Using Image Processing to Automatically Measure Pearl Oyster Size for Selective Breeding

Adrian Lapico*, Mangalam Sankupellay*, Louis Cianciullo*, Trina Myers*, Dmitry A. Konovalov*,
Dean R. Jerry*†, Preston Toole*†, David B. Jones*†, Kyall R. Zenger*†,
*College of Science and Engineering, James Cook University, Townsville, Queensland, 4811, Australia
trina.myers@jcu.edu.au, dmitry.konovalov@jcu.edu.au
†Centre for Sustainable Tropical Fisheries and Aquaculture, James Cook University, Townsville, Queensland, 4811, Australia
kyall.zenger@jcu.edu.au, dean.jerry@jcu.edu.au

***Abstract*—The growth rate is a genetic trait that is often recorded in pearl oyster farming for use in selective breeding programs. By tracking the growth rate of a pearl oyster, farmers can make better decisions on which oysters to breed or manage in order to produce healthier offspring and higher quality pearls. However, the current practice of measurement by hand results in measurement inaccuracies, slow processing, and unnecessary employee costs. To rectify this, we propose automating the workflow via computer vision techniques, which can be used to capture images of pearl oysters and process the images to obtain the absolute measurements of each oyster. Specifically, we utilise and compare a set of edge detection algorithms to produce an image-processing algorithm that automatically segments an image containing multiple oysters and returns the height and width of the oyster shell. Our final algorithm was tested on images containing 2523 oysters (*Pinctada maxima*) captured on farming boats in Indonesia. This algorithm achieved reliability (of identifying at least one required oyster measurement correctly) equal to 92.1%.**

I. Introduction

Aquaculture is one of the world's fastest-growing food sectors, with the industry expected to grow to US$202 billion by 2020 [1]. Despite the size of this industry, technology-assisted selective breeding is still in the infancy stages of development compared to the corresponding terrestrial breading programs [2], which allow for the easy measurement and analysis of traits such as size and weight [3]–[5]. Measuring and recording these types of genetic traits in an efficient manner allows agricultural workers to save both money and time, as well as deliver a superior product. As such, these types of technologies are greatly needed in the aquaculture industry. Recently, there has been an increase in research for uses of cameras and sensors for genetic trait capture in aquaculture, however, for the most part, the focus has been on fish [6]. A particular aquaculture industry that would greatly benefit from this sort of technology-assisted selective breeding is the pearl oyster industry.

Pearl oyster growth rate tracking is beneficial for the selective breeding of pearl oysters to optimise future pearl production. Pearl oyster cultivation currently requires oysters to be retrieved from the ocean monthly to be cleaned. During this cleaning time, farms with a growth rate tracking program measure each oyster individually, using a ruler or callipers. Pearl oyster growth rates are used to determine pearl oysters' health and estimate the quality and size of pearls they may potentially produce [7].

The main breeding objective is to get the oysters to seeding size as soon and as predictable as possible and to maintain uniformity amongst all animal groups as much as possible. Secondly, understanding growth rates over time allows better farm management decisions to be made earlier on in their life cycle. It allows correlated information on health or growth to be monitored in association with environmental data. Studies have shown that pearl oysters with slower growth rates but higher starting size have a higher survival rate than smaller pearl oysters with fast growth rates [8]. As such, pearl oyster growth rate is a primary trait used for selective breeding of pearl oysters, so that a higher percentage of oyster stock could survive the 2-5 years of farming and cultivation [9].

The current manual tracking of pearl oyster growth rates is not scalable or cost-effective for large farming operations and selective breeding programs since additional effort is required to measure sizes. An automatic solution to track the growth rates using cameras would allow for a non-invasive measuring system to be implemented in larger farms to replace the manual process.

Existing automatic sizing systems often use equipment which is expensive and/or difficult to deploy into aquaculture farming environments. An example can be seen in the large and complex automatic oyster grading machines design by ShellQuip [10]. Capture technology in existing systems such as X-Ray, NIR- cameras and lasers [11]–[13] are expensive, with near infer-red (NIR) cameras and lasers ranging from US$1,000 - US$5,000 [14]. A simple and cost-effective solution for image capture, in close proximity to water, is to use action cameras, which are water-resistant/waterproof cameras that can capture high-resolution images and do not require a specially trained operator [15]. The use of action cameras can be combined with image processing to conduct oyster measurements automatically from the captured images.

