A Methodology for Agricultural Robotics development

Simon Birrell

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Corpus Christi College
August 2022
This thesis is dedicated to the memories of Eli Zelkha and Colin Dean.
Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Simon Birrell
August 2022
Acknowledgements

This project was possible thanks to EPSRC Grant EP/L015889/1, the Royal Society – ERA Foundation Translation Award (TA160113), EPSRC Doctoral Training Program ICASE Award RG84492 (co-funded by G’s Growers), EPSRC Small Partnership Award RG86264 (in collaboration with G’s Growers) and the BBSRC Small Partnership Grant RG81275.

In addition, I am extremely grateful for the support and valuable time input from G’s Growers, in particular Charlie Kisby, John Currah, James Green and Jacob Kirwan. I would also like to thank Dr. Alex Jones from the Sainsburys Laboratory and many who have contributed to the iterations of Vegebot: especially Josie Hughes, Fumiya Iida and Julia Cai for their contributions and camaraderie, but also Luca Scimeca, Andre Rosendo, Fabio Giardina, Claudio Ravasio, Vivian Wong and Arsen Abdulali.

It felt like coming home to return to Corpus Christi College after all these years; my thanks to the college and its staff. I would like to thank my supervisor Fumiya Iida for his help, encouragement and patience; this journey is all thanks to him. It was a true pleasure to work alongside my fellow PhD students, who accepted such an atypical colleague with good grace. Thank you Josie Hughes, Luca Scimeca, Fabio Giardina, Toby Howison and Elijah Almanzor. Good luck to you all.

Finally, thanks to my sister Rachel for advice on the text, to my parents for buying me that BBC Micro and to my indulgent wife Pepa and children David and Sylvia, for putting up with me.
Abstract

Agricultural robotics is a discipline necessary for the future health and welfare of humanity, yet it is harder than suspected by those outside the field. Many challenges remain, economic as well as the technological ones of performing in unstructured, harsh environments. A general methodology for the design and development of agricultural robots is proposed using the thesis’ central work, the Vegebot lettuce-picking robot, as an example. An embodied approach to design is recommended, with development based on rapid iterations of prototypes in the field, together with deep involvement of the customer. Three key building blocks are described for automating the harvesting of the iceberg lettuce under challenging and uncertain field conditions. First, the lettuces are localised and classified using a data-driven method; this process is referred to here as Detection. Second, the end effector is moved into position under conditions of high environmental noise and uncertainty; this process is referred to here as Approach. Third, the lettuces are harvested with a custom designed end effector that incorporates a camera, pneumatics, a belt drive and a soft gripper. The general name given to this process is Manipulation. The Vegebot project is described in detail and the derived methodology is outlined.
Preface

Chapters 2–4 of this thesis are based on three peer-reviewed publications. Each publication has been edited to fit the narrative of this thesis; however, some overlap may occur. The contribution of each author is summarised at the beginning of the corresponding chapters. The publications are:


# Table of contents

List of figures xix  
List of tables xxv

1 Introduction 1  
1.1 The Challenges of Agricultural Robotics 2  
1.1.1 Structured versus unstructured environments 2  
1.1.2 A Harsh Environment 3  
1.1.3 Economics and the Developmental Process 4  
1.2 The Vegebot Project 5  
1.2.1 Background 5  
1.2.2 State of the Art 6  
1.2.3 Requirements Analysis 9  
1.2.3.1 Users 9  
1.2.3.2 Functional Requirements 9  
1.2.3.3 Non-functional Requirements 10  
1.2.3.4 Form Factor 10  
1.2.3.5 Cycle Time and Robustness 11  
1.3 Embodiment 12  
1.4 A General Approach to Building an Agricultural Robot 14  
1.4.1 Customer collaboration 14  
1.4.2 Rapid iterative development in the field 16  
1.4.3 Test multiple approaches 17  
1.4.4 Three key tracks: Detection, Approach, Manipulation 17  
1.4.5 Compensate for Seasonality 20  
1.4.6 Embodied-first solutions 20  
1.5 Contributions 21  
1.5.1 Contribution: the creation and testing of a lettuce-harvesting robot 22
| 1.5.2 | Contribution: the elaboration of a general design approach to agricultural robotics projects | 22 |
| 1.5.3 | Contribution: optimising harvesting speed under environmental fluctuations using embodied learning mechanisms | 23 |
| 1.5.4 | Contribution: use of force feedback to define the cut position on a lettuce stalk | 23 |

2 The Implementation of Vegebot

2.1 Used Materials

2.1.1 Vegebot in the Field

2.1.1.1 System Architecture

2.1.1.2 Control & Processes

2.1.1.3 Software

2.1.1.4 User Interfaces

2.1.2 Vegebot in the Lab

2.1.3 Vegebot in Simulation

2.1.4 Datasets

2.2 Application of general approach

2.2.1 Customer Collaboration: G’s Involvement

2.2.2 Rapid iterative development in the field: The Field Trips

2.2.3 Test Multiple Approaches

2.2.4 Three key tracks: Detection, Approach, Manipulation

2.2.5 Compensate for seasonality

2.2.6 Embodied-first solutions

3 Detection

3.1 Problem Statement

3.2 Application of Method

3.2.1 Customer Collaboration

3.2.2 Rapid Iterative Development in the Field

3.2.3 Compensating for Seasonality

3.2.4 Embodied First Techniques

3.3 Development Timeline

3.3.1 Gen 1: Haar-Feature Cascade Classifiers with initial dataset

3.3.2 Gen 2: Haar-Feature Cascade Classifiers with full dataset

3.3.3 Gen 3: YOLOv3, Darknet Classifier

3.3.3.1 Localisation Dataset
3.3.3.2 Classification Dataset ........................................ 51
3.3.3.3 Localisation Results ......................................... 52
3.3.3.4 Classification Results ....................................... 56
3.3.4 Detection in Simulation and Lab Work .......................... 57
3.4 Discussion ............................................................. 58
3.5 Future Work .......................................................... 59

4 Approach ................................................................. 61
4.1 Problem Statement ................................................... 61
4.2 Application of Method ............................................... 61  
  4.2.1 Rapid Iterative Development in the Field ....................... 61
  4.2.2 Compensating for Seasonality ................................ 61
  4.2.3 Test multiple approaches ..................................... 62
  4.2.4 Embodied-first solutions ..................................... 62
4.3 Development Timeline ............................................... 62  
  4.3.1 Gen 1: Open Loop Position Control .......................... 65
  4.3.2 Gen 2: Position Control with Calibration ...................... 65
  4.3.3 Gen 3: Position Control, Calibration with Visual Servoing on the End Effector ........................................ 68
    4.3.3.1 Final field tests of Gen 3 ................................ 68
  4.3.4 Gen 4: Two Stage Servoing .................................. 70
  4.3.5 Gen 5: Learned Open Loop ................................... 72
    4.3.5.1 Motivation ................................................. 72
    4.3.5.2 Method .................................................... 73
4.4 Experiments: Comparing Three Control Approaches (Gen 1, 4 and 5) ... 75
  4.4.1 Experiment One: Three different control systems ............ 75
    4.4.1.1 Experiment: ............................................... 75
    4.4.1.2 Results in simulation: .................................. 75
    4.4.1.3 Results on real robot: .................................. 77
  4.4.2 Experiment Two: Robustness to perturbations and distance .... 77
    4.4.2.1 Experiment: ............................................... 77
    4.4.2.2 Results in simulation: .................................. 78
    4.4.2.3 Results on real robot: .................................. 78
4.5 Conclusion and Contributions ....................................... 79
4.6 Future Work .......................................................... 80
# Manipulation

5.1 Problem Statement .............................................. 83
5.2 Development Timeline .......................................... 84
   5.2.1 Early cutting prototypes .................................. 84
      5.2.1.1 Rotary vs Linear End Effector Prototypes ........ 84
      5.2.1.2 Electric vs Pneumatic Linear Actuators .......... 86
   5.2.2 Integrated End Effector Prototypes ......................... 86
      5.2.2.1 Gen 1: The Bucket .................................. 86
      5.2.2.2 Gen 2: Vertical Actuators ......................... 89
      5.2.2.3 Gen 3: Cage with lever ............................. 89
      5.2.2.4 Gen 4: Steel cage ................................ 91
      5.2.2.5 Gen 5: Lightened steel cage ....................... 91
5.3 The Final End Effector ......................................... 92
   5.3.1 Force Feedback Driven Harvesting ....................... 92
      5.3.1.1 End Effector Design .............................. 92
      5.3.1.2 Force Feedback Control ......................... 94
   5.3.2 Future Challenges ...................................... 96

# Vegebot: Final field trials and experimental results

6.1 Final Field Trials and Harvesting Performance .................. 99
   6.1.1 Overall Harvesting Performance Metrics .................. 99
6.2 Future Work .................................................. 102

# Conclusions

7.1 Thesis objectives ............................................. 105
   7.1.1 The Creation and Testing of Vegebot .................... 105
   7.1.2 The elaboration of a general design methodology for agricultural robotics projects .................................................. 107
   7.1.3 To develop control methods for the "Approach" problem appropriate for the agricultural context .............................. 108
7.2 Agricultural Robotics as a Research Challenge ................. 109
   7.2.1 Challenges in Perception .............................. 109
   7.2.2 Challenges in Manipulation ........................... 110
   7.2.3 Challenges in Learning .............................. 110
7.3 Agricultural Robotics as a Business Opportunity .............. 111
   7.3.1 The Opportunities .................................. 111
   7.3.2 The Weaknesses .................................. 113
7.3.3 Business Models and Strategies .................................. 115
  7.3.3.1 What is the product? ........................................... 115
  7.3.3.2 Going to Market ............................................... 117
7.4 An Alternative View of the Industrialisation of Agriculture ............. 118
  7.4.1 Factories instead of fields ....................................... 118
  7.4.2 Structuring fields ............................................... 119
7.5 Conclusion .............................................................. 119

References ................................................................. 123
List of figures

1.1 Conventional lettuce harvesting. ............................................. 8
1.2 Video frame captures of a human worker (a) cutting through the lettuce stalk with a knife in the dominant right hand while grasping with the left, (b) lifting the head, (c) swapping the head to the dominant hand, (d) peeling off and discarding the outer leaves, (e) grasping and opening a plastic bag and (f) bagging the lettuce, ready to toss onto the carousel. The cycle time is around 6 seconds. ................................................................. 9
1.3 (a) The classical AI pipeline for autonomous agents (b) An alternative, layered view of the processes that make up an embodied agent. ............ 12
1.4 The different faculties and techniques within the detection, approach and manipulation layers, both for the Vegebot (top) and agricultural robots in general (bottom). Future work is marked with *......................... 14
1.5 A general methodology for designing and building agricultural robots. ... 15

2.1 The Vegebot harvesting system, shown undergoing field experiments. ... 26
2.2 Block diagram of the robotic lettuce harvester system developed. .......... 26
2.3 Processes for training and operation of the Vegebot, showing the key processes in green. The trajectory diagram for the lowest level pick sequence is shown in Figure 4.5 .......................................................... 29
2.4 The software architecture of Vegebot showing the structure and various packages used. ......................................................... 30
2.5 The web-based user interface for Vegebot. ................................. 31
2.6 The Dataset Gathering UI: keyboard-driven for detection, data-gathering, labeling and correction. ................................................. 32
2.7 The Vegebot simulated using CoppeliaSim. Virtual camera feeds are inset and a "virtual" lettuce is shown. The vectors shown are used in Learned Open Loop, described in section 4.3.5.2. ..................................... 34
2.8 The simulated lettuce. ......................................................... 35
2.9 The Vegebot Software Architecture: (a) on the real Vegebot (b) in simulation, with simulated cameras and IMU, plus a simulated arm tracking the output of the URSim simulated controller. ........................................ 35
2.10 The different versions of Vegebot together with the project timeline. .... 37

3.1 The different generations of Detection System. .............................. 44
3.2 The Haar-feature Cascade Classifier at work on imagery. Both false positives and negatives are present. ................................................. 46
3.3 Detection system performance under different conditions ................. 46
3.4 An early version of the User Interface (UI) being used in the field with the Haar-feature Cascade Classifier to select candidates for picking. Some false positives are clearly present, so a human operator manually selected which of the candidate bounding boxes should be used to perform the pick. ...... 47
3.5 The one-handed dataset gathering UI. The operator holds a webcam with one hand to capture datasets at different heights, and adds bounding boxes and labels with the other. .................................................. 49
3.6 The vision system pipeline showing the two stages of convolutional neural network. First, the lettuces are localised using one network. A second network using both the lettuces localised from the first network and pre-segmented lettuce images from a classification dataset is used. ......... 49
3.7 a) The requirements for successful lettuce harvesting determined by the physical end effector. The lettuce centre must be detected within a distance such that the lettuce is fully within the footprint of the end effector when cutting. b) The distribution of accuracy of the lettuce localisation system for the two different cameras used, with images from sub-datasets C and E respectively. .......................................................... 53
3.8 Localisation performance with varying brightness and image contrast. The precision and recall are given in both cases. The images below show the contrast and brightness enhancement applied to a typical image in the test dataset. ................................................................. 54
3.9 Examples of the localisation system working on different lettuce and with camera setups with different heights and angles and showing usage on different crops and different fields demonstrating robustness. Blue bounding boxes indicate the entire head of lettuce could be seen, green indicate where only part of the head is visible. ............................................. 55
3.10 Classification network performance. ............................................ 57
3.11 Simulating, or "mocking" the Detection system in integrated lab tests. Field photos are enlarged and can generally be successfully localised and classified by the Detection system, allowing other aspects of the robot, such as Approach, to be tested. .............................................................. 58

4.1 Summary of the five Approach systems developed. ........................................ 63
4.2 Control Information Flow for the five Approach systems. The sources of delay are indicated in red. ................................................................. 64
4.3 Calibration method, showing how position and camera co-ordinates are gained from 3 positions to allow a mapping from camera to real world co-ordinates to be achieved. ................................................................. 66
4.4 Visual servoing was added to Gen 3 to fine tune the position of the end effector after being positioned above the lettuce using calibrated position control. The overhead camera view is shown on the left and the end effector camera view on the right. Visual servoing modifies the trajectory of the end effector in the XY plane to move the centre of the lettuce bounding box (red) to the centre of the visual field (yellow arrow). The photo shows a test in the lab. .................................................................................................................. 69
4.5 End effector trajectories when undergoing the field experiments (Gen 3). Shows all trajectories centred on cutting (at 0 seconds) and an example representative trajectory. The vertical divisions correspond to the different stages of the Pick Sequence from Fig. 2.3 ............................................. 70
4.6 (a) The training procedure for Learned Open Loop. (b) A proposed process for incorporating online learning and exploitation of Learned Open Loop into Vegebot. ................................................................. 73
4.7 Learned Open Loop: (a) The inputs and outputs to the neural network. (b) Training the neural network over 2000 iterations. .......................... 75
4.8 Experiment One: End effector trajectories under the different control systems, (a) in simulation and (b) on the real robot. The left graphs show the end effector's trajectory in the X-Y plane with the start and end points labelled with time t. The right graphs show the end effector's trajectory along the Z axis over time t. The vertical purple lines indicate the start and end of the picking sequence. ................................................................. 76
4.9 Experiment Two: The three control systems in simulation and reality as perturbations are added (left) and the distance of the parked end effector to the lettuce is increased (right). The top graphs in each block show the final distance of the end effector centre from the lettuce. The middle graphs show the trajectory time. The lower graphs show failed picks from a total of 5 attempts.  

5.1 The initial attempt to directly develop a fully integrated two-handed solution was a failure.  

5.2 Early prototypes focused solely on the cutting problem, comparing linear and rotary electric cutting mechanisms. The plastic bucket enclosures had the effect of stabilising and gripping the lettuce head.  

5.3 Summary of the integrated end effector versions.  

5.4 The linear pneumatic prototype was the only variant that reliably cut through lettuce stalks.  

5.5 Gen 1 End Effector: The first integrated end effector used two pneumatic actuators for cutting, incorporated a webcam and a foam cushion for positioning the lettuce head.  

5.6 Gen-2 End Effector: The two cutting pneumatic actuators are mounted vertically and pull a belt that moves the blade. A horizontal pneumatic actuator moves the soft gripper to hold the lettuce in place.  

5.7 Gen 3 End Effector: Two powerful actuators drive the blade via a lever mechanism. A smaller actuator drives the soft gripper. Robustness has been improved, but not fully achieved, by using a steel top.  

5.8 Gen 4 End Effector: A single powerful actuator drives the blade via a new belt drive. The robust structure has been made entirely out of steel.  

5.9 Gen 5 End Effector: The principal change from Gen 4 is weight reduction, achieved by cutting holes in the centre steel plate.  

5.10 The final end effector developed, showing the belt drive mechanisms and dual pneumatic actuator system.  

5.11 The force feedback method, allowing a repeatable height between the ground and the knife to be achieved.  

6.1 Examples of harvested lettuce showing some with an ideal cut, others with unwanted outer leaves and damaged outer leaves.
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2</td>
<td>Distribution of the cycle times, leaves to remove and extra cuts required for the various lettuce harvesting experiments. The cycle time varies greatly depending on how far the arm needs to travel from the rest position to the lettuce.</td>
<td>101</td>
</tr>
<tr>
<td>7.1</td>
<td>The author’s son demonstrating their home Farmbot.</td>
<td>113</td>
</tr>
<tr>
<td>7.2</td>
<td>The Thorvald mobility platform.</td>
<td>115</td>
</tr>
</tbody>
</table>
List of tables

1.1 Conditions for the design and development of a lettuce harvesting system determined by the agricultural environment. . . . . . . . . . . . . . . . . . . 10
1.2 Generic metrics for an Agricultural Robot. The task might be harvesting, weeding, spraying etc. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

3.1 Details of the different sub-datasets used to create the localisation dataset including the number of lettuce and conditions in which the images were taken. 48
3.2 Classification dataset, showing the number of each type of lettuce in the dataset. 52
3.3 Overall system harvesting tests showing the localisation performance. . . . 56

4.1 Input and output variables to the Learned Open Loop neural network. . . . 74

5.1 Specification of the end effector developed . . . . . . . . . . . . . . . . . . 95

6.1 Overall system performance in the harvesting tests. Total lettuces attempted considers only lettuces within restrictions imposed by arm strength. . . . . . 100
Chapter 1

Introduction

Agriculture is a necessity, not a luxury. As it is buffeted by labour shortages, climate change, demands for safer food and newly-imposed trade barriers, the world is naturally turning to robotics to increase yield while lowering costs. Doing so appears to be a logical extension of the widespread move in the industry to mechanisation. Is a robot harvester not simply a more sophisticated combine harvester? Car production is now largely executed by robots; why not farming?

As it turns out, this is easier said than done. The problems are not simply economic (in many places in the world it is still cheaper to employ a human than use a machine), but also technological. It is not enough to make robots cheaper; robotic farming is hard to do at any price. The unstructured nature of a field is far removed from the predictable, ordered world of a car factory, and it requires hitherto unnecessary sophistication in perception, manipulation and planning. To thrive in such environments, robots will need to have more of the skills we take for granted in ourselves. The way agricultural robots detect, approach and manipulate their targets will require the solving of novel problems. To develop such robots we will need a tailored design methodology that has the reality of the field at its heart and that iterates towards the final design in deep collaboration with the end users from the farming community.

As a case study, the thesis details the process of creating and testing Vegebot, the world’s first lettuce harvesting robot. The progress from early experimental prototypes that separately tackled the problems of detection, approach and manipulation through to rapid field-tested iterations of an integrated platform is described, together with the support lab and simulation work that enabled development to continue during winter and over the pandemic. The design methodology is abstracted and its wider applicability discussed. Where the author collaborated with other team members (particularly on detection and manipulation) this is noted; the work on approach methods is entirely his own.
This thesis has three primary objectives:

• To document the creation and testing of Vegebot, of a lettuce-harvesting robot.

• The elaboration of a general design methodology for agricultural robotics projects.

• To develop control methods for the "Approach" problem appropriate for the agricultural context

To start, we will examine why agricultural robotics is hard.

1.1 The Challenges of Agricultural Robotics

To perform tasks such as seeding, crop care and harvesting, agricultural robots face challenges radically different from their siblings in automated mass-production factories. They must operate in unstructured, open environments with high variability; be robust to outdoor conditions; and be developed and customised for a bewildering of crops, pests [79] and farming conditions [20]. Solving these challenges will require multi-disciplinary teams spanning robotics, plant science and economics [112].

1.1.1 Structured versus unstructured environments

Factory robots depend upon a predictable environment and initial task configuration from which to execute [80]. The car part is always the same size, shape and weight; it is always positioned to sub-millimetre accuracy in fixtures before the robot’s movement begins [88]. In many cases, they do not require any kind of perception to perform [35]. A slight change in position of a part to be grasped is treated as an error, with the production line shut down until it is corrected. The technology required for these implementations is mature.

In warehouse picking and placing, the task is harder. Instead of a single car part, a robot must now recognise and handle thousands of different stock-keeping units (SKUs), requiring sensors and modern, deep-learning-based perception techniques [28]. The grasp positions vary with each SKU and may need to be learned [49]. The position of an SKU within a pick basket varies, and other items may need to be moved before a target can be reached. However, the overall environment is still reasonably predictable. A known and appropriate level of lighting can be guaranteed. The pick baskets can all be the same size and shape. One example of an SKU will closely resemble another in appearance and mass as they are, by their nature, mass-produced [28].

An agricultural robot has none of these certainties. Like SKUs in pick and place, plants are not pre-positioned exactly in the ground; neither are they uniform in size and shape.
All individual examples of a fruit or vegetable will vary in appearance, mass and structure [5]. While lettuces may be planted in standardised rows, the leafing on each plant will differ, changing their appearance and occluding the head [47]. Plants develop at different rates: even when their external appearance is similar, they may vary in important ways in internal structure. A lettuce head may contain voids that reduce its mass below a supermarket-acceptable threshold. A tomato vine may have been planted at a known position, but the structure and the position of the tomatoes themselves will vary from plant to plant.

On top of the variation in targets, the lighting conditions in a field vary radically both within a single day (imagine the difference between shadows at noon and dusk) and across the seasons [5, 40]. The factory robot has the luxury of controlled, optimised and constant lighting; the lettuce-picking robot does not.

All of these environmental factors complicate the perceptive, manipulative and navigational functions of agricultural robots, as well as the difficulty of making them human-safe [78] and compliant with laws and regulations [8]. Some of the difficulties caused by this variation can be mitigated by the use of precision agriculture, as will be discussed in section 7.4 of the Conclusions.

1.1.2 A Harsh Environment

Not only is the environment varied, but in many cases, it is also harsh. Some crops can be grown in sheltered polytunnels or greenhouses [102], but not all. In an exposed field, an agricultural robot is subject to wind and rain. Without an inflexibly rigid structure, strong winds may misalign cameras from known coordinates from moment to moment. Rain may spatter camera lenses and obscure the view.

Robots are subject to strong vibrations if attached to conventional farm machinery [personal communication, G Growers]. This not only requires highly robust enclosures for delicate mechanisms, but it may introduce noise into the perception and exact localisation of a target.

Even autonomous robots are subject to strong wear and tear. They travel through bumpy, muddy fields, lurching from one pot-hole to another, complicating both navigation and object detection [65]. Jolts are not occasional mishaps to be remedied by downtime, repairs or calibration, but part of everyday operation. A fleet of initially identical robots will soon drift to subtly different morphologies.

