Using Mobile-Based Augmented Reality and Object Detection for Real-Time Abalone Growth Monitoring

By

Thomas Napier

Dissertation submitted by Thomas Napier in partial fulfilment of the requirements for the Degree of Bachelor of Information Technology with Honours in the Discipline of Computer Science & Information Technology, College of Science & Engineering, Division of Tropical Environments & Societies at James Cook University.

Computer Science & Information Technology, College of Science & Engineering, Division of Tropical Environments & Societies, James Cook University.

21st June 2021

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Thomas Napier
21st June 2021

Statement of Access to Thesis

I, the undersigned, the author of this thesis, understand the James Cook University will make it available for use within the University Library and, by microfilm or other photographic means, allow access to users in other approved libraries. All users consulting this thesis will have to abide by the following statement:

"In consulting this thesis, I agree not to copy or closely paraphrase it in whole or in part without making proper written acknowledgement for any assistance which I have obtained from it."

Beyond this, I do not wish to place any restriction on access to this thesis.

. . . .

Acknowledgements

I would like to express my deep and sincere gratitude towards my primary research supervisor, Professor Ickjai Lee, for giving me the opportunity to do this research project, and for providing me with his invaluable wisdom and guidance throughout the process. His dynamism, vision, sincerity, and motivation have greatly inspired me. His expert teachings provided me with the skills and understanding of scientific methodology that were key to successfully carrying out this research, and he has given me the confidence to continue into an academia further. It was a great privilege and a true honour to work and study under his guidance, and I look forward to continuing to work with him in future.

I would also like to thank my family for their support and backing of an enquiring mind, without either of which I would not be where I am today. However, the completion of this project could not have been accomplished if it were not for the positive optimism and endless encouragement of my father, Darren Napier. Thank you for helping me stay the course through the highs and lows and inspiring me to persevere through to the very end.

Abstract

Abalone are becoming increasingly popular for human consumption which has resulted in significant production-related issues arising in attempt to meet market demand. Farming practices have remained mostly unchanged and traditionally are heavily reliant on human inspections and approximations which is time consuming resulting in high labour costs. Alternatively, machine vision can be used to automate growth monitoring, by providing fast, objective, and accurate results. These techniques have been successfully applied to other aquatic products such as fish and oysters, but suitable techniques are notably lacking in abalone aquaculture. This study introduces a new mobile-based method of counting and measuring abalone, that is both network and location independent. We propose, and substantiate through several aquaculture-centric experiments, that the instrument outperforms traditional counting and measuring techniques in both speed, and accuracy, and outline some limitations discovered when applying the system under certain situations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

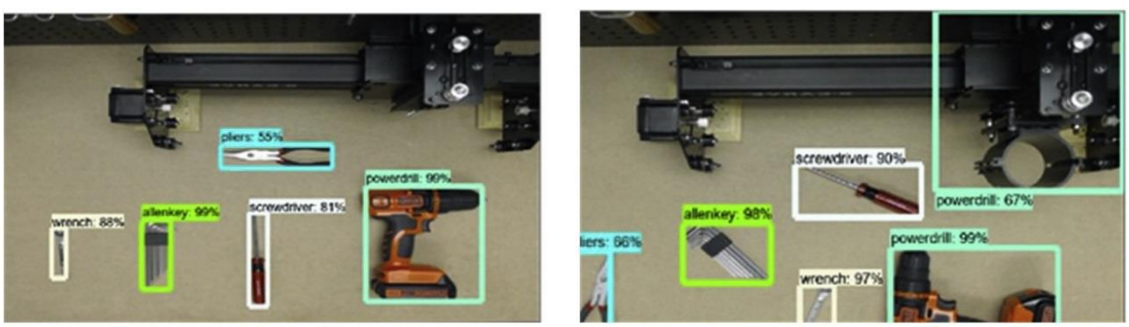
Aquaculture is the practice of farming aquatic plants and animals through the modification and manipulation of natural ecosystems. In recent times, aquaculture has seen a rapid increase in uptake, at a rate of approximately 4-11% annually due to escalating market demands based on increased human consumption combined with general decreases in wild commercial yield [1]. Aquaculture has thus been regarded as a reliable means of increasing food security for people worldwide [2].

One of the most highly sought after aquacultural creatures is the Abalone. Being one of the rarest and most expensive of any seafood, Abalone are a type of single-shelled marine snail that are found in very few parts of the world. As such, the commercial Abalone aquaculture sectors in Australia, particularly the Greenlip Abalone (*Haliotis laevigata*) have gained significant popularity in particular because of their high demand and export potential. Consequently, this uptake in interest has resulted in a proportional increase in price, and thus, abalone production and growth mechanism have become areas of high potential [3]. However, the methods of abalone farming have remained mostly unchanged and require comprehensive support from trained technical staff. Beginning from the abalone's nursery stages through to when they are market-ready, farmers are typically required to dedicate approximately 4 years of observation and labour which attests to the need of optimizing production management for consistently high yield rates.

Traditionally, farmers have used a combination of manual inspection and approximation techniques in favour of total population counting and measuring by hand [4, 6]. However, as the scale of production increases based on market demand, the inaccuracies inherent within these techniques become magnified. For abalone in the nursery stages, farmers select a small sample of juvenile abalone growth plates, count the amount, and then apply averaging techniques to approximate their stock count [25]. Similar counting methods are applied for the later stages of growth, but in addition, abalone are usually weighed, and their lengths are hand-measured which results in significant disturbance of the animals when they are removed from their substrate [5]. Moreover, these practices are highly time consuming, inaccurate, and susceptible to human error which further demonstrates the need for a superior system.

Thus, techniques to automate abalone measuring and counting processes have become areas of high potential. Much effort has already gone into applying vision-based techniques to other aquaculture areas such as fish, however, little research has been conducted in the domain of automating abalone farming [8]. Moreover, there are inherent issues with other techniques that prevent them being applicable to this research, and problematic to translate to an abalone context. In general, issues with current methods such as location dependence, expensive setup and maintenance costs, network reliance and non-real time systems causes them to be arguably ineffective and suboptimal for monitoring abalone growth. This is because the operating environment of onshore and offshore abalone farms can be unpredictable in terms of environmental conditions, weather, and network coverage [25]. Thus, real-time results, freedom of location and non-obligatory network connectivity would allow farmers to enhance their current monitoring practices without the burden of additional running expenses from complex server-client based setups.

Rapid advances in mobile technology by way of computational efficiency and power, combined with their widespread availability and convenience in modern times has made it an ideal candidate for addressing the limitations of previous approaches. Leveraging this increase in power, more opportunities to utilise recent technology paradigms including Augmented Reality (AR) and Deep Learning (DL) object recognition have become possible to run effectively on mobile devices. These techniques can provide measuring and counting functionality and by nature, will overcome some issues with previous approaches. Moreover, there have been several instances in research where these techniques have been utilized effectively in other domains for a variety of real-time inspection tasks, ranging from tourism to on-site training, and medicine, but have yet to be widely adopted in aquaculture [4, 7, 10, 11].



(a) (b)

Figure 1 – An Example of an AR Instructional System Integrated with Object Detection [11]

1.2 Problem Statement and Research Questions

Thus, to develop an improved abalone measuring and counting system, there is a core research question that must be addressed by the study to achieve the desired result. The answer to this question should result in a mobile-based system that is capable of accurately and efficiently counting and measuring abalone in real-time, where the system is both location-independent and network-independent.

Therefore, the primary question below encapsulates the core research concepts that are being investigated and it will ultimately be used to evaluate the effectiveness of the proposed instrument addressing the deficiencies identified in current literature.

How can an integration of object detection and augmented reality techniques be used for real-time, network independent abalone counting and size measurement using a mobile device?

This focus question was then broken into several sub-questions. The first of these questions will be used to provide a guideline for the related techniques and methods for measuring abalone and other aquatic life, with a particular emphasis on real-time. Using this question as a guide should ideally result in a mobile-based instrument capable of capturing object measurement data in real-time. An ideal implementation in the context of abalone measuring would mean enabling users to capture variable sizes of objects efficiently, independent of location, and network connectivity.

1 – *What object measuring techniques can measure Abalone accurately in real-time?*

The second research question then relates to the other form of real-time data capturing to be used, object counting. Successfully answering this question should result in a chosen mobile-based DL technique capable of recognising multiple abalone within a single image.

2 – *Which object detection techniques can accurately recognise and count multiple Abalone in real-time?*

Thirdly, this question will then focus on the integration and congregation of both previous techniques into one unified system. Success in this sense will mean that both object measuring and counting can occur in real-time using a mobile device.

3 – *How can object measuring and recognition be combined to provide a real-time user experience for Abalone monitoring on a mobile device?*

1.3 Relevance and Significance

The current techniques for counting and measuring abalone use a manual or approximation method to track population numbers and growth rates [4, 6]. The inherent issues with using these techniques can be dramatically improved in terms of accuracy, speed, and reliability using object recognition and AR on a mobile device. A well-designed system to count and measure the amount of abalone in commercial farming, in real time, would decrease the number of human mistakes and increase the efficiency, and accuracy, of stock taking. As a result, farmers would be to re-distribute human resources to other areas more effectively and reallocate funding to further improve their abalone growth monitoring process.

In addition, the proposed features of location and network independency will extend the range of uses to include not only onshore farms, but also offshore-based abalone farms. Daily inspections of abalone can thus be conducted more accurately, and relevant population data can be made more widely available. Such data would enable farmers to track their population and analyse growth data to better detect the presence of disease, high morbidity, or unusual patterns, thereby increasing the effectiveness of early preventative measures. Through the accompaniment of a mobile-based automated system with the existing monitoring procedures, more accurate and efficient stock taking will be made possible, and thus, farming procedures can be optimised to maximise profits.

However, there is a notable gap in the current literature surrounding the automation of counting and measuring abalone, yet the methods which produce high magnitudes of error are currently in use, which demonstrates the need for more research in this area. In addition, not enough research has been done on real-time measuring and object detection on mobile devices that is non-network reliant. Traditionally, in aquaculture and non-aquacultural areas, the use of complex setups and server or cloud-based for visual inspection tasks is used. Often these techniques require the use of networks and hard-wired computers to transfer images to a hosted system, which create unnecessary overhead and are often complex to setup and maintain. Chapter 2 of this thesis will further highlight several of the main approaches observed in the literature and outline their inherent issues when applied to the problem domain of abalone.

1.4 Research Aims

The aims of this project are multi fold. The first aim is to determine the most suitable vision-based techniques for counting abalone on mobile in real time without the need for a network connection. The chosen techniques will need to be able to detect abalone throughout its various life stages; be efficient and portable such that it can be used in real time on a mobile device and be fully self-contained to where it requires no communication with any external networks so that it can be used anywhere. It is further aimed that the most suitable augmented reality method for measuring abalone will be determined based on a similar set of metrics, with the added goal of being performant in a variety of environmental and lightning conditions.

Experiments will then be conducted to identify if the proposed solution improves the accuracy and speed of both the counting and measuring of abalone when compared to existing techniques. As a secondary aim, this research should establish, through investigation and interpretation of the experimental results, a foundational baseline from which further research into more robust and improved techniques can be conducted.

Thus, it is hypothesised that a selective integration of AR combined with object recognition will result in a real-time, network independent Android mobile application suitable for counting and measuring abalone.

1.5 Thesis Structure

The remainder of the thesis is structured as follows: Chapter 2 covers the relevant literature for the key concepts to be used in this research project and reviews the current approaches used to count and measure objects both in and outside of aquaculture. Chapter 3 details the framework design used and broadly discusses technical concepts. Chapter 4 describes how and why the abalone counting and measuring tool was implemented the way it was, from a development standpoint. Chapter 5 discusses the experimental methodologies used for testing the validity of counting both mature and juvenile abalone. Chapter 6 covers the experiments used for testing the measuring tool on several objects. Finally, Chapter 7 summarises the results and discusses their implications as well as the limitations of the proposed solution and how it could be improved on in future research.

Chapter 2

Related Works

2.1 Abalone Farming

As it has been briefly discussed already, the current state of abalone farming, and by large, the aquacultural industry have been limited to manual detection, counting and grading systems. For example, in most fisheries, where these manual forms of stock taking are highly prevalent, such tasks can take multiple people several hours to individually screen tens of thousands of fishes, with no definite assurance that their numbers are accurate or true [30]. In the case of abalone, manual inspection for counting and measuring is typically performed by hand by trained staff at both the juvenile and mature stages. Here, often any mode of disturbance decreases growth rates or can even cause the abalone to die if enough damage to the soft underlying body is done [5].

At the nursery stages, the juvenile abalone are even more fragile and susceptible to death if handled incorrectly. For farmers to achieve high nursery plate yield rates, trained staff must follow a strict procedure for any out-of-water counting, measuring or health inspections. When they are at this very delicate stage, they cannot be exposed to air for more than 1 minute at any one time, and only then, when air temperatures are between 10-25°C [25]. Such conditions require workers to be careful and precise, which can significantly increase the time taken to count and measure abalone stock. As abalone only grow, on average, only two to three centimetres annually, it is imperative to see a return on investment

[3]. Thus, ensuring their survival and constant growth at their mature stages is also highly important.

Currently, matured grow-out and weaner stage abalone are counted and weighed also using manual methods [4, 5, 6]. At these stages, it is less likely that they are as delicate as they were as juveniles, however, monitoring stock growth statistics are still highly important for identifying rapid rises and drops in population. In addition, close monitoring of the population can also assist farms in determining optimal selling points and can help assess the performance of different feeding strategies. However, due to the current methods of counting and measuring stock, there are still issues with accuracy, scalability and variability which only become more apparent as operation sizes expand, and stock volume increases.

2.2 Object Detection and Augmented Reality

2.2.1 Deep Learning and Object Detection

DL is a function of Artificial Intelligence (AI) which attempts to mimic the human brain, whereby models are shown large amounts of training data (such as images). Through repeated exposure, DL algorithms can learn patterns and extracting features to achieve a desired output. These goals can range from detecting objects to recognizing speech patterns. Object recognition, for example, is fundamentally a visual detection problem that is made possible using DL. The term object recognition is a generalization that is used to describe the marriage of both image classification and object localization [22].

Image classification relates to a process of prediction, whereby a deep neural network is trained to recognize and extract the characteristics of an object within a labelled example image using repeated exposure, so that can be used to assign a label to a previously unseen example. Object localization is the process used to locate the presence of objects within an image and indicate their position using bounding boxes. Historically, these types of mechanisms are computationally demanding and, in the past, have required powerful hardware to be able to run them effectively which limited the algorithms to pre-configured or cloud-based systems [26]. However, the feasibility of running DL models using mobile phones has increased multifold. This, in turn, has led to an advance in efficient mobile based DL frameworks such as Google's TensorFlow Lite and Facebook's PyTorch. These systems have been designed to harness the limited power of mobile devices to allow for the real-time extraction and identification of objects within an image [32, 33].

