VALIDITY OF LIDAR BASED IOS DEVICES FOR ON-TREE FRUIT SIZING

A TECHNICAL REPORT SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF COMPUTER AND INFORMATION SCIENCES

Supervisor

Associate Professor Minh Nguyen

Submission Date: 15/11/2022

By

Kooshan Aryana

School of Engineering, Computer and Mathematical Sciences

Abstract

Fruit sizing systems are an emerging technology in the agricultural industry, specifically orchard management. Fruit sizing systems employ various computer vision approaches such as those based around thermal cameras, RBG-D cameras, and LiDAR. Most of the current systems for fruit sizing occur post-harvest and the current On-Tree or in-field fruit sizing is commonly done manually which costs valuable time, labour and can be costly in cases where orchard owners outsource the work to alternative expert services. Furthermore, the reliance on said services can cause issues to arise such as miscommunication, missing appointments for sampling periods, or general lack of control. For this reason, this thesis proposes a novel proof-of-concept application that involves the employment of a LiDAR-based IOS device for On-Tree fruit sizing. This thesis documents the development of said approach and contains data derived from simple experiments to gain answers as to whether this proof-of-concept application is a viable method for On-Tree fruit sizing. The results show a positive precedent as the On-Tree apple experiment as well as the cherry bunch experiments demonstrated error range percentages within 6% between 0.5m and 2m away from the apple. Furthermore, the post-modification apple experiment was conducted on the same apple data-set produced error range percentages within 0.5% between 0.5m and 2m. The post-modification experiment purpose highlights the requirement for a dynamic approach to the exponential regression best-fit parameters applied. An alternative experiment was done to test the long-range capabilities of the proof-of-concept as well as determine the

reliance on accurate pixel width measurements. The results of the said experiment are promising, showing an error range percentage within 4% between 0.5m and 5m. As the machine learning model is derivative based on [1] provided by Hectre, the results for the detection count experiment were lackluster demonstrating odd values such as $29/52$ cherries detected. The limitations of the project are the heavy reliance on an accurate calibration of the parameters as well as an accurate pixel width of fruit (aka bounding box measurements from the machine learning model). Overall, the benefits of the proof-of-concept include lightweight accessibility, high sample rate, cost-efficient as well as competitive with alternative fruit sizing systems and far exceeds that of manual intervention-based On-Tree fruit sizing.

Contents

Abstract **2**

Attestation of Authorship **10**

Acknowledgements **11**

1 Introduction **12**
1.1 Summary 15

2 Background and Literature Review **16**
2.1 Light Detection and Ranging (LiDAR) 16
2.1.1 LiDAR Applications 18
2.1.2 Automated Vehicles 18
2.1.3 Agriculture 19
2.1.4 iPhone/iPad and LiDAR 21
2.1.5 Miscellaneous Applications 22
2.2 Object Detection 23
2.2.1 Viola-Jones Object Detection 23
2.2.2 Scale Invariant Feature Transform (SIFT) 25
2.2.3 R-CNN, FAST R-CNN and Faster R-CNN 30
2.2.4 Single Shot MultiBox Detection (SSD) 39
2.2.5 Object Detection Significance 46
2.3 Agricultural Developments 47
2.3.1 Smart Agriculture 47
2.4 Summary 50

3 Methodology **52**
3.1 Research Methodology 52
3.1.1 Research Questions, Aims and Goals 52
3.1.2 Research Design and Methods 53
3.2 Proof-Of-Concept Application Development 58
3.2.1 Front-End Development Components 58
3.2.2 Back-End Development Components 62
3.3 Summary 74

4 Results and Findings **75**
4.1 Results and Findings 75
4.2 Retrospective 76
4.3 Summary 79

5 Discussion **84**
5.1 Findings Analysis 84
5.1.1 Apple Experiments 84
5.1.2 Cherry Experiments 86
5.1.3 Flat Surface Manual Intervention Experiment 87
5.1.4 Count Experiment 87
5.1.5 Interpretations of Findings 88
5.2 Validity and Viability 93
5.2.1 Overview Goals and Failed Attempts 93
5.2.2 Realistic Benefits 99
5.3 Limitations and Future Improvements 101
5.3.1 Limitations 101
5.3.2 Future Considerations and Improvements 104
5.4 Summary 106

6 Conclusion **107**

References **110**

List of Tables

3.1 Example Calibration data gathering 1.25m away from object. 70
3.2 Example Calibration data gathering 1.60m away from object. 71

4.1 Findings from Apple Experiment 1 76
4.2 Findings from Apple Experiment 2 77
4.3 Findings from Cherry Experiment 1 78
4.4 Findings from Cherry Experiment 2 78
4.5 Findings of Flat Surface Manual Intervention Experiment 80
4.6 Findings Post Modification Applied to the Apple Experiment 1 Dataset 80

List of Figures

2.1 Is the original figure from [2] which covers the basic parts of the object detection process. Refer to [2] for further details regarding Regions with CNN features. 33
2.2 Represents the architecture from the original Fast R-CNN paper [3]. . 34
2.3 A fully connected network representing the RPN module as the staple addition to the existing former architecture as further detailed in [4]. . 35
2.4 the improved faster R-CNN method for the use case in [5]. 37
2.5 Represents the SSD model utilising the base network as VGG-16 network. Comparatively to the YOLO model [6] tests done on VOC2007 show that it outperforms in speed and accuracy [7]. 40
2.6 This is the figure provided by the original paper that illustrated the 5 extra deconvolusion layers [8]. 42
2.7 Improved modified SSD target detection model [9]. 44
2.8 Provided by [10] is the simplified model which utilises the VGG16 partial void of the convolutional layer. 46

3.1 This is the general high-level view of the pipeline from data acquisition to diameter measurement gained. 59
3.2 Contained in the Renderer.swift file, is the contents of the LiDAR data generation using the AVFoundation framework [11], as well as the storing methods for the data packet for exportation. 60
3.3 The ViewController.swift is common on most swift programming that is responsible for the construction of the front end artifacts. This is the button in which the capture will commence upon pressing. 61
3.4 This is the initial view when the application is launched. 61
3.5 The LiDAR representation drawn on the screen originally derived from the base code. 62
3.6 Share functionality allows for the user to choose where the information gets sent. This makes it much more flexible for some users. This will change in future iterations of the application to increase the speeds of the capture time. 63
3.7 Shows the type of files that are being saved. This includes the LiDAR .txt file, depth map representation of the LiDAR information as well as the original image in JPEG format. 64

3.8 The above consists of one of the essential inputs for the complete process to work correctly. 64
3.9 This function is, as mentioned, responsible for the determining the estimated distance from the object of interest i.e., fruit centre. 65
3.10 This function much like the previous, serves to procure a useful value from the input data frame derived via the machine learning component of the project. 66
3.11 The figure indicates the bounding box (black) co-ordinate corners given initially (red), the shifted corner (purple) and the wanted length indicated by x (blue). 67
3.12 68
3.13 68
3.14 Is the amalgamation of all functions bound together included with it some of the separation techniques to organise the data frame data that required specific attribute type alteration. 69
3.15 Each increment on the paper placed on the wall is an increment of 50mm. 71
3.16 This is the makeshift tripod, used to attempt to eliminate the shaking that occurs when holding a device manually causing human error. . . 72
3.17 Graphical representation of the 3 main variables calibrated based on collected measurement data. 73

4.1 An example capture taken of an apple 0.5m away from the camera origin. The Depth map (right image) represents all LiDAR data per pixel gained from the IOS device. 75
4.2 Graphical comparisons between attributes in 4.1 76
4.3 Graphical comparisons between attributes in 4.2. 77
4.4 An example capture taken of a cherry bunch 0.3m away from the camera origin. The Depth map (right image) represents all LiDAR data per pixel gained from the IOS device. 78
4.5 Graphical comparisons between attributes in 4.3. 79
4.6 Graphical comparisons between attributes in 4.4. 79
4.7 This is the specimen of interest in the captures conducted in 4.5. . . . 80
4.8 Graphical representations of 4.5. This is discussed further in 5.1.3. . . 81
4.9 Graphical representations of 4.6. This is discussed further in 5.1.5. . . 81
4.10 This experiment is responsible to test the machine learning model that is derivative of this paper [1]. The total cherries in this bunch are a total 52. The bounding box have indicated only 29 detected cherries. 82
4.11 This experiment is responsible to test the machine learning model that is derivative of this paper [1]. The total cherries in this bunch are a total 52. The bounding box have indicated 71 detected cherries. 83

5.1 Simple if-else statements to apply rudimentary dynamism to alter parameters based on distance from object of interest. 91

5.2 The initial point cloud approach to gain the distance information. The bottom sector of the figure displays the inaccuracies of this approach as the matching does not line up with the image. 95
5.3 The conversion values (purple). 103

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of student

Acknowledgements

Firstly, I would like to thank my mentor Dr. Minh Nguyen of Auckland University of Technology for diligently being supportive throughout the entire process of constructing this thesis. Furthermore, the team at Hectre, specifically Anwar Iqbal, Luke Butters, Dilum De Silva, Ariel Burgos, and (former team member) Zhaohan Xiong as without their constant professional support this project would not reach its final state. I would like to acknowledge Auckland University of Technology as well as Callaghan Innovation for providing funding and resources to fuel this research. Finally, I would like to thank my family and friends for providing constant motivation during the entire process

Chapter 1

Introduction

The agriculture industry continues to increase its endeavor to become more autonomous day by day with implementations such as [12], [13] and [14]. There remain some realms of the agricultural industry that rely on outdated standards such as fruit sampling. It is common for those who own orchards to outsource the sampling process of fruit to expert services. Externally employed manual sampling strategies can falter to common problems such as miscommunication, rescheduling causing sampling miss timing issues and can cause lack of control to farmers [15]. Even if such issues do not occur in a scenario, the data gained is hardly encompassing the full yield as discussed in [16]. Technological improvements can help to gain more data and in turn provide a larger yield for fruit orchards.

LiDAR technology is an ever-growing field in research, specifically in computer vision spanning several industries such as agriculture [17], robotics [18] and security [19]. The main draw to this sensor is its high accuracy capabilities and its newfound accessibility in recent years [20], [21]. Furthermore, companies such as Apple have created affordable devices that hold such sensors at the tip of a user's fingers. The rapid boom in artificial intelligence is also incredible in creating amazing tools for fruit counting [22] and even early attempts at fruit sizing [1]. In agriculture, computer

vision techniques harnessing artificial intelligence are employed for size estimations commonly applied during the post-harvest process for sorting or grading purposes as in [23], [24] and [1]. Yet, On-Tree or in-field fruit sizing is usually done manually, which can cost large amounts of time, labour, and money in cases where the work is outsourced as well as faltering to human error [15].

Existing works in for in-field size estimations have been common in the current research space. Varying approaches include thermal camera-based methods [25], RGB-D camera-based methods [26] as well as LiDAR-based methods shown in [27]. The RGB-D-based method [26] was capable of gaining results such as R^2 = 0.95-0.96 when compared with manual standards for mango measurements. Previous works such as the [27] and [28] employ LiDAR as the source for the size estimations where [28] was capable of keeping error range percentages no higher than 9.5%. Despite the beneficial accuracy results surrounding LiDAR, one limitation seemingly hold these approaches hostage. The limitation is the accessibility and the equipment complexity as mentioned in [27]. Hence the mission of this thesis document.

This paper proposes a proof-of-concept application that involves the usage of the LiDAR functionalities on the recently available IOS devices as the main method of gaining accurate distance values to then provide the diameter of a fruit detected via a machine learning model. These diameter measurements from the application will in turn provide simpler sampling for On-Tree scenarios to circumvent the previously established concerns with external expertise services. The LiDAR sensors on said devices have a maximum functioning range of 5m. This maximum of five metres is more than enough for an application to exploit the high accuracy to gain sizing estimations of fruit located on trees without manual intervention. One alternative hope of this research is to set a precedent for those who manage orchards to have the power to make informed decisions effectively without the requirement of outsourced expertise or reliance. This application would also severely cut costs involved in large LiDAR-based

equipment as mentioned earlier in [27], [15]. Computer vision techniques surrounding machine learning and object detection are also a large component in this paper as the provided machine learning model will be based on [1] as it is currently being used for post-harvest size estimation at Hectre (Orchard Management Company). With the ever-growing strides that machine learning has taken, strides towards higher accuracy can be made in On-Tree scenarios for fruit size estimations. The main objective of this paper is to examine the validity of LiDAR-based IOS devices for fruit sizing estimation. Byproduct hope is to create an efficient and comparable proof-of-concept that can compete with the previously e mentioned estimation approaches. The potential of the proof-of-concept application is highly positive as the amount of data that could potentially be gained is far exceeding at least that of manual standards.

This thesis will contain several sections first of which is the background and literature review which will cover diverse topics regarding machine learning smart agriculture as well as common architecture that revolve around convolutional neural networks. the following section labeled methodology will outline the developmental processes taken to develop the application itself as well as provide the methodology in which the data collection follows. Furthermore, this section will also outline the main goals of the project and will provide a clear concise aim for this proof of concept to achieve. the proceeding finding section will comprise tables and graphical representations of the results determined via the simple experiments done on fruits such as apple and cherry bunches. Included in the finding sections are also counting tests and post-modification experiments highlighting the possible improvements to the current proof of concept application. The discussion aspect of this thesis will consist of information regarding the findings analysis as well as exploring the limitations and future improvements to be made for this proof of concept. finally, the conclusion will provide a concise summary of all the aspects explored in this thesis as well as provide the implications of such proof-of-concept application developed.

1.1 Summary

To enclose, this chapter briefly summarised the components of this thesis that will fleshed out in later chapters. Essentially highlighting relevant road-map for the thesis goal of creating a proof-of-concept application that provides high efficiency in the field of fruit sampling for On-Tree scenarios.

Chapter 2

Background and Literature Review

The following section will comprise assorted topics that are either involved in the proof-of-concept applications development or adjacent to it. The main aim of this section is to demonstrate a general understanding of the topics to provide further insight later during the developmental process of the application.

2.1 Light Detection and Ranging (LiDAR)

As already established, LiDAR technology will be the focus of the project, hence exploring recent developments would help outline various limitations and advantageous aspects. LiDAR can be considered a sister sensory method to that radar and sonar methods. LiDAR is commonly recognised as being of higher fidelity, having a larger detection range than that of the other sister methods [29]. While the other mentioned methods use sound and radio wavelengths as detection, LiDAR relies on the use of light. To elaborate, LiDAR sensors function much like its acronym entails – light pulses are projected from the source of the sensor aka transmitter and the reflected laser light is returned via bouncing from objects in the vicinity [30]. The light that is returned is caught by a receiver and the below calculation known as the time of flight (TOF) is used

to map out surrounding environmental obstacles.

$$D = \frac{c \times t}{2} \tag{2.1}$$

Where D is the distance from an object, t is the time taken for the light to be received and c is the speed of light. The division of 2 is done to account for a double trip to the object and from the object.

Although LiDAR has great benefits drawbacks are also present. Specifically, adversarial weather conditions can cause early spurious debris detection that in turn leads to inaccuracy as explored in papers [31], [32]. Another study, [33], resolved that the detection of objects was reduced by 50% in weather adjacent to fogginess. Back-scattering is the action that light does when reflected off a surface. Back-scattering can determine various aspects of a target depending on the intensity of the returning signal [32] (pg. 678). LiDAR can employ two kinds of detection schemes, coherent and incoherent energy detection. The latter measures amplitude changes of the light that is back-scattered whereas the former provides accurate measurements of Doppler shifts or phase changes in the back-scattered light. In the context of light, it is important to note that coherent means that photons in light oscillate at the same frequency whereas incoherent is the opposite of this. Hence the various applications that LiDAR can cover. For example, coherent light exploits optical heterodyne detection [33] (pg. 32) methods which are much more sensitive than that of direct detection or incoherent and allow the operation to run on lower power (however it will require more complex transceivers). Laser components in a LiDAR sensor are a customizable aspect depending on the applications. It is common for lasers to range between 600-1000nm for non-scientific tasks. A common trend for these lasers is that they should be eye-safe (for terrain mapping on air crafts).

Despite some of the drawbacks mentioned earlier, LiDAR has taken large strides

into becoming a leading sensory technology in many industries such as automation, agriculture, civil engineering, and 3D technology.

2.1.1 LiDAR Applications

LiDAR has historically been a bumpy technology where in some cases it was cheaper over other sensory technologies [34] but the sister sensor, radar quickly surpassed LiDAR as production costs for it had decreased since its inception in the 1960s, resulting in a slow development of LiDAR technology. Despite the slow development of LiDAR since 1995, the advent of automata and robotics has revived the interest in LiDAR. Major developments in simultaneous localization and mapping have been made by implementing LiDAR sensors to accurately recreate environments for navigation [34]. Further developments are discussed in the proceeding sections.

2.1.2 Automated Vehicles

Automated vehicles have been the subject of large conversation in recent years, more so related to their efficacy in real-world settings. One manner of how Autonomous vehicles function is through the extensive implementations of sensors. The paper [35] highlights the usage of point cloud technology derived from LiDAR sensors for automated vehicles. The understanding of external obstacles is imperative in car automation to make correct decisions when navigating the road. LiDAR sensors have been known to create 3D point cloud representations of environments and objects with high precision [36], [37]. The paper proposes a novel way to reduce the storage requirements when using point cloud information for road scenes as well as increase the efficiency computationally of processes [35]. It is a 3-step process, and it is as follows:

1. Point cloud data filtration through linear interpolation.

2. Statistical method applied for removal of outliers.

3. Down sampling LiDAR point cloud data using a voxel grid filtration.

Due to the sheer number of points that can be received via LiDAR sensors, the above method is applied for the reduction of the unwanted noise [35]. The paper showed a reduction of 50% in data volume without losing valuable environmental information [35]. Another paper [38] also implements LiDAR for increased safety of electronic-controlled vehicles. This paper acts as a survey to highlight the efficacy of the tool being used in scenarios for accurate measurement regarding long distances. An experimental model has been applied to determine that the distance of LiDAR between 0.4, and 12m is of significant accuracy despite colour change [38]. Also, if the signal of the returning LiDAR is less than 100 units it is considered unreliable for use [38]. The paper acknowledges the shortcomings of LiDAR which can be affected by light intensity, returning intensity, fragmented noise as well as reflection or colour of the target object and determines that it is a necessary step to further the LiDAR sensor to ensure safe application in automated vehicle systems like braking systems, etc [38].

2.1.3 Agriculture

As the focus of this project is agriculturally adjacent, the paper [39] uses LiDAR for tree row volume estimation. The reason for this estimation is to optimise the distribution of phytosanitary products in orchard locales. The paper proves to be far more practical than initially expected, combining hydraulic systems with 12 controllable nozzles that are responsible for the regulation of the amount of phytosanitary product required for application [39]. The LiDAR sensors are responsible for the even distribution. This distribution is determined via the Tree Row Volume as shown below:

$$TRV = \frac{a \times h \times 10000}{c} \tag{2.2}$$

In this equation a is referring to the tree canopy diameter, h is referring to tree heights and C is the distance between the tree rows [39]. For the nozzle control of the spraying mechanism, a small needle is implemented to induce smaller or larger holes [39]. The preliminary tests of utilising LiDAR as the mediator of broth application, show a reduction of 50%, although further testing is required for validation [39].

Humanitarian technologies are referred to as technology that can increase the efficiency for humans such as the following [40], [41]. Much like that of this project, smart agriculture is a part of that category of technology [42]. The LiDAR sensors are not restricted to physical alterations of the physical realm, but it is also capable of creating meaningful data for analysis that is more perception-based much like this paper [42]. The paper proposes a method to determine the ripeness of Oil Palm Fresh bunches. Much like the data collection tied to this project, it is bound by basic standard .csv packaging for further analysis [42]. The system proposed utilises an Arduino UNO and two servo motors as the standard components [42]. LiDAR in this case is used to collect the data and compare the mean intensity values as this is what will determine the ripeness of chosen fruit [42]. The proposed method is shown to be successful as it can create point clouds from the oil palm fresh fruit bunches and has given valuable data as to whether it is a ripe bunch or not [42]. This is determined by the mean intensity values, lower being ripe while higher being unripe. The point cloud reconstruction of the fresh fruit bunches was done via LiDAR Lite V3 deployed like that of a scanning system on the Arduino UNO [42]. The proposed method is considered novel at this stage and suggests further future development for deployment on drones for an easier mobilised scanning system [42].

2.1.4 iPhone/iPad and LiDAR

As well established, this project makes use of the IOS LiDAR capabilities to capture valuable information for back-end processing. Although sparse papers such as [43], [44] are good examples of the current evolution of LiDAR technologies. Sensors that initially were difficult to obtain are now accessible to everyday users to test, develop and create solutions. The potential of the smartphone in surface detection or mapping is already prevalent as mentioned in the following [45], [46]. With the advent of the iPad Pro 2020, LiDAR has become commonplace and accessible to the consumer. Since its release, the sensor has been tested and has shown to prove decent quality results when detecting tree stems above a 10cm diameter threshold [47], [48]. Testing was done in the context of geomorphological, geological, forestry, and archaeological related industries. It was determined that the primary application use case for mentioned context would range from small to medium scale situations – i.e., centimetres to several hundreds of meters [43]. Although, the iPhone may be of advantage due to its mounting capabilities. The paper highlights that the inertial measurement unit may have caused the inaccuracies of the iPhone reading of the distance measurement and is not related to the LiDAR scanner itself [43]. The limitations are also discussed arguing that only small to medium-scale study sites are viable options for the LiDAR scanner to be used as the limitation of range reduces the applications [43]. When and if the power supply for handheld devices increases in standards, the range has the potential to increase. This is highly promising for fields outside of photogrammetry and opens doors to many more solutions much like this current project being proposed (referring to the proposed On-Tree Fruit Sizing Application).

2.1.5 Miscellaneous Applications

Larger scale studies for terrestrial management are also commonly using the LiDAR sensor to measure the elevation of the terrain such as in this paper [49]. The time required for conventional methods is extensive and involves the preliminary tasks of vehicular traffic safety considerations as well as dense vegetation conditions must be considered [49]. LiDAR poses to be a superior alternative to the above methods as it can do the photogrammetry-based unmanned aerial vehicle (UAV) scans of elevation on the land below which is precisely what the paper [49] aims to achieve. The paper assesses the efficacy of the UAV-based LiDAR data to be used for specific tasks surrounding highway design. The study suggests that agencies should start to move towards making increased use of remote data sources like that produced by terrestrial LiDAR for tasks such as highway design [49]. Future endeavours that were mentioned for the extension of this study are to tie in more focused highway design software to further the quality of the results and for validation purposes [49]. Despite this the results exclaiming the viability of LiDAR is promising.

There are many more applications that LiDAR has achieved such as the monitoring of pollution [50] and cloud density via atmospheric research [51]. The one unique aspect of LiDAR that has not been mentioned is the versatility of the laser scanning itself. Light is capable of many forms such as ultraviolet, visible infrared, or non-visible infrared and can be applied to the LiDAR paradigm for usage. The flexibility allows for non-reflective metallic surfaces, boulders, clouds, chemical compounds, and astonishingly single molecules to be converted into valuable point-cloud or standard CSV data [52]. Much like the previous paper [49], it is quite common for regular terrestrial research to be conducted through aerial scanning of the terrain such as done in [53].

2.2 Object Detection

Object detection is a subcategory of computer vision and image processing that employs various techniques to detect semantic objects of a specific type i.e., cars, trees, and buildings in digital media [54]. The various domains containing significant remnants of object detection are video surveillance, land surveying, and in various automation applications. There are several approaches for achieving object detection and many of them have their advantages or disadvantages depending on the situations in which they are used. Below we will briefly cover the history and timelines of some of the approaches conceived over the years for object detection purposes.

2.2.1 Viola-Jones Object Detection

This approach was a machine learning object detection framework that was conceived by Paul Viola and Michael Jones [55], [56]. The algorithm was conceived due to the nature of the problem at the time as a method for face detection as mentioned in their paper [55], [56]. This method is by far lower than the convolutional neural network methods that we have today, however, because of its compact nature it is used in specific use cases that require lower power systems such as those mentioned in the original paper, the Compaq iPad (2 fps device) [55]. Lower computational power means it can be compact and can be used for devices that are not meeting requirements for other more robust methodologies, broadening options for engineers or developers.

The problem description for this detection approach was as follows – In a photograph, does it contain a face or faces, and formulate bounding boxes for those faces? Face detection is considered as both a localization problem fused with a binary classification problem. For this task, the Viola-Jones algorithm only detects faces with restrictions such as full views, front orientations, no rotation, occupation of most of the frame and fixed resolutions [55], [56]. While these restrictions may seem to hinder the

viability of such an approach the use cases can be considered quite wide. This is due to the ease of picture processing that can be done to match the requirements needed for correct readings. Images can be scaled, rotated, processed by brightness, processed with blob detection – a method to locate an object within a larger image, and window markdowns of faces within a view. The only true restrictions for this approach for face detection would be the frontal view and the full-view requirements.

A deep dive analysis [57] explores the algorithm for this approach in depth. Viola-Jones is a boosted feature learning algorithm, although quite basic, it is light on computational power and praise to the custom adaptive boost, which can create exceptionally reliable classifiers out of copious amounts of weak ones [57]. When an image is inputted into the algorithm and produces at any point a value of 0, it means there are no faces detected. Opposingly, if all the classifiers return a 1 as the output, the algorithm returns an indication of successful face detection. Viola-Jones classification is also named the Haar cascade classification.

The name Haar-like features are derived from the mathematical sequence Haar wavelet – which are square-shaped functions. The Viola-Jones approach to object detection modified and adapted the idea of Haar wavelets to produce Haar-like features. Haar-like features function as explained below.

- A series of rectangular shaped regions at given locations are detected in a scene.
- The pixel intensities are added together for each region in the scene.
- Finally, the difference between the sums of each of the regions are gained.
- The gained differences are used to separate greater regions of the input image.

In the Viola-Jones object detection architecture, these differences are the necessary attributes to determine the separation between non-objects or objects. Due to the nature of Haar-like features being a lesser classifier, a substantial number of them are required

to produce reliable segmentation [55], [56]. Hence the name Haar Cascade Classifier. The cascading nature of this algorithm gradually increases in complexity, therefore, achieving greater results for detection rates, Although the results are sufficient for images, it is not as viable for real-time functionality. To circumvent this issue, the strong classifiers are arranged in a succession of an order of complexity along the cascade. This means that sub-windows that require no further processing will move on to the next sub-window, resulting in a degenerate tree. In face detection, the initial classifier within a cascade, namely the attention operator, requires only two features to achieve a false positive of 40% and a false negative of 0% [56]. Finally, if object tracking (specifically in the video) is a necessary task for a project, options such as the KLT algorithm can help to reduce the detection to that of simple salient features rather than resorting to a per-frame detection basis like that of the Viola-Jones detection framework [58].