The use of protected cultivation, and the alternative path of industrialising agriculture will be discussed in section 7.4 of the conclusions.
1.1.3 Economics and the Developmental Process

The tasks required of a factory robot are predictable and can usually be handled by simple reconfiguration of the control of standard hardware. Once defined, a single control method will work for all identically constituted and positioned parts. Target parts are generally rigid and can be handled with standard grippers. Warehouse pick and place may require a wider variety of grippers, such as soft and rigid or vacuum-driven "universal" grippers. Control strategies will vary depending on the size, shape and compliance of the SKU.

Nevertheless, these problems have massive scale. Factories are the same the world over. The issues faced by pick-and-place robots are common to hundreds of thousands of warehouses. The economies of scale lead to increased development budgets for robotic solutions. Agriculture, in contrast, is highly fragmented, so tailored robotic solutions vary across the many different crops, such as radicchio [39], grape [82], cucumber [120, 119, 118], asparagus [55], sugar beet [6], broccoli [68], kiwi [104], strawberry [51], apple [7] and citrus [50]. For each crop, image datasets must be gathered (although some transfer learning may be possible [14]), and end effectors [83] and control algorithms developed.

The literature of harvesting robotics suggests that every new crop requires at least a custom end effector and control algorithm [2]. Agricultural robots need to contend with the aforementioned variability of mass, size and shape within a given crop. In addition, they must avoid picking or even touching diseased targets.

With unlimited budgets, this increased specialisation and repeated customisation could be feasible. However agriculture still needs to operate at similar razor-thin margins to factories and warehouses [3]. A potato-planting robot on the island of Jersey may have unique constraints that entail a specific solution, yet the size of that market niche is tiny [2]. The layout and terrain of lettuce fields differs between the UK and Spain [Private communication, G Growers]. Even it it were technically feasible to reconfigure all environments to be homogeneous the world over, the practical need to gradually introduce robots side by side with existing human workers further complicates it.

The constraints of small market niches combined with high variability in end effector design and control algorithms require new methods of development and careful trade-offs between reuse of standardised platforms components and customised solutions [46, 2, 11]. Public investment in Agritech R&D may be necessary to compensate for these market failures; agricultural R&D has high returns but a long gestation period [41].

Major agricultural machinery manufacturers have traditionally navigated this varied landscape by focusing on solutions to generic high-level problems, such as tractors for general mobility. Tractors solve a significant part of many different farming tasks: how to move tools close to target crops. The business opportunity to produce solutions to brand new
1.2 The Vegebot Project

1.2.1 Background

A central task of this PhD was to develop a lettuce-picking robot that could operate in real fields around Cambridgeshire. To the best of the author's knowledge, this had never before been achieved. Many of the described general challenges for agricultural robots were encountered in the project; in the majority of cases, embodied techniques provided a solution.

Iceberg lettuce is an example of a crop that is still harvested by hand using a knife (see Fig. 1.1a and Fig. 1.2). The process presents two main challenges to automation. First, visually identifying the vegetable's location and suitability for harvesting in what appears to be a sea of green leaves is hard even for humans (Figure 1.1b). Any solution must be robust to the variation in individual lettuces, with their appearance varying greatly over weather conditions, maturity and surrounding vegetation. Second, in a terrain with an uneven ground the lettuce stem must be cut cleanly at a specified height to meet commercial standards, while the lettuce head can easily be damaged by unpractised handling. A lettuce harvesting solution should therefore incorporate a high-precision, high force cutting mechanism while being capable of handling the vegetable delicately. There is a growing need for automated, robotic iceberg lettuce harvesting due to increasing uncertainty in the reliability of labour and to allow for more flexible, ‘on-demand’ harvesting of lettuce ([9]).

This project investigated automating the harvesting of iceberg lettuce with three key research goals. Firstly, how vision systems can be developed using off-the-shelf convolutional neural networks as opposed to hand-tailored computer vision pipelines, with pragmatic architectural adjustments made to allow for the datasets available. Secondly, how mechanical systems can be developed to work within the operational constraints imposed by the agricultural environment. Finally, how field robots can be developed to allow rapid integration and hence testing in the field. As the project progressed, the outlines of a generalised approach to developing agricultural robots became clearer.

Vegebot was developed using an approach of rapid iterative design, prototyping and field testing. Three key building blocks are described for automating the harvesting of the iceberg lettuce under challenging and uncertain field conditions.

First, the lettuces are localised and classified using a data-driven method; this process is referred to here as Detection and dealt with in chapter 3. The Detection system was
implemented using two convolutional neural networks, the architecture being shaped by the datasets available. Using this method in field tests, a visual-based localisation success of 91% in field tests was achieved, and the crop accurately classified.

Second, the end effector is moved into position under conditions of high environmental noise and uncertainty; this process is referred to here as Approach and described in chapter 4.

Third, the lettuces are harvested with a custom designed end effector that incorporates a camera, pneumatics, a belt drive and a soft gripper. The end effector cuts the lettuce stems efficiently while grasping the lettuce head in a way that avoids damage. As the ground is uneven and its depth hard to detect under the foliage, a force feedback control system is used to detect when the end effector has reached the correct position to make the cut and achieve a consistent cutting height. The general name given to this process is Manipulation and its development is detailed in chapter 5.

1.2.2 State of the Art

There is prior work on vision techniques for agriculture. Many of the examples in the literature available at the projects’s commencement were from before the use of convolutional neural networks (CNNs) in the late 2000s, and so used a wide variety of hand-crafted features. The detection of volunteer potato plants was performed using adaptive Bayesian classification of Canny Edge Detectors among other features [87]. Broad-leaved dock detection (a weeding task) was performed using a texture-based approach, where image tiles were subjected to a Fourier Analysis [38]. (Weeding is a similar task to harvesting, just with less concern for the fate of the extracted plant). An alternative approach to weed detection used wavelet features of Near Infrared (NIR) imagery [104], subsequently passed to a PCA component and a k-means classifier [64]. Grapes have also been detected with Canny Edge filters, using Decision Trees as the classification mechanism [10]. Foliage detection on the same project required a separate algorithm. Grapes were classified on another project using the AdaBoost framework, which combined the results of four weak classifiers into one strong one [71]. Radicchio’s have been detected by thresholding Hue Saturation Luminance (HSL) images and applying particle filters [39]. Cucumbers were detected using NIR photography at two positions 5cm apart, to give stereoscopic depth information [119] and classified for maturity by estimating their weight from the perceived volume [120]. A more recent experiment detected Broccoli heads using an RGB-D sensor had the disadvantage that the robot had to move a tent across the field to prevent interference from outdoor light. Point clouds were clustered from the depth information, outliers were removed and Viewpoint Feature Histograms constructed. A Support Vector Machine performed the actual classification of the broccoli heads [68]. The use of vision to provide control through methods including visual
1.2 The Vegebot Project

Servoing has also been shown to increase positional accuracy when harvesting citrus fruit [76, 77].

These solutions are not appropriate for iceberg lettuce. Colour cues as used in [39, 10, 31] are less useful because the lettuces appear to be part of a 'sea of green'. Depth cues, as used in [68, 94] also provide limited information because the plants and their leaves overlap and the heads are often hidden.

Similarly, there are a number of existing autonomous harvesting systems. Harvesting is a challenging task to automate and a recent review came to the gloomy conclusion that almost no progress had been made in the past thirty years [5]. Many research projects have been performed, but little has filtered through into the commercial world. The more successful projects include a harvester for apples [110] using a suction method, rice harvesting using custom harvesting systems [67] and a sweet pepper harvesting system [4]. There has also been significant work in the development of autonomous weeding or grading systems including a sugar beet classifying system [70] and a grape pruning system [15].

There are a number of patents specifically relating to the harvesting of iceberg lettuce [90, 109, 89], however, these have not been demonstrated under field conditions and do not clearly demonstrate how non-destructive selective plant harvesting is possible. [89] describes a horizontal conveyor that would be automatically loaded with freshly-cut lettuce with their cut stems pointed downwards. External leaves would apparently fall off under the influence of gravity while undersized lettuces would fall through gaps in the conveyor to be discarded. [90] describes the use of pneumatic bladders to grip the lettuce stem, while a vortex of air is blasted downwards to move away the excess outer leaves; the head is then separated, either by a sharp applied downwards or upwards movement. In both cases the core is removed from the lettuce head, rendering it inappropriate for the use case here. Finally, [109] describes a proposed end effector with blade-tipped "fingers" that would remove the outer leaves and cut the stem. The author can find no record of any of these patents being transformed into an actual device.

Other previous approaches include using a belt driven band-saw type mechanisms or water jet cutting. These approaches have limitations, most notably that the outer-leaves of the lettuce can be easily damaged when harvesting and there is a lack of reliability in stem cutting height and quality.
Introduction

(a) Human workers in front of a slowly-advancing harvesting rig.

(b) The challenging localisation and classification problem posed by the lettuce field.

Fig. 1.1 Conventional lettuce harvesting.
1.2 The Vegebot Project

Fig. 1.2 Video frame captures of a human worker (a) cutting through the lettuce stalk with a knife in the dominant right hand while grasping with the left, (b) lifting the head, (c) swapping the head to the dominant hand, (d) peeling off and discarding the outer leaves, (e) grasping and opening a plastic bag and (f) bagging the lettuce, ready to toss onto the carousel. The cycle time is around 6 seconds.

1.2.3 Requirements Analysis

1.2.3.1 Users

The customer for the lettuce picking robot would be a farming company, along the lines of a major enterprise like G Growers (sponsors of the PhD) and potentially also small holdings. It is envisioned that an operator would have the ability to manage a fleet of Vegebots that are either autonomous or fixed to a rig (see below). This necessitates the ability to manage the robot remotely.

It assumed, as with all farm machinery, that there will be a role for servicing and maintaining the robot.

1.2.3.2 Functional Requirements

The robot should be able to detect lettuces in the field, classifying them according to their suitability to be harvested. The identified classes should include at a minimum (1) harvest-ready lettuces (which may be picked immediately) (2) immature lettuces (which can be returned to later) and (3) infected lettuces, which should be reported but not be touched with the end effector, so as to avoid spreading the infection.

The robot should be able to harvest a lettuce that has been detected and classified as harvest-ready. The stalk will need to be cut through cleanly and the lettuce head lifted from the rest of the plant.

A valuable, but optional requirement is that outer leaves should be removed by the robot to confirm to supermarket standards. If this proves to be beyond the robot’s capabilities, this function can be implemented by humans (or robots), later in the production pipeline.
Table 1.1 Conditions for the design and development of a lettuce harvesting system determined by the agricultural environment.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
<th>Influence on design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Environment</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width of lettuce lanes</td>
<td>2</td>
<td>Determines width of platform</td>
</tr>
<tr>
<td>Spacing between lettuce</td>
<td>30cm</td>
<td>Determines max size of end effector</td>
</tr>
<tr>
<td>Height of lettuce plants</td>
<td>30cm</td>
<td>Determines of height of platform</td>
</tr>
<tr>
<td>Diameter of lettuce</td>
<td>20cm</td>
<td>Determines size of end effector</td>
</tr>
<tr>
<td>Diameter of lettuce stem</td>
<td>~30mm</td>
<td>Determined blade specification</td>
</tr>
<tr>
<td><strong>Robot</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generator Power</td>
<td>240V, 2kW</td>
<td>Sufficient to power all systems</td>
</tr>
<tr>
<td>Compressor Air Pressure</td>
<td>8 bar</td>
<td>Sufficient for pneumatics</td>
</tr>
<tr>
<td>Vegebot Dimensions</td>
<td>2m x 0.6m x 0.5m</td>
<td>Fits within Lettuce lanes</td>
</tr>
</tbody>
</table>

The automated removal of leaves has proven to be a challenging manipulation problem in itself [53].

The harvested lettuce should be placed at a position within the robot’s workspace where it can be fed into the next automated stage of the production pipeline. On the example harvesting rig shown in Fig. 1.1a, this is a rotating carousel.

1.2.3.3 Non-functional Requirements

There are a number of strong constraints arising for the agricultural environment, which dictate the design decisions; these are summarised in Table 1.1 (top).

Additionally, there are quality standards for the harvested lettuce, imposed by the supermarkets (but are more flexible with smaller retail outlets), that strongly influence the harvested crops commercial value.

The first is the length of remaining stem protruding from the lettuce head. This should be 1-2mm, clean and with minimal browning.

The second is that outer leaves should be removed and the remaining outer layer should be undamaged.

1.2.3.4 Form Factor

One major design decision to be taken before commercial deployment is the form factor of the finished robot. Two broad possibilities present themselves:

Vegebot could be implemented as a tool attached to an existing harvesting rig (see Fig 1.1a). In this case, it could operate alongside human workers (assuming the relevant safety regulations have been followed) and allow their incremental replacement over time, according to labour availability and cost. For this to work, the cycle time would need to be
comparable to the human worker’s, to avoid slowing down the whole rig. Even in the case of a slower cycle time, this could potentially be achieved by operating several robots in parallel on the same lane. The rig form factor has the additional benefit of leveraging the existing human and mechanical infrastructure for lettuce peeling, quality control, storage etc. Rigs are subject to vibrations, and any attached system would need to be robust to them.

Alternatively, Vegebot could be implemented as an autonomous vehicle that could be sent to the field to independently harvest a crop of lettuces. The navigation and locomotion system could be custom-designed, or more likely, use an existing solution such as Thorvald [48]. This would have the downside of a more limited storage capacity than the rig and the need to have an additional process for offloading and incorporating the harvested lettuces into the existing production pipeline. Nevertheless, it has a number of potential benefits. First, it would allow the rapid fulfillment of small on-demand orders. Supermarkets order supplies of lettuce with a very short lead time, often for the next day [G Growers, private communication], making the marshalling of human and mechanical resources difficult at such short notice, with work sometimes taking place at night. Rigs are slow to move; humans are in short supply and have conflicting priorities. By having a nimble autonomous solution, independent of the presence of human workers, such orders would be more easily fulfilled. Finally, a smaller autonomous solution could make Vegebot more accessible to small holdings that cannot afford large harvesting rigs.

The project’s customer, G Growers, were open to both possible form factors. It was jointly decided that the project should focus on the core unsolved problem of harvesting a single lettuce, which is common to both form factors and leave the final form factor decision to future work.

1.2.3.5 Cycle Time and Robustness

Ideally, the finished robot would match or improve on the human worker cycle time of 10 seconds. However, this constraint is more important in the case of the rig form factor (see above, section 1.2.3.4). Even here it may be possible to attain by the use of multiple slower Vegebots operating in parallel on a single lane. In other circumstances, the cycle time is important, but less critical. It is less important that it matches a human’s cycle time, than that it is sufficiently fast to render the automated harvesting process economically viable.

The detection system should be robust to variations in lighting over the day, weather conditions and seasons. Ideally, it would also be capable of harvesting at night.

The integrated system as a whole should be robust to likely weather conditions in the field, including wind and rain. It should continue to operate effectively as it is moved through the field, suffering knocks, blows, and in the case of rig attachment, vibrations.
1.3 Embodiment

Embodiment, or embodied cognitive intelligence, is an approach to robotics that developed in reaction to the failure of classical twentieth century AI methods to solve problems in anything but highly constrained "toy" environments. Early work by Brooks in the 1990s stated that robotics should be advanced by building complete agents that were to be tested in the real world and that these agents should be both embodied (have a body and interact dynamically with the environment) and situated (learn about the world through its own sensors and exploration) [17, 19, 18]. This perspective was broadened and deepended by Pfeiffer [93, 92], Duffy and Joue [37] and many others.

Broadly speaking, embodiment today comprises three tenets: (i) that intelligence cannot be understood as an abstract faculty in isolation from an agent's body and its interaction with the environment (ii) that the morphology, behaviour and physics of the body can solve many of the problems associated with achieving competence in complex unstructured environments and (iii) that the way to develop embodied agents is by iteratively building and testing robots in the real world.

Building robots according to embodied principles assumes that they inhabit an ecological niche, but that within that niche the environment is varied, changeable and difficult or impossible to model accurately. The classical AI pipeline (see Fig. 1.3a) of building an
accurate model of the world from perception, and using this model to plan actions, was eschewed, especially by Brooks who believed that representation itself may be unnecessary [18]. Instead, a multi-layered approach is proposed, with parallel processes separated into "layers", each with their own perception, actuation, adaptation and local goals [19]. Such robots are posited to be more robust to environmental variation and to forces and damage imposed on the robot itself.

The theoretical relevance to agricultural robotics should be clear: such robots must acquire competence in real world, unstructured environments; they must adapt to variation in their surroundings and be robust to environmental pressures and damage. It then remains to be seen how useful embodied design techniques really are in the development an agricultural robot such as Vegebot.

One clear principle of embodied design is to progress from simpler to more complex designs through testing in the real world target environment [19]. In the present case this corresponds to rapid iterative development and frequent testing in the field.

A second design principle is to resolve problems with hardware wherever possible, rather than with complex software that depends on world models of doubtful accuracy. This will be seen in the development of the vegebot’s end effector.

Rather than breaking the complex behaviour of a robot into a serial pipeline of processes, from perception through to representation, planning and finally to action (see Fig 1.3a), behaviours are decomposed into "layers", each with their own perception, action and adaptation faculties (see Fig 1.3b). These layers may operate in parallel, interacting or even competing with each other, and the emergent behaviour is what allows the robot competent and able to adapt to its environment [19].

Vegebot’s development naturally decomposed into three parallel tracks, which are here named as Detection, Approach and Manipulation (see Fig. 1.3b). They were developed side-by-side, with frequent integrated field testing of the complete robot. They each had their own perceptual, adaptive and action faculties. Not all were developed from the start in an embodied fashion: manipulation was always highly embodied and practically software-free, while the approach layer began with a more classical perception-representation-action pipeline and evolved over time towards a fully embodied implementation.

Table 1.4 (top) outlines the faculties used in Vegebot’s detection, approach and manipulation tracks, together with those proposed for future work. These will be detailed in the coming chapters. Table 1.4 (bottom) broadens the list of techniques suitable for use in a wide range of agricultural robots.
1.4 A General Approach to Building an Agricultural Robot

The general approach to agricultural robotics that this thesis recommends was developed and shaped by the above requirements of the Vegebot project and the insights provided by embodiment. The process is *customer-centric*, *iterative* and is driven by *embodied* development *in the field*. The goal is to produce a solution that works *in situ* and for the intended end-users (see Fig. 1.5).

1.4.1 Customer collaboration

The involvement of customers is a given in any agile development process, but it is more critical than usual in agricultural robotics. It is important in four ways: in making sure that the correct problem is being solved, in leveraging the vast body of existing knowledge of the farmer, in modelling his or her manual skills and understanding the social impact of the solution under development.

Collaboration is essential with farmers proposing the "what" of the problem to be solved, and researchers suggesting the "how" of potential solutions. This is not a one-off task at the start of development, but runs through the course of the project. Does harvesting a lettuce necessarily require the removal of outer leaves? The latter may be a highly complex technical challenge, when other opportunities to remove leaves are readily and cheaply available later.
A General Approach to Building an Agricultural Robot

Fig. 1.5 A general methodology for designing and building agricultural robots.
in the overall food delivery process. The customer has this knowledge and the researcher almost certainly does not. By working together with the customer throughout the project, the scope of the goal can be adapted as necessary.

This knowledge is also hugely valuable in coming up with new solutions. The stereotype of the conservative, uninquisitive farmer helped by the innovative, imaginative technologist is very far from the truth: experimentation and innovation in farming is the norm, not the exception [52]. If a problem is important, it is highly likely that it has already been tackled in multiple manual and automated ways. The farmer may know little of robotics; it is highly likely that the roboticist knows even less of farming. A casual remark that mechanical prototypes in past prototypes have been destroyed by vibrations when attached to a farming "rig" can save months of development time.

The skills being modelled by the robot are generally elaborate manual skills driven by the tacit knowledge of its practitioners. These skills are often communicated in formal training for farm workers, and by word of mouth communication [52]. At the very least, the roboticist should film and study the way manual workers go about the task (see Fig. 1.2 for an example); the problem at hand may end up being solved in a different manner by a robot, but it is key to understand the existing embodied solution.

Finally, the social impact of the project can best be understood "from the inside", by talking to the participants. Is the work being replaced undertaken by local communities for whom it has high value, or is it performed by casual migrant labour? Is the work pleasant, in the opinion of its practitioners, not the researchers? What will be the local impact of the change begin implemented? [105]

1.4.2 Rapid iterative development in the field

Researchers generally prefer to develop and test in the lab; it is more comfortable, conditions are more controllable and everything is to hand. Yet these advantages may quickly become weaknesses in the solution under development: the robot turns out to lose calibration as it moves through a bumpy field; it only functions correctly at lab temperatures; it is subtly dependent on equipment not present in the field.

A recreated "field" in the lab may differ from real world conditions to such an extent that that the problem being solved ends up being a different one: cultivars under lab conditions may display less variance in appearance; harvesting lettuces grown in shallow trays may constitute a totally different mechanical / dynamic problem to harvesting from the cold wet earth. The lab test bed is akin to a computer simulation: it may differ in important ways from reality and lead to the wrong problem being solved. In addition, the number of lab-grown cultivars may be quickly exhausted by tests.
1.4 A General Approach to Building an Agricultural Robot

It is even harder to reproduce field conditions in simulation; the structure, compliance and variability of vegetable produce and the environmental conditions do not lend themselves to accurate mathematical modelling. Simulation may help to develop for certain defined processes (see Chapter 4 for an example), but it is no substitute for the field. To quote Brooks, "The world is its own best model. It is always exactly up to date. It always contains every detail there is to be known." [17]

It is impractical to 3D print a part or write code in a field, yet the higher the frequency of field trips, the faster real world problems will be identified and solved. Lab tests and computer simulations are a useful complement to mitigate the restrictions of seasonality (and pandemics), but they can never be the primary environment for development of agricultural robotics. Field trips have the additional benefit of deepening collaboration with the customer.

1.4.3 Test multiple approaches

There are potentially many ways to harvest a lettuce. For each part of the overall problem there are multiple potential solutions. Rather than using intuition to guess the optimal solution, the proposed method suggests rapidly prototyping multiple approaches before choosing one to optimize further. The faster these "quick and dirty" early prototypes can be thrown together and tested, the better. One of the earliest field trips of the Vegebot project was to test in parallel no less than four different approaches to cutting the lettuce stalk, each with different mix of technologies and developed by a different team of students.

This approach works well because of the unstructured nature of the target environment. Again, both the field and target are extremely hard to model either mentally or in a computer even before considering the best technical approach. Testing against reality is faster.

Some of the approaches tried should be existing ones, if only to provide a baseline; it is rare that every aspect of a problem needs to be solved with innovation, and many solutions are "good enough" despite not being cutting edge. A conventional computer vision method was quickly deployed for lettuce detection and later contrasted with the then cutting-edge YOLO algorithm. Reuse of existing solutions even if not optimal can make a project targeting small commercial niches economically viable.

1.4.4 Three key tracks: Detection, Approach, Manipulation

There are different tasks in agricultural robotics: spraying, weeding, harvesting, but each can generally be broken down into at least three processes: the detection of the target, the approach of the end effector to the target and the manipulation of the target. These three key processes can to some extent be tackled separately early on in the project and are tightly
coupled to the problem at hand. In addition, there is usually a navigation process for the robot but this is less tightly coupled and the solution is more general; platforms like Thorvald (see Fig. 7.2) completely decouple navigation from the rest of the agritech solution [48]. These three developmental tracks are shown in Fig. 1.5.

**Detection** is concerned with the localisation and classification of the target fruit or vegetable. This may use standard vision, or include depth data, the non-visible spectrum and touch. Perception may be passive or active, where the viewpoint is adjusted to get a better view. The output is a list of targets, together with some positional information.