At the cost of some general loss in accuracy due to limitations of mobile-based hardware, the benefits of on-device processing are multi-faceted when compared to traditional systems. The most compelling reason for using on-board resources is that overhead from server-client communication is greatly reduced which conjointly offsets or negates any hosting costs and connection reliability issues [9]. Furthermore, tasks which make use of local resources typically result in faster response times and increased information security because it is never sending or receiving information from a network. As such, applications that use this processing technique eliminate the need for network connectivity, meaning that they are ideal for DL tasks which take place in regions with unpredictable network connections, such as offshore abalone farms.

2.2.2 Augmented Reality

Fundamentally, AR is a virtual experience whereby real-time, interactable objects such as three-dimensional (3D) models, text and images are superimposed onto the physical world and are made viewable through a digital medium, such as a smartphone or tablet. Such technology can be useful for projecting virtual content and manipulating it within a 3D environment, in real-time.

Objects that in an AR scene are strictly digital, and as such, the boundaries of the physical reality can be extended through technology in such a way that it perceptually enriches a user's actual environment. To fully understand AR, it is important to distinguish where the boundaries between physical reality ends, augmentation begins, and where the line is drawn for virtual reality (VR). To define such boundaries, we use the reality-virtuality continuum, as seen in Figure 2 below.

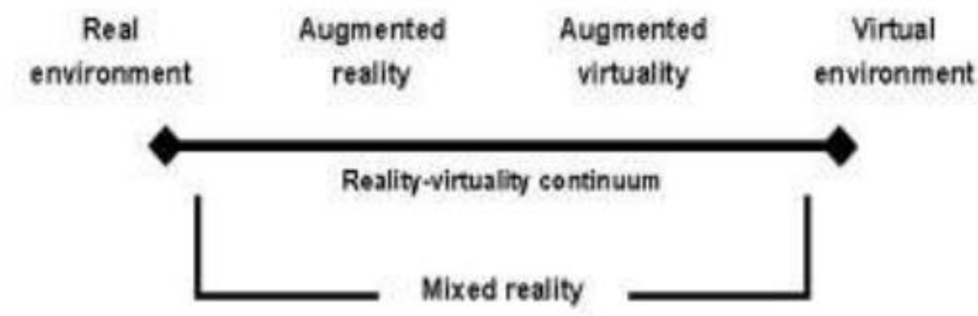


Figure 2 – The reality-virtuality continuum [35]

Full VR and AR greatly differ from each other. Rather than enhancing the user's environment with digital objects as with AR, VR completely immerses the user in a fully virtual environment without any view of the real world [36]. This distinction, though slight, greatly defines the scope and limitations of each concept. Thus, AR is not restricted to the boundaries of wearable technology such as head mounted displays, which take up the entirety of the sense of sight [35].

With the ubiquity of mobile devices in modern society, AR has seen an increase in usage in mobile applications. We need only look back to 2016 with the recent example of Pokémon GO app produced by Niantic, which quickly became the top mobile game in the US at the time; to see the potential that AR-enabled applications can achieve [24]. However, Pokémon GO is only one case which illustrates the potential of AR. The technology itself has already been implemented in a variety of industries ranging from healthcare to tourism [24, 27, 28, 37].

AR and DL have mainly been applied to areas outside of aquaculture, namely in areas of medicine, on-site teaching, and education, with relatively high success. One such paper where these techniques have been successfully used is in a 2018 study that focuses on using AR to create visual aids through display for early childhood training. Here, they use CNN based image classification to detect markers, which are the letters of the alphabet, and use AR to motion track relevant objects to the markers [27]. This approach has a strong focus on using marker-based AR, with an integration of deep learning image classification as well. However, the design is limited to only very visually distinct alphabetical letters, it requires camera calibration for motion tracking, and it is not made for mobile devices.

For approaches which employ the use a mobile device, there have been medical-based tools that use image classification and AR to detect and measure medical bedsores such as a study proposed in 2020 [7]. To measure these physical ailments, researchers chose to use a web hosted server which classifies the bedsore image using PyTorch and then they use OpenCV to measure bedsore (marked out beforehand) which is then shown graphically with Easy AR once processed through the measurement tool [28]. This article is useful because it combines both AR and DL together, for the detection and measurement of bedsores within an image. However, the authors make use of web services to achieve this which creates unnecessary overhead, is overly dependent on user interaction for the initial measurement line drawing, and the solution is not in real-time.

2.3 Object Counting and Measuring in Aquaculture

Previously proposed methods of counting and measuring abalone typically involved data capturing either manually or using commercially available optical detection systems and using approximation techniques to estimate farm populations. More recently however, there has been an increase in popularity in applying machine vision systems to the domain of aquaculture. The four main groups of approaches for counting and measuring aquatic products observed throughout the literature are sensor-based methods, acoustic based methods, video analysis methods and image processing methods [12]. These techniques have all demonstrated high potential across a multitude of tasks for other aquatic products such as fish, oysters, scallop, and prawns but is notably missing in abalone aquaculture. Figure 3 below displays the fundamental elements typically found in an aquacultural machine vision system (MVS).

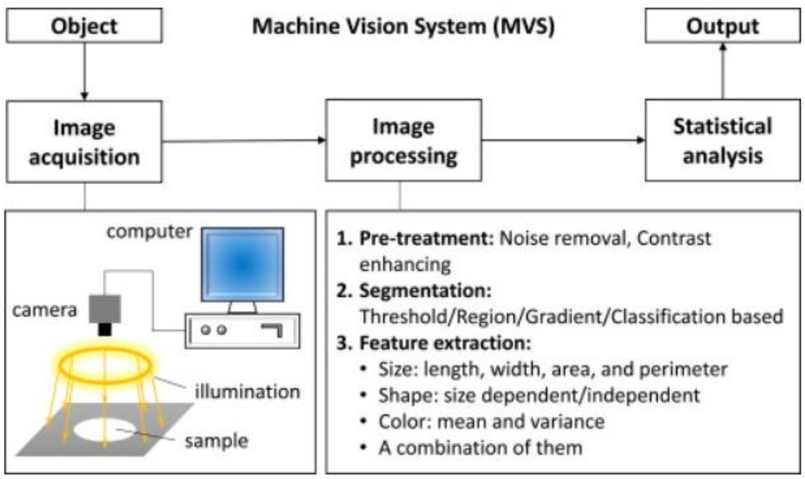


Figure 3 – Elements of a Typical Machine Vision System (MVS) [4]

2.3.1 Sensor-based Methods

Sensor-based methods are approaches used by researchers that typically employ the use of electronic and mechanical devices including optical sensors, infra-red and electrode resistivity for counting aquatic life. For abalone, approaches based on these types of methods have been previously investigated, such as in a study conducted in 2015 for the mechanical grading and weight estimation of market-ready samples [31]. Able to achieve a weight estimation accuracy to within 8g, the proposed capturing system here, as pictured in Figure 4, uses a complex, rail-guided, LED infrared backlit data collection method, where images are taken and then transmitted over an Ethernet network to a nearby storage device for analysis. Such a system has several drawbacks, the first of which is that it is limited to adult-sized abalone only, as juveniles would be too fragile for such a system. Secondly, an implementation like this has multiple points of failure due to its complex setup and reliance on network connectivity to the attached computer. This also indicates that special care and expense is required for system maintenance.

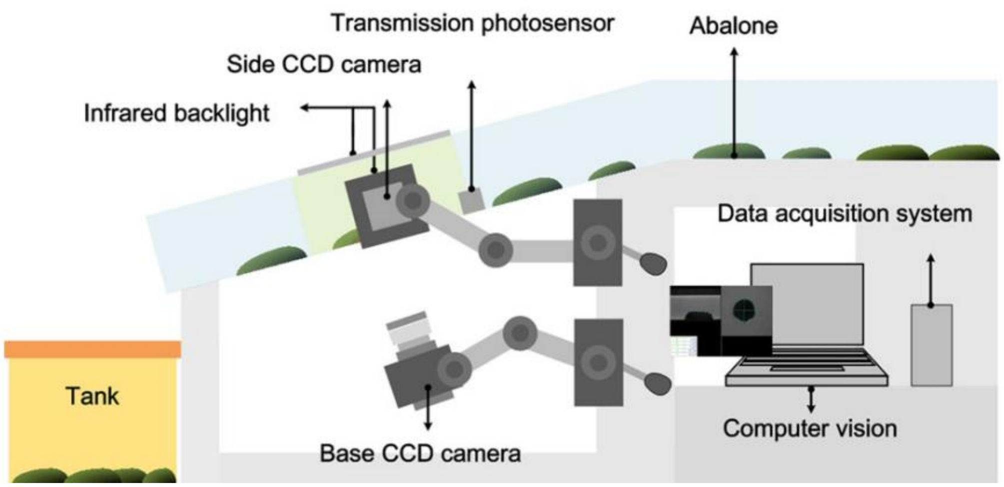


Figure 4 – Features of a sensor-based abalone measurement method as detailed in [31]

For an anatomically similar creature to abalone such as the oyster, sensor-based systems have also been successfully applied such as in study conducted where the method proposed could sort them into three size categories with an accuracy of 88% in quick succession using a fixed camera and conveyer-based system [13]. A similar method named "The CatchMeter" was proposed in 2006, which rather than being applied within a controlled environment, was put onboard a vessel for automatic catch logging, and thus subject to oceanic conditions. In their approach, they too used a mechanical conveyer-camera setup interfaced to a nearby computer for the automatic sorting of fish which achieved a 99.8% sorting reliability for seven species of fish and standard deviation of measuring equal to 1.2mm [14]. While this solution can perform real-time measuring and counting, it has the limitations of a complex setup including lightbox and feeder components to ensure that the fish are in the correct orientation. In addition, it is also vessel-bound meaning that it is limited in the scope of application.

In another sensor-based approach, researchers used an infra-red counter-based mesh turbine system based within a silver perch aquaculture tank [15]. The system would count and measure fish that pass through the unit and break the infra-red beams. One of the major drawbacks of this approach, however, is that it would detect multiple fish that passed through the mesh at once, as a single instance. The clear limitations this, and by large, sensor-based approaches have, is that they are often immobile and expensive, requiring specialist-made equipment and constant network connectivity to a nearby computer to function. Such approaches not only require the solution to be pre-arranged to function, but it also may have performance issues due to bottlenecks from back-and-forth communication between the system and the client.

2.3.2 Acoustic-based Methods

Acoustic-based methods are approaches that use sound waves to overcome issues with underwater image capturing such as in murky conditions and low light levels. Included within the acoustic based methods are techniques that use imaging sonar and hydroacoustic pulses to count and track underwater life. For approaches which use sonar there are studies where the researchers have used a multi-beam mobile sailing robot to automatically estimate fish density in open waters [16]. Here, the tool was designed with cost-effectiveness in mind, thus, a cheap sonar module was mounted onto a sailing robot and data was archived locally to a maximum of 22 Gigabytes (GB), with the option of wireless transmission. While this method was efficient at counting and measuring aquatic life in areas of uncertain water conditions with a measurement error of less than 8%, it still has some limiting factors. The solution is still relatively expensive to produce at around 2500 Euros per unit, and the images captured are not available in real-time as it must be either stored locally and retrieved manually or transferred wirelessly to a computer within a range of 1 kilometre, and then analysed to determine fish counts.

Acoustical techniques have also been used for monitoring fish density, behaviour, and growth rate within fish tanks such as in a study conducted in 2005 [38]. Using remotely-activatable acoustical emitters and receivers to monitor fish within a cylindrical tank, researchers were able to harmlessly record the reverberation time series in the tank with swimming fish to estimate their density and growth rates, without the need for human intervention. However, the system was limited in that measurements are only taken every 10 minutes at a time, thus meaning that results are not in real-time and sudden changes in population or behaviour would not be discovered until results were collated and analysed. The initial setup of this system under different conditions must also first be verified to ensure that results are reliable prior to use, meaning that the system lacks efficient adaptability.

Another method is the usage of hydroacoustic estimation, such as a study done on non-invasive fish counting methods via echo sounding [12, 16]. However, such solutions also are generally expensive and are required to be affixed to a floatation vessel to operate which limits their overall usefulness. In addition, they are unable to recognize small and dense, overlapping objects because the acoustic systems rely on the echo sound pulse reflected from objects with different densities, and are not precise enough to distinguish small schools of fish.

2.3.3 Video-based Analysis Methods

For video-based analysis methods, typically multiple cameras, and other sensors are employed for data capturing, and are often restricted to controlled environments. These types of systems are good for establishing continuity between image frames to establish aquatic creature trajectory and behavioural analysis. In two studies, video recording sequences are captured from aquaculture tanks using multi-camera systems for fish behaviour analysis [18, 19]. These methods are susceptible to similar limitations as those identified in sensor-based methods. However, they also possess the added factors in that they generally do not include the integration of AR, unlike many similar methods seen outside of aquaculture. Furthermore, they are often strictly limited to client-based systems which demand constant network connectivity.

Video-based techniques have also been applied for the real-time detection and counting fish in low quality underwater sequences. For a given underwater video, the system proposed in [39] can achieve an accuracy of 85% using a combination of blob shapes and histograms for tracking fish within a sequence. However, the method suffers from continuity issues relating to fish tracking between video frames, as it is unable to reliably distinguish unique individuals from low frame rate recordings. In addition, the system is also restricted to underwater sequences that feature low creature density.

Another example of this can be observed in a study done in 2020 which proposes a network-enabled crab detection system implemented on a configured microcomputer aboard a water floatation device [17]. Furthermore, these types of systems are generally not real-time, and videos are recorded for transmission to a web-based service for later analysis. This also means that video footage is required to be retained, and thus, server hosting costs and storage considerations become additional limiting factors to these types of approaches.