2.2.2 Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) is an algorithm commonly used in computer vision for detection and local feature descriptions in images [59] (pg. 112). For simplicity, the following steps comprise the steps in which SIFT functions.

1. Key point or Feature point detection: The scale invariance refers to the inherent value of an object or objects in the world such that they exist only in meaning when observed at certain scale ranges [60]. An example of this would be to consider a pebble. The pebble is only relevant in the meaning of scale of a few centimetres at most. If one were to view the pebble at the nanometre level, it would be pointless as by that point it would be more relevant to explore the molecular constructions of said pebble instead. If on the other hand, the pebble was viewed from kilometres away, the scale would be considered unintuitive as by

that measure it would be more relevant to view the surrounding environment. In the mentioned example, the emphasis on the relevant scale is made in connection with the object of interest. Scale invariance in an image is the general diameter within said image denoted by σ which represent the pixels counted. The image in question is blurred using a Gaussian kernel. After this, the sum of the Laplacian or the second-order derivatives is calculated with the blurred version which locates the edges within. The LoG (Laplacian of Gaussian) is commonly implemented for the blob detection process. To gain the true scale invariance, the multiplication between the LoG and $\sigma 2$ is needed to retrieve the maxima or minima.

2. Key point or Feature point localization: After the rough localization of the key points, they are further refined in the sub-pixel localization step. This step involves removing any features that are unwanted while increasing the definition of the prior localization [59].

3. Positional Allocation: Histogram of Orientation or HoG is used to create thirty-six bins covering a 360-degree set of orientations. This orientation histogram is formed around the key point localized in the prior step. Each of the samples that become inserted into the histogram is weighted by a Gaussian round window and its own gradient magnitude with a σ that is 1.5 increased the scale of the original key point [59].

4. Generation Feature Descriptor At this stage, the algorithm provides a list of feature points that are portrayed with attributes such as orientation, scale, and location. The descriptor generation involves a normalized vector of 128-dimensions. The attributes allow for the feature points to be recognised from varying perspectives. The descriptor is formed via the implementation of a 4x4 grid of gradient angle histograms as it is dealing with a gray-scale gradient of the original image [59]. The key point determinations as before such as the orientation and centre location

are applied and are significant in altering the dimensions of the grid. The angle histograms are divided into 8 parts and are contained under each spatial bin [59]. Miss-registered errors are avoided using the Gaussian window, which only allows those closest to those gradients closest to the centre [59]. The histogram formation also employs the trilinear interpolation [59].

Applications of Swift

SIFT in the current era is lesser used compared to that big data computer vision approaches such as convolutional neural networks. Despite this, researchers have employed and improved the existing SIFT algorithm for various specific use cases.

The paper [61] has proposed an improved SIFT algorithm for stitching images retrieved via UAVs. The usage of this is for achieving fast and accurate detection of straw cover over expansive areas. Slightly contrary to the prior statements made in this review, this paper states the common concerns of the standard SIFT algorithm such as...

- High dimensionality of feature descriptors.
- Strenuous computational effort.
- Low matching efficiency.

This is assumed to be comparative to current standards of object detection surrounding the neural network realm as well as the use case of fast stitching. Regardless, the study employs the following upgrades to improve performance.

- Down sample high-resolution images prior to feature detection: This helps to reduce the mass of the feature points which in turn increases the efficiency of feature extraction [61].

- Gradient normalised feature descriptors used to match among features: Feature matching is done via MN-SIFT feature descriptor which holds better texture features that improve the matching accuracy for the use case[61].

- Implement Progressive Sample Consistency Algorithm (PROSAC): The PROSAC algorithm exterminates the false positives produced, further improves the accuracy of the feature point, and determines the transformation matrix model employing the least-squares approach [61].

- Image fusion with optimal stitching combined with fade-in and fade-out. A smooth transition between overlapping regions and non-overlapping regions of the images is achieved via a fading-in and fading-out method [61]. This is to present common issues when stitching images automatically together such as misalignment, stitching gaps and ghosting [61]. Furthermore, the quality of the images is preserved during this process.

The paper concludes the findings to be successful in achieving the major goals of the study. The proposed method was capable of accurately and in a fast manner retrieving straw target feature points from the high-resolution images provided by the UAV. The method was also capable of distinguishing between the object of interest i.e., straw amongst noise in complex background situations. The experiments that were conducted as part of the study using the proposed method determined the following.

- The proposed algorithm compared to that of the other algorithms reduced the feature point number by 90% and 58% [61].

- Compared to the traditional SIFT algorithm and the SURF algorithm, the proposed algorithm held matching efficiency of 22.18% (exceedingly improved) [61].

- The detail of image integrity was preserved thanks to the employed image stitching methods [61].

Due to the rapid UAVs and this approach to image stitching achievements in real-time automatic panoramic images for crop assessment will be frequent in occurrences with the integration of machine learning.

Next, is a study [62] dated in 2015 yet amid the big data boom researchers still opted for SIFT as a method. In this use case Dense SIFT feature descriptors to identify patterns of an object. This feature descriptor is combined with extreme learning machine (ELM) [63] which is a classifier that can identify effectively, vehicles amongst noisy backgrounds as it provides sufficient general performance. The cost of ELM is said to be cheap for training and testing, thus the system can function in real-time scenarios. The paper concluded that the proposed method showed increased performance for detecting vehicles moving at increased speeds which tends to obscure vision and could possibly circumvent the need for motion cameras [62].

The study [64] is a recent paper that proposes a four-part novel approach, that includes...

1. The construction of a 3D Gaussian pyramid to obtain multi-scale pollen images
2. Computation of local differential vector for identification of key points.
3. Filtration of said key points via an inner layer contrast.
4. Finally the extraction of the statistical histogram of the key points as singular features for automatic classification of 3D pollen images [64].

As the paper proposed, the Spatial Local Key Point descriptor extraction method based on SIFT was successful in the classification of pollen recognition. The experiments determined that the local key point extraction for the original image o can persist information of the extrema of the image such that the large dimensionality becomes reduced as a result, The experimental results also show that the method has greater

flexibility sensitivity to illumination, rotation scale transformation, and affine transformation. [64]. The reason for the initial contribution to pollen type detection is to cater to those individuals who suffer from pollen hypersensitivity [65]. To be able to identify allergenic pollen archetypes in the air may help to alleviate potential discomfort or threats that pollen hypersensitivity can cause. The real-time monitoring of the pollen can open doors regarding the pollen type, concentration, as well as propagation path that could help forecast the allergenic pollen trajectory or concentration [64].

2.2.3 R-CNN, FAST R-CNN and Faster R-CNN

After concluding the brief discussion on non-neural approaches for object detection, we can now safely explore neural network approaches which are newer approaches than the previously discussed. the approach that will particularly be covered in this section will be region-based convolutional neural networks also known as R-CNN. We will also cover fast R-CNN, faster R-CNN and cascade R-CNN, these offspring of the original region-based convolutional neural network method are one and the same however they have improvements then that of its original version which is a topic to subsequently talk about below.

The original purpose of R-CNN was to produce a bounding box that encloses an object of a pertaining class for example a person can be considered a pedestrian or a car can be considered a vehicle. Since its inception in 2013, the authors describe R-CNN as a method that applies a technique namely selective searching which extracts regions of interest (ROI) [2]. Selective search specifically is a data-driven approach based on the highest-grade assumptions of segmentation and exhaustive search (time sensitive) [66]. Key inspirations also include - Exploitation of bottom-up segmentation for object location generation derived from the structure of the input image and capturing all object locations like that of exhaustive search [66]. The algorithm follows the following

steps...

1. Generate initial sub-segmentations of the original input image via the approach in [67]

2. Combining similar regions into larger regions. Selective search utilises a greedy algorithm for these combinations and is designed as follows. The criteria that must be considered for this process are similarities in colour, texture, region size, and the region fill.

   (a) Colour Similarity: Channels of the colours of the individual regions are contorted to a histogram. The paper [66], uses 25 bins for each, R, G, B channels combined into a single vector of total 75 bins. The equation below is employed to determine the similarity by histogram intersection:

   $$s_{colour}(r_i, r_j) = \sum_{k=1}^{n} = min(c_i^k, c_j^k). \tag{2.3}$$

   Refer to [66] for further details.

   (b) Texture Similarity: Texture is represented employing fast SIFT-like readings as it works exceptionally for material recognition [68]. Gaussian derivatives of 8 positions are taken with σ = 1 for each colour channel. For each of the colour channels, a bin size of 10 is used for histogram derivation. Therefore, each region r_i has a texture histogram of $T_i = \{t_i^1, \cdots, t_i^n\}$. Once again histogram intersection is employed:

   $$s_{texture}(r_i, r_j) = \sum_{k=1}^{n} = min(t_i^k, t_j^k). \tag{2.4}$$

   Refer to [66] for further details.

   (c) - Size Similarity: This allows for regions that are negligibly small to merge

earlier. The original paper explains that this is important as to not create undesirable situations where one region is responsible ensnaring all other regions one by one which would cause the scaling from one location [66]. As shown in the original paper [66]:

$$s_{size}(r_i, r_j) = 1 - \frac{size(r_i) + size(r_j)}{size(im)}. \tag{2.5}$$

Refer to [66] for further details.

(d) - Fill Similarity: This is the process of merging smaller regions together. In equation terms expressed in the original paper, it can be assumed that if r_i and r_j contain remnants of each other they would be combined. Opposingly if they both do not touch each other or contain lingering remnants of each other they would create strange merges that should not be aligned or would be an unlikely condition for merging processes.

$$s_{Fill}(r_i, r_j) = 1 - \frac{size(BB_{ij}) - size(r_i) - size(r_i)}{size(im)}. \tag{2.6}$$

Refer to [66] for further details.

(e) The paper then simplifies the mentioned before considerations into the following:

$$s(r_i, r_j) = a_1 {}^S colour(r_i, r_j) + a_2 {}^S texture(r_i, r_j) + a_3 {}^S size(r_i, r_j) + a_4 {}^S fill(r_i, r_j)$$

3. Finally, the candidate object locations are generated from the segmented region proposals.

The paper identified the strengths of selective search as being highly effective at producing a set of high-quality object hypotheses using restrained count of boxes whereas the quality was consistent across the object classes [66]. This is highly competent at least for the time of the publication of the paper. R-CNN and Fast R-CNN use selective search to retrieve important regions within an image, however, it would be important to mention that due to window processing times ranging from 1.8s to 3.7s, it would be inadequate in a real-time circumstance. After the regions have been warped, the inputs are then plugged into the remaining parts of the architecture.

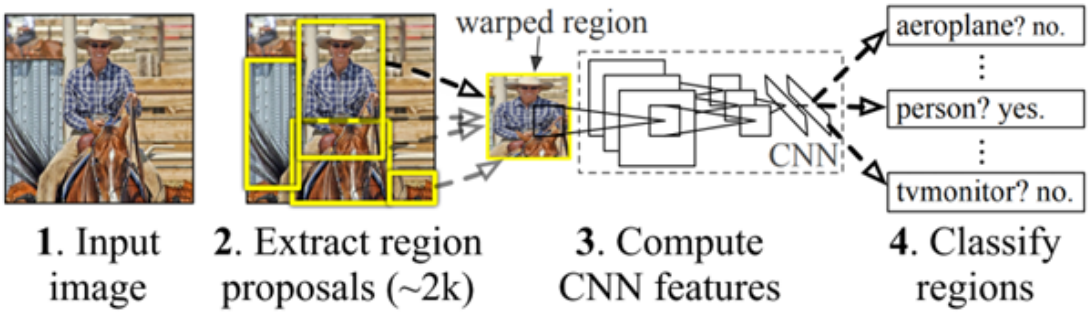


Figure 2.1: Is the original figure from [2] which covers the basic parts of the object detection process. Refer to [2] for further details regarding Regions with CNN features.

Each proposal is passed on to the convolutional network which is where the R-CNN object detection can get highly vary in terms of results depending on the network chosen. The paper proved this by employing two different networks namely, [67], [69]. Comparatively, one outperformed the other with results such as mAP scores of 58.5% to 66.0% respectively of networks mentioned [2]. Although the paper notes that the drawback with the better performing network by Simonyana and Zisserman [69] hinders the length of time taken for the forward pass time being seven times longer [2].

The Support Vector Machines are then responsible for the retrieval of the feature vector generated by the CNN to output confidence scores. Bounding box regression is applied based on the error analysis of detection. The original paper [2] applies a method inspired by DPM (Deformable Parts Model) [70] bounding-box regression. The

purpose of this is to minimise the number of miss-localized detections which in turn boosts mAP scores between 3 and 4 points [2].

R-CNN Off-Spring Architectures

The offspring from R-CNN is the innovations of Fast R-CNN [3] and Faster R-CNN [4]. These both are the main offspring of R-CNN and have proven to ensure great applications in object detection. What were these improvements? Before answering this, the drawbacks of R-CNN need to be identified. Regular R-CNN is slow, this is because the training and testing require 2000 regions to be calculated for completion. Each of the regions enters the CNN where it takes a large amount of time to extract features.

The Fast R-CNN attempts to increase proficiency by taking great inspiration from SPPNet – Spatial Pyramid Pooling Network – which converts the entire input on the feature map a single time. Fast R-CNN in a similar fashion using a single feature map for all the regions. This entails that one layer namely the pooling layer is used to convert the features with the maximum pooling [3].

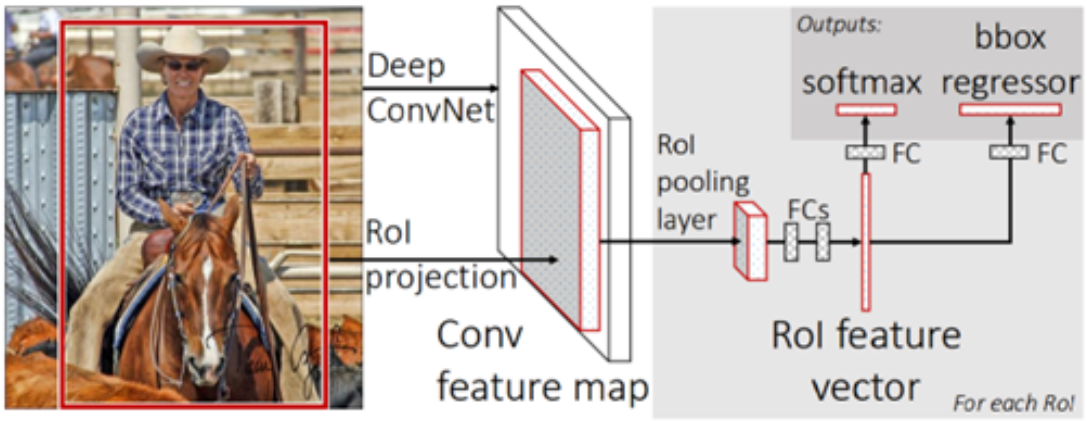


Figure 2.2: Represents the architecture from the original Fast R-CNN paper [3].

The next iterations of the R-CNN bloodline would be named Faster R-CNN. Up until this point, we are aware that the selective search method was used for both R-CNN

and Fast R-CNN region proposals. Faster R-CNN provides replaces the selective search method for region proposals with another named a region proposal network [4]. The reason for its replacement with a new method is due to the time taken for each image, being 2 seconds per image at a CPU implementation [4]. This network is an extra CNN that takes the features map as input to then output adequate region proposals for further processing in the ROI pooling layer [4].

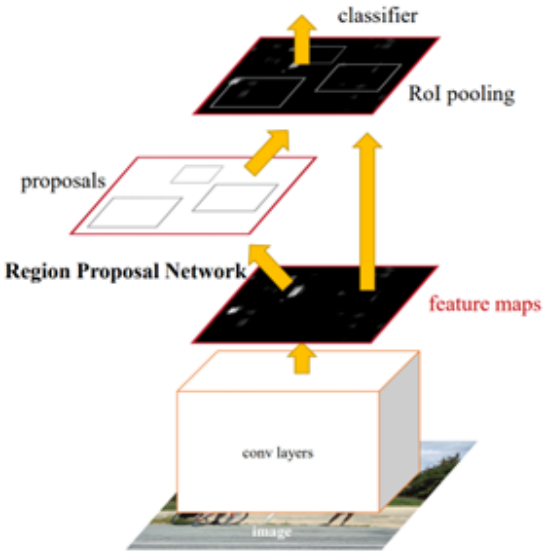


Figure 2.3: A fully connected network representing the RPN module as the staple addition to the existing former architecture as further detailed in [4].

R-CNN.architecture Applications

The applications of R-CNN, Fast R-CNN, and Faster R-CNN are vast. The below sections will briefly cover some of the developments in the object detection space which implements the method of discussion.

The paper was done by Moue et al., (2019) [5] is a perfect example of R-CNN being employed in marine target detection specifically. The paper states the importance of marine target detection in maritime transportation, marine environment detection, and national defense [5]. Usually, it is commonly known that radar is the medium for

maritime navigation or detection using electromagnetic waves which do not hinder the restrains of weather rain, or fog, nor is it time restrictive. This then calls into question what and why would the paper [5] needs to propose an R-CNN-based method for target detection. Before the proposed method radar methods would employ traditional methods that would include statistical processing and nonlinear processing [71]. On top of the obsolete pathway for the task itself, we must consider the drawbacks such as that radar for this task succumbs to difficulties such as low echo signal-to-noise ratio, complicated non-uniform backgrounds [71], and with complex marine environments. The specific R-CNN sub-type used was the Faster R-CNN with further improvements made such as...

1. Original classification loss replaced with focal loss [72]. This is done to solve the imbalance of the samples as well as improve the training results.

2. Replace the original non-maximum suppression (NMS) to a soft-NMS [73]. This was done to improve missing detection results

3. Instead of the original ROI Pooling, the paper opted for a Precise ROI pooling method [74] to further the accuracy provided which in turn reduces the precision loss of scale uniform function.

4. Lastly, the rectified linear activation function is replaced with the exponential linear unit [75] to rectify the vanishing of gradients and to increase the speed of the convergence tasks.

This paper helps to appreciate the flexibility of the R-CNN architecture which only gives this method further merit. Other papers within the maritime industry that utilise the R-CNN family of methods include [76], [77] and [78].

Due to the nature of this research project being agricultural-centric, it would be a disservice to avoid developments in object detection for the industry. A paper [79] uses

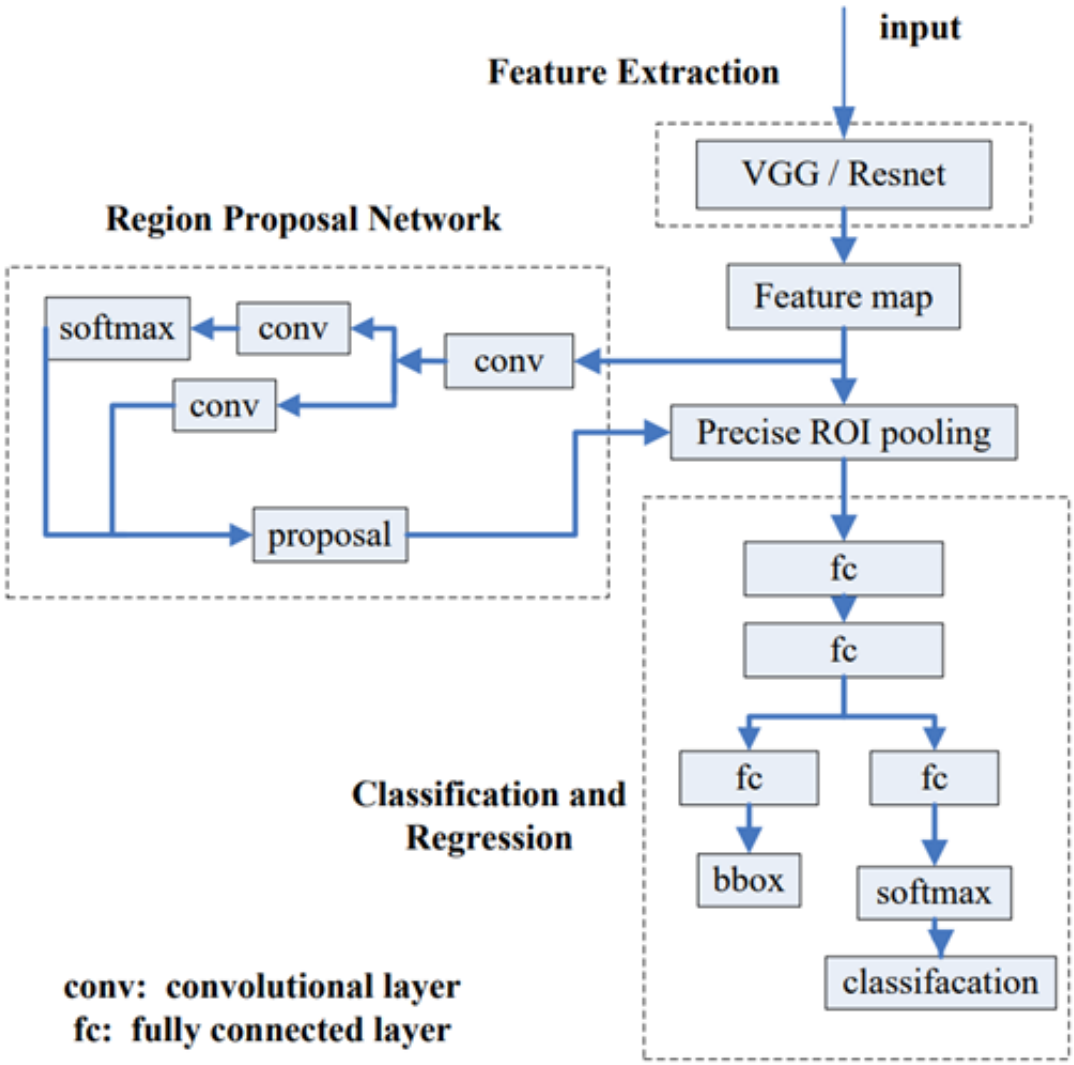


Figure 2.4: the improved faster R-CNN method for the use case in [5].

another alternative R-CNN method namely, Mask R-CNN. Mask R-CNN, to briefly cover, has comparable properties in terms of the process to the previous version however it focuses on creating a mesh-like mask on the objects being detected. This was used extensively in the paper [79] as they provide a solution for smart irrigation for crop management. The paper provides a deep learning solution that utilises an unmanned aerial vehicle and transfer learning from Mask R-CNN framework to locate water. The main purpose of this application is to identify malfunctioning sprinklers that could cause under-watering or over-watering [79]. During the conclusion, they express that

the application still requires further work for practical on-site work. This is due to the sensitive data that neural networks are hindered by such as background, light, image quality and camera angle [80].

The paper conducted by Deserno and Briassouli, [81] uses Faster R-CNN to identify varying insects within sticky traps. These traps are yellow sticky traps that are a crucial tool in agriculture to monitor the types of insect pests so that they may be classified and counted [81]. The project was devised to nullify the need for manual counting/identifications of said insects and what better way than the automated solution?

Another interesting Application of R-CNN is creating an effective method for weed killing on crop fields as shown in [82]. Agriculture and automation have become a formidable combination for researchers to flex their expertise especially for the replacement of mundane tasks shown as in the following [83], [84]. The paper proposes a quadratic traversal algorithm to determine the shortest path for weed killing for robotic hosts [82]. The proposed method uses the faster R-CNN method specifically in conjunction with the VGG-16 feature extraction network- a series of CNN layers followed by few fully connected layers- which is one of the most used algorithms for image classification as it is permissible with transferable learning [82]. The paper also employed and modified an existing algorithm, namely the OTSU algorithm which is commonly used for image processing in agriculture [85]. The algorithm has been improved specifically by compression of the range of the search gray-scale difference [82]. The proposed method was successful in the goal of reducing the distance required for weed killing for robotic hosts. Due to the externality of the proposed project, power consumption optimization and camera issues regarding environmental conditions were the future focuses [82]. A common trend in the paper discussed shows a lack of heavy focus on faster R-CNN development more so than it was the optimizations of the surrounding tasks which fulfill automation goals. The intertwining of various systems together in conjunction with architecture/models like R-CNN is how this field can move

further in the goals of full automation.

Automation in this industry can be derived from the combination of object detection, deep learning, and data analysis. The paper [86] is of substantial relevance due to the nature of the proposed method for apple detection. The [86] employs R-CNN to automatically detect passion fruit – fruit with high medicinal applications, flavour, and nutritional benefactors – to then produce yield maps. The paper uses modified multiple scales faster R-CNN or MS-FRCNN architecture based on Residual Network – Aka ResNet, which is another specific type of neural network – that was deemed successful in the detection tasks [86]. The paper also mentions the difficulties of detecting smaller objects in a yield estimation system due to the lower resolution of the input [86]. Comparatively to the Tu et al., (2018) paper [87], the MS-FRCNN had the 3rd, 4th, and 5th convolutional layers of the network to be combined. This combination was able to produce promising results as it outclassed the standard faster R-CNN method [86]. For future endeavours [86], the authors express interest in YOLO-V3 compared to the proposed solution. This is because YOLO-V3 would allow for adversarial training qualities to be incorporated which would help to tie the training to itself to then produce a self-contained improvement mechanism for the model.

2.2.4 Single Shot MultiBox Detection (SSD)

Single Shot MultiBox Detection or SSD [7], is an alternative algorithm/architecture commonly used in object detection scenarios or computer vision applications. Object detection using SSD is composed of 2 main sections, one of which is the extraction of feature maps and the second is the application of convolutional filters [7].

Unlike R-CNN the SSD model does not use a region proposal network for the determination of ROI. SSD simultaneously resolves the class scores and locations via convolutional filters. SSD is inspired by a feed-forward CNN that is responsible

to produce a static sized group of bounding boxes and provides the scores for the availability of an instance within said boxes [7]. After this, the non-max suppression is used to find the final detections.

The early layers are considered a base network that SSD implements for truncation of high-quality image classification. The model then follows the truncated base network with the convolutional feature layers which are responsible for decreasing size sequentially, to then, in turn, predict detections in multiple scales [7]. This also means that SSD has considerable issues with detecting small objects. To offset such an issue the input image should be of high resolution. Compared to other methods such as the You Only Look Once detection model [6], SSD implements varying convolutional models for predictors [7]. Individual feature layers added to use a set of convolutional filters to produce detection predictions which are also in group sets [7]. As shown below in the architecture design of the model...

