parkinsonia from *Cluden* (19°19′02″ S, 146°51′02″ E), snake weed from *Douglas* (19°19′29″ S, 146°45′44″ E), Chinee apple from *Hervey Range* (19°19′35″ S, 146°38′50″ E), parthenium from *Kelso* (19°22′38″ S, 146°43′05″ E), prickly acacia from *McKinlay* (21°20′21″ S, 141°31′27″ E) and lantana from *Paluma* (18°57′23″ S, 146°02′17″ E).

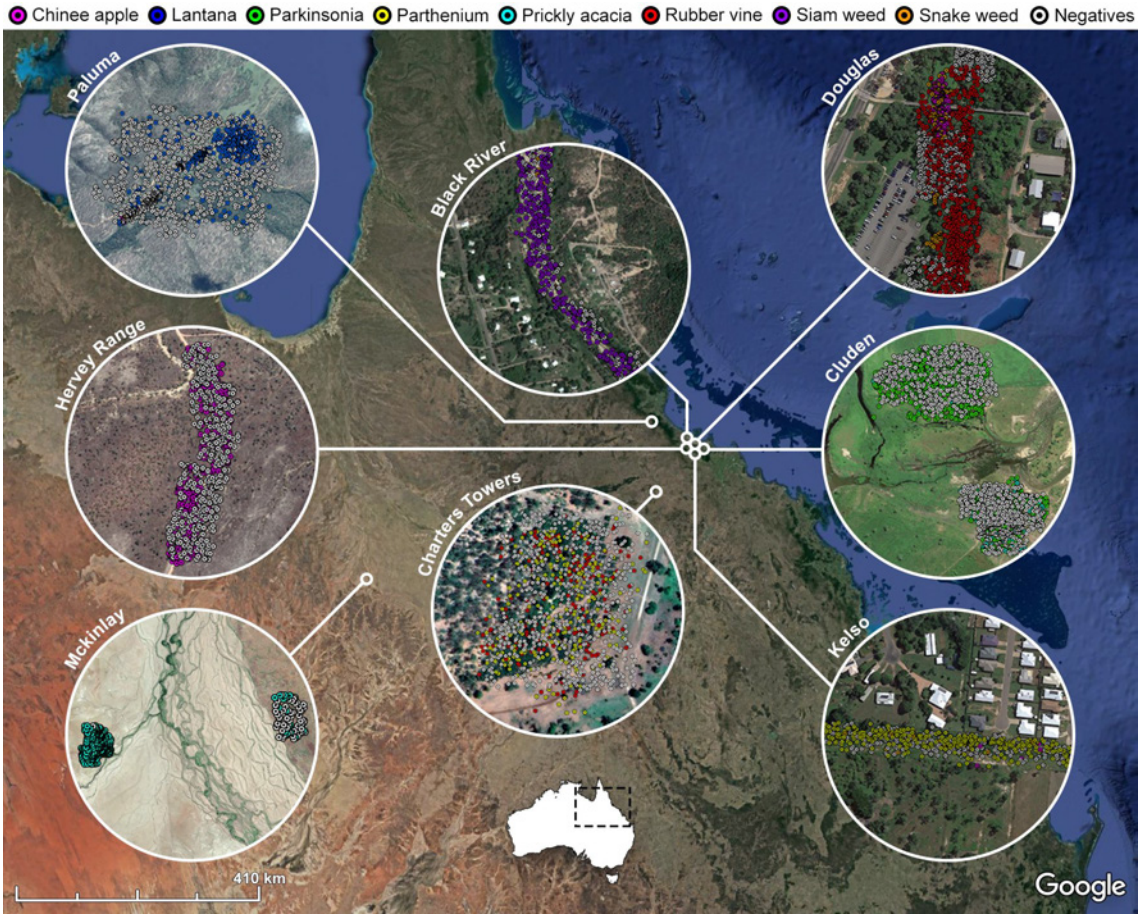


Figure 3.7: The geographical distribution of *DeepWeeds* images across northern Australia (Data: Google, SIO, NOAA, U.S. Navy, NGA, GEBCO; Image © 2018 Landsat / Copernicus; Image © 2018 DigitalGlobe; Image © 2018 CNES / Airbus).

3.3.4 *DeepWeeds*

From June 2017 to March 2018, images were collected from sites across northern Australia using the *WeedLogger* in-field instrument. The result is *DeepWeeds*, a large multiclass dataset comprising 17,509 images of eight different weed species and various off-target (or negative) plant life that have naturalised in Australia.

Table 3.1: The distribution of *DeepWeeds* images by weed species (row) and location (column).

	Black River	*Charters Towers*	*Cluden*	*Douglas*	*Hervey Range*	*Kelso*	*McKinlay*	*Paluma*	Total
Chinee apple	0	0	0	718	340	20	0	47	1125
Lantana	0	0	0	9	0	0	0	1055	1064
Parkinsonia	0	0	1031	0	0	0	0	0	1031
Parthenium	0	246	0	0	0	776	0	0	1022
Prickly acacia	0	0	132	1	0	0	929	0	1062
Rubber vine	0	188	1	815	0	5	0	0	1009
Siam weed	1072	0	0	0	0	0	0	2	1074
Snake weed	10	0	0	928	1	34	0	43	1016
Negatives	1200	605	1234	2606	471	893	943	1154	9106
Total	2282	1039	2398	5077	812	1728	1872	2301	17509

Table 3.1 shows that the two performance targets have been met. All species have over 1,000 images collected with Chinee apple as the most abundantly collected with 1,125 images. There is also a total of 9,106 negative class images collected. Examining each location, it can be seen that at least a 1:1 ratio of positive to negative images has been collected at each site. This will ensure that location bias is mitigated in the development of classification models.

The dataset was published [97] and made publicly available on GitHub[1]. As of October 2019 and since its publication in February 2019, the work has been accessed 5,155 times on Nature.com. The dataset is also viewed approximately 387 times per month by 93 unique visitors and has 28 stars on GitHub.com. Furthermore, in August 2019, the *DeepWeeds* dataset was added to TensorFlow's official dataset catalog [98] for release v1.2.0 and beyond. It is now one of 70 image datasets worldwide that are hard-coded into the TensorFlow machine learning library for end users to train, test and benchmark their algorithms against. This ensures a wide base of machine learning experts will be able to try their hand at classifying the unique weed species image dataset.

[1] *DeepWeeds* is publicly available on GitHub at: https://github.com/AlexOlsen/DeepWeeds

3.4 Chapter Summary

This chapter has reviewed the current landscape of datasets presented for weed species detection in both the image and spectral domains. The common pitfalls in data acquisition (gleaned from the literature) have been mitigated in devising a strategy for dataset collection that will lead to the development of successful real-time detection models. This chapter has made the following contributions:

- Introduced an image acquisition and dataset collection instrument, *WeedLogger*, for fast in-field image collection of weed species.
- Developed custom labelling software for fast curation of weed species images for multiclass classification.
- Introduced DeepWeeds, a 19,507 labelled image dataset of eight weed species from eight different locations across northern Australia.

The latter dataset will serve as the basis for the subsequent investigation into the use of deep learning for real time weed species detection, as discussed in Chapter 4.

Chapter 4

Deep Learning for Weed Detection

This chapter begins with an examination of the current literature behind the application of deep learning to real-time detection and classification of weed species for robotic weed control in complex environments. A case study is presented to illustrate the power of automatic feature selection presented by deep learning in comparison to the traditional image analysis techniques investigated in Chapter 2. State-of-the-art performance in weed species classification accuracy is presented with a cross-section of cutting edge deep CNNs on the DeepWeeds dataset presented in Chapter 3. The real-time speed and power utility of these networks are then analysed to pursue their efficacy as the detection system for a weed control robot.

4.1 Background

4.1.1 Weed Detection

The automatic classification and detection of plant species using computer vision algorithms is an important academic and practical challenge [76]. Solving this challenge requires quantifying image features of the target plants that represent their visible features [30–37]. A variety of algorithms and methods have been developed to solve this problem [99]. Traditional approaches include hand-crafted image features that exploit shape, colour and texture characteristics to describe certain weed and plant species. One such technique, showed that leaf texture discrimination resulted in 86% accurate classification of *Lantana camara* (lantana) as shown in Chapter 2 and [76]. Another study [30] used the morphological features of leaves to accurately differentiate between 32 different plant species found on the Nanjing University campus and Sun Yat-Sen arboretum in Nanking, China. Bruno et al. [56] extracted complex venation leaf patterns (Figure 4.1) and described them with fractal dimensions in order to achieve approximately 98% accuracy classifying ten different plant species from the Brazilian Atlantic forest and Brazilian Cerrado scrublands. All such traditional methods require a human expert to identify which human-literate features discriminate species, then craft a complex mathematical descriptor of that feature in order to differentiate between the species.

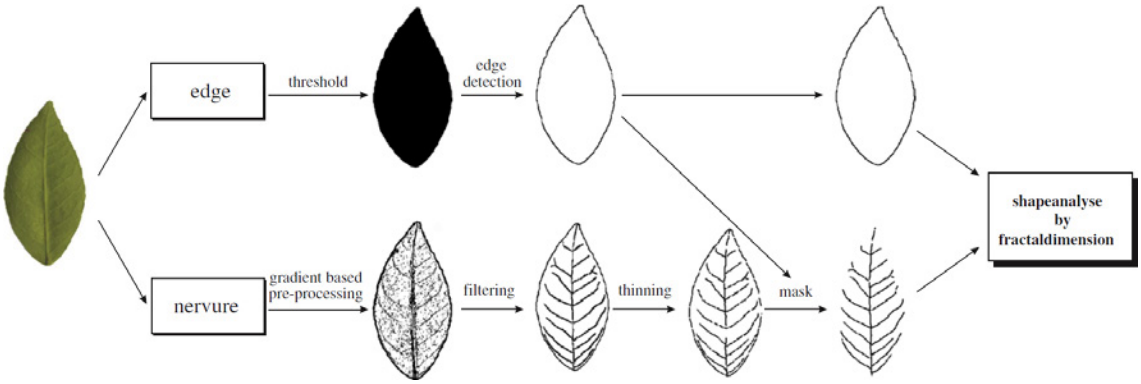


Figure 4.1: Stages of the leaf venation segmentation process outlined in [56] prior to application of the fractal dimension, illustrating the depth of complexity of traditional image analysis models for plant species classification.

Convolutional neural networks (CNNs) on the other hand, present a more palatable approach whereby a mathematical model can be configured to automatically extract image features and optimise which features best discriminate labelled images via a feedback mechanism. All the most promising recent leaf-classification methods are based on deep learning models, such as CNNs [34–36, 38]; which now dominate many computer vi-

sion related fields. For example, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has been dominated by CNN variants since 2012 when a CNN [100] outperformed all other techniques for the first time, and by a wide margin. This and other recent successes behoove the use of deep learning for weed species detection and classification for the highest possible detection accuracy.

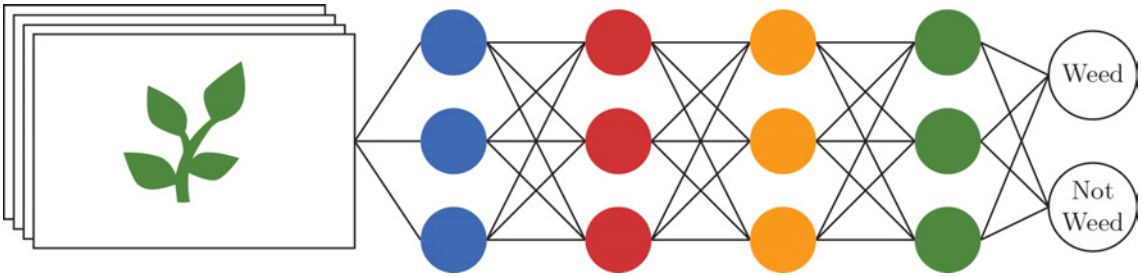


Figure 4.2: A crude illustration of the structure of a CNN that takes an input image through a series of convolutional layers (coloured dots) to produce a labelled output.

CNNs work with the image as a raw input. The image is passed through a series of convolutional filters of varying size and shape. Convolving an image with different convolutional filters (or kernels) extracts different image features. CNNs essentially optimise the convolutional filter coefficients until the extracted image features best describe the class labels associated with the raw images used for training, as illustrated in Figure 4.2.

Irrespective of the features being used, the performance of every detection model is bound by the dataset it is learning. The literature boasts many weed and plant life image datasets [30, 31, 36, 76]. The annual LifeCLEF plant identification challenge [89–91] presented a 2015 dataset [89] composed of 113,205 images belonging to 41,794 observations of 1,000 species of trees, herbs and ferns. This sprawling dataset is quite unique, with most other works presenting site-specific datasets for their weeds of interest [30, 36, 76]. These approaches all deliver high classification accuracy for their target datasets. However, an obvious shortcoming is that most datasets capture their target plant life under perfect lab conditions [30,36]. While the perfect lab conditions allow for strong theoretical classification results, deploying a classification model on a weed control robot requires an image dataset that photographs the plants under realistic environmental conditions, like *DeepWeeds* in Chapter 3. There is an inherent difficulty classifying weed species in situ, as opposed to in the lab (Figure 4.3).

Figure 4.3: A comparison of the inherent difficulty associated with classifying weed species in the lab versus in the field. (a) An image of a lantana leaf taken in a controlled lab environment. (b) A sample image of lantana from the *DeepWeeds* dataset, taken in situ of an entire plant.

Another important feature of any weed detection model is the relative speed of its computation. When implemented in real-time on a weed control platform, the model must be able to *infer* a result from an image fast enough so that it is finished before 1) the next image in sequence arrives and 2) the weed target identified in the image passes outside the control range of the robot. Traditional techniques vary tremendously in complexity and inference speed depending on the quantity and length of the feature descriptors being utilised. The detection model in Chapter 2 was uniquely constructed to reduce inference speed by using texture as a sole feature discriminator. Deep CNNs involve computation of highly complex structures which, until recently, have prevented their utility for real-time tasks. However, new techniques for reducing the computation of network architectures and new more powerful hardware have emerged to make deep CNNs the now dominant approach in terms of real-time performance and classification accuracy.

It is notable that the majority of current weed species classification methods lean towards weed control in cropping applications [34, 101–103], where classification using machine vision is simplified because the land is often flat, the vegetation is homogeneous and the light conditions can be controlled. Classification of weeds in more complex rangeland environments, however, has been largely ignored. These complex environments pose unique challenges for weed management and classification because they tend to be remote and extensive, with rough and uneven terrain, and present complex target backgrounds. Furthermore, many different species of weeds and native plants may also be present in the same area, all at varying distances from the camera, each experiencing different levels of light and shade, with some weeds being entirely hidden. To allow the classification methods deployed in this environment to be successful, site-specific and highly variable weed species image datasets, like *DeepWeeds* (Chapter 3), are needed.

4.1.2 Deep Learning

The dominant tools of modern computer vision recognition are deep CNNs [80]. Numerous works have shown its applicability in various domains [93, 100, 104], including plant and weed species detection [105]. This chapter presents the first application of deep learning for robotic weed control in complex environments that, when imaged, encompass immense variability. There is, therefore a requirement to review the history of deep learning architectures to determine the best approach to solve this challenge.

The First CNN

LeNet [106] is widely held as the foundational CNN for modern computer vision. Its pioneering thesis was that better pattern recognition systems can be built by relying on more automatic learning, and less on hand-designed heuristics. This was achieved by adapting the techniques of gradient based learning and back-propagation to create a neural network that learns from raw input data. This brute force computational architecture was made possible because of recent progress in machine learning and computer technology. The fruit of their labour was, LeNet-5, a groundbreaking 7-layer CNN that classified handwritten digits on banking checks with 82% accuracy. The architecture for this network is illustrated in Figure 4.4 below. Their proposal of back-propagation to learn kernel coefficients directly from images made learning fully automatic and widely applicable to broad recognition problems. The LeNet architecture constitutes five alternating layers of convolution and pooling, followed by two fully connected layers.

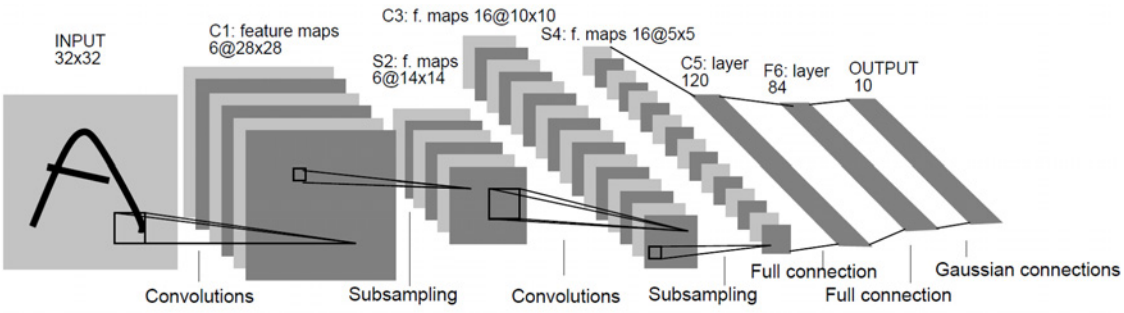


Figure 4.4: An illustration of the LeNet-5 CNN architecture with raw hand drawn digit image inputs and a sequence of seven layers with trainable parameters [106].

Going Deeper

The first *deep* CNN is historically noted to be AlexNet [100], a variant of the LeNet architecture that improved performance by exploiting depth and introducing regularisation

through the use of dropout. AlexNet was the first CNN to win the ILSVRC, an international large scale image recognition competition, in 2012 – and did so by a margin of more than 10.8% over the closest runner up. This was a significant leap forward in the history of computer vision recognition which showed that the depth of the model was essential for high performance, and to be taken advantage of, required the utilisation of high powered GPUs during training.

Explosion of Innovation

The step improvement by AlexNet showed what was achievable in driving computing hardware further and pushing model complexity to its limits. This advancement in computing power however, was paralleled at this time by innovations and new ideas in CNN architectural development. Starting in 2014, the quality of network architectures significantly improved with new ideas for making networks deeper and wider without sacrificing computational complexity. VGGNets [79] pushed layer depth to the extreme in 2014. Network-in-Network increased model complexity with little computation cost by exploiting smaller convolutions [107]. The Inception architecture [92, 104] creatively deployed wider layer blocks with many parallel convolutions. Residual networks like, ResNet [93] and ResNeXt [94], allowed substantially deeper models to be trained with a unique residual learning framework. Light weight networks, like MobileNets [108, 109], utilised depthwise separable convolutions to train hyper efficient networks for edge computing.

As a result of this innovation, edge computing has become possible for more and more applications. This chapter will look to make use of the state-of-the-art CNN architectures based on their optimal suitability to the task of robotic weed control.

4.1.3 Goals of Chapter

1. Demonstrate the superiority of deep learning methods compared to traditional image analysis techniques for weed species detection with a site-specific lantana dataset (Chapter 2).

2. Deploy CNN architectures for multiclass classification of the DeepWeeds dataset (Chapter 3) and determine which CNN architecture, or set of architectures, is most suited to weed species classification.

3. Determine the best performing CNN architecture in terms of speed and power consumption across a variety of hardware platforms. Consequently, uncover the ideal CNN architecture in terms of detection accuracy and real-time performance for implementation in a robotic weed control prototype (Chapter 5).

4.2 Deep Learning Case Study

The first deep CNN to truly exploit modern computational resources, AlexNet, will be deployed on our site-specific lantana dataset of Chapter 2 to demonstrate the efficacy of deep learning in detecting weed species in complex rangeland environments.

4.2.1 Site-Specific Lantana Dataset

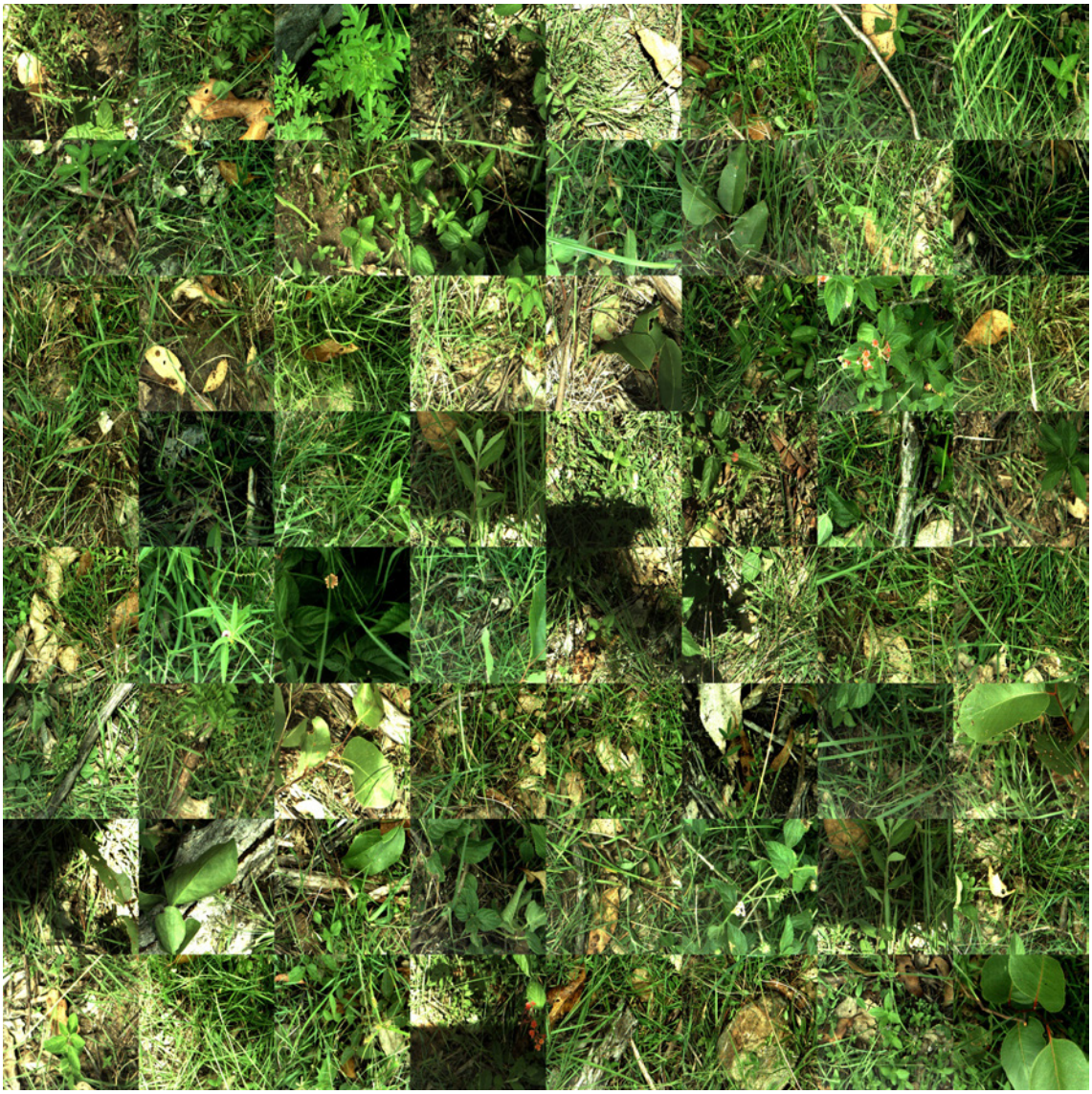


Figure 4.5: A sampling of images from the lantana site-specific dataset resized to 256 × 256 px as input for deep CNNs.

Chapter 2 introduced a challenging in situ image dataset of lantana under realistic and complex conditions. The texture of the leaves was quantified using a rotation and scale invariant adaptation of the HoG feature descriptor to classify the dataset with 86.07 % accuracy [76]. This dataset was made publicly available and will be utilised here to compare performance of this traditional feature set with deep learning. The dataset has been resized to 256×256 px to be more appropriate for the input shape required for most deep CNN architectures, which require less image resolution than texture recognition algorithms (Figure 4.5).

The dataset is not shared with pre-defined training and testing image subsets. Instead, the images have been used in a k-fold cross-validation configuration for training, validation and testing with $k = 10$. To allow a direct comparison to the results of Chapter 2, AlexNet will be deployed on the dataset using the same 10-fold cross validation scheme. The images are therefore partitioned into ten random splits of 80% training, 10% validation, and 10% training. Cross-validation ensures that all images will be exclusively present in the test set throughout the ten simulations. There is an approximate 1:3 distribution of negative to lantana image classes for this dataset (Table 4.1).

Table 4.1: Distribution of images within the lantana dataset after partitioning into training, validation and testing subsets.

Class	**Label**	**Train**	**Validation**	**Test**	**Total**
0	Negative	202	26	26	254
1	Lantana	67	8	8	83
Total		269	34	34	337

4.2.2 AlexNet Architecture

The AlexNet architecture [100] is a variant of the LeNet architecture [106] with five convolutional layers and three fully connected layers, as illustrated in Figure 4.6. The first convolutional layer filters the $227 \times 227 \times 3$ input image with 96 kernels of size 11×11 strided by 4×4 and 0×0 padding. This produces a tensor output shape of $55 \times 55 \times 96$ which is then normalised with a custom local response normalisation scheme. The response-normalised output is then passed through a max pooling layer, with overlapped pooling using 3×3 kernels spaced 2×2 pixels apart, producing a response-normalised and pooled output of shape $27 \times 27 \times 96$. The second convolutional layer then filters this output with 256 kernels of size 5×5 strided by 1×1 and 2×2 padding providing an output of size $27 \times 27 \times 256$. The output is then response-normalised and pooled, as before, resulting in an output shape of $13 \times 13 \times 256$.

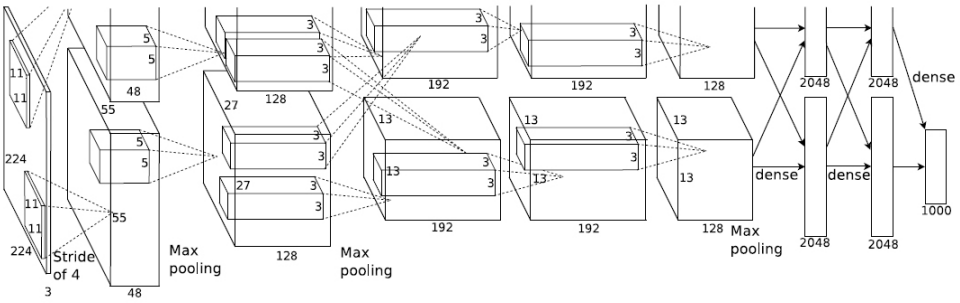


Figure 4.6: An illustration of the architecture of AlexNet whose structure is split into two layer streams in order to be executed on multiple GPUs [100].

The third convolutional layer filters this output with 384 kernels of size 3×3 strided by 1×1 with 2×2 padding to produce an output of $13 \times 13 \times 384$. This output is then filtered by the fourth and fifth convolutional layers with 384 3×3 sized kernels and 256 3×3 sized kernels, respectively. No normalisation or pooling interconnects the third, fourth and fifth convolutional layers. This results in an output of $13 \times 13 \times 256$ from the fifth layer which is max pooled using 3×3 kernels spaced 2×2 pixels apart producing a $6 \times 6 \times 256$ shaped output.

The output of the convolutional layers is then fed through two fully connected layers with 4096 neurons, each followed by a 0.5-probability dropout layer. The final fully connected 4096×1 output is then fed through a fully connected softmax output layer with 1,000 output neurons to match the ImageNet class count.

Input Shape

There is an error in the original AlexNet paper [100] concerning the reported input image size for the network. Krizhevsky et al. report the network accepts image sizes of $224 \times 224 \times 3$. However, the first convolutional layer has a depth of 96 and uses kernels of size $F = 11$ strided by $S = 4$ with $P = 0$ padding. With an input volume size of $W = 224$, the output volume size (O) of the first convolutional layer is shown to be,

$$O = \frac{W - F + 2P}{S} + 1 = \frac{224 - 11 + 2 \times 0}{4} + 1 = 54.25, \tag{4.1}$$

which is not an integer, and therefore surely not correct. The architecture shown in Figure 4.6 shows the output tensor shape of the first convolutional layer to be $55 \times 55 \times 96$. Let us reverse Equation 4.1 and calculate the expected input shape based on the reported output shape as shown in Equation 4.2.

$$W = S(O - 1) + F - 2P = 4 \times (55 - 1) + 11 - 2 \times 0 = 227 \tag{4.2}$$

Hence, the correct input shape accepted by the AlexNet architecture is $227 \times 227 \times 3$. It is assumed that input images were zero padded by an extra three pixels in [100] despite being explicitly mentioned. Consequently, we will be resizing the lantana dataset input images to 227×227 for our implementation of AlexNet.

Multiple GPUs

In 2012, computing hardware was less advanced and the network computations had to be split over two parallel layer streams which could then employ two GPUs simultaneously (Figure 4.6). Even with this design, the network took between five and six days to train ImageNet on two NVIDIA GTX 580 3GB GPUs. When implementing this architecture today, the network can be deployed on a single, more powerful GPU without having to split the layers. Our proposed implementation of AlexNet (Figure 4.7) has a single, traditional layer stream for execution on a single GPU, an NVIDIA GeForce 1080. Training the modified AlexNet architecture for 100 epochs on the lantana dataset took approximately ten hours.

Data Augmentation

To reduce overfitting, Krizhevsky et al. [100] employed two forms of data augmentation to artificially enlarge the dataset using label-preserving transformations. Firstly, the training data is augmented for translation invariance by generating image translations and horizontal reflections. 1,024 random 224×224 patches and their horizontal reflections are extracted from the downsampled 256×256 training subset of ImageNet images. This increases the size of the training set by a factor of 2048. Despite the resulting training subset being highly interdependent, without this scheme, AlexNet resulted in substantial overfitting of ImageNet. The second form of augmentation consisted of altering the pixel intensities of RGB channels in the training subset to bolster invariance of the model to illumination.

The same technique for artificially enlarging the training subset was employed here on the lantana dataset to evaluate AlexNet as its authors had intended. Each training and validation subset image was augmented into 25 random translations and their 25 horizontal reflections, effectually increasing the size of the training subset by 50. The held out test subset for cross validation, was not augmented at all. It was found that this data augmentation reduced overfitting and improved accuracy on the heldout test subset by 4.5%. The second technique for increasing the illumination invariance of the network was achieved with custom image pre-processing techniques, as is further discussed in Section 4.2.3.

Normalisation

The original AlexNet paper uses a custom local response normalisation algorithm to aid generalisation and reduce overfitting. The scheme denotes, $a^i_{x,y}$, the activity of a neuron computed by applying kernel i at position (x, y) and then applying the Rectified Linear Unit (ReLU) nonlinearity. The response-normalised activity, $b^i_{x,y}$, is given by,

$$b^i_{x,y} = \frac{a^i_{x,y}}{(k + \alpha \times \sum_{j=\max(0,i-\frac{n}{2})}^{\min(N-1,i+\frac{n}{2})} (\alpha^j_{x,y})^2)^\beta}, \tag{4.3}$$

where the sum runs over n adjacent kernel maps at the same spatial position and N is the total number of kernels in the layer. The hyperparameters were validated on the ImageNet dataset to be $k = 2, n = 5, \alpha = 10^{-4}$ and $\beta = 0.75$.

With this normalisation strategy, the AlexNet model was found to overfit the lantana training dataset. This was apparent because the validation loss began to increase as the training loss continued to decrease, indicating the model is getting better and better at predicting the training dataset to the detriment of performance on the validation dataset; and subsequently, any other generalised dataset. Consequently, the method of batch normalisation [110] was adopted and applied to the AlexNet structure. Batch normalisation was proposed by Ioffe and Szegedy [110] to reduce internal covariate shift; a phenomenon whereby the distribution of each layer's inputs changes during training making it notoriously hard to train models with saturating nonlinearities. Batch normalisation solves this problem by normalising layer inputs for each training mini-batch.

Dropout

AlexNet is also the first deep CNN architecture to demonstrate the benefit of dropout, a simple technique to prevent neural networks from overfitting [111]. Deep neural networks have an incredibly large number of trainable parameters. This increases what is known as the complexity of the model. Generally, the higher the model complexity, the larger the optimisation space that model has to fit to a complex dataset. Sometimes the model can overfit to a dataset, whereby the neurons and weights can co-adapt to learn the images being trained. Dropout prevents this co-adaption by randomly "dropping" neurons (along with their connections), with a given probability, from the neural network during training.

AlexNet employs dropout layers immediately following the two hidden fully connected layers with 50% dropout probability. Our implementation of the architecture is unchanged in its use of dropout.

Output Layers

The output dense layers of the AlexNet architecture consist of three fully connected layers. The first two have 4,096 kernels and apply the ReLU activation function. When ReLU is used as the activation function in each hidden layer of a network, deep CNNs train several times faster than their equivalents with tanh units [100]. The final output layer consists of a 1,000 kernel softmax activated fully connected layer. Softmax is used to provide a probability distribution over the 1,000 class labels which correspond to selected classes from the ImageNet competitive dataset.

For our implementation of the AlexNet architecture, Figure 4.7, we use the same 2,048 kernel, ReLU activated fully connected layers. However, the final 1,000 neuron fully connected layer is replaced by a two neuron softmax activated fully connected layer in order to produce a distribution of probability for the lantana and negative classes of our binary classification problem.

Revised Architecture

Implementing the aforementioned methodology we arrive at our revised AlexNet architecture for weed species classification, see Figure 4.7. This revised architecture has 56.91 million network parameters compared to the 56.87 million parameters of the original.

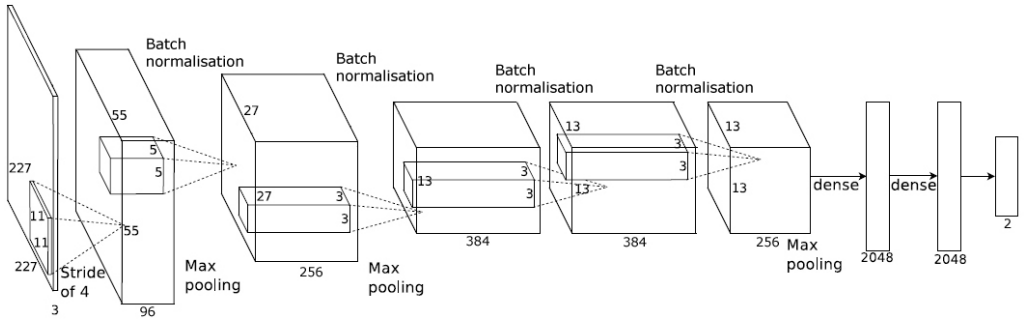


Figure 4.7: A single GPU revision of the AlexNet architecture for implementation on the site-specific lantana dataset.

4.2.3 Training Regime

This subsection details the training methods and algorithms used for deploying the revised AlexNet CNN to the lantana dataset.

Transfer Learning

When training an entire deep CNN *from scratch* (i.e. with random initialisation) one requires a dataset of sufficient size to match the complexity of the model. If a model is more complex than the dataset being learned is large, the model will quickly overfit and produce a model which does not generalise well to practical implementation. This gives rise to *transfer learning*, where a CNN is pretrained on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1,000 categories) and then used as an initialisation for a new learning task and new dataset (Figure 4.8).

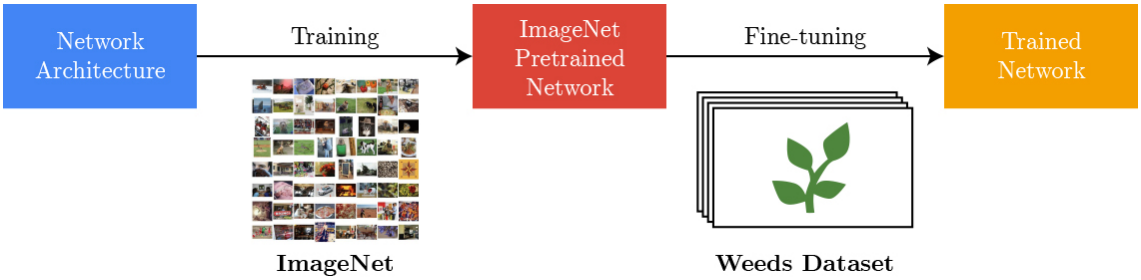


Figure 4.8: Illustration of the transfer learning process where a pretrained network is adapted for a new dataset and learning task.

When a dataset as large and varied as ImageNet is used, the pretrained CNN features exhibit great generality that can be mapped to almost any other learning task. This methodology also has the benefit of reducing the total required training time of the CNN, allowing highly complex models to be trained for new learning problems in a matter of hours instead of weeks.

Our focus is to train a deep CNN to our DeepWeeds dataset for the purpose of delivering a real-time detection model for a weed control robot. In deciding whether to use transfer learning or perform entirely specific hyperparameter optimisation we must consider 1) the dataset size, 2) the nature of the learning problem and 3) the speed requirement for training. The DeepWeeds dataset contains 17,509 images. This is quite small relative to the 1,431,467 image ImageNet dataset, comprising just over 1% of its total. A smaller dataset is unlikely to exhibit the variability required to generalise performance of a complex CNN to its learning problem. Transfer learning in this case would allow the general features of an ImageNet pretrained model to be mapped onto the DeepWeeds dataset during initialisation, then further training of the CNN would adapt the CNN to classifying the DeepWeeds images. This is only possible if the nature of the learning problem is somewhat similar, so that the general features of the pretrained model can be used in the transfer problem. The more general image features (i.e. lines, edges and gradients)

are extracted from the earlier layers of a CNN while the later layer features become more specific (i.e. shapes and patterns) (Figure 4.9). Because the fundamental image features present in ImageNet (i.e. lines, edges and gradients) must be present in DeepWeeds, transfer learning is an applicable approach.

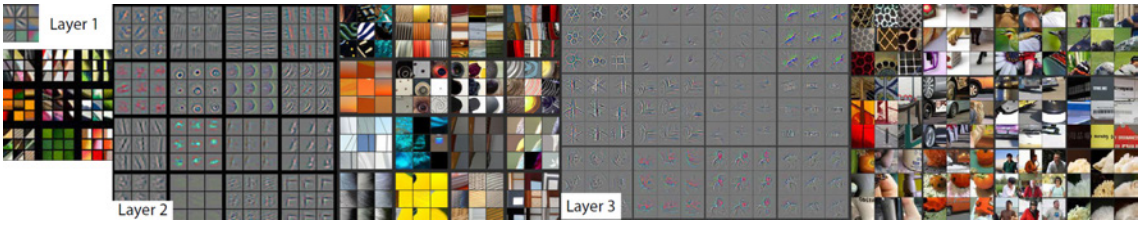


Figure 4.9: Visualisation of the weights of subsequent layers of a deep CNN illustrating that earlier layers exhibit simpler features and deeper layers, more complex features [112].

A dichotomy of approaches then emerges within transfer learning: 1) train all network parameters or 2) freeze certain lower level parameters and only train (or *fine tune*) the top-level parameters. Fine tuning would even further reduce the required training time by limiting the number of trainable parameters. However, this approach is only recommended in scenarios where the new dataset and learning problem is very similar to the original dataset. Although the image characteristics between DeepWeeds and ImageNet may be similar (i.e. lines, edges) we reason that the learning problem is much more specific with only nine classes compared to 1,000 classes. Therefore, all network weights should be trained after being initialised with pretrained ImageNet weights.

Image Pre-Processing

To overcome the highly variable nature of weed classification, a series of augmentations were performed on both the training and validation image subsets to account for variations in rotation, scale, colour, illumination and perspective. Image augmentation was performed using Keras' [113] preprocessing library and OpenCV [72]. All images were first resized to 256×256 pixels in size and randomly augmented for each epoch of training, *i.e.* one pass through all available training and validation images. Each image was also randomly rotated in the range of $[-360, +360]$ degrees. Then, each image was randomly scaled both vertically and horizontally in the range of $[0.5, 1]$. Each colour channel was randomly shifted within the range of ± 25 (i.e. approximately $\pm 10\%$ of the maximum available 8-bit colour encoding range $[0, 255]$). To account for illumination variance, pixel intensity was randomly shifted within the $[-25, +25]$ range, shifting all colour channels uniformly. In addition, pixel intensity was randomly scaled within the $[0.75, 1.25]$ range.

Random perspective transformations were applied to each image to simulate a large variation of viewing distances and angles. Finally, the images were flipped horizontally with a 50% probability and then cropped to retain the 227×227 pixels required for the AlexNet architecture's input layer.

This more extensive data augmentation methodology replaces AlexNet's original rudimentary algorithm [100], where only the pixel intensities of the RGB channels are randomly augmented. More thorough augmentation is required due to the small size of the lantana dataset being considered.

Activation and Loss Function

The output of the AlexNet architecture is fed through a Softmax activation layer [114]. The Softmax model is commonly used to apply logistic regression to multinomial problems. Softmax regression determines a discrete probability distribution of each class, p_i for the output of the final layer in our revised AlexNet architecture,

$$p_i = \frac{e^{y'_i}}{\sum_{i=1}^{N} e^{y'_i}}, \tag{4.4}$$

where y'_i represents the predicted class. In addition, $\sum_{i=1}^{N} e^{y'_i} = 1$ where N is the number of classes to be distinguished. Softmax regression is commonly used in tandem with cross-entropy loss to enable the network to learn different classes during backward propagation. As shown in Equation 4.5,

$$\lambda = \frac{-1}{N} \sum_{i=1}^{N} y'_i \times \log(y_i) + (1 - y'_i) \times \log(1 - y_i), \tag{4.5}$$

where λ is the computed loss and y_i is the class label.

Stochastic Gradient Descent

The Keras implementation of Adam [115], a first-order gradient-based method for stochastic optimisation, was used to optimise the cross entropy loss function. Adam (or Adaptive Moment Estimation) has little memory requirements, is invariant to diagonal rescaling of the gradients and is well suited for problems that are large in terms of data and/or parameters.

Learning Schedule

Typically, fine-tuning a pretrained model using transfer learning requires a smaller initial learning rate because the pretrained model weights have already undergone substantial

optimisation. For this task, the initial learning-rate (η_0) was set to $lr = 1 \times 10^{-4}$. To maximise the network's performance, a decaying learning rate schedule is used during the training process. The learning rate, η, is decayed by a factor of two whenever the learning stagnates. Learning stagnation is defined as every time the *validation loss* did not decrease after 16 epochs. Note that the validation loss refers to the classification error computed on the validation subset of images.

Batch Size and Early Stopping

A batch size was chosen to maximise the training speed to the capabilities of the hardware being used. The revised AlexNet architecture permitted a maximum batch size of 32 images for use on a GeForce GTX 1080 GPU.