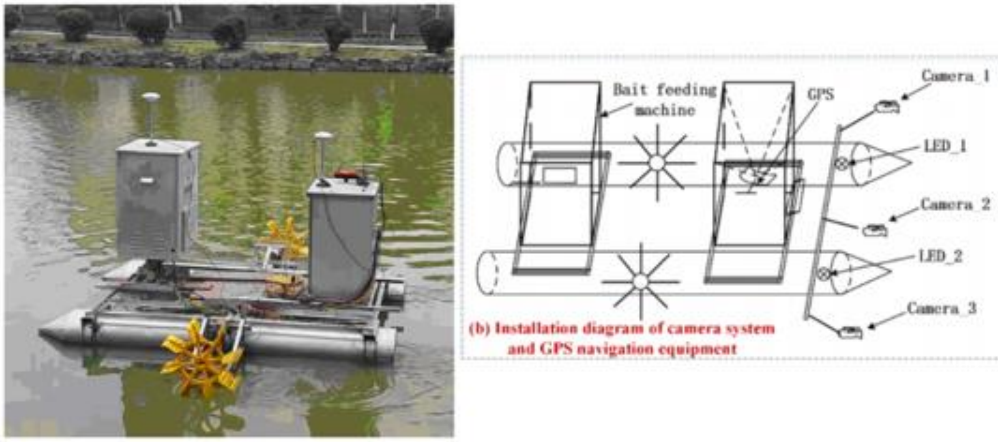


Figure 5 – Crab video-analysis based detection method which employs the use of multiple cameras [17]

2.3.4 Embedded Image Processing Methods

Finally, for embedded image processing, typically these methods are non-invasive, location-independent, and highly efficient because they have power and computation restrictions. For one such recent method applied to the task of scallop detection, researchers used image processing methods embedded in an autonomous underwater vehicle (AUV) system [20]. The system they used was equipped with sonar obstacle avoidance, an antenna, a propulsion system, and a downwards-facing camera which sequentially captured images for later counting analysis using a convolutional neural network based on the YOLOv2 architecture. This method has the reoccurring issue of the need to offset processing through data transmission over a network, meaning that it is not in real-time.

Further methods such as one proposed in 2018 [40], focus on using low-cost image processing methods embedded on a Raspberry Bi 2 for the counting of fish with an average accuracy equal to 96.64%. Their approach is both real-time, low cost and easily portable. However, in a real-life environment, the method requires a specialised setup where fish are fed through a two-tank interconnected system which force fish to travel from one tank, past the counting system into a collection tank. Alternatively, a multi-arrangement including multiple cameras would need to be disbursed strategically over a tank to ensure all areas are covered before their counting result is sent to a centralised system via a wired or wireless connection.

Another observation of embedded image processing from the literature can be seen in a study on lobster grading using convolutional neural networks [23]. In their methodology, they can achieve positive results using a mobile-based approach when attempting the task of determining a lobster's weight, carapace size and overall grade. Here, the researchers employ the use of a client-server model where photos taken with the application are sent to a server for analysis before results are returned to the user. This further confirms that this study, and similar embedded systems are typically network dependent and not real-time.

2.4 Summary of Literature Gap

The deficiencies in the literature are mainly that the areas of DL and AR have not been applied in the real-time measuring and counting of abalone, leaving a considerable gap. The most prevalent limitations of related approaches are that they are network-reliant and/or not real-time, requiring constant connectivity for the system to work. For a system such as the previously mentioned mechanical method proposed in 2006, they require a direct interface from a computer to the vision system via an Ethernet cable link [14]. Such a solution not only requires the solution to be pre-arranged to function like many other approaches [13, 14, 15, 16], but it also creates unnecessary overhead from back-and-forth communication between the system and the client.

Other approaches that incorporate the use of video-analysis suffer from network-reliance as the footage is transferred from the capture system to the web-hosted tool for processing, resulting in latency and susceptibility to poor signal conditions in remote areas [18, 19]. In addition, these systems also suffer from complex, often immovable setups which are more susceptible to error due to the increase in points of failure. Such approaches would not be as suitable for abalone, as the smallest changes in population or growth patterns could be potentially costly for farmers due to the slow growing rates. As summarised from reviewing the advantages and disadvantages of the approaches as per Table 1 below, it can be deduced from the identified thematic groups that none of the existing aquacultural approaches are entirely suitable for the given task. In addition, few of the review approaches are transferrable for counting and measuring abalone in real-time, where the approach uses a mobile device, and is both network and location independent.

Approach	**Strengths**	**Weaknesses**
Sensor-based	Fast response, capable of producing real-time results, and easy to understand	Often requires complex setups and special equipment, generally required to be networked for data transmission and analysis
Acoustic-based	Non-invasive, good for capturing underwater images in unclear conditions	Difficulties detecting smaller objects in high densities, sometimes they need to be affixed to vessels and results are generally not in real-time
Video-Analysis based	Good for tracking general behaviours and establishing trajectories of individual creatures in low-density situations	Issues regarding video storage and transference over a network, often unable to establish reliable continuity between

		frames for accurate counting
Embedded Image Processing	Often portable and non-invasive, typically highly power efficient due to hardware restrictions	Limited by hardware for continuous counting, sometimes requires network connectivity to transfer locally stored data

Table 1 – Strengths and Weakness of Current Aquaculture Counting and Measuring Methods

Thus, the research conducted in this thesis aims to determine, through experimental research, the effectiveness of mobile-based AR and object detection for counting and measuring abalone in both their juvenile and grow out/weaner stages. Through this, the system will overcome issues of immobility from identified methods in areas such as video-based analysis and sensor-based methods which often require special components and equipment configurations for operation. The mobile aspect of the approach will be more desirable for reasons of convenience and speed, where real-time results can be made available as opposed to being stored for ensuing analysis. Moreover, the proposed instrument will be operable in network independent conditions unlike many other approaches, such that it can be used in edge cases where connections are unreliable or slow.

Chapter 3

Framework of Mobile Abalone Measuring and Detection

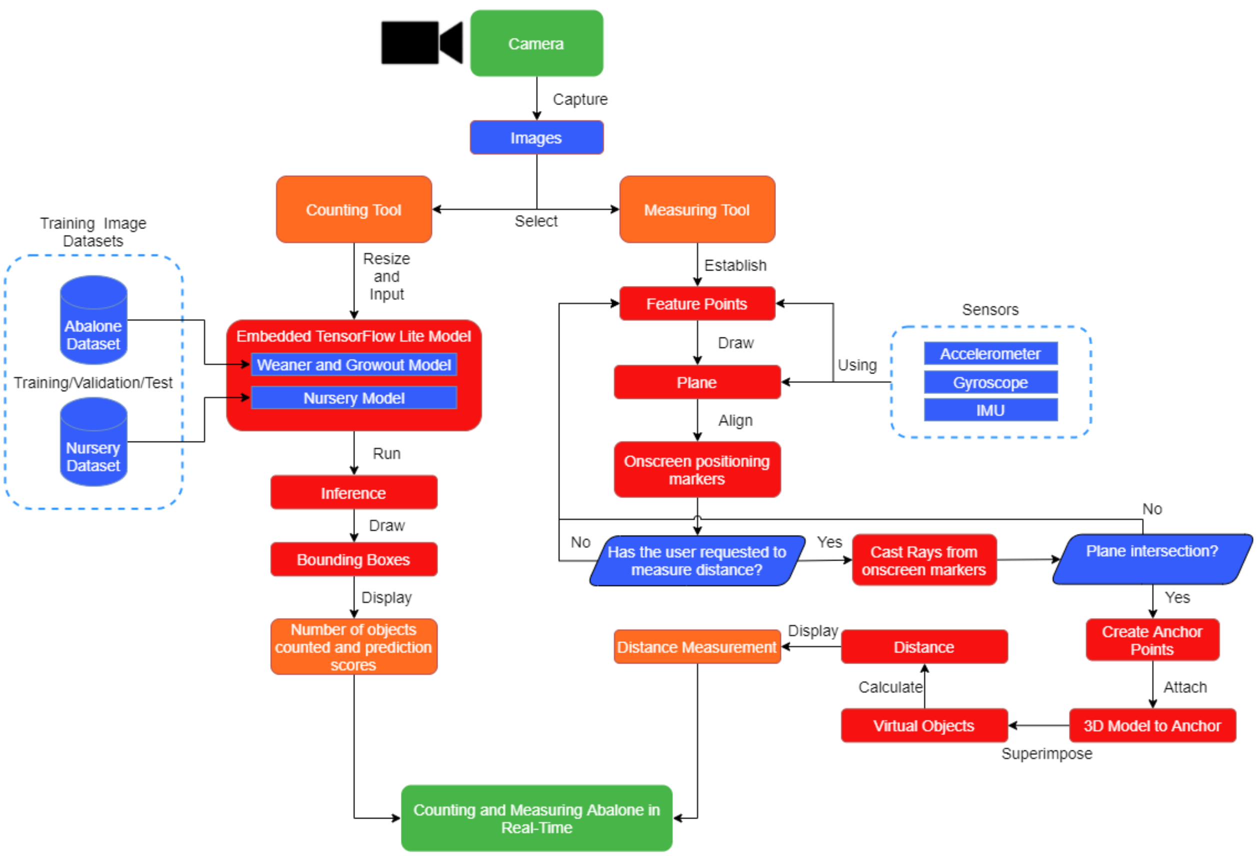


Figure 6 – Conceptual Framework for Counting and Measuring Abalone

The system designed for counting and measuring abalone in real-time, without a network connection has been defined as beginning with the on-device camera, as image capturing is at the core of the instrument. With image capturing handled, it is up to the decision of the user and the nature of the task they are trying to complete. In the context of this project, the decision must occur at the start of usage, as simultaneous operation was outside of the project time and resource scope. As outlined in Figure 6, the proposed framework consists of two main conceptual branches contained within a single android application, with no external connections required during its execution or operation.

For counting, images are taken from the camera, and morphed into a usable 416x416 size ready for input into the embedded, TensorFlow Lite model pre-train on custom data using the process of transfer learning. Two TensorFlow Lite models were trained on two sets of training data, one for nursery abalone and the other for weaner/grow-out stage abalone. The output of these models – once trained – was a self-contained .tflite file and an associated class labels text file, both of which were transferred to the local storage on the mobile device and embedded such that it could be used by the mobile application. Images captured by the system would then be passed through these models and the output results, including the prediction scores, object locational coordinates within the image and number of objects would be returned. From here, the bounding box coordinates are stretched to match the original image size and are drawn on-screen to indicate the position of the object along with the accuracy from prediction scores, class of object, and overall number of objects within the image.

In the second component, for the task of measuring objects, Google's ARCore was used, and it requires several steps of user interaction in conjunction with data made available from on-device sensors including the camera, inertial measurement unit (IMU), accelerometer and gyroscope. Initially a plane must be established within the operating user's environment which is achieved by using the sensory data available to locate and track feature points at areas of visual distinction. Using clusters of feature points, ARCore can then overlay and establish the boundaries of a two-dimensional (2D) plane which enable for the accurate positioning of anchor points and other digital objects relative to real surfaces, but within the augmented environment. With a plane established, the object to be measured must be situated close to it, and using the on-screen positional markers, users simply need to align the dots with either side of the object that they are trying to measure.

Once aligned, a user can request to measure the distance using a single button click. ARCore then takes the pixel axis coordinates from the 2D screen and casts a straight line – or ray – from both on-screen marker points and sends them into the world space within the camera's view. If either of these rays intersect the established plane, an anchor point is created at the point of intersection. From this, we can attach viewable 3D models and render them to the screen as visual indicators of where the anchor points were placed. It is then a matter of calculating the distance between the two established anchor points using the Euclidean Distance formula, and then relaying that to the user.

Both component functionalities allow for the user of the application to rapidly measure singular abalone or larger objects at once or count multiple abalone in real time. Such a design means that users are not forced to use additional devices or change the main screen of the application as both tasks can be easily swapped to. The ease of resetting the application, and additional advantage of real-time enabled features supports the operator in making one or more measurements without the need to take extensive time to reset the observation. In both cases, the focal objects do not need to be manually handled except to lie the creature flat, and it allows operators to easily collect objective data for growth inspections and population monitoring.

Chapter 4

Implementation

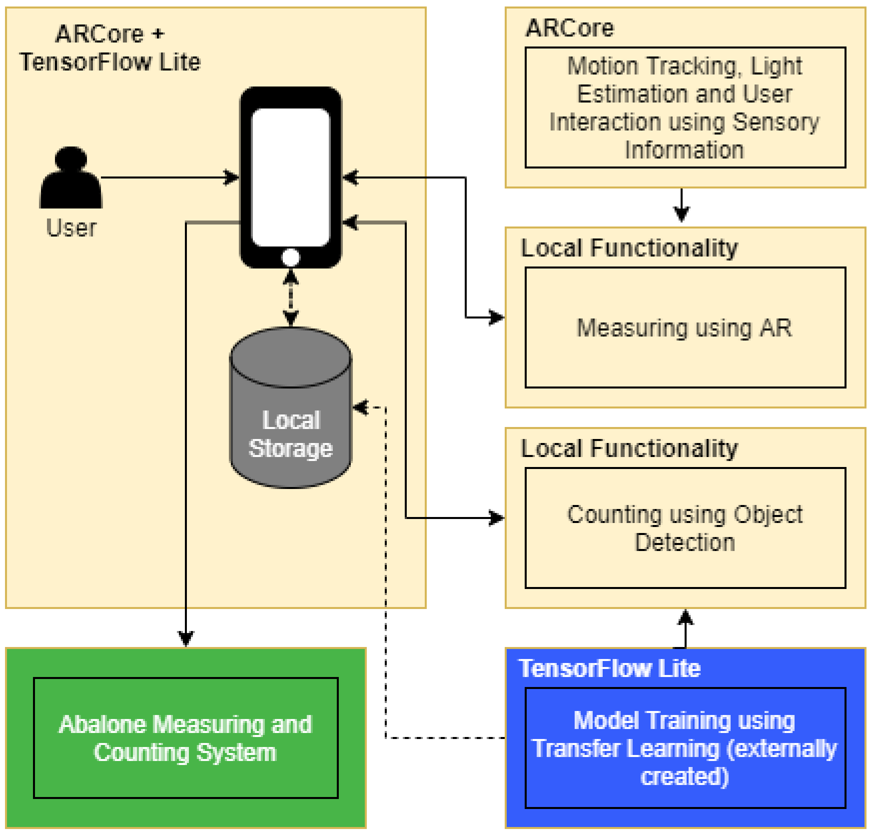


Figure 7 – Overall Architecture for the AR Object Detection-based App Implementation

The framework model described in the previous section has been implemented as an Android AR and object detection-based application, the architecture of which is visualised above in Figure 7. The figure shows that ARCore and TensorFlow Lite are the chosen technologies that will be combined and constructed using local storage, sensory information and externally created TensorFlow Lite models re-trained using transfer learning. TensorFlow Lite is shown to provide the local counting functionality using object detection, and likewise, ARCore is shown to facilitate measuring using AR. The layout of the diagram shows how the user interacts with the system, and how each component with the architecture relates and ultimately how abalone measuring, and counting is achieved.