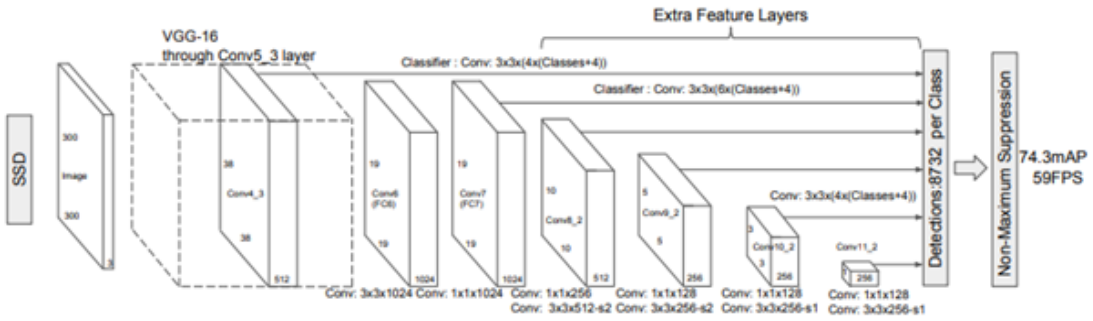


Figure 2.5: Represents the SSD model utilising the base network as VGG-16 network. Comparatively to the YOLO model [6] tests done on VOC2007 show that it outperforms in speed and accuracy [7].

It is significant to note that the bounding box revolves around the centre point, namely the default. As the original paper mentions that the bounding box offset results are measured relative to the default box position to each of the feature map locations [7]. Due to the convolutional nature, the feature map is separated in a tile-like manner so that the locations are relative to a fixed cell [7]. In each of the given cells, there is a

prediction made of the offsets relative to that of the default box shape inside the given cell as well as a score that indicates the matching of an instance in each of the boxes [7]. In Faster R-CNN [4] the default box is named the anchor box and does not apply them in varying resolutions of the abundance of feature maps. The varying in the default box shapes allows SSD to be efficient in discretisation (the process of transferring variables, functions, equations, and models into separate pieces) of space in the possible result box shapes [7].

SSD Applications

In this brief section, the applications of SSD in modern settings and industries will be discussed. One thing to note is that application sections much like before will try to illustrate the wide range of the methods used with justification as to why alternatives are disregarded. The goal would be to be familiar with the varying scenarios in which the methods are mostly applied.

Like previous analysis regarding R-CNN, SSD also is extensively applied in maritime applications as shown in the example [88]. The premise of the paper is surrounding the importance of synthetic aperture radar (SAR) technology in the space of marine transport and fishing law enforcement [88]. A method is proposed using SSD to address ship detection in noise-heavy SAR input images. This paper is chosen as it is significant in highlighting the benefits of the over other methods, namely that SSD is faster than faster R-CNN [88]. Results showed that the SSD-512 is 50% faster than Faster R-CNN [88]. Although the speed is revered here, the paper also mentions false positives that occur when ships are close to locations like the harbour or islands. This is due to the generation of the bounding boxes of the features starting from the 15th layer convolution which does not consider the silhouette of the ship itself [88]. This unfortunately results in the port cranes or buildings being considered as ships when detection occurs [88]. The paper proposes future work to absolve these issues by pinpointing the low-level

features such as edge separation discrimination via edge detectors etc [88]. The results of the paper were shown to be fruitful and future endeavours include the reduction of false positives.

In a similar vein, the paper [8] provides more insight into SAR detection in marine applications. This paper focuses on the difficulties in the detection of ships being small in the SAR images, which in turn causes poor detection [8]. This much like the prior paper is an upgrade to the existing SSD model. The paper proposes two upgrades, one of which is a default box optimization via K-means clustering and the latter is feature fusion based on deconvolution which improves low-level feature map representations [8]. The paper mentions the drawback of SSD, that being the issue of poor accuracy with small targets [8]. Below shows the mods made to the regular SSD model from the original paper.

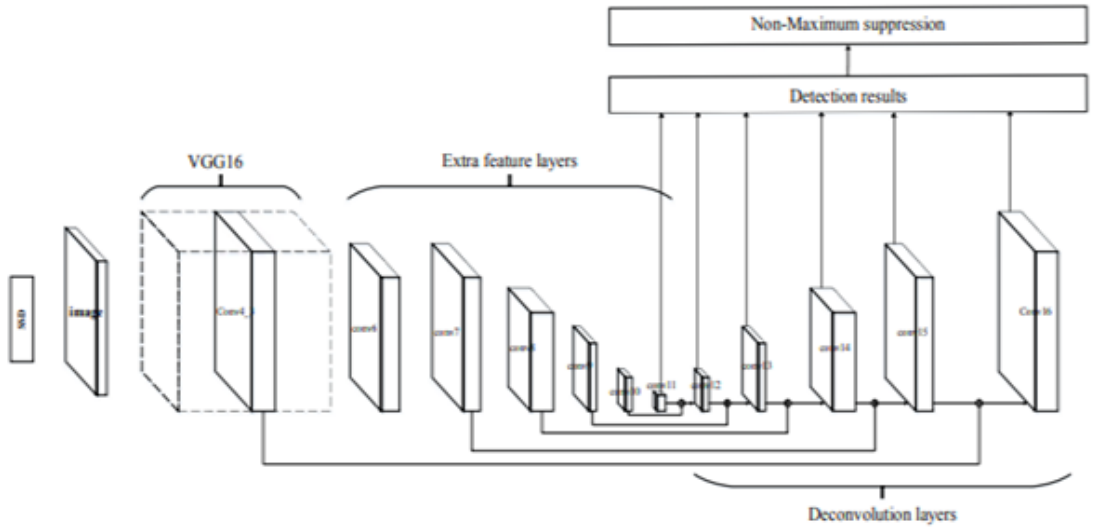


Figure 2.6: This is the figure provided by the original paper that illustrated the 5 extra deconvolusion layers [8].

Deconvolutional layers in this specific scenario function like upsampling without the fixed parameters. Because of the nature of the deconvolution network being able to learn the parameters during training, the up sampling becomes more manageable deconvolutions [8]. The whole reason for deconvolution-based feature fusion is to

mitigate the problems with small targets containing weak representation leading to underfitting problems i.e., low recognition. On the other hand, if the convolutional layers were to increase the representation would become significantly better for the low-level feature maps yet it would cause overfitting [8]. The feature fusion provided by the paper is inspired by the top-down network structure shown in [89] and is based on the feature fusion method done using deconvolution by [90]. This ultimately is responsible for combining the high-level semantic features with low-level feature maps to increase the quality of small target features [8]. The paper discussed showed effective results in improving the SSD model for the use case and was successful in recognising small targets in SAR images[8].

As mentioned earlier due to the manner of the origin project, agricultural development using SSD is imperative to set definitions for progress in the industry so as to attain guidelines. This guideline is that of automata, as the farm industry steadily leaps forward with the concept, the trends start to show some more niche applications such as the paper done by Wang et al., (2022) [9]. The paper exclaims the economic importance of Lingwu long jujubes, a forest fruit situated in Ningxia Hui Region further citing the following [91], [92] The fruit is, as per the name, oval-shaped, thick, and long yielding a high amount of vitamin C [93]. Due to SSD's well-recognised possibility for low computation, it becomes a valid method for achieving such detection tasks. One issue is the requirements for pre-trained or preloaded weights. The pre-loaded weights are undesirable due to the following reasons:

1. The network has poor flexibility, The pre-trained model is large and the parameters are extensive. This limits the application scenarios [9].

2. The optimisation availability is sparse which is developed due to the different allocations of the loss functions and classification of the detections being varied [9].

3. Despite the fine-tuning benefits on the reduction of the of variability of the distribution of target categories, it is too significant for the effect to be noticeable [9].

The authors have cited previous endeavours via [94], [95], [96] for the identification of jujubes but have noted that these methods are not accurate enough for detection or classification while being slow. Due to the mentioned gripes, Wang et al., (2022) [9], have developed a lightweight, void of pre-loaded weights, SSD-based, containing fewer parameters and with better detection accuracy for Lingwu long jujubes [9]. Below is the structure of the proposed modified SSD model.

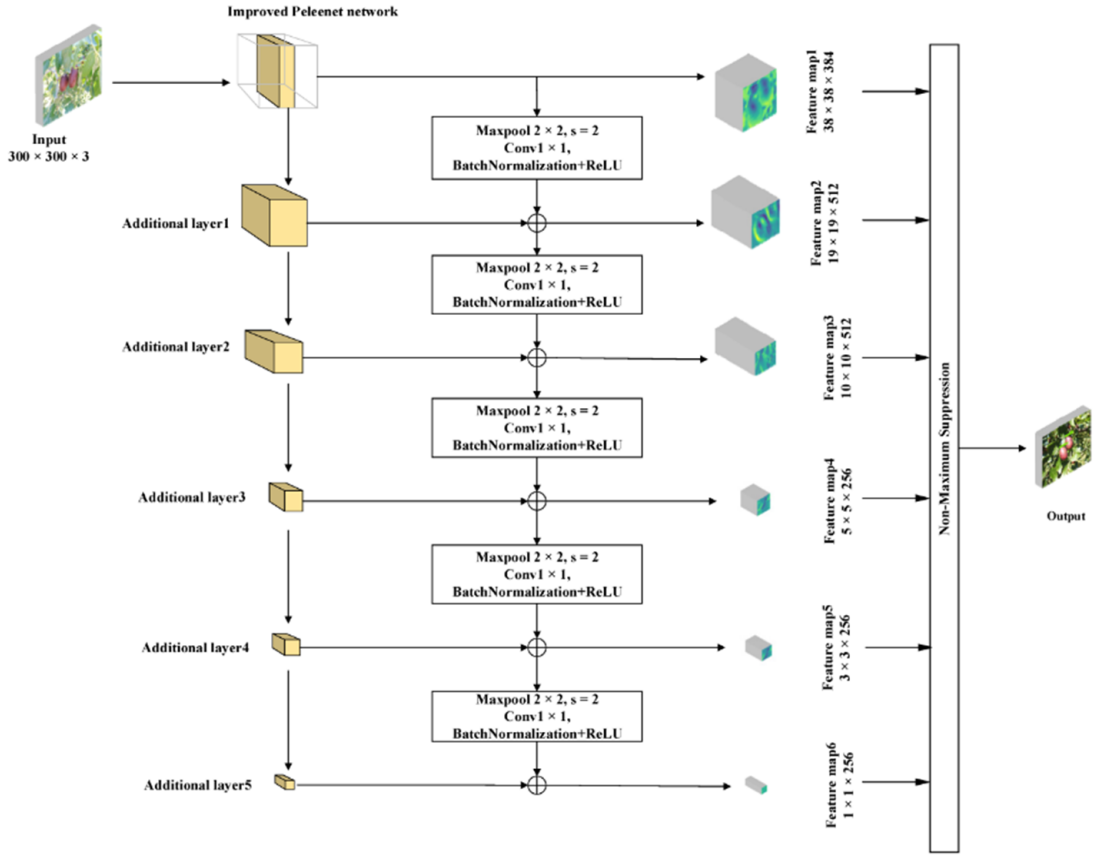


Figure 2.7: Improved modified SSD target detection model [9].

The original VGG16 network that was commonly used in the original SSD structure has been replaced with a modified PeleeNet module as the base trunk of the network.

The PeleeNet CNN is a variant of the DenseNet CNN that is optimized to meet low computational requirements as well as meet memory budgets [97]. Inceptionv2 module which is an alternative network replaces the original first few CNN layers to further strengthen the detection capabilities [9]. The passing of image feature messages between the plethora of levels is done via appending the additional levels [9]. These were modifications to better the overall performance for real-time detection for the detection use case. The results have indicated to be successful in both accuracy and speed. Compared to the conventional SSD algorithm, the improved algorithm yielded in higher mAP score of 97.32% with an AR reaching 78.23% with compressed parameters reaching 30.37% [9]. Future works include the poor accuracy results for heavily obscured Lingwu long jujubes and the practicality of a picking robot armed with the detection method [9].

The next paper in the agricultural scene to review is [10]. The paper is responsible for proposing an SSD solution for the identification of single tree detection. The paper highlights the importance of trees in the field of forestry ecology, as well as environmental management [98], [99]. The benefactor of using a CNN-based method is again tied to the time mentioned automata. Being able to aptly detect features automatically and apply a sense of the low-level features such as contours or colours is highly beneficial [100]. As mentioned, the SSD model is advantageous due to its speed and high accuracy of detection [7]. This paper takes advantage, once again, of SSD and applies it with sufficient optimisation for the specific use case [10]. Specifically, the modified SSD model was employed in quantity identification and location management of urban plantations [10]. The simplified SSD framework is as follows:

The SSD model has proved to be superior amongst previous endeavours for single tree detection such as the watershed method and the CV method [10]. In the experimental results provided by [10], the SSD model produces an average accuracy of 96.32% as it can segment the canopy features of a single tree from high-resolution

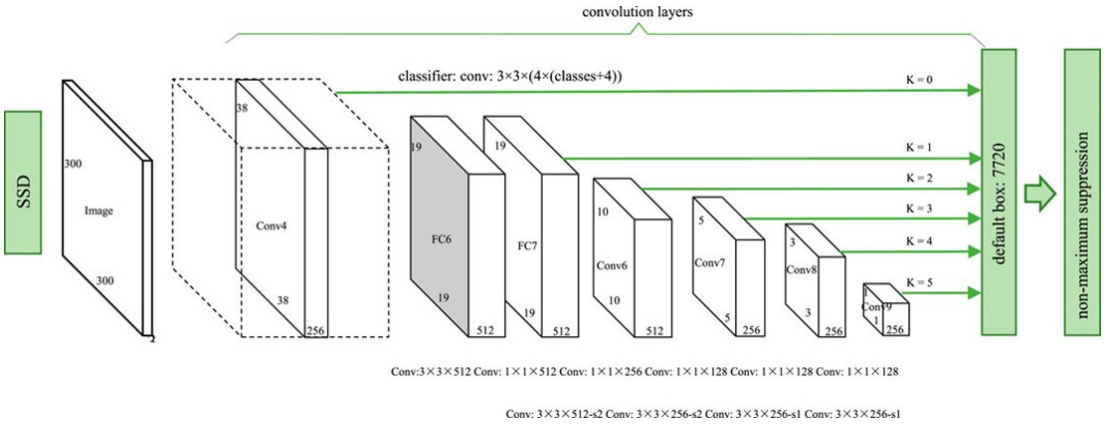


Figure 2.8: Provided by [10] is the simplified model which utilises the VGG16 partial void of the convolutional layer.

images. Also, recall rates of the SSD model show a 97.94% average recall rate which is the highest amongst the experimental results [10]. As per the SSD model developed, it is capable of segmenting single tree crowns, and it can recognize them in complex backgrounds [10]. It is capable of bolstering anti-inferencing capabilities and mostly negates interference of branches of a single tree detection scenario [10]. Finally, the application potential is promising as it has proven to perform exceptionally in all regards [10]. SSD much like R-CNN has paved the way for new iterations of the same network allowing for flexibility in many industries, not just those discussed in this section.

2.2.5 Object Detection Significance

The significance of object detection has opened large new pathways in human research. These pathways improve, as should most research, the human experience via efficiency. Efficiency is heavily knotted with automata as discussed earlier during this review section. Automata is one of the keystones of improving various industries whether it be agricultural or maritime and one method of automata in the physical realm is that of object detection. Object detection is unique that it opens various doors to solutions that without the advent of these various enabling algorithms would not be able to

comprehend or attempt to apply to issues. Overall, the section above has highlighted the benefits to some of the industry as well as a chronological timeline to show how object detection has evolved from that of equation-centric solutions such as the Viola-Jones method to the big data-centric solutions like the R-CNN or Faster R-CNN algorithms. Although this paper refrains from complex algorithm modification or complex equation crunching, the rationale for the overview of these algorithms is to show the greater door that they have provided for the solutions like this thesis aims to achieve to be realised.

2.3 Agricultural Developments

As well established, many industries are slowly and steadily adopting automata as part of the workflow to increase efficiency in general. Naturally, agricultural developments have also led in the direction of automata such as those discussed subsequently.

2.3.1 Smart Agriculture

Smart agriculture is a newer term related to the integration of automation or AI and the internet of things (IoT) technologies in farm management to further expedite the mundane tasks that would otherwise be either time-consuming or in niche cases create solutions [101] much like this project aims to achieve. Smart farming is nipping the long-term issue of food shortages as the population of humans is projected to increase to 9.73 billion [102] (pg, 26). The paper [103], has surveyed smart farming technologies that have been implemented from 2019 to 2021 which adhere to data gathering, storage data analysis, transmission, and adequate niche solutions. The paper cites various industry-related difficulties that result in a reduction in production efficiency such as soil salinity in arid conditions [104], [105]. Further issues may array due to climate conditions that may cause the land to become sensitive to desertification [106], [107]. The paper [103] has compiled a series of 58 documents relating to various countries worldwide. The list

below indicates the topics covered in [103] regarding smart agriculture:

1. IoT in smart agriculture: IoT is a robust new outlook to the way humans create efficient solutions that are increasingly becoming more popular and will become more prevalent in the coming years. With the advent of this technological advances such as these, agriculture can utilise wireless sensor networks to help keep all aspects connected, therefore creating better solutions for irrigation and fertilization optimization [108]. One thing to note is that the IoT is dependent on the speed of the internet and therefore the faster the better. One paper conducted by [109] was able to use IoT in real time to track, diagnose and notify via web network farmers of leaf diseases that would cause obstructions in growth.

2. 5G networks on smart farming: As mentioned earlier IoT, in general, is dependent on the speeds of the internet itself, and thankfully in recent times, the fifth-generation aka 5g communication network is possible for fast transfers of data. The 5g network provides beneficial merits such as high data transfer adhering to low latency, high connectivity density compared to previous networks, spectral efficiency improvement, increased communication performance, large coverage, and high efficiency of network energy [103].

3. Smart Sensing in agriculture: Sensors are a topic that has been covered previously in this literature review and is a large part of this project. The usage of sensors in agriculture can be beneficial for monitoring livestock, soil moisture, water levels, nutrient levels, etc. commonly used sensors in farms nowadays are mostly related to moisture management in soil [110]. Statuses that are monitored further include the temperature of the soil, air temperature, pH values, and humidity [110]. An emerging sensor in smart agriculture would be LiDAR.

4. Applications of said IoT in smart farming: As established briefly before IoT

has been applied to the monitoring of various aspects of the agricultural space. IoT systems can extend the versatility of the technology. An example of this is the unmanned aerial vehicle or UAV. UAVs are commonly used in other fields for terrain elevation analysis and can harbour various sensors to enhance their flexibility in specific locale scenarios [111]. In agriculture UAVs (Unmanned Aerial Vehicles) can be equipped with several sensors such as cameras, thermal, multi-spectral, and more recently LiDAR [103]. The many attachments on a drone allow for arrestation, spraying, terrain mapping, field prediction, plant counting, plant measuring, exploration, and nitrogen measuring [112]. Multi-Spectral capabilities on drones allow for vegetation analysis [113].

5. Limitations of Drones in agriculture solutions: Despite the benefits of the drone, there are drawbacks like the following:

    - Depending on the drone model and specifications required it can be expensive
    - Laws need to be abided as well as drone permits needed to be issued for usage and this may be difficult to obtain depending on the country [103].
    - Climate impacts the operation times of the drones as well as wind speeds ruining pre-determined flight paths [103].
    - The drone has an active status of an hour or less. Due to this, flight paths need to be determined beforehand [103].

6. Smart farming approaches in developing countries: In many developing countries such as India are willing to adopt such technologies that increase efficiency but due to the obstacles involved such as the availability of infrastructure being owned by the state rather than the individuals running the farming facilities [114] can pose difficulties adopting smart agriculture. Excluding country-specific

obstacles, the main implementation obstacles include, sensor availability, suitable 5G network establishment, device/equipment availability and trained experts for said navigation of such technologies [103]. African regions face alternative issues regarding the space of operation itself such as water scarcity, environment changes and climate fluctuations [115].

7. Smart decision support systems (SDSS): The last topic covered in [103] was the smart decision support systems or SDSS. The SDSS is designed to support those farmers who required assistance in making rational and informed decisions regarding smart farming utilization [99]. The goals are to provide diverse options for irrigation management, fertilization, and numerous other endeavours for given situations. SDSS can be considered a gateway for user-friendly smart agriculture as it provides various tools for those uncertain to be eased into the process.

Finally, the paper [103] concludes with the importance of smart agriculture and the significance of IoTs (Internet of Things) presence as the defining pillar for many of these systems. Also, when developing countries slowly implement these systems, they will provide a propelling effect as farm sustainability can be achieved. The paper also suggests that those governments in said developing countries should highly consider smart agriculture to increase the production of food as well as efficient use of water or other resources [103].

2.4 Summary

To briefly summarise the purpose of the aforementioned chapter, it provided various faucets of information that revolved around computer vision, smart agriculture and machine learning algorithms surrounding the relevant field. The reasons for this is to inform and provide a clear understanding of the various techniques that alternative

research has implemented in hopes that this can be used or referenced as part of this thesis.

Chapter 3

Methodology

3.1 Research Methodology

In the following section, the research questions, aims, goals as well as design choices are discussed. Furthermore, the nature of this project leads to outline of the developmental approaches for the creation of the design solution which has been included as part of the following section.

3.1.1 Research Questions, Aims and Goals

Reminder, this project aims to develop a prototype or proof-of-concept of an IOS-based application that utilises the onboard LiDAR sensor to derive the size of a given fruit. The project objectives are broken down as follows:

- Use python programming language to develop prototype back-end for handling LiDAR and image data.
- Use Swift programming language to develop prototype application to harbour suffice user interface for data collection.
- Successfully extract valuable LiDAR and image data for back-end processing.

- Use the company provided machine learning model checkpoints to detect fruit locations in captured image data.
- Retrieve the distance from the detected fruit for each fruit in the image.
- Use all data gained to derive a size estimation of all fruits detected.
- Derive viability of method moving forward which includes time efficiency and accuracy compared to standard methods available nowadays.

While the above are considered objectives more related to that of the developmental process (see section 3.2) , Where as the main question to find an answer to is:

1. Does the proof-of-concept application create a viable beneficial alternative to standard methods of fruit sizing?

3.1.2 Research Design and Methods

In this section, several different subcategories are highlighting not only the data collection process but the approaches taken for the design solution development. Understanding the proof-of-concept development is relevant for the understanding of the data types for data analysis.

Research Strategy

Much like the papers that are data collection-focused such as [116], [117], [118], this paper will employ standard methods for data collection. The unique component of this methodology is the inclusion of developmental strategies. The detailed recount of the developmental process creates space for later critique in this thesis that is not necessarily data-driven but more speculative. This can be of benefit to the overall project as it creates suggestions for future development upon the proof-of-concept application.

Exploiting this approach allows for not only solid, tangible data-driven determinations but also for the speculative solutions that could be made for shortcomings of said application. it is common for the underpinning philosophy to be surrounded by the positivist approach. Although it is not common in recent papers to explore research philosophy, it is valuable to identify why it is chosen.

Data Gathering Strategy

To fulfill the aims of the project, the data must be adjacent to the intent of determining whether the application is a viable approach to measuring fruit On-Tree fruit sizes. Viability includes whether it compares to current standards. For clarity, the following tools are employed during the data collection period:

- Caliper (digital): As the project is focused around finding the validity of the application to measure fruit on tree, it is essential to determine and compare with that of the ground truth measurements of the fruit. Although this is prone to human error, this is- as of this stage- the most accessible method to gain the ground truth measurement.
- IOS device (iPad): The device that currently holds consistent LiDAR capabilities and therefore the mentioned application developed was also made for this platform.
- Stopwatch: This tool is an extension tool to help the validity analysis as it will provide valuable information regarding the time it takes to capture and save information.
- Measuring tape: Measuring tape was used to measure the negligible distance away from the object. This is prone to human error.
- Fruit: Less of a tool but more of the main assessed apparatus.

- Makeshift Tripod: This is used mainly during the calibration period to determine the best parameters (3.2.2).

The experiments are exhaustively straightforward as the ultimate derivations required are data that can identify the validity i.e., the accuracy of the application that was developed and comparisons of current standard approaches (manual for the most part). Employing the simple experiments keeps the data gathering to a less complicated process and provides a simple barrier to entry for alternative future projects that intend to build upon this work.

Before any experiments are conducted the fruit is measured and ground truth values are determined. This involves the following ground truth criteria:

- Fruit diameter is measured via digital calipers.
- Mark distances away from fruit position via measuring tape.

The mentioned provisions employ ground truth comparisons and in turn provide a manageable method to determine the error range of the application. After said provisions are complete, the experiment follows a simple loop like so:

1. Capture data at ground truth x distance away from the fruit position of ground truth diameter of fixed y.
2. Use in application share functionality to organise data into dedicated folder on google drive. This specific process should be timed with the stopwatch.
3. Loop step 1-2 until all data is gathered.
4. Transfer folder data to relevant locations for the back-end machine learning fruit size determination script.
5. Record final diameter results from back-end output to spreadsheet application such as Microsoft Excel.

6. Compare ground truth diameter value y and application derived diameter z value.

The important final information that is of high significance is the diameter of the fruit gained via the application. The simple percentage error calculation is applied as shown below:

$$pERROR = (\frac{(z-y)}{y}) \times 100 \tag{3.1}$$

Where $pError$ is the percentage error range derived.

The simple loop experiment mentioned earlier will also be applied to a flat surface scenario where the main subject will be placed against a wall for comparison with the On-Tree Scenario and the cherry bunch scenario experiments. The conducted experiment will contain manual intervention to gain the pixel values through the software Adobe Photoshop as the machine learning model will not be able to detect the wall reference. Despite this, it will provide information on the importance of the bounding boxes and the need for precise pixel information. It should help determine any anomalies amongst the fruit-specific data and the calibration importance. The test for this will also encompass a larger length from the object to identify if accuracy falters. This may also help to see the drop off in accuracy. This is key for the discussion section of the paper (see chapter 5.1.5).

As this is an experimental-based research strategy it is important to identify the control that should be adhered to during experimentation despite being obvious. As part of the controls, all tests will keep static the following-equipment, app, fruit specimens (regardless of varying size), the resolution of the image captured, the amount of depth/LiDAR information retrieved, and functions for the derivation of diameter size of the fruit. The attribute of the experiment that will change is the distance between the object and the camera. This will help determine the adequate functioning range of the application. The data gathered is cross-sectional in nature as the readings will all be

collected and processed during the same period.