**Approach** is the seemingly simple problem of moving the end effector to the target while avoiding obstacles and keeping cycle time low. The problem is complicated by wind, uncertainty in the position of the target and any obstacles.

**Manipulation** is the heart of the task, where the end effector makes contact with the target: to harvest a lettuce, to remove a weed, to apply a pesticide or to plant a seed.

The three tracks are not wholly independent: the tolerance for the starting position of the end effector for manipulation will drive the requirements for approach; approach requirements will drive the detection system. The three tracks can also be viewed as behaviours within the finished system (see Fig. 1.3b), either running sequentially or in parallel. Manipulation follows approach, but may begin before approach has completed: the human hands shapes itself to the target object during reaching [73]. Classification of the target may be re-evaluated on the fly as the end effector approaches and more sensory information is available.

Nevertheless, decoupling these behaviours as far as possible allows them to also be developed in parallel, and helps each become more robust to changes in the others. A case in point is the handover of information from Detection to Approach. The Approach system needs a list of targets each with some positional information. If the position of the target is in 3D robot-space coordinates, then the two behaviours are tightly-coupled. If on the other hand, the positional information is simply 2D screen coordinates and sufficient merely to differentiate the targets, then the two behaviours are loosely coupled. Approach can then use its own perceptual techniques to track the target and Detection can be "mocked" in development by simply clicking a desired target on the screen.

Within each track, new versions of the system are quickly fabricated and tested in the field together with the customer. Each version should focus on solving one key problem. Once the problem is solved, even in an overspecced manner, attention can be shift to the next key problem (see Fig. 1.5). Examples of this will be provided in the following chapters.

While all three parallel tracks can be prototyped separately at the start of a project, they soon need to be integrated to form frequent iterations of the overall solution. The generic metrics for each track and for the overall system are listed in Table 1.2.
### 1.4 A General Approach to Building an Agricultural Robot

<table>
<thead>
<tr>
<th>Metric</th>
<th>Detection Approach</th>
<th>Manipulation</th>
<th>Integrated</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Localisation success</td>
<td>X</td>
<td></td>
<td>X</td>
<td>Detected qualified / Real qualified</td>
</tr>
<tr>
<td>False positive rate</td>
<td>X</td>
<td></td>
<td>X</td>
<td>False qualified / Real qualified</td>
</tr>
<tr>
<td>Classification accuracy</td>
<td>X</td>
<td></td>
<td>X</td>
<td>% Correctly classified</td>
</tr>
<tr>
<td>Detection update rate</td>
<td>X</td>
<td></td>
<td></td>
<td>Updates per second</td>
</tr>
<tr>
<td>Trajectory duration</td>
<td></td>
<td>X</td>
<td></td>
<td>From rest to manipulation position</td>
</tr>
<tr>
<td>Trajectory success %</td>
<td></td>
<td>X</td>
<td></td>
<td>Arrived within tolerance / Total</td>
</tr>
<tr>
<td>Manipulation Success %</td>
<td></td>
<td>X</td>
<td>X</td>
<td>Successfully manipulated / Detected qualified</td>
</tr>
<tr>
<td>Damage Rate %</td>
<td></td>
<td>X</td>
<td>X</td>
<td>Damaged / Total</td>
</tr>
<tr>
<td>Task Success %</td>
<td></td>
<td></td>
<td>X</td>
<td>Localisation success \times Manipulation success</td>
</tr>
<tr>
<td>Cycle time</td>
<td></td>
<td></td>
<td>X</td>
<td>Duration per task</td>
</tr>
</tbody>
</table>

Table 1.2 Generic metrics for an Agricultural Robot. The task might be harvesting, weeding, spraying etc.
1.4.5 Compensate for Seasonality

While development in the field is the ideal, there is the practical problem of the seasonality of many agricultural crops. Lettuces are only available in the field for approximately six months in each year and it is impractical to down tools over the winter. What can be done to continue development all year round? The answer varies according to the track under consideration.

To the extent that Detection can be based on pre-recorded sensory datasets (and not active perception), then software development, training and testing may easily continue all year round.

Simulations and lab tests can to some extent allow testing of different methods for Approach when the field is unavailable. Simulation can be particularly useful for rapid experimentation with learning-based approaches that require many trials; these can then be validated with lab and field work.

Manipulation is the hardest to reproduce in the lab, where physical conditions inevitably differ and simulation, where physical conditions are generally very hard to simulate accurately. Interesting new work suggests the use of sensorized physical twins to overcome this difficulty [59].

Ideally, the system hardware and software should require a minimum of modification when passing between the three environments (field, lab, simulation), to avoid the problem of maintaining three divergent versions of the project.

The problem of lack of access to the field was acute in the later part of the Vegebot project, which coincided with the COVID pandemic. It should be emphasised again that any results obtained in the lab or in simulation can only be considered provisional; the field is the ultimate yardstick of success.

1.4.6 Embodied-first solutions

For any given problem to be solved within the overall task, the first question should be: can this problem be solved with embodiment? Altering the physical body of the robot: the morphology of the end effector, the placement of the sensors, should be the first solution considered; altering the software should be the last. Why is this? After all, software is apparently easier to alter than hardware.

First, the truth is that while software can be altered rapidly, it takes a considerable amount of time to perfect. This time increases rapidly in the face of the complexity and uncertainties
of the real field. In development time, for many tasks, it is faster to fabricate a new piece of hardware than to restructure software.

Second, many perceptual and control problems can be radically simplified by the appropriate morphology. In the Vegebot, the shape of the end effector’s enclosure helps to position the lettuce in the centre of the end effector; the soft material of the gripper allows the lettuce to be grasped without damage; the force feedback of the leaves on the end effector feet provide the signal on when to cut the stalk; having one camera placed overhead and another in the end effector allow quick target selection as well as deeper, close-up classification of the lettuce.

One way to drive the search for embodied solutions is to minimise the use of software in the manipulation system. As will be detailed in chapter 5, Vegebot’s end effector was developed almost without any software at all. Where morphology and materials are not enough then low-level analog circuitry, analogous to biological reflexes, can also provide additional embodied robustness without higher-level faculties [103, 60].

Embodiment is not only a theoretical AI framework [92], but also a very practical methodology for solving robotic challenges in unstructured environments.

### 1.5 Contributions

This thesis posits that successful agricultural robotics solutions will inevitably draw lessons from agents that have already adapted to similar environments, that is to say, animals and humans. There are many ways this can be done; this section details four specific contributions in this vein. They range from immediate, practically useful techniques to more exploratory work that opens up new lines of investigation.

The four contributions are:

- The creation and testing of a lettuce-harvesting robot.
- The elaboration of a general design approach to agricultural robotics projects.
- Optimising harvesting speed under environmental fluctuations using embodied learning mechanisms.
- Use of force feedback to define the cut position on a lettuce stalk.
1.5.1 Contribution: the creation and testing of a lettuce-harvesting robot

Vegebot, the prototype for an autonomous lettuce-picking robot, is to the best of the author’s knowledge the first robot to be able to automate lettuce harvesting in real fields on crops planted for manual picking without any environmental modifications. An earlier paper [26] devised a system for growing and harvesting lettuces in a factory environment using lettuces planted in cultivators and delivered on a conveyer belt, but until Vegebot, this had not been accomplished in the field, an intrinsically harder problem. Vegebot achieved a harvest success rate of 88.2%.

Localisation of lettuces in the field is made difficult by the dense covering of leaves, crop variability and changing lighting conditions. Vegebot used a neural-network-based vision pipeline\(^1\) to both localize and classify the harvest-readiness of lettuces, despite occlusions and distracting vegetation.

Cutting the lettuce stalk neither too close nor too far from the lettuce head is key to the finished produce’s commercial acceptability. This challenge is further complicated by the presence of outer leaves obscuring visual and physical access to the stalk. Vegebot incorporates a novel end effector that pushes the leaves down and away from the head\(^2\), measuring force feedback from the plant and ground and stopping to cut when a threshold value is reached.

Vegebot incorporated embodied elements: the use of the environmental constraint of the leaves to simplify manipulation (see Section 5.2), the use of soft grippers (see Section 5.2) and the use of visual servoing to reduce a 3D perceptual problem to a 2D one (see Section 3.3) and the learning of sensorimotor coordination to accelerate harvesting.

1.5.2 Contribution: the elaboration of a general design approach to agricultural robotics projects

Vegebot was developed in close contact with farmers, taking advantage of an iterative design cycle structured around experiment and divided into three semi-independent tracks: detection, approach and manipulation. This methodology is outlined in this chapter, discussed in the context of Vegebot in Chapter 2 and broken down into the three tracks in Chapters 3, 4 and 5.

Problems are solved with physical, embodied solutions wherever possible and the restricted seasonal availability of field access is mitigated by the judicious use of lab and simulation environments.

---

\(^1\)The vision system was developed by Julia Cai under the author’s supervision.
\(^2\)The physical design of the end effector was by Josie Hughes and Fumiya Iida
1.5.3 Contribution: optimising harvesting speed under environmental fluctuations using embodied learning mechanisms

Determining where to move the end effector (Approach) to cut the lettuce in a rough environment presented its own challenges. In particular, strong winds moved camera orientation and positioning; bumps and blows to the platform frequently knocked the Vegebot components out of alignment. This made it hard to reliably map camera coordinates to robot arm coordinates: open-loop control based on these misalignments would send the end effector to the wrong position. Early versions of Vegebot required frequent recalibrations, before a more robust, closed-loop visual servoing system was put in place. This, however, produced slower trajectories (constrained by the lettuce detector update rate), worsening cycle times.

These conditions and challenges are also faced by biological agents (see section 1.1.2). The human system for reaching and grasping objects inspired a novel solution for Vegebot. First, the end effector trajectory was divided into two phases: a fast open-loop reaching motion to the correct ballpark position (the "reach"), followed by a slower, closed-loop "grasp". By optimising the first phase, the overall cycle time can be reduced with respect to a fully closed-loop solution, while retaining most of the robustness.

Before attempting a "reach", Vegebot has access to multimodal sensory data to guide it: the lettuce’s bounding box in the overhead camera, the poses of the platform and end effectors with respect to the camera (generated from Aruco boards), as well as orientation data from an IMU on the platform. During learning, valid reach targets are generated using slower robust closed-loop trajectories in both phases. A neural network is trained using the sensory data as input and the target reach positions as output. After training, target reach positions can then be inferred by the neural network from initial sensory input. This “learned open loop” method proved almost as robust as a fully closed-loop one.

By using this “learned open loop” reaching method and falling back to renewed learning whenever it fails, the robot can adjust to changes in its own body alignment and be more robust to wear and tear in the field while maintaining lower cycle times.

This contribution is detailed in Chapter 4.

1.5.4 Contribution: use of force feedback to define the cut position on a lettuce stalk

The manipulation task for Vegebot was to harvest the lettuce by cutting through the lettuce stalk. The cut position is critical for acceptance by supermarket customers: neither too close nor too far from the head is acceptable.
Vegebot uses an embodied solution, by measuring and allowing force feedback from the lettuce leaves and ground to determine the point at which the end effector should stop descending and to make the cut.

This contribution is detailed in Section 5.3.1.2.
Chapter 2

The Implementation of Vegebot

2.1 Used Materials

2.1.1 Vegebot in the Field

This section describes the version of the Vegebot that was used in the final field trials on 16/5/2018 (see Fig. 2.10).

2.1.1.1 System Architecture

The system developed for autonomous iceberg lettuce harvesting (Vegebot) is shown in Figure 2.1. Vegebot comprises a laptop computer running control software, a standard 6 degree of freedom (DOF) UR10 robot arm, two cameras and a custom end effector, all housed on a mobile platform for field testing. A block diagram showing the integration of the system is shown in Figure 2.2.

Vegebot contains two cameras: an overhead camera positioned approximately 2 meters above the ground (see Fig. 2.1) and another end effector camera mounted inside the end effector (see Fig. 5.10). Both are ordinary, low-cost USB webcams and stream video to the control laptop. Together, these allow Vegebot to detect (localise and classify) lettuces, and to move the end effector into position. There are additional sensors built into the robot arm: the standard joint encoders and a force feedback sensor that records the force and torque being applied to the end effector.

The UR10 arm provides a wide range of movements, and provides force and torque information allowing force feedback to be implemented. A commercial implementation would likely have simpler arms each with an end effector, all operating in parallel (for an example of such a system, see [104]. The control laptop controls the end effector using two
Fig. 2.1 The Vegebot harvesting system, shown undergoing field experiments.

Fig. 2.2 Block diagram of the robotic lettuce harvester system developed.
digital I/O lines routed through the UR10 arm. These switch the two pneumatic actuators on and off, the gripper actuator causing the soft gripper to grasp and release the target lettuce, while the blade actuator causes the blade to slice through the lettuce stalk and retract.

The mobile platform supports the above hardware items and is at present moved manually around the field. The system is powered by a generator, which provides sufficient power to meet the peak demands of the system. An air compressor is used to enable actuation of the pneumatic systems. The generator and compressor can sit on the Vegebot to allow the system to be completely mobile.

The software architecture is shown in Figure 2.4 and detailed in Section 2.1.1.3. The web-based user interface is shown in Figure 2.5.

2.1.1.2 Control & Processes

The processes for training and operating Vegebot can be analyzed at three levels (see Figure 2.3). At the highest level, the Learning Cycle, datasets are gathered for the initial training of the Detection vision system, harvesting is performed and additional data is gathered. As soon as enough new data is gathered to merit it, the system can be retrained. In this way, the accuracy and generalization abilities of the Vegebot can in principle be improved as images are obtained from new fields and under different weather conditions. The testing of these improvements is the subject of Chapter 3.

The Harvesting Session outlines the structure of the work in the field. First the Vegebot is moved along the lettuce lanes (seen in Figure 2.1) to bring approximately 10 lettuces within the robot’s workspace and field of view. The current iteration of Vegebot is manually pushed into position. Next, the Vegebot is optionally calibrated, using the method described in Section 4.3.2. Calibration is always performed at the start of a session and then on an as-needed basis as discrepancy between the lettuce position inferred by the overhead camera and that detected by the end effector camera increases.

Next, the vision system detects lettuces in the video feed from the overhead camera. A human then selects a lettuce by clicking on the user interface. This was a manual process during the experiments for the sake of safety. Selection could be automated with a trivial modification. The Pick Sequence then begins, with the lettuce being picked and placed onto the platform. Once the reachable lettuces have been picked, the Vegebot can either be moved to a new position or the session finished.

The Pick Sequence is fully automated and comprises seven stages. First, the end effector approaches the pre-grasp position, a point centred approximately 10cm over the inferred top of the lettuce, based on the localisation predictions from the overhead camera. Because of the rugged nature of the environment and the impacts received by the Vegebot, this prediction is
inevitably inaccurate to a greater or lesser degree. At this point, the camera in the end effector takes over to fine tune the end effector position to be directly over the centre of the lettuce. The end effector then descend vertically down over the lettuce until the force feedback sensor registers the upward force of the ground resisting the downward trajectory. The soft gripper is then activated and grasps the lettuce. Next, the blade actuator is activated and the blade moves horizontally and cuts through the lettuce stalk. Still grasping the lettuce, the end effector then lifts vertically to the same height as the pre-grasp position, clearing it from contact with the surrounding lettuces. The arm then moves the end effector to a convenient place position where the soft gripper is deactivated and the lettuce is released.

2.1.1.3 Software

The software (see Figure 2.4) was written on the Kinetic release of Robot Operating System (ROS). Custom ROS modules for Vegebot were written in Python and are bundled as the package vegebot:

- vegebot_commander – this node is responsible for receiving user commands from the web-based user interface front-end and either executing them or passing them to the appropriate node.

- lettuce_detect – this node encapsulates the code that classifies and localises lettuces from a 2D image. It calls the two deep neural networks running on Darknet.

- lettuce_sampler – this node supplies sample 2D lettuce imagery for testing purposes when not in the field.

- vegebot_msgs – this node defines the custom ROS messages used for inter-node communication, including “lettuce hypotheses”.

- vegebot_webserver – this node serves the HTML front-end user interface to the robot operator.

- vegebot_run – this module contains the 3D model of the Vegebot (in URDF format) and the scripts for launching the entirety of the software under different conditions.

Standard ROS hardware drivers (universal_robot, ur_modern and usb_cam) are used to drive the UR10 arm and the webcams. A standard installation of Darknet [95] with YOLOv3 was accelerated by CUDA drivers version 9 to provide image detection services.

1https://bitbucket.org/robotlux/vegebot/src/master/
2.1 Used Materials

Fig. 2.3 Processes for training and operation of the Vegebot, showing the key processes in green. The trajectory diagram for the lowest level pick sequence is shown in Figure 4.5.
Fig. 2.4 The software architecture of Vegebot showing the structure and various packages used.
2.1.1.4 User Interfaces

The Vegebot user interface (UI) was designed to facilitate customer involvement, drive the testing and rapid iteration of the three individual tracks (detection, approach, manipulation) and to enable seasonality compensation by working in a seamless fashion across field, lab and simulation environments.

First, the UI was implemented as a web-based UI (see Figure 2.5), streamed from the Vegebot’s integrated WiFi access point. This meant that it could be operated on the control laptop by researchers, but also shared in the field with customer representatives using wireless tablets. The UI was designed to be easily comprehensible and usable by all stakeholders. Camera feeds were displayed, overlaid with detected bounding boxes, reference points and calibration markers, all rendered using the d3.js library. Force feedback on the end effector was shown by three bar graphs to the left of the display.

Second, the testing and rapid iteration of the three development tracks was aided by the ability to control Vegebot at different levels of abstraction. An operator could simply click a lettuce and trigger the whole picking sequence (see Fig. 2.3). Equally, they could individually execute any of the processes’ sub-tasks, such as lettuce detection, the different stages of the multiple approach methods (see Fig 4.2) and execute the grasping, cutting, lifting and release
The Implementation of Vegebot

Fig. 2.6 The Dataset Gathering UI: keyboard-driven for detection, data-gathering, labeling and correction.

parts of manipulation. A 3D model of the Vegebot and detected lettuces, updated in real time, meant that an operator or observer could follow the Vegebots actions on the UI alone, without dividing attention between screen and robot.

Finally, the UI functioned identically in field, lab and simulation, helping to compensate for seasonality. In simulation, virtual camera feeds substituted for the webcams used in the field and in the lab. All sequences and sub-tasks could be triggered and monitored in the same way, the same UR10 software calculating the inverse kinematics and the execution taking place within the simulation.

An additional **Dataset Gathering UI** was designed by the author and Julia Cai, and implemented by her, for the creation of the lettuce datasets. The goal was to test the current detector, gather new visual data and to label it, all in the field together with a customer representative. This meant that bounding boxes and classification labels for classification could be more accurate, with any doubts resolved in situ by a customer domain expert.

A laptop was suspended from the researcher’s shoulders, with a UI that could be easily operated with one hand, leaving the other free to hold the webcam (see Fig. 3.5). The standard point and click method of user interaction was rejected in favour of a keyboard-only
system. The UI could be cycled with key-presses through four states: webcam streaming, image capture, localisation correction and classification correction (see Fig. 2.6).

The first stage, "Live Stream", simply displayed the real time image from the handheld webcam. On triggering image capture with a key-press, the image was frozen and could be accepted or discarded in favour of another. The latest version of the lettuce detector was triggered, generating bounding boxes and class labels. The user could either reject the image or proceed to the next state: localisation correction. Here, bounding boxes could be rejected (in the case of false positives), adjusted (in the case of an inaccurate localisation) or new ones added (in the case of false negatives). In the final stage, the detected class labels could be changed based on customer feedback and the localisation data sample added to the database. All of this could be performed with one hand, allowing fast and customer driven data gathering in the field [21].

To the best of the author’s knowledge, this is the only software of its kind, completely keyboard driven (instead of the ubiquitous point and click metaphor) and designed for use in the field.

2.1.2 Vegebot in the Lab

The lab environment for Vegebot testing was principally of use for testing the different approach methodologies and for ensuring that the three processes of detection, approach and manipulation were suitably coordinated.

Initially, only parts of the Vegebot were tested individually in the lab (see Fig. 3.11b). Later, the full platform was transported from the field to the lab (see Fig. 3.11a).

When not in the field, detection was largely performed and tested in software, using the previously gathered datasets. In the lab environment, it was "mocked" by using blow-up photos of lettuces placed on the floor (see section 3.3.4 and Fig. 3.11). In this way, approach methods could be tested without any software or user interface alterations.

With the exception of some very early experiments, where cucumbers and lettuces were sliced by different experimental cutting mechanisms, no manipulation experiments were performed in the lab. In theory, lettuces could have been grown in the lab in a tray of earth and harvested by Vegebot. In practice, soil conditions would likely be very different to the field and the number of examplars too limited by space conditions to be of any practical value.

On the other hand, experiments in approach methods were perfectly feasible in the lab; it was easy to measure trajectory duration and the precision in targeting the centre of the "mocked" lettuce, from different end effector starting points and with different distortions introduced into the Vegebot’s body (see Fig. 4.9).
2.1.3 Vegebot in Simulation

Fig. 2.7 The Vegebot simulated using CoppeliaSim. Virtual camera feeds are inset and a "virtual" lettuce is shown. The vectors shown are used in Learned Open Loop, described in section 4.3.5.2.

The purpose of building a simulated Vegebot (see Fig. 2.7) was to provide a platform for experimentation of different control and learning strategies for the Approach system, in the absence of field access during winter and over the COVID pandemic. For the learning-based experiments, the advantage over the lab experiments is that it allowed rapid generation of training sample datasets and so sped up iterative algorithm development. The simulated results were subsequently verified by equivalent physical lab experiments, but not yet, due to circumstances, by final field validation.

CoppeliaSim 4.0 was selected as a simulation platform, which provides a ROS-compatible environment with a ready-made model of the UR10 and pluggable physics engines. The Newton physics engine provided the best stability, and a model of the Vegebot platform was created using the real robot’s dimensions and weights. The simulation ran at around 10 frames per second.

Virtual cameras were used for overhead and end effector cameras. Photographs of earth from the lettuce fields were texture-mapped onto the ground. Wrapping photographs of lettuces onto spheres did not work with the real robot’s vision system; presumably the features picked up by YOLO were distorted and no longer recognisable. Instead, an overhead
2.1 Used Materials

(a) A virtual lettuce, comprising a near-transparent sphere on top of a texture-mapped circle, seen from an angle.

(b) The same, seen from the overhead camera is recognised by the vision system as a lettuce and can also be grasped.

Fig. 2.8 The simulated lettuce.

A photograph of a lettuce was mapped onto a flat circle, and this was topped with an almost transparent sphere, to give the simulated end effector something to grip onto (see Fig. 2.8a). With that, the vision system worked from the virtual cameras in the same way as on the physical robot, providing bounding boxes (see Fig. 2.8b).

There remained the problem of reproducing the arm’s movements from inverse kinematics in simulation. Previous work found the onboard UR10 controller to be more reliable than the ROS inverse kinematics package. Fortunately, Universal Robotics provide a simulation of the controller called URSim which accepts position or velocity commands and streams the position of a virtual UR10 arm. This stream is translated into ROS messages and fed to the arm model in CoppeliaSim which then tracks the controller’s virtual arm model. The software arrangement can be seen in Fig. 2.9. The bulk of the Vegebot software can be used unaltered in the two configurations.