To prevent the network from overfitting, an early stopping strategy was employed during training. The overfitting criteria was defined as when the validation loss did not decrease after 32 successive epochs. When this criteria is met, the training is aborted and restarted with the initial learning rate $\eta = 0.5 \times 10^{-4}$. While training the model with the smallest running validation loss is saved in order to restart training.

Training Process

The revised AlexNet architecture was trained on the lantana dataset with the documented training regime for a maximum of 100 epochs with k-fold cross validation ($k = 10$). Each model took an average of 7.92 hours to complete 100 epochs on an NVIDIA GeForce GTX 1080. This resulted in ten independent models being trained and evaluated. It is useful to observe the test and validation accuracy versus epoch throughout the training process to monitor overfitting (Figure 4.10). Similarly, the cross entropy loss versus epoch can shed light on the success of training (Figure 4.11).

The validation loss decreases at the same rate as the training loss, with both reaching an asymptote after 60-70 epochs (Figure 4.11). This verifies that the network is learning the dataset generally well without overfitting. Meanwhile, the accuracy on the test and validation subsets increases with successive epochs until both plateau after roughly 50 epochs eventually reaching up to 100% accuracy on the validation set (Figure 4.10). Some noise is visible in the individual validation accuracy and loss responses. This can be attributed to randomness in the cross validated subsets. In order to evaluate the accuracy of the model, we must assess its performance against the unseen and heldout test subset for each fold of cross validation. For each fold, the epoch with the lowest validation loss was extracted as the optimal model for testing.

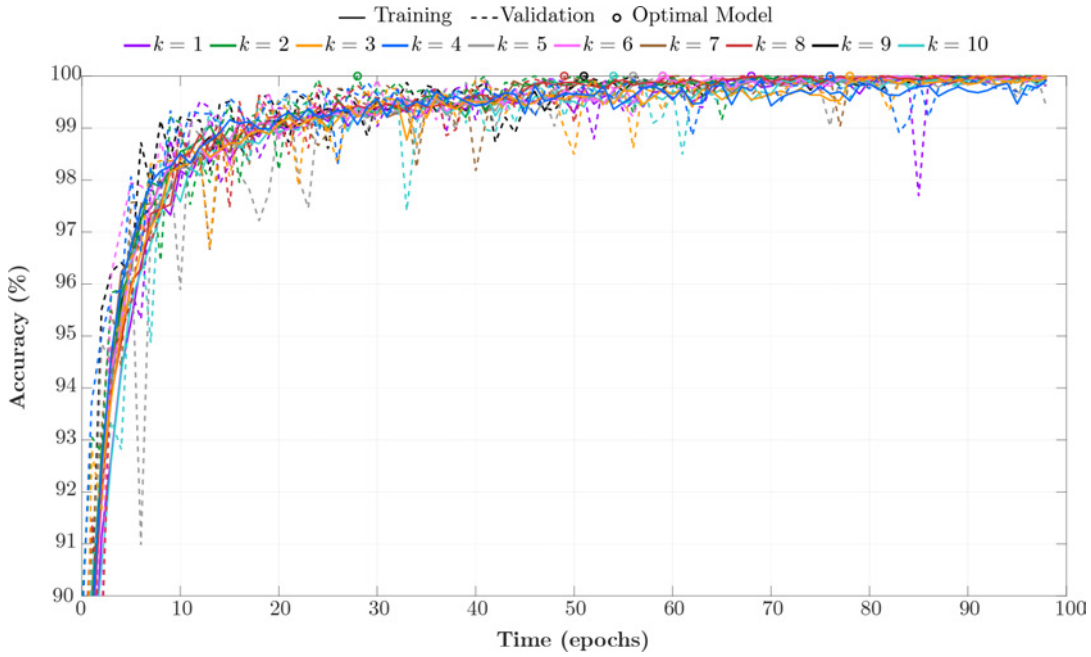


Figure 4.10: Visualisation of the learning process as the training and validation accuracy for each cross validated fold, k, increase with every epoch until reaching a plateau of accuracy after 50 epochs.

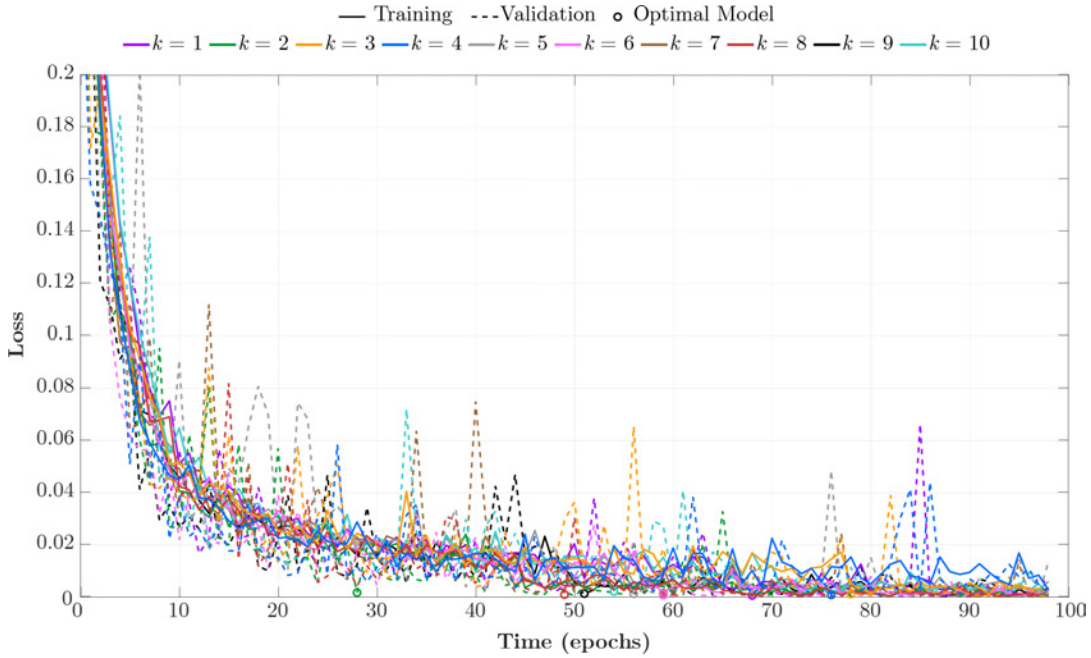
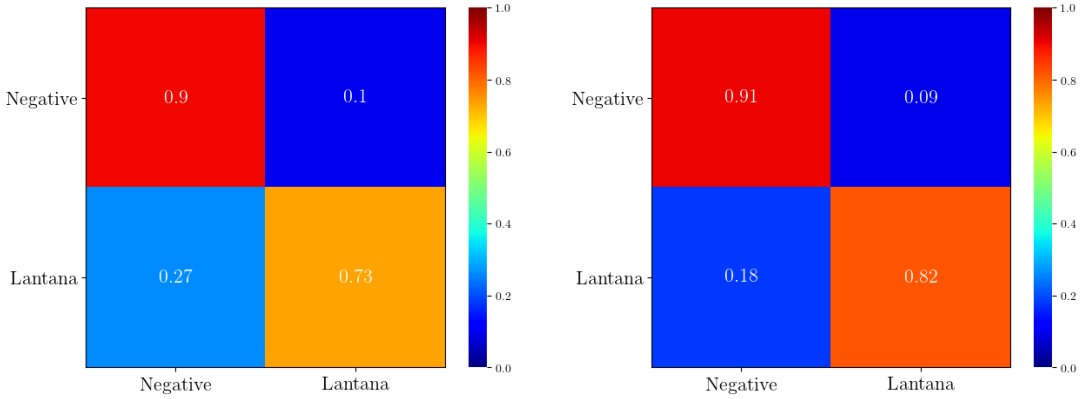


Figure 4.11: Visualisation of the learning process as the training and validation loss for each cross validated fold, k, decrease with successive epochs until reaching an asymptote after 70 epochs.

4.2.4 Classification Performance

Accuracy

Figure 4.12b reveals the inter-class performance of the AlexNet classifier with the average confusion matrix across all ten cross validated folds. AlexNet correctly classifies 91% of the negative class and 82% of the lantana class. This is a strong result. In comparison, the HoG leaf texture image feature set of Chapter 2 achieved 90% classification accuracy on the negative class and 73% classification accuracy on the lantana class. The deep CNN methodology has clearly outperformed the more traditional image classification methodology.



(a) HoG (b) AlexNet

Figure 4.12: A side-by-side comparison of the confusion matrices for the optimal HoG feature set classifier (a) and the superior AlexNet classifier on lantana (b).

Table 4.2: Comparison of the detection performance of AlexNet compared to the Histogram of Oriented Gradients descriptor of Chapter 2 averaged across ten folds of the lantana dataset.

Statistic	**HoG**	**AlexNet**
Accuracy (%)	86.1	88.4
Precision (%)	91.2	93.9
True positive rate (%)	90.2	90.6
True negative rate (%)	73.5	81.9
False positive rate (%)	26.5	18.1
False negative rate (%)	9.8	9.4

Further analysis of the classification performance is detailed in Table 4.2. The AlexNet

model outperforms the HoG feature set methodology in all binary classifier metrics. With a precision score of 93.9%, any positive label produced from the AlexNet model can be confident to within 93.9% certainty. The true positive and true negative rates echo the results for class-specific recall accuracy as indicated in the confusion matrices of Figure 4.12b and 4.12a. Also the slightly better false positive and false negative rates show that the model produces less false predictions than the HoG model.

Feature Analysis

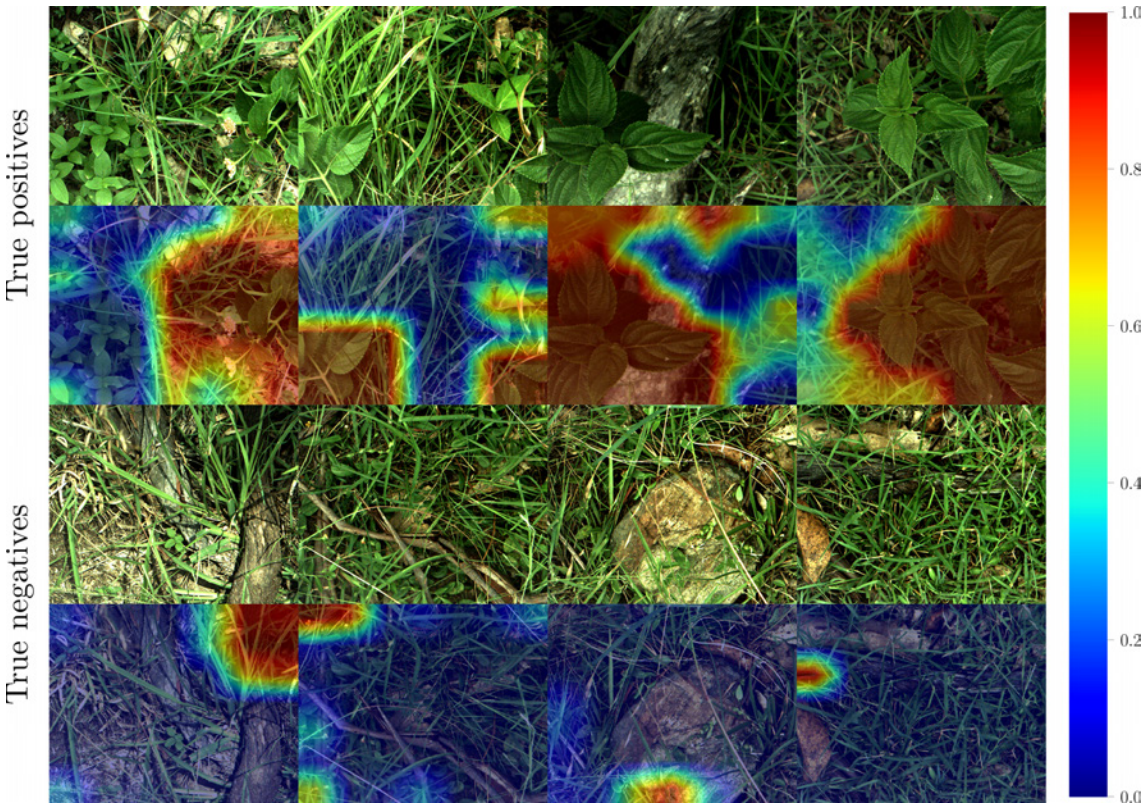


Figure 4.13: Class activation heatmaps for a subset of true positive and true negative images predicted by AlexNet. Red regions of the image indicate lantana features and blue regions indicate negative class features.

The HoG feature set of Chapter 2 was painstakingly handcrafted to exploit the leaf texture of the lantana weed species in order to classify it accurately from its background. Meanwhile, the AlexNet classifier was deployed and automatically learned its feature set with a stronger classification performance. A variety of techniques exist to visualise the image features learned by a deep CNN [112, 116–118]. Class activation maps or *heatmaps* [117] utilise the global average pooling layer to build a localisable representation of an image

passed through a trained CNN that exposes the implicit attention of the classifier. This technique has been adapted to our AlexNet model to represent the salient features being learned by the deep CNN in a similar fashion to that used by Calvert et al. [119] for the detection of Harrisia Cactus.

Figure 4.13 overlays the heatmaps with the original input images for a subset of true positive and true negative predictions from the AlexNet model. It is demonstrated that AlexNet is correctly locating the lantana leaves and distinguishing it from complex background imagery (including grass, wood, rocks and other plant leaves).

4.2.5 Summary

In comparison to the HoG feature set of chapter 2, deep CNNs present a much more robust solution that has the ability to learn which features separate the classes. A major setback of the texture-based HoG classifier developed in Chapter 2 is that it is not adaptable to every weed species. If a weed species were to have a similar leaf texture characteristic it would likely present as a false positive prediction. Even if that weed were to have other visible characteristics distinguishing it from lantana, the texture classifier is blind to such variations because its feature set is predetermined. In contrast, if a CNN was deployed it will automatically learn the differentiable features and thus should be inherently adaptable to all weed species.

The AlexNet architecture proved the superiority of deep learning compared to traditional methods for the task of detecting lantana. However, AlexNet is, by modern standards, a primitive deep CNN architecture. The literature for modern deep CNNs has extended to include many variants delivering much higher accuracies. The following section introduces a suite of new deep CNNs and investigates whether deep learning is capable of distinguishing between multiple weed species exhibiting variable characteristics.

4.3 DeepWeeds Classification

4.3.1 Scope

Can deep CNNs learn to differentiate between weed species with various different image features (shape, size, colour, texture) within extremely variable complex backgrounds? The first step in addressing this question has been made in Chapter 3 with the collection and curation of DeepWeeds, the first, large, multi-class weed species image dataset taken in situ from the northern Australian rangelands. This dataset allows the wider research community to improve classification accuracy on rangeland weed species and improve the viability of robotic weed control solutions.

Here we propose to investigate a cross-section of the most dominant CNN architectures operating today to comprehensively address this fundamental issue with an eye toward enhancing robotic weed control. The relevant literature has grown rapidly in the preceding decade with deep networks [100,106], very deep networks [79], deep and wide networks [92, 104], fully connected networks [120], residual networks [93,94], regional networks [118,121], light-weight networks [108,109], recurrent networks [122] and even amalgamations of these different architectures [123]. However, before choosing a network we must first clearly define our learning problem.

Learning Problem

There are predominantly four computer vision learning tasks: classification, localisation, object detection and segmentation (Figure 4.14). Classification asks if the target is present in an image, localisation tries to find where in the image the target is. Object detection asks where are each of the individual target objects within the image and segmentation tries to predict the exact shapes of said objects.

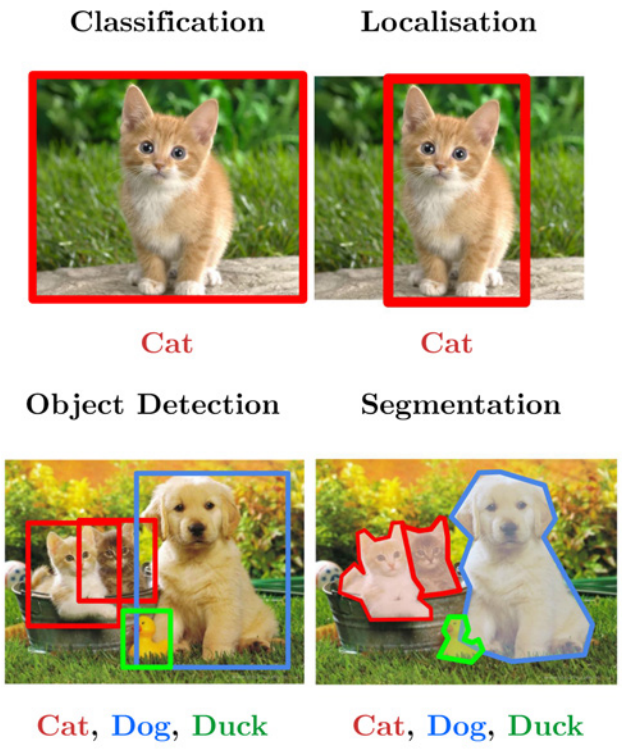


Figure 4.14: Simplistic rendering of the four predominant machine vision tasks in order from the simplest (top-left) to the most difficult (bottom-right).

These four tasks can be adapted to weed species detection to construct four specific learning problems. The tasks get more difficult from classification to segmentation, requiring larger datasets and more computational power.

Classification: What weed species, if any, are present in the image?

Localisation: Where in the image are the weeds?

Object Detection: Where are the locations of each single weed in the image?

Segmentation: What is the exact shape of the weeds in the image?

The litany of deep CNN architectures cited above have been adapted to reach high levels of accuracies across the different learning tasks.

Robotic Weed Control

The question of which task to address for weed detection depends on the precision required for the weed control strategy. In our case, we wish to implement real time weed species detection to activate a control mechanism as the vehicle passes by. This introduces a very strict real time requirement. Also the precision of the weed detection system is multivariately linked to the control being applied. For the prototype development undertaken in Chapters 2 and 5, we choose to apply spot-spraying herbicide application due to its relative ease and speed of electronic application.

Herbicide spray precision is limited to the size of the spray swath being applied. The smallest conventional nozzle spacing benchmarked by the prominent nozzle manufacturer, TeeJet, is 25 cm [124]. This means that the smallest application precision is 250 mm. The aim of this work is to develop weed recognition technology that can be applied readily with available hardware. Therefore, the learning problem will be shaped to fit with the current commonly used narrow sprayjet spacing of 250 mm. As a result, the camera field of view will be matched to this jet spacing.

If a camera mounted to the vehicle has the same 250 mm field of view as the spray swath precision, the most suited learning task for this problem is classification: asking, "is there a weed present in this image?" In this situation, performing object detection to identify the sub-location of weeds within the entire field of view to be sprayed is redundant and adds unnecessary cost to implementation accuracy and speed. However, if a camera viewing window is significantly larger than the precision of a single sprayer, object detection becomes more useful.

Accepting the constraints of real-time implementation, this work proposes to address image classification rather than the more complex and time-consuming object detection

for its robotic weed control solution. For instances where the field of view of the camera is larger than the individual spray precision, the camera field of view can be divided into sub-regions aligned to the individual sprayers. A batch of image classification tasks can then be performed simultaneously to identify which sprayers to trigger for the robotic solution. By framing our implementation to the simpler learning problem, we benefit from having higher attainable classification precision and faster real-time performance.

4.3.2 DeepWeeds Dataset

The DeepWeeds dataset was collected as part of this thesis (Chapter 3). It consists of 17,509 images including eight weed species classes and a highly complex negative class. For this analysis, the dataset has been resized to 256 × 256 px as a more appropriate size for input into the chosen CNN architectures (Figure 4.15).

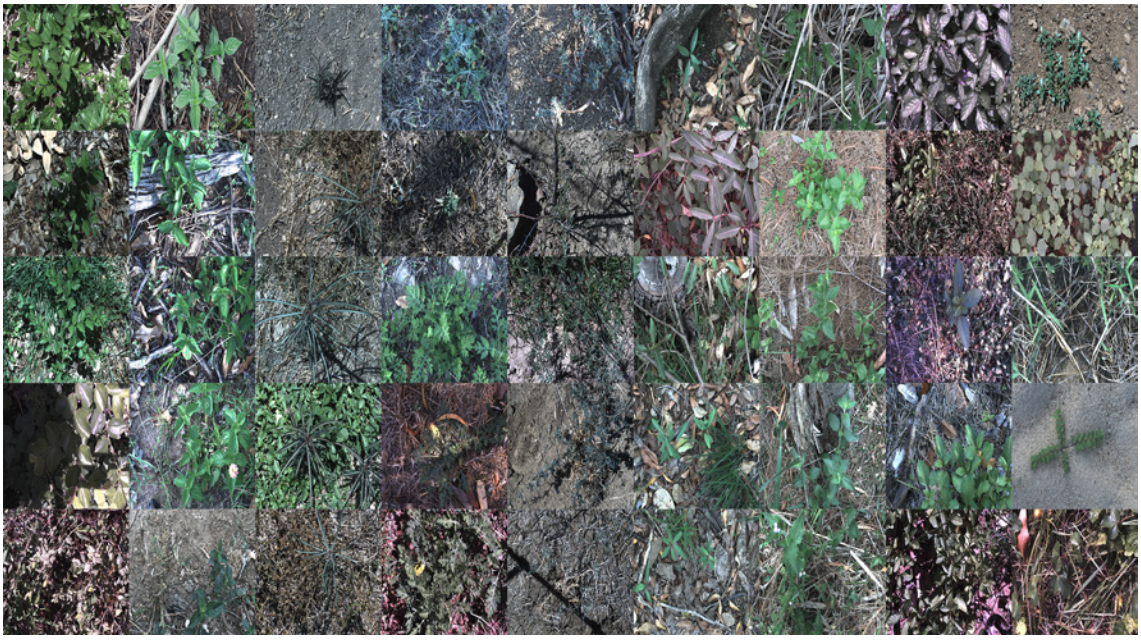


Figure 4.15: A sampling of images from the DeepWeeds dataset resized to 256 × 256 px with each column representing a specific class in the dataset.

To provide a thorough and randomly fair evaluation of the classification performance on the dataset, k-fold cross validation will be used. This involves iteratively partitioning the dataset into five random splits of 60% training, 20% validation, and 20% testing (Table 4.3). The training and validation subsets will be used for tuning of the hyperparameters while the testing subset will be entirely left out for evaluation.

Table 4.3: Distribution of *DeepWeeds* images into training, testing and validation subsets for k-fold cross validation with $k = 5$.

Class	Label	Train	Validation	Test	Total
0	Chinee apple	675	225	225	1,125
1	Lantana	638	213	213	1,064
2	Parkinsonia	617	207	207	1,032
3	Parthenium	612	205	205	1,022
4	Prickly acacia	636	213	213	1,062
5	Rubber vine	605	202	202	1,009
6	Siam weed	644	215	215	1,074
7	Snake weed	608	204	204	1,017
8	Negative	5462	1,822	1,822	9,106
Total		10,497	3,506	3,506	17,509

4.3.3 Network Architectures

The labelled DeepWeeds dataset allows us to benchmark the performance of some of the best CNN classifiers in the literature. The ImageNet competition presents the most challenging and widely participated image classification task on the planet [80]. From it, we can glean which best-performing architectures should be adapted to our weed species learning problem. In selecting these architectures, emphasis was given to off-the-shelf CNNs, which could be easily trained and deployed to facilitate wider use of the presented dataset. To that effect, the high-level neural network Application Programming Interface (API), Keras [113], was utilised; together with the machine learning framework, TensorFlow [125] and NVIDIA's library for optimising TensorFlow for real-time implementation, TensorRT [126].

Five popular CNN models were chosen for implementation based on their strong performance on the highly variable ImageNet dataset and their availability in Keras, TensorFlow and TensorRT. These models include: AlexNet, Inception-v3, MobileNetV2, ResNet-50 and VGG16. This sampling of CNN architectures gives a varied view in terms of network size, nature, speed and accuracy.

The diversity of the selected models is evident in terms of the number of model parameters, layer depth and classification accuracy when deployed on the ImageNet architecture (Table 4.4). AlexNet is much more primitive in its design with only eight layers of depth. Meanwhile, VGG16 presents an incredibly complex model with twice the number of parameters than the next closest model. Inception-v3 and ResNet-50 present competing methodologies of similar layer depth and parameter complexity with the strongest classi-

fication performance on ImageNet. While MobileNetV2 stands for the light-weight deep CNN architectures exhibiting a deep layer count but very slim parameter count due to its sparse convolutional operation.

Table 4.4: Summary of the characteristics and performance of the chosen CNN architectures when trained to the ImageNet dataset.

Model	**Year**	**Parameters (million)**	**Layers**	**Top-1 Accuracy (%)**
AlexNet	2012	60	8	62.5
Inception-v3	2015	25	42	82.8
MobileNetV2	2018	3.4	54	74.7
ResNet-50	2015	26	50	79.3
VGG16	2014	138	16	76.3

Other networks were removed from consideration for a variety of reasons. The 2016 ILSVRC winner, GBD-Net [127], was not considered because it requires within-image per-pixel labels for semantic segmentation. The 2017 ImageNet competition winner, Squeeze-and-Excitation (SE) networks [128] was not considered because the SE augmented models are not readily available in the Keras and TensorFlow backend. The 2019 variant of the residual family of networks, ResNeXt [94], was not considered due to limitations in hardware preventing deployment of such a complex model. The residual *Inception* model, Inception-ResNet [123] was not considered in favour of evaluating the Inception and residual architectures separately.

Now that the models have been chosen, we can review each architecture and decide how they shall be revised for deployment on the DeepWeeds dataset.

AlexNet

The revised AlexNet architecture presented in section 4.2.2 will be used again here and evaluated against the DeepWeeds dataset. The only significant and required change for implementing on the larger DeepWeeds dataset, is adjusting the final output layer for the nine weed species classes present (Figure 4.16). The original 1,000 kernel fully connected layer was adapted to a two kernel layer for the binary lantana classifier. Here, the final layer will have nine output kernels. The layer will also have a Sigmoid activation function to allow multi-label output predictions.

The resulting AlexNet model being implemented for DeepWeeds has 56.9 million network parameters, slightly less than the AlexNet architecture proposed for ImageNet classification. The AlexNet layer structure is simple, despite its high parameter count. We expect the simpler structure to not fare as well as the more complex and deeper structures

being investigated. However, it may have a significant real-time advantage because its simplicity will allow faster inference. Therefore, this network is an ideal one to investigate for robotic weed control if it can achieve useful accuracy.

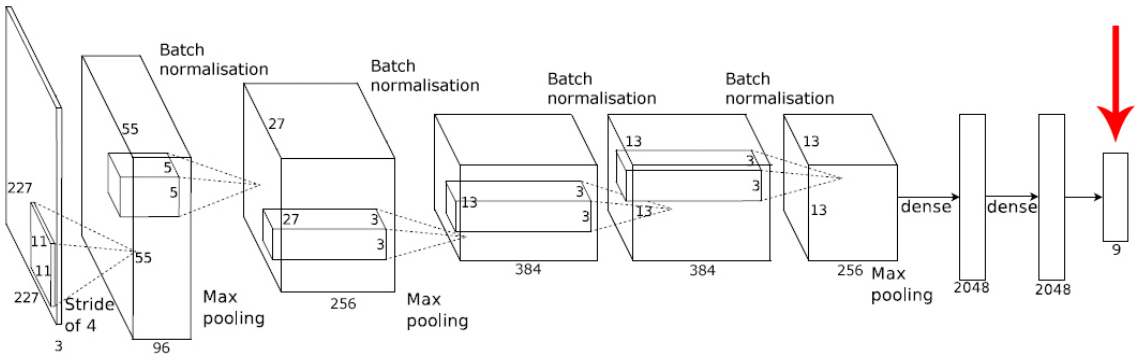


Figure 4.16: Final revision of the AlexNet architecture where the output layer has been replaced with a nine neuron fully connected layer to match the DeepWeeds classes.

VGG16

VGGNets were introduced by the Visual Geometry Group in 2014 [79] while investigating the effect of CNN depth on accuracy for large scale image recognition. VGGNets use the same principles of layer configuration as AlexNet [100] with very small convolution filters allowing the depth of the network to be increased steadily with more convolutional layers.

The input size for this network is a $224 \times 224 \times 3$ shaped image with three RGB colour channels. The input is sequentially passed through a stack of convolutional filters each with 3×3 sized kernels (the smallest possible size to capture horizontal and vertical gradients). For every convolution, the stride of the kernel is 1×1 px and padding is 1×1 px to ensure the spatial resolution of the input is preserved following convolution. Groups of convolutional layers are separated by max pooling layers with a filter size of 2×2 strided by two pixels. The pooling layers act to downsample the convolutional outputs at each depth, revealing more and more abstracted image features for optimisation.

The different variants of the VGGNet architectures have different numbers of convolutional filter stacks before being followed by the fully connected layers (Table 4.5). Every VGGNet architecture has three fully connected output layers: two hidden with 4096 kernels and the third is a 1000 neuron softmax activated layer which produces the network's output probabilities. This structure is simple, taking the primal elements of a CNN (like those that make up all of AlexNet) and pushing them further in terms of layer depth. The VGG16 variation on this architecture has 16 total weight layers including 13 convolutional layers and three fully connected layers.

Table 4.5: VGGNet configurations with increasing layer depth from left (A) to right (E) as more layers are added. *The VGG16 variant of interest is configuration (D) [79].

ConvNet Configuration					
A	A-LRN	B	C	D*	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

For this work we will be utilising the Keras [113] implementation of VGG16 which remains true to the original paper [79]. Several architecture specific revisions will be made to best tailor the network to the DeepWeeds dataset. The first alteration concerns the output layer. The 1,000 neuron ImageNet-specific layer is replaced with a 9-neuron fully connected layer mapping the layer kernels to the nine weed species classes of the DeepWeeds dataset. Next, the original VGG16 paper [79] uses very little data augmentation compared to AlexNet [100]. The only preprocessing performed is normalising the RGB colour channels of input images on the ImageNet training set. For our revision of this architecture we will be applying more significant preprocessing augmentations. The primary reason for this is to increase the variability of our dataset which only numbers 17,509 images.

Experimentation revealed that the base VGG16 implementation took an inordinately

long time to fit to the dataset and eventually began overfitting. One reason for this problem was reasoned to be internal covariate shift, whereby the inputs to each layer vary in scale which adds another level of complexity for the model to optimise. Batch normalisation was instituted to remedy this by normalising layer inputs and removing them from the optimisation process. Batch normalisation has the effect of allowing each layer of a network to learn by itself more independently of other layers. Batch normalisation layers were placed after every convolutional layer in the network. This significantly improved the learning rate of the network.

Inception-v3

The impetus behind the *Inception* deep CNN architecture was to continue the progress made by AlexNet and VGG16 from increasing model size and layer depth, respectively, but in an efficient computational manner such that the models are more applicable to mobile vision and big data tasks [104]. Network-in-Network [107] achieved this by adding small 1×1 convolutional layers to a network to increase its depth, while having a negligible effect on computational power. This improved the representational power of the network with little cost. Meanwhile, GoogLeNet [92], the first Inception model, outperformed both AlexNet and VGGNet by continuing to increase model complexity, but with nine times fewer parameters than AlexNet and more than 20 times less parameters than VGG16. How was this accomplished? The answer to this question lies in the foundational building blocks of the Inception architecture, the Inception modules, as shown in Figure 4.17.

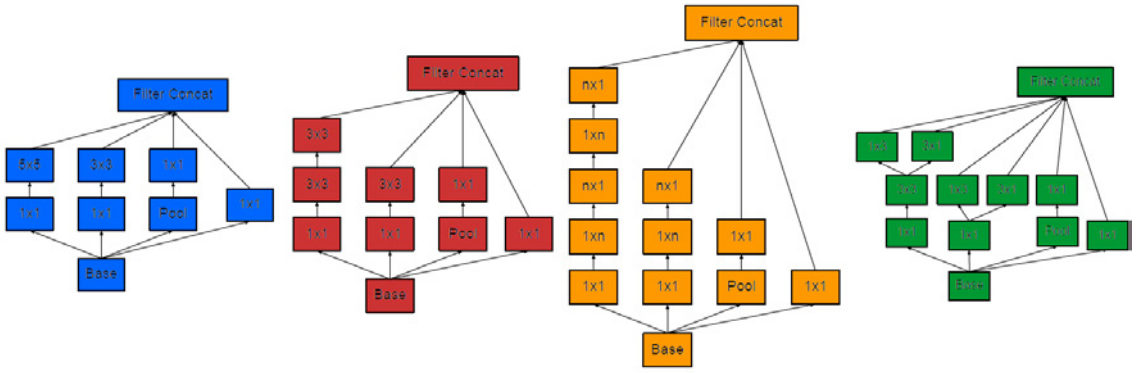


Figure 4.17: The original Inception module (a) from [92], and the expanded Inception modules (b), (c), (d) from [104].

Rather than stacking layer after layer of convolutions, like AlexNet and VGGNets, the Inception module allows the model to go wider, applying different convolution sizes at the same layer depth allowing model complexity to increase and greater abstractions to be learned in image features without abundant model depth. To achieve this with little resource increase required a variety of tricks in the utilisation of the 1×1 convolution filter. This filter provides a dimension reduction input to the larger convolutional filters (3×3 and 5×5). In general, an Inception network, is a network consisting of the aforementioned modules stacked one after the other with occasional pooling layers. The Inception-v3 architecture follows this format and includes a 1000 neuron softmax output layer for classifying the ImageNet dataset (Table 4.6).

Table 4.6: An outline of the Inception-v3 architecture with its Inception building blocks referenced from Figure 4.17 [104].

type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	Module (b)	35×35×288
5×Inception	Module (c)	17×17×768
2×Inception	Module (d)	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

The Keras [113] implementation of this Inception-v3 architecture is utilised for this work. The only modifications to the network was adapting the input size of the model to 224×224 px for consistency with our other networks. As well as modifying the output fully connected layer to have nine neurons instead of 1,000 neurons for use with DeepWeeds instead of ImageNet.

ResNet-50

ResNet-50 is a variant of the *deep residual learning* framework first proposed in [93]. At the time of this proposal, CNN architecture development had simplified to adding more layer complexity to the network, reaping the accuracy rewards and ignoring the required computational power. However, an upper limit exists otherwise known as the *degradation* problem. If the network depth increases to infinity, the accuracy of the network will

eventually saturate (i.e. the complexity of the model deepens beyond the complexity of the data creating an infinite singularity of solutions) and then degrade very quickly. Residual networks offer a band-aid for this problem with the introduction of the residual learning block, as shown in Figure 4.18.

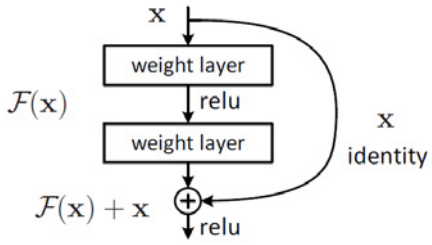


Figure 4.18: A residual learning connection, the foundational building block of ResNets [93].

Table 4.7: Overview of the defined layers within the different ResNet architectures. *ResNet-50 denotes the 50-layer architecture defined here [93].

layer name	output size	18-layer	34-layer	50-layer*	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

This residual building block essentially provides a shortcut connection between layers that allows the outputs from previous layers to skip its next layer and jump directly to an upstream layer. Skipping layers in effect, simplifies the network without reducing or increasing its parameter complexity. The skipped layers are gradually restored during training as the model learns the full feature space. This results in the ability to train much deeper networks with no added computational cost or saturated complexity. This building block can be inserted into any non-residual (or plain) network as long as the input and output dimensions of layers are satisfied during the skips. ResNet-50 is a 50

layer residual network that takes a 224×224 px input shape, passes it through a 7×7 convolutional layer and then passes it through a series of stacked convolutions with 1×1 and 3×3 kernel sizes, while residual connections are instituted for each layer. The 50-layer version is a medium sized variant amongst its various sized ResNet counterparts (Table 4.7).

For this work, we utilise the Keras [113] application library implementation of ResNet-50 as a baseline. The only revisions made to the architecture for use with DeepWeeds include replacing the final 1,000 neuron output layer with a nine neuron fully connected layer to match the DeepWeeds classes. Residual networks are able to outperform plain networks of similar complexity and can be trained much faster.

MobileNetV2

MobileNets are a class of efficient deep CNN architectures developed by Google [108, 109] for mobile and embedded vision applications. While the vast majority of works sought to develop deeper and more complicated networks in order to achieve higher accuracy, none advanced the efficiency of networks with respect to size and speed during the application of the models, otherwise known as inference time. MobileNets achieved accuracies on ImageNet that match the most complex and deep networks with but a fraction of the floating point operations and parameter counts. This was achieved by replacing standard convolution operations with depthwise separable convolution. The output feature map for convolution is represented by,

$$\mathbf{G}_{k,l,n} = \sum_{i,j,m} \mathbf{K}_{i,j,m,n} \times \mathbf{F}_{k+i-1,l+j-1,m}, \tag{4.6}$$

which has a computational cost of,

$$D_K \times D_K \times M \times N \times D_F \times D_F, \tag{4.7}$$

where M is the number of input channels, N is the number of output channels, $D_K \times D_K$ is the kernel size and $D_F \times D_F$ is the feature map size. Replacing the kernel of standard convolution with a depthwise convolutional kernel results in little loss of accuracy with extreme efficiency. However, it only filters the input channel. An additional layer of 1×1 convolution is required to map the output of the depth wise convolution into new features. This is known as depthwise separable convolutions and when used to replace a standard convolutional layer it results in a reduction of computational cost of,

$$\frac{1}{N} + \frac{1}{D_K^2}. \tag{4.8}$$

Standard MobileNets use $D_k = 3$ depthwise separable convolutions which equates to 8-9 times less computations than that of standard computations with a minor reduction in accuracy.

Table 4.8: An overview of the MobileNetV2 architecture where n denotes how many repetitions of the stated layer occur, c represents the number of output channels in each repeated layer, s represents the stride of the first layer in each sequence. All convolutions have a kernel size of 3×3 [109].

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

MobileNetV2 [109] marries this strategy with that of a residual network to improve the state-of-the-art performance of mobile models on classification tasks. The input for this network is of shape 224×224, as shown in Table 4.8. The unique convolutional operation of MobileNets allows this model to have staggering depth, 54 layers (more than all other networks investigated above), and yet only has 3.4 million parameters (7 times less than the closest network being investigated here).

For this work, we use the Keras [113] application library implementation of MobileNetV2. The only modification made to the architecture is swapping out the final 1,000 neuron output layer for ImageNet with our own fully connected nine neuron output layer. Investigating a light weight network, like MobileNetV2, affords this study a look at the real-time benefits for robotic weed control. If MobileNetV2 can achieve usable accuracy on the DeepWeeds dataset it is a prime candidate for field implementation.

4.3.4 Training Regime

This subsection outlines the training regime for deploying the five architectures on DeepWeeds. It was decided to evenly evaluate the models by having them adhere to the same benchmark training regime.

Transfer Learning

Transfer learning is a powerful and useful technique for achieving the highest possible accuracy on datasets that are relatively small compared to the benchmark deep learning datasets (such as ImageNet). The DeepWeeds dataset totals 17,509 images. While this is relatively large compared to other weed species datasets, it pales in comparison to the millions of images in the ImageNet dataset. Therefore, to maximise accuracy and shorten training time it was decided to use transfer learning with pretrained ImageNet models. This required utilising pretrained weights of the aforementioned models and restarting the training process on the DeepWeeds dataset. To ensure that the model fit well to the weeds classes, the entire model was fine tuned. The weights of the 9-neuron fully-connected layer (i.e. the output layer) were initialised by the uniform random distribution as per Glorot et al. in [129].

Image Pre-Processing

To overcome the highly variable nature of the target weed classification application, a series of augmentations were performed on both the training and validation image subsets to account for variations in rotation, scale, colour, illumination and perspective. Image augmentation, similar to that performed on the lantana dataset in section 4.2, was performed using Keras' pre-processing library [113] and OpenCV [72]. This further amplifies the variability of the DeepWeeds dataset, as shown in Figure 4.19.

All images were first resized to 256×256 pixels in size and randomly augmented for each epoch of training. Each image was also randomly rotated in the range of $[-360, +360]$ degrees. Then, each image was randomly scaled both vertically and horizontally in the range of $[0.5, 1]$. Each colour channel was randomly shifted within the range of ± 25 (*i.e.* approximately $\pm 10\%$ of the maximum available 8-bit colour encoding range $[0, 255]$). To account for illumination variance, pixel intensity was randomly shifted within the $[-25, +25]$ range, shifting all colour channels uniformly. In addition, pixel intensity was randomly scaled within the $[0.75, 1.25]$ range. Random perspective transformations were applied to each image to simulate a large variation of viewing distances and angles. Finally, the images were flipped horizontally with a 50% probability and then cropped to retain the model specific input size for each architecture's input layer. Without this extensive augmentation, the CNN networks drastically over-fitted the available images by memorising the training subsets.