Following the trend of simple experiments, a test for the current machine learning model version will be done. This experiment is less to do with the ultimate goal and more to do so with how the pixel values produced via the model affect the final sizing estimations. This project uses a machine learning model more akin to that used in [1]. The experiment will consist of counting the number of detections made per capture in a bunched scenario using cherries. The steps to this experiment are as follows:

1. Take the capture as done in the prior experiment loop.
2. Input the image data retrieved from the capture data into the machine learning model.
3. Finally present the image with bounding box counts.

Data Analysis Strategy

The data analysis is boiled down to the comparisons made between the true ground truth values and the estimated values gained via the proof-of-concept application. Alternatively, the count experiments will not be employed for comparison, rather they will serve as a confirmation test. As easily deduced, the success criteria for the experiments done excluding the count experiments is the error range percentage. The closer the percentage is to 0 the better the result as it means it is close to the ground truth. On the other hand, the count experiments should produce matching count values of fruit detection to be successful. For all experiments conducted, a specific type of analysis is adhered to, namely statistical-based descriptive data analysis. Applying this type of analysis should determine the eventual validity of the proof-of-concept application via the accuracy determinations with comparisons to the current standard. In these experiments, data that is duplicated is considered to be unique. Furthermore, outliers are identified, discarded, or kept and discussed as some provide insight into the inherent

limitations of the proof-of-concept application. The main software for data analysis will be Microsoft Excel which will provide tools to create graphical representations of the analysis conducted as well as tables that will harbour valuable data for readers.

3.2 Proof-Of-Concept Application Development

The Following sub sections will discuss the development process of the application solution for the mentioned goals and aims.

3.2.1 Front-End Development Components

As already established the platform of choice to deploy the solution for sizing fruit is the IOS devices harbouring the LiDAR capabilities. Due to the time constraints for the project, the bones of the front end have been derivative from an example project from the apple developer website [119]. The script incorporated is a great base to start, however, it does not meet the requirements alone. Therefore, modifications were made using the AVFoundation framework or library specifically related to capturing with LiDAR [120]. The following will discuss the developmental process of said modifications made to the base front-end swift script as well as the corresponding screenshots.

Renderer Script

The IOS Swift library is used to retrieve the LIDAR information in tandem with the image data. The AVFoundation [11] framework that is used has previously been used in other computer vision-based projects such as [121], [122]. For this use case, it is related to retrieving raw data that is formatted into a text file that is later processed in the back end. AVFoundation is a solution that appeared after a series of failed attempts to retrieve the LiDAR data (see chapter 5). The prior failed attempts provided beneficial accuracy

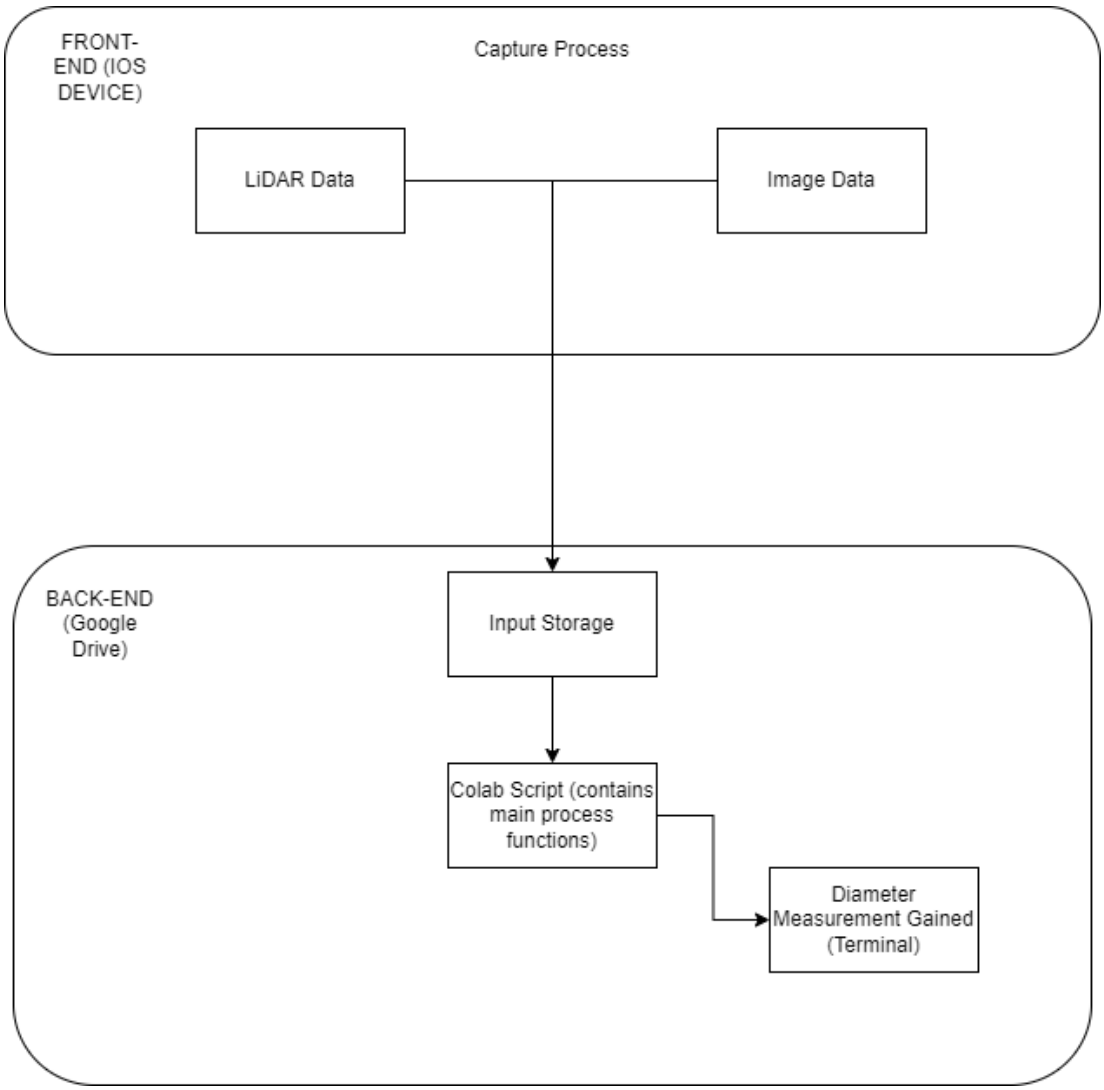
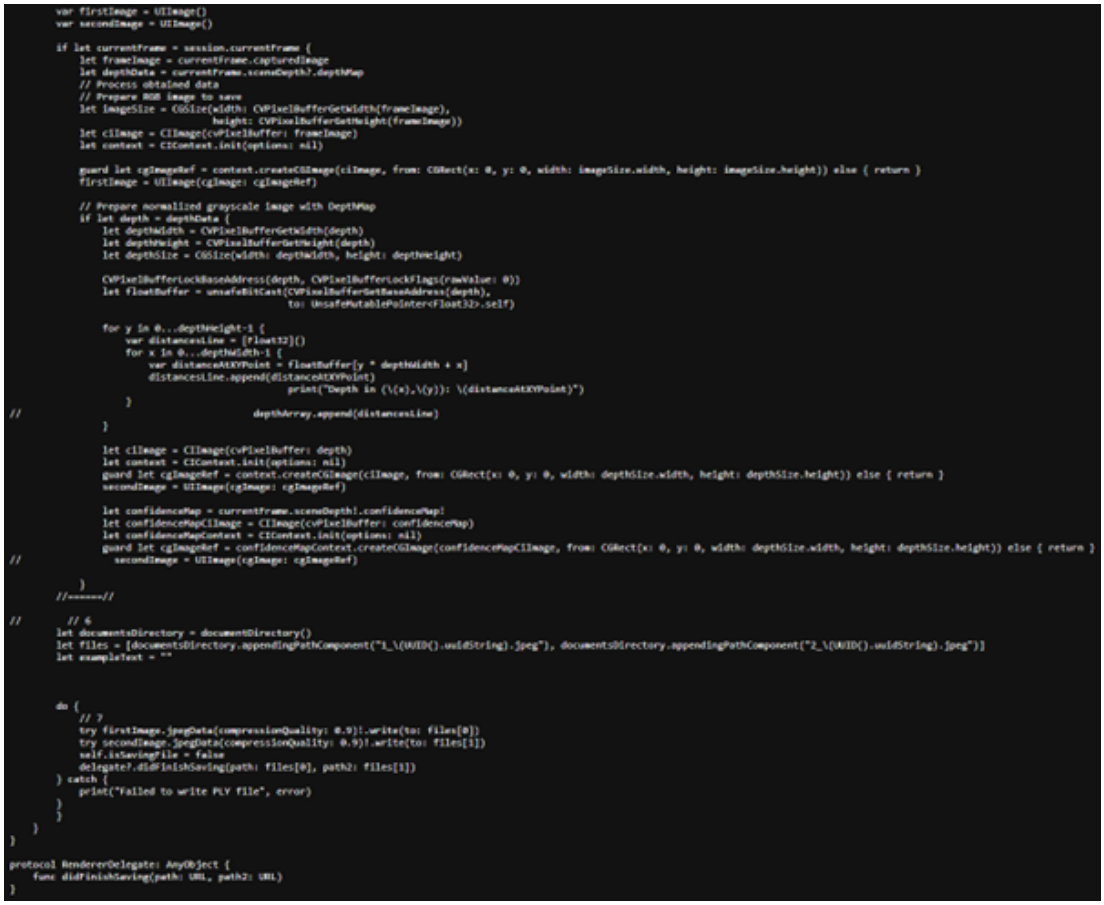


Figure 3.1: This is the general high-level view of the pipeline from data acquisition to diameter measurement gained.

however it was void of providing values for every pixel on the capture which in our case is necessary as the positions for the fruit can be put anywhere on the captured scene. This is accounted for in the AVFoundation approach provides an estimation for those pixels unable to be hit by the laser pulses and provides a highly accurate reading based on surrounding LiDAR readings. This is still however considered to be a drawback at least as it is still an estimation. Despite this, the results (see chapter 4) may show that this drawback is not necessarily anything to worry about.



```
        var firstImage = UIImage()
        var secondImage = UIImage()

        if let currentFrame = session.currentFrame {
            let frameImage = currentFrame.capturedImage
            let depthData = currentFrame.sceneDepth?.depthMap
            // Process obtained data
            // Prepare RGB image to save
            let imageSize = CGSize(width: CVPixelBufferGetWidth(frameImage),
                                   height: CVPixelBufferGetHeight(frameImage))
            let ciImage = CIImage(cvPixelBuffer: frameImage)
            let context = CIContext.init(options: nil)

            guard let cgImageRef = context.createCGImage(ciImage, from: CGRect(x: 0, y: 0, width: imageSize.width, height: imageSize.height)) else { return }
            firstImage = UIImage(cgImage: cgImageRef)

            // Prepare normalized grayscale image with DepthMap
            if let depth = depthData {
                let depthWidth = CVPixelBufferGetWidth(depth)
                let depthHeight = CVPixelBufferGetHeight(depth)
                let depthSize = CGSize(width: depthWidth, height: depthHeight)

                CVPixelBufferLockBaseAddress(depth, CVPixelBufferLockFlags(rawValue: 0))
                let floatBuffer = unsafeBitCast(CVPixelBufferGetBaseAddress(depth),
                                                to: UnsafeMutablePointer<Float32>.self)

                for y in 0...depthHeight-1 {
                    var distancesLine = [Float32]()
                    for x in 0...depthWidth-1 {
                        var distanceAtXYPoint = floatBuffer[y * depthWidth + x]
                        distancesLine.append(distanceAtXYPoint)
                                            print("Depth in (\(x),\(y)): \(distanceAtXYPoint)")
                    }
//                                  depthArray.append(distancesLine)
                }

                let ciImage = CIImage(cvPixelBuffer: depth)
                let context = CIContext.init(options: nil)
                guard let cgImageRef = context.createCGImage(ciImage, from: CGRect(x: 0, y: 0, width: depthSize.width, height: depthSize.height)) else { return }
                secondImage = UIImage(cgImage: cgImageRef)

                let confidenceMap = currentFrame.sceneDepth!.confidenceMap!
                let confidenceMapCIImage = CIImage(cvPixelBuffer: confidenceMap)
                let confidenceMapContext = CIContext.init(options: nil)
                guard let cgImageRef = confidenceMapContext.createCGImage(confidenceMapCIImage, from: CGRect(x: 0, y: 0, width: depthSize.width, height: depthSize.height)) else { return }
//                secondImage = UIImage(cgImage: cgImageRef)

            }
        //======//

//        // 6
        let documentsDirectory = documentDirectory()
        let files = [documentsDirectory.appendingPathComponent("1_\(UUID().uuidString).jpeg"), documentsDirectory.appendingPathComponent("2_\(UUID().uuidString).jpeg")]
        let exampleText = ""


        do {
            // 7
            try firstImage.jpegData(compressionQuality: 0.9)!.write(to: files[0])
            try secondImage.jpegData(compressionQuality: 0.9)!.write(to: files[1])
            self.isSavingFile = false
            delegate?.didFinishSaving(path: files[0], path2: files[1])
        } catch {
            print("Failed to write PLY file", error)
        }
        }
    }
}

protocol RendererDelegate: AnyObject {
    func didFinishSaving(path: URL, path2: URL)
}
```

Figure 3.2: Contained in the Renderer.swift file, is the contents of the LiDAR data generation using the AVFoundation framework [11], as well as the storing methods for the data packet for exportation.

Another unsavoury limitation of the approach is that the LiDAR information is accounted for a down-scaled version of the original image. The downscaling is between 1440 by 1920 and 256 by 192. AVFoundation allows for a higher resolution to be considered, however, the change is minuscule, and it will increase the capture time for the data. Capture time is to be considered as it is a key factor for the efficiency analysis post experimentation (see chapter 5).

```
override func viewDidLoad() {
    super.viewDidLoad()

    let button = UIButton(type: .system, primaryAction: UIAction(title: "Save", handler: { (action) in self.renderer.savePointsToFile()
        //self.sceneView.snapshot()
    }))
```

Figure 3.3: The ViewController.swift is common on most swift programming that is responsible for the construction of the front end artifacts. This is the button in which the capture will commence upon pressing.

View Controller Script

This is standard practice in IOS development to make use of the view controller script that is mostly responsible for artifacts that appear on the screen for the user. As the name intends, the view controller is also used to place artifacts on the screen in distinct positions to better the look of the interference. However, this project is a proof of concept the interface aesthetics were not at the forefront of development hence below are the snapshots of the application in action.

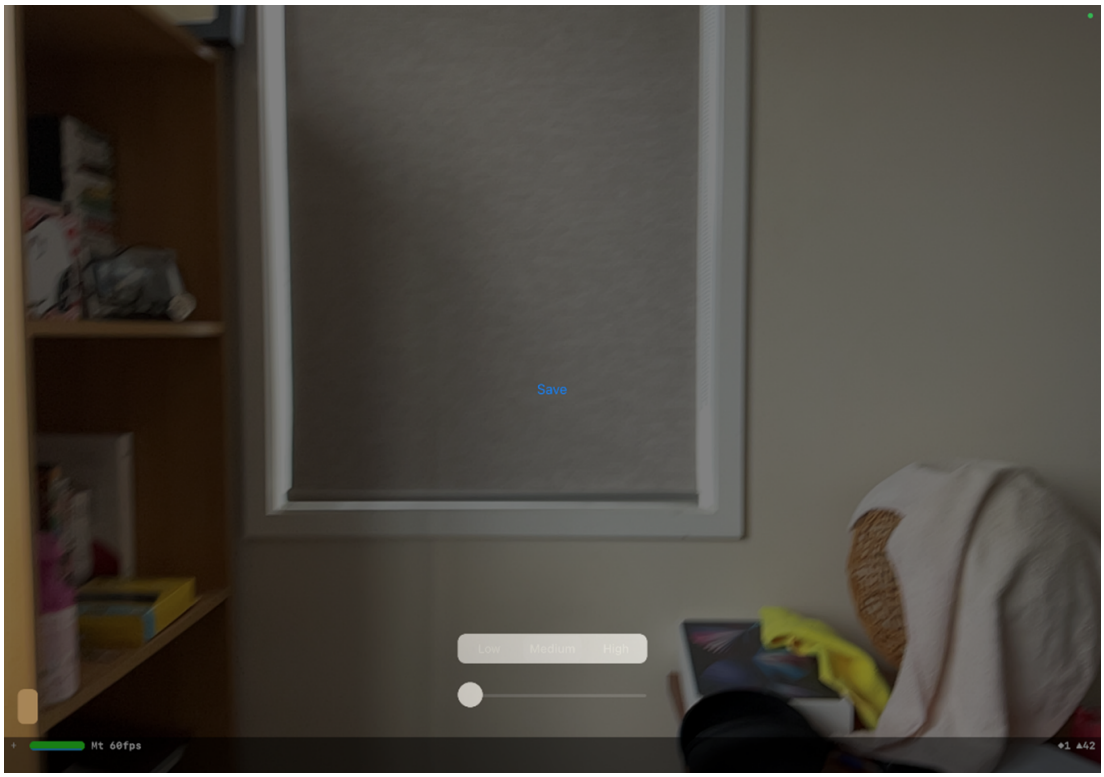


Figure 3.4: This is the initial view when the application is launched.

Figure 3.4 shows buttons located at the bottom of the screen that are non-functional from the original base script. The “Save” button in the centre of the screen is how the captures are taken.

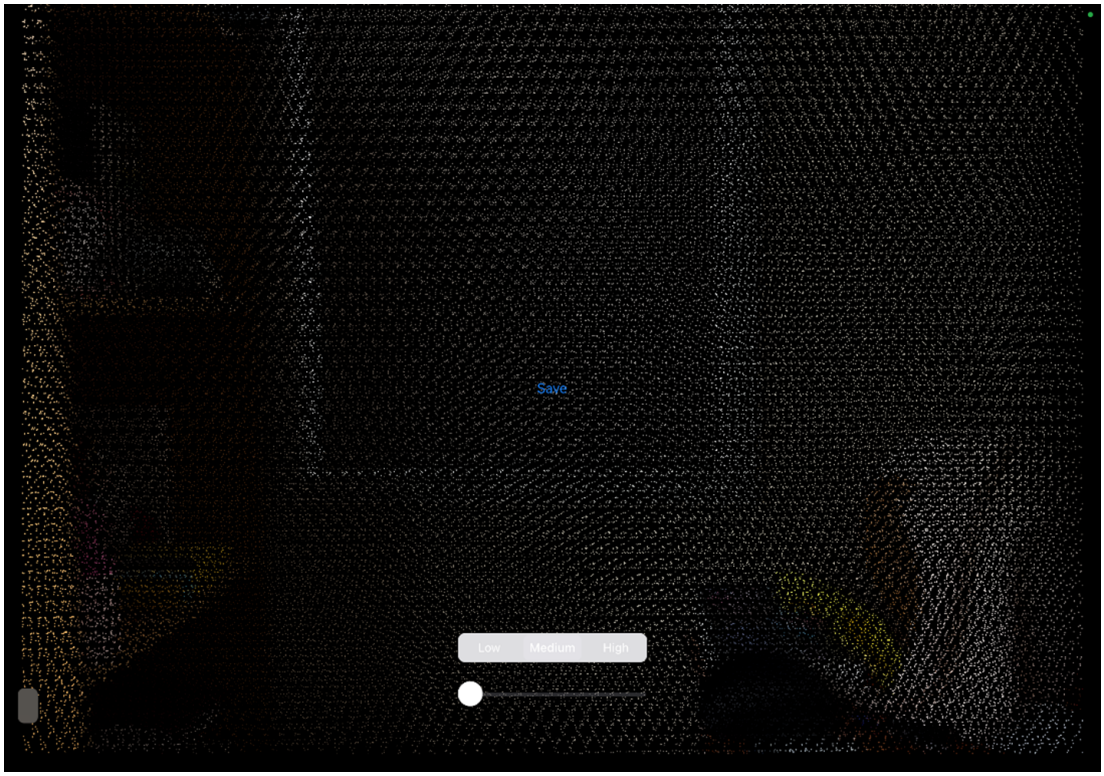


Figure 3.5: The LiDAR representation drawn on the screen originally derived from the base code.

3.2.2 Back-End Development Components

The back-end is where the input retrieved from the front is absorbed for further processing. This section will contain varying developmental approaches for the design solution.

Functions

The back-end component of the project is simple to understand in conception but was difficult to execute during development. As a concept, it is as simple as the following:

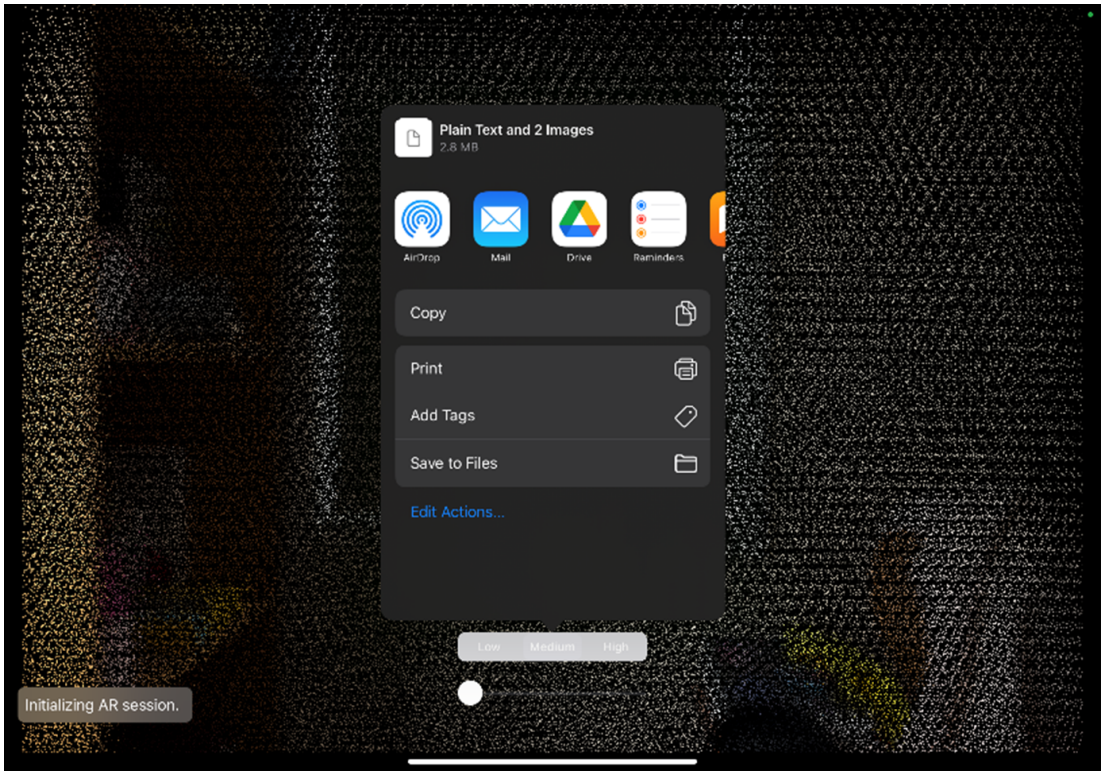


Figure 3.6: Share functionality allows for the user to choose where the information gets sent. This makes it much more flexible for some users. This will change in future iterations of the application to increase the speeds of the capture time.

- Take the input capture consisting of the following information.
  - x: Distance from the fruit(s).
  - y: Pixel Width of fruit(s).
- From the known distance x and the known pixel width y gain the z true width of the fruit.

The back-end consists of various functions which achieve concepts above. Below are the screenshots and subsequent explanations of each of these functions.

The figure 3.8 is responsible for reading CSV files containing mixed data regarding the fruit of interest that also contains the bounding box co-ordinates. The data is cleaned to exclude unwanted data and is copied to use for further processing.