Fig. 2.9 The Vegebot Software Architecture: (a) on the real Vegebot (b) in simulation, with simulated cameras and IMU, plus a simulated arm tracking the output of the URSim simulated controller.
2.1.4 Datasets

Only one pre-existing dataset was of use for Vegebot, gathered for a student project called "DEEPFarm" [84]. It comprised 891 high-quality images of lettuces together with their classification by the Vegebot's customer. Unfortunately, each image showed a single lettuce in the centre, making the dataset unsuitable for localisation purposes.

Additional datasets were gathered throughout the project until the final field trials. Details of these datasets are given in section 3.3.3.

2.2 Application of general approach

2.2.1 Customer Collaboration: G’s Involvement

The Vegebot project was customer driven from the start. As well as sponsoring the project (and this PhD), G Growers provided access to the fields as often as needed, as well as to executives, shareholders, labourers and managers. This enabled our research team to have a holistic view of the lettuce lifecycle as a business, economic and social process.

2.2.2 Rapid iterative development in the field: The Field Trips

In total, around 10-15 field trips took place over the springs and summers of 2016, 2017 and 2018. Key field trips and project milestones are shown in Fig. 2.10. Early on, a few researchers might drive to a field with a customer representative to test a prototype end effector, or to gather some data. As the project progressed, the full integrated platform was wheeled out on each visit by a larger team.

Field trips were generally separated by weeks when major revisions to the design were required. In other cases, new prototypes were tested after only a few days, and in one memorable occasion, the same afternoon.

Different fields were visited on each excursion according to crop readiness. G Growers work on many fields around Norfolk, Suffolk and Cambridgeshire and this variety enriched the datasets and allowed the testing of Vegebot in subtly different environments.

2.2.3 Test Multiple Approaches

At the start of the project, during the summer of 2016, a plethora of very different prototypes were rapidly built, tested and rejected in parallel. A two-handed Baxter robot that closely mimicked the manual human operations lasted for only two field trips. Linear, rotary,
Fig. 2.10 The different versions of Vegebot together with the project timeline.
pneumatic and electric actuators were all tried in different configurations. Conventional computer vision detectors were developed and tested in a matter of days.

Towards the end of the summer of 2016, the general techniques to be used for detection, approach and manipulation had been narrowed down and work began on an integrated platform. While the design used for each track continued to evolve and change, the testing and evolution of each part of the project became sequential, iterative and focused on solving one key problem at a time. From spring 2017 onwards, Vegebot was tested on each field trip as an integrated platform.

2.2.4 Three key tracks: Detection, Approach, Manipulation

There were three parallel development tracks: Detection, Approach and Manipulation, with overlapping teams working on each problem, as well as an overall integration effort. The author managed and devised detection methods (collaborating with Julia Cai on YOLO), developed all the approach methodologies, contributed the use of force feedback to Manipulation (which was driven by Josie Hughes and Fumiya Iida) and designed and managed the overall integrated platform.

The Detection track began with a conventional computer vision detector, the Cascaded HAAR classifier (Vegebot version 0.1, Fig. 2.10). This worked "well enough" to allow full platform testing with a human filtering out false positives, and was replaced in 2018 by the deep learning based YOLO detector (Vegebot version 1.6, Fig. 2.10).

The Approach track went through many iterations, starting with open loop position control based on a static model of the robot body (Vegebot version 1.0, Fig. 2.10). As the extent of model uncertainty and environmental noise became apparent, a calibration scheme was introduced with version 1.4. Visual servoing was added with 1.6 to fine tune the end effector position and this combination of position control, calibration and servoing became the method used for the final field trials in 2018. After a project hiatus, a method was sought that could reduce trajectory time while solving the difficulties of position estimation in the field. This led to Two-Stage Servoing and Learned Open Loop, tested in versions 2.1 and 2.0.

The Manipulation track began in 2016 with rapid-fire experimental cutting prototypes that tested the performance of linear vs rotary cutting motion and electric vs pneumatic actuators. By 2017, linear pneumatic actuators had been chosen, and subsequent prototypes iterated through different configurations of transmission, soft grippers and enclosures to solve one key problem after another. By the field trials 2018 (Vegebot version 1.6, end effector Gen 5), the remaining issues were end effector weight and exact vertical positioning of the cutting blade.
2.2 Application of general approach

2.2.5 Compensate for seasonality

While field trips are the ultimate measure of success or failure in such a project, the seasonality of lettuce harvesting restricts them to 6 months of the year. In addition, the second half of the project coincided with the COVID pandemic where there was no opportunity at all to access the fields. During these periods, development continued as best it could.

Detection methods were freely developed year round, using datasets gathered during the harvesting season. Approach methods could also be tested in the lab and later in simulation, with the latter allowing the easy generation of datasets for learning. End effector development was largely halted during the off season, as neither lab nor simulation testing was close enough to the reality of the field to be of much use.

2.2.6 Embodied-first solutions

The most important problem was tackled early on in 2016: how to reliably cut through a lettuce stalk. An early attempt to program a Baxter robot to wield a knife was abandoned, and different cutting mechanisms were tested and iterated on without any software at all.

Rather than an elaborate perception and positioning mechanism to get the final position of the end effector precisely over the lettuce, instead the lettuce head was nudged into position by the physical enclosure. Initially, this was a plastic bucket and later a combination of soft gripper and a foam block. As the project progressed, the approach methods themselves used less software and eliminated internal models in favour of more embodied methods such as visual servoing using the end effector’s embedded camera.

Finally, the vertical positioning of the cutting blade was set with embodied solutions rather than complex software. Adjustable feet on the bottom of the end effector interacted with the lettuce foliage and ground, providing force feedback through the arm and triggering the end of the end effector’s descent trajectory.

The more the project used physical, embodied methods, and the less it used complex software, the more robust the robot became.
Chapter 3

Detection

3.1 Problem Statement

The Detection process comprises the *localisation* of the target and its *classification*. For Vegebot\(^1\), localisation means discovery of where the lettuce is relative to the robot and classification means determining whether the lettuce is currently a suitable candidate for being harvested. Lettuces heads are variable in appearance and are typically partially or wholly occluded by their own leaves and by leaves of neighbouring lettuces. The outdoor lighting conditions also vary drastically with different weather, including very different levels of brightness and contrast. The lettuces need to be classified as "Harvest Ready" (for immediate picking), "Immature" (for picking at a later date) or "Infected" (to be avoided and reported). Additionally, the localisation system must output sufficient positional information on each target in order for the Approach system to be able to work: this may be robot-centric coordinates or simply bounding boxes in the camera viewport. All these operations must be performed in close to real time given that Vegebot uses localisation information dynamically to fine-tune the trajectory of its end effector.

3.2 Application of Method

3.2.1 Customer Collaboration

The customer, in this case G Growers, the sponsor of this PhD, were involved even before the Vegebot project began, as they had sponsored an earlier project to build a lettuce classifier [84]. In the earlier project, the basic classification scheme was defined with G Growers and  

\(^1\)The work on YOLO in this chapter is based on a project by Julia Cai, under the author’s supervision.
each target image was classified for the researchers by one of their domain experts. This
dataset was later reused for the early versions of the YOLO-based Vegebot classifier (see
section 3.3.3).

The datasets were continuously augmented during the course of the Vegebot project, with
newly gathered images classified on the spot by a domain expert working for G Growers,
while still in the field.

3.2.2 Rapid Iterative Development in the Field

The first approach attempted by the author was the use of Haars Classifiers, a standard pre-
deep learning, computer vision method. This had the advantage of being quick to implement,
as the available libraries were at that time more mature and accessible than deep-learning
alternatives.

A basic dataset of 300 lettuce images was gathered over two field trips (12/7/2016 and
20/7/2016) using a cellphone, and the first iteration of the detector was tested on a third
field trip (25/7/2016). This first detector, which localised lettuces but did not classify them,
showed a high false positive rate (119.2%), but was sufficient to allow other experiments to
proceed, providing that the final selection of a target (from the set of localised lettuces) was
manual. The dataset was further augmented on a subsequent field trip (11/10/2016), on the
hypothesis that more data would improve performance, but a limit had clearly been reached.
The detector was still in use on the first full integration of the Vegebot platform (15/5/2017).

The second attempted approach was to use one of the currently popular deep learning
algorithms. An initial selection of algorithms that could both localise and classify at high
speed yielded Fast-RCNN [45], Faster-RCNN [99] and YOLO [96–98]. the latter was
selected by the author as it reportedly had the fastest performance (albeit much slower than
HAARS), which would permit real-time feedback to the end effector control. One known
disadvantage of YOLO was poor performance with small, close-together objects, but this
was not an impediment for the Vegebot application. The YOLO project was then delegated
to a student Julia Cai, under the author’s supervision, who gathered further datasets. After
initial tests, YOLO was dedicated to pure localisation and a separate classification stage was
added to the pipeline. Iterations of the detector were developed and tested in field trips and
then the lab over the autumn and winter of 2017, and integrated and field tested in April
before the final trials (16/5/2018). On every visit to the field, the dataset was augmented.
3.2.3 Compensating for Seasonality

Lettuces are grown in fields local to Cambridge from May to October, the exact dates depending on the weather; for 6 months of the year, field trips are impractical. To compensate for this, the summer months were focused on gathering data and performing field tests, while software development and the testing of simulated lighting conditions were scheduled for the winter.

3.2.4 Embodied First Techniques

When the project started it was assumed that the output of the Detection system should include robot arm-centric lettuce 3D positions to move the end effector to. As work progressed, it became clear that the Approach system needed to use more robust, embodied methods and that the transformation of 2D screen coordinates to a 3D position was not trivial, and in some cases, not even necessary. Two Stage Servoing and Learned Open Loop don’t require a 3D position estimate for the lettuce at all.

Redefining the output of Detection system to use 2D screen coordinates decreases coupling between the systems and also enables the Approach system to be tested in isolation by clicking on a target on the screen. In hindsight, the perceptual system of the Approach system and Detection system can be designed and optimised separately, even when, as in this project, they use much of the same hardware and software.

3.3 Development Timeline

3.3.1 Gen 1: Haar-Feature Cascade Classifiers with initial dataset

*Key Problem: Lettuce localisation*

The first implemented object detector was based on the Haar-feature based cascade classifier method [121], well understood, tested in real world situations and available as part of the OpenCV library. It had the additional advantage of being one of the fastest detection mechanisms, which would allow its use in real time feedback control of the end effector during approach.

Haar features measure the difference in luminance between neighbouring rectangular regions of an image, either horizontally aligned, vertically or at 45 degrees. As an example, a key feature in face detection is the contrast between a dark horizontal rectangle over the eyes and an adjoining lighter rectangle over the cheeks just below. Classifiers built from a combination of such features can be *strong*, requiring more features, more processing
<table>
<thead>
<tr>
<th>Version</th>
<th>Algorithm</th>
<th>Field Tested</th>
<th>Key Problem Tackled</th>
<th>Method</th>
<th>Performance</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen 1</td>
<td>Haar-Feature Cascade Classifier, initial dataset</td>
<td>✓</td>
<td>Lettuce localisation</td>
<td>• Haar feature cascade classifier 10-stages</td>
<td>• Localisation success 55.6%</td>
<td>• Very high false positive rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Lettuce localization only</td>
<td>• False positive rate 119.2%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Manual false positive rejection</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Initial localization dataset of 300 images</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen 2</td>
<td>Haar-Feature Cascade Classifier, full dataset</td>
<td>✓</td>
<td>False positive reduction</td>
<td>• Haar feature cascade classifier 17-stages</td>
<td>• Localisation success 80.9%</td>
<td>• High false positive rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Lettuce localization only</td>
<td>• False positive rate 17.4%</td>
<td>• No classification</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Manual false positive rejection</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Full localization dataset of 1505 images</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen 3</td>
<td>YOLOv3, Darknet Classifier, full dataset</td>
<td>✓</td>
<td>Lettuce classification</td>
<td>• YOLOv3 performs localization</td>
<td>• Localisation success 91.0%</td>
<td>• Slow update rate (5Hz)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Localised lettuces classified by second CNN</td>
<td>• False positive rate 1.5%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Full localization dataset of 1505 images</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Full classification dataset of 665 images</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3.1 The different generations of Detection System.
3.3 Development Timeline

time and yielding greater accuracy, or weak, which use fewer features, are faster and yield an accuracy somewhat better than random. To avoid the computational burden of strong classifiers, a cascade of weak classifiers is used, each stage tuned to reduce false negatives while rejecting sub-windows of the image that very likely contain no targets. At each stage in the cascade, the classifiers have a harder classification task and require more features, but by eliminating most of the image so quickly, the average computation per region is very low. This produces a fast, real-time classifier with a tendency to produce false positives, and there are different techniques for mitigating this effect [121].

On the first two field trips (12/7/2016 and 20/7/2016), overhead videos of lettuce lanes were recorded, taken from about 1m above the ground. Frames from these videos were later labelled with bounding boxes around the lettuces. In all, 300 images were used to form a training, validation and test set. A few days later, on the third field trip (25/7/2016), the working detector, using 10 classification stages and running at 61 frames per second, was tested.

As can be seen in Figure 3.2, the accuracy is by no means perfect. Out of seven detected lettuces, four are correct and three are false positives. There are also two clear false negatives. It was however good enough for early experiments in the field, with the system making the initial detections and a human operator manually selecting the correct detected lettuces from the User Interface (UI) (see Fig. 3.4). The localisation success was 55.6% and the false positive rate was 119.2% (see Fig. 3.1 and Fig. 3.3, 1a).

3.3.2 Gen 2: Haar-Feature Cascade Classifiers with full dataset

Key Problem: False Positive Reduction

The Haars code was later retested by Julia Cai using the full dataset gathered for the YOLO detector, and increasing the number of classification stages to 17 (see Fig. 3.3, points 1b and 1c). Both localisation success and false positive rates improved, without attaining the performance of YOLO (See Fig. 3.1).

This first iteration proved prone to false positives and overly sensitive to scale: the detector would fail to work at a distance above the ground much different to 1m. All of this would later be improved by a larger and more varied dataset, but at this point a further tuning and comparison with other techniques was postponed to later in the project, in order to tackle other challenges. Most lettuces were detected successfully and false positives could be skipped with a simple UI. The detector was "good enough" for the time being.
Fig. 3.2 The Haar-feature Cascade Classifier at work on imagery. Both false positives and negatives are present.

Fig. 3.3 Detection system performance under different conditions
3.3 Development Timeline

3.3.3 Gen 3: YOLOv3, Darknet Classifier

Key problem: Lettuce classification

In principle, any of the latest deep-learning based object detectors could fulfill the Detection function\(^2\). Candidates such as YOLOv3 and Faster R-CNN \([98, 99]\) can both provide object bounding boxes and class labels in real time \([54]\). In this case, YOLOv3 was chosen by the author as it gave the fastest detection times and its principal disadvantage (poor performance on very small close-together objects) was irrelevant in this use case. Fast detection times on a laptop implied the possibility of later re-implementing the algorithm on more modest, embedded hardware.

With a large enough detection dataset, rich in examples of all lettuce categories, there would be little more to do. In the present project there were only two datasets available. The first was a detection dataset gathered by Julia Cai (see Figure 3.5 and Section 2.1.1.4), with images captured by a webcam and bounding boxes and class labels added manually using the keyboard-only user interface. This dataset (detailed in Table 3.1) was rich in positional data but the less common classes such as "Infected" were under-represented. The second dataset originated from a previous student project \([84]\) in lettuce classification and was rich in examples of all classes, but had no useful positional information, all lettuces being in the centre of each image.

\(^2\)This section incorporates work by Julia Cai, under the supervision of the author.
Table 3.1 Details of the different sub-datasets used to create the localisation dataset including the number of lettuce and conditions in which the images were taken.

<table>
<thead>
<tr>
<th>Sub Dataset</th>
<th>Number of Images</th>
<th>Number of Lettuce per Image</th>
<th>Camera Height From Ground</th>
<th>Weather Conditions</th>
<th>Image Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>157</td>
<td>7-10</td>
<td>≈1.8m</td>
<td>cloudy/sunny</td>
<td>medium</td>
</tr>
<tr>
<td>B</td>
<td>209</td>
<td>8-14</td>
<td>≈ 2m</td>
<td>sunny</td>
<td>high</td>
</tr>
<tr>
<td>C</td>
<td>117</td>
<td>3-6</td>
<td>≈1m</td>
<td>cloudy</td>
<td>medium</td>
</tr>
<tr>
<td>D</td>
<td>131</td>
<td>4-11</td>
<td>≈1.2</td>
<td>cloudy/rainy</td>
<td>low</td>
</tr>
<tr>
<td>E</td>
<td>891</td>
<td>1</td>
<td>≈0.3m</td>
<td>cloudy/sunny/rainy</td>
<td>high</td>
</tr>
</tbody>
</table>

Ideally, a more extensive detection database would have been gathered from multiple fields and stages of the crop cycle, to fully represent the position and location of exemplars of all classes. Alternatively, the existing classification images could have been inserted over other backgrounds to produce an artificial training set for detection. This latter strategy runs the risk of the network learning to detect artefacts in the synthetic images, rather than genuinely localising the vegetables based on natural visual cues.

Instead, the solution chosen by Julia Cai was to divide the pipeline into two networks (see Figure 3.6), each trained by one of the existing datasets. The first network, a YOLOv3 object detector would be used simply to discover the presence and location of lettuces (the number of classes being reduced to a single ‘lettuce’ class) and output their bounding boxes. Narrow bounding boxes, likely caused by lettuces at the edge of the viewport and out of reach of the arm, are rejected as candidates. Each of the remaining bounded boxes is then cropped (adding a small margin round the outside of the bounding box to provide more visual information to the next stage) and then a second Darknet Object Classification Network was applied to each. Finally, bounding boxes predicted by the first stage and the class labels predicted by the second stage are merged. Although requiring a two-stage network, this approach offers greater performance of both localisation and classification, at the cost of a slower update rate. The architecture has been chosen to achieve the best performance with the datasets available and given the information content of those datasets.

There is an additional advantage to using a two-stage network. Images input to YOLO are re-sized from 1920x1080 to a resolution of 320 by 320. This is still enough visual information to distinguish, say, a man from a dog, but may not be enough to determine whether one of the ten lettuces visible in the overhead camera is infected or not. By first localising the bounding boxes on the original 1920x1080 pixel image, cropping out each lettuce (approximately 260x260) and resizing them to 224x224, much more visual information on each lettuce is available for the classification network. This improves the likelihood of a correct classification on images from the overhead camera.
Fig. 3.5 The one-handed dataset gathering UI. The operator holds a webcam with one hand to capture datasets at different heights, and adds bounding boxes and labels with the other.

Fig. 3.6 The vision system pipeline showing the two stages of convolutional neural network. First, the lettuces are localised using one network. A second network using both the lettuces localised from the first network and pre-segmented lettuce images from a classification dataset is used.
Predictions on the network took 0.082s for localisation in the first stage and 0.013s classification time for each detected lettuce passed to the second stage. Assuming 10 candidate lettuces per image the total time for localisation and classification on the current hardware is approximately 0.212s, slower than a single YOLO object detection network would be, but still sufficiently fast for real-time adjustments. The end effector camera typically has only one lettuce in view during fine tuning, reducing the detection time to 0.095s. The harvesting time is somewhat longer, and thus this is not the time limiting step. The pipeline processes images from both overhead and end effector cameras. The overhead camera provides candidates for picking and the end effector camera is used to fine tune the approach of the end effector to the desired lettuce.

The two-stage network uses the existing datasets to maximum advantage and provides better classification by maintaining a higher resolution on the images of individual lettuces.

### 3.3.3.1 Localisation Dataset

Training a deep CNN object detector requires a large amount of data. The dataset also needed to be a good representation of the real scenarios the Vegebot would encounter. Since there was no existing dataset suitable for the propose of this project, a new lettuce localisation dataset was collected, labelled and assembled. Images were collected from three different sources: images taken by the overhead camera on the Vegebot platform, images taken directly with a camera and extracted images from videos taken by mobile phones and webcams. Figure 3.5 shows the process of obtaining images from the field using a webcam.

Images were divided into 5 sub-datasets (A, B, C, D and E) according to the characteristics of the images and corresponding to the different field experiments in which they were obtained. This allowed better tracking of the dataset to make sure the assembled dataset was well balanced. Figure 3.6 shows some sample images from each of the five datasets. The images cover different weather conditions, camera heights, lettuce fields, lettuce layouts, lettuce maturity and image qualities (a subjective evaluation of blurriness), since these are factors that can vary during lettuce harvesting. Table 3.1 gives a detailed overview for each subset including the number of images, number of lettuces per image, camera heights, weather conditions and image quality. Image quality refers to the subjectively evaluated blurriness of the images.

The images were labelled manually in square bounding boxes using the VoTT Visual Object Tagging Tool [122]. The lettuce images were labelled such that centre of the bounding box is the geometrical centre of the corresponding lettuce and the dimensions of the bounding box are 10% larger than the lettuce head. Only the lettuces whose heads are fully included in the image were labelled. The dataset was randomly separated into training (70%), validation
(20%), and test (10%) sets, where the validation set is used for hyperparameter tuning and the test set is only used for benchmarking the final performance.

Even though only lettuces that were fully visible within the image were labelled, the YOLO algorithm was robust enough to detect lettuces at the edges as well. Classifying these partial lettuces would have increased the complexity of the problem unnecessarily. Practically, these lettuces were likely to be out of the reach of the Vegebot robot arm and therefore they were rejected from the detected candidates. There were also cases where lettuces were blocked by weeds, the Vegebot itself or other obstacles, which led to narrow bounding boxes instead of square ones. Lettuce rejection algorithms were implemented to reject such candidates. A candidate was rejected if it met either of the following criteria:

- Rejection of non-square bounding boxes which are on the edges of the images
  \[ \frac{L}{w} > 1.15 \text{ and } d < \text{margin} \quad \text{where} \quad \text{margin} = \frac{L+W}{75} \]

- Rejection of narrow bounding boxes
  \[ \frac{l}{w} > 1.4 \]

  where \( w \) and \( l \) are the lengths of the bounding box edges, with \( w \) being the longer of the two. \( L \) and \( W \) are the width and height of the overall image, and \( d \) is the distance between the bounding box and the edge of the image.

The localisation network was based on the YOLOv3 architecture and was trained with a batch size of 64, subdivision of 8 and 10,000 iterations. The network was trained on a PC with a 4.5Ghz Intel i7-7700k CPU and an nVidia 1080Ti GeForce GTX GPU. Training took around 12 hours. Pre-trained weights based on ImageNet were used. No data augmentation was applied: this could improve localisation performance and remains for future work.

3.3.3.2 Classification Dataset

The goal of the classification network is to pick out the harvest-ready (i.e. mature and healthy) lettuces among all the lettuces recognised from the previous localisation step. Immature and infected lettuces should be left in the field. False negative localisation results can be hazardous: reaching for a non-lettuce object can damage the robot (if the object is a rock) as well as the object itself (if the object is a human hand or robot part). Adding a negative 'background' class acted as an additional filter to prevent false positives: by explicitly labelling edge cases as not being lettuces, the classification network’s performance improved.

The images were labelled by one of the authors with assistance provided by cultivation experts to allow labelling and classification of the dataset. Figure 3.6 shows sample images from each of the four classes. Table 3.2 is an overview of the size of the dataset. The
665 images were randomly separated into training (87.5%) and test (12.5%) sets. A higher portion of images were allocated to the training set deliberately due to the limitation of the images available.

Table 3.2 Classification dataset, showing the number of each type of lettuce in the dataset.