4.1 Augmented Reality on Mobile

4.1.1 Augmented Reality Mobile Library

The primary evaluation step in the AR toolkit selection process is to define the differences between the two primary categories of AR applications, which are either marker-based or location-based. The main distinction between the two types is tied on how positional information is gathered from the virtual scene, and how it is subsequently used to superimpose digital objects. Marker-based AR is when the system places digital markers into a scene at visually distinct points such as object edges, areas of contrasting colours, or on real objects using in-built recognition techniques. Markers act as virtual anchors, allowing for digital objects to be positioned and tracked relative to them. This mechanism allows a system to significantly simplify and streamline the positioning of virtual objects by ‘pinning’ them to the real environment.

Furthermore, this technique reduces the computational complexity required during position and orientation calculations, making it ideal for mobile devices due to their limited available resources [41]. Comparatively, location-based AR allows users to position objects anywhere within the range of the camera view by using a combination of geographical and smartphone sensory information [43]. Importantly, this means that location-based AR is well suited for applications that use predefined points of interest (POIs) or geographical information to display vicinity relevant content. For example, when walking around a major city with its numerous tourist attractions, a location-based mobile app will be able to detect these areas and provide insightful information [42].

4.1.2 Evaluation

For the purposes of an object detection and length measurement AR application however, it is likely that measurements will need to be taken dynamically, that is, users of the application should not be required to go to a specific geographic location to operate it as with location-based AR. Thus, the selection of AR SDKs will be reduced to only those which offer marker-based functionality. In addition, a set of criteria was developed to enable the suitability of each SDK to be related to the necessary functions of the app. As seen below in Table 2, a cross comparison of the top six AR SDKs was developed from the respective developer documentation and examined for their supported platforms, primary differentiating features, and pricing options.

SDK	Supported Platforms	Key Features	Pricing
Vuforia	• iOS • Android • Universal Windows Platform • Unity	• Cylinder tracking capability • Customisable markers	• Basic version ($42/mo) • Basic Cloud version ($99/mo)
ARToolKitX	• iOS • Android • Linux • Windows • Mac OS • Smart Glasses	• Smart glasses integration • Automatic camera calibration	Free (Open Source)
Google ARCore	• Android 8.1 or later	• Environmental and depth	Free

	• iOS 9.0 or later • Unity • Unreal	understanding for motion tracking • Light estimation	
Apple ARKit	• iPhone 6s and higher • iPad Pro • iPad Air (3^{rd} gen) • iPad (5^{th} gen+) • iPad mini (5^{th} gen+) • iPod touch (7^{th} gen)	• People occlusion • Multiple face tracking	Free
MAXST	• Android • iOS • Mac OS • Windows • Unity	• QR and barcode reader • Cloud recognizer	• Free trial • One-time fee ($699) • Subscription ($50/mo)
Wikitude	• Android • iOS • Windows • Smart Glasses	• Geo AR • Multiple trackers • Multiple image targets	• One-time fee (€2490) • Subscription (€2990/yr)

Table 2 – Comparison of the Top AR SDKs. [45, 46, 47, 48, 49, 50].

From Table 2, the primary attributes of the supported platforms, differentiating features and pricing were evaluated for each SDK. The available platforms that each SDK supports is vital to the decision-making process because it directly relates to the flexibility of devices which the development of the app will be constrained to. As seen from the table, the majority the SDKs are supported on most platforms, apart from ARKit which is only available for modern Apple products. Interestingly, other options like MAXST and ARToolKitX also have the added advantage of being available on computers and other smart wearable devices. However, with the focus being from a mobile device standpoint, all apart from ARKit are equally adaptable, being available both on iOS and Android.

Key differentiating features is another criterion which further aids in the selective process. As the primary functions of the proposed AR app will be object measuring and detection, a marker-based SDK will be used since the function of the app will likely not rely on or use geographical information. Several key features from both Wikitude and Vuforia are quite niche and superfluous for the task at hand, such as Vuforia's cylindrical tracking feature. These types of functions would be much better suited for enterprise or commercial applications, which is indicative through the pricing options targeted towards those types of customers. Other key features such as ARCore's environmental understanding and ARToolKitX's automatic camera calibration, however, almost certainly align with the proposed AR app and will assist with the development process.

Pricing was the final key consideration, and due to the nature of the project, a free and readily available SDK was necessary, as it not only implies that there will be more documentation and previous examples on it, but in general there is more knowledge surrounding it because more people have access to it, meaning development times can be drastically reduced.

Thus, from the cross examination of the top SDKs, ARCore and ARToolKitX consistently meet the requirements set out in the criteria. Both toolkits are free and widely used already, are available on both major mobile platforms, and possess useful features. However, upon further inspection, ARCore has the advantage over ARToolKitX due to the plethora of pre-existing example applications, and complete sets of developer guides constructed by Google’s development team which will assist in lowering development times. In addition, the inbuilt features such as environmental awareness and depth understanding will also be greatly advantageous to assist with AR integration.

4.1.3 Development

To implement an Android application using the ARCore framework, the project requires dependencies to be installed, including the SceneForm SDK. Once enabled, the first stage is to check the user’s device for AR ‘required’ availability upon start-up, meaning that their android SDK must be at minimum version 24 (Android 7.0 (Nougat)) or later[1]. If it meets the requirements, the user is then prompted to allow the app to access the device’s camera before the ARCore session is established. When permissions have been granted to access to the camera, the remaining operations to implement an object measuring application can defined in a series of logical steps as seen below in Figure 8.

[1] Minimum requirement enforced for AR to function https://developers.google.com/ar/develop/java/enable-arcore

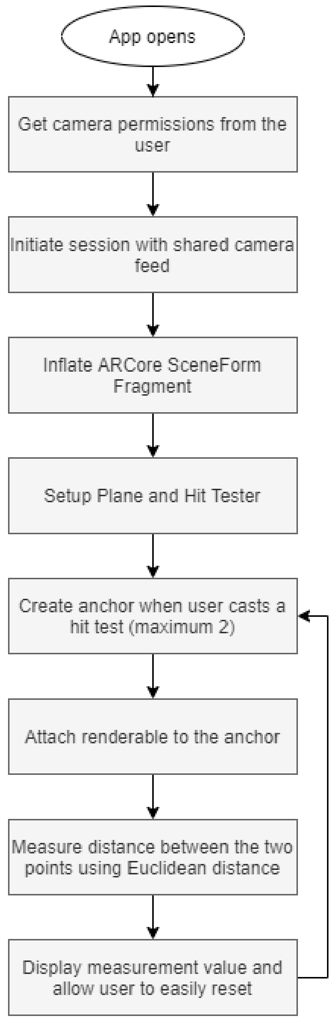


Figure 8 – ARCore Implementation Flowchart

Once camera permission is provided from the user after they launch the app, the initialization of the virtual scene is handled through an inflated SceneForm fragment, whereby any 3D assets are rendered within the camera's view. At this stage, the fragment also handles camera initialization and permission handling, and once successful, a shared camera feed from the on-board camera interlinked with ARCore.

After a camera uplink has been established, the next stage is to find a plane within the environment using feature points. A user will do this by moving their device throughout their environment, on any flat surfaces. In most cases, the more varied the surface is, the easier it is for ARCore to establish a plane. This is due to its innate environmental understanding feature-set, whereby contrasts in texture, colour and shape enable ARCore to find feature points easier.

Once a plane has been found within the environment, the user can then request to measure the object, and through the press of a button, ARCore will project a ray from that coordinate point on the two-dimensional screen into the three-dimensional virtual environment. If it is found that it intersects the plane, or anchor, data about the objects found is made available, allowing for user interaction to be managed. Figure 9 below demonstrates how a plane might be established from the associated feature points.

Figure 9 – Establishing a Plane Using ARCore (Image is taken within a virtual environment)

An anchor is created at the point of intersection on an established plane, after the initial input from the user's mobile device's screen. Anchors define a three-dimensional pose in world space, that is then pinned in the context of the trackable, in this case, the plane. This means that the anchor point itself can be properly tracked during motion, irrespective of where the object is placed in world space[2].

Once an anchor is established, a Renderable marker, such as a three-dimensional (3D) sphere, is then programmatically attached to the anchor to visually denote the anchor placement. From there, ARCore's SceneForm API handles interaction-based events that rely on further hit tests when using an object type of TransformableNode. These nodes allow the world position of anchors to be moved within the virtual environment if it remains on an established plane. Once two anchor points in the environment have been established, as seen in Figure 10 below, the distance between the spheres can be measured using the Euclidean distance formula, where

$$\text{distance} = \sqrt{\left[(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2\right]}.$$

[2] Working with Google ARCore Anchors as per: https://developers.google.com/ar/develop/developer-guides/anchors

Figure 10 – 3D Marker Placement for Measuring Objects

Here, distance will a measurement value in centimetres which equates to the length of the true straight line between the world position coordinates of the first anchor point x, y and z, and x1, y1, and z1 respectively and similarly for the second point. The resulting number is then returned and can be parsed to any layout object which uses string text.

4.1.4 Validation Metrics

For validating our approach, the ground truth values, to be measured by hand with a ruler or measuring tape, will be compared with both the farmer's technique of estimating, and the recorded measurements taken using the proposed instrument. In addition, the time taken to measure a series of object by hand, using estimation, and using the measuring instrument will also be compared. Through this, we will be able to verify if our methods outperform traditional abalone measurement approaches in terms of accuracy and speed and confirm if the degree of error is reduced overall. These results will then be congregated and critically compared and benchmarked against existing approaches to demonstrate the degree of improvement it provides.

4.2 Deep Learning on Mobile

4.2.1 Deep Learning Mobile Library

Many larger technology companies such as Google and Facebook have developed frameworks for deep learning applications on mobile devices. Selecting the most suitable SDK based on requirements is a vital component in determining the success of an application, because it directly affects the development times, and can alter the performance of the final product. TensorFlow Lite and PyTorch are two widely used, publicly available machine learning frameworks that support deep neural network models on mobile devices such as image classification and object detection [32, 33]. Both are in wide scale production use and have Application Programming Interfaces (APIs) written in the most popular programming languages, meaning they can be adapted into not only an Android application, but other operating environments as well.

TensorFlow Lite is an open-source machine learning framework developed by the Google Brain Team which is designed to streamline the process of incorporating AI into mobile applications. Through the design of an intuitive and high-level API, many of the underlying machine learning parameters and software details are abstracted [9]. This simplification not only results in quicker learning and deployment speeds from a development standpoint, but it also means that more comprehensive documentation guides and example GitHub repositories are available to showcase its range of features. Furthermore, Google uses TensorFlow in products such as Google Translate and Gmail, which are widely used today. This incorporation indicates the use of TensorFlow lite should also have good synergy with Google's Augmented Reality library, ARCore, which suggests that it will be a favourable pair for the project.

PyTorch on the other hand, is a deep learning library developed by Facebook's AI Research Lab (FAIR). Compared to TensorFlow, the supporting guides, documentation, and direct examples offered by PyTorch are fewer. This is because the primary purpose of the library is catered towards accelerated research prototyping, making it a popular choice among academic researchers [33]. It provides APIs that can cover most of the common pre-processing and integration tasks, but one of its major downfalls is that it is still in a beta stage, meaning documentation is rapidly changing and becomes equally obsolete as the library is continually updated.

4.2.2 Evaluation

For the purposes of an object detection and measurement Android application, both PyTorch and TensorFlow Lite, a more extensive investigation must be performed to determine their suitability to the given task. To do so, four major criteria have been selected: 1) What platforms is the SDK available on, 2) What language APIs does the SDK support, 3) What are the main features of each SDK, and 4) Is it free to use? As seen below in Table 3 below, a cross-examination of both SDKs was developed from the documentation.

	TensorFlow Lite	**PyTorch**
Supported Platforms	iOS, Android	iOS, Android, Linux, Mac, Windows
Supported Languages	Python, Java, C++	Python, Java, C++
Main Features	3 Object Detection 4 Image Classification 5 Pose Estimation 6 Text Classification 7 Image Segmentation 8 Speech and Gesture Recognition' 9 Digit Classification	• Object Detection • Image Classification • Pose Estimation • Text Classification • Image Segmentation • Speech and Gesture Recognition
Pricing	Free	Free

Table 3 – Comparison between TensorFlow Lite and PyTorch SDKs

From this, both SDKs appear to possess similar qualities, both supporting the major mobile platforms, programming languages, and main features required for the project. However, upon further inspection, due to the unstable nature of PyTorch's development stage – still being in beta, it offers fewer officially published examples and well-defined, up to date use cases. In comparison, TensorFlow Lite is a mature library that has numerous officially published tutorials, guides, and examples, as well as a plethora of third-party projects freely available on GitHub. In addition, TensorFlow Lite also shares similarities and commonalities with ARCore, thus making it the preferred choice for this implementation.

4.2.3 Datasets

Using TensorFlow Lite, the general steps outlined in Figure 9 below were used to retrain and embed two models – one for nursery, and another for weaner/grow out stage abalone.

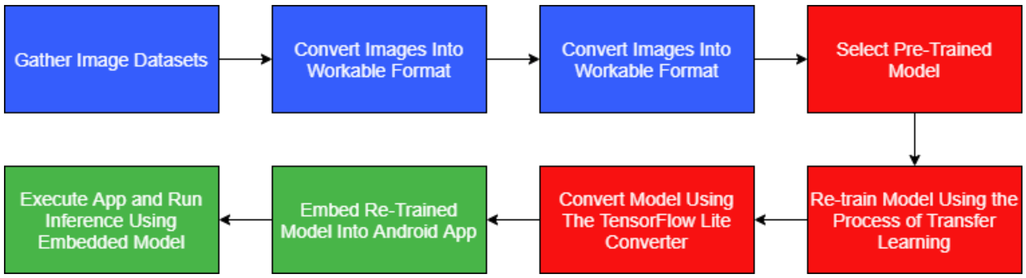


Figure 11 – Model Implementation Flowchart

A sample collection of around n=105 images of grow-out, weaner and n=130 nursery abalone obtained by during a James Cook University (JCU) research field trip in December 2020 from a farm in Victoria. Images would consist of either nursery growth plates with between 0 to upwards of 20 juvenile abalone per image, or for grow out and weaner stages, which had from to 10 to 50 per plate. With these images collected, a series of pre-processing steps as outlined in Figure 10, were used to ensure that the data was of quality and acceptable for model training.

First, all images were resized to 416x416 pixels as this is the required input for the chosen network architecture. Once resized, and due to the limited amount of quality images available for model training, some data augmentation steps were applied such as mirroring, rotation, brightness, and noise changes. The aim of this technique is to generate more variety and conditions from a single image, to simulate the image capturing under different circumstances. This was performed to simultaneously test the robustness of the implementation process, increase the amount of training data, and prevent issues with overfitting, where the model learns the detail and noise in the dataset resulting in reduced performance, or underfitting, where the model fails to capture sufficient detail.