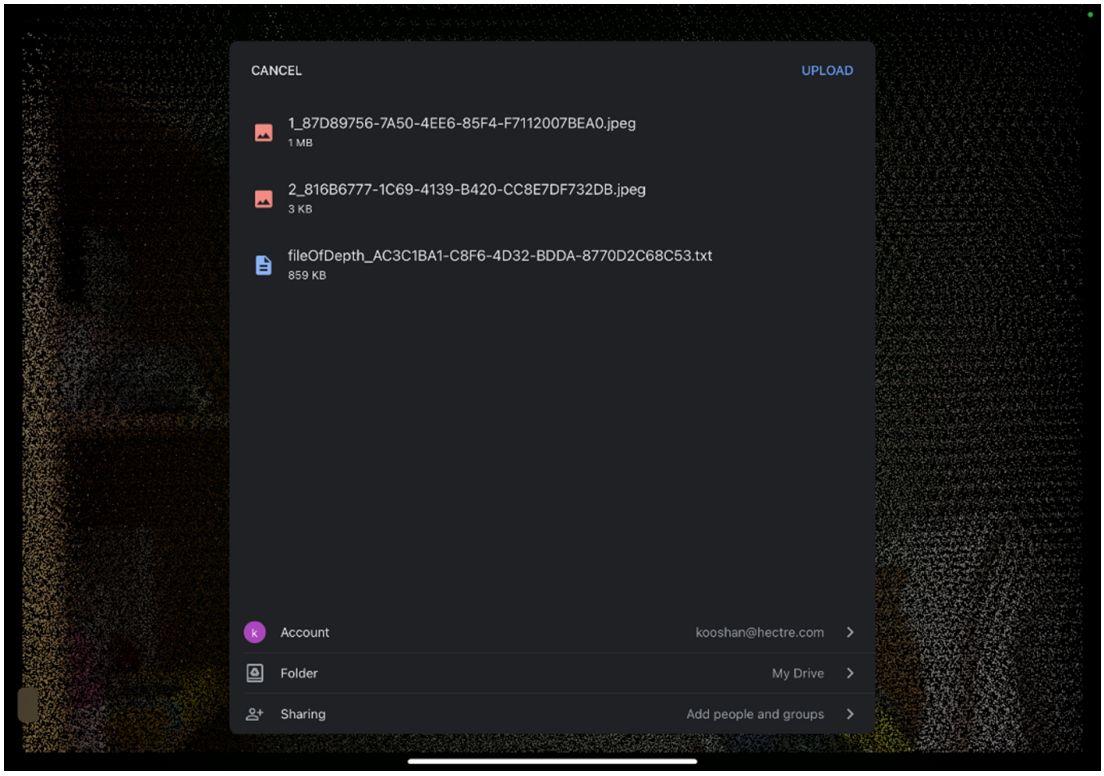


Figure 3.7: Shows the type of files that are being saved. This includes the LiDAR .txt file, depth map representation of the LiDAR information as well as the original image in JPEG format.

```
def readCSVfile():
    """
    Cleaning input CSV file to give new dataframe containing only bounding box co-ordinate.

    :return: final_data which is the copy of the dataframe containing all relevant min/max
    values of the bounding boxes.
    """
    data = pd.read_csv('/content/drive/MyDrive/output/results.csv')
    data = data.iloc[0]['Bounding Box [ymin, xmin, ymax, xmax]']
    new_list = data.split(" [")

    print(data)

    for i, w in enumerate(new_list):
        if ']]' in w:
            new_list[i] = w.replace(']]', '')
        if '[[' in w:
            new_list[i] = w.replace('[[', '')

    for i, w in enumerate(new_list):
        if '],' in w:
            new_list[i] = w.replace('],', '')

    print(new_list)

    new_data = pd.DataFrame(new_list, columns = ['Bounding Box (min_y, min_x, max_y, max_x)'])

    new_data['min_y'], new_data['min_x'], new_data['max_y'], new_data['max_x'] = new_data['Bounding Box (min_y, min_x, max_y, max_x)'].str.split(", ", 3).str

    final_data = new_data[['min_y','min_x', 'max_y', 'max_x']].copy()
    return final_data
```

Figure 3.8: The above consists of one of the essential inputs for the complete process to work correctly.

```python
def centreOfObjDistance(x1, y1, x2, y2):
    """
    Determines the centre of the object of interest and determines the distance
    based on the txt file containing all LiDAR distance value for scene.

    :param x1: Pixel x2 postion of corner bounding box based on dataframe.
    :param y1: Pixel y1 postion of corner bounding box based on dataframe.
    :param x2: Pixel x2 postion of corner bounding box based on dataframe.
    :param y2: Pixel y2 postion of corner bounding box based on dataframe.
    :return: The distance value from the object, in our case the fruit.
    """
    # Convert large image px measurement to map with depth px.
    depthX1 = x1 * 0.1328125
    depthX2 = x2 * 0.1328125
    depthY1 = y1 * 0.1326388888888889
    depthY2 = y2 * 0.1326388888888889

    # Find the midpoint between two points bounding box specific.
    midpointX = str(int((depthX2+depthX1)/2))
    midpointY = str(int((depthY2+depthY1)/2))
    # print(midpointX,midpointY)

    # File handling.
    # all_files = os.listdir("/validation")
    # txt_file = filter(lambda x: x[-4:] == '.txt', all_files)
    for file in os.listdir("/content/drive/MyDrive/MachineLearning/images/depth"):
        if file.endswith(".txt"):
            fileHandle = open("/content/drive/MyDrive/MachineLearning/images/depth/"+file, "r")
            lines = fileHandle.readlines()

    # Find the midpoint's distance.
    text = midpointX+","+midpointY+"::"
    target = ""
    for line in lines:
        if text in line:
            target = line
    fileHandle.close()

    # Separate redundant information and save as float.
    pattern = r'::'
    target = re.split(pattern, target)
    new_re = []

    for x in target:
        new_re.append(x.replace("\n",""))

    distFromObj = float(new_re[1])
    return distFromObj
```

Figure 3.9: This function is, as mentioned, responsible for the determining the estimated distance from the object of interest i.e., fruit centre.

In the figure 3.9, the return value is not only an essential component for determining the estimated true diameter of the object but also serves as a fail-safe to the ground truth measurements that are done during the experiments. This function also contains an

important conversion that is made between 2 resolution sizes of the original image being 1920px by 1440px and the LiDAR depth map being 192px by 256px (see chapter 4). This is to ensure the correct matching between the bounding box co-ordinates retrieved via machine learning and the .txt file containing every LiDAR distance measurement per pixel on the original down-scaled (see section 3.2).

```
def distance_between_pixel(x1, y1, x2, y2):
    """
    Determines the width in pixel of object of interest from bounding box co-ordinates.

    :param x1: Pixel x2 postion of corner bounding box based on dataframe.
    :param y1: Pixel y1 postion of corner bounding box based on dataframe.
    :param x2: Pixel x2 postion of corner bounding box based on dataframe.
    :param y2: Pixel y2 postion of corner bounding box based on dataframe.
    :return: Pixel distance between edge of object of interest in our case the fruit.
    """

    xUltima = x2 - x1
    yUltima = y1 - y1

    xUltima = pow(xUltima, 2)
    yUltima = pow(yUltima, 2)
    UltimaSum = xUltima + yUltima
    distance = math.sqrt(UltimaSum)

    return float(distance)
```

Figure 3.10: This function much like the previous, serves to procure a useful value from the input data frame derived via the machine learning component of the project.

Figure 3.10 is responsible to produce pixel width distance between the edges of an object of interest i.e., the fruit. The equation in question is basic and should be familiar to most people.

Length of a Line

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]} \tag{3.2}$$

The reason to employ the above equation is to exploit the bounding box co-ordinates that are produced from the machine learning model. 3.11 is a visual representation of what is occurring because of the above script 3.10.

Figure 3.11: The figure indicates the bounding box (black) co-ordinate corners given initially (red), the shifted corner (purple) and the wanted length indicated by x (blue).

The reason for point shifting explained in 3.11 is to target the horizontal cross-section of the object to ensure the bounding box is touching that of the fruit. Whereas if the original corner coordinates were used with the length of the line function it would produce a larger number due to the Pythagorean theorem. The larger length produced by the Pythagoras centre cross-section from the original corner coordinates is not of use since they do not encompass the shape of the fruit. This relates to the fact that the corners are outside of the constraints of the fruit. A consideration for future development is, for giving the averages of both the vertical cross-section and horizontal

section of the fruit which is further discussed in section 5. Because this project was tasked to be a proof-of-concept task, the simplicity was taken into account as to clearly show the validity of the processes (see 5).

```
def planarOfBestFit(data, a, b, c):
    """

    :param data: Contains depth distance and pixel distance.
    :param a: Based on previous calibration adjustable paraemter (up to 4).
    :param b: Based on previous calibration adjustable paraemter (up to 4).
    :param c: Based on previous calibration adjustable paraemter (up to 4).
    :return: The estimate measurement based on LiDAR depth information and pixel data.
    """
    # print(a)
    # print(b)
    # print(c)
    # sprint(data)
    x = data[0]
    y = data[1]

    return a*(x**b)*(y**c)
```

Figure 3.12

```
def bestFitFunction(pix_dist, depth_dist_meters):
    """
    Best fit planar is applied to input and given best estimate for diameter based
    planar of best fit function.

    :param pix_dist: Holds pixel distance derived from distance_between_pixel function.
    :param depth_dist_meters: Contains depth information derived from centreOfObjDistance.
    """
    # Planar curve parameter for the distance measurements based on two given inputs.
    parameters = [6.25e-04, 10e-01, 10e-01]

    # Unnecessary for measurement calculations.
    covariance = [[8e-10, -2e-07, -2.5e-07],
                  [-2e-07, 5e-05, 6e-05],
                  [-2.5e-07, 6e-05, 1.5e-04]]

    # Applying the 3D function for planar of best fit
    real_measurement_meters = planarOfBestFit(np.array([pix_dist,depth_dist_meters]),*parameters)
    print(str("{:.2f}".format(real_measurement_meters*1000)) + "mm")
```

Figure 3.13

The functions highlighted in 3.12 and 3.13 are where most of the heavy lifting occurs for the application. This is because it is essentially the main vertex function for deriving the true width of the fruit. The attributes a, b, c required for 3.12 are derived

from 3.13 where $parameters$ contains previously determined calibrated values (see subsection 3.2.2). The $data$ attribute in 3.12 holds the inputs of the pixel distance and distance from fruit values. 3.12 applies a Planar Curve of Best Fit equation that gives the estimated true diameter measurements. The equation in the return statement in 3.12 may seem familiar as it is a basic 3-variable Exponential Regression to Fit Function which is also applied in the calibration process 3.2.2:

$$z = a(x^b)(y^c) \tag{3.3}$$

Where the 3 variables in the case of this project is the pixel width, true width of object and distance from the object.

```
def main():
    """
    The main function that orders the process in which tasks should occur.
    """
    # read csv for all fruit bounding box max and min y and x
    boundingBoxCoords = readCSVfile()

    boundingBoxCoords['min_y'] = boundingBoxCoords['min_y'].apply(lambda x: float(x))
    boundingBoxCoords['min_x'] = boundingBoxCoords['min_x'].apply(lambda x: float(x))
    boundingBoxCoords['max_x'] = boundingBoxCoords['max_x'].apply(lambda x: float(x))
    boundingBoxCoords['max_y'] = boundingBoxCoords['max_y'].apply(lambda x: float(x))

    print("Bounding Box Co-ordinates")
    print(boundingBoxCoords)
    print("------------------------------------------------------------------------")

    count = 1
    for index, row in boundingBoxCoords.iterrows():
        print("Fruit = " + str(count))
        count += 1
        x1,y1,x2,y2 = row['min_x'], row['min_y'], row['max_x'], row['max_y']
        pix_dist = distance_between_pixel(x1, y1, x2, y2)
        depth_dist_meters = centreOfObjDistance(x1, y1, x2, y2)
        bestFitFunction(pix_dist, depth_dist_meters)
```

Figure 3.14: Is the amalgamation of all functions bound together included with it some of the separation techniques to organise the data frame data that required specific attribute type alteration.

Collectively, the above functions become the meat of the design solution for achieving the goal of gaining the size of the fruit. The beneficial aspect of the collective scripts is that they can easily be inserted into other existing projects. This is demonstrated in the design solution itself as the machine learning component of the project is provided by Hectre on a separate Google Colab file. The above scripts were simply added which then functions as intended.

Calibration

Before the experiments, it was necessary to find the trend calibration parameters for the Exponential Regression to Fit Function or the Planar Curve of the best fit function previously mentioned (see subsection 3.2.2). The calibration is done by collecting data on all known and unknown values through preliminary experiments. The difference in this experiment compared to the prior mentioned experiment loop is that the machine learning model will be void in this version. This is to identify the reliance on the varying components of the proof-of-concept including the machine learning component.

Table 3.1: Example Calibration data gathering 1.25m away from object.

Ground Truth dist. (m)	**1.25**
Real Width	Pixel Dist.
0.02	25
0.04	50
0.06	75
0.08	100
0.1	125
0.12	150
0.14	175
0.16	200
0.18	225
0.2	250
0.22	275

Calibrations conducted include distances between 0.5m and 4m gradually obtained in increments of 0.05m. Once calibration data is gained, application of the Planar Curve

Figure 3.15: Each increment on the paper placed on the wall is an increment of 50mm.

Table 3.2: Example Calibration data gathering 1.60m away from object.

Ground Truth dist. (m)	**1.60**
Real Width	Pixel Dist.
0.2	250
0.22	275
0.02	20
0.04	40
0.06	60
0.08	80
0.1	100
0.12	120
0.14	140
0.16	160
0.18	180

of Best Fit is applied (see 3.3). The planar curve produced (3.17) is then used as a vertex function where 2 input values will produce the 3 unknown values, in the case of

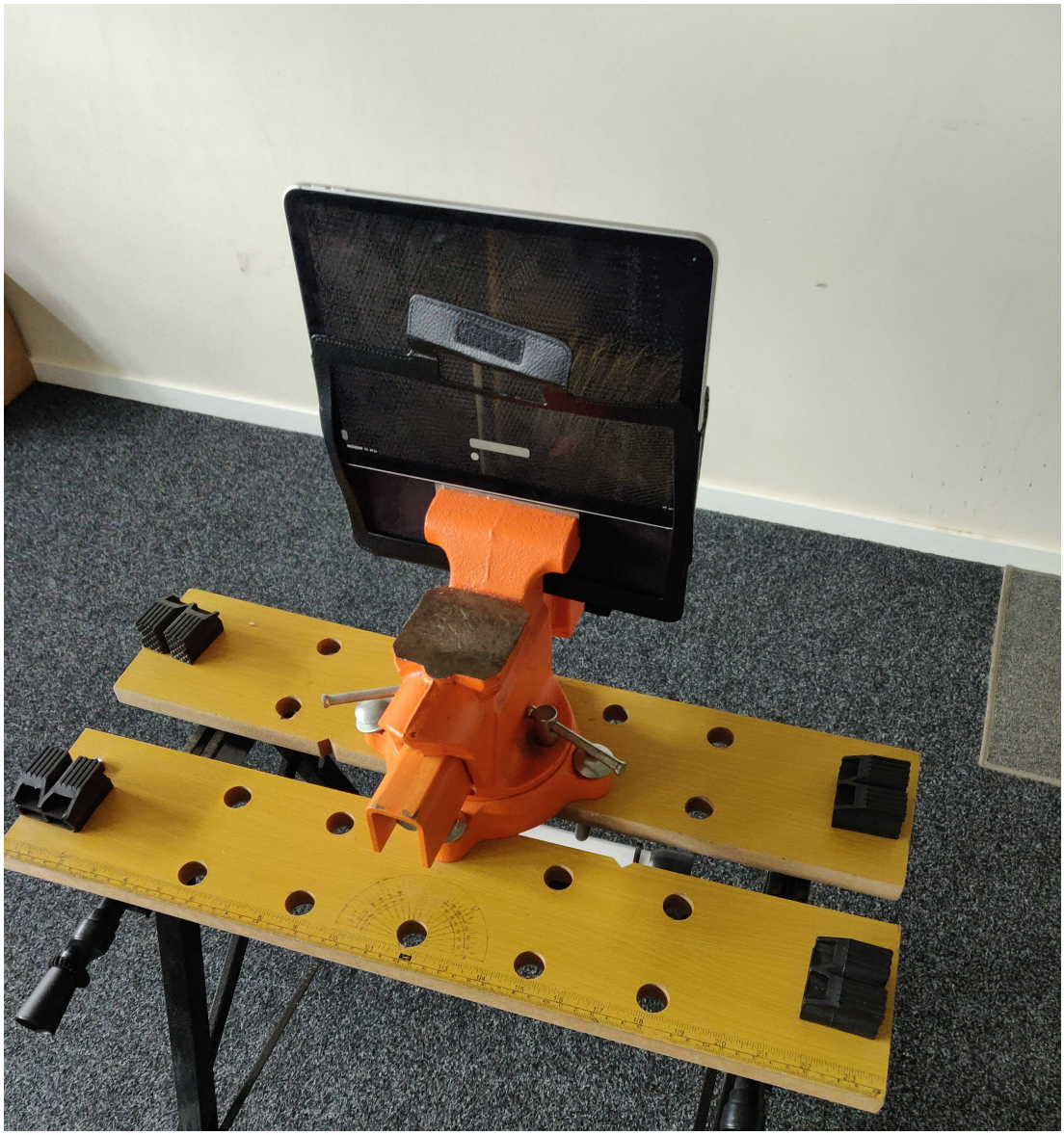


Figure 3.16: This is the makeshift tripod, used to attempt to eliminate the shaking that occurs when holding a device manually causing human error.

this project, the real width.

Methodology Limitations

As expected the main concern throughout both the calibration and experimental periods is the human error aspect which is generally common in any measurement-based studies that involve human intervention. Therefore, all results should take into consideration the

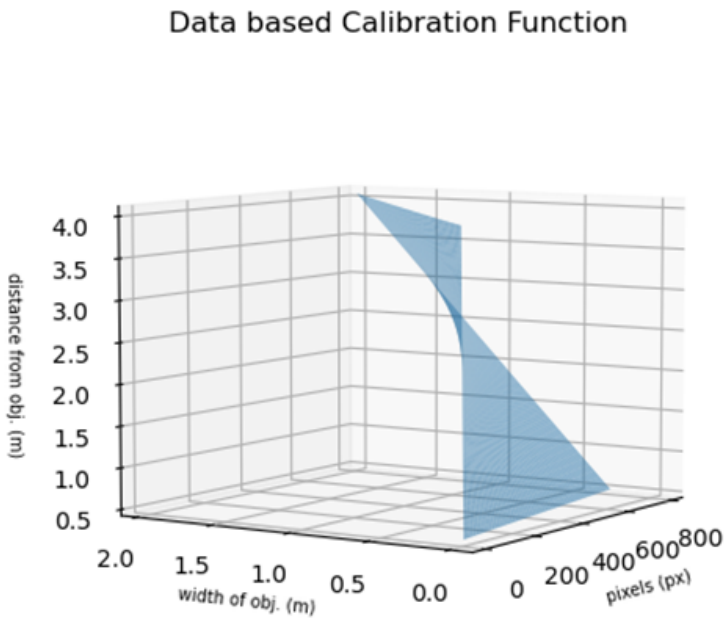


Figure 3.17: Graphical representation of the 3 main variables calibrated based on collected measurement data.

plus or minus 1% human error rate. The human error applies to the caliper calculations for ground truth, application results as it was ground truth based calibrated, as well as the distance from the fruit. The tools that will cause human errors are calipers, measuring tape, and the stopwatch. To further mitigate any external effects on the results, experiments were all done during the same period and in the same manner producing reduced possible tool-caused errors. Further limitations surrounding the proof-of-concept application itself are discussed in section 5.

3.3 Summary

This chapter consisted of detailed analysis and critique of the employed method of choice. The provided screenshots show the process in which the backend functions to gain the real distance values from the pixel information from the image captured and the distance from the object by utilising a vertex-adjacent function. The said function is calibrated via tools such as the tripod, manual measurements and collecting data to derive function parameters. In the next chapter this function is tested to determine the validity of the method of choice.

Chapter 4

Results and Findings

4.1 Results and Findings

All findings and collected data is present in the following section. The section will contain the experiments that collect data regarding the error range percentage, time taken for capture process, as well as tests against the currently accessible machine learning model for fruit count and post modification results. The experiments are ran for both an On-Tree Scenario and a bunch scenario using cherries.

Figure 4.1: An example capture taken of an apple 0.5m away from the camera origin. The Depth map (right image) represents all LiDAR data per pixel gained from the IOS device.

Table 4.1: Findings from Apple Experiment 1

LiDAR Distance (m)	Fruit Distance (m)	Estimated (mm)	Ground truth (mm)	Error (%)	Approx. Time (s)
0.52197266	0.5	70.3793118	70.38	-0.001%	16.66
0.7529297	0.75	72.64305063	70.38	3.216%	15.53
1.0107422	1	73.03072174	70.38	3.766%	15.35
1.25	1.25	74.36353102	70.38	5.660%	14.82
1.4667969	1.5	72.31008023	70.38	2.742%	15.55
1.5859375	1.6	72.14037877	70.38	2.501%	15.12
1.9912109	2	71.67805639	70.38	1.844%	15.29

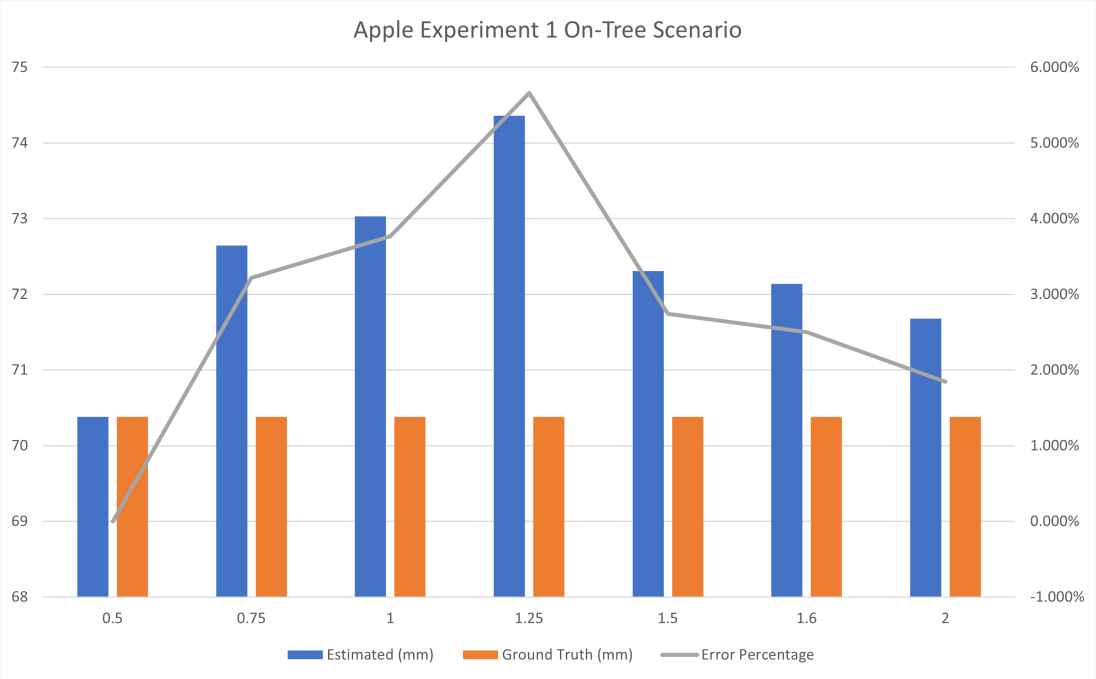


Figure 4.2: Graphical comparisons between attributes in 4.1

4.2 Retrospective

The experiments for each respective fruit were done during a single session ensuring the as much control over external factors as discussed earlier (see section 3). Aggregation of data was smoothly run although if more time was allocated for calibration higher distance ranges could be tested for although not a large concern. For on-tree scenarios, real access to an orchard for testing may increase the variety for the data as well as

Table 4.2: Findings from Apple Experiment 2

LiDAR Distance (m)	Fruit Distance (m)	Estimated (mm)	Ground truth (mm)	Error (%)	Approx. Time (s)
0.5213831	0.5	67.17328403	68.15	-1.433%	14.94
0.75122348	0.75	68.34927855	68.15	0.292%	15.87
1.003541	1	69.88664265	68.15	2.548%	14.93
1.25332	1.25	68.36353102	68.15	0.313%	16.21
1.533675	1.5	69.81017245	68.15	2.436%	16.50
1.5986313	1.6	70.11375479	68.15	2.882%	16.13
2.121	2	70.67663039	68.15	3.708%	14.61

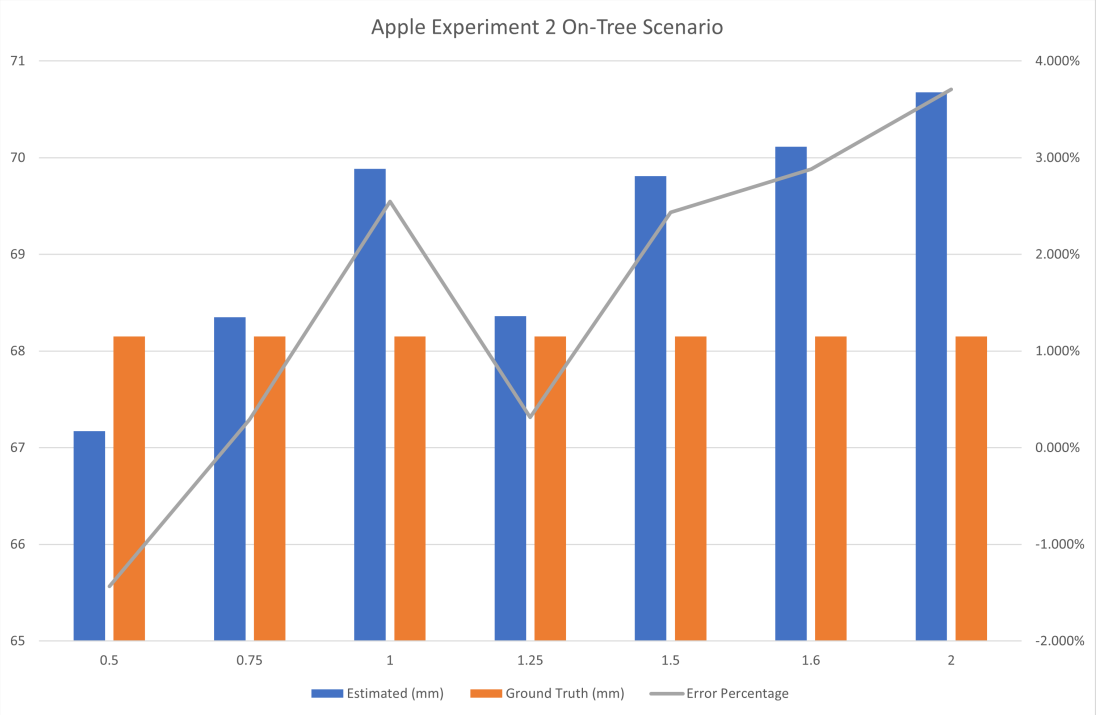


Figure 4.3: Graphical comparisons between attributes in 4.2.

conduct bunching experiments in tandem. Although discussed during section 5, the limitation of the machine learning model did effect the amount of quality data that could be recorded. Despite these factors, the experiments ran without issue and provided sufficient data which is simple to understand for the realisation of the efficacy of the proof-of-concept application. The main factors that effected the data gathering should be considered during future experimentation is the human error and machine learning

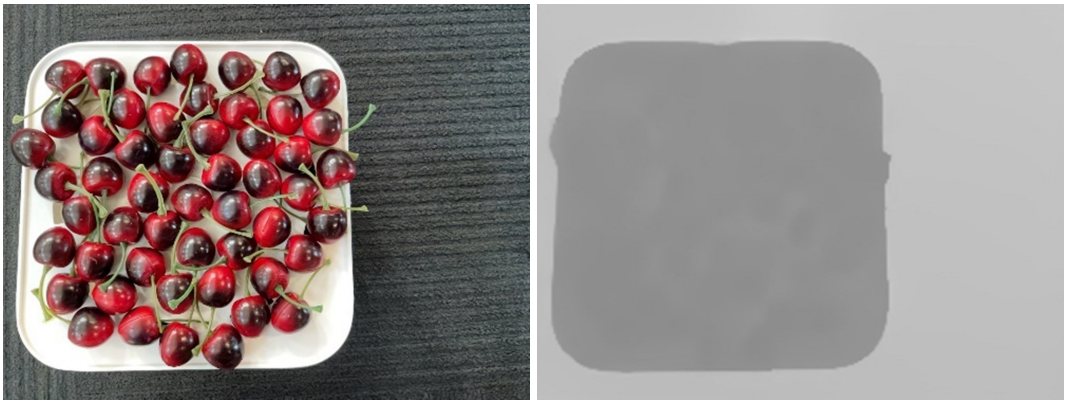


Figure 4.4: An example capture taken of a cherry bunch 0.3m away from the camera origin. The Depth map (right image) represents all LiDAR data per pixel gained from the IOS device.