<table>
<thead>
<tr>
<th>Lettuce Class</th>
<th>Harvest Ready</th>
<th>Immature</th>
<th>Infected</th>
<th>Background</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Images</td>
<td>181</td>
<td>149</td>
<td>121</td>
<td>214</td>
<td>665</td>
</tr>
</tbody>
</table>

The classification network used was the standard Object Classifier supplied with Darknet, with no transfer learning (the use of pre-trained weights would likely increase performance further). The batch size was 64, the subdivision was 4 and the network was trained to 260 iterations. The training was on the same hardware as the localisation network and took 2 hours.

### 3.3.3.3 Localisation Results

In order for a lettuce to be successfully picked, the centre of the end effector must be placed with a tolerance, $D$, of the true centre of the lettuce. The tolerance, $D$, which is determined by the mechanical design of the end effector is approximately 2cm for average sized lettuce (approximately 15-20cm diameter). For successful harvesting, the localisation system must predict the centre of the lettuce, such that the absolute difference from the ground truth, $\Delta D$ is less than the tolerance ($\Delta D < D$). In practice, for a given camera height the threshold was specified in pixels, calculated taking into account the scale of the image. This threshold is illustrated by Figure 3.7a.

To test the ability of the system to localise lettuce heads with sufficient accuracy to allow success harvesting, images taken with both low level and high level cameras were used (approximately 30cm and 170cm above the crop respectively). The difference between the detected and ground truth of the lettuce centre was found. The distributions of the accuracy in the localisation performance of the two cameras is shown in Figure 3.7b.

In the field, the lighting and weather conditions may vary significantly. To test robustness to different lighting conditions, the test subsets of datasets A-E in Figure 3.6 were artificially modified with image processing (using ImageEnhance Brightness and ImageEnhance contrast functions in the Python Willow library) to different levels of brightness and contrast, producing 6 times (7200) the original number of test images (1200). The localisation system was then tested on this set of images (Figure 3.8). The precision and recall were then

---

3The Darknet classifier has no separate validation dataset; the experimenter chooses the length of training based on periodically evaluating against the test set. For the robustness evaluation below, fresh data was used.
3.3 Development Timeline

(a) Fig. 3.7 a) The requirements for successful lettuce harvesting determined by the physical end effector. The lettuce centre must be detected within a distance such that the lettuce is fully within the footprint of the end effector when cutting. b) The distribution of accuracy of the lettuce localisation system for the two different cameras used, with images from sub-datasets C and E respectively.
Fig. 3.8 Localisation performance with varying brightness and image contrast. The precision and recall are given in both cases. The images below show the contrast and brightness enhancement applied to a typical image in the test dataset.

found. The system showed a high robustness to changes in image brightness (the most likely changing field conditions), with minimal changes in precision and recall. For the variation in image contrast, although the precision remained high, the recall dropped significantly for high changes in contrast. It is likely that using data augmentation techniques on the original training dataset would have improved this.

Figure 3.9 shows some examples of the localisation results. Figure 3.9a, 3.9b and 3.9c show the robustness at different camera heights, different angles (3.9d) and different parts of the field (middle and edges). The system was able to avoid detecting weed (3.9a and 3.9c), human feet (3.9a and 3.9b) as well as lettuces that fail to form lettuce heads (3.9b). Figure 3.9b also shows that the lettuce rejection algorithm is able to effectively reject lettuces which are on the edge of the image. Localisation was also effective at different heights (ranging from 20cm to 170cm) and with the camera tilted by up to 45 degrees.

When integrated into the full system, the overall performance of the localisation system could be tested in harvesting trials. The success rate (number of correctly identified lettuce over total number of lettuce observed) and false positive detections were recorded. The results from this overall system results include over 60 individual lettuce harvesting experiments, where the localisation results of all lettuce that could be visible observed by the system were recorded. The results are shown in Table 3.3.
Fig. 3.9 Examples of the localisation system working on different lettuce and with camera setups with different heights and angles and showing usage on different crops and different fields demonstrating robustness. Blue bounding boxes indicate the entire head of lettuce could be seen, green indicate where only part of the head is visible.
Table 3.3 Overall system harvesting tests showing the localisation performance.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Result</th>
<th>STD</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lettuce Localisation Success</td>
<td>91.0%</td>
<td>12.7%</td>
<td>Number of detected qualified (\frac{\text{Number of detected qualified}}{\text{Number of real qualified}})</td>
</tr>
<tr>
<td>False Positive Detection</td>
<td>1.5%</td>
<td>4.3%</td>
<td>Number of false qualified (\frac{\text{Number of false qualified}}{\text{Number of real qualified}})</td>
</tr>
</tbody>
</table>

### 3.3.3.4 Classification Results

Robustness and accuracy of the classification system is critical for avoiding infected or damaged crops which could infect the harvesting system. By skipping immature heads and avoiding unnecessary harvesting the efficiency of the harvester can be maximised. To test the robustness of the system, the same images from the localisation experiments (modified for brightness and contrast) were passed to the classification network and the accuracy recorded. The results are shown in Figure 3.10a. For classification, the network showed greatest robustness to contrast as opposed to brightness variations; this could be because the training data showed greater variation in contrast as opposed to brightness. Images taken in bright sunlight were high contrast rather than high brightness and there were no late-night images in the dataset to train for low brightness. Judicious data augmentation before training should improve performance.

To understand the classification decisions made by the network a confusion matrix of the field tests has been generated and is shown in Figure 3.10b. The diagonal shows the correctly classified lettuce, showing that the classification performs adequately for identifying background, infected and harvest ready lettuce. Identifying infected lettuce is crucial for avoiding contamination and further work should be undertaken to further improve the classification.

The network struggles to separate harvest ready and immature lettuces. One of the reasons is that the boundary between harvest ready and immature lettuces is very vague and changes accordingly to current market requirements, and thus creating a meaningful dataset is challenging. The classification dataset was labelled under the rules that a 'harvest-ready' lettuce head is around 18cm in diameter, which for the majority of the time is the harvesting requirement. On the day of the field test, there was a change in harvesting specification: lettuces that would normally be treated as 'immature’ and left in the field were also harvested, which explains why many of the ‘immature’ predictions got corrected to ‘harvest-ready’.

When entire system tests of the Vegebot were later ran in the field, the system provide 100% accuracy when classifying lettuce. Although a reasonable number of experiments were
3.3 Development Timeline

(a) Accuracy of the classification network with changes in image brightness and image contrast.

(b) The Confusion matrix showing the classification performance of lettuce.

Fig. 3.10 Classification network performance.

ran (69), the number of non-ideal (i.e. diseased or immature) lettuce in this experiment was low, so there was little variation in the classification of lettuce seen.

3.3.4 Detection in Simulation and Lab Work

The problem of seasonality means that field trials are only available for six months each year. During the latter part of the Vegebot project there was the additional impediment of COVID lockdowns. To continue development over such periods, alternative methods must be found. For most agricultural robotics projects, these will be lab work or simulation.

No integrated solution can be considered valid without *in situ* trials. The lab is a very different environment from the field; the variations in lighting, available exemplars, neighbouring foliage are extremely hard to reproduce. Simulation is generally even further removed from reality. Nevertheless, if sufficient photographic and visual data has already been gathered, then network training, optimisation and testing can proceed.

Additionally, it is useful to be able to "mock" the Detection system to allow development of Approach to continue and be tested as part of the integrated system, either in the lab or in simulation. In the lab, field photographs were enlarged and placed on the floor. Isolated lettuce photos (see Fig. 3.11a) were prone to false positives in the floor background, which was not present in the training set. Full background photos (see Fig. 3.11b) worked better.

Similarly, during testing of Approach methods in simulation, virtual lettuces were used to allow the Detection system to work without alteration (see Fig. 2.8 and section 2.1.3).
Fig. 3.11 Simulating, or "mocking" the Detection system in integrated lab tests. Field photos are enlarged and can generally be successfully localised and classified by the Detection system, allowing other aspects of the robot, such as Approach, to be tested.

3.4 Discussion

The gathering, together with the customer, correction and labelling of data in the field where ambiguities can quickly and easily be resolved is a widely applicable principle. The use of a keyboard-only UI facilitates this and should be broadly useful.

The use of convolutional neural networks was new at the start of the Vegebot project, but has since become standard practise. YOLO was chosen on the basis of a fast update rate, and its principal disadvantage, the poor localisation of small, close-together targets, was irrelevant given the layout of lettuces in the field. This would not be the case with all crops: hanging fruit like strawberries would overlap in the visual field and detection would be harder, likely requiring camera movement to disambiguate.

The use of a two-stage pipeline (a contribution by Julia Cai) was originally intended as a temporary fix for the problem of unbalanced datasets. Nevertheless, it has wider applicability to other projects given its potential to increase classification performance by sending higher-resolution images to the classification network, at the cost of a slower update rate.

In general, the "Approach" perceptual problem of tracking the targets during movement (which can require a high update rate) should in future projects be separated from the
"Detection" perceptual problem of generating a list of targets together with some positional information (which does not). The two systems may share hardware, some software and initial datasets, but they are subtly different problems.

3.5 Future Work

While the detection system implemented on Vegebot fulfilled its purpose for the field trials, more work remains to be done in three areas: online data gathering, harvest readiness testing and disease detection.

During development, datasets of lettuce photos were collected and labelled manually in preparation for the trials. Once Vegebot prototypes are in more frequent use, online data gathering would allow the datasets to vastly augmented. Each fleet member could transmit webcam footage to a central server (buffering on a local hard disk to allow for intermittent connectivity) for later labelling. This footage, useful in both supervised and unsupervised learning, should provide a greater variety of lighting, lettuce strains and field types and enable better localisation and classification. In addition, failed picks signalled by a human operator or automated processes, could be investigated and corrected by the development team.

It has been noted above that the definition of "harvest readiness" was flexible and changed according to market and crop conditions. It was largely determined by the diameter of the lettuce head. This suggests that it should be captured by a separate metric and measurement process, and not embedded into the neural network itself. An RGBD camera with depth sensor could allow the lettuce head diameter to be measured independently of the main classifier, and the threshold diameter varied as needed without any retraining.

Finally, disease detection could potentially be improved by the use of an ultraviolet light and spectrometer in an enclosed bucket. Experiments early in the project, in collaboration with Dr. Alexander Jones, suggested that mildew could be detected by bathing the lettuce head in ultraviolet light and measuring the frequency response with multispectral analysis.
Chapter 4

Approach

4.1 Problem Statement

Given a detected lettuce, localised and classified as suitable for harvesting, how should Vegebot move the end effector into position for subsequent manipulation? A solution to the Approach problem should be robust to the harsh environment of the field, to wear and tear on the robot itself and the consequent inaccuracies of its self-model. The solution should minimise trajectory time, while preserving accuracy and reliability in arriving at the target, as the trajectory duration is the largest component of overall harvesting cycle time.

4.2 Application of Method

4.2.1 Rapid Iterative Development in the Field

The Approach problem was tackled from the first integrated version of the Vegebot (version 1.0, Fig. 2.10). In all field trips through the summer of 2017 and to the final field trials in May 2018, different approach strategies were implemented, tested, improved and discarded. Methods developed and tested successfully in the lab were quickly discovered to be inadequate in the rugged environment of a lettuce field.

4.2.2 Compensating for Seasonality

After a project hiatus and unlike the detection and manipulation tracks, the development of the approach system continued in the lab and simulation during winter and the COVID pandemic. The final methods developed remain to be tested in the field in future work, for ultimate validation.
4.2.3 **Test multiple approaches**

A variety of methods were implemented and tested over the course of the project, from classic static models and position control, through feedback driven visual servoing and later to learning based methods. In all, five generations of Approach control systems were developed (see section 4.3 below).

4.2.4 **Embodied-first solutions**

In this project, the embodied solutions were arrived at by a process of elimination, rather than being the starting point. The first two generations of approach control relied on static models and position control. Only with the third generation, which added visual servoing to fine tune the end effector’s position, did the system become reliable in the field. Visual servoing eliminates the need for an internal model, relying purely on visual feedback from the body.

Using exclusively visual servoing and completely eliminating all internal models was later shown in the lab to be the most robust method, but also the slowest. By adding a control system that let the Vegebot practise with a slow, robust method and then switch to a faster open loop method once its own body image had been learned, was demonstrated to be a good compromise between robustness and speed. This method still awaits final field testing.

In general, the simplest solutions driven by direct visual feedback proved superior to those that assumed their knowledge of the external world to be accurate.

4.3 **Development Timeline**

Five generations of Approach systems were developed during the course of the Vegebot project. These are described in this section, summarised in Fig. 4.1 and their information flows detailed in Fig. 4.2.

Each method has two sequential stages (Fig 4.2) and begins with the **Overhead Camera** feeding a stream of 640x480 images to the **Lettuce Detector**, which generates one or more bounding boxes $BB_o$. Once the vision system has detected the bounding box of a target lettuce, the end effector must be moved to a "pre-grasp" position, a short vertical distance above a selected target, and then brought down to the ground to envelope, grip and cut it. The first three generations of approach system were tested in the field. Later, three of the five generations were compared in simulation and on the real Vegebot in the lab.
### 4.3 Development Timeline

<table>
<thead>
<tr>
<th>Version</th>
<th>Algorithm</th>
<th>Field Tested</th>
<th>Internal Representation</th>
<th>Method</th>
<th>Weaknesses</th>
</tr>
</thead>
</table>
| Gen 1 15/5/2017 | Open-loop Position Control | ✓ | 3D Static Projection Model | • Bounding box to 3D lettuce coordinates  
• Position control | • Highly vulnerable to projection model inaccuracies |
| Gen 2 19/10/2017 | Open-loop with calibration | ✓ | 2D Static Projection Model generated by calibration | • Bounding box to 3D position in calibration plane  
• Position control | • Frequent recalibration takes time  
• Pre-grasp often still mis-positioned |
| Gen 3 16/5/2018 | Open-loop with calibration and visual servoing | ✓ | 2D Static Projection Model generated by calibration | • Bounding box to 3D position in calibration plane  
• Position control to pre-grasp  
• Visual servoing to fine tune grasp | • Frequent recalibration takes time |
| Gen 4 18/12/2019 | Two Stage Servoing | | None: Dynamic, closed-loop tracking used instead | • Visual servoing to pre-grasp  
• Visual servoing to fine tune grasp | • Slower than previous methods  
• Limited by detector update rate |
| Gen 5 24/3/2020 | Learned Open Loop | | Sensorimotor coordination + dynamic closed-loop tracking | • Learning from Two Stage Servoing associates bounding box with target 3D position  
• Position control to pre-grasp  
• Visual servoing to fine tune grasp | |

Fig. 4.1 Summary of the five Approach systems developed.
Fig. 4.2 Control Information Flow for the five Approach systems. The sources of delay are indicated in red.
4.3 Development Timeline

4.3.1 Gen 1: Open Loop Position Control

*Key Problem: Reaching pre-grasp position*

The first method tested on the approach problem (Vegebot version 1.0, see Fig. 2.10) was the classic one of modelling the robot and its coordinate systems, calibrating the camera parameters and then transforming the target centre pixel of the lettuce (the centre of the bounding box) to a position in 3D space and finally using inverse kinematics to move the arm as required. This is shown in the first column of Fig. 4.2.

In the first of the systems, **Open Loop** control, a static **Projection Model** transforms the bounding box coordinates $BB_o$ into an estimated 3D location in the robot arm space $T$. This model derives $T$ by using knowledge of the camera’s geometry to project a vector from the camera through the centre of the bounding box to where it intersects with an estimated ground plane. The system then calculates the **Pre-Grasp Position** $T^*$, which is some 20cm vertically above $T$.

In Stage 1 of the movement, the end effector is moved using inverse kinematics to the **Pre-grasp position** $T^*$, and then in Stage 2 drops down to $T$ so that the lettuce is left in the centre of the end effector cage. This **End Effector Inverse Kinematics Control** is executed by the UR10’s controller (or URSim in simulation). The resulting movements are fast, but not robust to noise or to inaccuracies in the described projection model.

The problem encountered was that the system worked well in the lab, but would fail once subjected to knocks and bumps in the field. Even small deviations in the ground truth position of the overhead camera compared with the internal model, would mean that the robot might incorrectly locate its target by up to 10cm.

While the Vegebot’s overhead camera could have been made more rigid in its positioning, G Growers, our agricultural customer, showed us that any equipment attached to their harvesting rig (a very likely commercial form factor) would be subjected to far more shocks, vibrations and wear and tear than our stationary experimental platform. This led the author to augment position control with calibration.

4.3.2 Gen 2: Position Control with Calibration

*Key Problem: Compensate for model inaccuracies*

In the next method developed (shown in the second column of Fig. 4.2), the robot could self-calibrate the transformation from viewport pixels to arm position, using Aruco markers positioned on the top of the end effector. An occasional self-calibration was sufficient to reset the transformation, for example after moving the platform. Calibration also reset the target location of the lettuce centre within the viewport of the end effector camera. It was
assumed that the platform was kept approximately level with respect to the field, thanks to the tracks in which Vegebot moves.

The full calibration procedure (summarized in Figure 4.3) was as follows:

1. Manually position the end effector over any lettuce X using standard UR10 controls.
2. Manually raise the end effector vertically until approximately 10cm clear of the lettuce.
3. Trigger automatic calibration:
   (a) The centre pixel of the bounding box for lettuce X in the end effector camera is recorded as the target centre pixel for fine tuning (the camera is not exactly centred in the end effector for space reasons)
   (b) The calibration records the vertical position of the end effector (Z axis in ROS) and assumes this to be the height of the plane containing all future "pre-grasp" positions.
   (c) The end effector then moves to three positions at the edges of the viewport, in the same horizontal plane. Each position is recorded in terms of the X,Y,Z of the end effector in the robot arm’s coordinate frame and in terms of the u,v centre pixel of the detected Aruco marker.

The three calibration positions define a horizontal plane with respect to the ground, around 10cm over the tops of the lettuces. Given any pixel u,v in the viewport, the corresponding
x, y, z in the horizontal plane can be found by linear interpolation between these three points. The UR10’s built-in inverse kinematics were then used to move the end effector into position in the "Approach pre-grasp position" phase of the Pick Sequence (see Figure 2.3).

Three calibration points in robot arm space (see Fig. 4.3) are found \((P_1, P_2, P_3)\) and their equivalent viewpoint co-ordinate are found in the camera space \((C_1, C_2, C_3)\). Any viewpoint co-ordinate, \(C_t (u_t, v_t)\) can be expressed as the sum of two vectors:

\[
C_t = aC_2 + bC_3 \quad \text{where} \quad C_2 = C_2 - C_1
\]
\[
C_3 = C_3 - C_1
\]
\[
C_t = C_t - C_1
\]

The values of \(a\) and \(b\) can be found as:

\[
b = \frac{\bar{v}_t - \bar{u}_t\bar{v}_2}{\bar{u}_2}
\]

\[
a = \frac{\bar{v}_3 - b\bar{u}_3}{\bar{u}_2}
\]

This allows an equivalent point in robot space to be found:

\[
\tilde{P}_t = P_t - P_1
\]
\[
= a\tilde{P}_2 + b\tilde{P}_3
\]

Such that the point \(P_t\) in robot space can be calculated by:

\[
P_t = P_1 + a\tilde{P}_2 + b\tilde{P}_3
\]

While the full calibration sequence involved human input to position the end effector over a sample lettuce, the re-sampling of the horizontal plane itself was automatic and could be triggered without human intervention on an as-needed basis.
4.3.3  Gen 3: Position Control, Calibration with Visual Servoing on the End Effector

**Key Problem: Fine-tune pre-grasp end effector position**

In the version of the approach control system used in the final field trials, shown in the third column of Fig. 4.2, visual servoing was added in the descent phase to fine-tune the position of the end effector.

The calibrated positioning method proved robust enough to move the end effector into the pre-grasp position, but not to exactly centre it accurately over the top of the lettuce. At this point, the end effector "fine tunes" the position using a simple visual servoing method. The bounding box of the target lettuce is now visible in the end effector video feed (see Figure 4.4, the centre point is calculated and then the arm is moved in the horizontal plane (along the X and Y axes) until this centre point is close to the target pixel recorded in step 3a of the calibration sequence. The end effector is now positioned over the centre of the target lettuce and can then descend vertically.

Additionally, the time taken to fine tune the end effector position after pre-grasp reflects the inaccuracy of the calibrated projection model, and could potentially be used as a cue to automatically trigger the recalibration process. For Gen 3, the calibration procedure was always undertaken when the Vegebot was positioned at the start of a lettuce lane. When the platform was manually moved between harvesting sessions, there was an additional human decision (see Figure 2.3) on whether to recalibrate, if for example the change in terrain had caused the relative position of the platform to the field to change.

This combination of position control, calibration and visual servoing for fine tuning gave positive results in the field trials and were sufficient to declare the basic Approach problem solved.

The field trials of this version are detailed in the following section.

4.3.3.1  Final field tests of Gen 3

During the final field experiments on 16th May 2018, 69 qualified lettuces were detected by the vision system. Of these, attempts were made to pick 60, the remainder being out of range of the robot arm. 31 pick attempts were successful, with 29 failures, almost entirely due to the weight of the end effector causing mechanical failures on the arm which made attempting harvesting impossible.

The 31 successful trajectories of the end effector are shown in grey in Figure 4.5, with a representative trajectory highlighted in black. This representative trajectory shows a single experiment which reflects the desired trajectory and demonstrates the different parts of the
harvesting process. The breakdown of the time series into the processes from Figure 2.3 is shown. The X, Y and Z coordinates are shown with respect to the base of robot platform, with X pointing forwards in the direction of travel, Y pointing to the left and Z pointing up.

With the exception of the Grasp-Cut section, all of the other trajectory sections were slowed considerably by the burden of the end effector weight on the robot arm. This led to an average cycle time of 31.7s. Critically, the rate limiting step, the grasping and cutting, required only 2 seconds. Thus, using a lighter end effector, for example constructed from a lighter material such as carbon fibre, or using a stronger arm could lead to a significantly lower cycle time.

The trajectories also clearly show the impact of the force feedback (detailed in Chapter 5), with the robot arm descending in the Z axis at a consistent rate until the force threshold is met. This shows that the end height of arm varies considerably for different lettuce, showing how using force feedback allows a consistent height to be achieved. There is also slight variability in the X and Y axis close to when the force threshold is reached as the end effector self levels on the ground.

The problems that remained with this approach system were a long trajectory duration (leading to a long overall cycle time) and the time required to periodically recalibrate the
Fig. 4.5 End effector trajectories when undergoing the field experiments (Gen 3). Shows all trajectories centred on cutting (at 0 seconds) and an example representative trajectory. The vertical divisions correspond to the different stages of the Pick Sequence from Fig. 2.3.

positioning system each time the Vegebot moved. Solutions to these issues were sought in subsequent iterations, tested, due to circumstances exclusively in the lab and in simulation.

### 4.3.4 Gen 4: Two Stage Servoing

**Key Problem: Robustness to harsh environment**

After the final field trials, the remaining approach challenges were to reduce or eliminate the need for calibration and to lower the overall cycle time.

Visual servoing had successfully compensated for minor positioning errors in the previous generation. A new method, detailed in the fourth column of Fig. 4.2, was devised that eliminated completely the internal static model, the calibration and used visual servoing also for the initial placement of the end effector in the pre-grasp position.