In this study, we proposed an automatic solution for pearl oyster growth rate measurement. In the next section, we review some of the methods and research that has been utilised within this field previously.

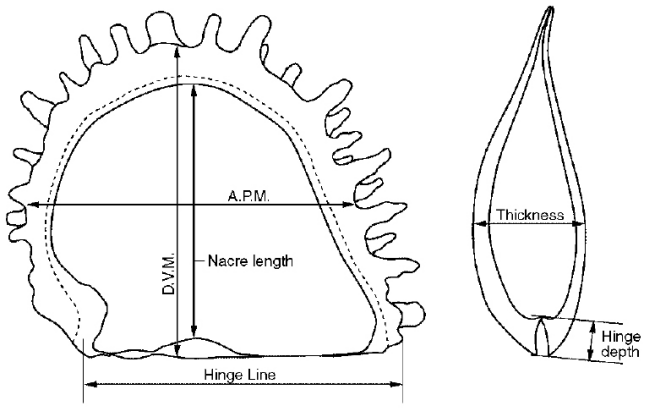

Fig. 1: Morphometric measurements for *Pinctada maxima* [16].

II. Background and Motivation

A. Pearl Oyster Shell Measurements

Pinctada maxima oyster has strongly correlated morphological measurements (Fig. 1). The three key measurements of the *Pinctada maxima* oyster are the doro-ventral measurement (DVM), anterior-posterior measurement (APM), thickness, and hinge depth (Fig. 1). DVM is the most common size measurement of an oyster [16], which has a high correlation ($r^2 > 0.95$) to both the APM and thickness measurements. APM can be calculated by

$$\text{APM} = 0.84 \times \text{DVM} + 14.9, \quad r^2 = 0.96, \tag{1}$$

similarly, thickness is calculated by

$$\text{Thickness} = 0.23 \times \text{DVM} - 1.89, \quad r^2 = 0.95, \tag{2}$$

with DVM in millimetres [16]. The existence of these high correlations allows for only one measurement to be extracted from an oyster to obtain all other key measurements, which may reduce the complexity of automatic size calculation using computer vision. However, to compute the size of an oyster, we must first identify the actual oyster itself, using computer vision and object detection.

B. Computer Vision and Object Detection

Object detection could be achieved through a computer vision system by processing images via segmentation, edge detection and/or feature extraction. For this study, edge detection was the primary technique used, where it could be defined as the process of identifying sharp discontinuities in an image [17]–[19]. The discontinuities are abrupt changes in pixel intensity. The identified discontinuities are then referred to as edges. Edge detection methods are generally grouped into two categories, gradient-based methods and Laplacian-based methods:

- *Gradient-based methods*: Gradient-based edge detection is accomplished by taking the first-order derivative of an image. Different operators known as masks are applied to the images to calculate a gradient magnitude for an image signal. The magnitude of the gradient is then used to calculate edge strength.
- *Laplacian-based methods*: Laplacian-based edge detection is completed by searching for zero crossings in the second derivative of an image signal to find edges [18].

We utilised both methods in this study to determine the best approach for pearl oysters.

C. Measuring Object Sizes in Aquaculture

In this study, the focus was on obtaining the required measurements (Fig. 1) from singular images. This was achieved by calculating a pixel size ratio using an object of known size within a scene or by using the distance of an object and re-adjusting for lens distortion [20], [21]. Once a pixel size ratio was calculated, systems could then count the number of pixels between points within an image to return an estimated measurement.

The majority of studies in this area focusing on aquaculture tend to focus on the size extraction for fish [22]–[25], however, the same techniques still apply to extract size of pearl oysters. Furthermore, if the oysters are identified and localized correctly, the actual measurement extraction could be easily performed using standard software tools, e.g. with OpenCV computer vision library [26]. Therefore, the main contribution of this work was in developing a reliable oyster segmentation technique.