Each image in both datasets were then manually annotated using the graphical interface, LabelImg [51], which produced a series of images and Extensible Markup Language (XML) annotation files. The data was then subsequently split into training, test, and validation subsets of roughly a 70/15/15% split, respectively. This is done so that once the model has been re-trained to classify objects based on a specific set of data, the model can be evaluated on new data to assess its accuracy.

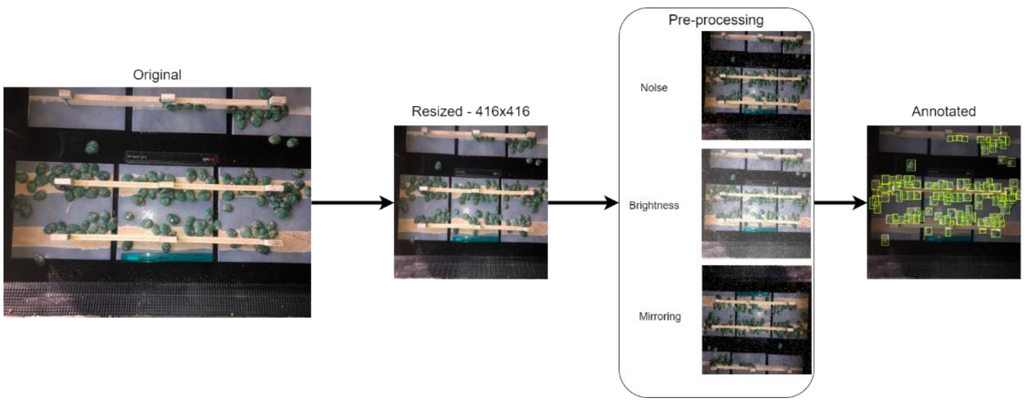


Figure 12 – Image Pre-processing and Annotation Flow

Moreover, for the nursery stage abalone dataset, since we were unable to obtain specimens for the experimentation phase, a decision was made to introduce a subset of images of groups of small rounded grey pebbles ranging in lengths from 10-20mm. A small dataset of these pebbles, annotated and labelled as nursery abalone was included in the final dataset for training. Such an approach would give us the flexibility to test the counting implementation on weaner/grow-out, and nursery stage abalone without hugely impacting the overall dataset accuracy.

4.2.4 Training

To implement an object detection model, it is imperative to understand how the transfer learning process can be used to retrain an existing object detection model. Training a model from scratch can take immense amounts of computational power and time, so instead, the use of a pre-trained model allows for the transference of knowledge from one domain, to another similar one, through the reuse of learned information [29]. As such, using a base model such as the YOLOv4 checkpoint that is trained on the Common Objects in Context dataset – a collection of 330 thousand images with over 80 object classes – enables for the transference of knowledge to be applied to new domains, such as detecting abalone, without needing to fully train a model [44]. YOLOv4 was specifically chosen as our network architecture because it has the nearest real-time speeds, is lightweight, and still maintains a high level of accuracy compared to others which makes it ideal for mobile-based abalone detection.

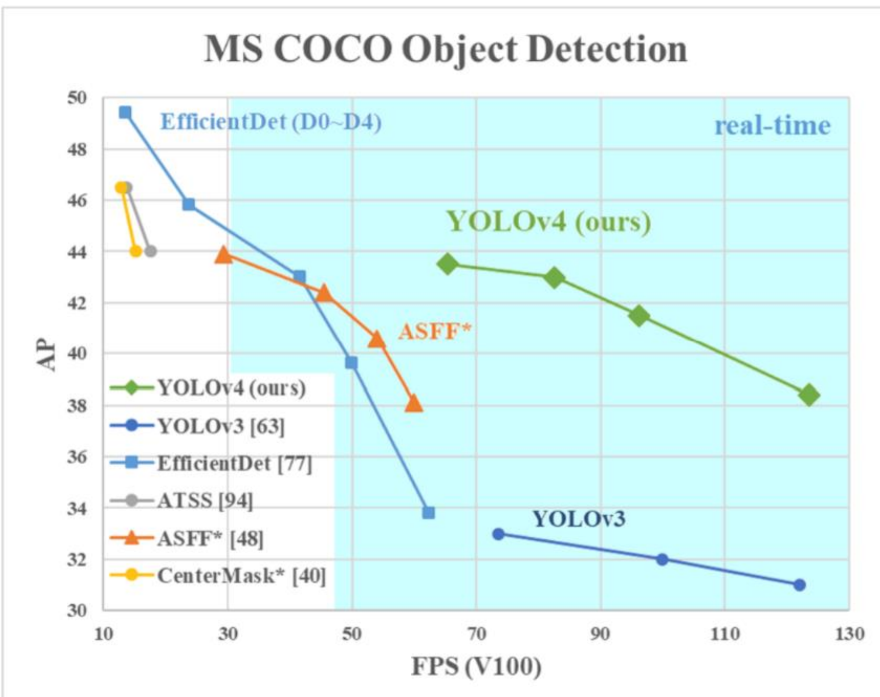


Figure 13 – YOLOv4 Average Precision versus Frames Per Second compared to other Algorithms [34]

During the training process, the initial layers of a Convolution Neural Network (CNN) incorporate more generic traits such as edges and blobs, with more specific features tailored to the dataset being included in the subsequent upper layers of the network [29]. This means that the removal of the top layers that are built around the original task enables the model to be re-trained to perform a different, but related task, without the need to build it from the initial layers. As part of the training process, the model is fine-tuned part-way through to ensure that feature extraction is performant and that its associations are adjusted according to the new dataset. The result of this is that the new training rate is significantly lower. Each model, for both datasets, was subsequently trained for 2000 epochs on custom data using YOLOv4 within a the Darknet framework. The resulting training performance metrics based on validation data can be seen in Table 4 below.

	Nursery Dataset	**Weaner/Abalone Dataset**
Average Precision (AP)	85.12	82.46
Precision	0.93	0.9
Recall	0.87	0.84
F1-Score	0.9	0.87
Mean Intersection over Union (mIoU)	65.49	84.17
Mean Average Precision (mAP)	0.8512	0.8246

Table 4 – Comparison of Training Performance Metrics for Both Datasets

4.2.5 Model Integration

After training was complete, the model is then converted into a mobile compatible format using conversion libraries. The library chosen for this was TensorFlow Lite, and this was because it possessed the qualities which supported our project, including it being quite a mature library, so there was plenty of information on it, it was interlinked with other AR libraries already such as ARCore, it was free and it supports Python and Java, two well-known languages to the primary researcher.

Using TensorFlow Lite, a simple labels text file is required in the same order that the model was trained in so that the objects can be correctly classified. The two residual files after the retraining process were then embedded in the project structure of the mobile app and the pathing variables were updated to ensure the new files associations were correctly connected. From here, images could be passed through the model and using the return output, bounding boxes could be drawn and their confidence scores could be produced.

4.2.6 Validation Metrics

In vision-based counting approaches, the count error will enable us to assess how accurate the implementation is versus the existing techniques. Counting accuracy is calculated by recording the average number of objects counted and the ground truth amount across a series of tests and cross referencing the results. By taking the ratio of correctly and incorrectly predicted instances to the total instances can be used to further analyse the reliability and robustness of the model.

$$Counting\ Accuracy\ = \left(\frac{Number\ of\ Correctly\ Counted\ Objects}{Total\ Number\ of\ Counted\ Objects}\right) * 100$$

There are also classification-based metrics based upon the confusion matrix by which we can evaluate the accuracy of the model. Firstly, during training the model can be tested against the 15% validation data subset for performance analysis, the results of which can be found in Table 4. Here, metrics such as recall, F1-score, and mIoU are calculated. Recall or sensitivity is used to calculate the true predictions from all correctly predicted data, and it involves taking the number of correctly counted objects, divided by the total number of actual, relevant objects.

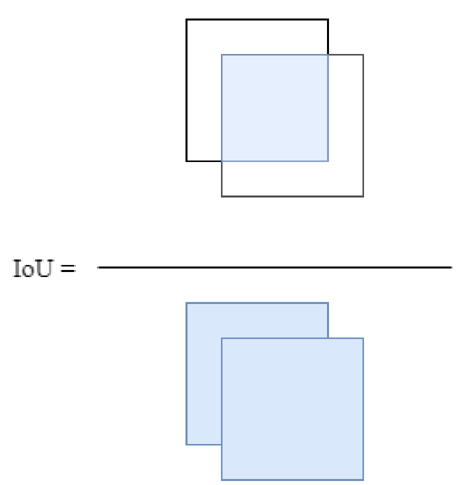
$$Recall = \frac{Number\ of\ Correctly\ Counted\ Objects}{Number\ of\ Actual\ Objects}$$

F1 score is the weighted average of precision and recall. The F1 score takes the precision and recall into account which leads it to usually be a better metric for evaluating a model than accuracy as long as false positives and false negatives have a similar cost. The F1 score is calculated using the equation:

$$F1 - Score = 2 * \frac{Recall\ x\ Precision}{Recall + Precision}$$

Furthermore, Intersection Over Union (IoU) is a metric used specifically for visual detection tasks because it computes the difference between ground truth annotations and predicted bounding boxes produced by the model. The output from an object detection model is the prediction confidence score, and bounding box coordinates for each object within the image. Based on the scores of each box, unnecessary boxes are removed based on an established threshold value, whereby any scores that fail to meet it are not used.

$$\mathrm{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Chapter 5

Experiments with AR Measuring Tool

5.1 Abalone Experimental Setup

For this research, an experimental design will be used where the independent variable will be the method of measuring and counting abalone – between ground truth, manual estimation to simulate performance from traditional techniques used by farmers and comparing it to the automated system. For dependent variables, we will be evaluating the change in the speed and accuracy of the measuring process when using both a manual approach compared against the proposed system, controlling for the species and life stage of abalone being evaluated – the Greenlip Abalone.

We tested the prototype system on 24 live mature-sized abalone at the James Cook University Campus to evaluate both the counting and measuring method and validate the results. As pictured in Figure 13, we followed a general process for testing and collecting data using both methods, in a way that was parallel to each other. Each method started by collecting data on the speed and accuracy of manual methods, before we used estimative counting and measuring techniques supported by abalone farming literature, before testing our methods.

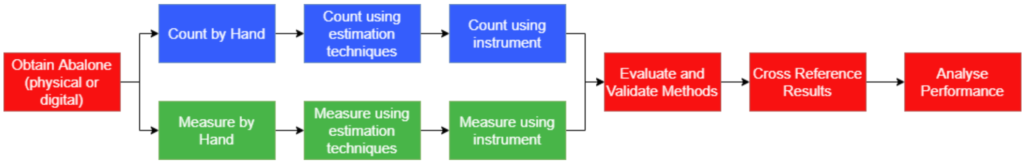


Figure 14 – Overview of Experiment Design

5.1.1 Experiment Conditions

Figure 15 – Abalone Measuring Experimental Setup

Experiments with the 24 live Greenlip Abalone were conducted within a typical laboratory setup with overhead lighting. The 24 abalone were placed on a gridded plastic tray and were roughly spaced apart. The measuring device was a Samsung Galaxy S8 Plus which is considered to be slightly outdated at the time of writing, being an older flagship initially released in April 2017. However, the model of phone was still capable of running AR-enabled applications. The phone was seated in a 32cm long camera gimbal with approximately a 45^{0} tilt. The camera gimbal was attached at a 90^{0} angle to a tripod set at a height of 40cm tall. The tray of abalone was then placed within the camera's view. The measuring tool was selected from with the android application and the tripod was moved for approximately 11 seconds before a plane was established from environmental feature points.

5.2 Original Method

For data collection, the first step was to measure each of the abalone individually by hand with a ruler, record their lengths for use as the ground truth values while simultaneously counting the time taken to measure all of them using a stopwatch. The resulting Table 5 below was constructed from the collection of ground truth measurements which took 148 seconds to record.

	Column 1 (mm)	**Column 2 (mm)**	**Column 3 (mm)**
A	62	70	49
B	72	60	54
C	64	53	62
D	66	61	60
E	70	45	60
F	63	56	51
G	32	60	50
H	32	46	52

Table 5 – Abalone Ground Truth Measurements

For approximation techniques such as those used by farmers, a similar approach as detailed in [25] was used. Using pseudorandom number generation, we randomly selected 20% of our abalone and used their measurements to approximate the lengths of the population. As such, 5 abalone were selected, and were hand measured for example, in one instance, our random numbers are 24, 11, 6, 13 and 9, where Column 1A = 1, Column 2A =2, etc. Keeping the ordering of abalone intact, we then measured each abalone by hand such that we ended up with a series of lengths corresponding to their position in Table 5. In our case, the lengths recorded were 52, 61, 54, 70, 62. When averaged, it resulted with a length of 59.8cm which was then applied to the remainder of the abalone. We repeated this process several times, and recorded the time taken with each instance.

We then measured using the proposed instrument mounted in the gimble arm, with a plane already established. It was observed to be near-impossible to measure the abalone object using a stationary camera position, because the size of the visual marker would block the other side of the object, and thus, a second marker could not be placed. In response, the phone was removed from the gimble and moved around from side to side at an arm's length away which allowed for the second marker to be consistently placed. The results of these measurements are placed in Table 6 below.

	Column 1 (mm)	**Column 2 (mm)**	**Column 3 (mm)**
A	51.7	65.2	56
B	78	68	79
C	68	51	51
D	69	72	55
E	68	49	67
F	67	60.7	56.9
G	45	71	62
H	43	56	67

Table 6 – Abalone Measurements Using the Proposed Instrument

5.3 Problems with Original Method

A large contributing factor to the loss of accuracy in these tests may be due to the surface in which they were conducted on. ARCore has a difficulty placing feature points on surfaces with visual uniformity, such as white tray which is used in these tests can lead to a decrease in average precision. However, a more likely explanation is that the methods used to place the measurement markers can be incorrect and imprecise because it is entirely dependent on where the user places the anchoring positions. Due to this reliance on user input, many factors can alter the end results such as the viewing angle of the device itself, and the precision of where the anchors are positioned relative to the edges of the object.

5.4 Modified Method

Thus, to reduce the need for user interaction, the original measurement method was modified the include two permanent on-screen markers, denoted by the red squares, as visualised in Figure 15 below.

Figure 16 – Modified Measuring Method On-screen Markers

With these defined 2D coordinates on screen, the user simply then needs to align the dots with the edges of the object they wish to measure and tap the single button at the bottom of the screen, and two rays will be cast from either marker.

This way, instead of relying on the user to point and tap to create an anchor inaccurately, this allows for much more precision.