Visual servoing has been extensively studied; see [25] for a review and [77] for an example application in harvesting. The use of multiple cameras for visual servoing along with authority switching algorithms has also been explored, for example in [32], which uses distance derived from an RGBD sensor to determine the change of authority. RGBD sensors are problematic outdoors, so Vegebot simply uses the presence or absence of the target in the end effector camera to switch authority.
This method proved to be robust, eliminating the need for calibration, but was slower than other methods. Visual servoing is used to continually adjust the velocity $v_{xyz}$ in robot arm space of the end effector to approach the visual target. The **Lettuce Detector** provides a bounding box $BB_o$ for the lettuce as before, but the overhead camera image stream is also passed to an **Aruco Board Detector (EE)** which returns a Bounding Box $BB_a$ for the top of the end effector.

An offset $d_{uv}$ between the centres of the two bounding boxes $BB_o$ and $BB_a$ is calculated in normalized pixel coordinates, and this is used to derive the overall desired end effector velocity $v_{xyz}$ in the module **Calculate Offset in Overhead Camera**:

$$
\begin{align*}
  v_{xy} &= -\alpha_{cam} d_{uv} \\
  v_z &= -\beta_{cam} [(z - z_{target}) + \frac{\gamma_{cam}}{|d_{uv}| + 0.1}]
\end{align*}
$$

(4.6) \quad (4.7)

where $z$ is the $z$ position of the end effector in robot arm space, $z_{target}$ is the estimated $z$ position of the ground and $\alpha_{cam}$, $\beta_{ca}$ and $\gamma_{cam}$ are constants that are defined separately for each camera. The equation for $v_z$ slows the speed of descent as the target is approached, as well as further reducing it if a large correction in the XY plane is needed. The resulting velocity commands $v_{xyz}$ are passed to the UR10 Controller (or URSim) in the module **End Effector Velocity Control**.

The overhead camera retains authority until the lettuce becomes visible in the end effector camera, at which point Stage 2 commences. Because the lettuces are predictably spaced apart, at the standard pre-grasp height there is no risk of parts of two lettuces appearing simultaneously in the end effector camera and confusing the robot over which target to approach. It also doesn’t matter if only part of the lettuce is visible initially; because of its compact shape and size the lettuce is guaranteed to eventually fit completely within the viewport and so the above equations will always reliably centre the lettuce. This changeover technique is, to the best of the author’s knowledge, a novel contribution, appropriate for similarly shaped, spaced and sized targets.

Authority is then switched to the **End Effector Camera** the output of which is used by the **Lettuce Detector** to derive a new, close-up bounding box $BB_{ee}$. The offset $d_{uv}$ is now the offset of the centre of $BB_{ee}$ from the centre of the visual field and the module **Calculate Lettuce Offset from Centre** derives the desired velocity $v_{xyz}$ using the same equations (4.6) and (4.7) but with different constants for $\alpha_{cam}$, $\beta_{cam}$ and $\gamma_{cam}$. The end effector continues to approach the centre of the lettuce while now descending faster; given that the lettuce is close in the XY place, there is less chance of missing the target by moving down too quickly.
The Two Stage Servoing approach is limited in velocity by the speed of updates from the perceptual system (around 5Hz on the current hardware) but is more robust to environmental noise and changes in body shape: as the Vegebot receives blows and its morphology alters, the trajectory, within limits, can self-correct. Pending validation in the field, this was the most robust method, but also the slowest.

4.3.5 Gen 5: Learned Open Loop

**Key Problem: Reduce cycle time**

4.3.5.1 Motivation

The feedback-driven trajectory to the pre-grasp position in Two Stage Servoing was slower than simply picking a target position and moving there using inverse kinematics. The velocity was limited by the lettuce detector’s update rate, which was in turn limited by the available GPU hardware. In the field, the final descent from the pre-grasp position to the ground was consistently found to require closed-loop corrections. Nevertheless, it seemed feasible that the pre-grasp position could be estimated without the need for feedback or constant recalibration.

Inspired by the two stages of human grasping: a fast, ballistic open look reach followed by a slower, feedback-driven grasping motion [73, 62], it was decided to revert to open loop motion to reach the pre-grasp position, but to learn this position from sensory data and successful trials gathered using the slower, more robust Two Stage Servoing method (see Fig. 4.6a). The trajectory is then generated in the manner shown in the fifth column of Fig. 4.2.

By learning target arm positions from successful closed-loop trials, there is no need to have access to a pre-existing 3D projection model; the system effectively learns the sensorimotor coordination necessary for the task at hand in an embodied fashion.

It also opens the possibility of adaptively falling back to the more robust Two Stage Servoing method whenever the open loop method degrades in accuracy or fails completely. The system could then be retrained online with newly gathered data from successful Two Stage Servoing trials (see Fig. 4.6b). In contrast to the periodic recalibration of Gen 3, there would be no wasted harvesting time: each approach trajectory, whether Learned Open Loop or Two Stage Servoing, would be productive.

The learning function proposed uses multi-sensory input to a neural network. In principle, if damage to the robot has simply shifted the camera position, the coordinate transformation between camera and robot arm could be derived in a closed form fashion or by a least
squares method. On the other hand, the effects of damage and environmental noise may be non-linear: wind causes the overhead camera support to oscillate; vibrations from a harvesting rig may have an even stronger effect. A loose camera would show a different coordinate transformation depending on the inclination of the platform; the neural network of the Learned Open Loop method includes an IMU input to allow this to be learned. The embodied nature of the neural network input provides flexibility to different types of damage.

This method potentially had the advantage of being more robust than Gen 1, eliminating the calibration of Gen 2 while maintaining its trajectory speed and preserved the successful fine-tuning of Gen 3.

Fig. 4.6 (a) The training procedure for Learned Open Loop. (b) A proposed process for incorporating online learning and exploitation of Learned Open Loop into Vegebot.

4.3.5.2 Method

As well as the Lettuce Detector which generates bounding box $BB_o$, the Aruco Detector (EE) now generates an estimated pose $P_a$ for the end effector, and an additional Aruco Detector (Platform) derives an estimated pose $P_v$ for the platform using the second Aruco Board, both poses being estimated in overhead camera space (see Fig. 2.7). This is combined with the absolute orientation $R_v$ of the platform, measured by an IMU as a quaternion in world space, to form a multi-modal sensory input vector to the neural network (see Table 4.1 and Fig. 4.7a).

The Neural Network accepts $BB_o$, $P_a$, $P_v$ and $R_v$ as inputs and generates position $T$ as an output, estimating the lettuce pose in robot arm space using the available sensory data (equation 4.8). $T^*$ is derived in the usual way, begin 20cm above $T$. These variables are
Table 4.1 Input and output variables to the Learned Open Loop neural network.

<table>
<thead>
<tr>
<th>Var.</th>
<th>Description</th>
<th>Units</th>
<th>Frame of reference</th>
<th>Dim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_a$</td>
<td>Pose of end effector Aruco board</td>
<td>Position $(x, y, z)$ Orientation $(x, y, z, w)$</td>
<td>Overhead camera</td>
<td>7</td>
</tr>
<tr>
<td>$P_v$</td>
<td>Pose of platform Aruco board</td>
<td>Position $(x, y, z)$ Orientation $(x, y, z, w)$</td>
<td>Overhead camera</td>
<td>7</td>
</tr>
<tr>
<td>$R_v$</td>
<td>IMU orientation</td>
<td>Orientation $(x, y, z, w)$</td>
<td>World</td>
<td>4</td>
</tr>
<tr>
<td>$BB_o$</td>
<td>Bounding box of lettuce in overhead camera</td>
<td>Centre, width, height $(u, v, w, h)$ in normalized pixel coordinates</td>
<td>Overhead camera image</td>
<td>4</td>
</tr>
<tr>
<td>$T$</td>
<td>Pose of end effector that places it over the lettuce</td>
<td>Position $(x, y, z)$ Orientation $(x, y, z, w)$</td>
<td>Robot arm</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 4.1 Input and output variables to the Learned Open Loop neural network.

listed in Table 4.1 and are shown in Fig. 2.7.

\[ T = f(P_a, P_v, R_v, BB_o) \]  

(4.8)

The input variables are mapped to $T$ using 4 fully-connected layers, tanh activation functions, a mean squared error loss function and the Adam optimizer. The input and output layer size is determined by the dimensions of the variables in Table 4.1. The size and number of the hidden layers was arrived at by iterative experimentation, trading off training speed against accuracy. The neural network was trained for 2000 iterations (Fig. 4.7b), using data taken from successful Two-Stage Servoing picks. Separate datasets were gathered in simulation and reality: in simulation 2000 samples were gathered, while on the real robot a smaller dataset of 50 successful picks was duplicated to form 2000 samples and found to work quite acceptably.

Stage 1 of the movement is open loop and fast, with the end effector moving swiftly to $T^*$. Stage 2 is identical to stage 2 of Two-Stage Servoing, with the end effector descending over the lettuce, self-correcting as it goes. As a result, Learned Open Loop is faster than Two Stage Servoing, while retaining much of its robustness.
4.4 Experiments: Comparing Three Control Approaches (Gen 1, 4 and 5)

During the COVID pandemic when field testing was unavailable, the new Approach methods of Gen 4 and Gen 5 were tested in the laboratory and simulation, using Gen 1 as a baseline.

4.4.1 Experiment One: Three different control systems

4.4.1.1 Experiment:

These three different Approach methods were tested for speed in simulation and then on the real Vegebot robot in the lab environment. A lettuce (either a paper photograph for the lab environment or a sphere and texture mapped disc in simulation) was placed on the ground and each control method was tested 5 times.

4.4.1.2 Results in simulation:

The **Open Loop** method is the fastest method (2.4s duration) as it can ignore any further sensory information once the trajectory has started (see Fig. 4.8). The end effector accelerates and moves rapidly in a straight line towards $T^*$ as quickly as the hardware permits. On reaching $T^*$ it accelerates downwards towards the ground. The changeover between the two stages can be seen as a kink in the time series graph of Fig. 4.8 (a) (Open loop, Simulation), at $t = 8.0s$. Note that the end point of the trajectory does not coincide perfectly with the lettuce centre, due to noise in the vision system and an imperfect static model.
Approach

Fig. 4.8 Experiment One: End effector trajectories under the different control systems, (a) in simulation and (b) on the real robot. The left graphs show the end effector's trajectory in the X-Y plane with the start and end points labeled with time $t$. The right graphs show the end effector's trajectory along the Z axis over time $t$. The vertical purple lines indicate the start and end of the picking sequence.
The **Two Stage Servoing** method is the slowest (7.3s duration). While the arm could travel faster, the bounding box updates will then lag behind and the end effector may miss the lettuce. In the first stage the descent in Z is slower (between $t = 2.0 - 4.5s$) than the first method (Fig. 4.8a, centre right). Descending too rapidly runs the risk of the lettuce falling outside the field of view of the end effector camera, which then misses the target. In the second stage (between $t = 4.5 - 9.3s$) the descent is faster, but still limited by the need to arrive cleanly at the XY position of the lettuce before touching ground.

The **Learned Open Loop** is a compromise between the first two methods (4.0s duration). The first stage is open loop (between $t = 6.0 - 7.4s$) with rapid descent. The second stage uses the same algorithm and parameters as the Two Stage Servoing, using the bounding box $BB_{ee}$ to drive the velocity control (between $t = 7.4 - 10.0s$). This second stage could potentially be further optimised for speed, sacrificing robustness to errors in the learned value of $T$. This would steepen the second stage curve (Fig. 4.8, bottom right) reducing overall travel time to close to the Open Loop value, but missing the lettuce in the face of extreme environmental noise.

### 4.4.1.3 Results on real robot:

The trajectories on the real robot (on the right of Fig. 4.8) are slower, but follow the same pattern as in simulation. Open Loop is the fastest (4.8s duration), Two Stage Servoing is the slowest (8.9s duration) and Learned Open Loop is a compromise (6.2s duration).

The velocities of the different stages of the trajectories had to be adjusted downwards on the real robot for two main reasons. Firstly, more protective stops were triggered by the UR10 controller on the real robot than on the URSim software. This suggests that the URSim’s simulation of the behaviour of the arm controller under real conditions is not exact. Secondly, the geometry of the real USB cameras differed from the simulated ones, meaning that the successful handover from overhead to end effector cameras was more sensitive to end effector velocity. The simulated cameras will be altered in future work.

Overall though, the three different control methods that were prototyped in simulation worked similarly in reality, with minimal adjustments.

### 4.4.2 Experiment Two: Robustness to perturbations and distance

#### 4.4.2.1 Experiment:

The second experiment was designed to test the control methods’ robustness to perturbations in the system and to the initial distance of the lettuce from the parked end effector.
Approach

Perturbations were added to the simulation by moving the UR10 arm off its usual alignment. In the simulator, a virtual ball joint was introduced under the arm and the yaw, pitch and roll modified. On the real robot, the platform itself was inclined while keeping the overhead camera support vertical. The magnitude of the perturbations is given by the IMU and measured in radians, referring to the yaw pitch and roll angles introduced.

On the real Vegebot platform, for practical reasons, the perturbations were modelled slightly differently. The end of the platform was jacked up to incline it to a similar angle to the ball joint method. The overhead camera was then adjusted back to a vertical orientation. This is similar, but not geometrically identical to the method used in simulation. 5 sample picks were made for each combination of perturbation size and control method.

In separate tests, the distance of the lettuce from the parked end effector was varied to ensure that the methods worked over the range of the workspace. 5 sample picks were made for each combination of distance and control method.

4.4.2.2 Results in simulation:

The results in simulation can be seen on the top half of Fig. 4.9. The position error of Open Loop increases as the noise is augmented, to the point where 100% of the picks fail when the noise reaches 0.25 radians. Learned Open Loop and Two Stage servoing are robust to the noise, as the visual servoing component brings the end effector back to the correct position. In terms of picking time though, Open Loop is fastest as once the initial lettuce localisation is performed, no further vision system updates are required. Two Stage Servoing is slowest, as it needs to move slow enough not to outpace the localisation update rate. In some ways, it is an over-cautious strategy: it would work even if the lettuce moved. Learned Open Loop, a combination of a fast open loop approach for stage 1 and slower servoing servoing in stage 2, resulting in a picking time between the two extremes.

The results for varying lettuce distance are as expected: there is no variation in picking error with distance from the parked end effector, suggesting that the projection model is accurate.

4.4.2.3 Results on real robot:

Results on the real robot follow the trends seen in simulation. The increase in position error on Open Loop (from 3 to 9 cm) is not as extreme as in simulation: practical considerations prevented noise being applied to all three rotational exes. Learned Open Loop and Two Stage Servoing are robust to low noise, as in simulation, and the differences in picking time follow
the same ordering. In general, it proved necessary to run the real robot at a lower velocity than the simulated version to prevent misses and protective stops.

Failures in Open Loop start at a lower noise level (0.10 radians), than in simulation (0.25 radians). In addition, Learned Open Loop fails as much as Open Loop does at the highest noise level (0.25 radians): the targeted positions are too far from the real ones to give the servoing time to compensate. Picking error distance increases with lettuce distance, suggesting inaccuracies in the projection model that grow towards the edge of the visual field.

4.5 Conclusion and Contributions

All the tested methods show the expected trade-offs in simulation and in reality. Open loop harvesting is fast but vulnerable to pertubations and to model imperfections. Two Stage Servoing is more accurate and robust, but slower where the frequency of vision updates is constrained by the vision system hardware. The switching scheme, appropriate for this and similarly constrained problems appears to be novel.

The speed of the Learned Open Loop method lies between the other two, trading off some robustness to improve speed. By using Learned Open Loop, the approach time of 20s can be reduced to 6.2s and so the overall cycle time decreased from 31s to 17.2s, a 45% improvement and closer to the target human value of 6s.

There is an analogy here to human reaching, where an open-loop initial ballistic reach moves the wrist to the vicinity of the target object, followed by a more measured feedback-driven grasping process [73]. The combination of these two stages appears novel, as does the use of multi-modal sensory input to compensate for complex damage scenarios. The resulting fast but robust performance is a contribution.

The development of the five Approach methods represented a journey away from the perceive-model-plan-act pipeline towards more embodied methods, where internal representations were simpler or even absent. In hindsight, embodied methods were the best place start, because of the uncertain nature of the unstructured environment and the difficulty of inferring sufficiently accurate models. This insight contributed to the development of this thesis’ proposed overall method.
4.6 Future Work

The comparative experiments should be replicated in a field study before definite conclusions can be drawn. Two Stage Servoing and Learned Open Loop are promising, but need validation in the target environment.

An adaptive process leveraging Learned Open Loop (see Fig. 4.6b) should also be tested and compared to the use of Gen 3 and Gen 4 in isolation. Theoretically, it would allow the Vegebot to rapidly adapt to wear and tear. If computational hardware is a limitation, network retraining does not have to take place on the Vegebot: it could be performed in the cloud and downloaded to the robot when ready.

At present, the perceptual system used for Approach is the unmodified Detection system. The Approach system however has no need to continually reclassify the targets; this second stage of the Detection pipeline slows the update rate. A modified Approach perceptual system trained on the same datasets and optimised for speed could focus purely on localisation, perhaps with an object tracking algorithm. Such a system could potentially speed up the visual servoing part of the trajectories.
Fig. 4.9 Experiment Two: The three control systems in simulation and reality as perturbations are added (left) and the distance of the parked end effector to the lettuce is increased (right). The top graphs in each block show the final distance of the end effector centre from the lettuce. The middle graphs show the trajectory time. The lower graphs show failed picks from a total of 5 attempts.
Chapter 5

Manipulation

5.1 Problem Statement

The manipulation challenge for the Vegebot was to grasp and detach the lettuce head cleanly from the rest of the plant, cutting through the stalk close, but not too close to, the head. In addition, the removal of outer leaves needed to be contemplated, either by the Vegebot itself, or later in a subsequent food handling process. This removal operation is currently performed by human workers using a knife.

(a) Baxter being positioned in the lettuce field.  
(b) The two handed Baxter mimicked the human worker configuration.

Fig. 5.1 The initial attempt to directly develop a fully integrated two-handed solution was a failure.
5.2 Development Timeline

Initial attempts in early 2016 to produce a complete, integrated system failed\(^1\). A two-handed Baxter robot with a soft gripper and knife (see Fig. 5.1) repeatedly missed lettuces and even when it reached them, was unable to muster the force to cut through the stalk. There were too many points of failure in the system and it was decided to decompose the project into separate tracks, each focusing on single part of the overall problem.

The subsequent development of the Manipulation solution can be divided into two phases. In the first, a large number of simple prototypes were produced that were intended to field-test and compare different cutting methods, rather than be complete end effectors. In the second phase, the development of the end effector was characterised by rapid iterative development in the field of an embodied-first solution over the remainder of 2016 and 2017 (see Fig. 2.10). At each iteration of the end effector, the focus was on solving one key problem before passing on to the next.

5.2.1 Early cutting prototypes

5.2.1.1 Rotary vs Linear End Effector Prototypes

After abandoning the integrated approach of the Baxter robot, it was decided to produce end effector prototypes that concentrated on the central manipulation problem: how to cleanly cut through the stalk of a lettuce? Detection and approach were delegated to human experimenters, who placed the experimental end effectors directly over the lettuces to test their cutting abilities (see Fig. 5.2).

Two initial prototypes with electric actuators compared the performance of rotary and linear cutting (see Figs. 5.2b and 5.2c). In both cases an angled plastic bucket acted as an enclosure that both grasped and stabilised the lettuce while cutting took place, a purely embodied solution to the positioning problem.

As with many of the early experiments, no formal metrics were collected as the results were not even close. The rotary cutter hacked repeatedly at the lettuce stalk, making only small cuts. The linear cutter was more successful, reaching half way or more through the stalk with a single cut. The linear approach was selected for the next pair of prototypes.

\(^1\)The lead work on the end effector in this chapter was by Josie Hughes and Fumiya Iida, with the collaboration of the author. The force feedback control system was devised by the author.
5.2 Development Timeline

(a) Placing and testing the cutting prototypes by hand.

(b) The rotary electric cutter.

(c) The linear electric cutter.

Fig. 5.2 Early prototypes focused solely on the cutting problem, comparing linear and rotary electric cutting mechanisms. The plastic bucket enclosures had the effect of stabilising and gripping the lettuce head.
5.2.1.2 Electric vs Pneumatic Linear Actuators

The next two cutting prototypes compared the use of electric and pneumatic linear actuators. The new electric prototype increased the power of the linear electric actuator over the previous version.

The new prototypes were initially tested in the lab, checking to see if they could slice lettuces and cucumbers in half. The electric prototype achieved a higher velocity than before, but failed to cut cleanly. The higher-force pneumatic actuator was much more successful.

Two pneumatic prototypes were then tested in the field, with smaller (see Fig. 5.4) and larger actuators. The larger actuators could reliably cut through lettuce stalks.

This cutting problem was now solved in principle. The plastic bucket, beside positioning and stabilising the lettuce head, had the additional advantage that it peeled back many of the outer leaves of the lettuce head, potentially obviating the need for a subsequent peeling process.

Subsequent end effector prototypes (Gen 1 to 5) maintained this basic design and were built to be integrated into, and tested with the full platform. The progress of these versions is summarised below and in Fig. 5.3.

5.2.2 Integrated End Effector Prototypes

5.2.2.1 Gen 1: The Bucket

Key Problem: Cutting

By the first field trip at the beginning of the new lettuce season (15/5/2017) the full Vegebot platform had been designed and assembled. A new version of the linear pneumatic prototype with the larger actuators was developed for attaching to the UR-10 arm (see Fig. 5.5).

A webcam was fixed to the end effector to facilitate experiments on the Approach track. A foam cushion inside the bucket helped to position the lettuce head. Two linear pneumatic actuators drove the blade for cutting.

While the cutting problem was solved by this version, two further problems were revealed on the field trip. First, the actuators were long and would get caught on lettuces adjacent to the one being harvested, making the positioning of the end effector hard. Second, it was difficult to judge at what vertical height to position the end effector before cutting. It was decided to add legs to the next generation, to enable the blade to be vertically positioned with respect to the ground.
<table>
<thead>
<tr>
<th>Version</th>
<th>Key Problem</th>
<th>Characteristics</th>
<th>Performance</th>
<th>Weaknesses</th>
<th>Photo</th>
</tr>
</thead>
</table>
| Gen 1 15/5/2017 | Cutting | • Cutter 2 x 1.0 MPa  
• Bucket shape & foam act as passive gripper | • Cutting good | • Actuators too low and long  
• Need to adjust cutting height | ![Image](image1.png) |
| Gen 2 22/6/2017 | Reducing footprint | • Cutter 2 x 1.0 Mpa (vertical)  
• Gripper 1.0 Mpa  
• Acrylic cage  
• Belt drive  
• Fixed legs | • Cutting inconsistent | • Fragile structure  
• Cutter needs more power | ![Image](image2.png) |
| Gen 3 13/7/2017 | Fragility | • Cutter 2 x 1.5 Mpa  
• Gripper 1.0 Mpa  
• Acrylic / steel cage  
• Lever drive | • Lever sticks, otherwise cuts well | • Fragile structure  
• Lever sticks  
• Hard to adjust cutting height | ![Image](image3.png) |
| Gen 4 19/10/2017 | Fragility, cutting | • Cutter 1 x 1.5 Mpa  
• Gripper 1.0 Mpa  
• Steel cage  
• Belt drive  
• Adjustable feet | • Reliable cutting | • Too heavy for arm | ![Image](image4.png) |
| Gen 5 8/11/2017 | Weight reduction | • Cutter 1 x 1.5MPa  
• Gripper 1.0 Mpa  
• Steel cage (weight reduced)  
• Belt drive  
• Adjustable feet | • Detachment 97%  
• Damage 38% | • Still heavy  
• Cutting height imprecise  
• Instability on ground contact | ![Image](image5.png) |

Fig. 5.3 Summary of the integrated end effector versions.
Fig. 5.4 The linear pneumatic prototype was the only variant that reliably cut through lettuce stalks.