Table 4.3: Findings from Cherry Experiment 1

LiDAR Distance (m)	**Fruit Distance (m)**	**Estimated (mm)**	**Ground truth (mm)**	**Error (%)**	**Approx. Time (s)**
0.3077052	0.3	26.85323097	27.89	-3.717%	16.42
0.6315052	0.6	27.01314906	27.89	-3.144%	16.20
0.921012	0.9	28.17749381	27.89	1.031%	15.04
1.280971	1.2	29.20715837	27.89	4.723%	14.65
1.5709379	1.5	28.79901473	27.89	3.259%	15.30
1.87319	1.8	29.53703816	27.89	5.906%	16.28

Table 4.4: Findings from Cherry Experiment 2

LiDAR Distance (m)	**Fruit Distance (m)**	**Estimated (mm)**	**Ground truth (mm)**	**Error (%)**	**Approx. Time (s)**
0.3025509	0.3	26.23662445	26.15	0.331%	16.28
0.611109	0.6	26.17634321	26.15	0.101%	15.30
0.91952	0.9	26.27827547	26.15	0.491%	16.29
1.246439	1.2	27.20774317	26.15	4.045%	15.11
1.537166	1.5	27.10948979	26.15	3.669%	16.49
1.8779037	1.8	27.17544928	26.15	3.921%	15.92

model (see section 5).

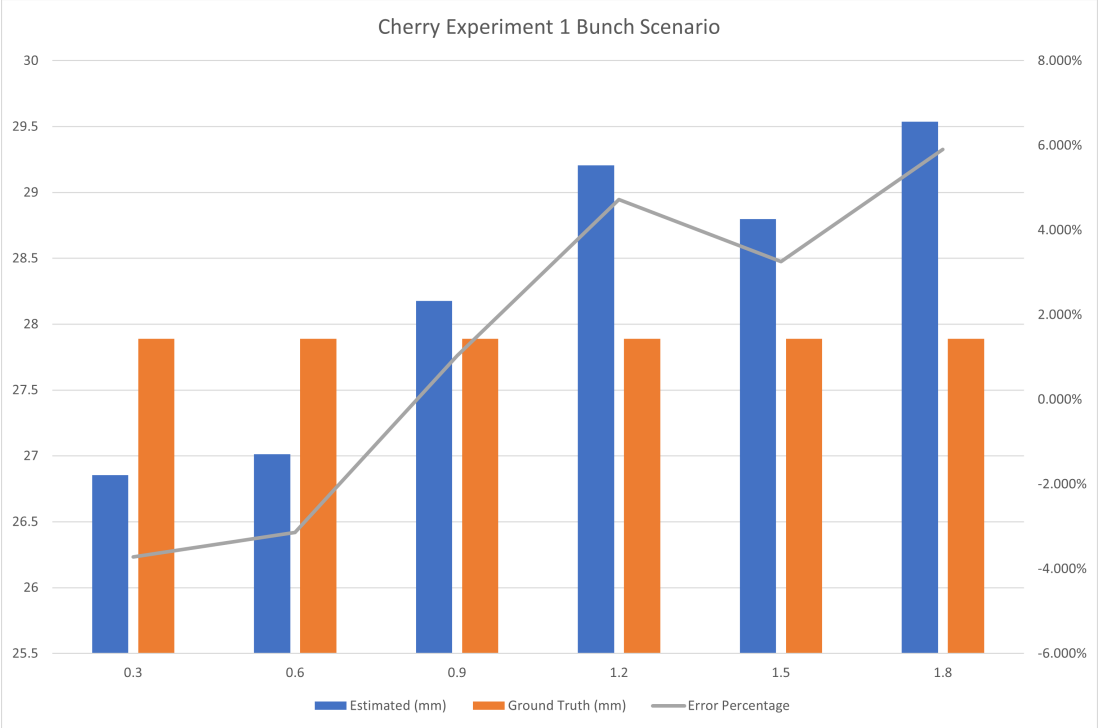


Figure 4.5: Graphical comparisons between attributes in 4.3.

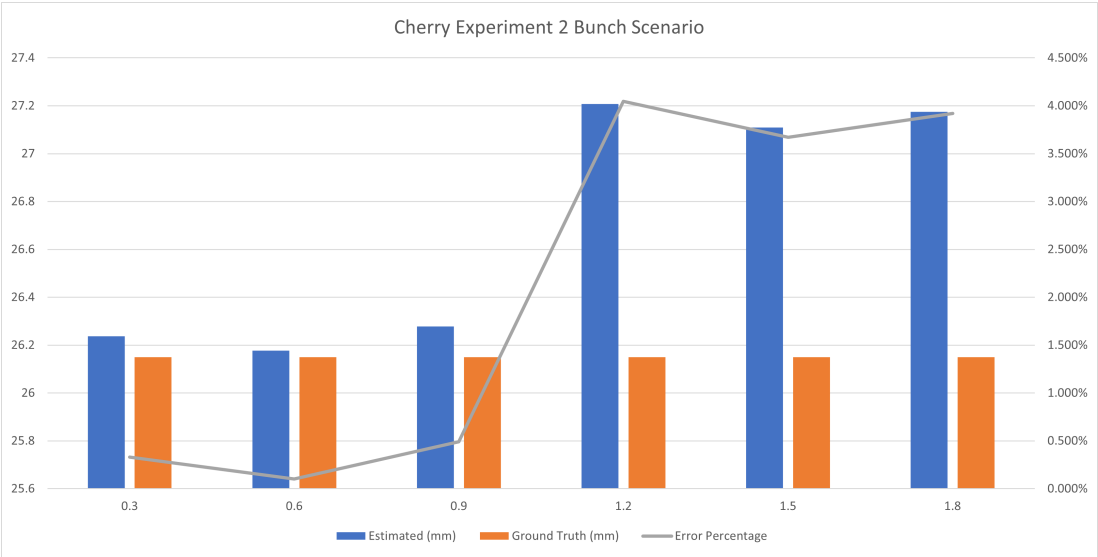


Figure 4.6: Graphical comparisons between attributes in 4.4.

4.3 Summary

The purpose of this chapter was to present data as clearly as possible, to determine the validity of the IOS application employing the aforementioned method. Diverse fruits

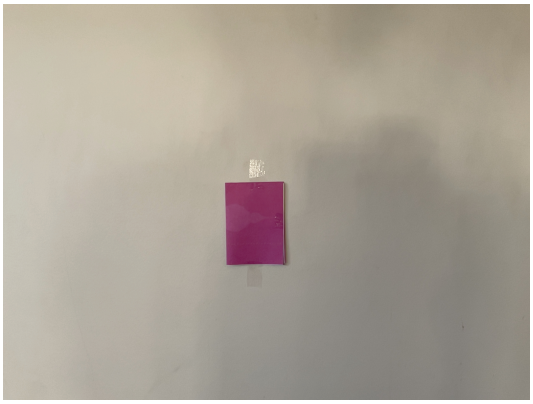


Figure 4.7: This is the specimen of interest in the captures conducted in 4.5.

Table 4.5: Findings of Flat Surface Manual Intervention Experiment

LiDAR Distance (m)	**Fruit Distance (m)**	**Estimated (mm)**	**Ground truth (mm)**	**Error (%)**	**Approx. Time (s)**
0.55322266	0.5	74.34	75	-0.880%	16.04
1.1123047	1.1	73.69	75	-1.727%	15.63
1.7089844	1.7	74.77	75	-0.307%	15.39
2.2792969	2.3	75.5	75	0.667%	16.72
2.765625	2.75	74.33	75	-0.893%	15.42
3.3632812	3.3	75.67	75	0.893%	16.41
3.9394531	4	76.33	75	1.773%	15.78
4.375	4.3	73.83	75	-1.560%	16.78
5.0273438	5	72.27	75	-3.640%	16.34

Table 4.6: Findings Post Modification Applied to the Apple Experiment 1 Dataset

LiDAR Distance (m)	**Fruit Distance (m)**	**Estimated (mm)**	**Ground truth (mm)**	**Error (%)**	**Approx. Time (s)**
0.52197266	0.5	70.3793118	70.38	-0.001%	16.47
0.7529297	0.75	70.23435051	70.38	-0.207%	16.43
1.0107422	1	70.33067959	70.38	-0.070%	15.61
1.25	1.25	70.36024972	70.38	-0.028%	15.99
1.4667969	1.5	70.31322559	70.38	-0.095%	14.95
1.5859375	1.6	70.35031596	70.38	-0.042%	15.26
1.9912109	2	70.72046553	70.38	0.484%	16.23

were shown to better show whether major discrepancies occur among varying fruit sizes. These discrepancies are discussed in the next chapter.

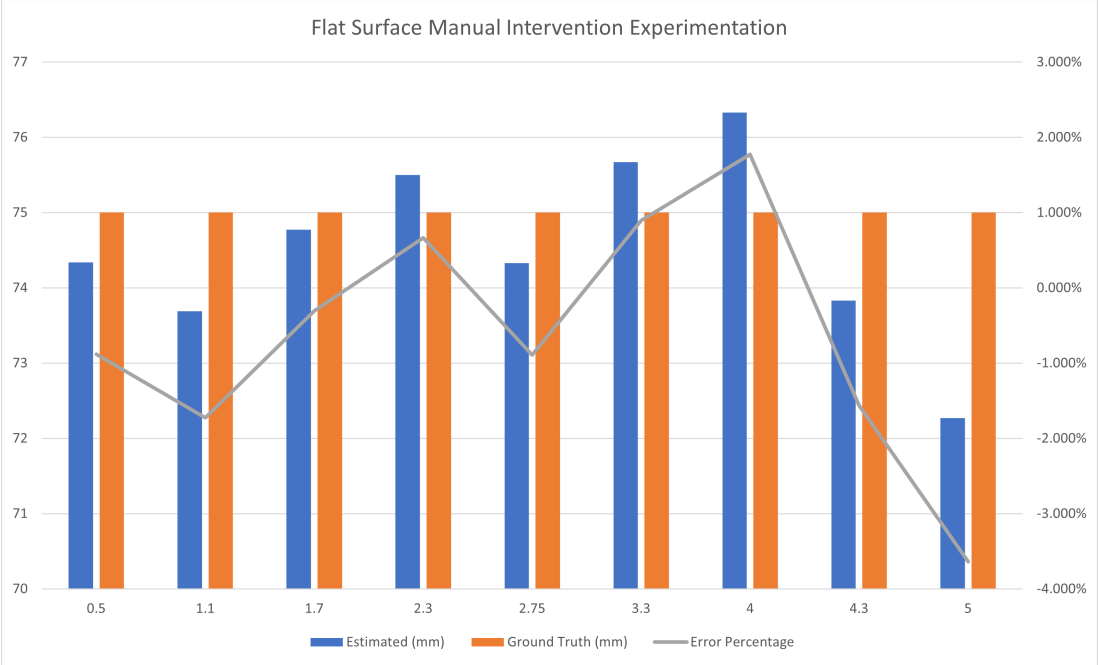


Figure 4.8: Graphical representations of 4.5. This is discussed further in 5.1.3.

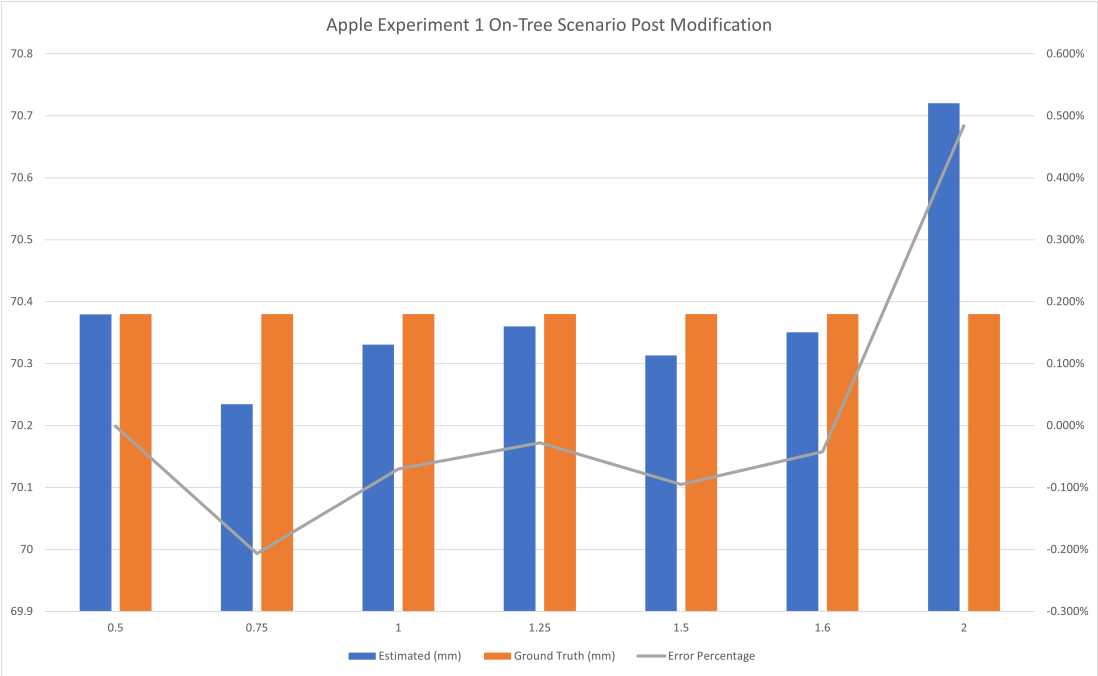


Figure 4.9: Graphical representations of 4.6. This is discussed further in 5.1.5.

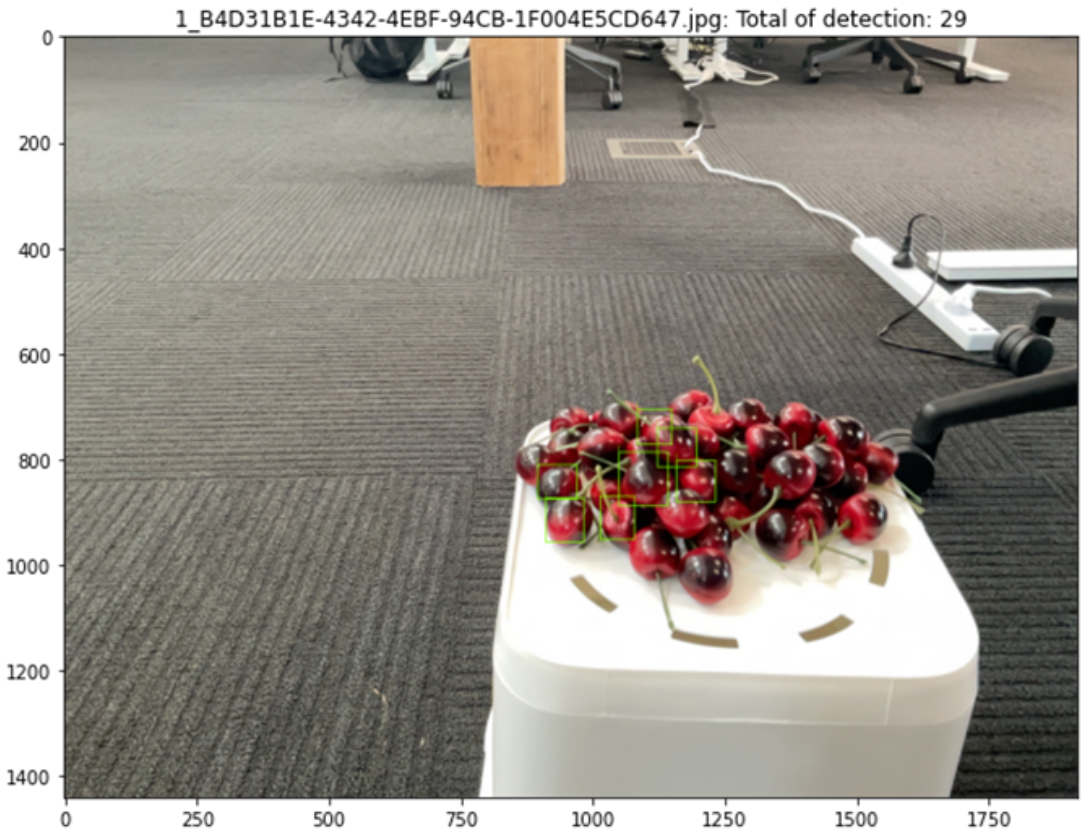


Figure 4.10: This experiment is responsible to test the machine learning model that is derivative of this paper [1]. The total cherries in this bunch are a total 52. The bounding box have indicated only 29 detected cherries.

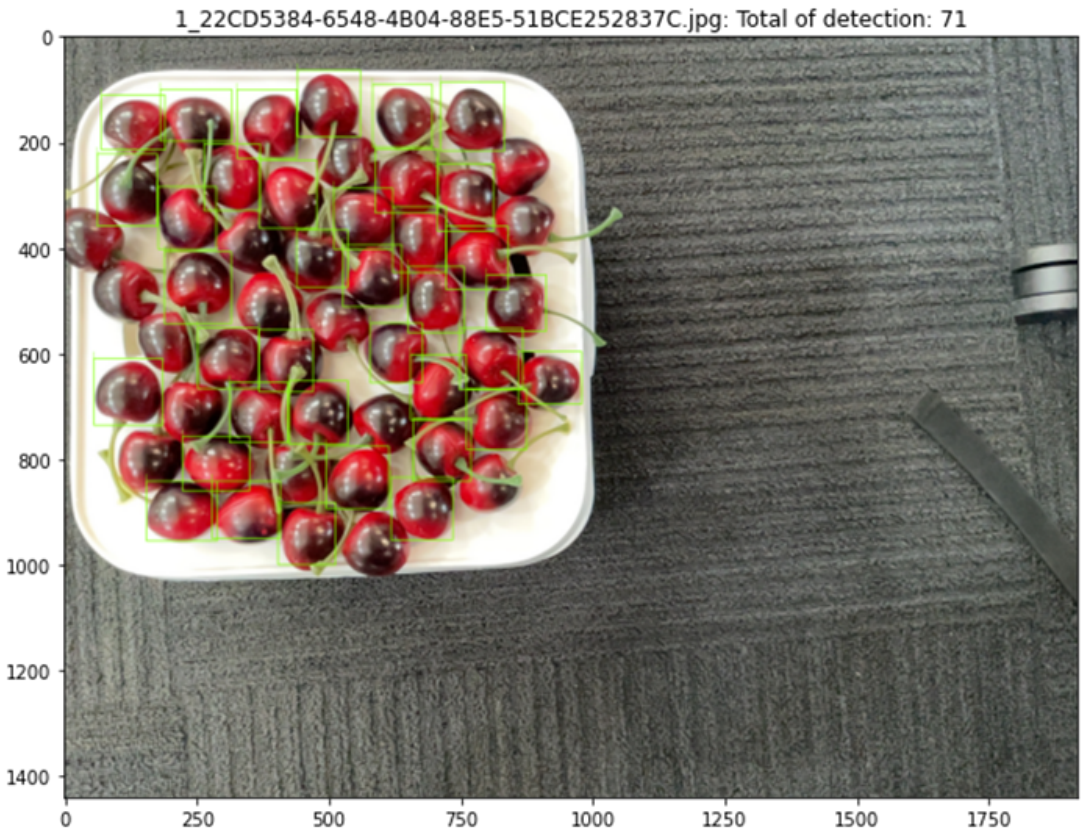


Figure 4.11: This experiment is responsible to test the machine learning model that is derivative of this paper [1]. The total cherries in this bunch are a total 52. The bounding box have indicated 71 detected cherries.

Chapter 5

Discussion

This section will cover the findings in detail as well as attempt to answer the shortcomings of the proof-of-concept application. Furthermore, the discussion will contain a dedicated component related to the benefits of the application and whether it has met the initial goals mentioned in 3.1.1.

5.1 Findings Analysis

To remind, the focus aims of the experiments is to determine the validity of the developed applications in terms of accuracy and to determine the efficiency gain over standard conventional methods (see 3.1.1). Determining these objective goals would also provide valuable information as to whether this application is worth investing further for development, time and resources.

5.1.1 Apple Experiments

To remind, the focus aims of the experiments are to determine the validity of the developed applications in terms of accuracy and to determine the efficiency gain over standard conventional methods (see 3.1.1). Determining these objective goals would

also provide valuable information as to whether this application is worth investing further in development, time, and resources.

Findings

1. Apple Experiment 1
    - Overall error range percentages between 0.5m and 2m are within 6%.
    - The worst error range percentage was 5.660% at 1.25m distance.
    - The best error range percentage is boasting -0.001% at 0.5m.
    - Despite 1 result, the trend shows steady increase in error range percentage as distance from object increases.
    - The average time taken for each capture is 15.47s.
2. Apple Experiment 2
    - Overall error range percentages between 0.5m and 2m are within 4%.
    - The worst error range percentage was 3.708% at 2m distance.
    - The best error range percentage is boasting 0.292% at 0.75m.
    - The trend shows steady increase in error range percentage as distance from object increases.
    - the average tine taken for each capture is 15.60s
3. Apple Experiment 1 Post Modification
    - Overall error range percentages between 0.5m and 2m are within 0.5%.
    - The worst error range percentage was 0.484% at 2m distance.
    - The best error range percentage is boasting -0.001% at 0.5m.

- The trend in this experiment is less obvious however, most of the error ranges are negative and show that the results are lower than the ground truth.
- The average time taken for each capture is 15.85s.

5.1.2 Cherry Experiments

Many fruit archetypes like blueberries apples and bananas tend to grow in a group-like manner. Therefore, the experiments on cherries in a bunch were required to determine whether the diameter estimations would be significantly varied compared to that of the single fruit measurements. For reference, the results that are discussed in this section are 4.3 and 4.4.

Findings

1. Cherry Experiment 1
    - Overall error range percentages between 0.5m and 1.8m are within 6%.
    - The worst error range percentage was 5.906% at 1.8m distance.
    - The best error range percentage is boasting 1.031% at 0.9m.
    - The trend shows steady increase in error range percentage as distance from object increases.
    - The average time taken for each capture is 15.65s.
2. Cherry Experiment 2
    - Overall error range percentages between 0.5m and 1.8m are within 5%.
    - The worst error range percentage was 4.045% at 1.2m distance.
    - The best error range percentage is boasting 0.101% at 0.6m.

- The trend shows steady increase in error range percentage as distance from object increases.
- the average tine taken for each capture is 15.90s

5.1.3 Flat Surface Manual Intervention Experiment

As mentioned in the data gathering strategies, a test for flat surfaces was required to compare with the results produced via fruit specimens as the data comparison could show some trend or not (see 3.1.2). The experiment also tests for longer ranges beyond the distance calibrated for (see 3.2.2). The pixel measurements are where the manual intervention occurs for this experiment which creates an extra test for whether the machine learning model has a heavy influence on the accuracy produced.

Findings

1. Flat Surface Manual Intervention Experiment
    - Overall error range percentages between 0.5m and 5m are within 4%.
    - The worst error range percentage was -3.640% at 5m distance.
    - The best error range percentage is boasting -0.307% at 1.7m.
    - There is no obvious trend in the data isolated.
    - The average time taken for each capture is 16.06s.

5.1.4 Count Experiment

This particular experiment is not so much a focused experiment on the accuracy of the design more so than it is the test of the actual machine learning model itself. As mentioned before the model was provided externally by Hectre and is a derivative model from this paper [1]. The model is designed to be detecting fruits in a bin. The results

shown in 4.10 and 4.11 are significant in that it demonstrates the need for a dedicated machine learning model.

Findings

1. Count Test 1: 29/52 Cherries detected (see Fig. 4.10).

2. Count Test 2: 71/52 Cherries detected (see Fig. 4.11).

5.1.5 Interpretations of Findings

This section attempts to understand the findings that were produced by the proof-of-concept application.

Apple and Cherry Findings Interpretations

To remind the experiments were to essentially identify the worthiness of the proof-of-concept without any manual intervention. The common trend amongst the apple experiments is that they were accurate as they boasted a low error range percentage under the 6% window which comparatively competes with alternative strategies of fruit sizing such as [123] and [15]. A surprising result is the lowest recorded error range percentage of -0.001% during the apple experiment 1. With accuracy like such, greater deductions can be made for the fruit maturities and better harvesting decisions can be made. Despite the impressive results of some readings, some readings can be considered sporadic, hence the anomaly values of 5.660% and 3.708%. Although the reasons for the production of such values are mute, there are deductions via some of the alternative values from the tests that can give hints. One said hint is the calibration itself mentioned during the calibrations section (see section 3.2.2). The reason to suspect this is the nature of how the calibration was done as well as the impact it has on the result size values. In all cases of the data, the calibration is responsible to tie in the

information between the distance away, true width of, and pixel width of/from the object (see section 3.2.2). Therefore making this component is effective in the fluctuations of the produced accuracy due to human error. Another aspect to realize is that a trend was present in the data. This trend was shown to be an increase in the error range percentage as distance increased from the fruit of interest. This could likely be a combination of issues regarding the bounding box production as well as the post-calibration parameter being applied to the 3 variable exponential curve of best fit function mentioned during the back-end development section(3.2). From these experiments an early assessment can be made, this assessment is surrounding the heavy importance of the calibration and the bounding boxes produced by the machine learning model (see 5.3.1). The results for the average time taken for both apple experiment capturing times taken were 15.47s and 15.60s respectively. What is the significance of these results? The main significance is the fact that the time taken and the accuracy produced are capable of competing with other methods such as [123], [15] and [26] at an arguably lower cost. Due to these reasons the application far exceeds the initial expectations and far exceeds the goals validity test for On-Tree measurement.

Much like the prior apple experiments the results are similar in showing the accuracy values. The trend of increased error is also displayed in both cherry experiments. This particular experiment identifies the efficacy of the proof-of-concept for smaller fruit types such as cherries or blueberries. The cherries being smaller in comparison to that of other fruits like the apple means the error range percentages induced by the application mean more in comparison to other fruits as it affects the yield results during post-analysis of orchard management. During the experiments conducted, the shortest error range percentage amongst the 2 cherry experiments is 0.101% retrieved from the 0.9m range. The highest error range percentage is 5.906% taken at 1.8m. The highest value amongst all of the experiments is the aforementioned and furthers the notion of retrieving sizes of smaller fruits. Furthermore, as observed in the apple experiments, the

trend of an increase in error range percentage is shown as the distance is increased. It is also important to note the first 2 readings of cherry experiment 1 4.3 shows negative values indicating the under-sizing of the cherry. The size of the fruit is also affected more by the parameters in the back-end part of the proof-of-concept as the human error discrepancies can apply large changes to the resulting diameter. The time results for each capture are shown to be quick just as before resulting in averages such as 15.65s and 15.90s. Overall results in the occluded bunch tests are promising for the individual cherry readings and show similar efficacy to the On-Tree apple experiments. The next section (see section 5.1.5) will explore some modifications made to apply a band-aid-fix to the standard error range percentage increase trend observed in the prior apple and cherry experiments.