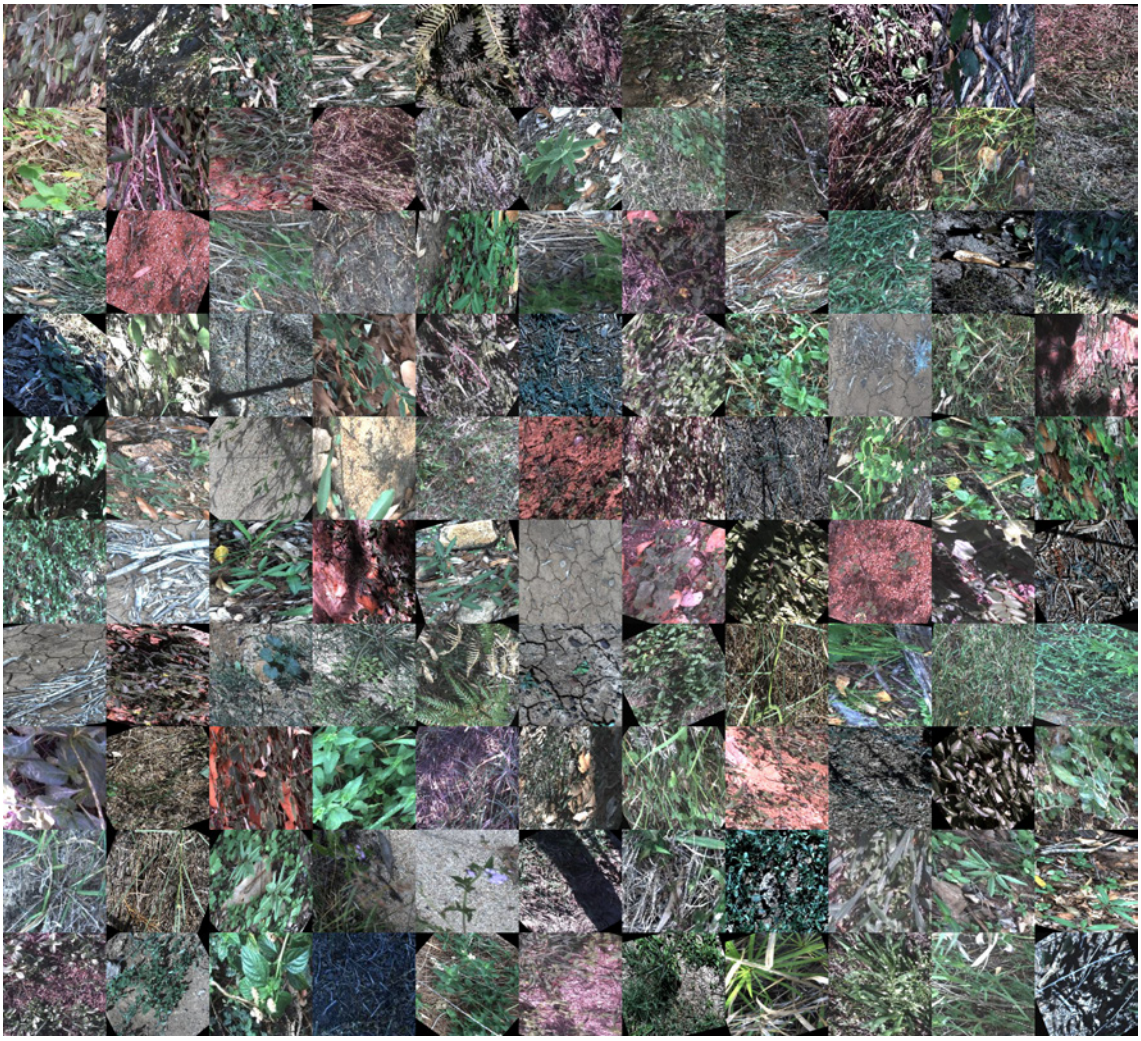


Figure 4.19: A sampling of training and validation subsets of the DeepWeeds dataset that have undergone random rotation, scale, colour, illumination and perspective pre-processing augmentations.

Activation and Loss Function

The nature of this classification task for the *DeepWeeds* dataset allows for multiple weed species to be present in each image. Therefore, it may be useful to generate an output probability for each class in the dataset on a single image. To accomplish this, we utilise the sigmoid activation function, which defines the probability distribution of each class p_i for the final output layer as,

$$p_i = \frac{1}{1 + e^{-y_i'}}, \tag{4.9}$$

where y_i' is the predicted class score. This allows an output of probabilities for each class to identify the likelihood that an image belonged to each class. Furthermore, an image is classified as one of the target weeds if that class' sigmoid-activated neuron probability was the greatest amongst all others and its likelihood was greater than $1/9 = 11.\overline{1}\%$ (i.e. a random guess). The random guess threshold was implemented to overcome the immense variation in the negative *DeepWeeds* class, which causes its target probability to be less strongly weighted towards specific image features than the eight positive classes – whose images are more consistent.

In concert with the sigmoid activation function we deploy the binary cross entropy loss function to enable the network to learn different classes during backpropagation.

Stochastic Gradient Descent

The Keras implementation of Adam [115], a first-order gradient-based method for stochastic optimisation, was used for optimising the binary cross entropy loss function.

Learning Schedule and Early Stopping

A small initial learning-rate ($\eta_0 = 1 \times 10^{-4}$) was used to begin the transfer learning process, since the pretrained model starting point has already undergone optimisation. The schedule of the learning rate is established such that the learning rate η successively halved every time the *validation loss* does not decrease after 16 epochs. The training was performed in batches of 32 images, and aborted if the validation loss did not decrease after 32 epochs. While training, the model with the smallest running validation loss was continuously saved, in order to re-start the training after an abortion. In such cases, training was repeated with the initial learning rate $lr = 0.5 \times 10^{-4}$. The batch size of 32 images was chosen by experimenting with the maximum possible size that could be fit onto an NVIDIA GeForce GTX 1080 GPU card for each of the networks.

Training Process

The five revised architectures were each trained on the DeepWeeds dataset with the documented training regime for a maximum of 200 epochs with k fold cross validation ($k = 5$). Training for 200 epochs on an NVIDIA GeForce GTX 1080, the AlexNet models took an average of 9.10 hours to train, Inception-v3 models took an average of 10.80 hours to train, MobileNetV2 took 9.38 hours, ResNet-50 took 9.38 hours and VGG16 took 9.33 hours.

None of the models appear to be overfit the training dataset as the validation accuracy consistently sits below the training accuracy, as is to be expected (Figure 4.20). Furthermore, the validation loss for all models plateaus without ever rising, which also indicates

the models are not overfitting to the training subset (Figure 4.21). Hierarchical tiers of model classification performance on the training and validation subsets are also evident (Figure 4.20). ResNet-50 and Inception-v3 outperform the other models with a steeper learning curve in the early epochs. Both then begin to plateau after 100 epochs of training, eventually reaching optimal validation accuracies of 97.6% and 96.7%, respectively. MobileNetV2 and VGG16 closely follow these two models with strong classification of the training and validation accuracy. Both models have a slightly delayed learning curve with large jumps between 60 and 80 epochs. MobileNetV2 and VGG16 then begin to plateau in accuracy after 100 epochs resting at optimal validation accuracies of 95.8% and 95.1%, respectively. The bottom tier performing model is then AlexNet, which is also the most primitive. It has an elongated learning curve, plateauing after 140 epochs and reaching a peak optimal validation accuracy of 91.7%.

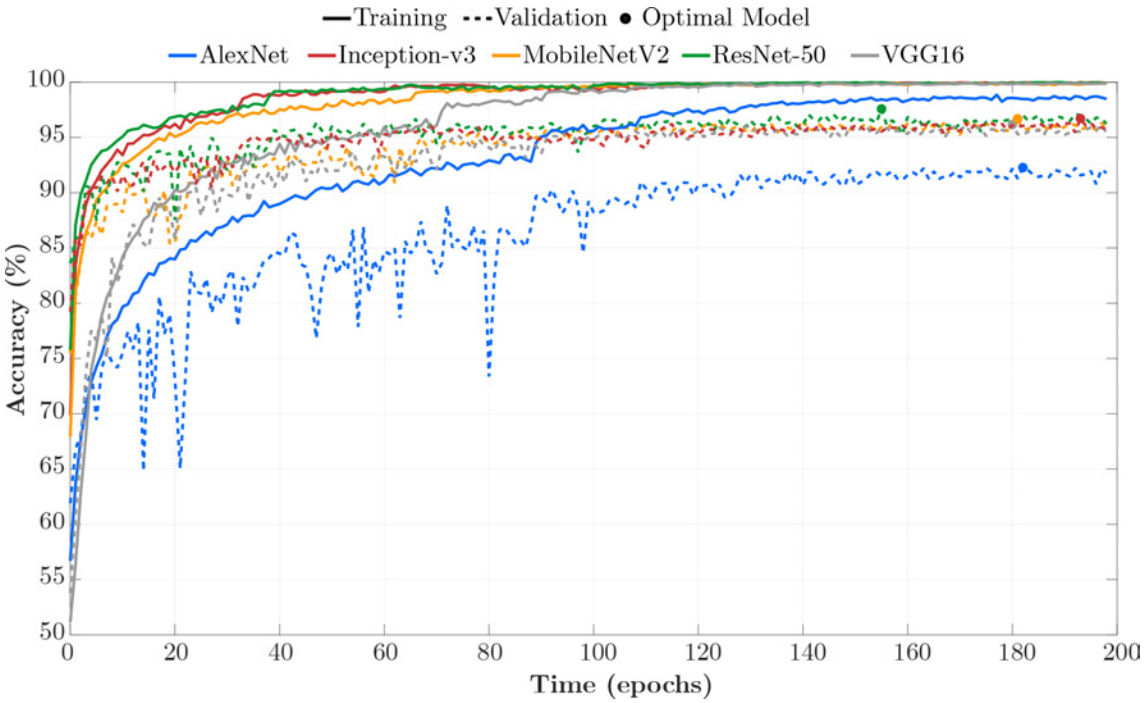


Figure 4.20: Training and validation accuracy for each epoch of the training process for a single fold of each of the five networks under investigation.

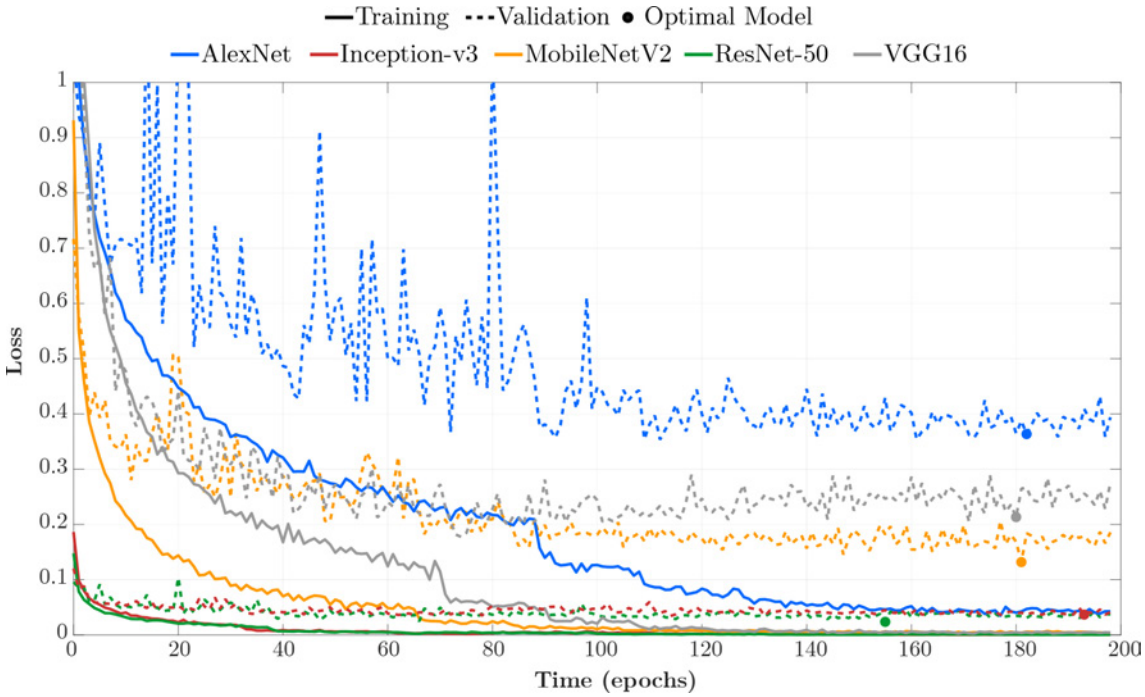


Figure 4.21: Training and validation loss for each epoch of the training process for a single fold of each of the five networks under investigation.

4.3.5 Classification Performance

After training and optimising the models against the training and validation subset, the optimal model epoch (defined by the lowest validation cross entropy loss) was extracted for evaluation against the testing subset. This process was performed for every cross validated fold of every model. Evaluating performance against the heldout test subset removes bias and gives an indication of how the models will generalise to an independent dataset.

The individual class accuracy and weighted overall average accuracy of each model, which takes into account the number of samples within each class, provides the best measure of model classification performance (Table 4.9). A visual comparison of the weighted average accuracy for each model clearly shows the hierarchy of model performance (Figure 4.22). The top tier models are again, ResNet-50 and Inception-v3 with 95.7% and 95.1% respective accuracy on the DeepWeeds dataset. VGG16 and MobileNetV2 follow along in the second tier with 92.3% and 91.6% average classification accuracy. Finally, AlexNet achieves 86.2% accuracy. All models show little variance in weighted accuracy across the randomised folds indicating the models generalise well with randomly varied datasets. The performance of these five models mirrors their performance on the ImageNet dataset, indicating that the learned features of the weeds species dataset has much

in common with the general world ImageNet dataset. It can be seen that model complexity plays an integral role in classification performance. The top-tier models, ResNet-50 and Inception-v3, have the highest complexity with their combined number of parameters and layers. MobileNetV2's unique architecture and sparse convolution operation allows it to have a complex layer structure with far less parameters. Its performance matches the more complex VGG16 model. Meanwhile, AlexNet's overtly simpler layer structure results in the lowest comparative performance despite its high parameter count.

Table 4.9: Average test classification accuracies (%) across all five cross validated folds for the five architectures. The network with the highest accuracy is emboldened for each species. The number of network parameters and layer depth for each model are also listed.

Weed species	**AlexNet**	**Inception-v3**	**MobileNetV2**	**ResNet-50**	**VGG16**
Chinee apple	63.6 ± 7.6	85.3 ± 5.3	83.6 ± 2.8	$\mathbf{88.5 \pm 3.7}$	86.0 ± 2.5
Lantana	80.8 ± 4.4	94.4 ± 1.2	$\mathbf{95.5 \pm 1.5}$	95.0 ± 1.8	91.4 ± 3.1
Parkinsonia	74.3 ± 13.7	96.8 ± 1.8	94.0 ± 2.0	$\mathbf{97.2 \pm 2.3}$	95.2 ± 2.8
Parthenium	81.8 ± 7.0	94.9 ± 1.0	86.9 ± 4.6	$\mathbf{95.8 \pm 3.0}$	89.5 ± 3.8
Prickly acacia	89.0 ± 2.2	92.8 ± 2.4	88.3 ± 8.5	95.5 ± 1.6	$\mathbf{96.2 \pm 1.8}$
Rubber vine	78.3 ± 7.9	$\mathbf{93.1 \pm 2.6}$	90.0 ± 2.0	92.5 ± 3.1	92.4 ± 3.0
Siam weed	88.8 ± 3.9	$\mathbf{97.6 \pm 1.1}$	91.8 ± 2.5	96.5 ± 0.5	94.5 ± 1.5
Snake weed	75.0 ± 4.3	88.0 ± 3.1	85.1 ± 4.6	$\mathbf{88.8 \pm 2.5}$	87.3 ± 3.2
Negatives	92.9 ± 1.4	97.2 ± 0.6	93.6 ± 0.7	$\mathbf{97.6 \pm 0.4}$	93.1 ± 2.9
Weighted average	86.2 ± 2.3	95.1 ± 0.1	91.6 ± 1.4	$\mathbf{95.7 \pm 0.5}$	92.3 ± 1.7
Parameters	56,910,601	21,786,793	2,235,401	23,553,033	33,635,605
Layers	8	42	54	50	16

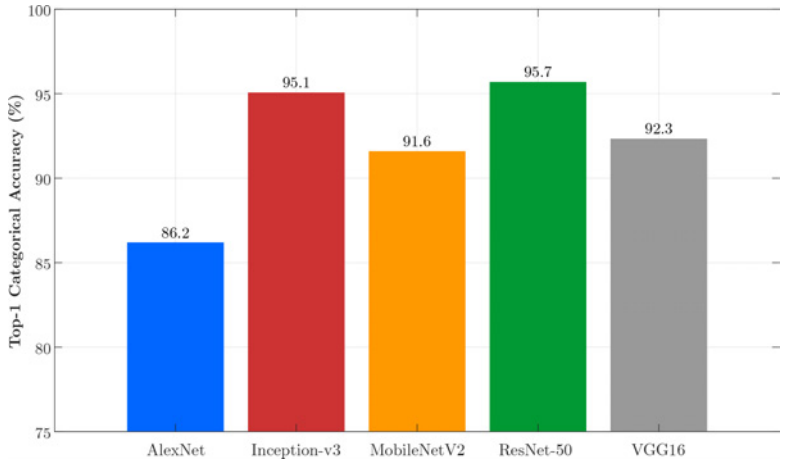


Figure 4.22: Comparison of the average test classification accuracy achieved on the DeepWeeds dataset for the five investigated CNN architectures.

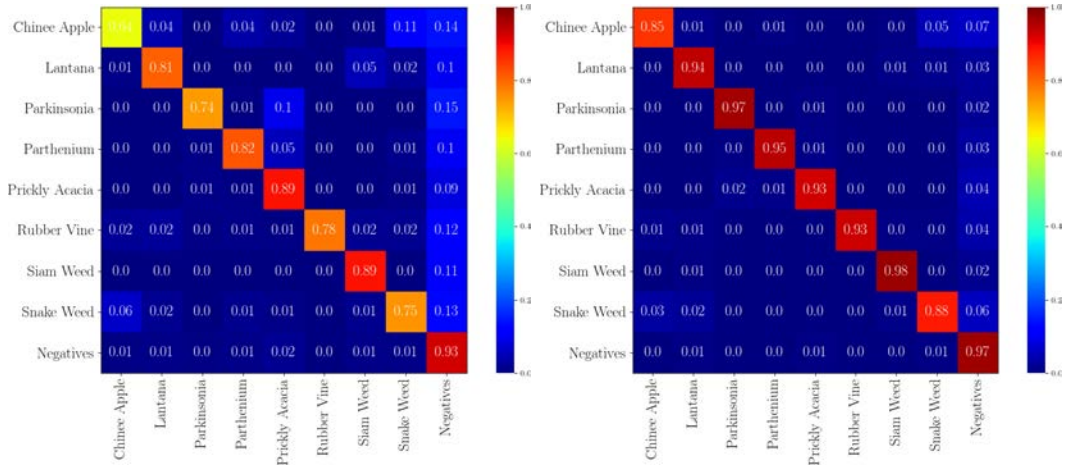
The inter-species performance of each model indicates which species are generally easier for the deep CNNs to learn and which are not (Table 4.9). From model to model, the negative class appears consistently to be the most well predicted class, never dipping below 90%. One can reason this is due to the relative size of the negative class to the others, providing more training examples to learn. The weed species that are the most consistently well recalled across models are Siam weed and Prickly acacia; while the most difficult weed species are consistently Chinee apple and snake weed. We suspect the relatively low performance on Chinee apple and snake weed is due to their strongly correlated features leading to high interspecies confusions.

Confusions

Confusion matrices resulting from aggregating each model's performance across the five cross validated test subsets are provided in Figure 4.24. The confusion matrices help us to see where common species confusions are occurring and which species may be correlated. The models are performing well with a strong diagonal of class recall and relatively low confusion counts overall (Figure 4.24).

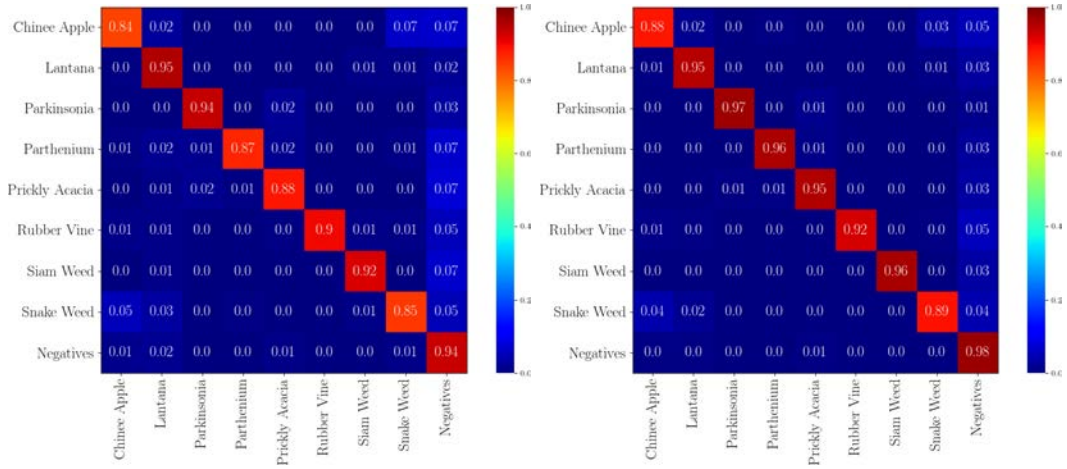
The models commonly confuse between 3-11% of Chinee apple images with snake weed and between 3-6% vice versa. Reviewing these particular samples shows that under certain lighting conditions the leaf material of Chinee apple looks strikingly similar to that of snake weed. This is illustrated in the sample misclassification of snake weed in Figure 4.23. Furthermore, the models incorrectly classify between 1-10% of parkinsonia images as prickly acacia. One of these misclassifications is presented in Figure 4.23. It should be noted that parkinsonia and prickly acacia are from the same genus, and are commonly both known as *prickle bush* species. In addition to their similar shape and size, they both produce thorns, yellow flowers and bean-like seed pods. This likeness is considered to be the most likely reason for these false positives in our model.

Figure 4.23: Example images highlighting confusions between classes of weed species. Specifically, (a) correctly classified snake weed, (b) Chinee apple falsely classified as snake weed, (c) correctly classified prickly acacia, and (d) parkinsonia falsely classified as prickly acacia.



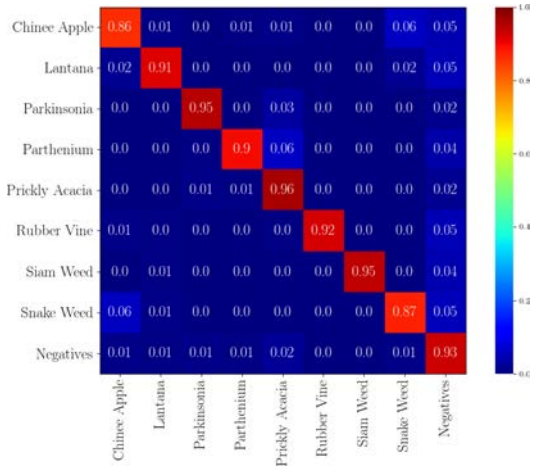
(a) AlexNet

(b) Inception-v3



(c) MobileNetV2

(d) ResNet-50



(e) VGG16

Figure 4.24: The confusion matrices for the five models aggregated across their five cross validated folds indicating the inter-class accuracy when evaluating the models on unseen heldout test subsets.

Minor confusions are limitations of the learning model, however these weeds – Chinee apple, snake weed, parkinsonia and prickly acacia – are all harmful and need to be controlled. Therefore, confusing one for the other can be considered inconsequential when the goal is to control them all. Of more pressing concern for the application of robotic weed control are false positives, where established native plant life or other desired non-targets are incorrectly classified as weeds. This is a deeply important issue as spraying off-target plant life may cause harm to the native ecosystem, lower crop yields, reduce pasture coverage, as well as result in unnecessary herbicide use.

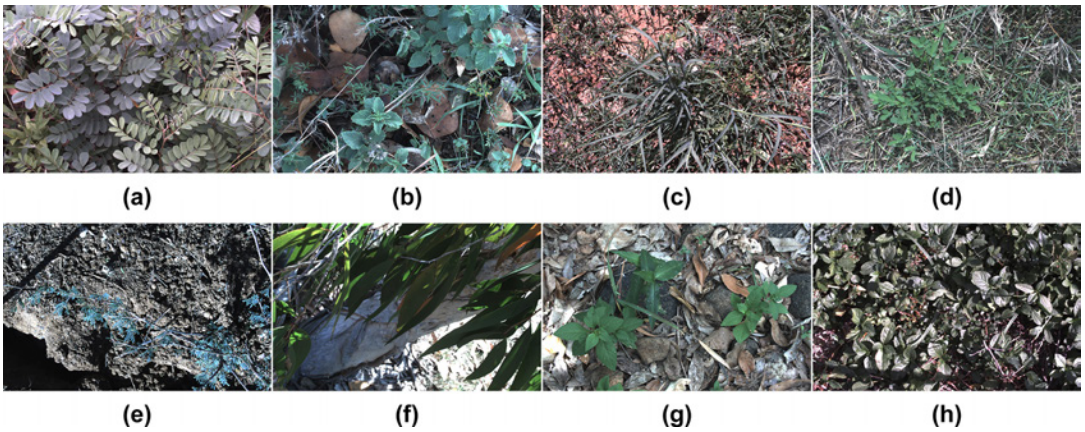


Figure 4.25: Example false positives where images from the negative class were falsely classified as (a) Chinee apple, (b) lantana, (c) parkinsonia, (d) parthenium, (e) prickly acacia, (f) rubber vine, (g) Siam weed and (h) snake weed.

Examples of negative plant life that have been misclassified as each of the eight target weed species are presented in Figure 4.25. In every case, the native plant exhibits image features that closely resemble the species it has been confused with. This shows that the immense variation in plant life that makes up the negative class does introduce unavoidable false positives. Fortunately, these cases are few and far between for the best performing models. ResNet-50 and Inception-v3 have a false negative occurrence of just 2% and 3% (Figure 4.24). While the three other models, MobileNetV2, VGG16 and AlexNet have false negative rates of 6%, 7% and 7%, respectively.

Model Complexity

This section has found that ResNet-50 and Inception-v3 offer the strongest classification performance. This is no surprise when comparing the complexity of these models – defined by the number of trainable network parameters – to lower performing models. Typically, the more parameters that are available for optimisation (i.e. the more complex the model), the larger the solution space of the network and therefore the more ability it has to fit

to a given dataset and exhibit strong performance. However, this presents a challenging tradeoff when considering real-time model inference. The more complex model will take longer to classify a given image in the field. One way of quantifying this tradeoff is with *accuracy density* (the percentage accuracy of a model divided by its parameter complexity).

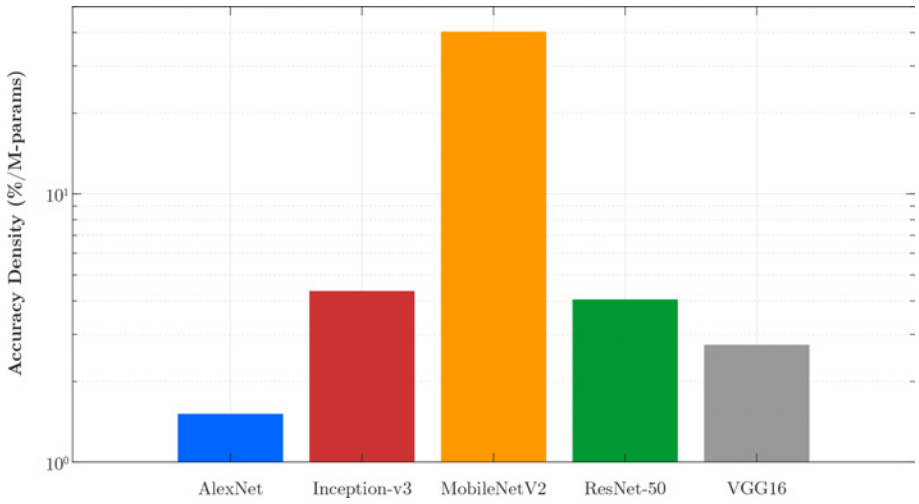


Figure 4.26: Comparison of the accuracy density of the five investigated models which gives an indication for the ideal model for in-field performance.

MobileNetV2 is the clear standout in terms of accuracy density due to its incredibly low parameter count and good classification accuracy (Figure 4.26). This indicates that MobileNetV2 may be the model of choice for real-time implementation despite its second tier classification accuracy. The two best classifiers, Inception-v3 and ResNet-50, tie for second place with accuracy density due to their highly complex model structures. VGG16 closely follows with similar complexity and diminished accuracy. Finally, AlexNet has the lowest accuracy density due to a low relative accuracy. However, its complexity is perhaps unfairly represented by its parameter count. Since the AlexNet structure only has eight layers it would likely be very fast. Nonetheless, its relative accuracy excludes it from further consideration.

4.3.6 Summary

In summary, this section has revealed that modern CNNs can be used effectively to learn the differences between a variety of weed species simultaneously. They are also well suited to the highly complex background imagery exhibited in the DeepWeeds dataset and are thus ideal for in situ classification. ResNet-50 provided the strongest classification accuracy on the DeepWeeds dataset achieving 95.7%. A short analysis of accuracy density revealed that MobileNetV2 may be the most ideal model based on the simplicity of its model

structure. It achieved 91.6% classification accuracy with approximately ten times less parameters than ResNet-50. But rather than relying on accuracy density to estimate real-time performance, it would be more beneficial to measure inference time and power consumption of the models during their real time implementation.

4.4 Real-Time Inference

As we progress toward the application of robotic weed control, we must also assess our chosen model's viability for real-time and in-field performance. In the previous section, we benchmarked the classification performance of five popular deep CNN architectures on the DeepWeeds dataset. We demonstrated that deep learning can provide high accuracy in differentiating weed species and detecting them from their natural complex environment. But can these highly complex algorithms be implemented fast enough so that weed control robots function at speeds that make them practically useful for farmers and landholders?

4.4.1 Related Work

Early Work

The harsh real-time requirement of weed and plant species detection with machine vision has long been recognised as a major issue. A study in 2000 by Yang [130] distinguished young corn plants from weeds using an Artificial Neural Network (ANN) with up to 80% accuracy, performing training in 20 hours and inference in close to one second per image. This was a tremendous result for its time, however falls well short of useful implementation speed required for a real-time robot.

The advent of powerful GPUs led to rapid increases in accuracy, as we have seen. However performing real-time inference on high-performance GPUs such as the NVIDIA GTX 1080, is putative to consume large amounts of power and is hence, ill-suited to deployment on edge computing devices. Nonetheless, a variety of studies have deployed deep CNNs on power hungry desktop GPUs specifically for robotic weed control to achieve the required speed and accuracy albeit at the cost of high power consumption [131–133]. All three studies utilise the BoniRob [26] agricultural platform, a low-speed highly functional agricultural robot for cropping scenarios. The first such work [131] uses an on-board NVIDIA GeForce GTX 1080Ti GPU to achieve highly accurate classification with a novel sequential crop-weed fully convolutional network that runs at 5 fps or 200 ms inference time. The second study [132] employs a GTX 1080Ti GPU to achieve 89% precision detection of weeds in carrot fields at 18.56 fps or 53.8 ms inference time. From there, they can sacrifice detection accuracy for faster processing speeds up to 56 fps or 17.9 ms inference time per

image. The latter study [133] used online semi-supervised crop and weed detection on an NVIDIA GeForce GTX 1080 to classify sugar beet with up to 95% accuracy at 8 fps (125 ms inference time per image) or operate with degraded performance at 5 fps (200 ms inference).

These studies used the high-end GPU hardware and achieved high levels of accuracy. However, some inference times still don't match up to the requirements of real-time throughput required in agricultural production. Meanwhile, the high power consumption of the weed control robot limits its battery cycle and demands a heavy power supply requirement. Fortunately, innovations in edge computing have opened new pathways for implementation.

Edge Computing Innovation

One solution proffered by developers such as NVIDIA, are embedded GPUs for mobile application or edge computing devices. The Jetson family of compute modules provide a power-efficient platform for deploying deep learning architectures, some of which are already being used in agriculture [134–138]. Partel et al. employed the YOLO [121] object detection CNN to achieve an overall precision and recall of 59% and 44% of weed targets at 2.4 fps, which was not practical [134]. They improved processing speed by deploying the lighter weight Tiny YOLO to achieve 22 fps (or 45.5 ms inference time) with some loss of accuracy. These studies collectively show that real-time detection is achievable. But they fail to quantify whether the speeds will be practical for the end-users, and thus whether these real time solutions are ready for industry uptake.

A prominent and low-cost edge computing device is the Raspberry Pi, which has prevailed in less strenuous real-time applications. However, one recent study [139] deployed a light-weight custom CNN integrating elements of U-Net [140], MobileNet [108], DenseNet [141] and ResNet [93] on a Raspberry Pi. This model was able to semantically segment weed areas with up to 67% accuracy at over 10 fps (or 100 ms inference time per image) on the Raspberry Pi 3B+ edge device.

NVIDIA Corporation has also developed a real-time software library to support edge computing, with the TensorRT library [126] for optimising deep learning architectures. The TensorRT library takes as input a trained deep learning model and employs a variety of strategies to accelerate its real-time performance then outputs an optimised inference engine for deploying said model in the field. Such strategies include: reducing the precision of the input network's weight parameters requiring less multiplications, removing redundant layers and operations from the network that are required for training but not inference, fusing convolution, bias and ReLU operations together to shorten their imple-

mentation time and aggregating operations with sufficiently similar parameters, just to name a few! This software library is made compatible with the most common deep learning development libraries. However, certain layer operations are not supported. Therefore, careful consideration of model selection should be made with consideration of supported TensorRT operations.

No studies have yet shown the power of integrating the NVIDIA TensorRT library for weed species detection. Similarly, there exists no such literature benchmarking the different NVIDIA Jetson family devices for real-time weed species detection. This section aims to deliver a benchmark of weed species detection using the TensorRT library across all NVIDIA Jetson devices. With direct relation to a real-time inference target based on the requirements of a weed control robot, this work will also provide the first look at what speeds are possible for a weed control robot to achieve in the field.

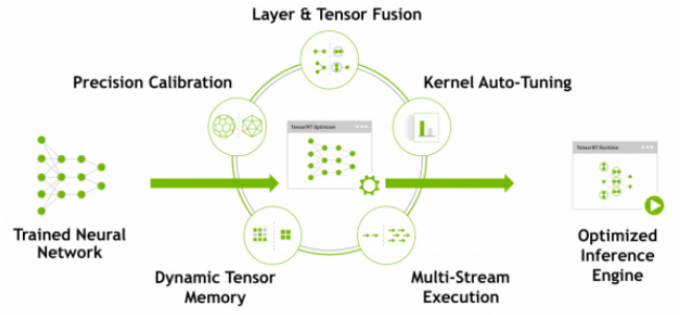


Figure 4.27: A diagram illustrating the TensorRT optimisation process, where a trained neural network undergoes a series of optimisation strategies to produce a deployment-ready runtime inference engine [142].

4.4.2 FPGAs

Another powerful solution to real time implementation of deep CNNs is using Field Programmable Gate Arrays (FPGAs) rather than GPUs. Although embedded GPUs offer substantial power improvements over conventional GPUs, they are still relatively power hungry when compared to FPGAs. While GPU-based implementations targeted towards image classification tasks are plentiful, there are few recent works [143, 144] that have detailed the implementation of custom hardware accelerators specifically tasked for agricultural purposes. Knoll et al. demonstrated real-time performance by implementing an FPGA-based DNN on a Terasic DE1-SoC for plant detection in organic farming [143]. This system can classify a target dataset with accuracy matching that of a state-of-the-art GPU while running at up to 42 frames per second with only 4 W of power consumption, which is 45 times lower than an NVIDIA GTX 1080Ti [143]. Meanwhile, Lammie et

al. [144] demonstrated that FPGA-accelerated binarised networks significantly outperform their GPU-counterparts, achieving a 7-fold decrease in power consumption while performing inference on weed images three times faster. These significant benefits are gained while losing only 1% classification accuracy benchmarked against the DeepWeeds dataset presented in this thesis.

4.4.3 Goals

This section will provide a detailed performance analysis of the inference time and power consumption costs associated with the five deep CNNs investigated for weed species classification earlier in Chapter 4: AlexNet, Inception-v3, MobileNetV2, ResNet-50 and VGG16. The networks will be optimised using the NVIDIA TensorRT real-time inference library for maximum performance. Four different NVIDIA hardware platforms will be investigated: the GeForce GTX 1080 GPU and the Jetson Xavier, Jetson TX2 and Jetson Nano embedded GPU devices. Of critical importance is also the number of images being processed at a time by the detection system. This parameter will be investigated by examining the performance of the networks with different batch sizes during inference. There are two central goals driving this research, as below.

1. Compute the real-time speed and power performance for the five CNN architectures, four hardware devices, five sets of batch sizes using TensorRT optimisation library.

2. Evaluate these benchmarks against the performance targets for a weed control robot to aid the development of the prototype.

4.4.4 Performance Targets

For a weed control robot, there are two physical dimensions that restrict the real-time operation of the system and provide us a performance target. Firstly, any detection system that is operating in the field must be able to complete processing an image in time for the next image to arrive. This time limit, t_I, is defined in Equation 4.10 as the time it takes for the weed control robot to traverse the field of view of its camera d_I while operating at a vehicle speed of v (Figure 4.28).

$$t_I \text{ [ms]} = \frac{d_I \text{ [mm]}}{v \text{ [km/hr]}} \times \frac{3.6 \times 10^6 \text{ [ms/hr]}}{1 \times 10^6 \text{ [mm/km]}} \tag{4.10}$$

Figure 4.28: An illustration of the first real time speed limitation whereby the time it takes to process an image must not exceed the time it takes to traverse its field of view.

Second, any weed control robot's detection system must complete the processing of an image in time for any potential targets in that image to reach the weed control mechanism. For the prototype system under development in this thesis, the robotic system is a herbicide spraying mechanism. This time constraint, t_{II}, is defined in Equation 4.11 as the time it takes the robot to traverse the distance (d_{II}) from the location of where the image was collected to where the control method can be applied while travelling at a speed of v (Figure 4.29).

$$t_{II}\ [\text{ms}] = \frac{d_{II}\ [\text{m}]}{v\ [\text{km/hr}]} \times \frac{3.6 \times 10^6\ [\text{ms/hr}]}{1 \times 10^3\ [\text{m/km}]} \tag{4.11}$$

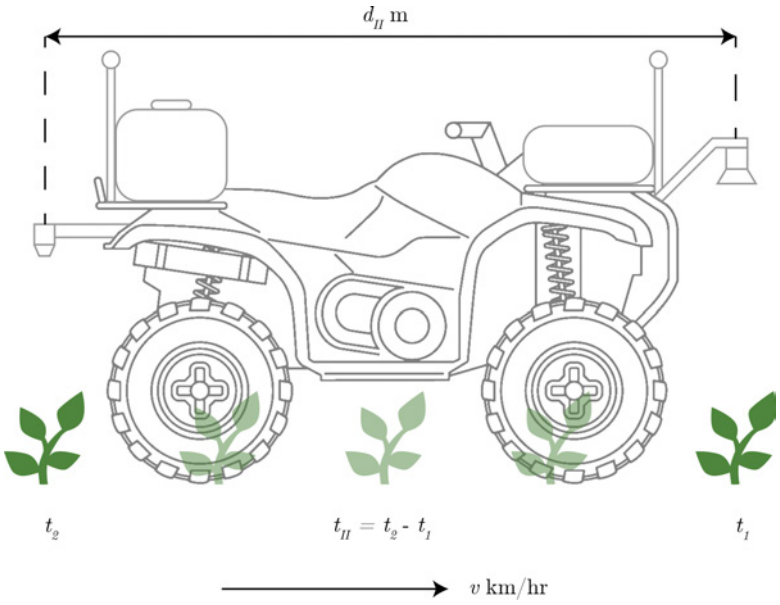


Figure 4.29: An illustration of the second real time speed limitation whereby the time it takes to process an image must not exceed the time it takes the vehicle to pass from the location of where the image was collected to the location of where the control method can be applied.

For the robot being developed in this thesis, the proposed field of view of the cameras in Chapter 5 are 450 × 280 mm. The cameras are oriented in landscape, therefore the distance d_I = 280 mm. The distance measured between the cameras and the herbicide sprayers is d_{II} = 2.2 m. It is therefore evident, that for this weed control robot, time constraint one outranks time constraint two because it will be much shorter.

Time Constraints

Let us now calculate time constraint one for this analysis. The vehicle speed constraint is determined by consultation with local landholders in a crop and rangeland situation, as well as by the recommended operating speed of existing boom sprayers which are subject to regulation [145]. Three tiers of speeds are decided upon:

Low speed 10 km/hr. Slower than ideal for boom spraying but perhaps a realistic tradeoff if robotic weed control achieves desirable accuracy.

Standard speed 15 km/hr. Recommended operating speed of a boom sprayer to prevent spray drift and maximise droplet activation.

High speed 20 km/hr. Maximum recommended operating speed that is a regular occurrence with the demands of broadacre cropping.

Table 4.10: Performance target time constraints based on the prototype robot dimensions and desired vehicle operating speeds.

Target	Vehicle speed (km/hr)	Inference time (ms)	Frame rate (fps)
Low	10	101	10
Standard	15	67.2	15
High	20	50.4	20

The real-time benchmarks for all hardware, models and batch sizes will be evaluated against these three tiers of performance requirements (Table 4.10). For the prototype vehicle to operate at 10 km/hr it must process images at 10 fps or 101 ms per image. To operate at 15 km/hr, the system must process images at 15 fps or 67.2 ms per image. Finally, to operate at 20 km/hr, the system must process images at 20 fps or 50.4 ms per image.

4.4.5 Hardware

As discussed, four NVIDIA devices have been selected for benchmarking real-time performance: GeForce GTX 1080 GPU card, Jetson Xavier, Jetson TX2 and Jetson Nano (Fig-

ure 4.30). These devices have been selected because of their compatibility with NVIDIA's CUDA and TensorRT libraries. The Jetson family of edge computing devices are of primary interest for use on the weed control prototype, while the desktop GPU card is included for perspective. They each offer unique performance specifications that will provide a useful comparison (Table 4.11).

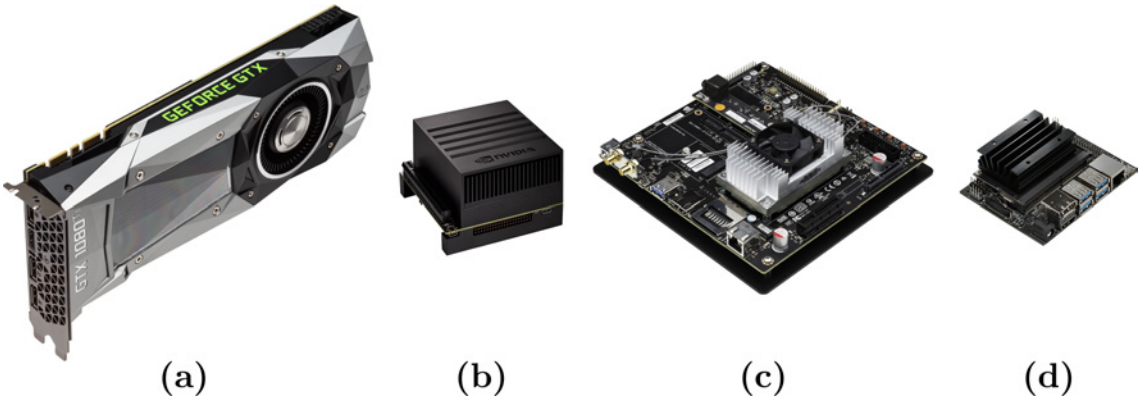


Figure 4.30: The NVIDIA suite of hardware devices being investigated, including: (a) GeForce GTX 1080 GPU desktop card, (b) Jetson Xavier, (c) Jetson TX2 and (d) Jetson Nano.