The implementation of this modified method was after the initial experiments were conducted, and the live abalone were no longer available for testing. As a substitute, several Greenlip Abalone shells were obtained and tested on in a similar fashion to the previous method with both their ground truth lengths, as observed in Table 7 and measurements using the instrument were recorded as seen in Table 8. However, the estimation techniques were not applied here and instead, a normalised accuracy was calculated between all experiments.

	Column 1 (mm)	**Column 2 (mm)**	**Column 3 (mm)**
A	52	47	41
B	40	40	37
C	39	79	43
D	40	43	44
E	41	46	50
F	47	81	49
G	37	42	45
H	35	43	45

Table 7 – Abalone Shell Ground Truth for the Modified Method

	Column 1 (mm)	**Column 2 (mm)**	**Column 3 (mm)**
A	54.3	47.7	42
B	42	43	34.9
C	41	77.5	43.5
D	39.8	41	42.7
E	40.1	47.4	48
F	44.1	80.4	48.6
G	35	43.5	41

H	32	44.4	42

Table 8 – Abalone Shell Measurements Using the Proposed Instrument for the Modified Method

An interesting observation was made when conducting the experiments with the modified method. It was discovered that sometimes, particularly with smaller shells, the augmented dots which signify where ARCore has established a plane can overlap and interfere with the measurement process, as it can obscure an edge. However, the issue can be easily resolved either by physically moving the object or re-establishing the plane by moving the device around.

5.5 Fish Experimental Setup

To further analyse the modified AR measuring method, the experiments were extended to other aquatic products to validate robustness in the technique. In these experiments, two reef fish were obtained – one Black Pomfret Trevally and a Red-Throat Emperor. Like previous measuring experiments, the ground truth length values were recorded by hand measuring each fish using a tape measure.

5.5.1 Experiment Conditions

Both fish were kept frozen and were thawed until use to ensure accurate measuring. Each fish was placed onto a flat horizontal surface and measured using the same phone, with a bright overhead light. Estimation could not be conducted in the previous way because only two fish were available for the experiment. As such, an assumed estimation accuracy based on multiple abalone experiments was substituted and applied to the fish experiments.

Unlike previous experiments for measuring abalone, no tripod was used, instead, a handheld method was employed, whereby the device was held at an arm’s length away and moved from side to side for plane establishment. The use of real fish for testing meant that the markers had to be oriented to be at the furthest distant points at either side of each fish, a demonstration of which can be observed in Figure 16 and 17 below.

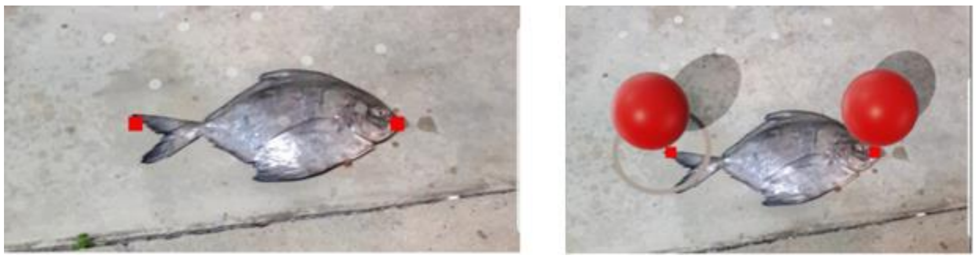


Figure 17 – Measuring Trevally Fish Using the Modified AR Method

Figure 18 – Measuring Red-Throat Emperor Fish Using the Modified AR Method

5.6 Results

From our counting experiments on both mature abalone and juveniles, we have found that through the usage of AR, our approach is successfully able to outperform manual and estimation techniques used by farmers. In terms of accuracy, which is calculated by comparing the total average ground truth length for abalone versus the total average measured length using our approach, our method shows that the tool initially was underperforming with a 6.8% loss in average accuracy across all measurements taken. This was later identified to be a result of over-reliance on user interaction, and thus a second iteration was developed which saw the use of on-screen markers to eliminate imprecisions. As such, the second method of measuring performed highly well during the experiments, achieving an average precision score equal to 99.08%. Figure 18 below highlights and compares each of the four methods of measuring.

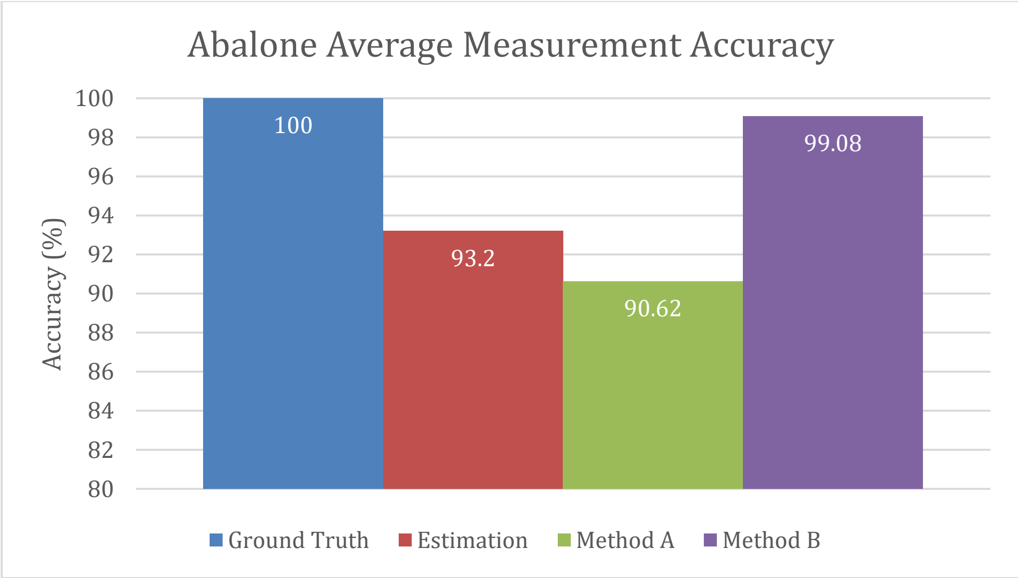


Figure 19 – Graph of the Abalone Average Measurement Accuracy

However, in terms of speed, the initial developed method did slightly improve upon hand measured systems, reducing the time taken from 150 to 125 seconds, and greatly improved on estimation techniques. Yet while the first method did increase efficiency, it still suffered in situations where small objects were being measured, because often the first marker placement would visually obstruct the second one from being placed. This issue was resolved with the second method, however, as both markers would be placed concurrently. This also had the double benefit of speeding up measurements, as users would only need to simply align the markers and tap once, whereas before, they needed to align, place one marker, then physically move to place the second, which increased the time taken for counting. As seen in Figure 19, the second method of counting was able to greatly outperform all other methods, decreasing the time from 125 to 93 seconds.

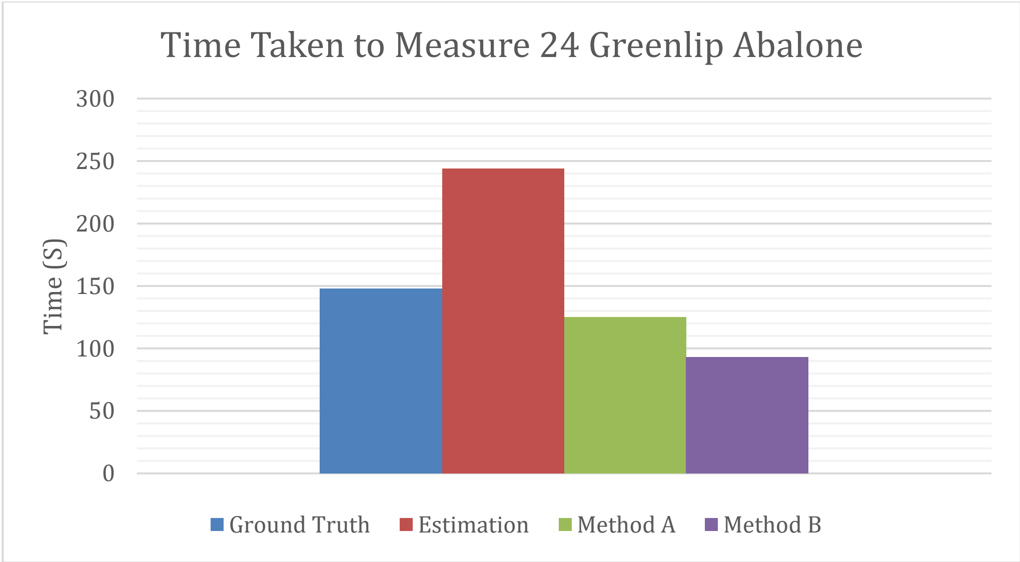


Figure 20 – Graph of the Time Taken to Measure 24 Greenlip Abalone

For reasons of robustness, the second method was also tested on two species of fish, and as observed below in Figure 20, the technique was able to produce comparable, high levels of accuracy despite the object being considerably larger.

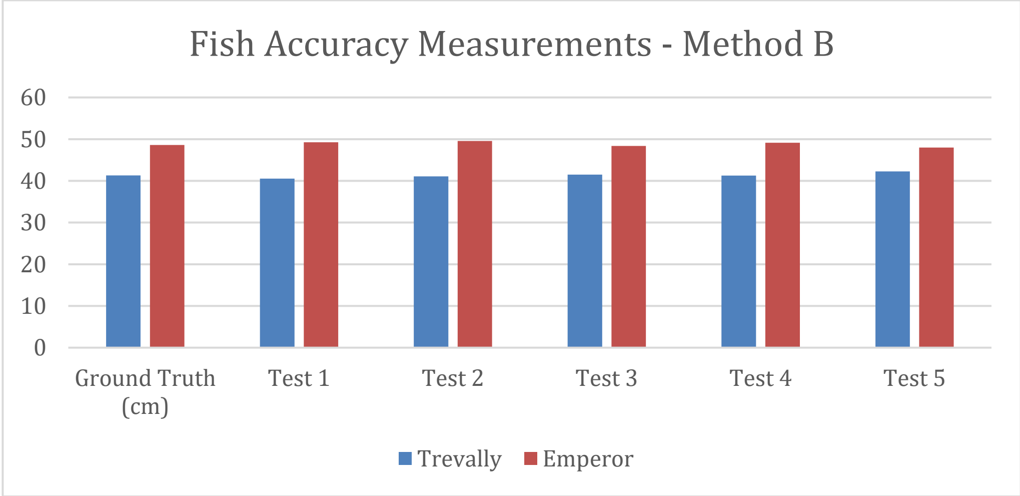


Figure 21 – Graph of the Fish Accuracy Measurements Using the Modified Method

Taking a closer look at the percentage error calculated from the fish measurement tests, it can be observed that the technique maintains accuracy to within a maximum of 2.5% of the object's actual size.

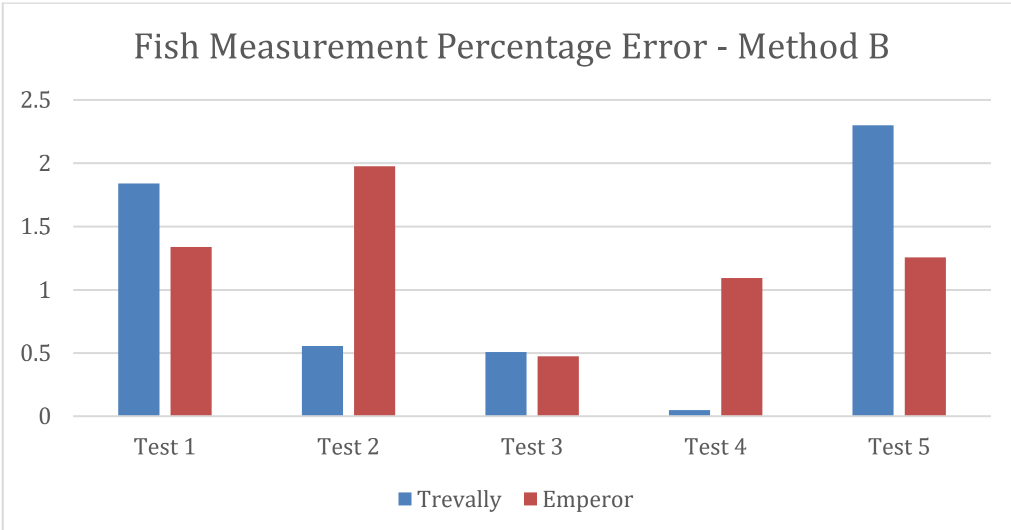


Figure 22 – Graph Showing the Percentage Error Calculated from the Fish Measurement Tests Using the Modified Method

Through the comparison of average measurement accuracies across all of the measured objects as visualised in Figure 22, it becomes apparent that the first method proposed fell short of expectation and performed poorly compared to the other methods. However, with the tweaks in framework design and decision to restrict user input, the second implemented method performs solidly in all situations with accuracies between 98-100%.