(a) The first integrated end effector maintained the bucket enclosure.
(b) End effector seen from below.

Fig. 5.5 Gen 1 End Effector: The first integrated end effector used two pneumatic actuators for cutting, incorporated a webcam and a foam cushion for positioning the lettuce head.
5.2 Development Timeline

5.2.2.2 Gen 2: Vertical Actuators

Key problem: Reducing footprint

The next iteration concentrated on reducing the footprint of the end effector to avoid interference with neighbouring lettuces. The cutting actuators were mounted vertically and transmitted force to the blade via a belt drive (see Fig. 5.6a). The bucket shape was replaced with a cage-like construction and a soft gripper that could be moved into position by a second, horizontally-mounted actuator.

The new end effector was successfully tested in the laboratory, but proved too fragile when taken on its first field trip (22/6/2017). The combination of acrylic plates, chosen for their light weight, and metal tubing was not rigid enough to resist the stresses of collisions with lettuces and ground, and the forces being transmitted through the structure by the cutting actuators. In addition, the vertical cutting actuators were not powerful enough to transmit the required force through the belt to consistently cut through the lettuce stalk.

5.2.2.3 Gen 3: Cage with lever

Key problem: Fragility

For the next generation of end effector, a lever mechanism replaced the belt drive and the power of the cutting actuators was massively increased. The idea was to establish a force profile that would have no issues cutting any lettuce stalk and then scale down from there. The actuators were once again mounted horizontally, but because of their position high above
Fig. 5.7 Gen 3 End Effector: Two powerful actuators drive the blade via a lever mechanism. A smaller actuator drives the soft gripper. Robustness has been improved, but not fully achieved, by using a steel top.

Apart from the continued fragility, the lever proved to be another weak link and would get stuck. In addition, the fixed feet meant that it was hard to adjust the vertical positioning of the end effector and lettuce heads were often cut instead of the stalk.
5.2 Development Timeline

Fig. 5.8 Gen 4 End Effector: A single powerful actuator drives the blade via a new belt drive. The robust structure has been made entirely out of steel.

5.2.2.4 Gen 4: Steel cage

*Key problem: Fragility, cutting*

The end effector was reconstructed using exclusively steel plates, eliminating the acrylic components. The belt drive was restored and the twin horizontal actuators reduced to one (see Fig. 5.8). In addition, adjustable height feet were added to the legs, allowing the vertical cutting position to be manually adjusted in the field.

This iteration was tested in the field on 3/10/2017 and 18/10/2017 at the very end of the lettuce season and proved to be robust and capable of reliably cutting through the lettuce stalks and could be adjusted for cutting height. Unfortunately, it was too heavy to be lifted by the UR-10 arm over the distal regions of the workspace and caused frequent protective stops.

Force feedback detection was added to the Approach software for this iteration, and the combination of adjustable height feet and force sensing allowed the end effector to be more successfully positioned with respect to the lettuce stalk.

5.2.2.5 Gen 5: Lightened steel cage

*Key problem: Weight reduction*

The final version of the end effector was developed in the lab during November 2017 and tested in the field the following year in the final field trials at the start of the next lettuce
season (16/5/2018). The weight was reduced from Gen 4 by cutting holes in the central steel plate (see Fig. 5.9) but the mechanism was otherwise unchanged from Gen 4 (see Fig. 5.10). The working of the complete, final end effector will now be fully described.

5.3 The Final End Effector

This section describes the final generation End Effector (Gen 5), used in the final field trials on 16/5/2018.

5.3.1 Force Feedback Driven Harvesting

The lettuce harvester was designed to achieve reliable, efficient harvesting of lettuce with minimal damage to the lettuce. To meet supermarket specifications the lettuce stem should be cut with a single consistent straight cut such that there is approximately 2mm of stem left. The outer leaves of the lettuce should also be removed where possible. A UR10 6 degree-of-freedom arm is used to provide movement of a custom end effector which has been specifically designed for lettuce harvesting. The UR10 arm is mounted on a mobile base which can be moved along the rows of lettuce.

The picking sequence (Figure 2.3 ‘Pick Sequence’) demonstrates how there are two stages to the physical cutting aspect of the harvesting procedure. To minimize the damage to the lettuce and also achieve a clean cut, a method where the end effector is made of two mechanisms has been used. Firstly, a soft clamping method is used to hold the lettuce throughout cutting and when lifting. Secondly, a cutting mechanisms is required to cut the stem of the lettuce at a given height. The cutting mechanism requires force (≈20N) to cut through the stem and outer leaves, whilst also requiring height adjustability and also a straight linear cut.

5.3.1.1 End Effector Design

To achieve sufficient cutting force to cut the stem, a high impact, straight cut is required at the base of the lettuce. A number of different mechanisms were tested to determine which could achieve sufficient force and quality of cut: soft gripper and knife hand, pneumatic actuation, belt drive and rotary chopping (see section 5.2).

The two handed approach lacked sufficient cutting force and required a high level of co-ordination between the two arms. A rotary electric motor approach lacked the force to reliably cut the stem and led to the mechanism having to ‘hack’ at the stem. Although the linear actuator approach provided sufficient force, the speed was low, leading to poor cut
Fig. 5.9 Gen 5 End Effector: The principal change from Gen 4 is weight reduction, achieved by cutting holes in the centre steel plate.
quality. The pneumatic cutting mechanics provides a high power-to-weight ratio, making it highly suited for this application where a fast clean cut is required. Although there is no position control, pneumatic actuation allows for easy to implement cut/open control.

The soft gripping mechanism has a single moving gripper and a fixed gripper lined with foam. Similar to other harvesting end effectors [34, 39], a pneumatic actuator is used to control the gripper as this can be used to provide controllable compliance by varying the air pressure such that the lettuce is held but not damaged with simple open/close control.

The end effector developed is shown in Figure 5.10, with the design parameters given in Table 5.1. The end effector used only two actuators, one for grasping and one for cutting to enable simple control. A timing belt system was used to transfer the linear motion from a single actuator to both sides of the blade to allow smooth movement. This allows the actuator to be mounted above the height of the lettuce, such that when cutting it does not interfere. The belt drive system allows for the height of the cutting mechanism to be easily altered.

5.3.1.2 Force Feedback Control

A key challenge to successful harvesting was reliably cutting the lettuce stalk at the correct height in an environment which is highly varying, uncertain and unknown. To achieve this, the ground was used as a fixed reference point and the stem was assumed to be a fixed...
Table 5.1 Specification of the end effector developed

<table>
<thead>
<tr>
<th>End Effector Parameters</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>8 kg</td>
</tr>
<tr>
<td>Height</td>
<td>45 cm</td>
</tr>
<tr>
<td>Width</td>
<td>45 cm</td>
</tr>
<tr>
<td>Depth</td>
<td>30 cm</td>
</tr>
<tr>
<td>Gripper Pneumatic Actuator Specification</td>
<td>1 MPa, Bore 10 mm, Stroke 15 cm</td>
</tr>
<tr>
<td>Cutter Pneumatic Actuator Specification</td>
<td>1.5MPa, Bore 15mm, Stroke 20cm</td>
</tr>
<tr>
<td>Timing Belt</td>
<td>5.08mm pitch, 203cm length, 20mm width</td>
</tr>
<tr>
<td>Length of Travel of Blade</td>
<td>200 mm</td>
</tr>
<tr>
<td>Cutting Knife Length</td>
<td>250 mm</td>
</tr>
<tr>
<td>Inner Area to encapsulate Lettuce</td>
<td>25 cm x 25 cm</td>
</tr>
</tbody>
</table>

distance above the surface. Using force-feedback from the joints of the UR10 robot arm, the end effector is lowered towards the ground, enveloping the lettuce, until a given force was achieved and contact with the ground could be assumed. The cutting height relative to the ground can be adjusted by manually varying the height of the cutting mechanism. A force threshold, $T$, was found by experimentally determining what force is required for the end effector to interact with the ground, i.e. when it overcomes the resistive force of the leaves and other ground reaction forces, $F_R$. The force threshold was experimentally determined to be 60N to ensure all leaves were pushed away from the lettuce head and the end effector was in contact and level with the ground. This approach is summarized in Figure 5.11.

This method helped push out the outer leaves of the lettuce which interfered with the cutting mechanism. This also allows the end effector to self-level on the ground, and provided stability and consistency. Small ‘feet’ were added to the end effector to allow stability to be achieved and prevent it from pressing too low into the ground. This approach allows the system to adapt to different field conditions, for example different soil heights relative to the tractor track heights.

Once fully positioned, the lettuce is grasped and the cutting takes place. Each of the pneumatic actuators is controlled by a valve which has two position controls. Two digital outputs from the UR10 end effector are used to control the valves. After the correct height is achieved using force feedback, cutting is triggered by first actuating the grabbing mechanism so the lettuce is held in a fixed place. The cutter pneumatic system is then actuated so the blade cuts the stem of the lettuce. The arm can then be lifted, with the knife released and then the grabber retracted to release the lettuce.
Besides these two challenges, an additional one was that the weight of the end effector was at the limit of the payload ability of the UR10. This restricted the arm to moving more slowly than would otherwise be necessary. This will be discussed in the experimental results.

### 5.3.2 Future Challenges

The last mechanical iteration of the end effector successfully harvested lettuces in the final field trials, but nevertheless had three areas for potential improvement: weight reduction, stabilisation and damage rate reduction.

The 8kg weight of the end effector was close to the UR10’s theoretical limit of 10kg. This limit is not binary, but manifests itself in an increased frequency of "protective stops" as the limit is approached. The frequency of protective stops also increases the faster the velocity of the end effector on its trajectory. A trade-off becomes necessary: each protective stop currently requires a manual reset of the UR10 and Vegebot software; a slower end effector means an increased cycle time for picking. Reducing the weight of the end effector is therefore highly desirable if robustness can be maintained. One approach would be to build the end effector from another material, such as carbon fibre.
As the end effector is lowered, the feet come into contact with the ground, or the layer of lettuce leaves on top of it. As the ground is uneven, the placement of the end effector can be unstable, with some feet in contact and others not. Making the feet or the legs compliant could help stabilise the end effector at rest.

The final major challenge would be to reduce the damage rate. As detailed elsewhere, the damage is largely a function of rather artificial supermarket standards for lettuce perfection. Nevertheless, it is desirable to be able to more precisely position the cut location on the stalk with respect to the lettuce head. One potential solution would be to allow the soft gripper to move vertically as well as horizontally, lifting the lettuce head and stretching the stalk. Some sort of sensory feedback could detect the height of the bottom of the lettuce, or a more embodied solution might be achievable simply be modifying the shape of the soft gripper.
Chapter 6

Vegebot: Final field trials and experimental results

6.1 Final Field Trials and Harvesting Performance

The final field tests of the full Vegebot platform were performed in May 2018 at a lettuce field in Cambridgeshire, UK. These final tests followed on from over 12 previous visits to the field with well over 300 lettuces harvested.

At the beginning of each experimental session, the Vegebot was assembled at the start of a lettuce lane. Typically, a three person crew participated, one operating the control laptop, one observer and one checking and resolving any physical issues and enabling the air compressor when required.

After positioning Vegebot at the start of a lettuce lane, the lettuces within the viewport of the overhead camera were detected and picks attempted. Once attempts had been made to pick all feasible lettuces, the platform was moved forward down the lane to the next unpicked rows. Each lettuce position, and false positives or negatives were recorded, together with the number and trajectory of all pick attempts. Finally, each lettuce was inspected for damage, in particular for the stalk being cut too close to the lettuce body. In total, 69 lettuces were detected by the vision system, 60 were in range of the robot arm and harvesting attempted with 31 lettuce harvested successfully. A video of the Vegebot in operation was recorded \(^1\).

6.1.1 Overall Harvesting Performance Metrics

The results of the field experiments are shown in Table 6.1. Considering all the harvesting attempts, the detachment success is found to be 52% (31 out of 60 lettuces correctly identified, \(^1\)https://youtu.be/UR-7LBdI7Z4
excluding false positives). However in 28 cases the harvesting failure was due to practical restrictions (weight of the arm, practical workspace of the robot arm and the range of the overhead camera viewport), such that it was physically not possible to pick some lettuce. If the limitations of the arm are ignored, and the denominator reflects only those lettuces within the practical workspace, then the Detachment Success rises to 97% (31 out of 32). In other words, with one exception, if the arm could reach the lettuce, the end effector could pick it. Although this is a considerable exception, it could be simply achieved by using a robot arm with increased torque output.

Table 6.1 Overall system performance in the harvesting tests. Total lettuces attempted considers only lettuces within restrictions imposed by arm strength.

| Metric                        | Result                  | Definition                                                     |
|-------------------------------|-------------------------|                                                               |
| Total Ground Truth Lettuces   | 69                      | All lettuces present in viewport                               |
| Total Lettuces Detected       | 61 (1 false positive)   | All lettuces detected in viewport                              |
| Total Lettuces Attempted      | 32                      | All lettuces present in workspace                              |
| Total Lettuces Detached       | 31                      | Number of successfully picked qualified                        |
|                               |                         | Number of detected qualified                                    |
| Detachment Success            | 97%                     | Number of successfully picked qualified / Number of detected qualified |
| Harvest Success               | 88%                     | Localisation Success x Detachment Success                       |
| Cycle Time                    | 31.7s, $\sigma^2 = 32.6$ | Complete cycle time from lettuce to next                       |
| Damage Rate                   | 38%                     | $N_{lettuce harvested in unsaleable condition} / Total number harvested |
| Leaves to be Removed          | 0.75, $\sigma^2 = 1.42$ | Average leaves to remove for saleability                       |
| Total lettuces attempted      | 69                      |                                                               |

Examples of the harvested lettuce are shown in Figure 6.1 showing high quality cuts and also showing those with unwanted outer leaves or damage. The distribution of the lettuces which required extra leaves to be removed, extra cutting attempts and the cycle time is shown in Figure 6.2. The cycle time varies greatly depending on how far the arm needs to travel from the rest position to the lettuce, exacerbated by end effector weight slowing the movements. In a few cases, one extra leaf needed to be removed (manually) to achieve supermarket perfection. Additionally, in some cases extra cuts were required. This was often due to the leaves of the lettuce and movement of the lettuce head within the cutting area. Additionally, the cuts were generally a little too close to the body to be acceptable in the current market.

The average Cycle Time was 31.7 seconds, with a variance of 32.6 seconds. Again, this value was largely due to the limitations of the arm and the weight of the end effector. Of the trajectory sections in Figure 4.5, all but the short Grasp-Cut section (2 seconds) have
6.1 Final Field Trials and Harvesting Performance

Fig. 6.1 Examples of harvested lettuce showing some with an ideal cut, others with unwanted outer leaves and damaged outer leaves.

Fig. 6.2 Distribution of the cycle times, leaves to remove and extra cuts required for the various lettuce harvesting experiments. The cycle time varies greatly depending on how far the arm needs to travel from the rest position to the lettuce.
their speed limited by the arm’s payload capacity. A much reduced Cycle Time should be achievable with a stronger arm or lighter end effector. In addition, around a quarter of the cycle time is taken by the Fine Tuning of the end effector position. Any improvements to the accuracy of the overhead camera localisation would further reduce the overall cycle time.

Reducing the Damage Rate (38%) will require further experimentation. Supermarket chains, the largest wholesale lettuce buyers, have strict standards for the length of the cut stalk to improve the vegetable’s appearance in packaging. According to these standards, aesthetic rather than relevant to the lettuce’s suitability for eating or not, the end effector often missed the ideal length, cutting in most cases slightly too close to the lettuce head. Of the 32 picks, only 2 actually resulted in inedible lettuces. Improvement can probably be made by refining the force feedback mechanism and perhaps introducing field-dependent depth calibration at the start of each session. This remains for future work.

Again, buyer standards dictate that a packaged lettuce should not have too many superfluous leaves in the packaging. At present, a human harvester will deftly remove a few leaves after each pick before passing the lettuce onto the harvesting rig. The end effector left the picked lettuce with an average of 0.75 additional leaves that are undesirable by these standards. These would have to be removed further down the production chain by hand, or in an automated fashion.

It is worth noting that both the metrics for Damage Rate and Leaves to Be Removed could be substantially improved by permitting a greater range of appearance of the vegetable on supermarket shelves. Until the robot improves, this suggests a dual pricing strategy, with a higher price paid by the consumer for a ‘perfect’ hand-picked lettuce and a lower price for a more variable but quite edible robot-picked one.

6.2 Future Work

Further work needs to be done to take Vegebot from research to practical usage.

As mentioned in chapter 4, the newer approach systems such as Two-Stage Servoing and Learned Open Loop still need to be tested in the field to be considered validated. The logic to switch dynamically between the two methods as necessary remains to be implemented.

It will be necessary to consider and develop the final form factor for the Vegebot platform. On the one hand, variants of the end effector and arm could be attached to an existing autonomous platform such as Thorvald. On the other hand, they could be attached to an existing harvesting rig (see Fig. 1.1a), to allow them to be used alongside human labour. The choice is a commercial one.
The end effector itself can be improved. Weight reduction will be important in improving cycle time, perhaps by manufacturing it from carbon fibre instead of steel. Making the feet of the end effector compliant would likely stabilise it more as it hits the foliage and ground. Lifting the grasped lettuce slightly could stretch the stalk and improve the position of the cut, improving compliance with supermarket standards. Automatic evaluation of the cut quality, using another convolutional neural network, would allow for better quality monitoring and drive improvements in subsequent hardware and software iterations.

The robot arm itself is expensive, over-engineered (only three of the seven degrees of freedom are currently used) and copes poorly with the heavy end effector. A cartesian system should be developed and tested for comparison. Additionally, a parallelised cartesian version that could support a row of end effectors working simultaneously could radically improve performance and harvest faster than human workers.
Chapter 7

Conclusions

This chapter will examine why it is worth pursuing agricultural robotics and recapitulate this thesis’ contributions. Beyond societal needs, it will detail why we should pursue the field from a purely research point of view and analyse the challenges of making a business from agricultural robotics.

7.1 Thesis objectives

7.1.1 The Creation and Testing of Vegebot

There is much remaining work required to achieve an iceberg lettuce harvester for commercial operation. Existing challenges include visual analysis, precise manipulator control, harvesting rig development, and reduction of the overall cycle time and costs. In this project the focus was not to develop a commercial product, but to demonstrate proof-of-concept experiments which provide research outcomes which can aid future development of agricultural robotics systems not only for iceberg lettuce, but many other crops. This section discusses the design rationale behind the development process.

The final prototype of Vegebot is a result of more than 15 iterations and on-site field tests which were carried out in the UK harvest seasons (July-Sept) between 2016-18, and also countless lab based experiments. In each iteration of the platform, new software and hardware redesigns were tested in the field, data gathered and results compared. The development approach adopted was to produce a modular system to enable rapid integration and testing of the architecture systematically. Frequent field tests were used to provide performance feedback, customer insight and to identify the improvements required. As a consequence of this approach, the physical design changed radically from week to week (see Figure 5.3). This process was kept grounded by the use of standard harvesting metrics [5] to monitor
Conclusions

progress. The author submits that this iterative approach is more likely to yield robust, field-worthy robots than careful upfront design based on an idealized version of the problem.

As an example of the approach taken, the available visual datasets of lettuces were not ideally suited for an optimal vision system. Two separate datasets, one for localisation and one for classification, were both of reasonable quality in themselves but in an ideal world would have been combined into one integrated whole. Rather than spend time and resources gathering yet another dataset to replace them, the Vegebot’s neural networks were quickly adapted to make use of what was available. This enabled the robot to detect lettuces correctly, solving the problem for the time being and allowing work on the overall system to continue. With future iterations and online data-gathering this architecture could be simplified once again into a single, fully-integrated CNN architecture, or separating out a distinct, faster tracking system for Approach.

It is noteworthy that a vision system based on a standard convolutional neural network architecture was able to achieve the localisation results that it did, given the difficulty of the task for a human harvester. Many of the previous harvesting robots detailed in Section 1.2.2 required vision systems carefully tailored to the fruit or vegetable in question (eg. detecting colour or depth). For example, broccoli heads are detected using an elaborate pipeline of RGB-D sensors, point clouds and feature extraction in [68] and radicchios using hand-crafted features and particle filters in [39]. CNNs, together with some rapid and informal data gathering, proved ‘good enough’ for the non-trivial localisation of iceberg and may turn out to be sufficient for other crops ([61]).

Considering the mechanical development, by making field testing in collaboration with the customer central to the project, the robot design naturally adapted itself to real-world commercial conditions. Vegebot operates in the same fields and along the same lane layout as human harvesters. Neither the environment nor the crop itself was altered in any way to facilitate the automated harvesting. By contrast, solutions using water knives require careful selection of the cultivar and modifications to the way they are planted [111]. Vegebot-derived solutions could be gradually deployed alongside existing methods, rather than requiring major changes to current practices. The control and calibration software was repeatedly simplified to provide a solution that worked robustly in the field. Complex algorithms to model in 3D and determine the optimal cutting position were replaced with mechanical legs that provided force feedback from the ground, giving the robot a simple signal on when to cut. A design change was considered an improvement whenever a mechanical feature or software module was eliminated. In the long-term, this preference for embodied simplicity over sophisticated software solutions may prove limiting, yet Vegebot has already achieved important results. The use of standard metrics as proposed by [5] kept the project on track
and focused on steady, incremental improvements. The authors’ feeling is that the iterative, simple approach can yield yet many more dividends before being exhausted.

As the project stands, the damage rate, caused by cutting the lettuce stem too short, is too high for supermarket standards, although the harvested vegetables were perfectly edible. The most recent sample size of 69 lettuces was enough to confirm this as the next problem to address (hundreds of lettuces had been harvested over previous iterations). Future versions of Vegebot will need to address and improve the damage rate, perhaps with visual feedback from the harvested lettuces dynamically adjusting the force threshold at which the cut is made. In parallel, the end effector needs to be made lighter to achieve a human-level cycle time, possibly by manufacturing with carbon fibre, or by using an alternative, stronger cartesian arm design.

### 7.1.2 The elaboration of a general design methodology for agricultural robotics projects

The methodology was both the driver and the end result of the Vegebot project. As development progressed, the process became clearer. How generally applicable is it?

As stated above, the breakdown into Detection, Manipulation and Approach is a simplification. Other projects may require more tracks: a potato-planting robot will need to detect, approach and grasp a seed potato and then detect, approach a planting site where the potato is to be dropped [2]. When considered as behavioural processes (not as development tracks), Detection, Approach and Manipulation will often overlap in time on other projects. Nevertheless, this decomposition into loosely-coupled processes is still useful and can be extended to generic agricultural robots (see Fig. 1.4).

*Detection* may require the use of multi-sensory data: depth-augmented, multi-spectral and near infrared (NIR) vision as well as touch and proprioception. Vegebot continually reclassifies the targets on the fly but only makes use of this information when selecting which one to pick. In general, the classification is a provisional estimate which may be revised at any point in the overall process by the same or different perceptual systems. Some fruits can’t be truly classified without weighing or palpating them [106]. Detection on the Vegebot was a passive perceptual process, but for other fruits or vegetables it may be an active one: moving aside occluding leaves on a strawberry plant to get a better view.