Table 4.11: A comparison of the selected NVIDIA devices' GPU specifications. Please note that the GTX 1080 is a high-performance card for desktop usage, while the other three are designed for mobile use.

Specification	**GeForce GTX 1080**	**Jetson Xavier**	**Jetson TX2**	**Jetson Nano**
CUDA cores	2,560	512	256	128
Memory size (GB)	8	16	8	4
Memory bandwidth	256-bit	256-bit	128-bit	64-bit
Max. temperature (°)	94	80	80	80
Max. power (W)	180	30	12	10
Dimensions (mm)	267 × 111	105 × 105	87 × 50	70 × 45
Price (USD)	$ 899	$ 699	$ 399	$ 99

NVIDIA have delivered distinct tiers of performance here and one will expect the inference speed to be directly proportional to the number of CUDA cores available on each GPU. Therefore, the hierarchy of performance will undoubtedly match the design tier, however, of more interest will be which hardware device will be able to meet the performance requirements for weed control. And of the ones that can, which are the cheapest and most efficient.

4.4.6 Testing Software

TensorRT

NVIDIA's TensorRT [126] library was implemented across all hardware devices. This required converting the trained Keras and TensorFlow DeepWeeds models into their TensorRT inference engine counterparts. This conversion process involves:

1. Saving the Keras model (and its weights) as a HDF5 model file.
2. Converting the HDF5 Keras model into a TensorFlow frozen protobuff (PB) model.
3. Converting the PB model into a Universal File Format (UFF) model using the TensorRT Python API.
4. Parsing the UFF file into a TensorRT inference engine using the TensorRT C library.

TensorRT inference engines are platform specific because the optimisations employed by TensorRT utilise the physical limitations of the GPU it is being targeted for. Therefore this conversion process was repeated for all models and all hardware devices.

Custom Software

Testing software was written in C++ and CUDA using the TensorRT library and the OpenCV image processing library [72]. The testing software deploys each of the five inference engine models with batch sizes of $1 - 5$ images per batch. The software is uniquely packaged and deployed on all four hardware devices.

Tensor Format

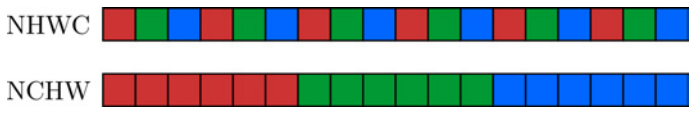


Figure 4.31: A simplified representation of the two most common tensor formats, NHWC, the TensorFlow default format and NCHW, the TensorRT default format.

In order to prepare images as a CUDA batches for inference, the tensor data had to be converted from the standard TensorFlow format, *NHWC*, to the tensor format accepted by TensorRT, *NCHW*. Where N is the number of images in a batch, C is the number of colour channels, H is the height of the image, W is the width of the image, and the order of letters indicates the order the tensor data is written and read (Figure 4.31).

Power Monitoring

Power consumption was monitored externally while the testing software is executed using the native GPU monitoring software present on each device. For the GeForce GTX 1080, the monitoring software was the NVIDIA System Management Interface (or `nvidia-smi`). For the Jetson family of devices, the monitoring software was Tegra Stats (or `tegrastats`). Both monitoring programs were executed with a time interval of 100 ms. The primary variable being monitored for each device is the GPU power consumption.

4.4.7 Benchmarks

The aforementioned testing software was executed for all CNN architectures, on all hardware devices for batch sizes of $1 - 5$. To ensure the power consumption and inference speed statistics are relative to the task of weed detection alone, no other tasks were running during testing and no display devices were connected. The final benchmarks are made available in Table 4.12. The inference times are colour coded to indicate how they measure up to required performance targets. The power consumption statistics are colour coded to arbitrary thresholds of efficiency.

The GeForce GTX 1080 far outperforms the mobile edge devices with inference speeds of all models well-within the 50 ms performance target. However, the downside of the GTX 1080 is its high power requirement with all power consumption measures far exceeding the edge devices. At a glance, it appears AlexNet and MobileNetV2 are the most efficient and best performing algorithms in terms of power and speed. The other three networks all appear to have more mixed performance.

Batch size has a significant impact on the speed of inference and the power consumption during operation. The lightweight networks, AlexNet and MobileNetV2, are the only networks able to achieve the most stringent 50 ms performance target on the Jetson Nano. AlexNet does so with up to four images per batch and MobileNetV2 does so with up to two images per batch. This is a very strong result for MobileNetV2, given its impressive accuracy in the previous section.

As batch sizes increase on the Jetson modules, we start to see mixed results in terms of which networks are able to meet the low, standard and high performance targets. Table 4.12 thus serves as an ideal glossary for model selection during the development of the robotic prototype in Chapter 5.

Table 4.12: Results of the inference time and power consumption benchmark tests for all CNN models, hardware devices and batch sizes. The resulting power consumption and inference time statistics are colour coded according to specific thresholds.

Model	Batch size (images)	Power consumption (W)				Inference speed (ms)			
		GTX 1080	Jetson Xavier	Jetson TX2	Jetson Nano	GTX 1080	Jetson Xavier	Jetson TX2	Jetson Nano
AlexNet	1	76.85	2.08	0.93	2.06	1.68	10.86	17.25	26.25
	2	80.89	2.36	1.47	2.53	1.95	12.97	19.47	34.31
	3	85.19	2.84	2.33	2.91	2.16	14.26	21.01	43.22
	4	85.99	3.08	2.56	3.03	2.24	15.55	22.61	49.96
	5	90.45	3.49	3.28	3.18	2.42	18.77	27.83	62.90
Inception-v3	1	88.47	3.23	4.47	3.25	7.49	36.74	42.62	91.93
	2	108.67	4.37	5.59	4.17	8.36	47.05	60.42	146.12
	3	119.31	4.71	6.64	4.36	10.07	65.58	84.44	216.83
	4	118.25	4.89	7.13	4.36	11.38	83.11	108.85	283.95
	5	117.59	5.06	7.35	4.48	13.00	100.46	135.33	354.45
MobileNetV2	1	71.14	2.23	1.77	2.31	3.22	11.39	17.40	29.05
	2	78.11	2.80	2.96	2.63	4.46	15.22	23.93	49.52
	3	86.75	3.14	2.93	2.79	5.51	20.16	35.09	71.98
	4	92.81	3.40	3.19	3.03	6.46	24.63	43.06	94.07
	5	96.43	3.44	3.51	2.92	7.56	31.29	52.94	116.41
ResNet-50	1	92.99	3.33	4.32	3.75	3.49	23.72	25.84	59.79
	2	114.86	4.11	5.20	4.24	4.44	32.70	40.80	95.63
	3	129.22	4.69	6.18	4.35	5.48	43.74	58.13	145.42
	4	127.59	4.95	6.34	4.44	6.65	52.42	71.64	180.08
	5	127.70	5.23	6.91	4.69	7.44	58.76	82.49	212.86
VGG16	1	146.41	5.25	7.05	4.14	5.81	43.05	62.31	166.36
	2	147.79	5.80	7.81	4.73	9.98	75.46	135.70	331.32
	3	144.43	5.81	8.41	4.77	13.57	111.72	179.63	464.30
	4	149.38	6.24	8.85	4.82	16.89	138.06	223.48	596.82
	5	152.69	6.40	9.56	5.01	20.62	168.14	243.15	704.62

Time (ms)	< 50.4	< 67.2	< 101	> 101
Speed (km/hr)	> 20	> 15	> 10	< 10
Power (W)	< 5	< 10	< 50	> 50

4.4.8 Inference Speed

Next we compare the inference time for each network with a batch size of one image (Figure 4.32). This offers a direct comparison of how each model would perform for a single camera, single nozzle spot-spraying prototype. All models when run on the GTX 1080 achieve the most stringent 50 ms threshold. This indicates that if using a GTX 1080 in the field on the weed control robot, the operating speed of 20 km/hr is easily achieved. However, this would present significant power requirements for the mobile platform which are not considered realistic.

The ranking of model performance for inference speed is consistent across all devices. The fastest network is AlexNet. This is obviously due to its simple architecture with only eight layers. However, AlexNet's accuracy on our DeepWeeds dataset is insufficient. The

next fastest model is MobileNetV2 with an inference speed of 29.0 ms on the least powerful Jetson Nano device. This is a tremendous result owing to the light weight design of the model's depthwise separable convolutions. The next to fastest architecture is ResNet-50, which on the Jetson Nano takes 59.8 ms to perform inference on a single image. The two slowest networks are Inception-v3 and VGG16. Inception runs at 91.9 ms with one image falling just within the slowest target threshold of 100 ms. Meanwhile, VGG16 does not meet the 10 km/hr performance requirement on the Jetson Nano for a single batch. The complexity of the latter three models are considered to be the reason for the relatively slower inference speed.

An important result here, is that all networks satisfy the most stringent 20 km/hr performance target when operating on the Jetson Xavier compute module. This indicates that not only is robotic weed control possible, it will be able to achieve an operating speed that would make it competitive in the broadacre cropping industry.

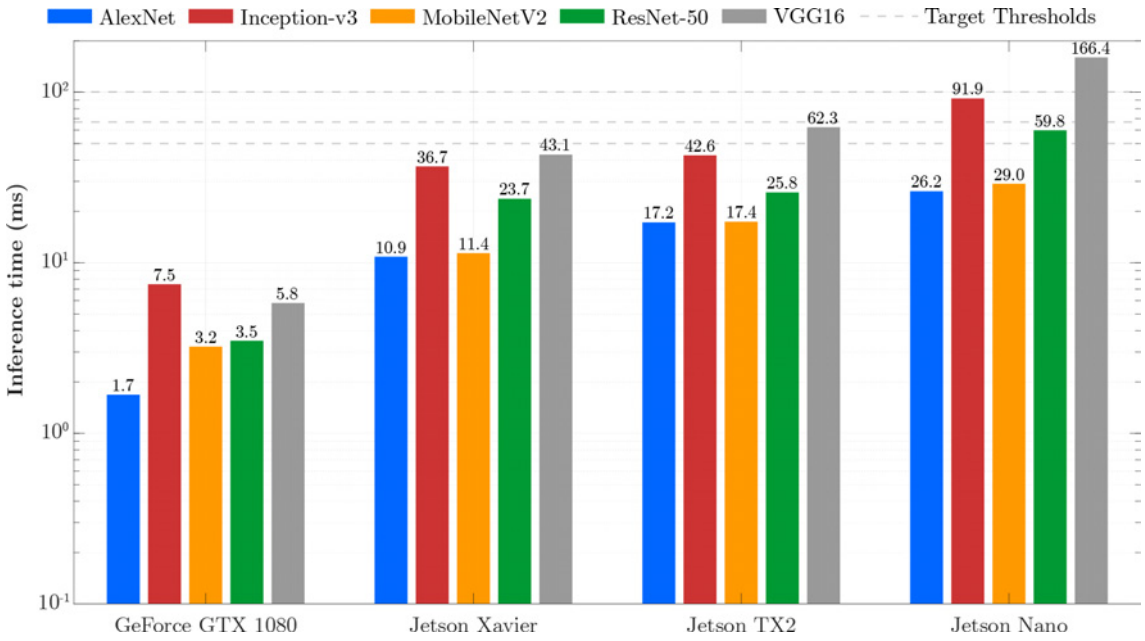


Figure 4.32: Comparison of the GPU inference time for the five networks when running on the four different hardware architectures with a batch size of 1.

4.4.9 Power Consumption

Next we compare the corresponding power consumption for each network on each architecture for a batch size of one single image, as illustrated in Figure 4.33. Again a log y-axis scale is used to visualise the incredibly hungry power statistics of the GTX 1080 GPU. Clearly, the GTX 1080 is not compatible for edge computing on a weed control

robot. This type of power consumption would render the robot's battery cycle unsuitable for end-users. In contrast, all the Jetson devices exhibit tremendously low and consistent power consumption across all the models during inference. The results from device to device are very evenly balanced. Power consumption therefore should have little effect on the decision to utilise one Jetson device over the other.

Model-to-model performance in terms of power consumption correlates to the inference speed. AlexNet is the least power hungry, followed by MobileNetV2, ResNet-50, Inception-v3 and finally, VGG16 as the most power hungry. There is notably a large gap of 3-6 W between AlexNet and VGG16 across the different architectures. This indicates just how much power the convolution operations consume, and that cutting down just one layer of convolutions can have a significant impact on speed and power.

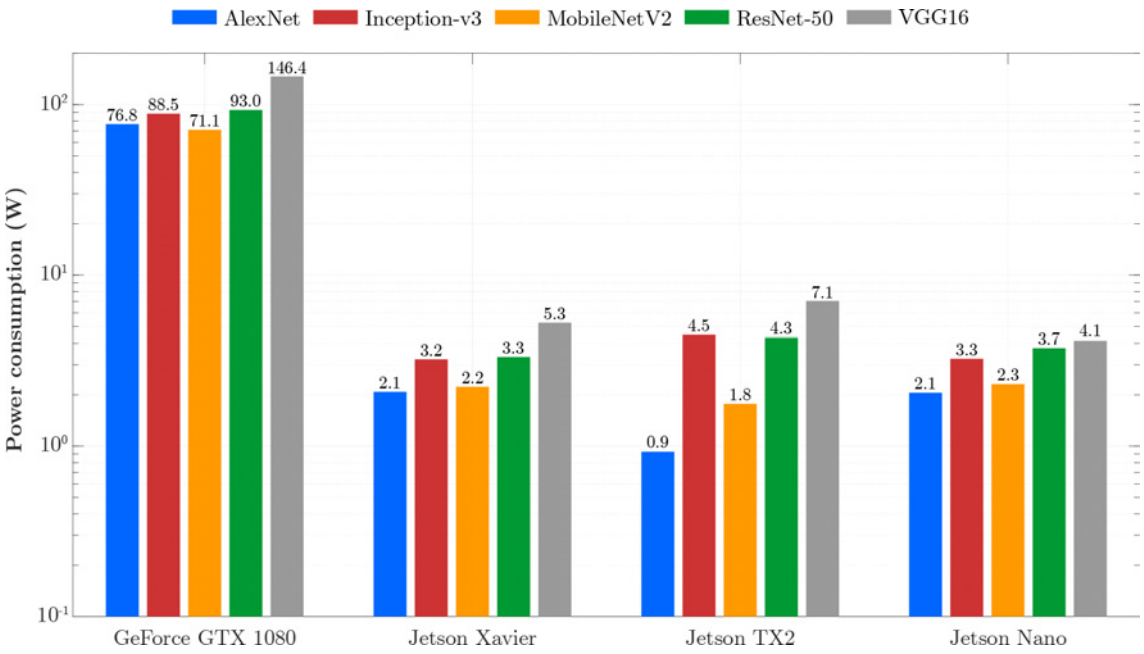


Figure 4.33: Comparison of the GPU power consumption during inference for the five networks when running on the four different hardware architectures with a batch size of 1.

4.4.10 Batch Size

Batch size is defined as the number of images being predicted at one time by the inference engine. Parallel computing is made possible because of the GPU architecture of each device. Batch size has important applications for robotic weed control and is realised by the number of cameras in operation on the robot. Multiple camera systems are ideal for large broadacre weed control solutions (that may have 12, 24 or 36 m boom sprayers) so

that a wider field of view can be captured. A cost saving decision then needs to be made regarding how many cameras are being driven and processed by a single processing unit.

Our benchmark indicated that MobileNetV2 could be executed with five cameras (batch size = 5) while maintaining the highest performance target and operating at 20 km/hr with a Jetson Xavier. Meanwhile, when running MobileNetV2 on the Jetson Nano, only two cameras can be operated simultaneously to achieve the target speed of 20 km/hr.

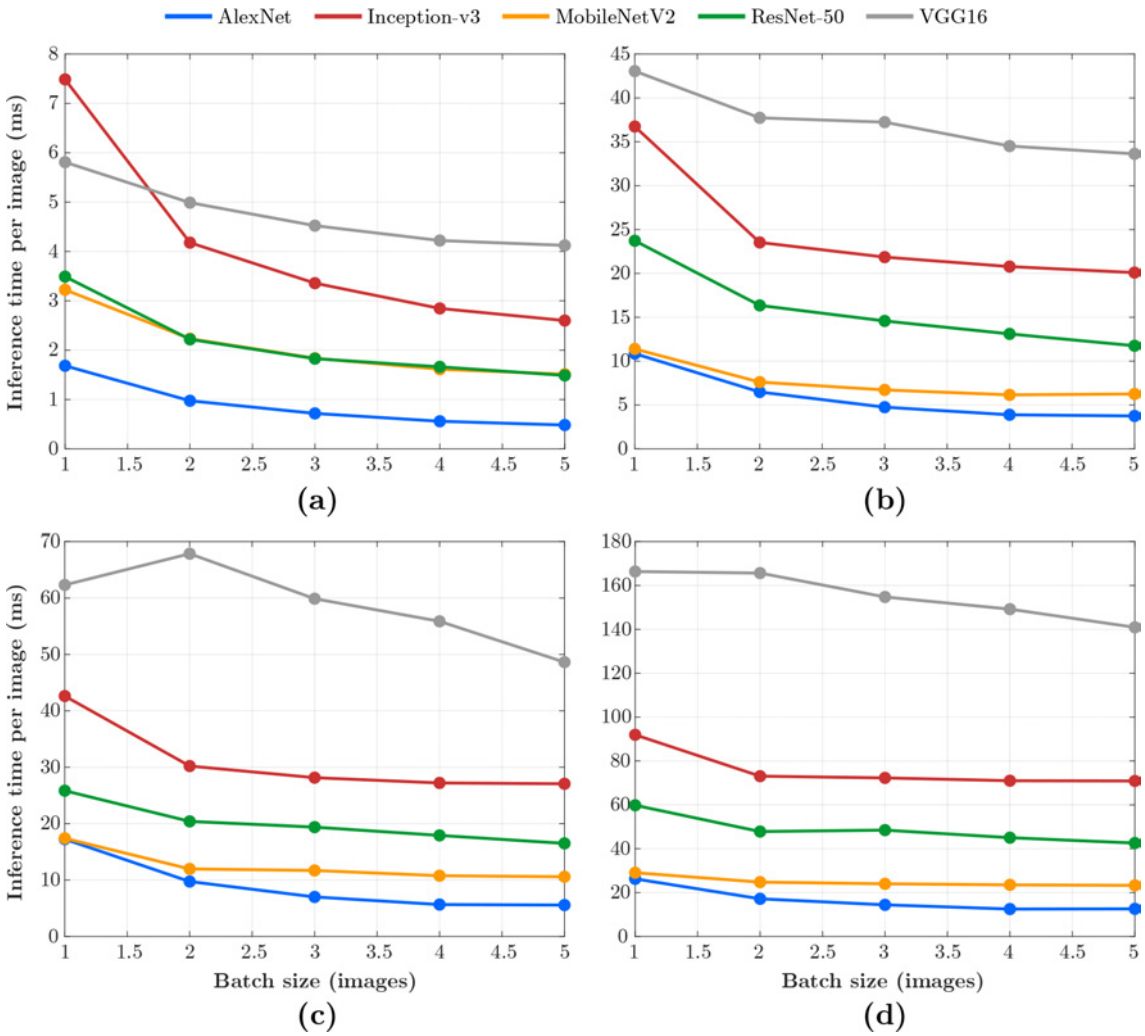


Figure 4.34: Inference time per image versus batch size for the four chosen architectures (a) GeForce GTX 1080 (b) Jetson Xavier (c) Jetson TX2 and (d) Jetson Nano.

An interesting result is presented in Figure 4.34 which plots the per-image inference time versus batch size. This shows that inference time per image has a linear relationship with batch size, however it is not one-to-one. This means that for example, two images processed together in a batch is performed faster than two images processed successively in batches of one. This is an important result for robotic weed control that indicates if you must use multiple cameras, you are better of processing them in a batch together than separately on the same device.

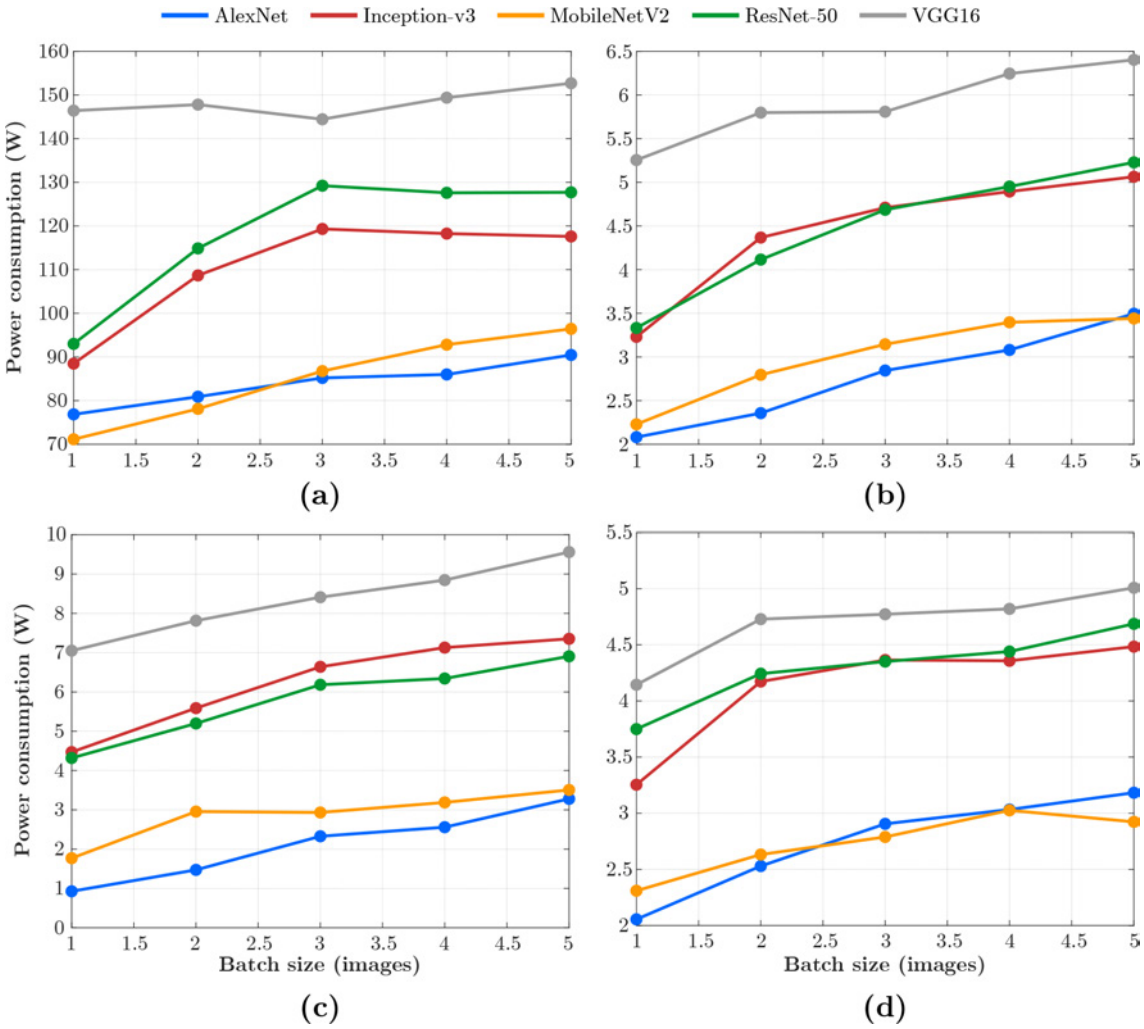


Figure 4.35: Power consumption versus batch size for the four chosen architectures (a) GeForce GTX 1080, (b) Jetson Xavier (c) Jetson TX2 and (d) Jetson Nano.

As expected, the more work being done (i.e. the larger the batch) the more power consumed (Figure 4.35). There is no free lunch here. Interestingly, the power consumption on the GeForce GTX 1080 appears to plateau for batch sizes of three and above.

Meanwhile, all models appear to have a similar linear relationship of power consumption to batch size. This indicates that no model has a batch-specific advantage over another. However, the power consumption hierarchy of models is still clearly evident with AlexNet and MobileNetV2 outperforming the other three architectures.

4.4.11 Robotic Weed Control

Let us now shape our analysis for the application of robotic weed control and discern which combination of hardware, model architecture and batch size is ideal for our prototype development. Chapter 5 covers in detail the steps involved in developing the detection system for the weed control prototype of this thesis. Therein, the decision is made to employ an array of four machine vision cameras in order to capture a 2 m horizontal field of view. For all four cameras, the field of view in the direction of travel of the robot remains fixed at 280 mm, therefore our performance target requirements remained fixed. Let us narrow our evaluation to the batch size of four in order to select the ideal hardware and model for the job.

The tradeoff between model accuracy and inference time, including a representation of the power consumption of each model for all four architectures is provided in Figure 4.36 with a batch size of four. Generally, the better the accuracy, the slower the inference time. Models that sit above the quadratic dashed line are maximising this tradeoff. The performance targets are also represented with vertical dashed lines, where applicable for the Jetson TX2 and Jetson Nano.

Clearly, ResNet-50's high accuracy and middle-of-the-road inference speed make it a strong contender for field use. What's more, it satisfies the harshest real-time requirement for all devices except the Jetson Nano. Inception-v3 is close behind ResNet-50, however it is slightly worse than ResNet-50 in all facets making it quite redundant. The next most competitive model for field implementation is MobileNetV2. It offers tremendous real-time performance at the cost of a few percentage points of accuracy. If a Jetson Nano were being used, the MobileNetV2 architecture is the only model capable of delivering high accuracy at the 20 km/hr performance requirement. AlexNet and VGG16 sit at the bottom of the list here for two opposite reasons. AlexNet is fast but inaccurate, while VGG16 is accurate but too slow to be implemented.

For the purpose of maximising the potential accuracy of our in-field prototype, it was decided that ResNet-50 was the ideal model for implementation. The prototype is not expected to be used for broadacre cropping until further iterations are ready for commercial development. Therefore, for the field trials conducted as part of this thesis, the minimum operating speed of 10 km/hr (with the time target of 100 ms) is deemed suitable. ResNet-

50 with a batch size of four comfortably satisfies this real-time requirement. We must then decide which embedded device is the most suitable for the prototype.

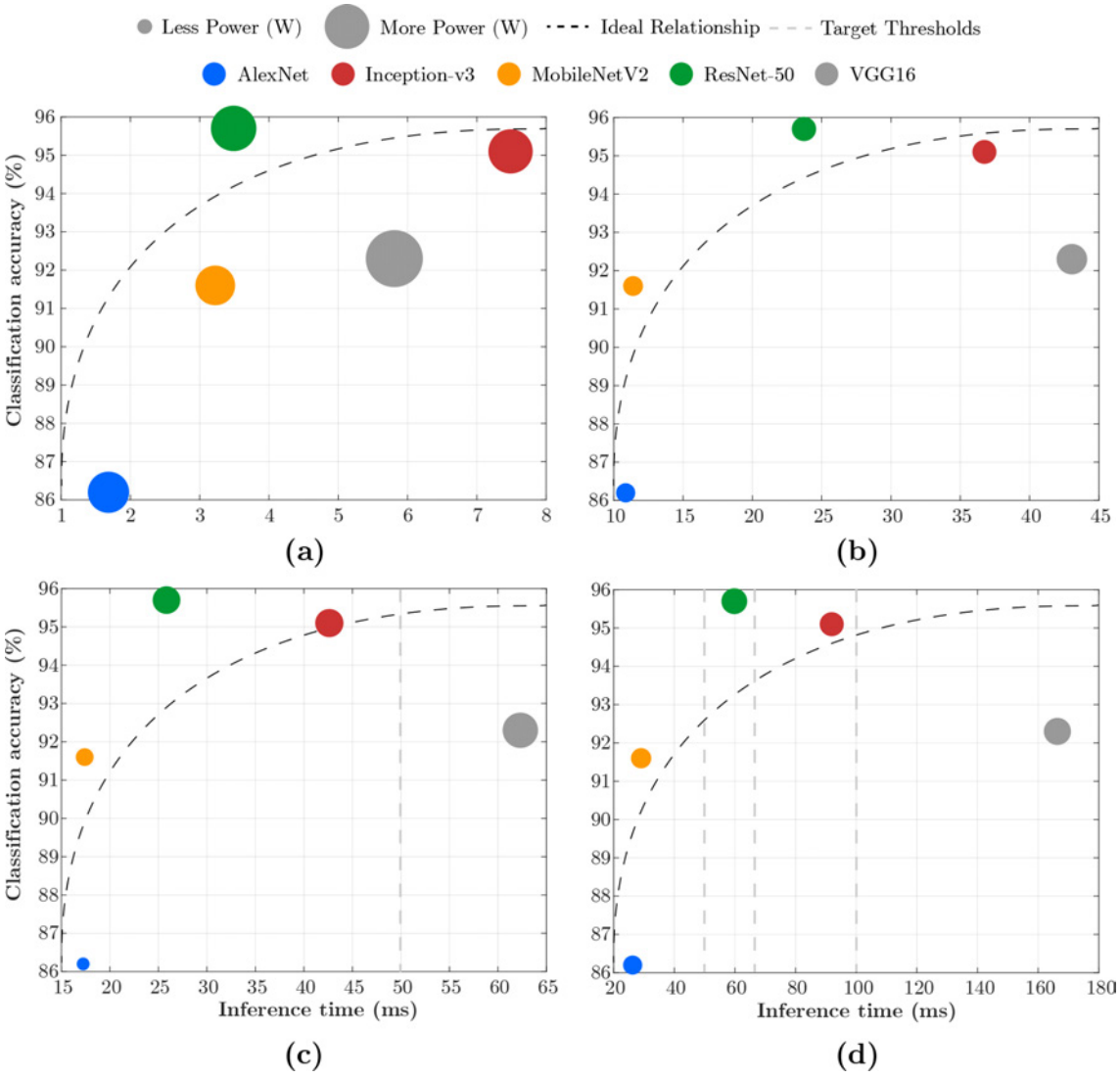


Figure 4.36: Plots of classification accuracy versus inference time with a batch size of four for all models across each architecture (a) GeForce GTX 1080, (b) Jetson Xavier, (c) Jetson TX2 and (d) Jetson Nano. The power of each model is also represented with the size of the marker.

The tradeoff between speed and power for each hardware device when deploying ResNet-50 with a batch of four images is illustrated in Figure 4.37. It also represents the price of the hardware and whether the performance targets are met. The GTX 1080 is a poor fit for edge computing and the Jetson Nano is unable to meet the 10 km/hr performance requirement for this model configuration and is also removed from consideration.

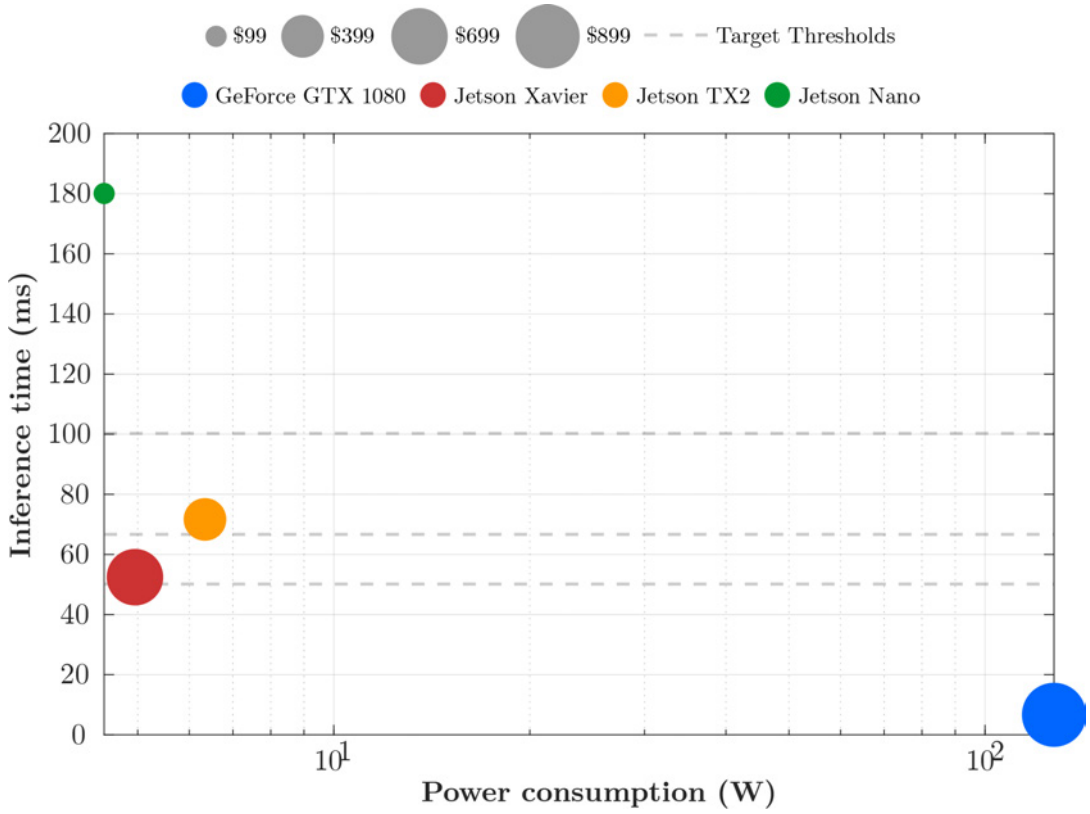


Figure 4.37: A plot of inference time versus power consumption for ResNet-50 processed with batches of four images across the four different architectures.

This leaves the Jetson Xavier ($699) and Jetson TX2 ($399). The Xavier is $300 USD more expensive but provides better operating power and speed performance. Despite this boost, the Xavier is just barely unable to meet the 20 km/hr performance requirement. Based on this tradeoff, it was decided to use the Jetson TX2 in combination with ResNet-50. The performance target of 10 km/hr will be accomplished allowing the robotic prototype to demonstrate maximum possible accuracy during its development.

The cost of this system is not only important given a fixed research budget, but it is also important for our stated end users – farmers and local landholders. As this technology comes to market, case-by-case decisions can be made as to which combination of hardware, model and batch size fit their requirements. For example, in a broadacre situation where weed spray accuracy is of paramount importance due to the value of the crop, design will favour highly accurate and fast systems despite the high cost. Whereas, for rangeland cattle farmers on pasture filled property, where spray accuracy is of slightly less consequence due to the relatively lower land value but land size is much larger, low-cost and faster designs would be favoured.

4.5 Chapter Summary

In summary, this expansive chapter has contributed the following:

- A brief review of the history of deep learning and its application to weed species detection informed the work being done to isolate deep learning as the weapon of choice, push its accuracy to new heights for weed detection and investigate its utility for real-time robotic weed control.

- The AlexNet architecture was revised and deployed to classify lantana with 88.4% accuracy, outperforming the handcrafted texture feature set developed in Chapter 2. Detailed feature analysis of the AlexNet architecture revealed that the deep learning model successfully learned the same descriptive features as the texture method, but did so automatically.

- A selection of state-of-the-art deep CNN architectures were revised and deployed to learn the DeepWeeds dataset presented in Chapter 3. AlexNet, Inception-v3, ResNet-50, MobileNetV2 and VGG16 classified DeepWeeds with 86.2%, 95.1%, 91.6%, 95.7% and 92.3% accuracy, respectively. These results were indicative of the model complexity present in each architecture.

- Some common confusions between different weed species were found to occur based on the similar image features. Similarly, the expansive class of negative plant life presents a challenge to deep CNNs with a rate of 2-7% false negatives among the models.

- The real-time performance of these models was then evaluated on a collection of edge computing devices to determine which combination thereof is most suited for robotic weed control. An expansive benchmark, the first of its kind, was produced detailing inference speed and power for the aforementioned models on the family of NVIDIA edge computing devices with varying batch sizes. This benchmark will serve as an ideal look up table for implementations for any works concerned with real-time weed or plant species detection.

- The AlexNet and MobileNetV2 architectures were found to be the fastest with inference speeds as low as 2 ms on a GeForce GTX 1080. The power consumption of the models was closely correlated with inference time. When factoring in the tradeoff between inference speed and classification accuracy, the ResNet-50 and MobileNetV2 architectures rose to the top.

- Based on the performance requirements for robotic weed control, it was decided to implement ResNet-50 on a Jetson TX2 edge computing device for the robotic weed control system to be developed in the subsequent chapter of this thesis.

The developed deep CNNs in this chapter, and the detailed selection process that investigated which model to utilise in the field based on its accuracy, speed and power performance, set the stage for the following chapter which concerns the development of the robotic weed control prototype, *AutoWeed*.

Chapter 5

Prototype Development

Agricultural automation has seen much innovation and commercialisation in the 2010s. This chapter reviews the industry need for robotic weed control to inform ground-up goals in the development of AutoWeed, a retro-fitted selective spot sprayer for crop and rangeland environments. The design concept and prototype methodology is then detailed focusing on the detection, positioning and spraying subsystems. AutoWeed is then taken through a series of experimental trials to validate its positioning and detection precision, as well as its weed control efficacy. The chapter ends with a look forward to how this successful prototype could be commercialised to provide farmers with a new tool for weed management.

5.1 Background

5.1.1 Robotic Weed Control

The recent boom in research and development in robotic weed control can be attributed to rapid developments in robotics [22, 23], deep learning [67] and edge computing devices [146, 147]. The long-term goal is a fleet of autonomous vehicles continually applying weed control treatment to paddocks with little-to-no supervision by the farmer or landholder [24, 148]. However, obstacles remain to the realisation of such a system; such as weed detection accuracy and the automation of control methods. These obstacles have rightly been the focus of cutting edge research [27, 97]. But new control methods will only be as good as the detection system it uses to target control. Therefore, accurate detection is perhaps the most critical obstacle for robotic weed control.

The primary focus of this research thesis is to improve the accuracy and robustness of weed detection for selective weed control. In order to advance this aim, we must exhaustively evaluate the detection models in situ, under the expected conditions upon which it would be utilised commercially. To support the evaluation of the weed detection systems developed in prior chapters, an in-field prototype must be developed that can deploy the weed detection systems and assess their in-field performance.

The first iteration of prototype development for this thesis (Chapter 2) utilised a towable trailer as the base vehicle for the spot spraying system. The maneuverability of the system suffered with the necessity of a tow vehicle and the required large turning radius of the trailer system. Being able to maneuver complex terrain at relatively high speeds is required when addressing weeds in large pastoral cattle stations. This differs to crop situations, where maneuverability is much simpler and speed is constant. The other important lessons learned from the first prototype include:

- Low-cost dead reckoning technology is not suitable for the required positioning accuracy and sophisticated positioning technology should be utilised.
- Much larger image datasets are required to capture the variability of the environment under which a prototype is tested in.
- A self-propelled system offers advantages over a towed system in the rangeland environment, where system maneuverability is key for operational efficiency. Therefore, a self-propelled narrow-boom sprayer is ideal.
- The robotic control method should remain simple (i.e. activation of a solenoid rather than engaging complex mechanical control technique) in order to focus efforts on image processing and computer vision.

5.1.2 Industry Need

The uptake of new agricultural technologies, which are the product of cutting edge research and development, often follow the Gartner hype cycle [149] of user adoption (Figure 5.1). The cycle begins as a technological innovation triggers press or industry interest from a publication or product launch. This creates a peak of inflated expectations where a wave of hype projects expectations beyond the current reality of its capabilities. Inevitably, these exceeding expectations are not met due to adoption pitfalls during the trough of disillusionment. Oftentimes, researchers will be funded to push a novel idea up the hype curve, cresting at the peak of inflated expectations only to have their innovation dumped in the trough of disillusionment because of no consumer pull and/or expired project funding. After this period, which can be finite or infinite; the technology may reemerge by overcoming the initial hurdles and begin the march toward a plateau of productivity.

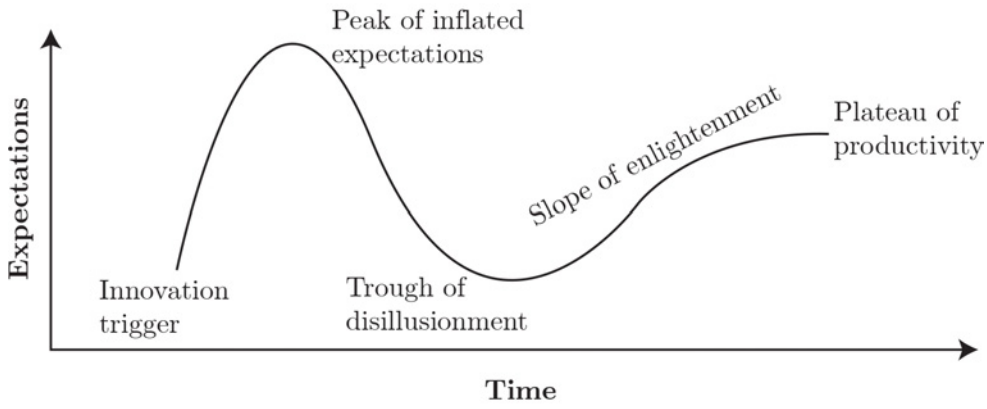


Figure 5.1: The Gartner hype cycle, a graphical depiction of a common pattern that arises with each new technology or other innovation [149].

Australian farmers are at a crisis point with an industry reliant on glyphosate for non-selective weed control and plagued by herbicide resistant weeds [150]. This is even more impacted by the threatened loss of glyphosate products due to perceived carcinogenic impacts [151]. Site-specific weed control systems that detect weeds from crop or pasture and spot-spray herbicide only where it is required have the potential to alleviate the economic and environmental impacts of current non-selective herbicidal control. With new efficient weed detection and spraying systems, there is also potential for introducing new and alternate herbicides as well as reducing the amount of herbicide in the environment. The primary goal of this research thesis is to develop a prototype for robotic weed control that

delivers on realistic expectations for commercial development. The application, utility, economic and environmental impacts of the new technology are then investigated with a view to support early and consistent user uptake.

A recent Gartner hype cycle analysis for emerging technologies in 2018 (Figure 5.2) forecasts the road to commercialisation for some of the building blocks required for robotic weed control solutions [152]. Edge AI was listed as a recently triggered innovation that may take 5-10 years to reach production. Smart and Autonomous Mobile Robots were deemed to be in the peak of inflated expectations with estimations of more than ten years until production plateau; primarily due to legal and ethical barriers to commercial uptake. Meanwhile, Deep Learning is touted at the crest of inflated expectations and expected to have an expedient 2-5 year path to production plateau. An analysis of these trends reveals that if a robotic weed control prototype relies on full automation, the path to commercialisation may be a long one. Therefore, this research thesis aims to simplify its robotic prototype by relying solely on deep learning, as opposed to other emerging technologies, in order to accomplish immediate commercial activity and provide farmers with a more useful tool in a shorter time span.