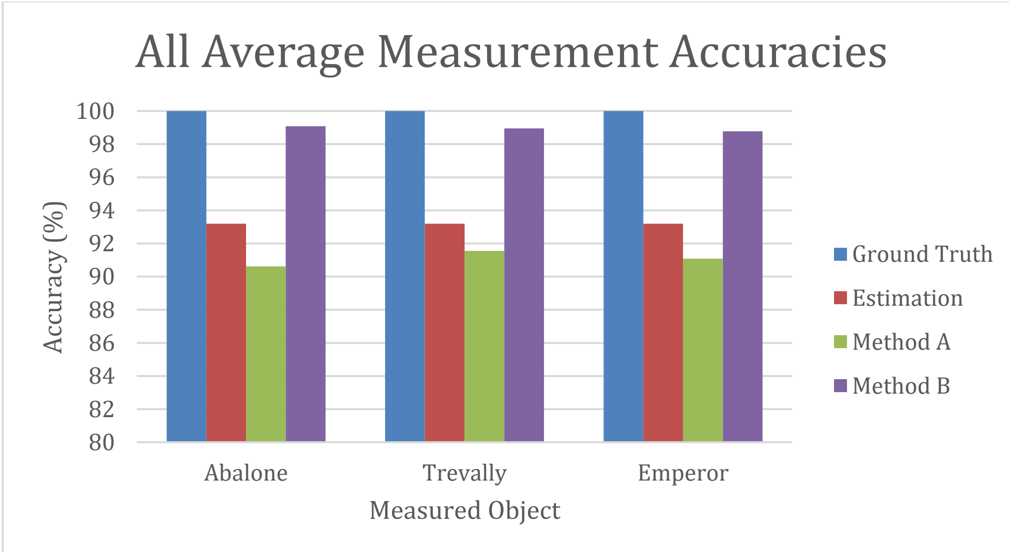


Figure 23 – Average Measurement Accuracies for all Experimented Methods

Chapter 6

Experiments with Object Detection Counting Tool

6.1 Abalone Grow out and Weaner Dataset

6.1.1 Experiment Conditions

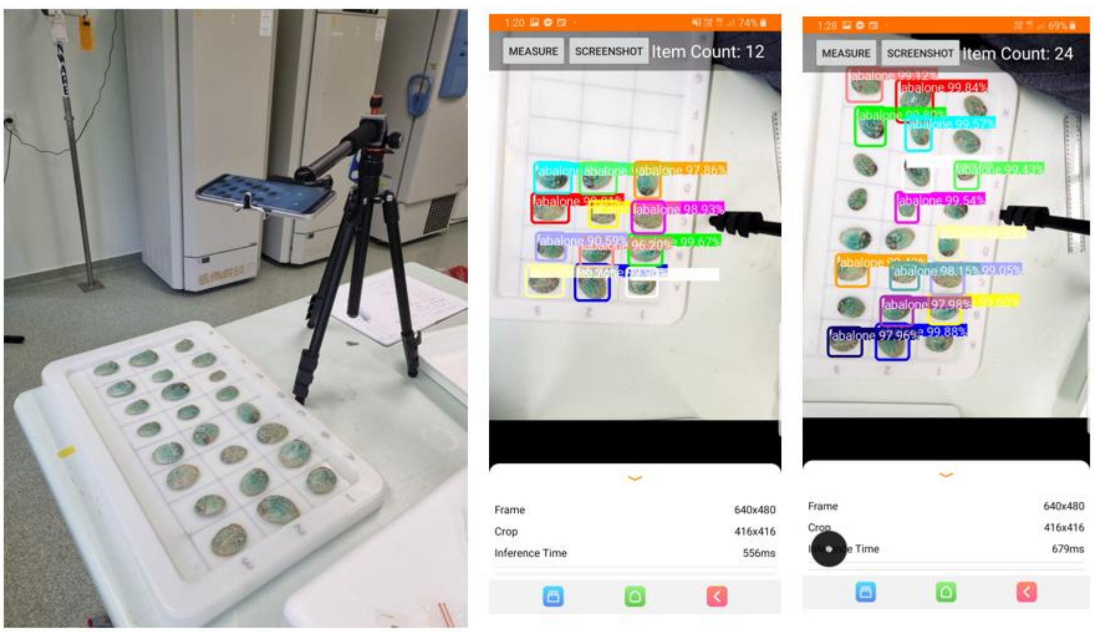


Figure 24 – Weaner and Grow out Abalone Counting Experimental Setup

Much like the experimental process undertaken for measuring objects, counting was achieved using a tripod mounted-camera setup within a laboratory environment at the JCU campus. Sitting flat and parallel to the 24 live abalone, the phone camera was situated 63cm above the tray, ensuring that the full tray was visible during the counting process.

Data collection was achieved by taking continuous screen captures of the counting process. Once 20 iterations of object counting were complete, abalone were slowly, individually removed from the tray, and again, 20 iterations of data were captured. In each iteration, the amount of objects correctly counted, incorrectly counted and the inference time was collected. This incrementally continued up until the dataset was halved to 12 abalone. This alteration was conducted to see if changing the amount of objects being detected would affect accuracy results.

6.2 Abalone Nursery Dataset

6.2.1 Experiment Conditions

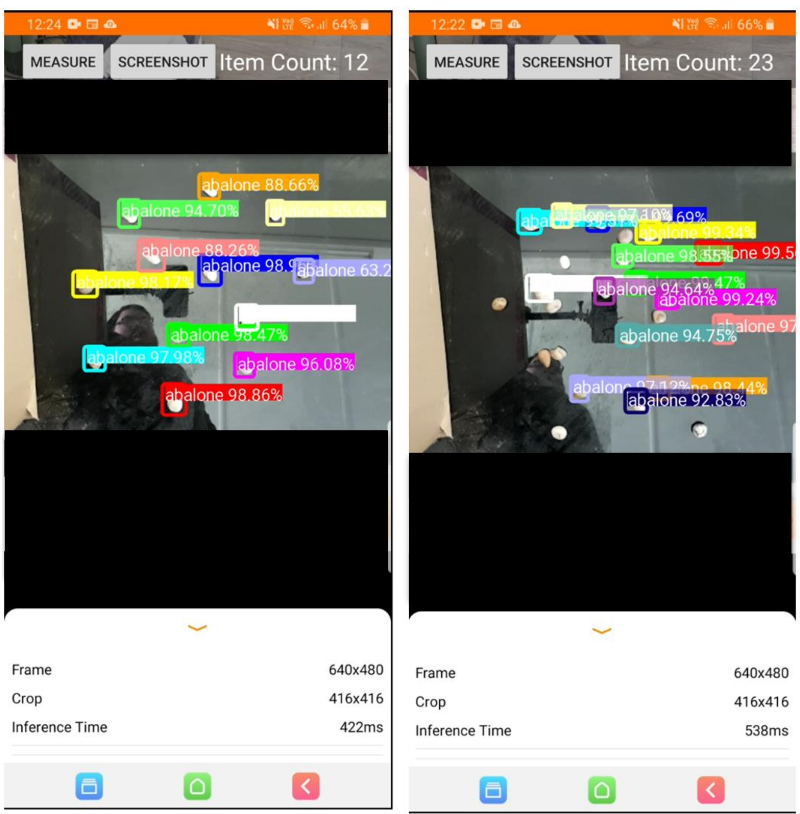


Figure 25 – Nursery Abalone Experimental Setup

For counting juvenile abalone, the lack of any real specimens for experimentation meant that they could be conducted outside of the laboratory. As such, the counting took place on top of a glossy, reflective black surface in an attempt to replicate similar conditions to the surface of water, or the nursery abalone growth plates. The counting instrument was suspended using a 32cm gimble such that the camera was 60cm and parallel above the round grey pebbles imitating juvenile abalone.

A process similar to ones previously described for counting was conducted, whereby the correct and incorrect number of objects counted was recorded for 20 iterations per number of objects. Starting at 24 juvenile abalone, the count was slowly decreased until there were only 12 remaining.

6.3 Results

From our experiments counting weaner/grow-out and nursery abalone, we have confirmed that object detection is capable of counting objects in real-time, where results are location and network independent, to an extent. When counting up to 15 objects, the current framework is capable of producing highly accurate results for both juvenile and adult abalone from around 95-100%. However, once the count of objects exceeds 15, the accuracy decreases linearly, as seen in Figure 25, as the technique struggles to recognise the presence of an additional abalone.

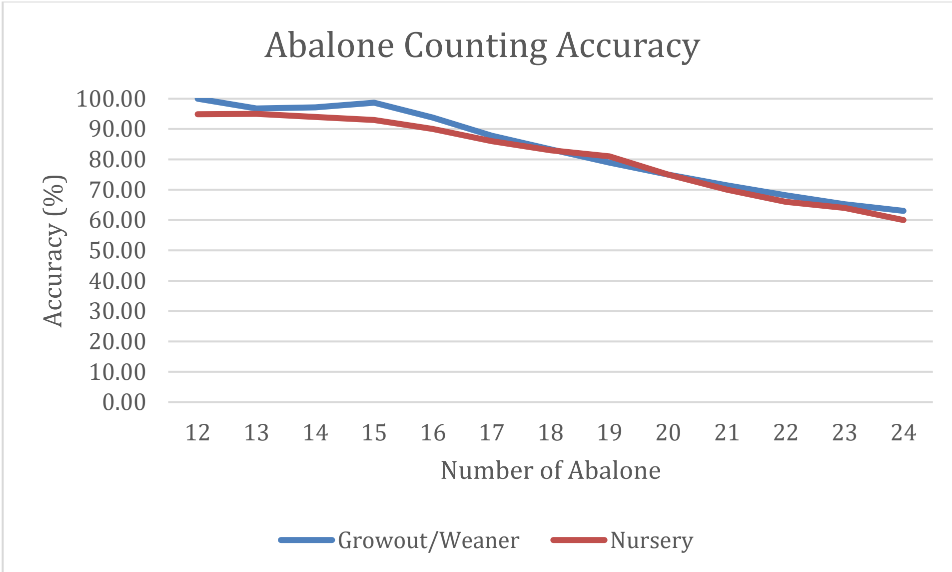


Figure 26 – Graph of Abalone Counting Accuracy

In terms of time taken, as highlighted in Figure 26, both models were able to achieve a modest average inference time for 24 objects equal to 606ms for weaner/grow-out and 394ms for nursery stages. This confirms that the implemented object detectors are capable of achieving real-time inference while still maintaining a high base level of accuracy for up until 15 objects at once.

Manual counting methods are difficult to quantify because it is highly dependent on the individual doing the counting. Assuming the average human response time is 231ms [52], it would take several seconds for an average person to count 24 objects with a high probability of achieving 100% accuracy, thus it was not considered in these results, as an image processing technique will nearly always be much faster.

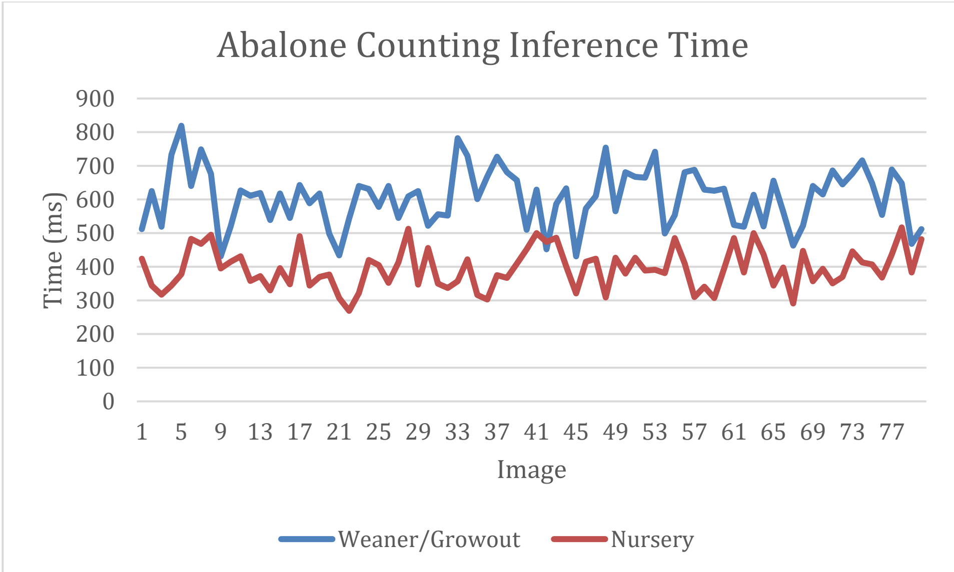


Figure 27 – Graph of Abalone Counting Inference Time

Chapter 7

Summary and Discussion

7.1 Summary of Contributions

Object detection and augmented reality are two rapidly evolving technology areas that have attracted many researchers. Through this research thesis, we have demonstrated a prototype approach to solving the issues experienced by abalone farmers when it comes to monitoring the growth of their populations. This research also has the potential the impact other areas of study, particularly fields in which more effective, image-based automation is required in unpredictable conditions or environments.

In Chapter 2, we analysed and critically reviewed a wide breadth and depth of literature relating to the area's abalone farming, DL, AR, and their applications in and outside of aquaculture. Outside of aquaculture, approaches were examined to evaluate the effectiveness of AR for measuring objects, and DL for counting. We highlight a diverse set of methods, organised into thematic groups, and derive the considerable gap where this research resides.

In Chapter 3, a conceptual framework overview is shown which seeks to combine and use the strengths of AR and object detection to compliment abalone farmer's growth monitoring practices. Here the major components, general decisions and program flow, and design choices for quantitative data collection and analysis are described.

In Chapter 4, more technical details are discussed and the method of implementation, and reasons behind the chosen technologies are discussed. A deeper look into the validation metrics to be used in the assessment of the proposed instrument.

In Chapter 5, experiments conducted using the proposed measuring system are explored. Here, materials and methods of efficiency and accuracy data collection are described between the three independent variables of the study, being manual, approximative techniques and our AR measuring approach. We also explore the findings and re-development of a new robust method for increased performance.

In Chapter 6, experiments conducted using the developed counting tool on weaner, grow-out and nursery abalone are described. A discussion and comparison of manual counting methods versus the proposed instrument is also conducted.

Overall, we have shown that our framework is indeed more efficient and effective than manual and estimation monitoring methods. This, in turn, alleviates the problems of inaccuracy and inefficiency identified within previous methods. Our proposed method is real-time, location and network independent and it does improve the accuracy and speed of counting and measuring. Object recognition has proven to be a promising approach to automate the processes of counting abalone at various stages of their lifecycle and can be a good foundation for further automation research. Likewise, AR has shown that it is a fast and robust way of measuring objects and can be applied many different types of objects outside of abalone.

7.2 Discussion

From the experimental results, we can confidently conclude that our methods are both more efficient and accurate than traditional and estimation techniques for counting and measuring abalone. While our methods cannot be used simultaneously as we originally intended, the tools by themselves are useful and provide a degree of improvement over traditional methods that warrant its field usage.

For counting, our method was able to effectively count for up to 15 abalone efficiently with between 98-100% accuracy. For more than 15 objects, the accuracy slowly decreases linearly. We found that this seemed odd and somewhat contradictory based on our research and experiments, and as such would like to discuss this further.

One seemingly explanatory cause is due to the limitations of YOLOv4 architecture which has historically had issues counting small objects, evident through its performance on other datasets with smaller image sizes [34]. We believe, due to its single-shot detection architecture, it is not as suitable for capturing finer details present in smaller objects and 416x416 images. The model possesses a tradeoff of speed over accuracy, which was valued for our purpose, but in doing so, it is unable to fully detect all objects in an image to a value greater than the minimum specified accuracy threshold value.

Another explanation is our method of experimentation, where the objects may have been too close together, or camera too distant away from the subjects themselves. Due to the scope of our research and time restrictions, we were unable to perform the diverse range of experiments to obtain further results for analysis. A potential future direction and solution may be to split the images up into smaller regions in real-time and have that be fed into the model for more counting performance.

From our testing, measuring using AR has proven to offer major speed advantages when it comes to measuring smaller objects, such as weaner abalone shells, or larger ones such as reef fish. Our method of measuring demonstrated robustness in either case, where the accuracy remained consistent across different sizes of objects, indicating that the method can be extended to other applications and fields such as agriculture, on-site training, or education. However, in the case of very small objects, AR may not be ideal solution as results from experiments revealed that sometimes the augmented information can obscure the edges of small objects, such that they are hidden from view.