Post Modification Interpretations

This particular experiment should not be considered a concrete solution to the issues that pertain to the functions in the back-end component responsible for the derivation of the estimated distance value. This is however a patchwork to the existing script to significantly increase the quality of the results based on the Apple Experiment 1 On-Tree Scenario data set. Crudely, this approach to improving the accuracy is to mimic a dynamically changing parameter.

The Fig. 5.1 shows the so-called modifications made to the back-end specifically to the script as seen in Fig. 3.13. The basic idea came to fruition after the apple experiment 2 data gathering. After noticing the increase in the error range percentage, it was worth an attempt to see if the parameters holding a static value would affect the outcome of the diameter results. The apple experiment 1 data set is chosen to employ the changes.

Although a band-aid fix, the results from the post-modification experiment (see table 4.6) are based on the original data set of apple experiment 1. Once again it would not be responsible to consider this as part of the design solution more than it is a confirmation

```
if 0.0 <= depth_dist_meters <= 1.0:
  parameters = [6.9e-04, 9.875e-01, 9.95e-01]
else:
  parameters = [6.25e-04, 10e-01, 10e-01]
```

Figure 5.1: Simple if-else statements to apply rudimentary dynamism to alter parameters based on distance from object of interest.

of the suspicion that the calibrated parameters are important to the overall outcome. The results procured through this fix are highly accurate and all error range percentages are within 1%. As this is data-set dependent it is strictly used as a matter of confirmation and future inspiration for a dynamically changing parameter as discussed further in 5.3.2.

Flat Surface Manual Intervention Interpretations

This experiment has a reason to be here and that is due to the way it will compare against the fruit-based experiments. This test will specifically test the range in which the error range percentage increases to where a drop-off range is identified, as well as provide valuable information regarding the accuracy trends compared to those experiments done on fruit. The interesting aspect of this experiment is that it provided little insight into any trends previously mentioned in 5.1.5. This experiment produced surprising results at long ranges with error range percentages not exceeding higher than 4% which is more than enough range in an orchard setting. The evidence gained has only proved that this method is comparable to competing methods much like the previously mentioned [15], [123] and [1]. Despite the positive notes, there seems to be a large increase in the error range percentage shown in the 5m capture. The cause can be directed to the calibration as only ranges between 0.5m and 4m were taken into consideration. Overall, these results are positive, proving that the calibrated parameter and pixel-perfect information

is highly important for high-accuracy estimations.

Count Findings Interpretations

Finally the test against the machine learning model derivative of the model based on scenarios as shown in [1]. Fruit counts such as 76 out of the actual 52 cherries were detected in the cherry bunch scenarios. The threshold for the model required reduction to 0.1 as values higher would procure no detection. The importance of taking these experiments and discussing them is to identify the areas that require work. The machine learning model requires reworking if this is to go further in development as the bounding boxes are a large component of accuracy as previously mentioned (see section 5.1.5). Furthermore, the rigidity of these bounding boxes shown in 4.10 shows how the threshold reductions relieve the tight attachment to the corner of the fruit. In the case of this project, the closer to the edge of the fruit the better the accuracy. Both experiments for machine learning produced lackluster results and this is simply because the scenarios that are applied during these experiments are not the standard on which the model was trained. During prior experiments to get the captures to recognize fruit several captures were taken hence the above experiments being unaffected by the situations where bounding boxes were not produced. Further discussion of the improvements of the machine learning model is made in section 5.3. Although, there were outliers as shown (see section 4). This means that the accuracy is also tied to that of the box construction and is a prerogative for future iterations to further the machine learning model to accommodate tighter boundaries and On-Tree scenarios. Another great demonstration of this experiment is the efficiency. Earlier it was noted that the times for these types of captures average between the 14 and 16 second range resulting in swift data gain in short amounts of time. This is discussed further in this section.

5.2 Validity and Viability

Although the application contains minor shortcomings, the benefits far exceed those negatives. As per section 3.1.1, the achievements will be discussed further below in section 5.2.1. Furthermore, some failed attempts the proof-of-concept will be discussed and explored.

5.2.1 Overview Goals and Failed Attempts

- Python and Swift
  - Although this goal is unrelated to the performance efficacy of the project itself yet provides general benefits for integration. Python was the choice for the back end to seamlessly merge with the existing machine learning model and save time overall. The merging was successful as the design of the code was based on a plug-and-play idea. Based on object-oriented programming, the functions were segmented to then be able to call them at later stages. This allowed for an easy transition when developing off-hand and provides further developmental benefits when modifying as discussed earlier.
  - As for the front end development there were some difficulties constructing the interface and functionalities (see section 5 Difficulties). Despite the difficulties, the interface was functional as well as the implementations of AVFoundation allowed for the retrieval of the LiDAR data captured by the sensor on the IOS device. Currently, the front end is based on prior code provided by the Apple Developer page which hosts demo code for free usage. For reference here is the specific demo that was utilized [119]. As the main point of this thesis is to provide a viable proof-of-concept, there was no need to focus on the aesthetic view of the application hence why

The interface is currently not flashy or appealing however its conception fulfills the goal of using Swift to develop a front end.

- LiDAR and Image Extraction

  - This process was done as part of the front-end and is discussed in detail in the methodology (see section 3.2). There were several previous attempts to retrieve LiDAR from the IOS device, one of the approaches revolves around the Photogrammetry approach in which point-cloud data would be extracted similar to this paper [15]. Point cloud data is universally used in Photogrammetry based tasks such as [124], [125] and [15]. The approach conceived initially for this project was to overlay point-cloud data on top of the image data to gain the location of the fruit with exact distance values from the fruit. It was also unneeded for the pixel values as the distance across the flattened point cloud was exactly in meters, therefore creating a bypass that may have produced quick processing. However, this approach was attempted and was a failure due to the sensor of the LiDAR being offset from the camera causing warping issues when it came to overlaying (see Fig. 5.2).

    Rotational matrices were also a large factor in the failed attempts as the gyroscopic orientation of the IOS affected the angles in which the point cloud would be constructed. These minor issues eventually combined to create a failed approach that did not meet the standards. This failure led to the development of an innovative approach, applying knowledge from others in the field of IOS development, and attention was drawn to the AVFoundation method. AVFoundation is a commonly used library in IOS to develop applications revolving around determining the depth values of a photograph. This method was highly beneficial as it was tied to the

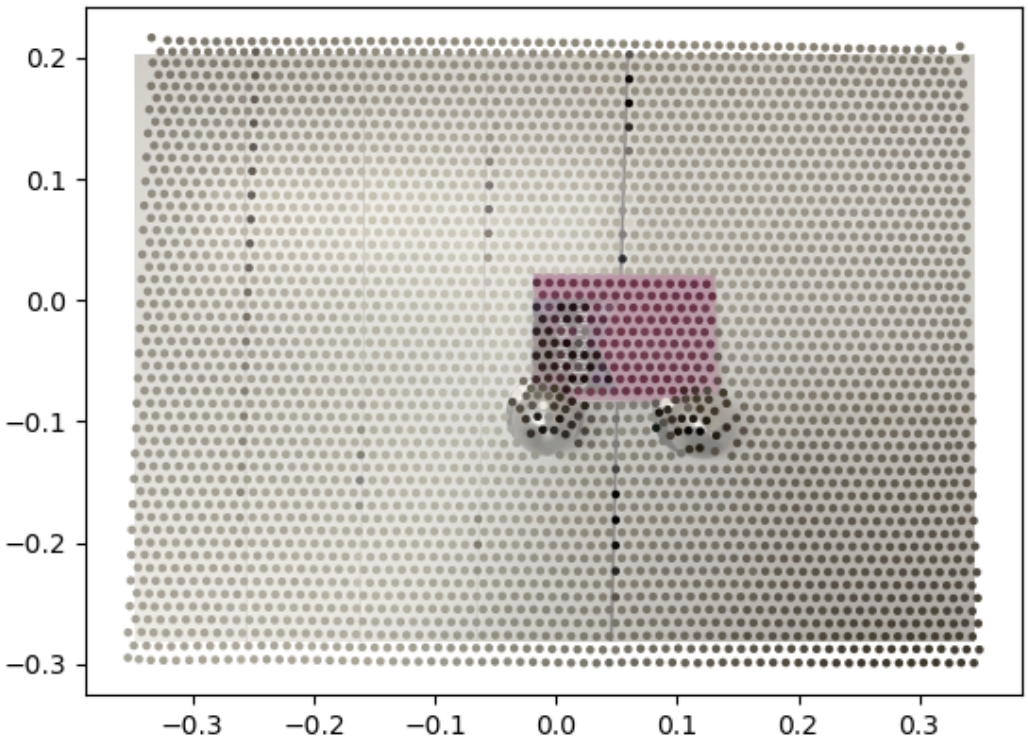


Figure 5.2: The initial point cloud approach to gain the distance information. The bottom sector of the figure displays the inaccuracies of this approach as the matching does not line up with the image.

captured image itself. There were slight drawbacks regarding the resolution downscaling for LiDAR map information, yet the results say otherwise in terms of the accuracy achieved. It also provided distance values for each pixel on the image, unlike the prior method which would be accurate but not provide distances for each pixel and would require further processing to achieve the same. The AVFoundation-based approach revolves around using the LiDAR camera like a point cloud construction yet it will follow through an estimation algorithm to determine the in-between pixels that do not get scanned. Despite the unwanted estimation-based calculations, it has proven to be of minor concern. The LiDAR values were saved in a simple

text file with specific artifacts separating the pixel locations as well as the distance values. This text file including the associated image data is part of the data packet that is sent via the share button.

Slightly less troublesome was the image-capturing method. It was based on a basic camera controller that standard IOS applications commonly apply. The extraction of the image would always result in JPEG extension images rather than JPG images, which did conflict later in the backend processing, however, a simple script was implemented to work around this to simply replace the latter extensions of the images. The resolutions were required to be high as the machine learning model was trained on decent-quality images. The prior mentioned failed attempt would not be able to produce high-resolution images as the point=cloud snapshot feature on the alternative Swift Arkit framework produces foggy image quality. The final approach that was successfully allowed for both LiDAR and image to be captured at the same time circumventing the need for matching algorithms in the backend post-processing. Finally, as mentioned earlier (see section 3.2) the extraction was done via a share button. The share button was beneficial to transfer the data packet containing all important data for post-processing as it was capable of flexibility. At this stage of the project, it was necessary to provide a method to move the files to other locations rather than a single target however it will be changing in future iterations to further decrease the average time for capture (see section 5.3.2).

- The Machine Learning Model
  - Surprisingly, this project was very minor in the involvement of the training of a model for On-Tree fruit detection. The model was based on the fruit that was in the bin scenario shown in [1]. As mentioned earlier the bounding

boxes produced by the machine learning model is the foundation for pixel width derivation in the backend. The workaround was simply lowering the threshold which alleviates the rigidity of the detection. This also meant that taking captures of consistent angles that produce the bounding boxes for the apple experiment was required. The tediousness came in finding those angles that would work with the machine learning model. Although, thanks to the speed at which captures are taken it was manageable during the experimental stages of the project. For future iterations, a new model that encompasses the true requirements of an On-Tree scenario should be considered as it is one of the backbones of the entire application gaining accurate pixel width readings.

- Retrieve distance for each fruit in the image
  - This aim came in tandem with the machine learning aspect of the project as essentially if the locations of the fruit are known it was possible to know the distance away from said fruit. Initially, there were concerns about getting unreliable readings in which distance values would miss the fruit via edge readings. This was a known issue that could arise and therefore accounted for it through the centre detection method. The centre detection is detailed in the methodology (see section 3.2) but for brevity, it utilizes basic mathematical midpoint calculations between the detected bounding box corners to isolate the centre of the fruit. This means that the target of interest is always going to be selected without obstruction or misinterpretation via the edge of the fruit. The process involves the conversion of the coordinate values gained from the machine learning to then map that onto the list of distance values in the text file. There is an unfortunate drawback that is inevitable with this approach – that being the

fact that overlapped fruit with hidden centres will not yield an accurate result. Because the midpoint detects the centre of the bounding box/fruit the centre is always the target and disregards artifacts that may obstruct that centre position yielding incorrect estimations of distance. Despite this, the results shown during the experimentation even in bunch scenarios were successful. Still, future improvements should be considered to at least account for partially covered fruit with the centre covered.

- Use all data gained to derive a size estimation of all fruits detected
  - The size estimation function is successful, and the aim was met yet there are some concerns about the approach. The one concern regarding the approach is evident in the experiments done, the ever-increasing gap in the error range percentage. The trend was observed showing an increase in the distance between the fruit and the camera. This issue is heavily tied to the backend and how the parameters affect the results. This was evident in the upgraded version of the script which was responsible for the manipulation of the parameters based on the range from the fruit of interest. The initial calibration was responsible for the resulting single parameter used for all ranges. However, a dynamic method should be explored to further the accuracy of the application. Despite this drawback trend in the error range percentage, the results are formidable with none of the values showing a range beyond 6%. This means that the goal of deriving the size estimations was met with moderate success with a bright future in terms of improvements (see section 5.3.2).
- Derive viability of method moving forward which includes time efficiency and accuracy compared to standard methods:

– All the goals have to lead to this final achievement and is the main conjecture as to whether this application is a successful proof-of-concept. The consensus based on the results is that it is a sufficient proof-of-concept that provides a substantial increase in the efficiency of yield gain in On-Tree fruit scenarios. This proof-of-concept competes with these papers [123], [15], [26] when it comes to accuracy efficacy. The main sell for this proof-of-concept is the speed at which data is collected and processed. The average times that were recorded to take one single capture of an entire scene – in this case, a tree – is between 14 and 16 seconds compared to alternative methods that claim slower processing times [15]. While the accuracy tests can be further improved (see section 5.3.2), the current version is a successful asset to those who wish to take control of their own data analysis. The conception of this lightweight novel approach to On-Tree fruit sizing will set a positive precedent in the field of smart farming. Not only does the proof-of-concept compete with those prior mentioned smart farming approaches, but compared to those common manual by-hand methods, it far exceeds in terms of efficiency. This significantly increases the sample rate compared to manual intervention methods (see section 5.2.2). As the experiments demonstrate, the application can gain many readings with adequate accuracy that will be further improved in future iterations already allowing a competitive alternative (see section 5.3.2).

5.2.2 Realistic Benefits

During this sub-section, the major benefits of this novel approach will be discussed. The benefits will compare the current standards of orchard yield analysis to this proof-of-concept application. There will be 3 main categories that will be discussed – Time

taken, Expertise, and Accuracy.

- Time: Although covered multiple times in the previous sections, the comparison to standard practice has yet to be done. The standard practice in many orchards makes use of tools such as calipers to size fruit. Few samples are usually done per tree to determine the prediction of the yield later in the year. Compared to a caliper approach the speed at which this application performs to gain multiple samples at the same time is significant. Furthermore, data availability is not barricaded by the Excel spreadsheet as it is done automatically which therefore further decreases the amount of time spent on recording data and increases time for creating an informed business decision based on the data gained.

- Expertise: Currently some orchard owners rely on those with the expertise to gain the data for the informed decisions to be made. Whether the expertise is valuable is not in question but more so the unwanted worry of relying on a third party to assess the orchard. There are also possibilities of appointments being missed and therefore causing further issues for the orchard. The reason why this is mentioned is that the sampling process is an important part of being able to deduce well-informed decisions. With the absence of a third party and the employment of the developed application, the sampling process is in the control of those who own the orchard. This will then allow orchard owners to understand their yields and gain expertise for themselves – creating better decisions for when harvest times occur.

- Accuracy of the Data: The main draw for orchard or farm adjacent owners to consider shifting resources like that of this proof of concept is the sheer amount of sampling possible. Current standards like that described in [16] show analysis that accounts for 25% of the canopy and provides estimates within 7% error range percentage. Compared to the results mentioned in [16], the application can

estimate within 6% of the true ground truth of the fruit. Furthermore, the tool can encompass the contents of the entire tree if it is visible to the camera. The data, at this stage, slightly surpasses the standard as well as demonstrates efficiency by encompassing the entire subject. This means the sampling rate is quick, accurate, not manual, and large. The emphasise on large is important as more samples mean better informed decisions.

5.3 Limitations and Future Improvements

The project while producing promising results still pertains to some significant drawbacks that either can be changed or cannot be due to the inherency of the technology itself. The section will be segmented into smaller sub-sections for stating the limitation.

5.3.1 Limitations

- Dead Angle:
  - The so-called 'dead' angle is a limitation of the application where the attempt to record the diameter size is heavily influenced by the angle/perspective of the edge. Despite the application being designed around this limitation as it simply takes the cross-section diameter across via the bounding box, it is still an issue. The limitation takes into effect mainly when the fruits are positioned in a manner where perspective obscures their true width as most fruits are not perfectly round. Because of the calibrations to get the conversion parameters for the diameter determination, the angles of the edges must be flat. This limitation is difficult to address as it would need to revolve around an entirely new approach to receiving the diameter values replacing the existing parameter function.

- Range from object:
  - This limitation has been addressed in prior sections and revolves around the limited range in which captures accurately derive the diameter size. Once again, this limitation is caused due to the lack of calibration for further ranges. At this stage, the range maximum is 4m. Despite the max calibrated range being 4m, there are significantly different diameters of the same fruit at 2m compared to 0.5m as shown in the results. This was addressed as part of a band-aid fix where the parameters would be calibrated for each range away from the object of interest and were encapsulated in a rudimentary if-else statement. This is not a permanent fix and should not be considered a fix. The best probable future fix is to create a function that can handle the varying ranges and correctly alter the parameters to produce accurate results void of the variety of data-set. This would make it robust enough to maximise the full range of the IOS LiDAR.
- Image Quality and Resolution:
  - As the machine learning models are trained on moderately high resolutions, the images that are passed into the model must be at least moderate quality. The pixel measurement adhered to in this project were 1440 by 1920 pixels. Due to the rigidity of the current code, the pixel count must be accurate as it would otherwise cause large issues when conversions occur for the retrieval of the distance values in the text file. This can be easily circumvented via a dynamic conversion function. By taking the shape of the image, it will provide the pixel dimensions of any image. After this is complete, we compare the pixels defined by the shape and auto-generate the conversion values. Fig. 5.3 shows the current resolution conversion.

```
# Convert large image px measurement to map with depth px.
depthX1 = x1 * 0.1328125
depthX2 = x2 * 0.1328125
depthY1 = y1 * 0.1326388888888889
depthY2 = y2 * 0.1326388888888889
```

Figure 5.3: The conversion values (purple).

- LiDAR inherent limitations:
  - These limitations unfortunately are out of the immediate control of the project itself as it revolves around the LiDAR sensor provided by IOS. As explored in section 2, LiDAR tends to struggle in adverse weather conditions that cause reflectivity [31], [32]. Reflectivity is an attribute in many objects, some more prevalent than others. Fruit-like apples have a reflectivity spectrum that fluctuates depending on the moisture in the air and the weather conditions such as sunshine or rain. As this thesis aimed to find the efficacy of the proof-of-concept application based mostly on the accuracy merit, hence the experiments were done in moderate sunshine conditions. The LiDAR sensor situated in the iPad pro and iPhone 12 functions at a maximum range of 5m which is more than enough for the tasks this application is responsible for. The 5m drop-off range is demonstrated during the flat surface manual intervention experiment as the error range percentage spikes (see section 4.5).
- Occluded Fruit:
  - There is an issue with the occlusion of the fruits as mentioned in similar approach papers such as [15]. When fruit is in bunches, fruits can become overlapped on top of each other. This limitation affects the method of

distance retrieval obscuring the centre of the fruit will cause distance values to fluctuate.

5.3.2 Future Considerations and Improvements

This section will piggyback off the previous, encompassing all the prior limitations, prioritising those suggested in the previous section as well as those mentioned throughout the whole body of work. This is a list of exploratory research that would benefit the robustness of the project in the future:

- Increase speed of capture and file package transfer fluidity:
  - Currently this process is fragmented and slower than it needs to be and can be further increased via a server implementation for backend processes. The time for capture would significantly increase and background upload implementation would also further the quality of life for users when deployed into a real setting.
- Account for the overlapped centres of fruit resulting in faulty distance readings:
  - As explained earlier, the distance readings are gained from the centre position of the fruit meaning any fruit with an obstruction of the centre would cause distance discrepancies. Absolving this issue would make a far more reliable tool that can easily be integrated into highly bunched fruit types such as blueberries or blackberries.
- The pixel length retrieval method:
  - Although dependent on the bounding boxes, it is the concern of how the pixels are retrieved in function calculation described in section 3. The current method accounts for one single horizontal cross-section disregarding

the vertical cross-section. By taking both the vertical pixel length and horizontal it is possible to derive an average pixel length which in turn may increase the volume calculations of a fruit.

- Consideration for varying fruit types:
  - This application will for future development be used to calculate the volume of the fruit. This is a concept that was initially disregarded due to time constraints. However, future exploratory research in this area should be considered as there are varying shape volumes for fruits. For example, cherries and mandarin volumes are different in orientation. Employing a conditional function that, based on the fruit detected will change the volume calculation based on the average volume structure of the fruit.
- Dedicated machine learning model:
  - This is one of the main improvements that require attention post-proof-of-concept. The nature of the project is based on the produced bounding boxes. Therefore, a dedicated training data set for specifically On-Tree scenarios should be collected to train a model for deployment post-proof-of-concept.
- The consideration of an automated system:
  - This application is based on IOS and currently intends to be used by humans. However, due to the lightweight nature of the product, it would be wise to consider further automation done via drones. Drones are known to be used for many modern automation tasks such as shown in these papers [126], [127], [128]. This would be farther down in terms of immediate improvements yet it should be considered as due to the lightweight and cost-efficiency of the proof-of-concept application funds can be employed towards a drone for automation.

- Occlusion investigations:
  - The occlusion problem for the detection of fruit has been recognised in papers such as [15] and is a current limitation in this proof-of-concept. The paper [15] proposes a solution of discarding heavily occluded apples resulting in lower sampling but an accurate sampling. Another consideration is the usage of a photogrammetry-based LiDAR point cloud to account for occluded fruits similar to that implemented in [15].

5.4 Summary

The discussion chapter extensively explores the findings of each experiment covering the noticeable shortcomings of the tool and benefits. As the title of the thesis has suggested, the validity and viability of the proof-of-concept application has been thoroughly assessed. Overall, the findings suggest that the validity is highly promising. With further refinement in the method and considerations to the limitations of the product, it is looking to be a highly promising application. The future prospects for the product are to mainly increase the flexibility of the machine learning aspect of the product while also applying the highly realistic tree scenarios on an orchard to further the confirmation of the validity. Despite few of the aforementioned drawbacks such as the dead angle, occlusion and LiDAR related issues the proof-of-concept has been proven and is likely to continue development.

Chapter 6

Conclusion

Finally, this thesis provides an outline of the steps taken to create a basic proof-of-concept application. The major goals for the project were to create an application that overall increases efficiency, quality of life, and accuracy using LiDAR sensors in On-Tree scenarios for fruit detection. Current manual approaches are tedious and do not appreciate the whole tree when recording the sample yield, whereas this proof-of-concept can record large amounts of data from a single tree. This project particularly builds upon this paper [1] where instead of manual intervention at any stage, sample rates can exponentially increase via the hands-off On-Tree approach. Section 2 of the paper highlights the adjacent knowledge of alternative works, covering the background work as well as the functionality of realms such as R-CNN, Smart Agriculture as well as the pre-big data era of object detection. Section 2 also provides the limitations inherently with LiDAR ranging from the lack of performance in adverse weather conditions and reflectivity issues mentioned further in section 5.3.1. Section 3 describes how development was conducted as well as the methodology of how data was to be collected. Furthermore, the intended goals or achievements were clearly defined as the following.

- Use python programming language to develop prototype backend.

- Use Swift programming language to develop prototype front-end.
- Successfully extracts valuable LiDAR and image data for backend processing.
- Use the provided machine learning model to derive the locations of fruit.
- Retrieve the distance from the detected fruit for each fruit in the image.
- Use all data gained to derive a size (diameter) estimation of all fruits detected.
- Derive viability of method moving forward which includes time efficiency and accuracy compared to current standard methods.

All of the goals mentioned were successful to the degree of a proof-of-concept standard. The viability of the proof-of-concept is key to answering the research mentioned in section 3.1.1. Section 4 surrounded simple experiments that tested the distance accuracy range percentages between 0.5m and 2m from the fruit of interest. Count experiments were done to illustrate the provided machine learning models' capabilities to determine that this model will be revamped for better results as the bounding boxes affect the accuracy of the diameter measurements. An extra experiment applied a hotfix that highlights the need for a dynamic function that alters the parameters based on the distance away from the fruit. The accuracy values gained from the experiments between the tested ranges were promising showing error range percentages within 6%. Furthermore, a flat surface manual intervention experiment was conducted to test the long-range validity of the proof-of-concept as well as further confirm the reliance on a strong machine learning model (see section 5.1.5). This draft product is on track to perform better than alternative approaches as the lack of manual intervention, the increase in sample rates per tree, and the control that the application provides to the orchard owner are highly favourable. Although this application is still a proof-of-concept it has provided enough positive leverage to consider the following future works.

- Capture speed increase via server integration and automatic upload.
- Rectify the centre overlapping issue with an alternative approach for receiving the distance of the object.
- Modify the pixel length retrieval method.
- Use the provided machine learning model to derive the locations of fruit.
- Considerations for volume difference amongst fruit types.
- Dedicated machine learning mode for On-Tree scenarios.
- Automation consideration with drone technology
- Occlusion management (related to the rectification of overlapping issues).

While the product developed may be in some regards rudimentary, it sets a positive stepping stone in the direction of efficiency in agricultural domains such as orchard management. As devices that inhabit technology such as LiDAR continue to reach the normal user, the more viable this type of approach becomes. Because the proof-of-concept application has seen success in terms of the initial goals set and questions answered, there is a promising precedent for future versions of this product to flourish. Finally, the validity of IOS-based LiDAR devices for fruit sizing has been tested and proven promising for the future market empowering those running their farms or orchards. With the advent of sampling methods derived from this proof-of-concept, manual intervention expertise-based sampling approaches may become a modicum of the past.