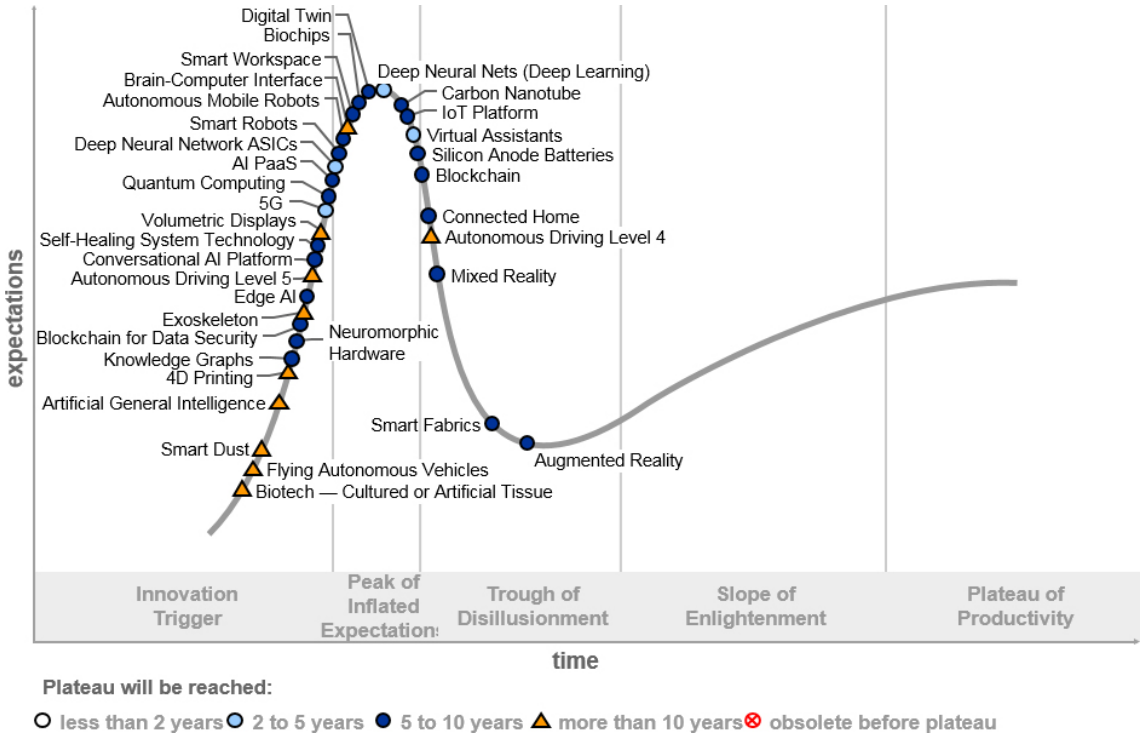


Figure 5.2: The Gartner hype cycle for emerging technologies, as of August 2018. © 2018 Gartner, Inc. [152]

5.1.3 Goals

Subsequently, the following development goals were set for the second stage prototype:

1. Employ the latest commercially available and sophisticated GPS, IMU and Kalman filter positioning technology to accomplish the required positioning accuracy. To achieve 90% spray coverage of weed targets 300 mm in diameter with a 250 mm spray swath, the required positioning accuracy is at least $\pm 135mm$.

2. Use the developed system to collect and label large site-specific image datasets with speed and ease in the field.

3. Utilise a small self-propelled farm vehicle; either an All-Terrain Vehicle (ATV) or a Utility-Terrain Vehicle (UTV) as the robotic platform base.

4. Have the vehicle be driven by an operator and automate weed detection and control.

5. Use herbicide application as the treatment due to its simplicity to automate (i.e. simple activation of a solenoid rather than complex mechanical componentry).

5.1.4 Funding

After completing the first prototype and site-specific case study, this research thesis led the successful application for a two-year $298,949.20 Commonwealth Grant from the Control Tools and Technologies for Established Pest Animals and Weeds Competitive Grants Programme in 2017 [9]. This grant allowed significant funding to further the robotic prototype discussed in this chapter.

Furthermore, Yamaha Motor Australia kindly donated a Yamaha Grizzly 700 ATV to the project to serve as the base vehicle for robotic weed control research. With these sponsorships, the scope for this chapter could be fully realised to address the needs of industry.

5.1.5 Design Concept

We propose to adapt a semi-autonomous intermediary step in the technological chain for robotic weed control; a manned agricultural vehicle that automatically detects and applies herbicide to weeds (Figure 5.3). This simpler solution combines proven spraying and motion sensing technology with the emerging technology of deep learning. As such, early adoption obstacles of autonomous robotics can be bypassed and a solution that is ready for rapid commercialisation can be delivered.

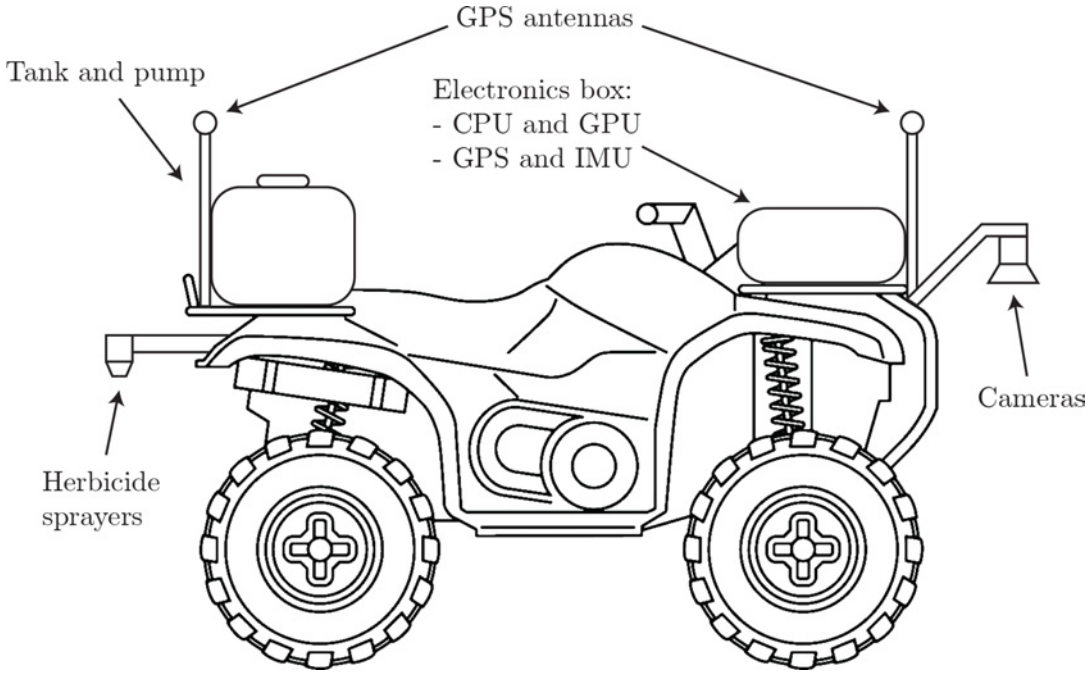


Figure 5.3: Illustrated design concept of the robotic weed control prototype.

5.2 Methodology

5.2.1 Overview

System

The prototype system consists of a novel design of a see-and-spray system that has been retro-fitted to a Yamaha Grizzly ATV (Figure 5.4). A row of four cameras mounted on the front of the ATV face downward and capture images of weeds as they are passing under the vehicle (Figure 5.5). The images are then processed to identify and locate weed targets. Upon weed detection, the tracking system will determine where the weed is located in reference to the vehicle using regular updates from the vehicle positioning system. When the tracking system predicts that a weed will pass under any of the five sprayers it will activate those sprayers for a time adequate to ensure weed plants receive appropriate coverage of herbicide treatments.

Three major subsystems within the prototype required significant software and hardware development: i) the detection system, encompassing the optical setup and weed detection algorithms, ii) the positioning system, including vehicle and target tracking, and iii) the spraying system, which focuses on the spray technology hardware and developed electronics.

Figure 5.4: Front (top) and back (bottom) views of the AutoWeed spot-spraying prototype photographed during a field trial on a sowthistle infested wheat crop in Arcturus Downs, QLD.

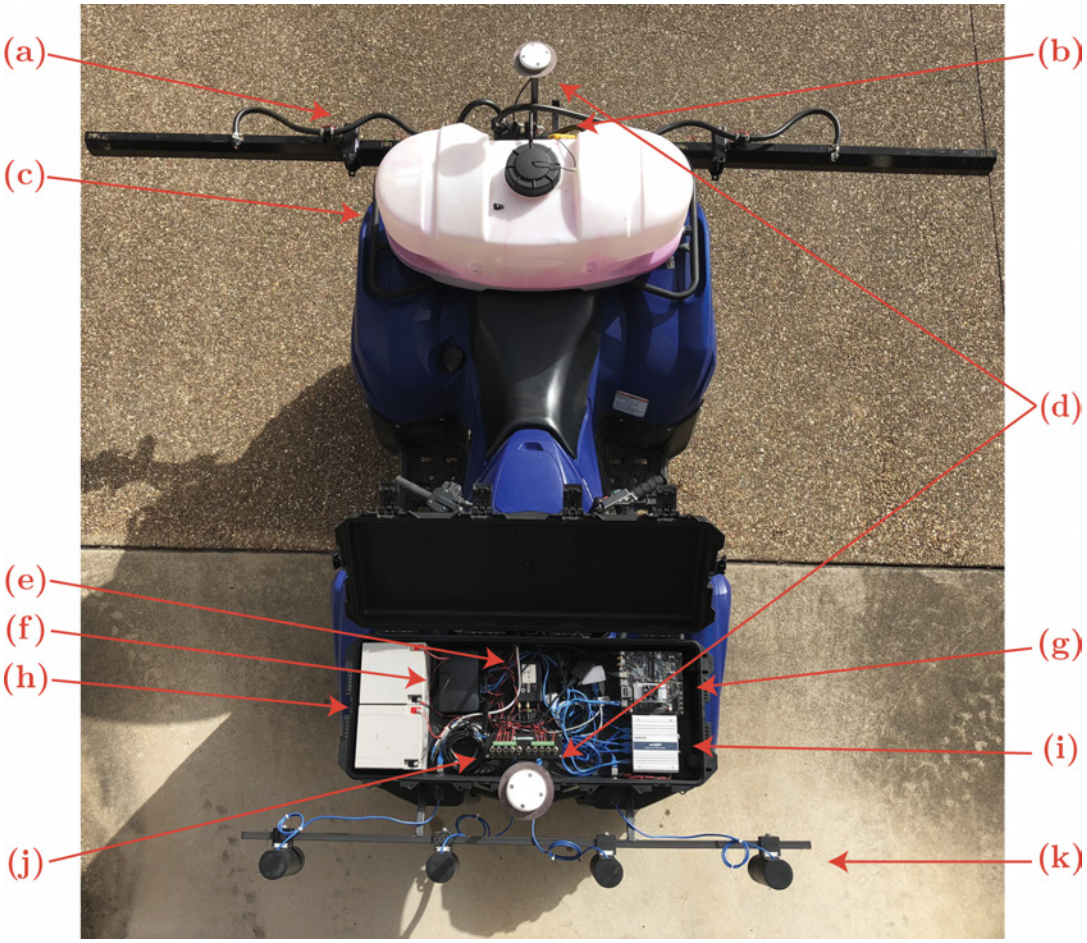


Figure 5.5: A top-down view of the AutoWeed system with the following labelled subcomponents: (a) the solenoid sprayers, (b) the solenoid control board, (c) the pump and tank system, (d) the two GPS antennas, (e) the GPS/IMU system, (f) a wired and wireless router, (g) the stacked NVIDIA Jetson TX2 compute modules, (h) two 12 V batteries, (i) an Ethernet switch, (j) the AutoWeed power switch board and (k) the camera and lens units.

Software

All software developed for the robotic prototype was written in C++ and utilises the methods, data structures, and messaging systems provided by the open source Robot Operating System (ROS). The primary component of ROS is the anonymous publisher/subscriber messaging system. This messaging system is used to generate messages about the state of the system and seamlessly deliver them to other programs which require them. Individual ROS nodes were written for each subsystem of the prototype system, which communicate via custom made ROS topic messages (Figure 5.6).

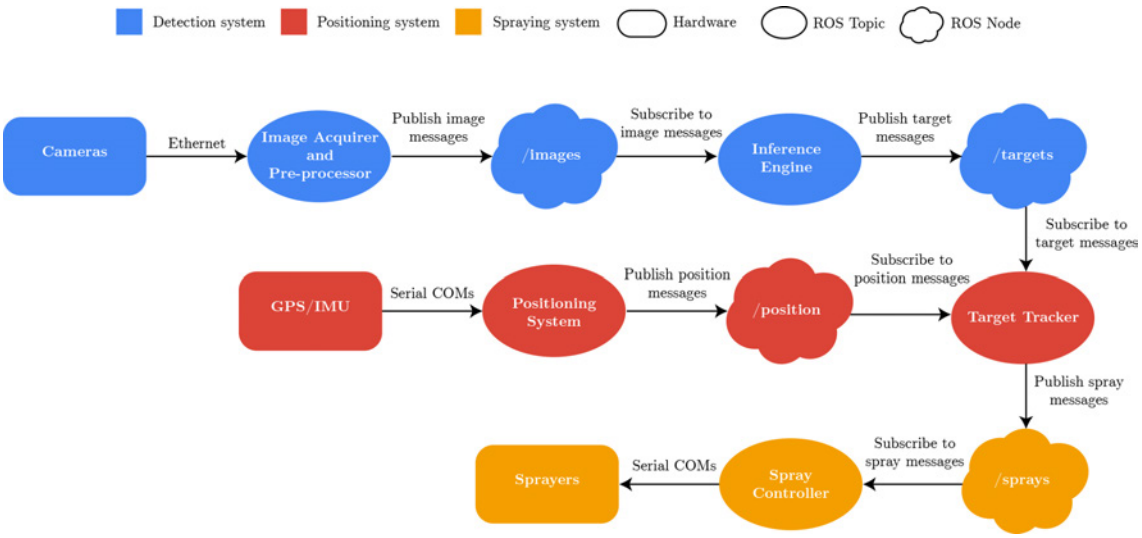


Figure 5.6: An illustrated system diagram flow chart for the ROS software and associated input and output hardware.

The three primary hardware components of the system; the cameras, positioning unit and sprayers, communicate with the custom-developed ROS software programs – or nodes – to simultaneously publish images, process images and weed targets, utilise the vehicle position and weed targets to publish sprayer commands Figure 5.6. This interconnected flow chart explains the entire system prototype.

ROS' publisher/subscriber system has been exploited in the construction of our software. By adhering to this system, we can utilise ROS' in-built logging and error handling routines, messaging system, and their open-source user-built libraries for interfacing with our cameras and positioning hardware.

5.2.2 Robotic Platform

Base Vehicle

One of the tools of choice for rangeland weed control currently, is hand gunning of herbicides from the back of an All Terrain Vehicle (ATV). A logical next step in the automation of this practice is to take these existing vehicles and retro-fit them with state-of-the-art weed detection and spraying technology. This approach is also scalable to crop situations, where the detection system can be mounted onto self-propelled or trailed boom sprayers.

The base vehicle for the new prototype was an industry-donated 2017 Yamaha Grizzly 700 ATV (Figure 5.7). It is perfectly suited for the difficult Australian rangelands and can be used in cropping paddocks to site-specifically target low density weed populations.

Figure 5.7: A 2017 Yamaha Grizzly 700, donated by Yamaha Motor Australia to serve as the base vehicle for our prototype.

NVIDIA Jetson TX2 Development Boards

After careful consideration of the range of available edge computing hardware, the NVIDIA family of Jetson compute modules were identified as the leading candidates for processing on our robotic prototype. These devices were chosen due to their compact form, processing capabilities, in-built input and outputs and future proof operating system. Designed specifically for machine vision and image processing in robotics, the NVIDIA Pascal GPU, which is native to the Jetson family architecture, enables parallel processing of images captured from the Ethernet machine vision cameras.

A suite of other edge computing processing units were considered, including the BeagleBone Black, Raspberry Pi 3, Intel Neural Compute Stick, Google Coral Development Board and the Movidius Neural Compute Stick. A comparison of the performance benchmarks for some of these devices is shown in (Figure 5.8) shows that the Jetson family offers top-of-the line performance compared to the Raspberry Pi 3 and a Raspberry Pi 3 augmented with an Intel Neural compute stick [154]. The Jetson Nano offering is low-cost with the lowest performance, while the remaining modules from the Jetson family of devices (the Jetson TX2 and Jetson Xavier) offer increased processing power at a higher

price. Also, because our deep learning development utilises the NVIDIA GPU library of software and devices, there is a seamless work-flow to transfer our developed detection models onto NVIDIA edge hardware.

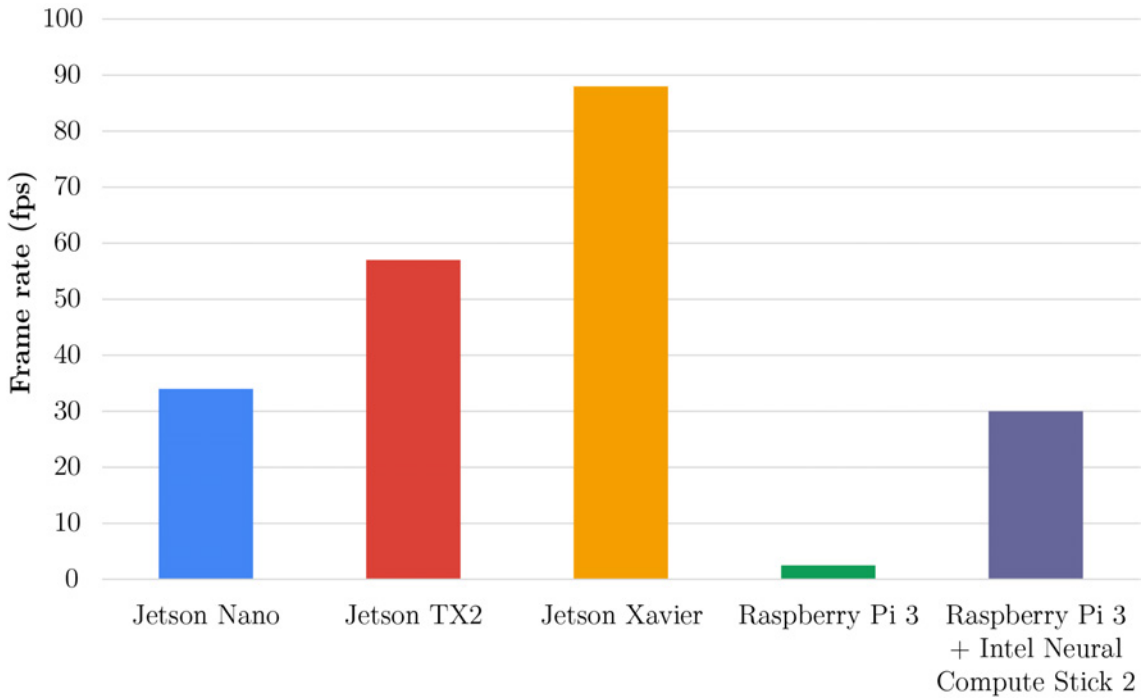


Figure 5.8: A comparison of inference performance frame rates (in frames per second) between various edge computing devices running a 300×300 px MobileNetV2 classification model trained on ImageNet [154].

One of the primary reasons for selecting the Jetson family of modules is that it has a selection of I/O available on board, specifically CAN, UART, I2C and Ethernet, which will be used to interface with the subsystems on the ATV in both the weed detection and spraying system. Furthermore, many of the interfaces on the Jetson are provided as pin headers, allowing the flexibility to extend the capabilities of the machine. This flexibility is similar to that of single board computers like the Raspberry Pi or Beagle Bone Black. The Jetson, however, offers computing performance far exceeding either of those competing single board computers.

After narrowing the design selection down to the NVIDIA Jetson family, it must still be decided which Jetson model to adopt. Section 4.4 of Chapter 4 presented a comparison of the core offerings in the NVIDIA Jetson family. All models were found to exceed the required 10 fps frame rate for deploying inference engines on a live robotic platform. However, it was found that the Jetson TX2 (Figure 5.9) presented the best trade-off

between price and computing power; with the Nano and Xavier on the lower and higher ends of the spectrum, respectively.

Figure 5.9: An NVIDIA Jetson TX2 compute module [155].

Astro and Elroy

The benchmarks comparing the inference speed and power requirements of the NVIDIA Jetson family of compute modules showed that each device was capable of meeting the required frame rates for processing a ResNet-50 model in real-time. However, this was under the assumption of one camera being used for the robotic system. The optical system for our prototype, which will be discussed subsequently in greater detail, utilises four cameras simultaneously to provide a 1.75 m field of view for the robot. This means our time requirement is quartered. The Jetson TX2 can inference the ResNet-50 (224 x 224) full precision network at 40 fps for one camera and one image per batch. Assuming the inference time increases with an increase in batch size, we reason that a batch of four images could be processed in at least 10 fps on the Jetson TX2; which meets the speed requirement for the prototype.

However, this analysis assumes one Jetson TX2 processing unit is entirely dedicated to performing inference of the weed detection models. As a result, this prototype utilises

two Jetson TX2 compute modules, hereafter named *Astro* and *Elroy*. Elroy performs sophisticated inference engine operation and publishes weed targets to the ROS server. Meanwhile, Astro performs all other mundane tasks including: acquiring images, collecting position data and activating the sprayers (Figure 5.10).

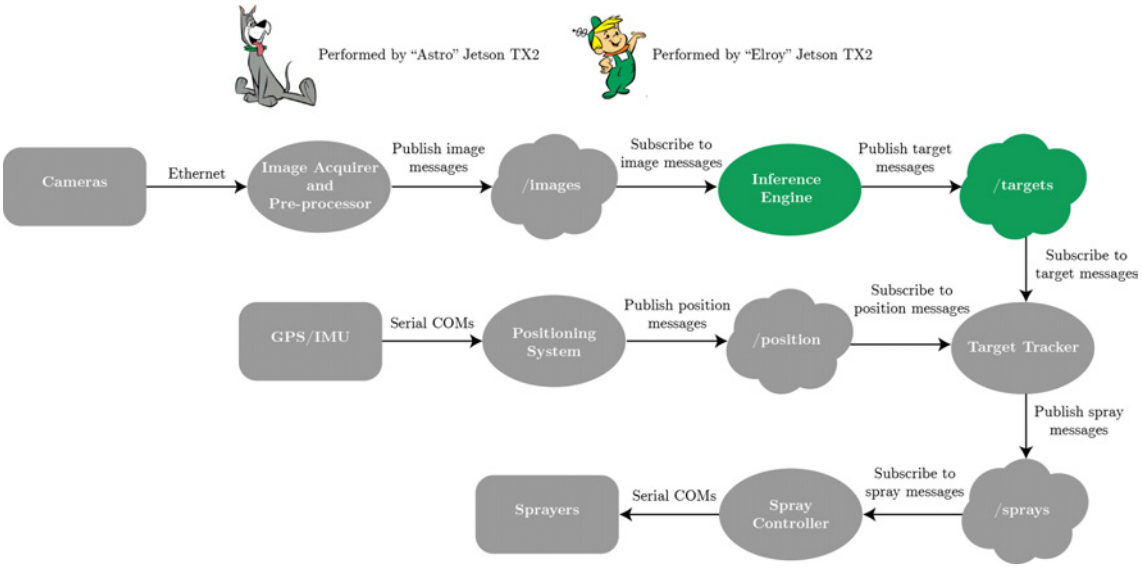


Figure 5.10: A component breakdown of which software tasks and hardware interfaces fall under the responsibility of the *Astro* and *Elroy* Jetson TX2 processing units.

This setup is possible due to the built-in distributed computing architecture of ROS. Multiple computers running different operating systems and software are seamlessly able to communicate provided they exist on the same Ethernet network. Astro and Elroy are both hardwired to a local Ethernet network on the prototype. The on-board Ethernet router is also wireless, allowing an external laptop or hand-held device to join the network and act as a control or user interface for the prototype.

Power Supply and Distribution

To power the subsystems of this prototype, a choice presented itself to either use the power generated by the magneto of the ATV or to use additional batteries. After reviewing the total power requirements of the subsystems, it was decided that both options would be required. The ATV battery and magneto are capable of supplying 120W at 12 V, however, this supply is very noisy due to the magneto generator and the fuel ignition system.

The sprayer system (including both the solenoids and the spraying pump) was powered entirely from the ATV supply and kept separate from any additional batteries used to power the electronics. This is due to their high power draw and noise generation on

the power line when activated. The sprayer solenoids individually require 18 W instantaneously, but drop to around 8 W during operation. The worst case power drawn from the prototype occurs when all five solenoids are activated at the same time, totaling 90 W momentarily before dropping (within 20 ms) to 40 W during operation. With the ATV able to provide 120 W at 12 V, the ATV battery and magneto is adequate to power the sprayer system.

Figure 5.11: A breakdown of which hardware components were powered by which battery power supply.

Table 5.1: Typical and maximum power consumption statistics for the various hardware components within the prototype.

Hardware	**Typical power (W)**	**Maximum power (W)**
Advanced Navigation Spatial Dual	7	7
Astro Jetson TX2	8	15
Elroy Jetson TX2	8	15
Netgear wireless router	12	12
Korenix PoE switch (incl. cameras)	45	95
Solenoids	8	18
Spraying pump	10	20
Total	98	182

The Jetson processing units and the wireless router were powered from an isolated battery setup (Figure 5.11). These electronics are susceptible to brown out or power cycling if there is too much noise or dips in the supply rail, which therefore needs to be separate from other, noisier power loads. The second battery was used to power the Power-over-Ethernet (PoE) switch for the cameras and the Advanced Navigation positioning system. The relatively high power consumption of the PoE switch had been found to cause brown outs of the sensitive electronics supplied by the first battery.

For testing and development purposes, the system required being able to run on battery power for the duration of a normal working day (i.e. approximately eight hours). This

minimises time lost due to charging and having to shutdown and restart the system if transferring the power source to mains. Based on the individual power requirements of the subcomponents, the typical expected power usage is 98 W $\approx$ 12 V @ 8 A (Table 5.1). Therefore, to power the system for eight hours, the battery capacity needs to be 64 Ah $\approx$ 8 A $\times$ 8 h.

The battery choice was determined by physical size, capacity and, battery type. A maximum height limit of 15 cm was required to fit into the mounting case of the vehicle. Two batteries were used to achieve the required capacity of 64 Ah, in addition to the existing battery of the ATV system. This is more than adequate as the typical idle current was 2 A ($\approx 24W$) and the peak current is 4 A ($\approx 48W$). The batteries were also required to be sealed gel type to ensure they did not pose an acid spill risk if they were tipped over for any reason. A Diamec 12 V 26 Ah sealed gel battery was chosen (Table 5.2).

Table 5.2: Technical specifications for the chosen Diamec 12 V battery.

Feature	**Specification**
Model	Diamec 12 V 26 Ah Battery (DMD12-26 GEL)
Type	Gel Type Lead Acid Deep Cycle
Dimensions	175 mm (L) x 134 mm (W) x 126 mm (H)
Voltage	12 Vnom
20 Hour Capacity	26.0 Ah @ 1.3 A
10 Hour Capacity	21.1 Ah @ 2.11 A
5 Hour Capacity	18.7 Ah @ 3.74 A
1 Hour Capacity	14.0 Ah @ 14 A

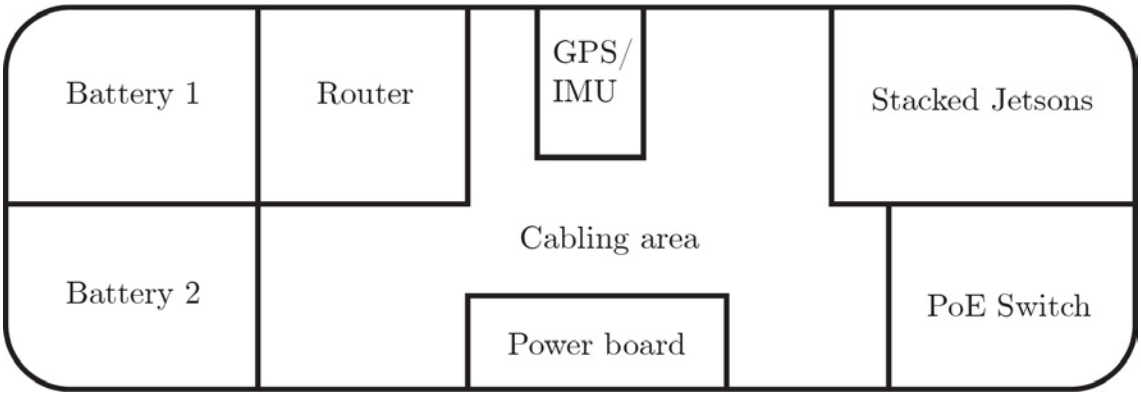
Mounting and Assembly

A key problem with the initial prototype in Chapter 2, was poor mounting of the electronics and hardware leaving the system exposed to the environment, shock, and vibration. This led to a variety of issues for hardware failure, making troubleshooting, debugging and fault finding extremely challenging. To mitigate this, a box was chosen that could fit all the required electronics, seal them from dust and water ingress, and fit on the front ATV rack so as to not impede the view of a human driver.

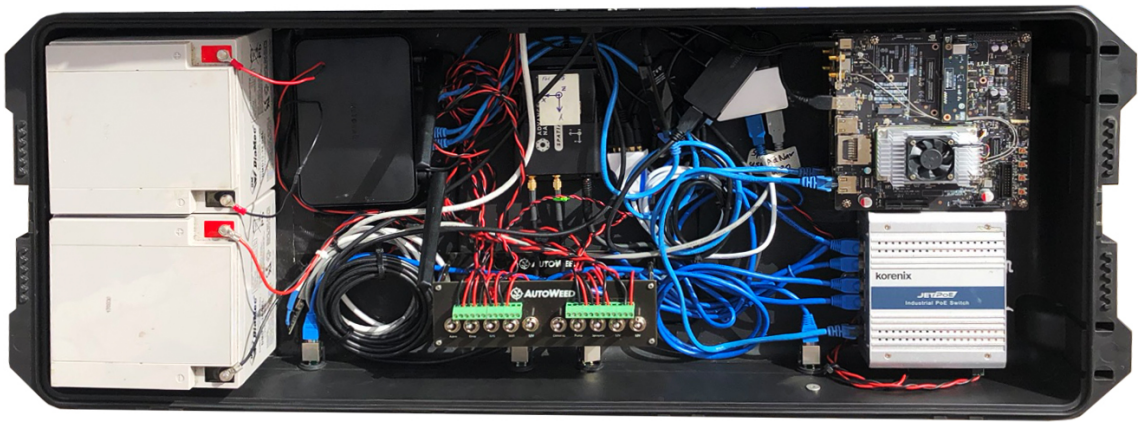
The only suitable box found was the Pelican iM3100 Storm Long Case based on its dimensions compared to the required size limits (Table 5.3). Building a custom-made box may have been possible but would have taken more time and money in comparison to purchasing an off-the-shelf case.

Table 5.3: Dimensions of the chosen iM3100 Pelican case compared against the derived size limitations.

Pelican iM3100	**Length (cm)**	**Width (cm)**	**Height (cm)**
Interior	92.7	35.6	15.2
Exterior	101.1	41.9	17
Size Limit	110	50	20



(a)



(b)

Figure 5.12: The planned (a) and actual (b) layouts of the electronics box mounted to the front rack of the ATV prototype.

Placement of items within the box required careful consideration (Figure 5.12). Items such as the Advanced Navigation unit required isolation distances from magnetic fields induced by Direct Current (DC) wires and the physical dimensions of the required electronics limited the internal capacity of the box. The batteries in particular required an internal height of at least 13 cm. The Pelican iM3100 box was chosen first and the bat-

tery model was later chosen to conform. All subcomponents were then held in place with Velcro to ensure robustness to travel and mechanical vibration.

5.2.3 Detection System

Optical System

Oftentimes image processing frameworks fail in real world application because they are hamstrung by unforeseen errors during the first and most important step in the framework: image acquisition [96]. The images acquired must match the target application as closely as possible for real world success. Therefore the design of the optical system is a very important undertaking.

The design of the optical system starts with an understanding of the application at hand which creates boundary conditions that inform the selection of a camera and lens combination to achieve the desired image. For our application, there are a number of boundary conditions which affect the design of the optical system. These include:

- The height from the camera lens to the ground was chosen to be 1 m in order to allow the solution to target weed regrowth for a variety of weed species. Similarly, the ground clearance underneath the robotic vehicle is 288 mm. Weed targets will rarely exceed this height, therefore the optical system should have a depth of field of approximately 288 mm.

- No external shading or lighting is to be used for the optical system, so as not to limit the vehicle's maneuverability. The camera and lens must be chosen and utilised to adequately capture the dynamic range lighting in the scene.

- The ideal vehicle speed while spraying is 10 km/hr. The chosen camera must be able to operate at a shutter speed fast enough to maintain focus despite vehicle motion.

- The ideal field of view of the weed treatment system should be approximately 2 m in total. This will be both the visible swath and the range set for the nozzle system.

- The system will operate under harsh environmental conditions. Therefore we look exclusively at machine vision cameras and lenses to provide a robust mechanical specification for in-field use. Similarly, it is beneficial for external camera and lens parameters (such as focus, iris and zoom) to be fixed for the system to be robust to mechanical vibration.

- Prior image analysis work in Chapter 2 revealed an ideal resolution of 4 px/mm is required for detailed texture discrimination between weed species. To ensure our

optical system can perform this work if required, we must have a minimum resolution of 4 px/mm.

The design methodology in selecting the camera and lens is documented below. Firstly, an image sensor was chosen based on the boundary conditions. FLIR machine vision cameras (formerly Point Grey) are renowned for their robustness in remote applications. The author of this thesis has significant experience developing software using the FLIR machine vision camera software development library. As such, the scope of cameras for selection was narrowed to the FLIR machine vision suite.

To achieve a resolution (r) of 4 px/mm with a total horizontal field of view ($HFOV_{total}$) of 1,800 mm, a combined image pixel width (IW_{total}) of 7,200 px is required, as defined in equation 5.1.

$$IW_{total} = HFOV_{total} \times r = 1800 \times 4 = 7200 \text{ px} \tag{5.1}$$

This is too large a resolution and field of view to adequately image with a single camera at a 1 m working distance without significant distortion. The decision was made to utilise four cameras to achieve the combined field of view, knowing that images could be processed in parallel. With four cameras ($n = 4$), a minimum pixel width for our desired image sensor (IW_{camera}) is 1,800 px.

$$IW_{camera} = \frac{IW_{total}}{n} = \frac{7200}{4} = 1800 \text{ px} \tag{5.2}$$

The FLIR BlackFly Gigabit Ethernet 23S6C-C camera, pictured in Figure 5.13 with a Sony IMX249 image sensor was chosen. It has a 1920 × 1200 px resolution image sensor, which would achieve a resolution of 4.267 px/mm for our 1800 mm field of view with four cameras. Other features of prominent importance include:

- Ethernet interface for data transmission and power supply. Ethernet was preferred over USB due to its more robust cabling connection and compatibility with the Jetson TX2 modules.

- A large dynamic range of 66.65 dB will achieve suitable acquisition of detail in the shadows and highlights of dynamically lit scenes in full sunlight. This image sensor has the third largest dynamic range of all FLIR machine vision camera models (Figure 5.14).

- The camera has a minimum exposure time of 0.019 ms. This more than meets the shutter speed requirement to resolve image focus at speeds greater than 10 km/hr.

- The camera software allows auto-adjustment of exposure and gain to balance image quality.
- The camera can achieve a maximum frame rate of 41 fps. This quadruples the required frame rate of 10 fps to operate at 10 km/hr.
- The camera image sensor has sensor format of 1/1.2" and an image sensor pixel size of 5.86 μm.
- The camera has a standard machine vision C-mount lens holder which allows the use of the widest possible range of lenses.

Figure 5.13: Image of the FLIR BLFY-PGE-23S6C-C camera.

Table 5.4: Some key technical specifications of the FLIR BFLY-PGE-23S6C-C camera.

Feature	**Value**
Resolution (px)	1920 × 1200
Image sensor format (")	1/1.2
Pixel size (μm)	5.86
Dynamic range (dB)	67.67
Interface	Gigabit Ethernet
Maximum frame rate (fps)	41
Lens mount	C-mount
Image sensor type	CMOS
Minimum shutter speed (ms)	0.019
Dynamic range (dB)	67.37
Dimensions (mm)	29 × 29 × 30
Power consumption (W)	2.5
Mass (g)	36

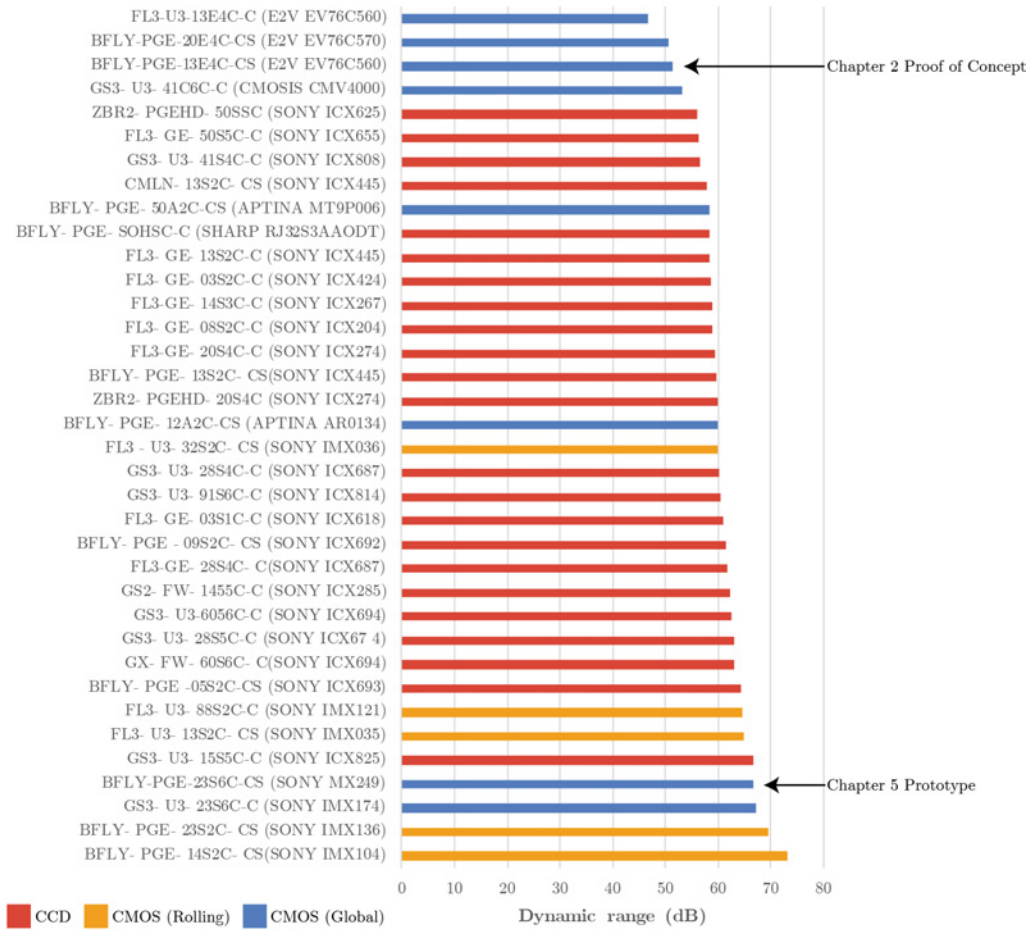


Figure 5.14: A comparison of the dynamic range of FLIR's 2017 range of machine vision image sensors.

Now that the camera and image sensor are decided upon, equation 5.3 can be used to calculate the required lens focal length (FL) from the desired field of view of one camera $HFOV_{camera}$), the horizontal sensor size of the camera image sensor (SS) and the working distance from the lens to the target (WD), all in mm.

$$FL = \frac{SS \times WD}{HFOV_{camera}} \tag{5.3}$$

The field of view of each camera can be determined from the number of cameras chosen and the desired total field of view ($HFOV_{total}$) (Equation 5.4). Each camera therefore is required to cover a field of view of 450 mm.

$$HFOV_{camera} = \frac{HFOV_{total}}{n} = \frac{1800}{4} = 450 \text{ mm} \tag{5.4}$$

The horizontal size of the image sensor can be approximated from the resolvable pixel size (PS) of the image sensor and the resolution width of the sensor in pixels (IW) (Equation 5.5). The horizontal image sensor size of our chosen camera is therefore approximately 11.25 mm.

$$SS = PS \times IW = (5.86\ \mu\text{m}) \times (1920\ \text{px}) = 11.25\ \text{mm} \tag{5.5}$$

Substituting the image sensor size, working distance and horizontal field of view for each camera into equation 5.3 we can determine the required focal length. Equation 5.6 shows that a lens with a focal length of 25 mm is desired for our target application.

$$FL = \frac{SS \times WD}{HFOV} = \frac{11.25 \times 1000}{450} = 25\ \text{mm} \tag{5.6}$$

The next important lens characteristic is the lens format. The lens format must be larger or equal to the chosen image sensor format in order for the lens to project an image large enough to be captured on the image sensor without vignetting. From Table 5.4, the camera image sensor format is 1/1.2". Therefore our desired lens must have a lens format $\geq$ 1/1.2". Secondly, to achieve a depth of field large enough to keep our weed targets in frame (approximately 288 mm), a small aperture size is required. We therefore restrict our lens criteria to include lenses with an aperture of at least $f/8$ in size. Finally, we require a C-mount lens to match with the lens holder on the chosen camera. This criteria led to the selection of the Fujinon CF25HA-1 mega-pixel 1", $f/1.4$ – $f/22$, 25 mm focal length lens, photographed in Figure 5.15.

Figure 5.15: Image of the Fujinon CF25HA-1 lens.

The lens' mode of operation was also designed to resolve detail in the shadows and highlights of high dynamic range scenes without motion blur while moving at high speeds. This was achieved by selecting an aperture size of $f/8$ to allow some sunlight in; while simultaneously restricting the shutter speed to less than 0.05 ms. The automatic exposure

and automatic white balance algorithms within FLIR's FlyCapture Software Development Kit were utilised to achieve acceptable imaging without the need for manual tuning between different sites. Together this camera and lens combination provides a 450 × 280 mm field of view for a single camera image. In total we have a theoretical field of view of 1800 × 280 mm. The final designed optical system is illustrated for posterity in Figure 5.16 below.