With these results, it indicates that these approaches can assist farmers in nearly doubling their efficiency when stock taking without risking harm or shock to the animal, and without them requiring the assistance of trained technical staff who are forced to precisely, and time consumingly, remove abalone in a manner to prevent death. While this technique will never fully replace the technical expertise required to grow abalone sustainably and efficiently, it can serve as an optimisation tool, to go alongside farmers for increased productivity and objectivity monitoring stock growth.

7.3 Future Work

The framework described in this research thesis exists mainly as a proof of concept. For true integration into an abalone aquacultural farm, additional research would be required to find the best approach and method of counting, which can recognize multiple, overlapping, moving abalone at various orientations and simultaneously measure their lengths, with the added difficulty of water reflectivity.

Our implementation also still does not fully overcome the issue of the need for trained technical staff, as it still requires manual operation by a user. Additional modifications to the approach would revolve around further removing the need for user-interaction such as auto-calibration and event triggering, which, in turn, would help improve measuring accuracy and reduce data capturing times.

The method proposed was entirely tested in laboratory conditions, where lighting was consistent, and conditions were calm. As a result, more testing of counting is required, with additional variations in conditions, such as moving objects, motion blur and ideally in-situ validation.

Finally, due to restrictions in time, full integration between ARCore and TensorFlow lite for both real-time measuring and counting simultaneously could not be achieved. Further investigation into the combination of these technologies could yield valuable results that are widely applicable to areas within and outside of aquaculture. The gap identified from our literature review still exists and may be worthy of future research with more automated methods, such as the employment of drones.

Bibliography

[1] M. Troell, P. Rönnbäck, C. Halling, N. Kautsky, and A. Buschmann, "Ecological engineering in aquaculture: use of seaweeds for removing nutrients from intensive mariculture," *Journal of Applied Phycology*, vol. 11, no. 1, pp. 89–97, Feb. 1999, doi: 10.1023/A:1008070400208.

[2] T. S. Dlaza, "GROWTH OF JUVENILE ABALONE UNDER AQUACULTURE CONDITIONS," p. 98. [Online]. Available: https://core.ac.uk/download/pdf/58912523.pdf

[3] L. Venter, D. T. Loots, A. Vosloo, P. Jansen van Rensburg, and J. Lindeque, "Abalone growth and associated aspects: Now from a metabolic perspective," *Reviews in Aquaculture*, vol. 10, pp. 451–473, Jun. 2018, doi: 10.1111/raq.12181.

[4] H. Hong, X. Yang, Z. You, and F. Cheng, "Visual quality detection of aquatic products using machine vision," *Aquacultural Engineering*, vol. 63, pp. 62–71, Dec. 2014, doi: 10.1016/j.aquaeng.2014.10.003.

[5] R. Reaburn and S. Edwards, "Modelling the growth of cultured greenlip abalone (Haliotis laevigata Leach): practical modes of measurement," *Aquaculture Research*, vol. 34, no. 3, pp. 253–259, 2003, doi: https://doi.org/10.1046/j.1365-2109.2003.00814.x.

[6] P. L. França Albuquerque *et al.*, "Automatic live fingerlings counting using computer vision," *Computers and Electronics in Agriculture*, vol. 167, p. 105015, Dec. 2019, doi: 10.1016/j.compag.2019.105015.

[7] F. Orciuoli, F. J. Orciuoli, and A. Peduto, "A Mobile Clinical DSS based on Augmented Reality and Deep Learning for the home cares of patients afflicted by bedsores," *Procedia Computer Science*, vol. 175, pp. 181–188, Jan. 2020, doi: 10.1016/j.procs.2020.07.028.

[8] P. A. Cook, "Recent Trends in Worldwide Abalone Production," *shre*, vol. 35, no. 3, pp. 581–583, Oct. 2016, doi: 10.2983/035.035.0302.

[9] Y. Deng, "Deep learning on mobile devices: a review," in Mobile Multimedia/Image Processing, Security, and Applications, 2019, doi: 10.1117/12.2518469

[10] D. Mourtzis, J. Angelopoulos, and N. Panopoulos, “A Framework for Automatic Generation of Augmented Reality Maintenance & Repair Instructions based on Convolutional Neural Networks,” *Procedia CIRP*, vol. 93, pp. 977–982, Jan. 2020, doi: 10.1016/j.procir.2020.04.130.

[11] Z.-H. Lai, W. Tao, M. C. Leu, and Z. Yin, “Smart augmented reality instructional system for mechanical assembly towards worker-centered intelligent manufacturing,” *Journal of Manufacturing Systems*, vol. 55, pp. 69–81, Apr. 2020, doi: 10.1016/j.jmsy.2020.02.010.

[12] D. Li et al., “Automatic counting methods in aquaculture: A review,” *Journal of the World Aquaculture Society*, vol. 52, no. 2, pp. 269–283, 2021, doi: https://doi.org/10.1111/jwas.12745.

[13] M. B. P. MS, R. K. B. PhD, P. E. Kenneth C. Diehl PhD, and C. R. H. PhD, “Machine Vision Based Oyster Meat Grading and Sorting Machine,” *Journal of Aquatic Food Product Technology*, Oct. 2008, doi: 10.1300/J030v03n04_02.

[14] D. J. White, C. Svellingen, and N. J. C. Strachan, “Automated measurement of species and length of fish by computer vision,” *Fisheries Research*, vol. 80, no. 2, pp. 203–210, Sep. 2006, doi: 10.1016/j.fishres.2006.04.009.

[15] L. J. Baumgartner, M. Bettanin, J. McPherson, M. Jones, B. Zampatti, and K. Beyer, “Influence of turbidity and passage rate on the efficiency of an infrared counter to enumerate and measure riverine fish,” *Journal of Applied Ichthyology*, vol. 28, no. 4, pp. 531–536, 2012, doi: https://doi.org/10.1111/j.1439-0426.2012.01947.x.

[16] R. Koprowski *et al.*, “Mobile sailing robot for automatic estimation of fish density and monitoring water quality,” *BioMed Eng OnLine*, vol. 12, no. 1, p. 60, 2013, doi: 10.1186/1475-925X-12-60.

[17] L. J. Baumgartner, M. Bettanin, J. McPherson, M. Jones, B. Zampatti, and K. Beyer, “Influence of turbidity and passage rate on the efficiency of an infrared counter to enumerate and measure riverine fish,” *Journal of Applied Ichthyology*, vol. 28, no. 4, pp. 531–536, 2012, doi: https://doi.org/10.1111/j.1439-0426.2012.01947.x.

[18] S. Cao, D. Zhao, X. Liu, and Y. Sun, "Real-time robust detector for underwater live crabs based on deep learning," *Computers and Electronics in Agriculture,* vol. 172, p. 105339, May 2020, doi: 10.1016/j.compag.2020.105339.

[19] T. H. Pinkiewicz, G. J. Purser, and R. N. Williams, "A computer vision system to analyse the swimming behaviour of farmed fish in commercial aquaculture facilities: A case study using cage-held Atlantic salmon," *Aquacultural Engineering*, vol. 45, no. 1, pp. 20–27, Jul. 2011, doi: 10.1016/j.aquaeng.2011.05.002.

[20] V. M. Papadakis, I. E. Papadakis, F. Lamprianidou, A. Glaropoulos, and M. Kentouri, "A computer-vision system and methodology for the analysis of fish behavior," *Aquacultural Engineering*, vol. 46, pp. 53–59, Jan. 2012, doi: 10.1016/j.aquaeng.2011.11.002.

[21] C. Rasmussen, J. Zhao, D. Ferraro, and A. Trembanis, "Deep Census: AUV-Based Scallop Population Monitoring," in 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, Oct. 2017, pp. 2865–2873. doi: 10.1109/ICCVW.2017.338

[22] Martignac, F., Daroux, A., Bagliniere, J.-L., Ombredane, D., & Guillard, J. "The use of acoustic cameras in shallow waters: new hydroacoustic tools for monitoring migratory fish population. A review of DIDSON technology." *Fish and Fisheries,* 16(3), 486–510. doi:10.1111/faf.12071

[23] S. A. Vo, J. Scanlan, and P. Turner, "An application of Convolutional Neural Network to lobster grading in the Southern Rock Lobster supply chain," *Food Control*, vol. 113, p. 107184, Jul. 2020, doi: 10.1016/j.foodcont.2020.107184.

[24] A. S. Eklind, Love, "An exploratory research of ARCore's feature detection," KTH, pp. 6-7, 2018. [Online]. Available: https://pdfs.semanticscholar.org/9741/4cccbbf2e298b4e29ca3f9c830c6a69dca34.pdf

[25] M. P. Heasman, N. Savva, New South Wales, Department of Primary Industries, Port Stephens Fisheries Centre (N.S.W.), and Fisheries Research & Development Corporation (Australia), "Manual for intensive hatchery production of abalone: theory and practice for year-round, high density seed

production of blacklip abalone (Haliotis rubra)," Nelson Bay, NSW: NSW Dept. of Primary Industries, Port Stephens Fisheries Centre, 2007. [Online]. Available: https://www.lib.washington.edu/msd/norestriction/b67512665.pdf

[26] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[27] A. K. Dash, S. K. Behera, D. P. Dogra, and P. P. Roy, "Designing of marker-based augmented reality learning environment for kids using convolutional neural network architecture," *Displays*, vol. 55, pp. 46–54, Dec. 2018, doi: 10.1016/j.displa.2018.10.003.

[28] F. Orciuoli, F. J. Orciuoli, and A. Peduto, "A Mobile Clinical DSS based on Augmented Reality and Deep Learning for the home cares of patients afflicted by bedsores," *Procedia Computer Science*, vol. 175, pp. 181–188, Jan. 2020, doi: 10.1016/j.procs.2020.07.028.

[29] Z. Gao, Y. Zhang, and Y. Li, "Extracting features from infrared images using convolutional neural networks and transfer learning," *Infrared Physics & Technology*, vol. 105, p. 103237, Mar. 2020, doi: 10.1016/j.infrared.2020.103237.

[30] P. L. França Albuquerque *et al.*, "Automatic live fingerlings counting using computer vision," *Computers and Electronics in Agriculture*, vol. 167, p. 105015, Dec. 2019, doi: 10.1016/j.compag.2019.105015.

[31] D. Lee, K. Lee, S. Kim, and Y. Yang, "Design of an Optimum Computer Vision-Based Automatic Abalone (Haliotis discus hannai) Grading Algorithm," *Journal of Food Science*, vol. 80, no. 4, pp. E729–E733, Feb. 2015, doi: 10.1111/1750-3841.12799.

[32] Google Brain Team, "TensorFlow Lite guide", TensorFlow, 2020. [Online]. Available: https://www.tensorflow.org/lite/guide.

[33] Facebook's AI Research Lab, "PyTorch Mobile," PyTorch. [Online]. Available from: https://www.pytorch.org/mobile/android

[34] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” arXiv:2004.10934 [cs, eess], Apr. 2020. [Online]. Available: http://arxiv.org/abs/2004.10934

[35] D. W. F. Van Krevelen and R. Poelman, “A Survey of Augmented Reality Technologies, Applications and Limitations,” IJVR, vol. 9, no. 2, pp. 1–20, Jan. 2010, doi: 10.20870/IJVR.2010.9.2.2767.

[36] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, “Augmented reality technologies, systems and applications,” *Multimedia Tools and Applications*, vol. 51, pp. 341–377, Jan 2011.

[37] R. Lovreglio and M. Kinateder, “Augmented reality for pedestrian evacuation research: Promises and limitations,” *Safety Science*, vol. 128, p. 104750, Aug. 2020, doi: 10.1016/j.ssci.2020.104750.

[38] S. G. Conti, P. Roux, C. Fauvel, B. D. Maurer, and D. A. Demer, “Acoustical monitoring of fish density, behavior, and growth rate in a tank,” *Aquaculture*, vol. 251, no. 2–4, pp. 314–323, Feb. 2006, doi: 10.1016/j.aquaculture.2005.06.018.

[39] “Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos:,” in Proceedings of the Third International Conference on *Computer Vision Theory and Applications*, Funchal, Madeira, Portugal, 2008, pp. 514–519. doi: 10.5220/0001077705140519.

[40] J. M. Hernández-Ontiveros et al., “Development and implementation of a fish counter by using an embedded system,” *Computers and Electronics in Agriculture*, vol. 145, pp. 53–62, Feb. 2018, doi: 10.1016/j.compag.2017.12.023.

[41] Z. Oufqir, A. El Abderrahmani, and K. Satori, “From Marker to Markerless in Augmented Reality,” *Embedded Systems and Artificial Intelligence*, Springer Singapore, 2020, pp. 599–612 [Online]. Available: http://dx.doi.org/10.1007/978-981-15-0947-6_57

[42] R. Pryss, M. Schickler, J. Schobel, M. Weilbach, P. Geiger, and M. Reichert, “Enabling Tracks in Location-Based Smart Mobile Augmented

Reality Applications," Procedia Computer Science, vol. 110, pp. 207–214, 2017, doi: 10.1016/j.procs.2017.06.086.

[43] S. Boonbrahm, P. Boonbrahm, and C. Kaewrat, "The Use of Marker-Based Augmented Reality in Space Measurement," Procedia Manufacturing, vol. 42, pp. 337–343, 2020, doi: 10.1016/j.promfg.2020.02.081.

[44] "COCO - Common Objects in Context." [Online]. Available: https://cocodataset.org/#home.

[45] G. Developers, "Fundamental concepts | ARCore ", Google Developers, 2020. [Online]. Available: https://developers.google.com/ar/discover/concepts.

[46] Apple Developers, "ARKit - Augmented Reality," Apple Developer, 2020. [Online]. Available: https://developer.apple.com/augmented-reality/arkit/.

[47] Vuforia Developers, "Getting Started with Vuforia for Android Development." Vuforia Developer Library, 2020. [Online]. Available: https://library.vuforia.com/articles/Solution/Getting-Started-with-Vuforia-for-Android-Development.html.

[48] P. Lamb, "Developing an application using ARToolKitX" GitHub, 2019. [Online]. Available: https://github.com/artoolkitx/artoolkitx.

[49] MAXST Developers, "MAXST | - The most convenient and the easiest AR software development kit." MAXST Developers, 2020. [Online]. Available: http://maxst.com/#/en/arsdk

[50] Wikitude Developers, "Augmented Reality Experiences with Wikitude Cross Platform SDK," Wikitude. [Online]. Available: https://www.wikitude.com/products/wikitude-sdk/

[51] darrenl, tzutalin/labelImg. 2021. [Online]. Available: https://github.com/tzutalin/labelImg

[52] D. L. Woods, J. M. Wyma, E. W. Yund, T. J. Herron, and B. Reed, "Factors influencing the latency of simple reaction time," Front. Hum. Neurosci., vol. 9, 2015, doi: 10.3389/fnhum.2015.00131.