References

[1] L. Butters, Z. Xu, K. Le Trung, and R. Klette, "Measuring apple size distribution from a near top–down image," in *Image and Video Technology*, C. Lee, Z. Su, and A. Sugimoto, Eds. Cham: Springer International Publishing, 2019, pp. 255–268.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[3] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[5] X. Mou, X. Chen, J. Guan, B. Chen, and Y. Dong, "Marine target detection based on improved faster r-cnn for navigation radar ppi images," in *2019 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2019, pp. 1–5.

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

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

[8] Y. Li, J. Chen, M. Ke, L. Li, Z. Ding, and Y. Wang, "Small targets recognition in sar ship image based on improved ssd," in *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*. IEEE, 2019, pp. 1–6.

[9] Y. Wang, Z. Xing, L. Ma, A. Qu, and J. Xue, "Object detection algorithm for lingwu long jujubes based on the improved ssd," *Agriculture*, vol. 12, no. 9, 2022. [Online]. Available: https://www.mdpi.com/2077-0472/12/9/1456

[10] Y. Zheng and G. Wu, "Single shot multibox detector for urban plantation single tree detection and location with high-resolution remote sensing imagery," *Frontiers in Environmental Science*, p. 585, 2021.

[11] "Avfoundation documentation." [Online]. Available: https://developer.apple.com/documentation/avfoundation

[12] P. A. Roussos and D. Gasparatos, "Apple tree growth and overall fruit quality under organic and conventional orchard management," *Scientia Horticulturae*, vol. 123, no. 2, pp. 247–252, 2009.

[13] M. R. Werner, "Soil quality characteristics during conversion to organic orchard management," *Applied Soil Ecology*, vol. 5, no. 2, pp. 151–167, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0929139396001394

[14] G. M. Peck, P. K. Andrews, J. P. Reganold, and J. K. Fellman, "Apple orchard productivity and fruit quality under organic, conventional, and integrated management," *HortScience*, vol. 41, no. 1, pp. 99–107, 2006.

[15] J. Gené-Mola, R. Sanz-Cortiella, J. R. Rosell-Polo, A. Escolà, and E. Gregorio, "In-field apple size estimation using photogrammetry-derived 3d point clouds: Comparison of 4 different methods considering fruit occlusions," *Computers and Electronics in Agriculture*, vol. 188, p. 106343, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169921003604

[16] R. P. Marini, J. R. Schupp, T. A. Baugher, and R. Crassweller, "Sampling apple trees to accurately estimate mean fruit weight and fruit size distribution," *HortScience horts*, vol. 54, no. 6, pp. 1017 – 1022, 2019. [Online]. Available: https://journals.ashs.org/hortsci/view/journals/hortsci/54/6/article-p1017.xml

[17] U. Weiss, P. Biber, S. Laible, K. Bohlmann, and A. Zell, "Plant species classification using a 3d lidar sensor and machine learning," in *2010 Ninth International Conference on Machine Learning and Applications*, 2010, pp. 339–345.

[18] Y. Sumi, Y. Kawai, T. Yoshimi, and F. Tomita, "3d object recognition in cluttered environments by segment-based stereo vision," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 5–23, 2002. [Online]. Available: https://doi.org/10.1023/A:1013240031067

[19] 2017. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://link.springer.com/10.1007/978-3-319-31293-4

[20] J.-K. Lee and K.-J. Yoon, "Temporally consistent road surface profile estimation using stereo vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1618–1628, 2018.

[21] E. A. Taşdelen and V. Sezer, "Comparison and application of multiple 3d lidar fusion methods for object detection and tracking," in *2020 5th International Conference on Robotics and Automation Engineering (ICRAE)*. IEEE, 2020, pp. 64–69.

[22] L. Fu, F. Gao, J. Wu, R. Li, M. Karkee, and Q. Zhang, "Application of consumer rgb-d cameras for fruit detection and localization in field: A critical review," *Computers and Electronics in Agriculture*, vol. 177, p. 105687, 2020.

[23] R. M. Perez, F. A. Cheein, and J. R. Rosell-Polo, "Flexible system of multiple rgb-d sensors for measuring and classifying fruits in agri-food industry," *Computers and Electronics in Agriculture*, vol. 139, pp. 231–242, 2017.

[24] S. Cubero, N. Aleixos, E. Moltó, J. Gómez-Sanchis, and J. Blasco, "Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables," *Food and bioprocess technology*, vol. 4, no. 4, pp. 487–504, 2011.

[25] D. Stajnko, M. Lakota, and M. Hočevar, "Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging," *Computers and Electronics in Agriculture*, vol. 42, no. 1, pp. 31–42, 2004.

[26] Z. Wang, K. B. Walsh, and B. Verma, "On-tree mango fruit size estimation using rgb-d images," *Sensors*, vol. 17, no. 12, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/12/2738

[27] V. Méndez, A. Pérez-Romero, R. Sola-Guirado, A. Miranda-Fuentes, F. Manzano-Agugliaro, A. Zapata-Sierra, and A. Rodríguez-Lizana, "In-field estimation of orange number and size by 3d laser scanning," *Agronomy*, vol. 9, no. 12, p. 885, 2019.

[28] N. Tsoulias, D. S. Paraforos, G. Xanthopoulos, and M. Zude-Sasse, "Apple shape detection based on geometric and radiometric features using a lidar laser scanner," *Remote Sensing*, vol. 12, no. 15, p. 2481, 2020.

[29] A. Boukerche and H. O. U. Zhijun, "Object detection using deep learning methods in traffic scenarios," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–35, 2021. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=bth&AN=150035762

[30] M. Dassot, T. Constant, and M. Fournier, "The use of terrestrial lidar technology in forest science: Application fields, benefits and challenges," *Annals of Forest Science*, vol. 68, pp. 959–974, 08 2011.

[31] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, "Cnn-based lidar point cloud de-noising in adverse weather," *IEEE Robotics and Automation Letters, Robotics and Automation Letters, IEEE, IEEE Robot.*

Autom. Lett., vol. 5, no. 2, pp. 2514–2521, 2020. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edseee&AN=edseee.8990038

[32] A. Rogalski and K. Chrzanowski, "Infrared devices and techniques," *Optoelectronics Review*, vol. 10, no. 2, pp. 111–136, 2002.

[33] R. Ganeev, *Laser - Surface Interactions*. Springer Netherlands, 2013. [Online]. Available: https://books.google.co.nz/books?id=4KUpngEACAAJ

[34] T. Ito, "Lidar for automated driving era." Institute of Electrical and Electronics Engineers Inc., Conference Proceedings, pp. 37–40–40. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edselc&AN=edselc.2-52.0-85124192148

[35] V. S. Saravanarajan, R. C. Chen, and L. S. Chen, "Lidar point cloud data processing in autonomous vehicles." Institute of Electrical and Electronics Engineers Inc., Conference Proceedings. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edselc&AN=edselc.2-52.0-85123452635

[36] Y. Wu, Y. Wang, S. Zhang, and H. Ogai, "Deep 3d object detection networks using lidar data: A review," *IEEE Sensors Journal, Sensors Journal, IEEE, IEEE Sensors J.*, vol. 21, no. 2, pp. 1152–1171, 2015. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edseee&AN=edseee.9181591

[37] L. Fuqiang and W. Zongyi, "Polishnet-2d and polishnet-3d: Deep learning-based workpiece recognition," *IEEE Access*, vol. 7, pp. 127 042–127 054, 2019. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edsdoj&AN=edsdoj.088f54e2034f4cb29e1bc81930a09e35

[38] E. Tudor, I. Vasile, G. Popa, and M. Gheti, "Lidar sensors used for improving safety of electronic-controlled vehicles." Institute of Electrical and Electronics Engineers Inc., Conference Proceedings. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edselc&AN=edselc.2-52.0-85106730649

[39] pp. 1–4, 2020. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edseee&AN=edseee.9181291

[40] M. S. M. Saleh, A. A. B. Sajak, R. Mohamad, and M. A. M. Zaaba, "Iot real-time soil monitoring based on lora for palm oil plantation," vol. 1874. IOP Publishing Ltd, Conference Proceedings. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edselc&AN=edselc.2-52.0-85109079192

[41] R. Anzum and J. Naeem, "Leveraging lorawan technology for smart agricultural monitoring of malaysian palm oil plantation," vol. 756. IOP Publishing Ltd, Conference Proceedings. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edselc&AN=edselc.2-52.0-85107201582

[42] pp. 1–8, 2021. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edseee&AN=edseee.9550981

[43] G. Luetzenburg, A. Kroon, and A. A. Bjørk, "Evaluation of the apple iphone 12 pro lidar for an application in geosciences," *Scientific Reports*, vol. 11, no. 1, 2021. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edselc&AN=edselc.2-52.0-85119073317

[44] B. Sirmacek and R. Lindenbergh, "Accuracy assessment of building point clouds automatically generated from iphone images," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5, pp. 547–552, 2014. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edsdoj&AN=edsdoj.7feebea960f94c1e9e0061305d63de1f

[45] M. D. Harley, M. A. Kinsela, E. Sánchez-García, and K. Vos, "Shoreline change mapping using crowd-sourced smartphone images," *Coastal Engineering*, vol. 150, pp. 175–189, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378383918304551

[46] M. Jaud, M. Kervot, C. Delacourt, and S. Bertin, "Potential of smartphone sfm photogrammetry to measure coastal morphodynamics," *Remote Sensing*, vol. 11, no. 19, 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/19/2242

[47] C. Gollob, T. Ritter, R. Kraßnitzer, A. Tockner, and A. Nothdurft, "Measurement of forest inventory parameters with apple ipad pro and integrated lidar technology," *Remote Sensing*, vol. 13, no. 16, 2021. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edselc&AN=edselc.2-52.0-85113712296

[48] M. Mokroš, T. Mikita, A. Singh, J. Tomaštík, J. Chudá, P. Wężyk, K. Kuželka, P. Surový, M. Klimánek, K. Zięba-Kulawik, R. Bobrowski, and X. Liang, "Novel low-cost mobile mapping systems for forest inventories as terrestrial laser scanning alternatives," *International Journal of Applied Earth Observation and Geoinformation*, vol. 104, p. 102512, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303243421002191

[49] M. Khanal, M. Hasan, N. Sterbentz, R. Johnson, and J. Weatherly, "Accuracy comparison of aerial lidar, mobile-terrestrial lidar, and uav photogrammetric

capture data elevations over different terrain types," *Infrastructures*, vol. 5, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2412-3811/5/8/65

[50] pp. 13–18, 2020. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edseee&AN=edseee.9189314

[51] A. Fernández, A. Apituley, I. Veselovskii, A. Suvorina, J. Henzing, M. Pujadas, and B. Artíñano, "Study of aerosol hygroscopic events over the cabauw experimental site for atmospheric research (cesar) using the multi-wavelength raman lidar caeli," *Atmospheric Environment*, vol. 120, pp. 484–498, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1352231015303241

[52] A. P. Cracknell, *Introduction to remote sensing*. CRC press, 2007.

[53] J. M. Bonisteel, A. Nayegandhi, C. W. Wright, J. C. Brock, and D. B. Nagle, "Experimental advanced airborne research lidar (eaarl) data processing manual," *US Geological Survey Open-File Report*, vol. 1078, p. 38, 2009.

[54] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.-K. Papastathis, and M. G. Strintzis, "Knowledge-assisted semantic video object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1210–1224, 2005.

[55] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[56] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[57] Y.-Q. Wang, "An Analysis of the Viola-Jones Face Detection Algorithm," *Image Processing On Line*, vol. 4, pp. 128–148, 2014, https://doi.org/10.5201/ipol.2014.104.

[58] B. D. Lucas, T. Kanade *et al.*, *An iterative image registration technique with an application to stereo vision*. Vancouver, 1981, vol. 81.

[59] A. R. Hurson, S. Wu, A. Hurson, and S. Wu, *AI and Cloud Computing*. San Diego, UNITED STATES: Elsevier Science Technology, 2021. [Online]. Available: http://ebookcentral.proquest.com/lib/aut/detail.action?docID=6456159

[60] T. Lindeberg, "Scale-space theory: A basic tool for analyzing structures at different scales," *Journal of applied statistics*, vol. 21, no. 1-2, pp. 225–270, 1994.

[61] Y. Liu, M. He, Y. Wang, Y. Sun, and X. Gao, "Farmland aerial images fast-stitching method and application based on improved sift algorithm," *IEEE Access*, vol. 10, pp. 95 411–95 424, 2022.

[62] Y. Cai, L. Li, S. Ni, J. Lv, W. Zeng, and Y. Yuanlong, "Moving vehicle detection based on dense sift and extreme learning machine for visual surveillance," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, pp. 1614–1618.

[63] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[64] Z. Wang, W. Bao, D. Lin, and Z. Wang, "A local feature descriptor based on sift for 3d pollen image recognition," *IEEE Access*, vol. 7, pp. 152 658–152 666, 2019.

[65] C. H. Katelaris and P. J. Beggs, "Climate change: allergens and allergic diseases," *Internal Medicine Journal*, vol. 48, no. 2, pp. 129–134, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/imj.13699

[66] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: https://doi.org/10.1007/s11263-013-0620-5

[67] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000022288.19776.77

[68] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring features in a bayesian framework for material recognition," in *2010 ieee computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 239–246.

[69] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[70] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[71] X. Chen, Y. Huang, J. Guan, and Y. He, "Sea clutter suppression and moving target detection method based on clutter map cancellation in frft domain," in *Proceedings of 2011 IEEE CIE International Conference on Radar*, vol. 1. IEEE, 2011, pp. 438–441.

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

[73] N. Bodla, B. Singh, R. Chellappa, and L. Davis, "Improving object detection with one line of code. arxiv 2017," *arXiv preprint arXiv:1704.04503*.

[74] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 784–799.

[75] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[76] T. Zhang, X. Zhang, J. Li, and J. Shi, "Contextual squeeze-and-excitation mask r-cnn for sar ship instance segmentation," in *2022 IEEE Radar Conference (RadarConf22)*. IEEE, 2022, pp. 1–6.

[77] Z. Zhang, F. Gui, X. Qu, and D. Feng, "Netting damage detection for marine aquaculture facilities based on improved mask r-cnn," *JOURNAL OF MARINE SCIENCE AND ENGINEERING*, vol. 10, no. 7, p. 996, 2022. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edswsc&AN=000832321700001

[78] S. Wei, H. Chen, X. Zhu, and H. Zhang, "Ship detection in remote sensing image based on faster r-cnn with dilated convolution," in *2020 39th Chinese Control Conference (CCC)*. IEEE, 2020, pp. 7148–7153.

[79] C. K. Albuquerque, S. Polimante, A. Torre-Neto, and R. C. Prati, "Water spray detection for smart irrigation systems with mask r-cnn and uav footage," in *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*. IEEE, 2020, pp. 236–240.

[80] D. Soekhoe, P. van der Putten, and A. Plaat, "On the impact of data set size in transfer learning using deep neural networks," in *Advances in Intelligent Data Analysis XV*, H. Boström, A. Knobbe, C. Soares, and P. Papapetrou, Eds. Cham: Springer International Publishing, 2016, pp. 50–60.

[81] M. Deserno and A. Briassouli, "Faster r-cnn and efficientnet for accurate insect identification in a relabeled yellow sticky traps dataset," in *2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*. IEEE, 2021, pp. 209–214.

[82] L. Zhang, R. Li, Z. Li, Y. Meng, J. Liang, L. Fu, X. Jin, and S. Li, "A quadratic traversal algorithm of shortest weeding path planning for agricultural mobile robots in cornfield," *Journal of Robotics*, vol. 2021, 2021.

[83] R. Bogue, "Robots poised to revolutionise agriculture," *Industrial Robot: An International Journal*, 2016.

[84] O. Bawden, J. Kulk, R. Russell, C. McCool, A. English, F. Dayoub, C. Lehnert, and T. Perez, "Robot for weed species plant-specific management," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1179–1199, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21727

[85] A. M. A. Talab, Z. Huang, F. Xi, and L. HaiMing, "Detection crack in image using otsu method and multiple filtering in image processing techniques," *Optik*, vol. 127, no. 3, pp. 1030–1033, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030402615012164

[86] S. Tu, J. Pang, H. Liu, N. Zhuang, Y. Chen, C. Zheng, H. Wan, and Y. Xue, "Passion fruit detection and counting based on multiple scale faster r-cnn using rgb-d images," *Precision Agriculture*, vol. 21, no. 5, pp. 1072–1091, 2020. [Online]. Available: https://doi.org/10.1007/s11119-020-09709-3

[87] S. Tu, Y. Xue, C. Zheng, Y. Qi, H. Wan, and L. Mao, "Detection of passion fruits and maturity classification using red-green-blue depth images," *Biosystems Engineering*, vol. 175, pp. 156–167, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511017309844

[88] Y. Wang, C. Wang, and H. Zhang, "Combining a single shot multibox detector with transfer learning for ship detection using sentinel-1 sar images," *Remote Sensing Letters*, vol. 9, no. 8, pp. 780–788, 2018. [Online]. Available: https://doi.org/10.1080/2150704X.2018.1475770

[89] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, "Beyond skip connections: Top-down modulation for object detection," *arXiv preprint arXiv:1612.06851*, 2016.

[90] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[91] K. Yu, Y. Zhao, X. Li, Y. Shao, F. Zhu, and Y. He, "Identification of crack features in fresh jujube using vis/nir hyperspectral imaging combined with image processing," *Computers and Electronics in Agriculture*, vol. 103, pp. 1–10, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169914000337

[92] J. Chen, X. Liu, Z. Li, A. Qi, P. Yao, Z. Zhou, T. T. Dong, and K. W. Tsim, "A review of dietary ziziphus jujuba fruit (jujube): Developing health food supplements for brain protection," *Evidence-Based Complementary and Alternative Medicine*, vol. 2017, 2017.

[93] L. Wu, J. He, G. Liu, S. Wang, and X. He, "Detection of common defects on jujube using vis-nir and nir hyperspectral imaging," *Postharvest Biology and Technology*, vol. 112, pp. 134–142, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925521415301083

[94] L. Geng, W. Xu, F. Zhang, Z. Xiao, and Y. Liu, "Dried jujube classification based on a double branch deep fusion convolution neural network," *Food Science and Technology Research*, vol. 24, no. 6, pp. 1007–1015, 2018.

[95] A. M. Al-Saif, M. Abdel-Sattar, A. M. Aboukarima, and D. H. Eshra, "Identification of indian jujube varieties cultivated in saudi arabia using an artificial neural network," *Saudi Journal of Biological Sciences*, vol. 28, no. 10, pp. 5765–5772, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319562X21004794

[96] X. Luo, B. Ma, W. Wang, S. Lei, Y. Hu, G. Yu, and X. Li, "Evaluation of surface texture of dried hami jujube using optimized support vector machine based on visual features fusion," *Food Science and Biotechnology*, vol. 29, no. 4, pp. 493–502, 2020. [Online]. Available: https://doi.org/10.1007/s10068-019-00683-9

[97] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," *Advances in neural information processing systems*, vol. 31, 2018.

[98] M. I. Faridatul and B. Wu, "Automatic classification of major urban land covers based on novel spectral indices," *ISPRS International Journal of Geo-Information*, vol. 7, no. 12, 2018. [Online]. Available: https://www.mdpi.com/2220-9964/7/12/453

[99] R. Ahl, J. Hogland, and S. Brown, "A comparison of standard modeling techniques using digital aerial imagery with national elevation datasets and airborne lidar to predict size and density forest metrics in the sapphire mountains mt, usa," *ISPRS International Journal of Geo-Information*, vol. 8, no. 1, 2019. [Online]. Available: https://www.mdpi.com/2220-9964/8/1/24

[100] M. Mokroš, X. Liang, P. Surový, P. Valent, J. Čerňava, F. Chudý, D. Tunák, Saloň, and J. Merganič, "Evaluation of close-range photogrammetry image collection methods for estimating tree diameters," *ISPRS International Journal of Geo-Information*, vol. 7, no. 3, 2018. [Online]. Available: https://www.mdpi.com/2220-9964/7/3/93

[101] M. Bacco, P. Barsocchi, E. Ferro, A. Gotta, and M. Ruggeri, "The digitisation of agriculture: a survey of research activities on smart farming," *Array*, vol. 3-4, p. 100009, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590005619300098

[102] F. FAO *et al.*, "The future of food and agriculture–trends and challenges," *Annual Report*, vol. 296, pp. 1–180, 2017.

[103] E. Said Mohamed, A. Belal, S. Kotb Abd-Elmabod, M. A. El-Shirbeny, A. Gad, and M. B. Zahran, "Smart farming for improving agricultural management," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 24, no. 3, Part 2, pp. 971–981, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110982321000582

[104] A. Hammam and E. Mohamed, "Mapping soil salinity in the east nile delta using several methodological approaches of salinity assessment," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 23, no. 2, pp. 125–131, 2020.

[105] M. K. Abdel-Fattah, S. K. Abd-Elmabod, A. A. Aldosari, A. S. Elrys, and E. S. Mohamed, "Multivariate analysis for assessing irrigation water quality: A case study of the bahr mouise canal, eastern nile delta," *Water*, vol. 12, no. 9, p. 2537, 2020.

[106] E. Mohamed, A. Saleh, and A. Belal, "Sustainability indicators for agricultural land use based on gis spatial modeling in north of sinai-egypt," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 17, no. 1, pp. 1–15, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110982314000088

[107] A. El-Zeiny, A. El-Hefni, and M. Sowilem, "Geospatial techniques for environmental modeling of mosquito breeding habitats at suez canal zone, egypt," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 20, no. 2, pp. 283–293, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110982316301284

[108] T. U. Kumar and A. Periasamy, "Iot based smart farming (e-farm)'s," *International Journal of Recent Advances in Multidisciplinary Topics*, vol. 2, no. 4, pp. 85–87, 2021.

[109] A. James, A. Saji, A. Nair, and D. Joseph, "Cropsense–a smart agricultural system using iot," *Journal of Electronic Design Engineering*, vol. 5, no. 3, 2019.

[110] R. Kumar, R. Mishra, H. P. Gupta, and T. Dutta, "Smart sensing for agriculture: Applications, advancements, and challenges," *IEEE Consumer Electronics Magazine*, vol. 10, no. 4, pp. 51–56, 2021.

[111] M. Khanal, M. Hasan, N. Sterbentz, R. Johnson, and J. Weatherly, "Accuracy comparison of aerial lidar, mobile-terrestrial lidar, and uav photogrammetric capture data elevations over different terrain types," *Infrastructures*, vol. 5, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2412-3811/5/8/65

[112] N. Islam, M. M. Rashid, F. Pasandideh, B. Ray, S. Moore, and R. Kadel, "A review of applications and communication technologies for internet of things (iot) and unmanned aerial vehicle (uav) based sustainable smart farming," *Sustainability*, vol. 13, no. 4, p. 1821, 2021.

[113] D. Glaroudis, A. Iossifides, and P. Chatzimisios, "Survey, comparison and research challenges of iot application protocols for smart farming," *Computer Networks*, vol. 168, p. 107037, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128619306942

[114] M. P. Senyolo, T. B. Long, V. Blok, and O. Omta, "How the characteristics of innovations impact their adoption: An exploration of climate-smart agricultural innovations in south africa," *Journal of Cleaner Production*, vol. 172, pp. 3825–3840, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652617311885

[115] ——, "How the characteristics of innovations impact their adoption: An exploration of climate-smart agricultural innovations in south africa," *Journal of Cleaner Production*, vol. 172, pp. 3825–3840, 2018.

[116] J. Venkatesh, K. K. Ramasamy, M. Aruna, K. Praveen Kumar Rao, N. Sasikala, and K. Nasani, "eagri: Smart agriculture monitoring scheme using machine learning strategies," in *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 2022, pp. 1–9.

[117] A. Goulart, A. S. R. Pinto, A. Boava, and K. R. L. J. C. Branco, "Iot off-grid, data collection from a machine learning classification using uav," *Sensors*, vol. 22, no. 19, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/19/7241

[118] C. Vanitha, N. Archana, and R. Sowmiya, "Agriculture analysis using data mining and machine learning techniques," in *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)*, 2019, pp. 984–990.

[119] "Displaying a point-cloud." [Online]. Available: https://developer.apple.com/documentation/arkit/environmental_analysis/displaying_a_point_cloud_using_scene_depth

[120] "Capturing depth using the lidar camera." [Online]. Available: https://developer.apple.com/documentation/avfoundation/additional_data_capture/capturing_depth_using_the_lidar_camera

[121] H. Chen, S. Ye, A. Kurilovich, R. Bohush, and S. Ablameyko, "Video-based content recognition of bank cards with mobile devices," *Programming Computer Software*, vol. 46, no. 6, pp. 373–383, 2020. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edb&AN=147387906

[122] S. Hira, A. Matsumoto, K. Kihara, and S. Ohtsuka, "Hue rotation (hr) and hue blending (hb): Real-time image enhancement methods for digital component video signals to support red-green color-defective observers," *Journal of the Society for Information Display*, vol. 27, no. 7, pp. 409–426, 2019. [Online]. Available: https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edb&AN=137469340

[123] A. Gongal, M. Karkee, and S. Amatya, "Apple fruit size estimation using a 3d machine vision system," *Information Processing in Agriculture*, vol. 5, no. 4, pp. 498–503, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214317317302408

[124] G.-h. Chen, Q. Jin, X.-q. Zhang, and H.-m. Zhang, "Airborne lidar point cloud classification fusing spectral information," in *2021 3rd International Conference on Intelligent Control, Measurement and Signal Processing and Intelligent Oil Field (ICMSP)*. IEEE, 2021, pp. 177–184.

[125] Y. Ono, A. Tsuji, J. Abe, and H. Noguchi, "Robust detection of surface anomaly using lidar point cloud with intensity," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 1129–1136, 2020.

[126] M. Kulbacki, J. Segen, W. Knieć, R. Klempous, K. Kluwak, J. Nikodem, J. Kulbacka, and A. Serester, "Survey of drones for agriculture automation from planting to harvest," in *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*, 2018, pp. 000 353–000 358.

[127] T. M. Murtha, E. N. Broadbent, C. Golden, A. Scherer, W. Schroder, B. Wilkinson, and A. A. Zambrano, "Drone-mounted lidar survey of maya settlement and landscape," *Latin American Antiquity*, vol. 30, no. 3, p. 630–636, 2019.

[128] M. L. M. Rudge, S. R. Levick, R. E. Bartolo, and P. D. Erskine, "Modelling the diameter distribution of savanna trees with drone-based lidar," *Remote Sensing*, vol. 13, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/7/1266