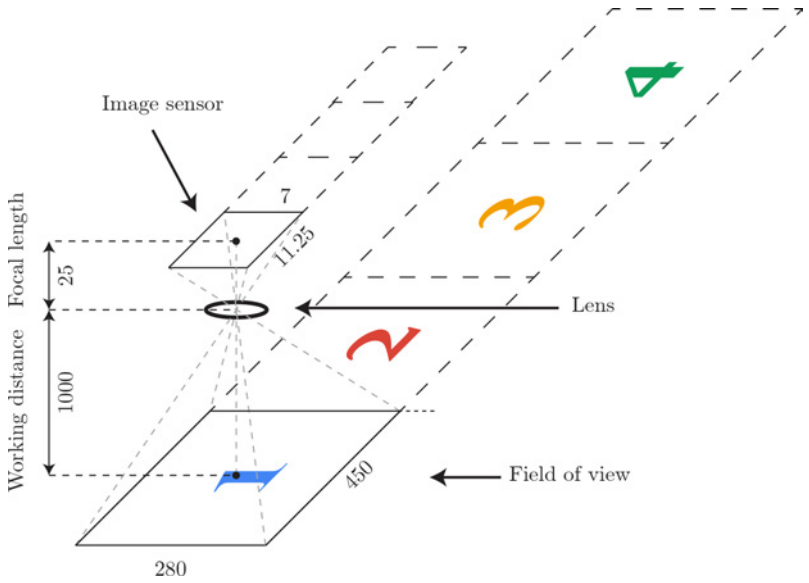


Figure 5.16: A dimensional rendering of the designed optical system. All measurements are in mm. (Note that this drawing is not to scale.)

Next we must consider the transmission of image data from the cameras to the processing units. With four Gigabit Ethernet cameras operating simultaneously, a four port Gigabit Ethernet switch is required to transmit the data to a single Ethernet port on the Jetson TX2 compute module. The switch is also required to provide Power-over-Ethernet for simpler powering of the cameras. The Korenix JetNet 3180G eight port power over Ethernet (PoE) switch (pictured in Figure 5.17) was used to fulfill these criteria.

Figure 5.17: PoE Korenix switch used for image data transmission on the prototype.

Detection Framework

A ROS software framework was developed to perform image acquisition, image pre-processing, real-time inference and target generation (Figure 5.18). Four identical instances of ROS nodes are generated for the four simultaneously operating cameras. The raw image from the camera is collected and published to its own ROS topic for the purpose of dataset collection and graphical user interface display. Four identical image preprocessor ROS nodes are also generated that perform the relevant pre-processing steps to prepare data for the inference engine. It is here that the four parallel streams merge into one as batches of four images are processed by the single TensorRT inference engine ROS topic which publishes found targets.

Figure 5.18: A detailed illustration of the ROS software flowchart for the detection system.

Camera Driver

The ROS open-source FLIR (formerly Point Grey) FlyCapture camera driver library [156] was used and modified to suit the specific purpose of this prototype. A single camera driver ROS node was developed that could be launched simultaneously with separate camera identification codes in order to connect to four cameras simultaneously. The camera driver was also modified to instantiate the cameras for image acquisition with the following settings:

- Fourfold pixel binning was adopted to reduce to the 1920 $\times$ 1200 px images to 480 $\times$ 300 px, since the deep learning methods developed in Chapter 4 achieve strong performance on significantly down-sampled images. This sub-sampling is performed on-board the camera, which allows for a boost in acquisition speed.
- Apply full 24 bit colour processing on-board the camera to achieve better image quality.
- Implement a custom auto exposure algorithm that restricts the camera exposure time to $\leq$ 0.05 ms and provides full gain range variation.
- Publish the acquired raw 480 $\times$ 300 px images to the camera-specific raw image topic.

Image Pre-processing

Each raw image is then fed into a camera-specific launched image pre-processing ROS node that performs relevant pre-processing steps prior to real-time inference. The pre-processing steps applied include:

1. Extract the timestamp of the collected raw image and optionally save along with the raw image for dataset collection.
2. Convert the image from the FLIR library image format into an OpenCV image format to allow the application of various OpenCV algorithms.
3. Down-sample the 480 $\times$ 300 px image to 255 $\times$ 255 px, the input size for a single image in our developed inference engine models.
4. Re-arrange the tensor data format from NHWC to NCHW, as discussed in Chapter 4 to match the default TensorRT tensor format.
5. Publish preprocessed image data to a unique topic where it will be collected into batches for real-time inference.

It is worth noting that no augmentations are applied to the newly collected data when deploying our detection models in the field. Pre-processing augmentations are applied to the training data when training models, as opposed to the test data for deploying models. Usually the test data has no pre-processing augmentations applied beyond resizing the image for input to the network.

Inference Engine

A single inference engine ROS topic was developed that utilises the TensorRT library [126] and performs inference on a batch of four images from the four cameras at an average frame rate of 70 fps. The inference engine ROS node performs the following tasks:

- Load, convert and construct the TensorRT inference engine from an HDF5 Keras/TensorFlow model file.
- Subscribe to the preprocessed data topic and wait until a new set of preprocessed image data is collected from the set of four cameras. Once the image data from all cameras are ready, prepare the data into a batch allocated to CUDA memory on the Jetson TX2 GPU.
- Perform inference on the CUDA batch and extract the resulting score and classification result.
- If the classification result is not negative (i.e. a weed of some type is found) and the classification score is greater than a given decision threshold, then publish the target for spraying.

Dataset Labelling

Perhaps the most time consuming part of deploying machine learning models for a real-world application is the curation and labelling of large datasets. As detailed in Chapter 3, a software application was developed to accelerate the labelling process. Images are collected conforming by filename to the input conditions for the software program. The program expedites labelling by presenting four images at a time for labelling and allowing the user to click (or touch) the presented image to label it as a target. A screenshot of the program, developed by Alex Olsen and Benjamin Girgenti, is provided in Figure 5.19.

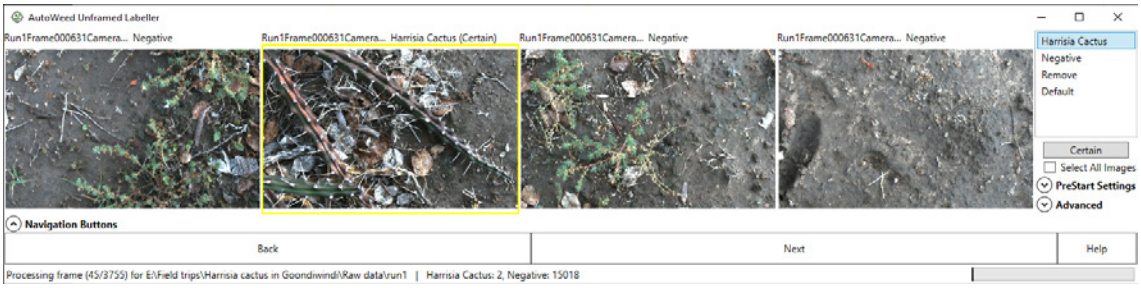


Figure 5.19: A screenshot of the AutoWeed Labelling software being used here to label Harrisia Cactus.

All dataset labelling for the AutoWeed trials in Section 5.3 was performed using the AutoWeed Labelling software. This software allows a single user to label $2,000 - 3,000$ images per hour.

5.2.4 Positioning System

Advanced Navigation Positioning System

The major lesson learned from the Chapter 2 case study, was that more sophisticated positioning technology should be utilised to achieve the required accuracy of weed spot-spraying. As a result the Advanced Navigation Spatial Dual system (Figure 5.20) was chosen to provide highly accurate and reliable global positioning, heading, and velocity information for the ATV. The Spatial Dual system utilises sensor fusion to incorporate dual RTK-GPS and an inertial navigation system. It also has the option to incorporate wheel speed sensor information from the vehicle and correction information from a stationary RTK-GPS base station to increase the accuracy of the system even further (Table 5.5).

Table 5.5: Specifications for the Advanced Navigation Spatial Dual GPS/IMU system.

Specification	**Value**
Horizontal accuracy (m)	0.8
Vertical accuracy (m)	0.5
Velocity accuracy (m/s)	0.007
Roll/pitch/yaw accuracy (°)	0.07
Maximum output data rate (Hz)	1,000

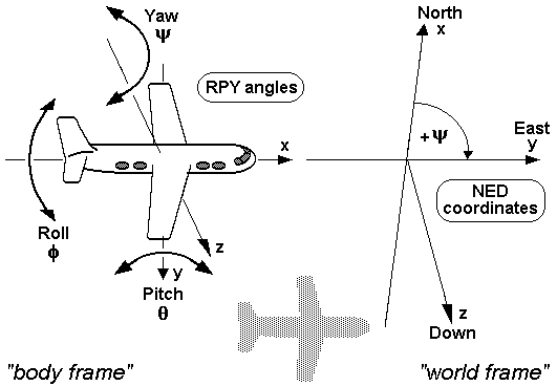
Figure 5.20: Image of the Advanced Navigation Spatial Dual GPS/IMU positioning and navigation system [157].

There are potentially other solutions available for considerably less cost than the Advanced Navigation Spatial Dual, however, the system was purchased for this prototype because it can easily be integrated with ROS, meets the accuracy requirements, and will allow the quickest development time. Future iterations of the prototype may involve the development of a cheap alternative to replace the Advanced Navigation Spatial Dual System using low cost sensors and custom Kalman filter fusion software.

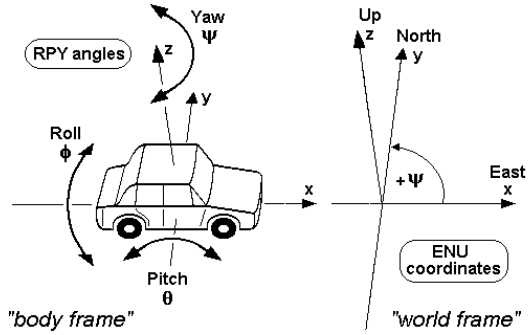
Positioning System

The positioning and navigation software was developed using a base ROS package developed by the Advanced Navigation company for the Spatial Dual system. This ROS package provides the base communication interface with packet encoding and decoding for RS-232 serial communication and comes pre-configured to output some basic packets in ROS message format. This software was heavily modified to output the required messages for navigation, and to conform to the ROS REP 105 navigation reference standard.

The Spatial Dual system uses a North-East-Down (NED) reference frame for navigation. This means the yaw is positive in the north-to-east direction, and the z-axis is positive in the direction of gravity. ROS uses an East-North-Up (ENU) reference frame, which means the yaw is positive from east-to-north and the z-axis is positive against gravity. Conversion is required between reference frames due to the difference between the NED and ENU reference frames (Figure 5.21). Conversion from the NED reference frame to the ENU reference frame was therefore required. To convert from NED to ENU the y-axis, z-axis, and yaw components were inverted. This allowed the ROS visualisation of the system to align with the real-world.



(a) North-East-Down (NED) [158]



(b) East-North-Up (ENU) [159]

Figure 5.21: Comparative illustrations of the ENU and NED reference frames.

After completing the reference frame conversion the latitude and longitude output from the positioning system need to be converted into a 2D Cartesian plane in units of metres using the GeographicLib API [160]. This was performed as it simplifies the tracking system into a single plane and limits potential errors caused by vertical position error. When the positioning system initialises, the first GPS position is taken as the vehicle's origin. The longitudinal position displacement in metres from the origin now represents the x-axis, and latitudinal position displacement in metres from the origin represents the y-axis. The positional displacement in metres between two GPS coordinates is calculated by an ellipsoidal function which can determine the difference accurate to ± 15 nm. Therefore the accuracy of the GPS position is the primary source of error in positioning the vehicle using this conversion.

The orientation of the vehicle is given by the Spatial Dual system, and is used to determine the position of the sprayers and cameras using the ROS *tf2* framework. The *tf2* framework is a set of tools for transforming points in 3D space and for converting systems of points from one reference frame to another. Using this framework the positions of the sprayers and cameras can also be visualised, as shown in Figure 5.22.

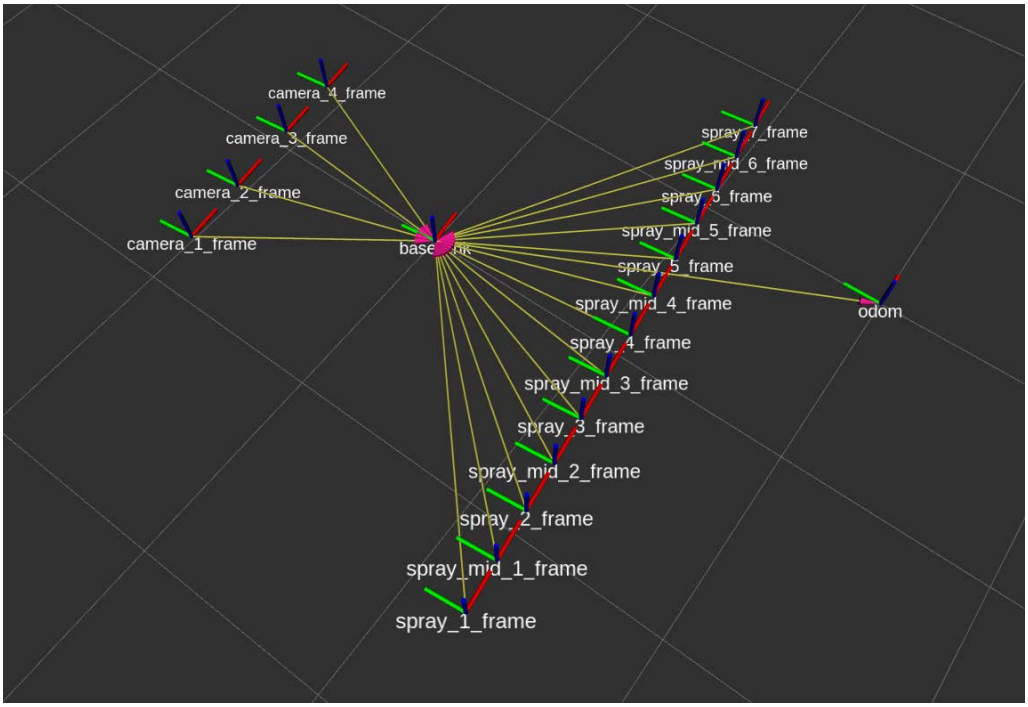


Figure 5.22: ROS visualisations using the *tf2* framework marking the positions of the camera and sprayer reference frames in relation to the IMU/GPS centre point and reference frame origin.

Target Tracking and Processing System

The tracking and spraying system will work almost identically to that of the proof of concept system derived in Chapter 2. Once a target is identified by the detection system, its position will continue to be tracked as the ATV moves. When a weed target passes under a sprayer reference frame, the sprayer will be engaged for a period of time that ensures adequate coverage of the target.

There are three pseudo-simultaneous phases to the tracking system, as indicated below. The tracking program will cycle through each phase continuously during operation, effectively performing each phase in parallel.

Phase 1: Wait on Callbacks

The tracking program waits for messages from the rest of the system. These messages are either updates to the current position, or new targets which have been found. The positioning system will publish odometry messages updating the current position, and the image processing system will publish target messages when new targets are found.

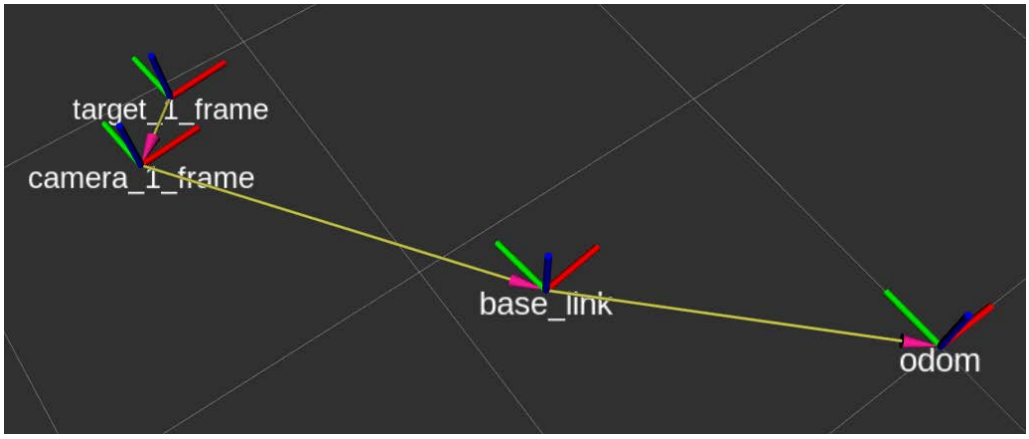


Figure 5.23: ROS visualisation demonstrating how the target position frame is rendered into a real world representation aligned with the *odom* or odometry real world reference frame.

Phase 2: Add New Targets

After receiving a new target, its location in the world frame must be determined based on when it was found and which camera identified it. The ROS tf2 framework stores a history of the vehicle positions for a specified amount of time allowing for previous positions to be queried. Using the time stamp from when the target was captured, its location can be determined from the position of the vehicle in past positions. The location of the target in the world frame is then determined by where it is in the image, from where the camera is located in reference to the position and orientation of the vehicle, and where the vehicle is in the world frame. This transformation chain can be seen in Figure 5.23. The "base_link" frame is the origin point of the vehicle and the "odom" or odometry frame represents the real world origin frame.

Phase 3: Update Tracking

After an update for the position is received, the position of each sprayer is updated using the position and orientation of the vehicle.

The position of each target is then checked to determine if it is about to pass under one of the sprayers. This is achieved by determining whether a target is within a circular radius of each sprayer. In practice, the algorithm calculates the magnitude of a straight line vector passing from the target point location to the sprayer point location. The algorithm then compares this magnitude to a preset threshold. If the distance magnitude is less than the threshold, the corresponding sprayer is activated. The tracking software also registers "mid-spray locations" that exist between each sprayer in order to account

for the situation when a target passes between two sprayers. The software will check these mid points against each target the same way the individual sprayer locations are evaluated. However, if a mid point is triggered, the two nearest sprayers will be engaged.

After determining which sprayers to activate, the system calculates the spray duration based on the speed of the vehicle and the size of the target. This spray duration ensures the target being sprayed is adequately covered by the herbicide regardless of vehicle speed or target size. The spray duration (SD) is calculated from the desired target size (TS) and vehicle velocity (v) using equation 5.7.

$$SD\ [\text{ms}] = \frac{TS\ [\text{m}]}{v\ [\text{m/s}]} * 1000\ [\text{ms/s}] \tag{5.7}$$

For each spray activation, a status message is generated and published indicating which sprayer was activated and for how long. This message is received by the spray controller program for it to electronically activate the sprayers. After a target has been tracked and sprayed it will be removed from the tracking program so that redundant target checks are ignored.

5.2.5 Spraying System

The spraying system designed for this prototype is based upon a modified Fimco ATV boom sprayer (Figure 5.24). The standard solenoids shipped with this unit were replaced with JP Fluid Control fast-acting solenoid valves (Figure 5.24) with an ON and OFF response time of 30 ms. The solenoids were custom fitted to a new sprayer spacing of 35 mm in order to align with the 1.8 m field of view of the detection system. Standard TeeJet nozzle varieties were utilised.

Figure 5.24: Left: Fimco ATV-25-71-QR 25 gallon High-Flo 3.8 GPM pump45 psi boom spraying ATV attachment. Right: JP Fluid Control 24 V 1/4" thread, 1mm orifice 30 ms solenoid.

A custom control system was developed to interface with the solenoids to be able to activate them at their rated response time. The modifications and custom board were chosen and developed by Brendan Calvert. The developed control circuit board is pictured in Figure 5.25. The interface for the sprayer control board was designed to use the standard RS-232 communication using a custom protocol for individually activating an array of the sprayers for a specified time. The firmware used on the custom control system was a modification of the firmware developed by Jake Wood for the case study of Chapter 2.

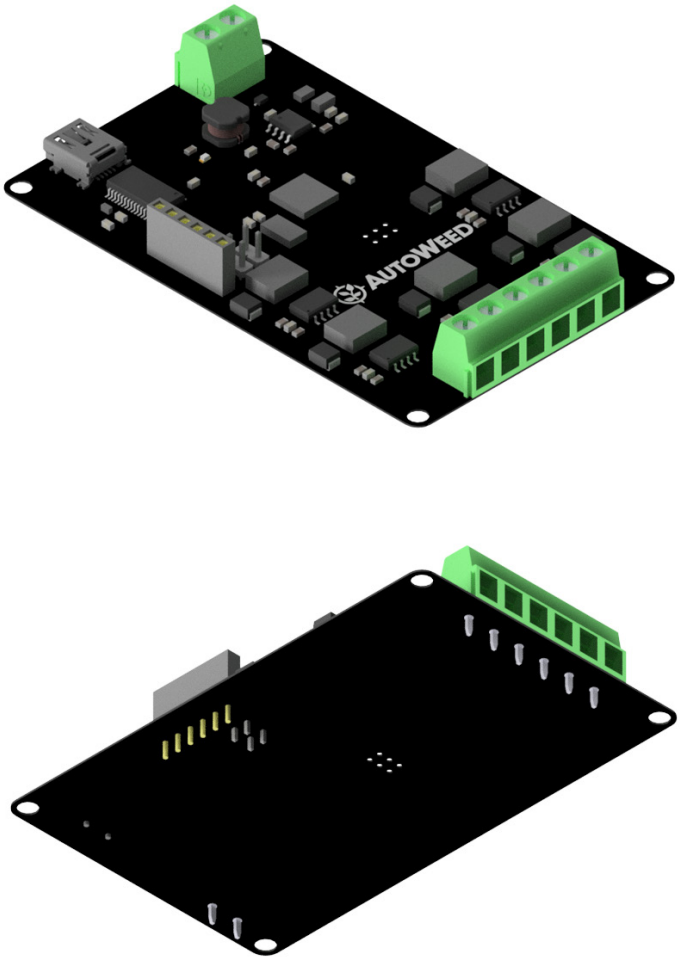


Figure 5.25: A 3D orthographic rendering of the custom designed sprayer control board.

After preliminary testing, it was apparent that the activation of the solenoids can be difficult to observe during broad daylight. To combat this issue, LED lights were installed above each sprayer and wired in parallel to the solenoid. This allowed demonstrated sprays to be more visible for the final spraying system (Figure 5.26).

Figure 5.26: Developed AutoWeed prototype sprayer system with 35 cm spaced solenoids and LED indicators above each nozzle.

5.2.6 Positioning Trials

A set of experiments were conducted to evaluate the positioning accuracy of the system. These trials utilise a simple orange blob detection algorithm to target orange cones for spraying. The three stages of positioning trials include: i) straight line, ii) straight line with multiple targets and iii) complex path with multiple targets. For all trials, a collection of data was recorded including: the vehicle path, ground truth targets, detected targets and spray activations.

For the straight line positioning tests, targets were placed on the ground directly in front of the prototype. The prototype then drove over the targets to determine the accuracy of the spray. Three runs of straight line tests were performed.

For the straight line trial with multiple targets, multiple targets were spread along the horizontal field of view of the cameras while the vehicle traveled in a straight line to perform detection and spraying. A single run of this trial was performed. This trial will indicate successful localisation of the all camera and sprayer positions.

For the final complex path trial, the AutoWeed prototype was taken through a more complex path to vigorously assess the positioning accuracy while spraying multiple targets. Along this path, several targets were placed to assess the spray accuracy.

5.2.7 Detection Trial

With successful targeting, positioning and spraying, the performance of the weed detection models could be evaluated based upon whether or not weeds were being sprayed in the field. The goals of this detection trial were to:

- Test if a DeepWeeds trained model performs well on a never-before-seen trial site, or if site-specific datasets will be required for effective performance.
- Establish a work flow for deploying the detection models in the field.
- Evaluate the accuracy of our best performing ResNet-50 model for weed species detection in the field.

Trial Site

Chinee apple was chosen to be the ideal weed species for this trial because of its prevalence in the local region of Townsville (Figure 5.27) and its availability in the DeepWeeds dataset. A Chinee apple trial site was identified for testing in the suburb of Condon. The trial site, a grassy residential paddock, represented an ideal test case for the detection system with several clearly identifiable Chinee apple targets against a homogeneous background of plant life that mostly contained grass.

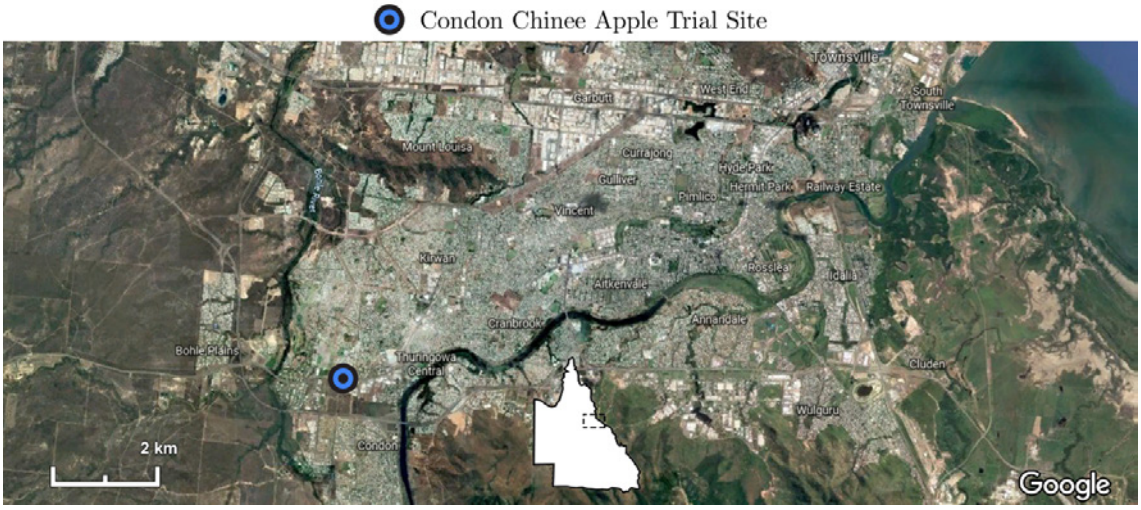


Figure 5.27: Townsville based trial site for testing the detection of Chinee apple using the AutoWeed spot spraying prototype (Data and Images from CNES / Airbus Data SIO, NOAA, U.S. Navy, NGA, GEBCO GBRMPA Maxar Technologies).

Model Preparation

Two binary classification models targeting Chinee apple were prepared for this investigation: 1) a DeepWeeds model and 2) a site-specific model. The dataset prepared for the first model was a partition of the DeepWeeds dataset, introduced in Chapter 3, that includes all Chinee apple images and negative images with all other classes removed. Meanwhile, the site-specific Chinee apple dataset was collected from the selected trial site. A consistent distribution of images in each class was maintained for both datasets (Table 5.6).

Table 5.6: Distribution of the two datasets used for the Chinee apple detection trials.

Dataset	Chinee apple	Negatives	Total
DeepWeeds	1,125	9,106	10,231
Site-Specific	1,179	9,760	10,939

The datasets were each partitioned into 60% training, 20% validation and 20% testing and then an ImageNet pretrained ResNet-50 architecture was trained on both datasets with: a batch size of 32 images, a maximum of 100 epochs, early stopping when the validation accuracy does not improve after 32 epochs, halving the learning rate if validation accuracy does not improve after 16 epochs and various augmentations (including colour, rotation, scale and shift). The DeepWeeds and site-specific models reached a plateau of accuracy at 98.4% and 97.8% after 40 and 70 epochs, respectively (Figure 5.28).

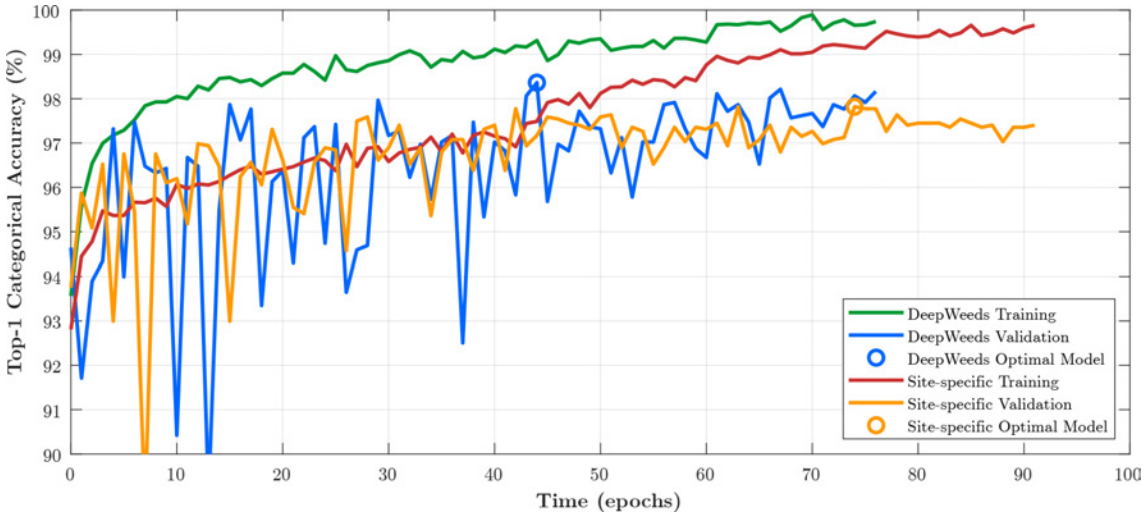


Figure 5.28: Training and validation accuracy of the DeepWeeds and site-specific ResNet-50 models during training.

The model with the highest validation accuracy during training was saved as the optimal model to be used for the detection trial. These optimal models for both the DeepWeeds and site-specific datasets were evaluated against the test partition in their respective datasets to assess their performance (Figure 5.29).

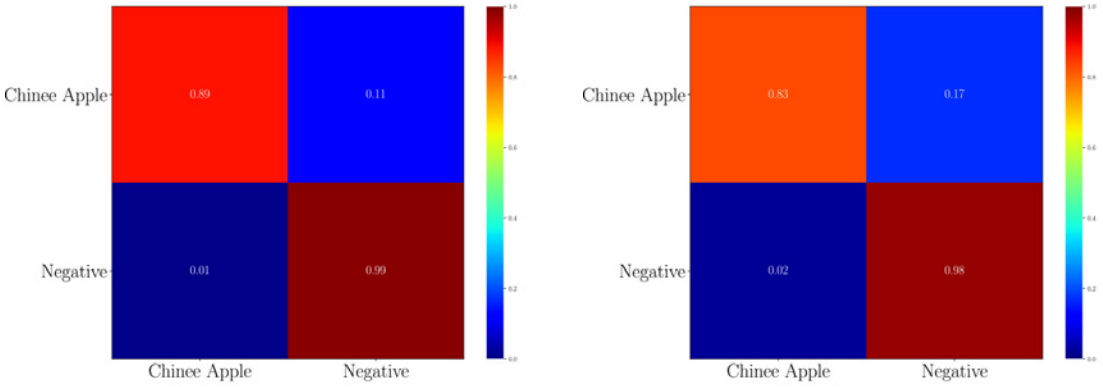


Figure 5.29: Confusion matrix showing the recall rate accuracy for the (a) DeepWeeds and (b) site-specific ResNet-50 models on their respective test subsets.

The confusion matrices for both models (Figure 5.29) reveals that the DeepWeeds model achieved 89% recall of Chinee apple and 99% recall of negatives on its test set; while the site-specific model achieved 83% recall of Chinee apple and 98% recall of negatives on its test set. Note this is not a direct comparison because each model was only evaluated on its own specific test dataset. It can be noted that both models classify their test set quite well, which is to be expected given each dataset comprises over 10,000 images.

Both models were then deployed in the field at the same trial site. This would provide an indication of whether a Chinee apple dataset from an entirely different location would work well on a new trial site, or if site-specific datasets were superior.

5.2.8 AutoWeed Trials

The complete AutoWeed spot-spraying system was evaluated in five field trials conducted across eastern Australia between March and August 2019 (Figure 5.30). The trials were conducted on various land types, with different weed species and in different climates. The trials in Spring Ridge and Arcturus included live demonstrations to local landholders at the invitation of North West Local Land Services and the Department of Agriculture and Fisheries. The trials therefore served as an opportunity to evaluate the performance of the system and to promote the system to local landholders in view of future commercialisation of the technology.

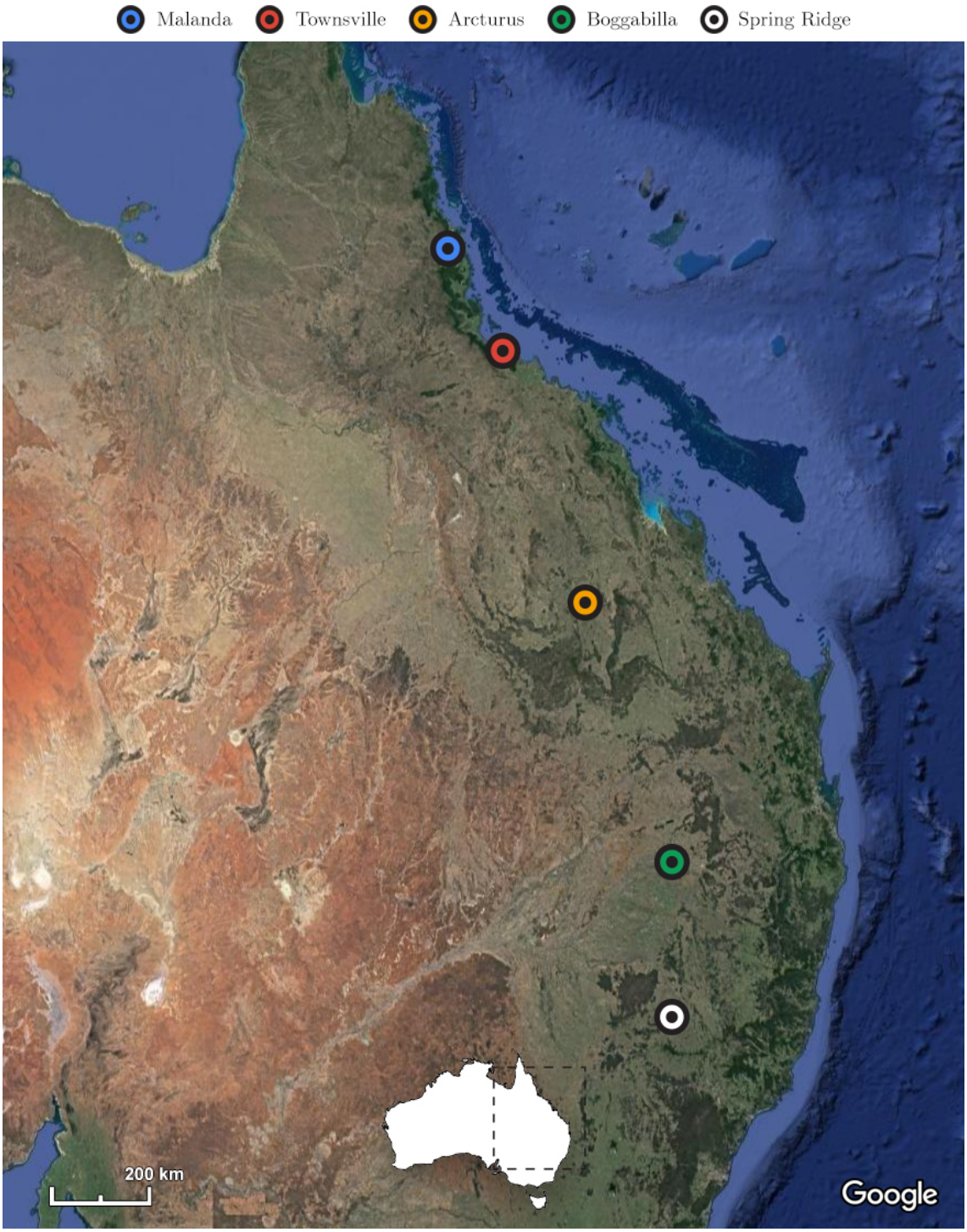


Figure 5.30: The geographical distribution of *AutoWeed* trial sites along the eastern coast of Australia (Imagery ©2019 TerraMetrics, Map data ©2019 Google).

Chinee Apple in Townsville

On Tuesday 5th March 2019, the AutoWeed prototype was utilised to treat a roadside infestation of Chinee apple in the suburb of Rasmussen with highly diverse plant life. Chinee apple is a large shrub or small tree that causes dense infestations and impenetrable thickets that seriously hamper stock management and reduce pasture production [161]. The broadleaf weed is readily identifiable by its leaf shape, colour and thorny base, as shown in Figure 5.31. With successful detection of this weed already shown in section 5.3.2, this trial of the full AutoWeed system aimed to show the efficacy of the system for weed treatment.

Figure 5.31: Left: An up close photograph of a Chinee apple shrub. Right: Chinee apple in field during the Townsville trial being passed over by the AutoWeed prototype.

Navua Sedge in Malanda

On Friday 24th May 2019, the AutoWeed prototype was utilised to treat a *Cyperus aromaticus* (Navua sedge) infested pasture paddock on a cattle station in Malanda, QLD. Navua sedge is a particularly devastating weed in North Queensland (Figure 5.32). It provides little feed value for cattle, quickly overtakes palatable pasture species and, consequently, reduces the carrying capacity of farming land for local landholders [163]. The weed is similar to pasture plant species but in sunlight is distinguishable by its yellow-tinted colouring. This difficulty in detection is echoed by local landholders who expend effort and time to regularly spray their paddocks by hand only to find they themselves have missed targets.

Figure 5.32: Photographs of Navua sedge present within dense pasture. The grass-like weed is very difficult to identify amongst grass.

Harrisia Cactus in Boggabilla

On Friday 7th June 2019, the AutoWeed prototype performed a spot-spraying trial on *Harrisia spp.* (Harrisia cactus) infested pasture within a paddock on Warakirri Cropping's Willaroo farm in Boggabilla. Harrisia cactus is a perennial weed that forms dense infestations to reduce pastures to levels unsuitable for stock [164]. It is identified by thick, spiky, green stems with red, round fruits (Figure 5.33). Harrisia cactus is a significant rangeland weed in Queensland and northern New South Wales. It competes against native and naturalised pasture species and can inhibit stock access and pose a threat of injury to wildlife.

Figure 5.33: Harrisia catus pictured by the roadside (left) and in pasture (right).

Turnip Weed in Spring Ridge

On Monday 29th July 2019, the AutoWeed prototype performed a green-on-green spot spraying crop trial of a turnip weed infested Oats crop at University of Sydney's Nowley Farm near Spring Ridge. The target weed was *Sisymbrium thellungii*, an African turnip weed, that affects many Australian crops. AutoWeed was invited to perform the trial by North West Local Land Services, who hoped to leverage the detection technology for a crop application [166]. This trial constitutes the first cropping trial of the detection technology. The uniformity of the crop and irregularity of the weed made for a simpler learning problem (Figure 5.34).

Figure 5.34: Turnip weed in an oats crop on the University of Sydney's Nowley Farm, 20 km northwest of Spring Ridge .

Sowthistle Weed in Arcturus

On Wednesday 14th August 2019, the AutoWeed prototype performed a green-on-green spot spraying crop trial on a wheat crop infested with *Sonchus oleraceus*, a common sowthistle weed, and unwanted chickpea plants on a large broadacre farm in Arcturus, 100 km southeast of Emerald. The wheat crop in question is typically blanket sprayed with a selective herbicide. Green-on-green spot-spraying of the unwanted sowthistle and chickpea (Figure 5.35) would allow for significant chemical savings.

Figure 5.35: Left: A close-up photograph of sowthistle in wheat. Middle: A broad view of the wheat crop where the size of the weeds is visible. Right: A close-up view of unwanted chickpea plants growing among the crop.

Dataset Collection

Prior to the trials, the AutoWeed prototype was used to collect data from a designated trial paddock in each farming location. Images were collected while continuously traversing the paddocks in random path. The total number of images collected across all sites was 103,857 (Table 5.7), far exceeding the image count of the DeepWeeds dataset (17,509). The custom developed AutoWeed labelling software was then used to annotate the weeds present in every image. For pasture datasets, approximately 2,000 images per hour could be annotated. This labelling speed was doubled for crop datasets with approximately 4,000 images annotated per hour. This significant difference in labelling simplicity is due to the homogeneity of the crop environment compared to the more variable plant life present in pastoral lands.

Table 5.7: Distribution of the image datasets collected for the five AutoWeed spot-spraying trials.

Location	**Target weed(s)**	**Negatives**	**Positives**	**−:+**	**Total**
Townsville	Chinee apple	9,059	2,476	3.66:1	11,535
Malanda	Navua sedge	13,727	3,985	3.44:1	17,712
Boggabilla	Harrisia cactus	16,887	2,630	6.42:1	19,517
Spring Ridge	Turnip weed	20,703	3,495	5.92:1	24,198
Arcturus	Sowthistle, chickpea	25,782	5,113	5.04:1	30,895
Total		86,158	17,699	4.87:1	103,857

Randomly traversing the paddock ensured the distribution of target (positive) images to non-target (negative) images matches the trial environment. On average, a ratio of five negative images to one positive image was recorded across the trial sites. The total number of images at each site increases over time due to the ease and familiarity with the dataset collection process from trial to trial.

It was decided to collect twice as many images for the Arcturus trial because there were two equally dominant weed targets: sowthistle and chickpea. Labelling was performed separating both weed species to allow multiclass classification rather than grouping the weed species together for binary classification. The positive count for this trial consists of 2,681 sowthistle weed images and 2,432 chickpea images. This is the only multiclass classification trial of the AutoWeed methodology.

Model Preparation

Following the methodology of Chapter 4, an ImageNet pretrained ResNet-50 architecture was trained on each dataset with: a batch size of 32 images, a maximum of 100 epochs, early stopping when the validation accuracy does not improve after 32 epochs, halving the learning rate if validation accuracy does not improve after 16 epochs and various augmentations (including colour, rotation, scale and shift).

The Chinee apple classifier for Townsville reached a plateau of validation accuracy at 97.4% after 91 epochs (Figure 5.36). The resulting optimised model was then evaluated against the held-out test set achieving 97% detection of negatives and 81% detection of Chinee apple (Figure 5.36). This strong detection result with only 11,535 images can be attributed to the distinctive features of Chinee apple.