The development of *Approach* on Vegebot ended with embodied, low-dimensional representation, closed loop methods; this thesis’ advice is to start from such methods on future projects. Humans and animals appear to use such methods to catch a baseball [24] and to catch a prey. It may be that some problems can only be solved by exhaustive modelling of the
local environment, but this should be the last resort, not the starting point. While this does not reflect the development of Vegebot, it is also important to consider the perceptual apparatus of Approach as distinct from that of Detection. Approach may reuse Detection’s camera hardware and CNNs, but it may optimize in a different way or use other methods entirely. Vegebot’s Approach system used the combined localisation and classification pipeline from Detection; it could have improved the update rate by using the localisation system alone.

The *Manipulation* method chosen is likely to be specific to each project. Again, the advice is to start with embodied methods and add software only as necessary. Compliance and morphology are likely to outperform the smartest perceptual apparatus.

For future work on refining the methodology, the author proposes three avenues for exploration and testing on future projects. First, developed Approach systems should begin with slower, robust closed-loop methods and look for ways to reduce the trajectory time once these have been attained. Second, the perceptual systems for Detection and Approach should be considered separately, as they optimise for the solution of different problems. Third, customer feedback should be collected together with experimental feedback in a more formal manner on each field test; the insights are valuable and should be shared with the team at large.

### 7.1.3 To develop control methods for the "Approach" problem appropriate for the agricultural context

Moving the end effector reliably to the target proved to be harder than anticipated. The real morphology of a robot under the pressures of the rough field environment drifts from its internal model; projections from 2D to 3D coordinates are no longer accurate. In hindsight, the perception-model-plan-act pipeline is the wrong starting point given the inherent uncertainty in the perception of the environment. Simpler, robust closed loop methods like Two Stage Servoing are superior as a baseline.

The final Approach system (Gen 3) tested in the field uses a calibration method that needs to be performed periodically. This works in the sense that the robot can reliably harvest lettuces, but is unsatisfactory in the long term because of the harvesting time it wastes. Two Stage Servoing and Learned Open Loop appear superior in lab tests, but it will be important to field test the newer methods.

Learned Open Loop holds promise, pending field-testing, particularly when embedded in an online adaptive learning process (see Fig. 4.6b). The robot should be able to harvest quickly until its body morphology and behaviour drifts from its learned "muscle memory", at which point it resorts to a slower more robust method and learns again. The method is
bio-inspired, embodied and, the author believes, well suited to the uncertain, harsh field environment in which agricultural robots operate.

7.2 Agricultural Robotics as a Research Challenge

At first glance, agricultural robotics appears to be simply another industrial application of robotics, more suited to business than to scientific research. Why should we, as a research community, care?

The principle reason is that agriculture forces us to confront hard automation problems that have not been encountered in the industries so far disrupted by robotics. Solving these problems will require intensive, basic research and not just everyday industrial R&D.

Agricultural robots must adapt to their environment and not vice versa. Car factory robots are pampered: in a very real sense, the factory is there for the benefit of the robots and not the other way round – it ensures the exact positioning of parts, perfect lighting conditions, etc. The rigid scaffolding that indoor robots depend on is missing in a wind-swept field. To adapt and thrive, an agricultural robot must solve many of the same problems that animals and humans have had to over millennia. By working to solve these same problems, we gain insights into the nature of life itself.

7.2.1 Challenges in Perception

Robots in car factories can in many case operate efficiently with no perceptual abilities at all. As long as the part to be manipulated is correctly positioned, the robot arm can follow a predetermined trajectory. However, a field is inherently unpredictable. Even lettuces planted in a rigid pattern will vary in size, shape, position and occluding vegetation. Perception is therefore essential.

Visual perception is challenged by this variation and by ever-changing lighting conditions. Convolutional neural networks, running algorithms such as YOLO, are now able to tackle this problem for the first time. However, much remains to be done.

Active perception will be necessary in many contexts, as fruit and vegetables are occluded by leaves. To see all the tomatoes on a vine, a robot arm must actively change viewpoint [74] or even move leaves out of the way. To pick a potato from a basket, the robot may need to move the camera to avoid occlusion by the basket handle [2].

Perception will need to become multisensory. Some aspects of plant health can be revealed by multi-spectral or ultraviolet photography [42]. To assess harvest readiness, lettuces may need to be weighed before their stalks are cut. Touch is a key human sense long
used in the evaluation of fruit and vegetable quality; we know a mango is ready to eat by gently pressing it [106]. These are all under-explored senses in the robotics community, and a given perceptual task may require the fusing of more than one. At the frontier of research knowledge are the artificial implementation of smell and taste [113], which may provide a further richness of data.

### 7.2.2 Challenges in Manipulation

Lettuce picking has been a hard task to automate. It has required the use of embodied techniques: a soft gripper, an evolving model of the body, closed-loop control and environmental constraints.

Fruit and vegetables are soft and prone to damage unless handled delicately. That delicacy requires soft grippers and fine control of the force applied [85]. The control of soft, compliant grippers is complex, and an active area of research [125]. To control the force applied, tactile sensing will need to be embedded into these soft grippers [117]. Soft grippers are liable to wear and tear and in the long term may need self-healing capabilities [116].

Manipulation control of soft end effectors is intrinsically hard, with machine learning techniques replacing inverse kinematics or dynamics [44]. In addition, the problem is non-stationary, as the harsh environment causes wear and tear on the grippers and robot body. The constant refreshing of the body image, as detailed above, is one approach to mitigating this problem.

Some manipulation problems are intrinsically difficult. The removal of the outer leaves of a lettuce is a condition for supermarket acceptability. The Vegebot’s cage pushes down and separates outer leaves to some extent, but more may need to be removed to match human performance. Peeling a lettuce is a hard problem, with solutions potentially requiring two arms or the use of environmental constraints [53]. So is cleanly removing a strawberry stalk.

Each individual problem may appear niche from the research point of view, but by solving them one by one we reveal common techniques and push forward the boundaries of knowledge.

### 7.2.3 Challenges in Learning

The variability of environment, task and target is such that learning will be a core component of agricultural robotic solutions.

Picking a lettuce in a field in Norfolk will differ from picking one in Murcia, Spain. The soil, the lighting and the plant growth will all vary between countries and even between fields.
To be economically viable, a robotic solution must adapt to this variation. Learning is a key tool to enable this.

Learning, in its many guises, is a research focus worldwide. Current frameworks such as reinforcement learning are highly sample inefficient [124] and may not be economically viable in the context of agriculture. On the other hand, fleets of robots can learn together, leveraging the latest in distributed learning algorithms [86].

Agriculture also provides a plethora of potential test cases for imitation learning, from robots directly observing their human coworkers to the use of filmed footage of harvesters being used to inspire new designs. Humans naturally optimise repeated movements for speed and energy minimisation [107]; what could we learn from using posture tracking to discern the trajectories followed by human harvesters?

Humans can also intervene in automated agricultural processes, with humans-in-the-loop providing positive or negative reward signals [36, 114]. By dealing with edge cases as they arise, they can both solve immediate problems beyond the robot’s autonomous capability and provide iterative feedback on the robot design.

7.3 Agricultural Robotics as a Business Opportunity

As well as being a fruitful research area, agricultural robotics is potentially a massive business opportunity. The goals and constraints differ markedly from those of the research community, and bridging the gap between the two worlds requires strategic planning and resources [36]. This section will outline the opportunities, the weaknesses and potential business models and strategies for the real-world deployment of agricultural robotics.

7.3.1 The Opportunities

A strong driver for the adoption of agricultural robotics is the increasing difficulty of finding and maintaining a cost-effective labour supply [36, 91]. This is a worldwide problem but more acute in rich countries. As people’s wealth and opportunities increase, they are less likely to choose the hard and repetitive labour that characterises many farm jobs, even before considering the issue of low pay [27]. In addition, the recent trends to de-globalize and restrict the free movement of labour (particularly in the UK) have constrained supply. At the same time, fierce competition and the market leverage of large supermarket chains have constrained the ability of farmers to raise prices [100].

From private conversations with G Growers, this PhD’s sponsors, it is clear that dependability of labour supply is at least as important a consideration as cost. Unpicked food rots in
the field and decreases yield. So, initially expensive robots may function as the "labourer of last resort" even while their cost per hour exceeds that of a human. The key economic consideration is not the cost of a robot versus the cost of a human. Rather, it is whether the marginal cost of employing a robot to execute a task that would otherwise not be performed is compensated for by the increase in profit that is attained.

Yield is everything. At least 25% of lettuces remain in the field unpicked, representing a clear opportunity for the optimization of margin [private communication, G Growers]. This, one could say, is low-hanging fruit. Part of the solution is precision agriculture: the mass gathering of data on the state of individual plants, via sensors in the field, drones or land-based robots [108, 36]. With the appropriate modelling, the optimum moment and scale of interventions for each plant (weeding, pesticide application, nurturing, harvesting, etc.) can be deduced. But a robot then needs to be able to apply these interventions at the level of individual plants. This requires the localization of plants and the optimization of fleet planning [33].

As the focus moves from the field to the individual plant, new opportunities emerge. The mass production model of agriculture is driven by economic as well as technical constraints. Until now, there has been little opportunity to deal effectively with each separate plant. The result is an environmental wasteland of identically planted crops, sprayed with pesticides whether the plant requires it or not. By applying pesticides selectively based on plant condition and the calculated threat posed by its neighbourhood, overall pesticide use can be reduced for both cost savings and health benefits [57].

By treating plants individually, alternative farming layouts such as permaculture and agroforestry become more feasible at scale. Permaculture proposes the mixing of different crops in the same field to promote better soil use and create a complementary ecosystem for the plants in the usage of nutrients and pest control [66]. Agroforestry goes further, mixing crops and livestock with trees; it claims further benefits in the quality of soil, cycling of nutrients, prevention of erosion and reduction of water loss [63, 58]. Both of these systems necessarily complicate agricultural interventions, but these could be reduced by the use of robots.

Finally, all these benefits can increase the viability of small-scale holdings, allowing for small, specialised farms to compete with the industry leaders. This depends though on the availability of either multi-purpose, flexible robots or robots as a service (RaaS) to compensate for the high capital outlay of purchasing a robot. Even families can participate, with allotment-scale robots such as Farmbot [30] providing a first step to the reclaiming of food supply by the masses (Fig. 7.1).
7.3.2 The Weaknesses

There are, however, many intrinsic weaknesses in the business opportunity and specific challenges to overcome.

The market is indeed worldwide; everyone needs food. However, the cost of labour differs radically between developing and developed countries. Apart from investment resources, one reason for the slowness of adoption of many new technologies in the third world is that the cost of labour is lower, and so the economic advantage of the alternative technological solution must be still greater [43]. In many cases, it is cheaper to simply let humans continue to work on farms as they always have done. The societal impact of mechanization should also not be overlooked [105]. All this reduces the market opportunity to supplying those countries where labour problems are most acute. A smaller market opportunity means less investment.

Compounding this is the fact that every crop, and every crop task, may require an individual, tailored solution. Harvesting lettuces and broccoli heads are two closely-related problems with potentially very different solutions. Harvesting iceberg lettuces may require one solution (e.g., pneumatic actuators to drive a blade [12]), while romaine lettuces allow another (e.g., the use of waterjets [22]) because of their shape. Even within the same crop, existing agricultural frameworks may differ from country to country (soft moist soil in the...
UK vs. sandy soil in Spain [private communication, G Growers]). As the solutions multiply and diversify, the market niches become smaller, as do the returns and available investment. Extensive reuse of platforms and software components, with the minimum of hardware and algorithmic modifications, will be required to keep robotic solutions economically viable.

For the foreseeable future, robots must co-exist with their human colleagues. Until robots are cheaper on a per-completed-task basis than human labour, farmers will necessarily use a labour mix, drawing on more expensive robotic labour once human resources have been exhausted. This increases the need for safety mechanisms and constrains the ability to completely rework the crop environment to optimise for robotic operations. Robots must be robust. Historically, they have been developed for use in sheltered, indoor environments. For many crops, this is not an option. Outdoor agricultural robots must be fortified against the weather and wear and tear, increasing their mass and exacerbating the difficulty of making them mobile. Any agrirobotic business must devise a maintenance solution that allows for on-premise repairs and competes in terms of service level with the networks established by leading players such as John Deere.

There is the difficulty of forcing a change in the capability acquisition model upon the farming community. Farmers pay for labour by the hour used; they do not pay for the creation of the human being. Someone has to cover the large capital investment required to manufacture a robot; requiring this of a business that already suffers from low margins is complicated. If the customer can’t cover the cost of financing, the provider must do so. This is independent of the cost of research and design, which (as stated above) is constrained by market niche size.

Mechanisation generally results in overall benefits for society, but may incur losses for sections of it. Ideally, displaced agricultural workers "share in the machine", but this is rare under capitalism; compensation for lost wages is only likely to occur where that labour is organised [105].

Farming has very low margins, with farms in the UK dependent on public subsidies to break even. Supermarket chains have consolidated into a few massive groups and exert enormous downward pressure on provider pricing [100]. The lower the margin, the harder it is to invest and the more risk-averse customers for agricultural robotics will be.

Robots are hardware-heavy and expensive. While costs have decreased, they are nowhere near following Moore’s law. As such, a purchased robot needs to be as effectively amortised as possible. Apart from other difficulties stated above, there remains the challenge of seasonality. A highly-optimised lettuce-picking robot is of use for just half of the year at best. To be deployed in the other half, it must be retooled for tasks such as weeding, selective nurturing or monitoring.
Finally, agricultural robotics must make the leap from being an area of academic research to being a viable business. The goals of research are different: to advance science, to publish papers or to submit a thesis. Yet research’s contributions are vital for solving technical issues and laying the foundations for a viable business. The constant turnover of students in a research context compares unfavourably with the long-term commitments and institutional knowledge built up in companies. Keeping research focused on real-world problems is key and has been part of the focus of this PhD.

### 7.3.3 Business Models and Strategies

Within the space of agricultural robotics, opportunities need to be carefully evaluated and selected to address the aforementioned weaknesses.

#### 7.3.3.1 What is the product?

First, a prospective company should select whether they provide platforms or solutions. On the face of it, platforms are attractive as they can be deployed to many different market niches. An example would be Saga Robotics’s Thorvald platform (see Fig. 7.2), which purports to solve the general problem of mobility and navigation within fields [48]. Tooling for specific tasks can be attached to Thorvald, making the device multi-purpose and so easier to amortise.

A platform, on the other hand, requires someone to augment it to a full solution. As we have seen, there are many different market niches and each may require a specific hardware and software combination. At one extreme, solutions are so individualised that they become a
services business, for which it is harder to obtain venture capital. Yet without the availability of solutions, who will buy the platform?

This suggests a two-tiered industry, with some players producing platforms and local partners customizing and extending those platforms to produce specific solutions. Both types of players must be economically feasible for the whole to survive. It is possible that to "bootstrap" the business, a company must initially develop both platforms and solutions, which complicates matters.

As mentioned in the introduction, major manufacturers like John Deere have conquered a varied worldwide market by focusing on solutions to generic high-level problems such as mobility. The traditional tractor has always had scale and may have its direct modern equivalent in an autonomous platform like Thorvald. Does the advent of agricultural robotics create any new high-level problems that can be solved with generic products?

As this thesis has stressed, the problems of Detection, Approach and Manipulation are linked but may be tackled separately in the early stages of a project. Could these give rise to generic products?

It is easy to imagine field-robust, self-contained edge devices, combining cameras, depth sensors and GPU processing power that could commodify detection. Such detectors could broadcast labels and camera-centric coordinates of targets, as well as uploading raw images and video to a cloud-based server, where models are retrained and packaged for distribution to the edge.

Approach systems rely on perception, whether supplied by detectors or self-contained. Robot arms are just one available form factor: apples have been harvested using tethered drones by the company Tevel Aerobatics [123], cartesian robots have been applied [30], delta robots are less common in agriculture [1], but are promising for their low cost and weight. Crops may be approached from above, from the side, from below; access may be easy or obstructed, targets may be light or heavy. Each of these use cases could become a separate product line, with its own mechanical, perceptual and control solution.

Manipulation seems, on the face of it, the least generic and most specific of the problems. Embodied solutions imply varied morphologies. Control systems must be developed for plucking, twisting, slicing and many other ways of handling the target fruit or vegetable. Ultimately though, we have the example of the highly generic, supremely useful human hand. Human-level robotic manipulation still seems far off [56], but may one day become flexible enough to become a generic solution.
Another strategic decision for an agricultural robotics business is how to charge the customer. Low margins deter upfront investment. Robots-as-a-service (RaaS) offers charge on a per-hour or per-task-completed basis, reducing risk for the customer but increasing financing costs for the provider [81].

If RaaS is offered, the provider must find alternative uses for the robots with the changing seasons. As stated above, this will mean retooling for other tasks. This will further drive the differentiation of physical platforms such as Thorvald (the equivalent of tractors) from specific end effector/software combinations for individual tasks.

Like any business, an agricultural robotics startup must decide who their customer is. Directly providing solutions to farmers is an obvious path, yet comes with all the stated difficulties. Autonomous robots based on Thorvald or other platforms may not be the best initial solution. It may be better to focus on end effector/software combinations that can be deployed on existing farm machinery such as tractors. While subject to its own engineering difficulties (e.g., strong vibrations) this would allow lower-cost deployment, while sacrificing some of the potential benefits. It may even be better to partner with the existing farm machinery industry giants such as John Deere to leverage their sales and service networks. This provokes the question of who the real customer is: the farmer or John Deere?

Interoperability standards will be key for agricultural robotics to succeed, both at the level of compatibility with existing or legacy physical platforms and at the level of data exchange. Plant status data captured by drones from many different manufacturers needs to be easily interpretable by nurturers and harvesters. Startups will need to contribute to and leverage the implementation of new standards.

Modern startups are encouraged to use the Customer Development and Lean Startup strategies [13, 101]; founders should be in constant contact with their customers to focus on producing solutions that solve real, not imaginary, problems. The slogan is "Get out of the building" – that is, spend more time on site with customers and less in your office [13]. Happily, this dovetails with the strategy outlined in this thesis of iterative development with as much in-field testing and development as possible.

Finally, potential agrirobotic companies should attempt to own and use data as a competitive advantage. There will naturally be a tension with customers over who owns the data, but the target should be at least exclusive access with respect to competitors, if not outright ownership. More data will improve robots’ perception and control as well as allow the automation of edge cases that would otherwise require human intervention.
# Conclusions

## 7.4 An Alternative View of the Industrialisation of Agriculture

It is beguilingly easy to view the world through the lens of one’s own profession. Agricultural robotics, as a discipline for the direct replacement of human labour in the field, is but part of a wider set of solutions to increase yield and lower the production cost of food. There may be alternative solutions already available, or yet to be developed, that greatly reduce the dependency on, and the complexity of agricultural robots.

### 7.4.1 Factories instead of fields

Factories have long standardised and removed variability from pre-existing production processes. Smart, data-driven factories are the core of the "Industry 4.0" vision; with decentralised, flexible and efficient production [69]. To what extent could the same happen to agriculture? Protected cultivation, the use of greenhouses and polytunnels to shield and nurture crops that would naturally grow in the open is one step in this direction. It is increasingly used in countries where the environment either never was suitable for the crop, or no longer is due to climate change [115]. A more radical approach is to dispense with the field altogether, and to grow crops in indoor structured environments such as vertical farms, using techniques like hydroponics [16].

Hydroponics promises improved cultivation performance on many fronts: up to 90% reduced water usage, 90-99% reduced land usage, reduction of the risk of infection and pests, easier soil and nutrient monitoring, reduced time between planting and harvesting, year-round production and a generally more controlled and less harsh environment [16]. The spectral properties of LED lighting can also be optimised for photosynthesis [23]. Lettuces can be economically grown using hydroponics in countries such as India that are less suited to the crop than the UK [75]. Yet hydroponics isn’t yet anywhere near taking over from soil-based production. The economic resources of farmers plays a part. Capital costs are high: equipment is expensive to acquire and requires constant maintenance. A field has often already been paid for. Energy costs for hydroponics are high, whereas the sun is free [16]. Crop access may be an issue. Monitoring must be constant to avoid catastrophic crop failure; field-grown crops are robust to periods of neglect. Used hydroponic solution must be processed to avoid environmental contamination [75]. Not all crops are suitable for hydroponic cultivation. Crops must still be harvested, but a structured, controlled environment simplifies the problem.
As with other aspects of farming, different crops and differing local circumstances will likely drive a mix of farming formats over the industry. Rather than replace soil-based production, hydroponics and urban indoor farming will complement the traditional field based solutions.

### 7.4.2 Structuring fields

Even in fields, variation can be reduced and yield increased through precision farming. A 2020 McKinsey report estimated the potential combined value creation of smart-crop monitoring and drone farming to be almost five times that of autonomous farming machinery [72]. IoT-based soil monitoring can monitor mineral content and predict fertility; moisture detectors enable closed-loop irrigation, reducing costs and improving yields; nutrient monitoring can modulate the application of fertilise; multispectral analysis by drone can monitor plant health [23].

In some countries, cultivars themselves can be genetically modified to improve their growth characteristics, although this is subject to consumer wariness and legal barriers in others. Genes for pest resistance can be imported from other species (transgenics), existing genes modified (gene editing) and new genotypes may simply be bred [23].

The further industrialisation of agriculture is subject to some limits. In the developed world, there is some backlash against farming methods perceived to be unnatural. There is clear evidence for the impact on biodiversity and soil quality of agricultural monocultures; perennial polycultures are one potential answer to this [29]. Again, the result is likely to be a mix of techniques, with different trade-offs taken in different geographical locations and different economic circumstances.

The development of precision and indoor farming will progress in parallel with the increasing sophistication of robotics, and the resulting implementation mix at any given time will depend on the state of the art and economics of both. Robots will become increasingly adept at operating in unstructured environments while the opportunities to restructure natural processes will increase.

### 7.5 Conclusion

“You know, farming looks mighty easy when your plow is a pencil, and you’re a thousand miles from the corn field.” – Dwight D. Eisenhower

The day before, the Vegebot’s end effector had exploded on first contact with a lettuce, scattering acrylic shards across the lettuce lanes and rendering the crop in that corner of
the field unsaleable. Now, under a dark grey sky and light rain, the wind was blowing hard, rocking the overhead camera back and forth. Confused, the mechanical arm lunged repeatedly, missing its targets. It had not been a good week. Then, with a peal of thunder, a bolt of lightning struck a nearby tree. The author and collaborators were stranded in the middle of a flat, desolate field, next to a large, malfunctioning metal robot.

Sometimes it can be hard to see the outlines of a revolution when you’re in the middle of it. Yet a confluence of new machine-learning techniques, cheaper sensors, computing power and demand driven by labour shortages have kickstarted a new surge in the mechanisation of agriculture. Robotic solutions to problems that have been intractable for decades, such as lettuce harvesting, suddenly appear within reach.

Yet robotics, when applied to problems in the real world – and farming is nothing if not real – is so much harder than even the technologically sophisticated would expect. The touching optimism of new entrants to the field contrasts with the battle-weariness and cynicism of experienced researchers and engineers. Humanoid robots topple over when trying to walk a few steps on a perfectly flat surface. Brushing aside leaves to find and pick a strawberry challenges the most advanced robots.

Fortunately, the world is full of creatures who have already solved these problems. A mountain goat leaps nimbly over the rockiest terrain, inspiring admiration and occasional flashes of resentment in the researcher toiling over locomotion algorithms. Birds not only brush aside leaves but dodge humans and penetrate netting to easily reach the hidden strawberry. They are not computers with sensors and actuators: they are physical creatures with shaped bones, flesh and feathers. They are embodied. We can profitably study how these creatures succeed and, without copying slavishly, discover what techniques and principles they offer to inspire us.
References