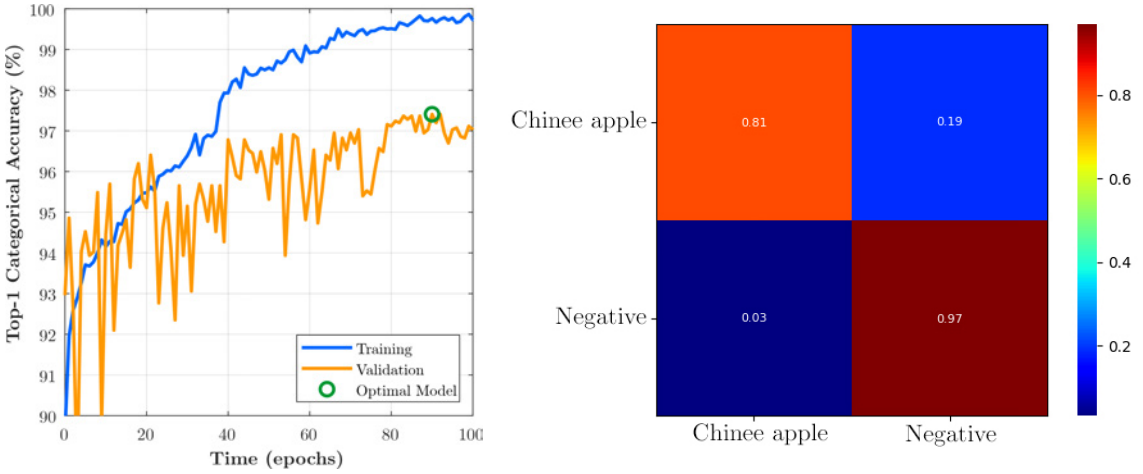


Figure 5.36: Left: The training and validation accuracy throughout successive epochs of the training process for the Chinee apple model. Right: The confusion matrix of the optimal model evaluated on the Chinee apple test subset.

The Navua sedge model for Malanda reached a plateau of validation accuracy at 93.1% after 45 epochs (Figure 5.37). The resulting optimised model was then evaluated against the held-out test set achieving 98% detection of negatives and 77% detection of Navua sedge (Figure 5.37). The comparatively less accurate Navua sedge classification shown here on the test set can be attributed to the weed's similar visible features to background pasture grass species. Of all the weeds investigated in this thesis, Navua sedge presents the most difficult vision-based identification challenge. A greater number of training images will likely be required to improve detection accuracy.

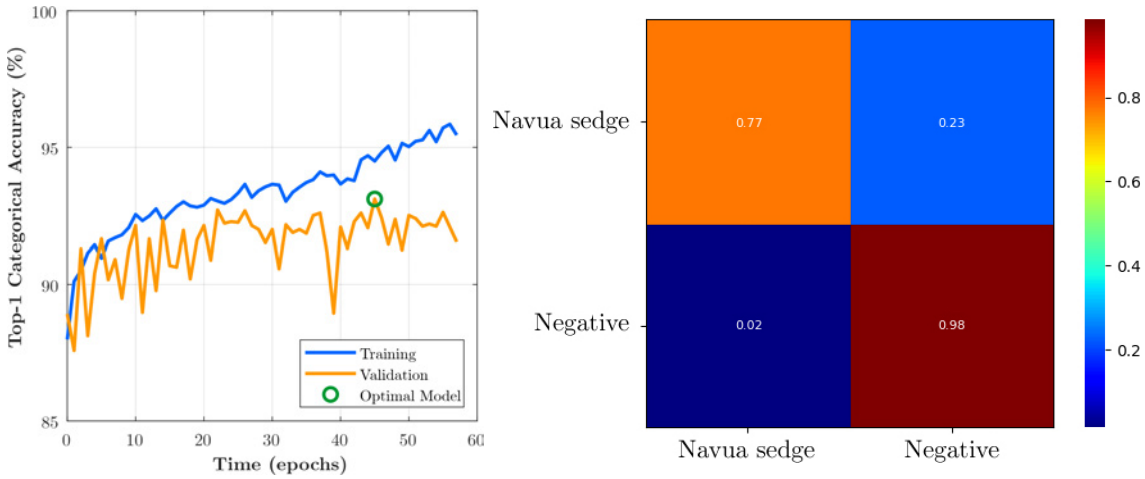


Figure 5.37: Left: The training and validation accuracy throughout successive epochs of the training process for the Navua sedge model. Right: The confusion matrix of the optimal model evaluated on the test Navua sedge subset.

The Harrisia cactus classifier for Boggabilla reached a plateau of validation accuracy at 96.6% after 34 epochs (Figure 5.38). The resulting optimised model was then evaluated against the held-out test set achieving 100% detection of negatives and 82% detection of Harrisia cactus (Figure 5.38). Achieving perfect classification of the negative class speaks to the uniquely and prominent visible features of the cactus compared to its background. The 18% non-detections are attributed to variance in the Harrisia class due to health of the plant. It is expected to see even higher recall of Harrisia cactus with larger training datasets.

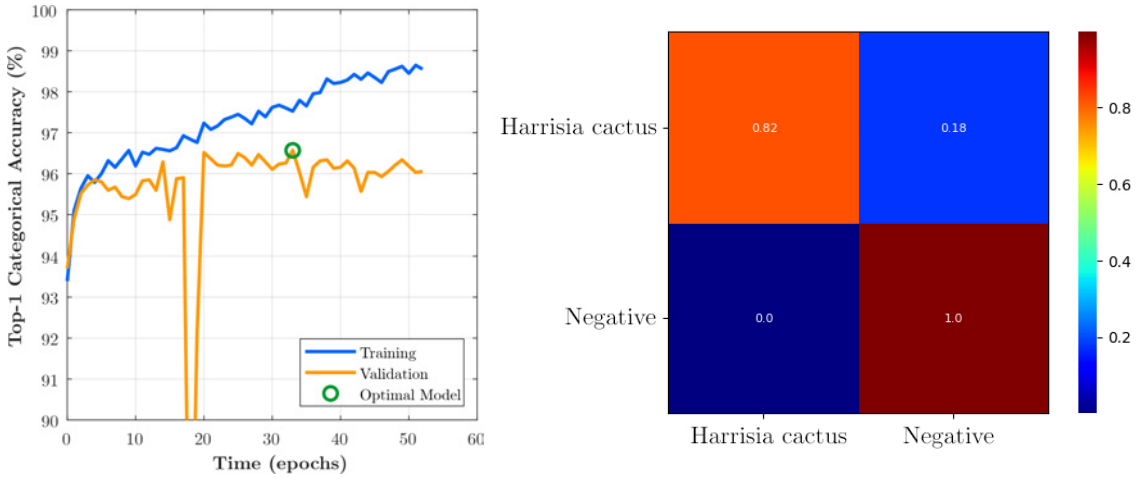


Figure 5.38: Left: The training and validation accuracy throughout successive epochs of the training process for the Harrisia cactus model. Right: The confusion matrix of the optimal model evaluated on the Harrisia cactus test subset.

The turnip weed classifier for Spring Ridge reached a plateau of validation accuracy at 95.5% after 63 epochs (Figure 5.39). The resulting optimised model was then evaluated against the held-out test set achieving 99% detection of the crop and 80% detection of turnip weed (Figure 5.39). The model correctly rejects the crop with near perfect accuracy. However, the lower than expected 80% classification of turnip weed is assumed to be due to the presence of at least three other weed varieties in the crop which were labelled as negatives. Better inter-species classification could be achieved by classifying each negative plant species separately.

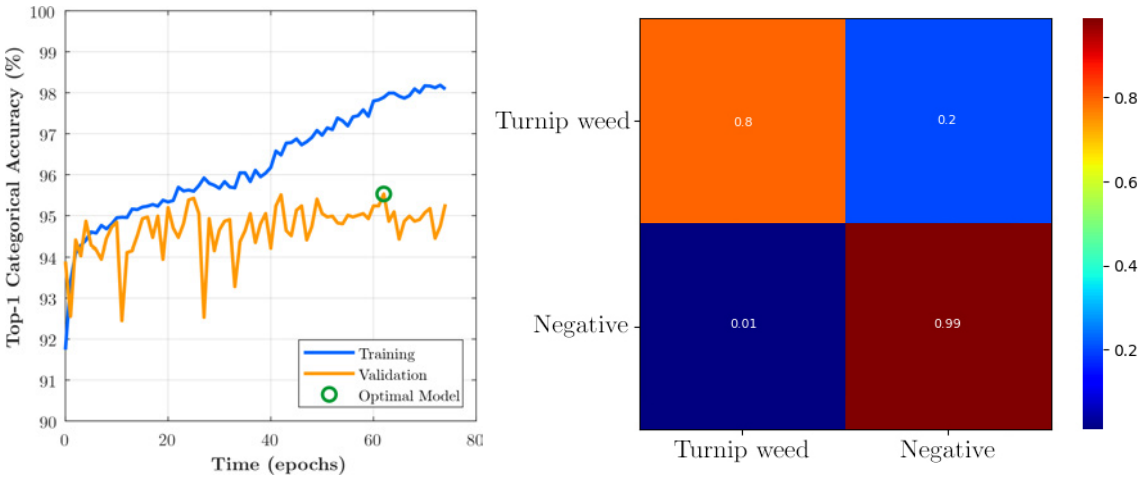


Figure 5.39: Left: The training and validation accuracy throughout successive epochs of the training process for the turnip weed model. Right: The confusion matrix of the optimal model evaluated on the turnip weed test subset.

The multiclass classifier for Arcturus reached a plateau of validation accuracy at 96.4% after 58 epochs (Figure 5.40). The resulting optimised model was then evaluated against the held-out test set achieving 99% crop detection and 92% detection of both sowthistle and chickpea (Figure 5.40). This is the strongest classification result of all trials conducted in this thesis. We assume this strong performance is owed to the size of the training dataset being the largest of all trials.

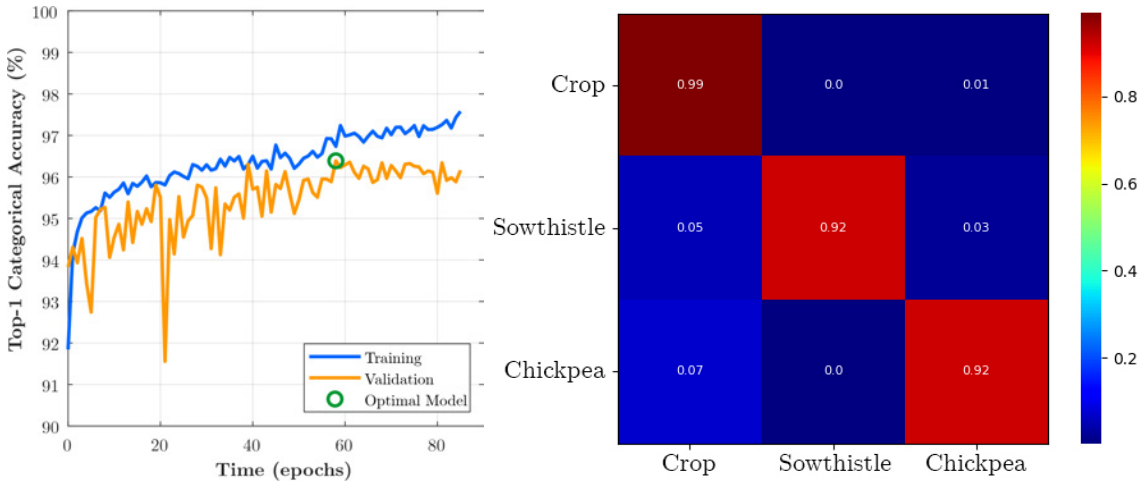


Figure 5.40: Left: The training and validation accuracy throughout successive epochs of the training process for the Arcturus model. Right: The confusion matrix of the optimal model evaluated on the Arcturus test subset.

Deployments

The resulting classification models were then loaded onto the AutoWeed prototype in order to perform spot-spraying trials on designated trial zones in each location. During the trials, data was logged to allow the determination of sprayed and unsprayed trial zone area. The accuracy of the prototype and its efficacy for reducing herbicide usage could then be evaluated.

5.3 Experimental Results

5.3.1 Positioning Trials

Straight Line

Overall, the system detected and sprayed every target in each of the three trials with 100% accuracy (Figure 5.41). This is a very strong result and suggests the software and hardware developed for our positioning system meets the accuracy requirements for straight line detection and spraying.

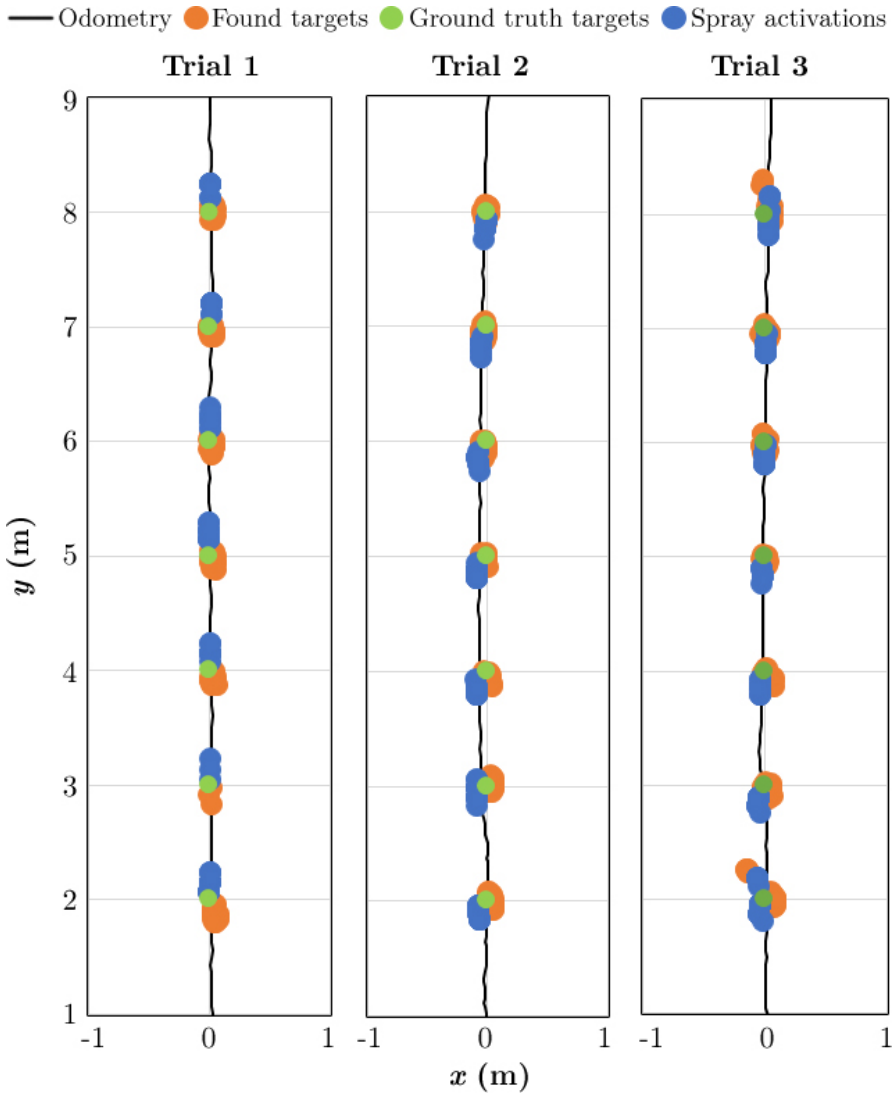


Figure 5.41: A compilation of data from the three straight line trials. Every target for each trial was successfully detected and sprayed.

Straight Line Multiple Targets

The results of this trial showed that three targets positioned 1 m out from the central path were not detected or sprayed (Figure 5.42). After careful review of the images resulting in these missed detections, it was determined that these targets were outside the field of view of the cameras and therefore could not be detected. Meanwhile, all other ground truth targets were detected and successfully sprayed with accuracy by the system. This test revealed that the positioning algorithm to deduce target positions from different cameras and trigger spraying from various sprayers was functional.

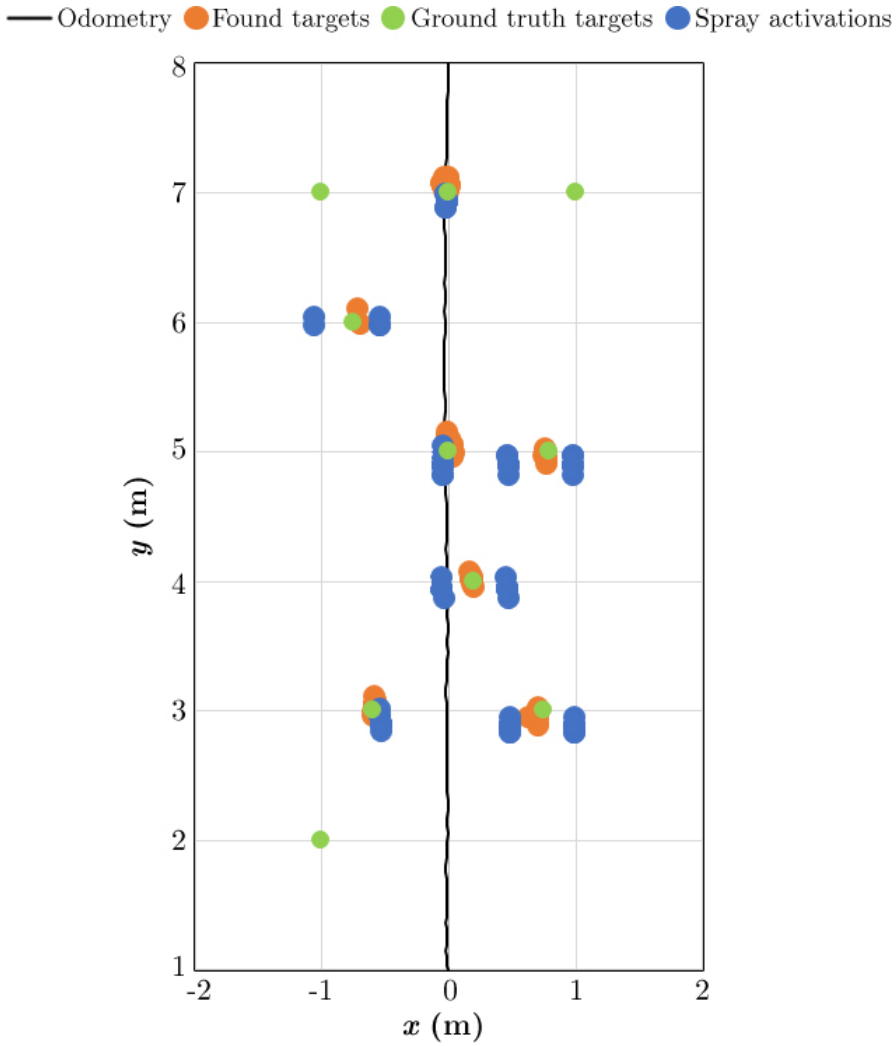


Figure 5.42: An illustration of the successful straight line, multiple target trial where all identified targets were successfully sprayed.

Complex Path Multiple Targets

The results from this positioning trial indicate that every target along the complex path was successfully detected, tracked, and sprayed (Figure 5.43). There are, however, two false positive detections where the crude orange blob detection software found two false targets. These targets were found to be caused by strong sunlight reflection off the bitumen. Nevertheless, the spraying and positioning system, which is under investigation here, successfully completed the tracking and spraying of the targets.

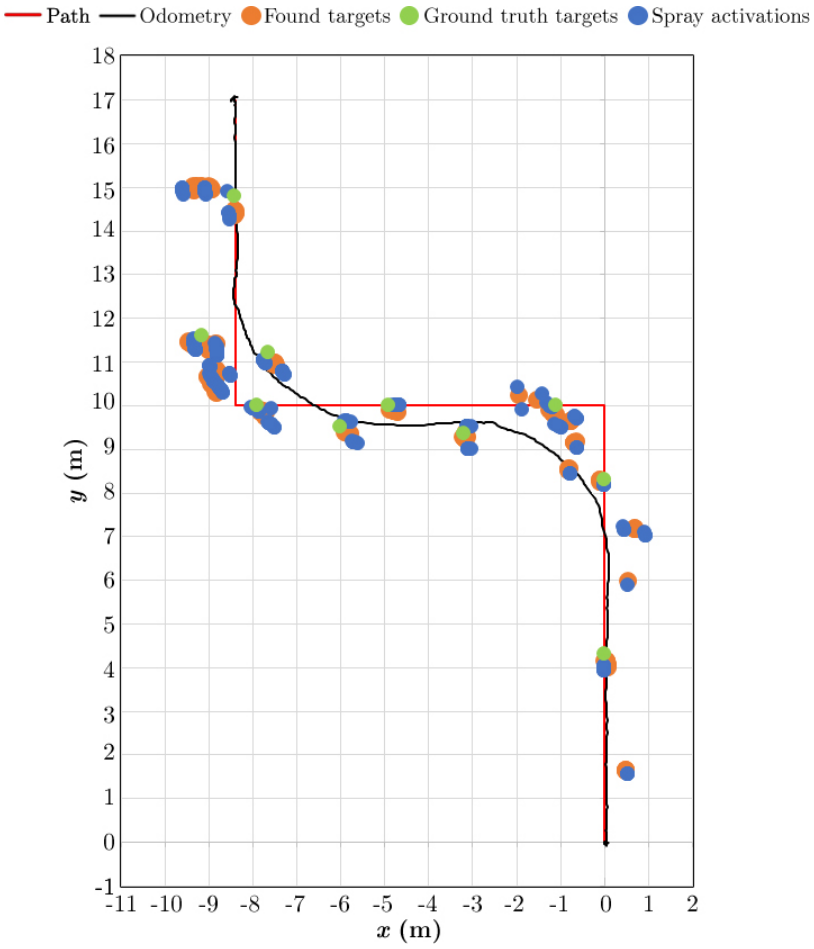
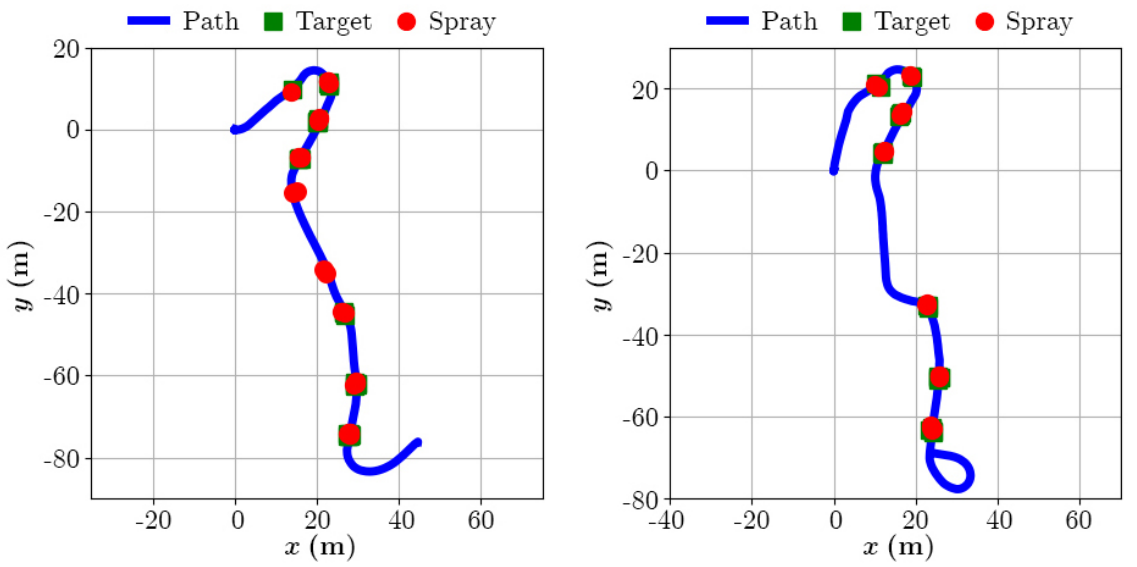


Figure 5.43: An illustration of the more complex path trial of the AutoWeed positioning and spraying system where all targets were successfully detected and sprayed.

5.3.2 Detection Trial

This detection trial evaluated two separate Chinee apple classifiers on the same paddock in order to determine if general weed species datasets from several locations could match the performance of a site-specific dataset at the same trial location. The trial site for evaluation contained seven ground truth Chinee apple targets amongst the grass background. Two separate trials were conducted at the site using both the DeepWeeds trained model and the site-specific trained model. Throughout the trials, the motion path of the vehicle, ground truth target locations and recorded spray locations were recorded and mapped.

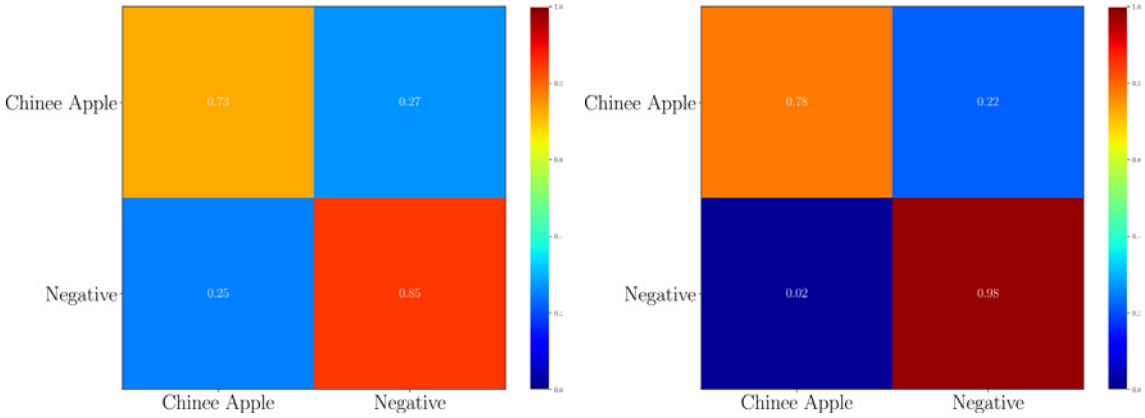
Both models correctly sprayed all seven targets (Figure 5.44). However, the DeepWeeds model sprayed slightly more false positives than the site-specific model, which had no false positive sprays.



(a) (b)

Figure 5.44: Visualisation of the motion path, ground truth target locations and recorded spray locations of trials conducted on the Chinee apple trial site using (a) the DeepWeeds trained model and (b) a site-specific trained model.

Images collected by the AutoWeed platform during both trials were then analysed to assess the prototype's recorded classifications. In total, there were 7,584 images and machine classifications for the trial of the DeepWeeds model and 7,200 for the trial of the site-specific model. Every single image and machine classification was reviewed by a human expert in order to assess the machine's performance.



(a) (b)

Figure 5.45: Confusion matrices after evaluating the (a) DeepWeeds and (b) site-specific models on the in-field dataset.

The site-specific model outperformed the DeepWeeds model in the field by correctly recalling Chinee apple targets with 78% accuracy compared to 73% accuracy (Figure 5.45). Furthermore, the DeepWeeds model incorrectly classified 25% of the negative images as Chinee apple (i.e. false positives). This is the reason for the four false spray activations that occurred during the trial (Figure 5.44). By comparison, the site-specific model predicted the negative class much better with only 2% false positives images and zero false positive sprays (Figure 5.45). This is to be expected, since the site-specific model has been trained on images from the same paddock as the trial location; while the DeepWeeds model was trained on Chinee apple from an entirely different location.

A collection of false positive image examples from the DeepWeeds field trial were analysed to better understand how the models failed (Figure 5.46). The majority of false positives are due to tall strands of grass approaching the lens of the camera and changes in dynamic lighting. The false positives however, all have confidence scores below 95%. This gives hope that the false detections can be mitigated by setting a higher decision threshold for spray activations.

The site-specific model also has 5% less false negatives (i.e. missed detections) than the DeepWeeds model. It is important to note that these are false negative *images* and not false negative *targets*. Because the detection system is operating at over 10 fps, each target will be imaged multiple times and the detection system will have multiple chances to detect a target. Despite both models having a false negative rate larger than 20%, they were both able to detect and spray all targets.

A collection of false negative example images from the DeepWeeds field trial were analysed to better understand how the models missed Chinee apple targets (Figure 5.47). It can be seen that the false negative classifications mostly occur when the weed is almost completely out of the frame, or it is heavily occluded by dynamic lighting and other plant life. It is assumed that the frames immediately in front or behind these false negatives, contained a better view of the weed and therefore resulted in positive detections.

Summary

It is clear from this trial that the site-specific model outperforms the more varied and non-specific DeepWeeds dataset. It was then decided for future trials to begin with entirely site-specific dataset collection. Following this successful validation of the detection system, and earlier validations of the spraying and positioning system, the AutoWeed prototype was then subjected to field evaluation.

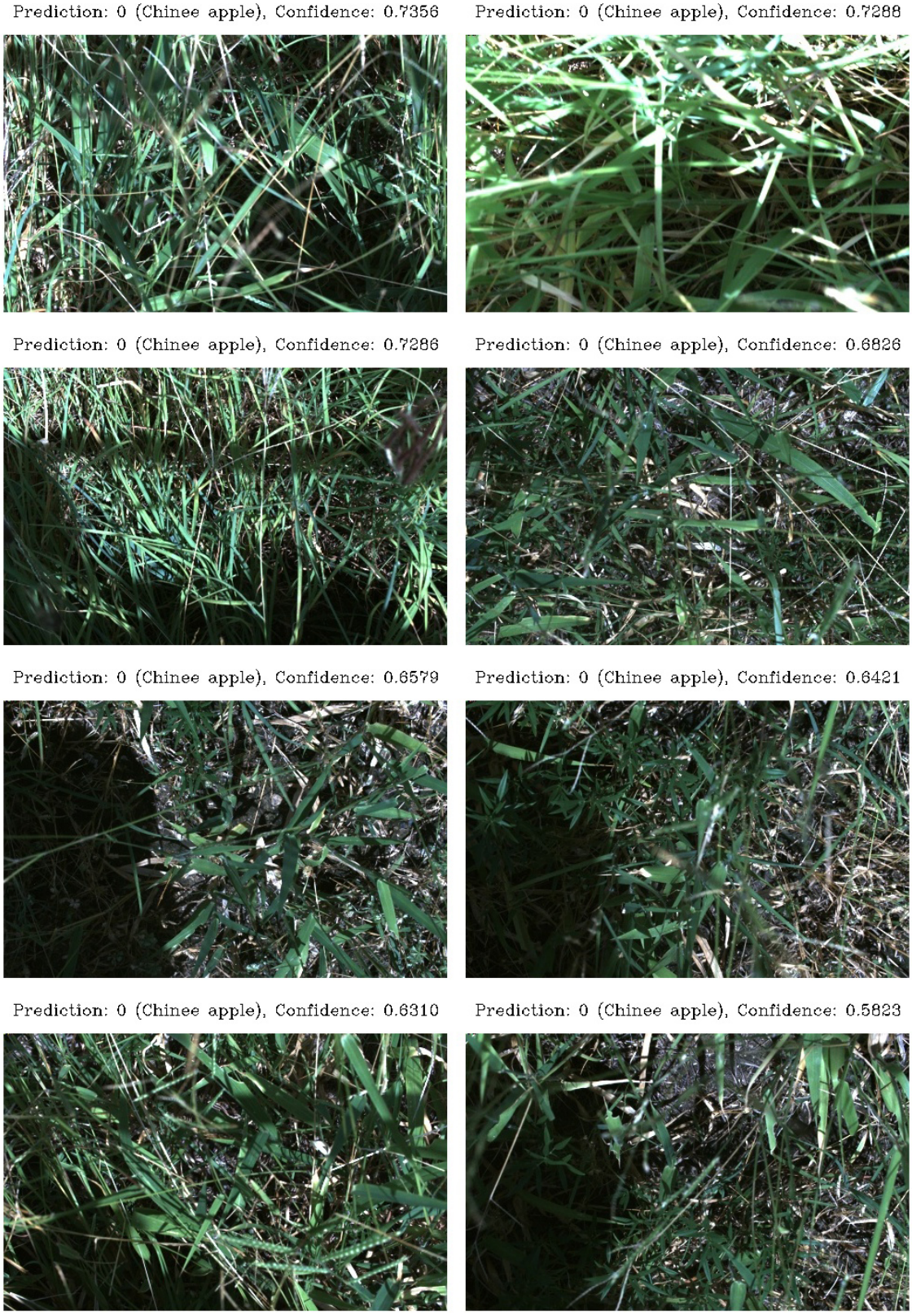


Figure 5.46: A collection of the highest confidence false positive classifications resulting from deploying the DeepWeeds model at the Chinee apple trial site.

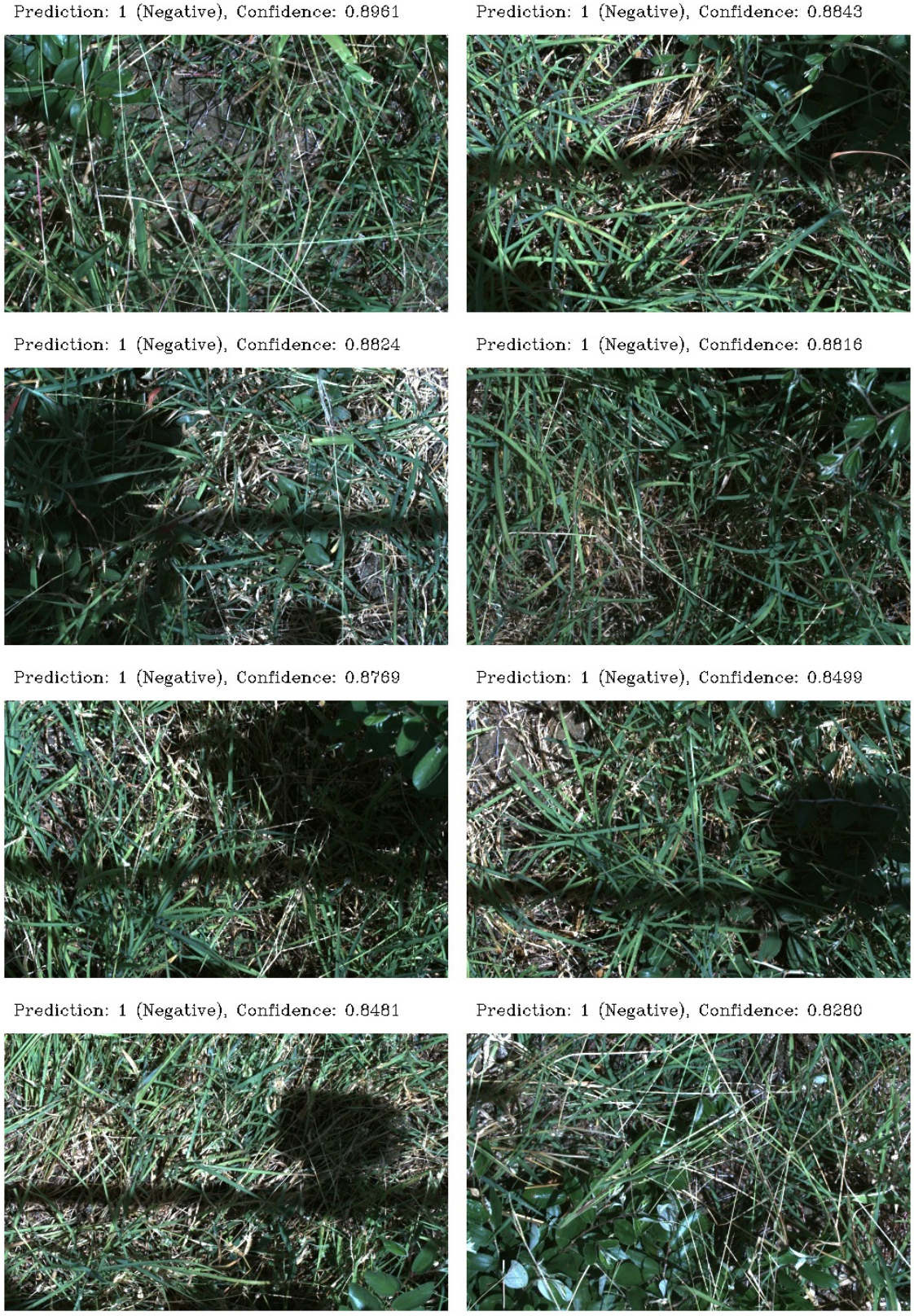


Figure 5.47: A collection of the highest confidence false negative classifications resulting from deploying the DeepWeeds model at the Chinee apple trial site.

5.3.3 AutoWeed Trials

Results

The models for each site-specific dataset were loaded onto the AutoWeed prototype to treat small trial zones in each location. Various data was logged during the trials including the vehicle's traversed path, identified target locations and triggered spray locations. This allowed the determination of sprayed and unsprayed trial zone area for each site (Table 5.8).

Table 5.8: Weed treatment statistics recorded by the AutoWeed spot-spraying prototype across the five trials.

Location **Target weed(s)**	Townsville Chinee apple	Malanda Navua sedge	Boggabilla Harrisia cactus	Spring Ridge Turnip weed	Arcturus Sowthistle, chickpea
Area covered (m^2)	440	101	303	2267	739
Area sprayed (m^2)	30	11	42	106	30
Area sprayed (%)	7	11	14	5	4
Herbicide reduction (%)	93	89	86	95	96
Average speed (km/h)	5.9	6.0	6.0	10	7.0

Each trial location exhibited low weed density, with majority of the trial area consisting of non-target pasture or crop. Robotic spot-spraying offers the most herbicide reduction in this situation in comparison to traditional blanket spraying, which covers the entire paddock with a selective herbicide. The two cropping trials saw the highest herbicide reduction. This is likely due to these crops having lower weed density due to effective non-selective herbicide treatment at the chosen trial sites. The average speed of the prototype was found to be higher during the cropping trials, due to the relatively flat terrain compared to the rangeland environments.

Chinee Apple in Townsville

Current control of roadside and rangeland Chinee apple regrowth is performed by regularly clearing the paddocks with machinery. When there is only a fraction of the paddock that needs to be controlled, like the 7% in this trial, green-on-green spot-spraying offers a tremendous financial incentive. This prototype is perfectly suited for spot-spraying of Chinee apple regrowth, particularly on roadsides across North Queensland.

Figure 5.48: Left: Alex Olsen and Brendan Calvert operating the AutoWeed prototype in Townsville. Right: Location data for the Townsville trial.

Navua Sedge in Malanda

The 89% herbicide reduction on Navua sedge in Malanda is a strong result given the inherent difficulty of detecting Navua sedge in pasture. It is the first successful trial on a sedge with a grassy background. Efforts have been made to develop selective herbicides that target Navua sedge and leave the pasture untouched [162]. Currently, Sempra is the only such registered herbicide, and it requires regular application with ten week withholding periods to be effective [163]. With AutoWeed, we introduce a more favourable selective treatment for Navua sedge. Because of our high accuracy in minimising false positives, the farmer can afford to use non-selective herbicides, such as glyphosate, to ensure the death of weeds without significantly affecting their pasture. However, this approach is only effective when there are low weed densities of the targeted weed.

Figure 5.49: Left: Alex Olsen operating the AutoWeed spot-spraying prototype on a cattle station 10 km southeast of Malanda. Right: Location data for the Malanda trial.

Harrisia Cactus in Boggabilla

This trial has proven the efficacy of the AutoWeed prototype to detect and spray Harrisia cactus in pasture situations without spraying surrounding desirable vegetation. While the North West Regional Strategic Weed Management Plan 2017-2022 has set an objective to contain Harrisia cactus within its present bounds [165], high control costs inhibit the management of this weed. The automatic weed detection and control technology shown here will significantly reduce the cost of management compared to traditional hand gunning and blanket spraying of infected pasture. In turn, this technology may play a key role in stopping the inexorable spread of Harrisia cactus in southern Queensland and northern New South Wales.

Figure 5.50: Left: Peter Ridd operating the AutoWeed prototype at Willaroo farm, Boggabilla, NSW. Left: Location data for the Boggabilla trial.

Turnip Weed in Spring Ridge

This represented the first AutoWeed trial in a cropping situation. This oats paddock would usually be blanket sprayed with a selective herbicide that targets broadleaf variety plants and weeds, like turnip weed, without affecting the thin leaf crop. With spot-spraying technology able to detect and spray the 5% of weeds present in the paddock, fantastic savings of herbicide usage can be achieved. This trial was also the most expansive of the AutoWeed trials in terms of area sprayed. Because of the flat and homogeneous cropping terrain the vehicle could be operated at much higher speeds than previous trials. At the higher speed, there was no negative impact on detection accuracy. This is important for cropping application because the farmer would want to traverse the crop as quickly as possible and usually operates at around 20 km/hr.

Figure 5.51: Left: Alex Olsen operating the AutoWeed prototype on Nowley Farm, Spring Ridge, NSW. Right: The location data for the Spring Ridge trial.

Sowthistle Weed in Arcturus

This was the second successful AutoWeed trial in a cropping application. It was also the first successful trial targeting two different weed targets with a single model. An important result from this work is the ability of the AutoWeed detection system to correctly discern the difference between chickpea and sowthistle, two broadleaf plant species. At present, there are few herbicides capable of selectively treating within broadleaf plant varieties. AutoWeed therefore presents a demonstrably useful method of controlling a broadleaf weed in a broadleaf crop. In this particular paddock, the non-broadleaf wheat crop would be blanket sprayed with a broadleaf selective herbicide to achieve control of the weeds. Therefore, this detection and spray technology could be used to spray 4% of the wheat paddock with a stronger herbicide to achieve better control at a lower cost in a shorter time frame.

Figure 5.52: Left: Alex Olsen conducting the AutoWeed trial at Arcturus with local landholders in attendance for WeedSmart Week Emerald 2019. Right: The location data for the Arcturus trial.

5.4 Chapter Summary

In summary, this chapter has contributed the following:

- The development of *AutoWeed*, a novel weed detection and spraying prototype for selectively applying herbicide to control weed species. This represents a new and efficient tool that will aid weed management in both crop and pasture.
- The development of novel ROS software to control the robotic prototype, deploy weed detection systems as inference engines in the field with input from cameras, and selectively activate solenoid sprayers.
- The utilisation of off-the-shelf highly precise positioning equipment to develop a positioning system that provided adequate accuracy for tracking and spraying of weed targets in real time. The spraying system achieved 100% target accuracy during all straight line and complex path motion tests.
- A suite of five trials were conducted across the eastern coast of Australia on Chinee apple, Navua sedge and Harrisia cactus in pasture paddocks; turnip weed in an oats crop; and sowthistle weed and unwanted chickpea plants in a wheat crop. The AutoWeed system achieved on average, 90% classification accuracy across the five trials with between 10-20% non-detections and less than 3% false detections. Compared to traditional blanket spraying of the trial areas, the AutoWeed system can provide up to a 95% reduction in herbicide usage.
- The successful prototype development in this chapter laid the foundation for a $50,000 grant project starting from July 2019 to June 2020 funded by round two of the Department of Agriculture and Water Resources' Smart Farms Small Grant scheme. The project will deliver a bespoke AutoWeed detection system retro-fitted to a side-by-side utility-terrain vehicle targeting Harrisia cactus in the Goondiwindi region.

Let us now turn a critical eye to this work, and ask: 1) Were the goals met? 2) What can be improved? and 3) What lies ahead for future research, development and commercialisation of this prototype?

Positioning

The positioning requirements of this system were made more complex by spacing the cameras and the sprayers over 1 m apart. This was initially conceived for two purposes:

1) to provide adequate time for our image processing algorithms and 2) to ensure the weed targets were first seen before being trampled by the machine. However, the first purpose is mitigated with the powerful innovations in hardware that allow complex deep learning architectures to achieve faster than required inference times (as shown in Chapter 4). And after significant in field testing, it was found that the clearance underneath the target vehicle rarely damaged the weeds as the vehicle passed over. As a result, the system may greatly benefit by moving the cameras closer to the sprayers and negating the need for sophisticated positioning algorithms and hardware.

The Advanced Navigation Spatial Dual is an extremely sophisticated GPS and IMU system that provides sufficient positioning accuracy at a high price point ($\geq$ \$10,000). With the cameras and the nozzles adjacently positioned, no tracking would be required to maintain knowledge of weed positions once detected. Instead, the sprayers can be immediately activated whenever a weed passes under a camera. This would also decrease the margin of error in the positioning system. The overall system robustness may improve with a reduced number of subsystems and therefore a reduction in potential sources of error.

This direction for future work would have an enormous impact on the commercial potential of this system by significantly reducing its price and potentially increasing its robustness to error.

Spraying

The spraying system utilised for this prototype is entirely conventional. All parts were purchased off-the-shelf, except for a solenoid driver PCB which activates the solenoids for a given amount of time after receiving a software command.

The spraying performance of the vehicle could be improved by changing the nozzles as required for each weed target. Foliar spraying of Harrisia cactus, for example, requires the entire stem of the weed to be doused in herbicide. While the prototype successfully detects and activates its sprayers on the cactus, the solenoid and nozzle does not provide enough herbicide volume to drench the cactus. As a result, a larger orifice solenoid and higher capacity nozzle would be desired when targeting this specific weed. This, in turn, may alter the tank, pump and plumbing system required for each weed.

Fortunately, little research and development is required to achieve the spraying performance required. This spraying technology exists and can be easily accessed as needed. The weed detection system, which is the focus of this thesis, can be retro-fitted to any existing agricultural vehicle for a variety of control mechanisms.

Detection

A workflow for performing dataset collection, labelling and detection was established during this chapter that was adapted to suit multiple weed species in multiple environments. Overall, the weed detection models achieved approximately 90% successful detections with approximately 5% false detections and 5% non-detections in the field. Success of the detection model was noticeably affected by the diversity of plant life in the native background environment of the weed being targeted. Generally, it was found, that more images were required to be collected in these more difficult environments to achieve the stated level of accuracy.

Our workflow was entirely dependent on site-specific datasets. The ability of our detection models to transfer their performance on a weed irrespective of location was mixed. It is reasoned this is due to the site-specific datasets not capturing the full variability of the target environment. Future work in this arena should investigate how many images it will take in each target environment to capture the full range of variability, perhaps to the point where no more images may need to be collected. This will be an extremely important factor surrounding the commercialisation of robotic weed control. The sheer size of weed species datasets that will be collected in the commercialisation of this technology will require innovative *big data* management to continually manage, re-train and deploy detection models to customers across the world.

The earlier suggestion of moving the cameras closer to the nozzles to mitigate the reliance on complex positioning and navigation requires the detection models to perform inference even faster. This will be a staple of future research and development as new and better deep learning architectures and edge computing devices are introduced that provide stronger performance.

Usability

The prototype developed throughout this chapter remains a proof of concept and is not yet ready for commercialisation in many ways. To facilitate user adoption, a simple and robust user interface will be required. For the farmer, this could be a touch screen interface or an ISOBUS compatible display in the cabin of the agricultural vehicle. The interface should display the current status of the weed control prototype (including detection and spray history, status of system and current positioning information).

The aforementioned suggestions to reduce the complexity required of the positioning system will also work towards reducing the sources of error present in the prototype. This would allow for a more modular and more robust system.

Partnering software must also be developed, extending from the labelling software de-

veloped for this thesis. This software could provide the ability for an unskilled user to collect, re-train and deploy weed detection models on their robotic product.

Economic and Environmental Impact

Despite the aforementioned limitations of the prototype developed for this thesis, the promise of this solution remains economically and environmentally viable. It is clear that such a solution will save landholders in some cases up to 95% of their herbicide costs by only treating the fraction of their paddock with weeds.

In certain cropping situations, where it is required to spray a broad-leaf weed in a broad-leaf crop, this solution is even more important. Currently few herbicides exists for this type of treatment. Whereas, the AutoWeed solution can be utilised to selectively treat weeds with a higher concentration herbicide to ensure effective control.

Meanwhile, the environmental impact in a pasture situation is readily apparent. Farms are often so large, and management of rangeland weeds so costly, that the practice is being ignored which results in weed species becoming more wide spread. With the adoption of robotic weed control technology, like that presented here, landholders with limited options can get their land back under control as the practice becomes more efficient and cost effective.

Chapter 6

Conclusion

In conclusion, this thesis has contributed new weed detection frameworks and robotic weed control systems that achieve high accuracy in complex real time environments as a result of scrupulous dataset collection, development and field testing. This chapter begins by summarising the novelty of each contribution and its value in the literature. We then discuss possible directions of future work in the application of deep learning in weed species detection. Finally, we look to the horizon and discuss how robotic weed detection will benefit the agricultural industry and why its commercialisation is needed to provide new tools for integrated weed management.

6.1 Contributions

There are many novel contributions presented in this thesis that advance the real-time performance and accuracy of weed species and their utility for robotic weed control. The early contributions of the work are focused on a case study targeting lantana using traditional image analysis. These contributions include:

- The development of a successful novel leaf area segmentation algorithm to isolate broad-shaped leaves from a complex in situ background based on traditional colour thresholding and image morphology operations.
- The innovation of the Histograms of Oriented Gradients feature set to be rotation and scale invariant by way of a dominant rotation alignment and multi-scale fusion approach, respectively.
- The implementation of the novel HoG approach to detect the lantana weed species in its native environment based solely on the texture of its leaf surface with real-time throughput.
- The introduction of a challenging 337 image in situ dataset of lantana upon which the texture-based classifier achieved 86.1% accuracy.
- The development of a novel dead-reckoning positioning algorithm using only an IMU for yaw and a WSS for speed to achieve $\pm$ 50 mm accuracy over 5 m of straight line motion.
- The proof of concept development of a novel weed detection and spraying system consisting of a towable trailer with mounted machine vision technology. The positioning system provided adequate accuracy for spraying during straight line motion, however failed in more complex paths. Meanwhile, the detection system performed well in scenarios matching the in situ dataset, but did not generalise well to unseen scenarios.

Following the development of the proof of concept case study, it was apparent that deep learning based approaches offered a more powerful approach to weed detection. But these new tools required extensive dataset collection matching the targeted application. The following contributions include new tools and techniques for dataset collection and deploying deep learning frameworks for weed species detection.

- The AlexNet deep CNN architecture was revised and deployed to classify lantana with 88.4% accuracy, outperforming the handcrafted texture feature set. This re-

vealed the power of deep learning for weed species detection to automatically learn requisite distinctive features.

- Designed and developed a fast, easy-to-use image acquisition and dataset collection instrument, *WeedLogger*, for capturing images of weed species in situ with the identical optical system of a prototype weed control robot. The system also allows in-field labelling via a touchscreen and rapidly accelerated dataset collection.

- Wrote custom labelling software for fast curation and manual classification of weed species images for binary and multiclass classification. The software allows labelling of thousands of images per hour.

- Introduced *DeepWeeds*, a 19,507 labelled image dataset of eight weed species from eight different locations across northern Australia. Shared publicly, this dataset is the first of its kind, providing the opportunity for researchers to develop and benchmark new detection algorithms for some important Australian rangeland weed species.

- Benchmark performance on the DeepWeeds dataset was presented for AlexNet, Inception-v3, ResNet-50, MobileNetV2 and VGG16 achieving 86.2%, 95.1%, 91.6%, 95.7% and 92.3% accuracy, respectively. These results were indicative of the model complexity of each architecture and present unique tradeoffs of speed and accuracy for real-time detection.

- The real-time performance of the aforementioned models was extensively evaluated on a variety of state-of-the-art edge computing devices. This presents the first benchmark of the NVIDIA family of edge computing devices for weed species detection. The benchmark provides researchers a look up table to determine which model and hardware architectures are ideal for their given performance requirement.

- It was shown that the AlexNet and MobileNetV2 architectures offered the fastest performance with inference speeds as low as 2 ms on a GeForce GTX 1080. The power consumption of all models was closely correlated with inference time. It was also found that the ResNet-50 and MobileNetV2 architectures offered the best tradeoff between inference speed and classification accuracy and thus ideal for real-time weed detection.

These advances in weed species detection using deep learning were then applied to develop a new prototype for robotic weed control. The contributions during the development and in-field testing of the novel prototype include:

- A novel weed detection and spraying prototype, *AutoWeed*, was designed and developed here providing a new, efficient tool to aid weed management in both crop and pasture.
- Novel software was developed deploying the aforementioned weed detection systems as inference engines that take images from cameras at the front of the vehicle and identify targets for the vehicle to track and spray as it passes over them.
- Off-the-shelf highly precise positioning equipment was utilised to program a positioning system that provided more than adequate accuracy for tracking and spraying weed targets in real time. The spraying system achieved 100% target accuracy during all straight line and complex path motion tests.
- A suite of five trials were conducted across the eastern coast of Australia on Chinee apple, Navua sedge and Harrisia cactus in pasture paddocks; turnip weed in an oats crop and sowthistle weed in a wheat crop. The AutoWeed system achieved on average, 90% classification accuracy across the five trials with between 10-20% non-detections and less than 3% false detections. Compared to traditional blanket spraying of the trial areas, the AutoWeed system provides up to a 95% reduction in herbicide usage.

In summary, the contributions of this thesis have advanced the efficacy of weed species detection in crop and pasture through the provision of a new robotic weed control prototype that is ripe for commercialisation. As a new tool for integrated weed management, this robotic weed control solution could provide a step change in the efficiency of agricultural production.

6.2 Future Work

There are a variety of future directions for extending this work within each field of interest, including new deep learning approaches, accelerating real-time inference, growing and fast-tracking dataset collection and extending detection to other weed control methods.

6.2.1 Deep Learning

The methods developed herein are focused on classification as the central learning task. This approach works for weed targets that are of similar size to the application spray size. More precise object detection and localisation of weed targets is required to detect seedlings and smaller weed species targets. To allow future work in this arena, the DeepWeeds

dataset has been labelled with bounding boxes identifying weed targets. These algorithms are more complex and will be more difficult to implement at high vehicle speeds in real time. Therefore, research into accelerating deep CNN architectures for bounding box detection (such as Tiny-YOLO [121], Faster R-CNN [167] and Single-Shot Box Detectors [78]) of weed species should be investigated.

Furthermore, the research space of modern deep learning architecture sees continual innovation. The latest architectures and their step improvements in accuracy, training time and real-time performance should continue to be investigated to maximise weed species detection performance.

Similarly, Moore's law of exponential growth in computing will continue to see new and more powerful edge computing devices beyond the Jetson family investigated here. The latest advances in hardware should continue to be exploited, this will allow future weed detection units to operate at faster real time speeds which will benefit landholders. Other hardware avenues of research include using FPGA-based implementations of deep learning architectures as in [143, 144].

Accelerating deep CNNs with architecture-level optimisations is also an active field of research. TensorRT has introduced new methods of optimisation from reducing weight precision to removing redundant layer operations [126]. Meanwhile, fully binarised weight precision implementations of deep CNN architectures offer tremendous speed-up of inference with little loss of accuracy [144].

6.2.2 Dataset Collection

As with most applications of machine learning in the modern world, the focus of the approach reduces to managing *big data*. This thesis has shown that more images of weed species captured under as many factors of variation as possible will continue to see the accuracy of weed species detection increase. Future work should prioritise the collection of larger weed species datasets. The AutoWeed robotic platform has already been used to collect and label over 200,000 images since its introduction in this thesis. The accuracy of the platform will continue to improve.

Another possible direction for future work is utilising very large banks of image dataset to automatically label new images reducing the need for manual labelling and greatly accelerating the learning process. Smart and efficient dataset management will provide the scaffolding for future commercialisation of machine vision based weed detection systems.

6.2.3 Robotic Weed Control

The *AutoWeed* prototype developed here in Chapter 5 is ready for commercialisation. Future work to facilitate commercial success includes making the unit more modular, user-friendly and cheaper. Future work will see the development of boom-mounted green-on-green detection systems that operate independently of one another to cover a larger field of view. These systems could ideally be retro-fitted to existing farming equipment or attached to state-of-the-art autonomous platforms like the SwarmFarm [24] fleet. This direction of future research and development will be the focus of a $50,000 research grant from the Australian Government under round two of the Smart Farms Small Grant program. This grant will extend from July 2019 to June 2020 and was made possible by the successful prototype development presented in this thesis.

This thesis exclusively used foliar spot-spraying as the control technique to evaluate our weed detection systems. A variety of other weed control techniques are being investigated for robotisation, including thermal weed control [168] and targeted physical control by mechanical means [25–27]. All such systems rely on accurate detection of the weed from its native background before applying the control technique. The work in this thesis therefore presents logical innovations for these other means of control to make them readily applicable.

The main contribution of this thesis is a commercially viable prototype for autonomous detection of weed species in both crop and pasture environments. The commercialisation of this new technology is a major focus of future work in the industry. This will likely provide a new tool for landholders to improve weed management practice and in turn, improve the efficiency of their farming operations.

Bibliography

[1] Standards Australia, *The International system of units (SI) and its application*. AS ISO 1000-1998, 1998.

[2] Macquarie Dictionary, *Macquarie Dictionary*. Macquarie Dictionary Publishers Pty Ltd, 7 ed., 2017.

[3] Invasive Plants and Animals Committee, *Australian Weeds Strategy 2017 to 2027*. Canberra, Australia: Australian Government Department of Agriculture and Water Resources, 2016.

[4] B. de Hayr, *Health of the Landcare Movement: Survey Results*. Sydney, Australia: Landcare Australia, 2013.

[5] H. Combellack, "Weed control pursuits in Australia," *Chemistry and Industry*, pp. 273–280, April 1987.

[6] J. Sinden, R. Jones, S. Hester, D. Odom, C. Kalisch, R. James, and O. Cacho, "The economic impact of weeds in australia," in *CRC for Australian Weed Management, Technical Series*, 8, (Adelaide, Australia), March 2004.

[7] R. McLeod, *Annual Costs of Weeds in Australia*. Canberra, Australia: Centre for Invasive Species Solutions, November 2018.

[8] Commonwealth of Australia, *Agricultural Competitiveness White Paper*. February 2015. ISBN: 978-1-925237-73-3.

[9] Department of Agriculture and Water Resources, "Control tools and technologies for established pest animals and weeds grant programme 2017." Online: `http://www.agriculture.gov.au/pests-diseases-weeds/pest-animals-and-weeds/wp-comp-grants-programme`. Accessed: 2019-08-09.

[10] Department of Agriculture and Fisheries, "Prickly acacia fact sheet 2016." Online: `https://www.daf.qld.gov.au/__data/assets/pdf_file/0007/73753/IPA-Prickly-Acacia-PP9.pdf`, 2016. Accessed: 2020-03-24.

[11] Natural Resource Management Ministerial Council, *The Australian Weeds Strategy: A national strategy for weed management in Australia*. Canberra, Australia: Australian Government Department of the Environment and Water Resources, 2006.

[12] J. Thorp and R. Lynch, *The Determination of Weeds of National Significance: Australia's twenty worst weeds – wanted dead*. Canberra, Australia: Australian Government Department of Agriculture Fisheries and Forestry, 2000.

[13] Australian Weeds Committee, *Weeds of national significance 2012*. Canberra, Australia: Department of Agriculture, Fisheries and Forestry, 2012. ISBN: 978 0 9803249 3 8.

[14] G. Bastin and the ACRIS Management Committee, *Rangelands 2008 – Taking the Pulse*. Canberra, Australia: National Land & Water Resources Audit, 2008.

[15] A. C. Grice, S. Campbell, R. Breaden, F. Bebawi, and W. Vogler, *Habitat management guide - Rangelands: Ecological principles for the strategic management of weeds in rangeland habitats*. Adelaide, Australia: CRC for Australian Weed Management, 2008.

[16] Australian Bureau of Agricultural and Resource Economics (ABARES), "AGSURF Data." Online: `https://apps.daff.gov.au/AGSURF/`, 2018. Accessed: 2019-04-20.

[17] R. E. C. McFadyen, "Biological control of weeds," *Annual Review of Entomology*, vol. 43, no. 1, pp. 369–393, 1998. PMID: 15012395.

[18] D. Stock, K. M. Johnson, A. Clark, and E. van Oosterhout, *Lantana best practice manual and decision support tool*. Queensland, Australia: Department of Employment, Economic Development and Innovation, 2009.

[19] C. J. Swanton and S. F. Weise, "Integrated weed management: The rationale and approach," *Weed Technology*, vol. 5, no. 3, p. 657–663, 1991.

[20] Croplands, "WEEDit Optical Spot Spraying." Online: `https://www.croplands.com.au/Products/WEEDit-Optical-Spot-Spraying`. Accessed: 2019-10-19.

[21] Trimble, "WeedSeeker Spot Spray System." Online: `https://agriculture.trimble.com/product/weedseeker-spot-spray-system/`. Accessed: 2019-10-19.

[22] D. C. Slaughter, D. K. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Computers and Electronics in Agriculture*, vol. 61, pp. 63–78, 2008.

[23] D. L. Shaner and H. J. Beckie, "The future for weed control and technology," *Pest Management Science*, vol. 70, pp. 1329–1339, 2014.

[24] SwarmFarm Robotics Pty Ltd, "SwarmFarm Robotics." Online: `https://www.swarmfarm.com/`. Accessed: 2019-09-08.

[25] O. Bawden, D. Ball, J. Kulk, T. Perez, and R. Russell, "A lightweight, modular robotic vehicle for the sustainable intensification of agriculture," in *Australian Conference on Robotics and Automation (ACRA 2014)*, (University of Melbourne, Melbourne, VIC), Australian Robotics & Automation Association ARAA, December 2014.

[26] A. Ruckelshausen, P. Biber, M. Dorna, H. Gremmes, R. Klose, A. Linz, R. Rahe, R. Resch, M. Thiel, D. Trautz, and U. Weiss, "BoniRob: An autonomous field robot platform for individual plant phenotyping," *Precision Agriculture*, vol. 9, pp. 841–847, January 2009.

[27] C. Peressini, A. Guzzomi, S. Powles, and M. Walsh, "Design of a novel mechanical targeted tillage tool for weed removal in large-scale crop fields," in *Proceedings of the 2016 CIGR*, (Aarhus, Denmark), pp. 1–8, 2016.

[28] "Blue River Technology." Online: `http://smartmachines.bluerivertechnology.com/`. Accessed: 2019-10-19.

[29] S. Sonka, "Big Data and the Ag Sector: More than Lots of Numbers," *International Food and Agribusiness Management Review*, vol. 17, p. 20, February 2014.

[30] S. G. Wu, F. S. Bao, E. Y. Xu, Y. Wang, Y. Chang, and Q. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, (Cairo, Egypt), pp. 11–16, December 2007.

[31] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. B. Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *Proceedings of the 2012 European Conference on Computer Vision (ECCV)*, (Berlin, Heidelberg), pp. 502–516, 2012.

[32] D. Hall, C. McCool, F. Dayoub, N. Sunderhauf, and B. Upcroft, "Evaluation of features for leaf classification in challenging conditions," in *Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (Hawaii, USA), pp. 797–804, 2015.

[33] C. Kalyoncu and Önsen Toygar, "Geometric leaf classification," *Computer Vision and Image Understanding*, vol. 133, pp. 102 – 109, 2015.

[34] A. dos Santos Ferreira, D. M. Freitas, G. G. da Silva, H. Pistori, and M. T. Folhes, "Weed detection in soybean crops using ConvNets," *Computers and Electronics in Agriculture*, vol. 143, pp. 314 – 324, 2017.

[35] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino, "How deep learning extracts and learns leaf features for plant classification," *Pattern Recognition*, vol. 71, pp. 1–13, November 2017.

[36] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, (Québec City, Canada), pp. 452–456, 2015.

[37] J. Carranza-Rojas, H. Goeau, P. Bonnet, E. Mata-Montero, and A. Joly, "Going deeper in the automated identification of herbarium specimens," *BMC Evolutionary Biology*, vol. 17, p. 181, 2017.

[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[39] G. Jourdain, H. Serrat, and J. Beguerie, "Bilberry." Online: `https://bilberry.io/`, 2019. Accessed: 2019-10-25.

[40] R. Shepherd, R. Richardson, and F. Richardson, *Plants of Importance to Australia: A checklist*. Meredith, Victoria, 2001.

[41] NSW Department of Primary Industries, "Review of the declaration of Lantana species in New South Wales." Online: `http://www.dpi.nsw.gov.au/__data/assets/pdf_file/0011/216848/Review-of-the-declaration-of-Lantana-species-in-NSW.pdf`, 2007. Accessed: 2020-03-25.

[42] Department of Agriculture and Fisheries, "Lantana fact sheet 2016." Online: `https://www.daf.qld.gov.au/__data/assets/pdf_file/0009/62010/IPA-Lantana-PP34.pdf`, 2016. Accessed: 2019-10-19.

[43] L. Smith and D. Smith, "The naturalised Lantana camara complex in eastern Australia," *Queensland Botany Bulletin*, vol. 1, pp. 1–26, 1982.

[44] J. Swarbrick, B. Wilson, and M. Hannan-Jones, "The Biology of Australian Weeds: 25. Lantana camara," *Plant Protection Quarterly*, vol. 10, no. 1, pp. 82–95, 1995.

[45] G. Sharma, A. Raghubanshi, and J. Singh, "Lantana invasion: An overview," *Weed Biology and Management*, vol. 5, no. 4, pp. 157–165, 2005.

[46] Biosecurity Queensland and National Lantana Management Group, *Plan to protect environmental assets from lantana*. Yeerongpilly, Queensland: Department of Employment, Economic Development and Innovation, 2010.

[47] Department of Environment and Resource Management, *Mapping lantana using Landsat: a remote sensing centre report*. Brisbane, Australia: Queensland Government, 2010.

[48] C. Gentle and J. Duggin, "Allelopathy as a competitive strategy in persistent thickets of Lantana camara l. in three Australian forest communities," *Plant Ecology*, vol. 132, pp. 85–95, 1997.

[49] R. Lamb, "Some effects of lantana camara on community dynamics of eucalypt woodlands," in *Proceedings of the 52nd ANZAAZ Congress*, (Macquarie University, New South Wales), p. 304, 1982.

[50] AEC Group, *Economic impact of lantana on the Australian grazing industry*. Queensland Government, Australia: Department of Natural Resources and Water, 2007.

[51] Australian Weeds Committee, *Lantana (Lantana camara L.) strategic plan 2012-17*. Canberra, Australia: Department of Agriculture, Fisheries and Forestry, 2012.

[52] Department of the Environment and Heritage and the CRC for Australian Weed Management, "Lantana (Lantana camara) weed management guide." Online: `https://www.environment.gov.au/biodiversity/invasive/weeds/publications/guidelines/wons/l-camara.html`. Accessed: 2019-10-19.

[53] A. Wang, W. Zhang, and X. Wei, "A review on weed detection using ground-based machine vision and image processing techniques," *Computers and Electronics in Agriculture*, vol. 158, pp. 226 – 240, 2019.

[54] Z. Wang, Z. Chi, and D. Feng, "Shape based leaf image retrieval," in *In Proceedings ofthe IEEE Conference on Vision, Image and Signal Procesing*, vol. 150, pp. 34–43, February 2003.

[55] J. X. Du, X. F. Wang, and G. J. Zhang, "Leaf shape based plant species recognition," *Applied Mathematics and Computation*, vol. 185, pp. 883–893, February 2007.

[56] O. M. Bruno, R. O. Plotze, M. Falvo, and M. Castro, "Fractal dimension applied to plant identification," *Information Sciences*, vol. 178, pp. 2722–2733, June 2008.

[57] A. R. Backes, D. Casanova, and O. M. Bruno, "Plant leaf identification based on volumetric fractal dimension," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, pp. 1145–1160, September 2009.

[58] J. Cope, P. Remagnino, S. Barman, and P. Wilkin, "Plant texture classification using Gabor co-occurences," *Advances in Visual Computing*, vol. 6454, 2010.

[59] Y. Li, Q. S. Zhu, Y. Cao, and C. L. Wang, "A leaf vein extraction method based on snakes technique," in *In Proceedings of the International Conference on Neural Networks and Brain*, vol. 2, pp. 885–888, October 2005.

[60] H. Fu and Z. Chi, "Combined thresholding and neural network approach for vein pattern extraction from leaf images," in *In Proceedings of the IEEE Conference on Vision, Image and Signal Processing*, vol. 153, pp. 881–892, December 2006.

[61] T. Beghin, J. S. Cope, P. Remagnino, and S. Barman, "Shape and texture based plant leaf classification," *Advanced Concepts for Intelligent Vision Systems*, vol. 6475, pp. 345–353, January 2010.

[62] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, "Leaf classification using shape, color and texture features," *International Journal of Computer Trends and Technology*, vol. 1, pp. 225–230, July 2011.

[63] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005.

[64] N. Dalal, *Finding people in images and videos.* PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 2006.

[65] O. Deniz, G. Bueno, J. Salido, and F. D. la Torre, "Face recognition using Histograms of Oriented Gradients," *Pattern Recognition Letters*, vol. 32, pp. 1598–1603, September 2011.

[66] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal on Computer Vision*, vol. 60, pp. 91–110, 2004.

[67] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[68] D. R. Cox, "The regression analysis of binary sequences (with discussion)," *Journal of the Royal Statistical Society*, vol. 20, pp. 215–242, 1958.

[69] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[70] R. Rojas, *Neural networks: a systematic introduction*. Berlin, Germany: Springer-Verlag, 1996.

[71] Y. Xu, K. Imou, Y. Kaizu, and K. Saga, "Two-stage approach for detecting slightly overlapping strawberries using HOG descriptor," *Biosystems Engineering*, vol. 115, pp. 144–153, June 2013.

[72] Itseez, "Open Source Computer Vision (OpenCV) Library." `https://github.com/itseez/opencv`, 2015.

[73] K. Park, H. Chung, J. Choi, and J. G. Lee, "Dead reckoning navigation for an autonomous mobile robot using a differential encoder and a gyroscope," in *Proceedings of the 8th International Conference on Advanced Robotics (ICAR)*, (Monterey, CA, USA), pp. 441–446, July 1997.

[74] F. Ahmed, H. A. Al-Mamun, A. H. Bari, E. Hossain, and P. Kwan, "Classification of crops and weeds from digital images: A support vector machine approach," *Crop Protection*, vol. 40, pp. 98 – 104, 2012.

[75] P. J. Herrera, J. Dorado, and n. Ribeiro, "A novel approach for weed type classification based on shape descriptors and a fuzzy decision-making method," *Sensors*, vol. 14, no. 8, pp. 15304–15324, 2014.

[76] A. Olsen, S. Han, B. Calvert, P. Ridd, and O. Kenny, "In situ leaf classification using histograms of oriented gradients," in *Proceedings of the 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, (Adelaide, Australia), pp. 441–448, 2015.

[77] J. Yu, S. M. Sharpe, A. W. Schumann, and N. S. Boyd, "Deep learning for image-based weed detection in turfgrass," *European Journal of Agronomy*, vol. 104, pp. 78 – 84, 2019.

[78] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *Lecture Notes in Computer Science*, p. 21–37, 2016.

[79] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv pre-print: arXiv/1409.1556*, 2014.

[80] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[81] Agropolis Foundation, "Pl@ntNet." Online: `https://identify.plantnet.org/`. Accessed: 2019-10-13.

[82] A. Bakhshipour and A. Jafari, "Evaluation of support vector machine and artificial neural networks in weed detection using shape features," *Computers and Electronics in Agriculture*, vol. 145, pp. 153 – 160, 2018.

[83] M. Dyrmann, R. N. Jørgensen, and H. S. Midtiby, "RoboWeedSupport - Detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network," *Advances in Animal Biosciences*, vol. 8, no. 2, p. 842–847, 2017.

[84] A. Shirzadifar, S. Bajwa, S. A. Mireei, K. Howatt, and J. Nowatzki, "Weed species discrimination based on SIMCA analysis of plant canopy spectral data," *Biosystems Engineering*, vol. 171, pp. 143 – 154, 2018.

[85] L. Li, X. Wei, H. Mao, and S. Wu, "Design and application of spectrum sensor for weed detection used in winter rape field," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 33, no. 18, pp. 127–133, 2017.

[86] M. Louargant, G. Jones, R. Faroux, J.-N. Paoli, T. Maillot, C. Gée, and S. Villette, "Unsupervised classification algorithm for early weed detection in row-crops by combining spatial and spectral information," *Remote Sensing*, vol. 10, no. 5, p. 761, 2018.

[87] F. Lin, D. Zhang, Y. Huang, X. Wang, and X. Chen, "Detection of corn and weed species by the combination of spectral, shape and textural features," *Sustainability*, vol. 9, no. 8, p. 1335, 2017.

[88] S. Mahesh, D. Jayas, J. Paliwal, and N. White, "Hyperspectral imaging to classify and monitor quality of agricultural materials," *Journal of Stored Products Research*, vol. 61, pp. 17 – 26, 2015.

[89] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planqué, A. Rauber, S. Plazzo, B. Fisher, and H. Müller, "LifeCLEF 2015: Multimedia Life

Species Identification Challenges," in *Proceedings of the 2015 International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF)*, vol. 1391, (Toulouse, France), pp. 462–483, November 2015.

[90] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, J. Champ, R. Planqué, S. Palazzo, and H. Müller, "LifeCLEF 2016: Multimedia Life Species Identification Challenges," in *Proceedings of the 2016 International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF)*, (Évora, Portugal), pp. 286–310, August 2016.

[91] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, J.-C. Lombardo, R. Planqué, S. Palazzo, and H. Müller, "LifeCLEF 2017 Lab Overview: Multimedia Species Identification Challenges," in *Proceedings of the 2017 International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF)*, (Dublin, Ireland), pp. 255–274, 2017.

[92] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Boston, USA), pp. 1–9, 2015.

[93] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, USA), pp. 770–778, 2016.

[94] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2016.

[95] M. Lasseck, "Image-based plant species identification with deep convolutional neural networks," in *LifeCLEF 2017*, (Dublin, Ireland), 2017.

[96] T. B. Moeslund, *Introduction to Video and Image Processing*. Springer-Verlag London, 2012.

[97] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, B. Calvert, M. Rahimi Azghadi, and R. D. White, "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning," *Scientific Reports*, vol. 9, February 2019.

[98] TensorFlow, "TensorFlow Datasets v1.3.0." Online: `https://www.tensorflow.org/datasets`, November 2019. Accessed: 2019-11-11.

[99] J. Wäldchen and P. Mäder, "Plant species identification using computer vision techniques: A systematic literature review," *Archives of Computational Methods in Engineering*, vol. 25, pp. 507–543, April 2017.

[100] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, vol. 1, (Lake Tahoe, USA), pp. 1097–1105, 2012.

[101] M. Pérez-Ortiz, J. Peña, P. Gutiérrez, J. Torres-Sánchez, C. Hervás-Martínez, and F. López-Granados, "A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method," *Applied Soft Computing*, vol. 37, pp. 533 – 544, 2015.

[102] J.-L. Tang, X.-Q. Chen, R.-H. Miao, and D. Wang, "Weed detection using image processing under different illumination for site-specific areas spraying," *Computers and Electronics in Agriculture*, vol. 122, pp. 103 – 111, 2016.

[103] O. Barrero, D. Rojas, C. Gonzalez, and S. Perdomo, "Weed detection in rice fields using aerial images and neural networks," in *Proceedings of the 2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*, (Bucaramanga, Colombia), pp. 1–4, August 2016.

[104] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, USA), pp. 2818–2826, 2016.

[105] H. Goëau, P. Bonnet, and A. Joly, "Plant identification based on noisy web data: the amazing performance of deep learning," in *LifeCLEF 2017*, (Dublin, Ireland), 2017.

[106] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, November 1998.

[107] M. Lin, Q. Chen, and S. Yan, "Network In Network," *arXiv pre-print: arXiv/1312.440*, 2013.

[108] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv pre-print: arXiv/1704.04861*, 2017.

[109] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv pre-print: arXiv/1801.04381*, 2018.

[110] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv pre-print: arXiv/1502.03167*, 2015.

[111] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[112] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *arXiv pre-print: arXiv/1311.2901*, 2013.

[113] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[114] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *Multiple Classifier Systems* (T. Windeatt and F. Roli, eds.), (Berlin, Heidelberg), pp. 125–134, Springer Berlin Heidelberg, 2003.

[115] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, (San Diego, USA), 2015.

[116] L. van der Maaten, "Barnes-Hut-SNE," *arXiv pre-print: arXiv/1301.3342*, 2013.

[117] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *arXiv pre-print: arXiv/1512.04150*, 2015.

[118] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv pre-print: arXiv/1311.2524*, 2013.

[119] B. Calvert, A. Olsen, B. Philippa, and M. R. Azghadi, "AutoWeed: Detecting Harrisia Cactus in the Goondiwindi region for selective spot-spraying," in *Proceedings of the 1st Queensland Pest Animal and Weed Symposium (PAWS)*, (Gold Coast, Australia), May 2019.

[120] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, June 2015.

[121] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv pre-print: arXiv/1506.02640*, 2015.

[122] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *arXiv pre-print: arXiv/1707.07012*, 2017.

[123] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv pre-print: arXiv/1602.07261*, 2016.

[124] TeeJet, "TeeJet: Nozzle spacing and minimum spray heights." Online: `https://teejet.it/english/home/tech-support/nozzle-technical-information/nozzle-spacing-and-minimum-spray-heights.aspx`, 2019. Accessed: 2019-10-4.

[125] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems." `https://www.tensorflow.org`, 2015.

[126] NVIDIA Corporation, "TensorRT." `https://developer.nvidia.com/tensorrt`, 2018.

[127] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, H. Zhou, and X. Wang, "Crafting GBD-Net for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 2109–2123, October 2016.

[128] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint: arXiv/1709.01507*, 2017.

[129] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, (Sardinia, Italy), pp. 249–256, 2010.

[130] C.-C. Yang, "Application of artificial neural networks in image recognition and classification of crop and weeds," *Canadian Biosystems Engineering / Le Genie des biosystems au Canada*, vol. 42, pp. 147–152, January 2000.

[131] P. Lottes, J. Behley, A. Milioto, and C. Stachniss, "Fully convolutional networks with sequential information for robust crop and weed detection in precision farming," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2870–2877, October 2018.

[132] V. Czymmek, L. O. Harders, F. J. Knoll, and S. Hussmann, "Vision-based deep learning approach for real-time detection of weeds in organic farming," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–5, May 2019.

[133] P. Lottes and C. Stachniss, "Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5155–5161, September 2017.

[134] V. Partel, S. C. Kakarla, and Y. Ampatzidis, "Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence," *Computers and Electronics in Agriculture*, vol. 157, pp. 339 – 350, 2019.

[135] A. Milioto, P. Lottes, and C. Stachniss, "Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2/W3, pp. 41–48, 2017.

[136] A. Milioto, P. Lottes, and C. Stachniss, "Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2229–2235, May 2018.

[137] J. Yu, S. M. Sharpe, A. W. Schumann, and N. S. Boyd, "Detection of broadleaf weeds growing in turfgrass with convolutional neural networks," *Pest Management Science*, vol. 75, no. 8, pp. 2211–2218, 2019.

[138] I. Sa, Z. Chen, M. Popović, R. Khanna, F. Liebisch, J. Nieto, and R. Siegwart, "WeedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming," *IEEE Robotics and Automation Letters*, vol. 3, pp. 588–595, January 2018.

[139] Ł. Chechliński, B. Siemiątkowska, and M. Majewski, "A System for Weeds and Crops Identification—Reaching over 10 FPS on Raspberry Pi with the Usage of MobileNets, DenseNet and Custom Modifications," *Sensors*, vol. 19, no. 17, 2019.

[140] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv pre-print: arXiv/1505.04597*, 2015.

[141] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *arXiv pre-print: arXiv/1608.06993*, 2016.

[142] NVIDIA Corporation, "TensorRT Developer Guide." Online: `https://docs.nvidia.com/deeplearning/sdk/tensorrt-developer-guide/index.html`, 2019. Accessed: 2019-10-6.

[143] F. J. Knoll, M. Grelcke, V. Czymmek, T. Holtorf, and S. Hussmann, "CPU architecture for a fast and energy-saving calculation of convolution neural networks," in *Digital Optical Technologies 2017* (B. C. Kress and P. Schelkens, eds.), vol. 10335, pp. 362 – 370, International Society for Optics and Photonics, SPIE, 2017.

[144] C. Lammie, A. Olsen, T. Carrick, and M. Rahimi Azghadi, "Low-Power and High-Speed Deep FPGA Inference Engines for Weed Classification at the Edge," *IEEE Access*, vol. 7, pp. 51171–51184, 2019.

[145] Grains Research & Development Corporation (GRDC), "Practical tips for spraying: Fact sheet." Online: `https://grdc.com.au/__data/assets/pdf_file/0027/142596/grdc_fs_spray-practical-tips_high-res-pdf.pdf.pdf`, 2014. Accessed: 2019-10-6.

[146] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, pp. 78–81, May 2016.

[147] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, pp. 30–39, January 2017.

[148] P. Gonzalez-de Santos, A. Ribeiro, C. Fernandez-Quintanilla, F. Lopez-Granados, M. Brandstoetter, S. Tomic, S. Pedrazzi, A. Peruzzi, G. Pajares, G. Kaplanis, M. Perez-Ruiz, C. Valero, J. del Cerro, M. Vieri, G. Rabatel, and B. Debilde, "Fleets of robots for environmentally-safe pest control in agriculture," *Precision Agriculture*, vol. 18, pp. 574–614, August 2017.

[149] Gartner, Inc., "Gartner hype cycle." Online: `https://www.gartner.com/en/research/methodologies/gartner-hype-cycle`. Accessed: 2019-09-08.

[150] I. Heap and S. O. Duke, "Overview of glyphosate-resistant weeds worldwide," *Pest Management Science*, vol. 74, no. 5, pp. 1040–1049, 2018.

[151] "First cancer lawsuit over weedkiller roundup filed in Australia." Online: https://www.smh.com.au/national/first-cancer-lawsuit-over-weedkiller-roundup-filed-in-australia-20190603-p51u1a.html. Accessed: 2019-09-08.

[152] M. Walker, *Hype cycle for emerging technologies, 2018.* Gartner, Inc, August 2018. Online: https://www.gartner.com/en/documents/3885468/hype-cycle-for-emerging-technologies-2018. Accessed: 2019-09-28.

[153] A. Allan, "The big benchmarking roundup." Online: https://blog.hackster.io/the-big-benchmarking-roundup-a561fbfe8719. Access: 2019-09-08.

[154] NVIDIA Corporation, "Jetson Nano: Deep Learning Inference Benchmarks." Online: https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks, 2019. Accessed: 2019-09-25.

[155] N. Corporation, "Harness AI at the Edge with the Jetson TX2 Developer Kit." Online: https://developer.nvidia.com/embedded/jetson-tx2-developer-kit. Accessed: 2019-09-08.

[156] M. Purvis, "ROS FlyCapture Camera Driver." Online: http://wiki.ros.org/pointgrey_camera_driver. Accessed: 2019-8-20.

[157] Advanced Navigation, "Spatial dual positioning system." Online: http://www.advancednavigation.com/product/spatial-dual, April 2018. Accessed: 2019-10-4.

[158] Qniemiec, "Roll-pitch-yaw angles of airplanes and other air vehicles." Online: https://en.wikipedia.org/wiki/File:RPY_angles_of_airplanes.png, July 2010. Accessed: 2019-10-4.

[159] Qniemiec, "Roll-pitch-yaw angles of cars and other land based vehicles." Online: https://en.wikipedia.org/wiki/File:RPY_angles_of_cars.png, July 2010. Accessed: 2019-10-4.

[160] C. F. F. Karney, "GeographicLib." Online: https://geographiclib.sourceforge.io/, August 2018. Accessed: 2019-10-4.

[161] Department of Agriculture and Fisheries, "Chinee apple fact sheet 2017." Online: https://www.daf.qld.gov.au/__data/assets/pdf_file/0008/52766/IPA-Chinee-Apple-PP26.pdf, 2017. Accessed: 2019-09-11.

[162] North Queensland Register, "Trial underway to control navua sedge on the atherton tablelands." Online: `https://www.northqueenslandregister.com.au/story/5291958/weed-eradication-trail-underway`. Accessed: 2019-09-11.

[163] Department of Agriculture and Fisheries, "Navua sedge fact sheet 2016." Online: `https://www.daf.qld.gov.au/__data/assets/pdf_file/0007/51010/IPA-Navua-Sedge-PP53.pdf`, 2016. Accessed: 2019-09-11.

[164] Department of Agriculture and Fisheries, "Harrisia cactus fact sheet 2017." Online: `https://www.daf.qld.gov.au/__data/assets/pdf_file/0003/49179/IPA-Harrisia-Cactus-PP22.pdf`, 2017. Accessed: 2019-09-11.

[165] North West Local Land Services, "North West Regional Strategic Weed Management Plan 2017-2022." Online: `https://northwest.lls.nsw.gov.au/__data/assets/pdf_file/0010/722917/North-West-Regional-Weed-Mgmt-Plan-web-version.pdf`, 2017. Accessed: 2019-09-11.

[166] V. Hohnke, "Resistant weeds focus of Spring Ridge field day," *Namoi Valley Independent*, July 2019. Accessed: 2019-09-11.

[167] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, p. 1137–1149, June 2017.

[168] A. M. Peerzada and B. S. Chauhan, "Chapter 2 – Thermal Weed Control: History, Mechanisms, and Impacts," in *Non-Chemical Weed Control* (K. Jabran and B. S. Chauhan, eds.), pp. 9 – 31, Academic Press, 2018.