D. Computer Vision for Automatic Oyster Processing

Automatic oyster sizing equipment does exist but it is primarily focused on oysters consumed for food and not pearl oysters, which are generally larger [16]. The majority of the world's oysters are consumed as food [1]. As such advancements in automatic oyster size measurement are limited to post-farmed oysters for accurate oyster grading. Oysters produced for food are generally sold by size, weight and volume as which they can be separated into 3 main grades: small, medium and large [17]. Automatic oyster grading machines are produced commercially but the measuring techniques are not publicly available as machine design is considered a trade secret.

The commercial grading equipment requires the use of a vision chamber or a controlled illumination environment. It would be difficult to deploy a large vision chamber on small pearling boats. In addition, since the algorithms/techniques for detecting and sizing the oysters are undisclosed, the equipment may not be able to detect and calculate the measurements of each oyster when there are multiple oysters in one image.

Even the non-commercial systems within the literature [11], [13] tend to focus on both post-farmed oysters and processing singular oysters through a controlled lighting environment. Furthermore, there are no applications of using newer image processing capabilities such as feature detection, convolutional

layers or modern edge detection on oysters to calculate the size of a given oyster. There has also been no exploration in applying computer vision to oysters during the farming process, where multiple oysters are captured in one image. Our study attempted to rectify this situation, as well as produce a practical system that could save both time and resources for pearl oyster farmers and producers.

III. Materials and methods

In this section, the experimental design, algorithms, and datasets are explained. As an overview, our experiment was divided into two stages. We started by utilising a small labelled dataset in a pilot study. Using this initial dataset we developed and then evaluated the accuracy of an algorithm for detecting and then measuring the size of an oyster from an image. Following this, we applied the developed algorithm to a larger dataset provided by a commercial pearl oyster farm. The reason for this split methodology was due to the unavailability of labelled (i.e. measured) data. By first developing our algorithm on the pilot study and then testing on the larger data set we were able to have more confidence in our results, even with the relatively small sample size used in the pilot study.

A. Datasets

1) Validation Dataset: The validation dataset used for the pilot study, consisted of photographs of 12 physical oyster shells. Images of oysters were captured using a GoPro Hero 4 Silver from a fixed orientation and height above a wooden surface. Each oyster shell was oriented with the hinge line perpendicular to the width of the camera sensor. The images were captured in an outdoors environment during the day to replicate the capture setup from the farm images as close as possible. To assist in algorithm calculations, a visible tape measure was also placed next to the oyster shells. The tape measure was planned to be used to calibrate the pixel value to the actual value ratio for the algorithms measurement calculations. The DVM and APM measurements of each oyster were also manually recorded with a ruler after images were captured.

2) Pearl Oyster Images: This data set consisted of 166 images, where each image contained 16 individual oysters. A custom image capture table was deployed on oyster cleaning vessels to gather image data for this research. The table consisted of a flat wood top with fixed GoPro Hero 4 camera1 mounted directly above the centre of the table. The distance and orientation between the camera in relation to the table remained constant for each captured image. Each image captured in the farm dataset contained one standard oyster net with sixteen pockets, holding one oyster each. Each oyster was oriented with the hinge line perpendicular to the camera sensors width. The net was placed on the table immediately after being cleaned to have the image captured, see an example in Fig. 2.

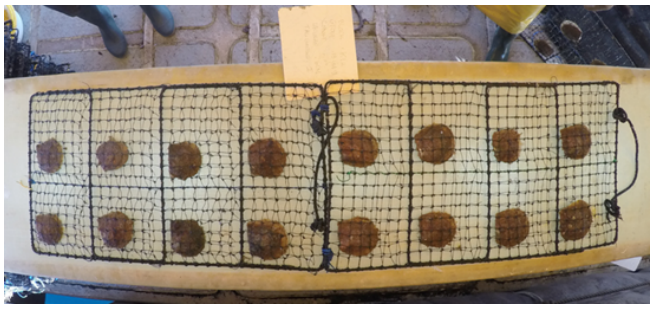

Fig. 2: An example of *Pinctada maxima* nets during cleaning [16]

B. Algorithm

Due to the lack of research conducted in this area, a new sizing algorithm was created, specifically for this study. The sizing algorithm consisted of the following eight steps:

1) Lens Correction: The fisheye lenses used in the experiment distort the captured image in order to achieve a greater field of view, however, this resulted in objects within images being distorted from the true shape and size of the object. To rectify this, an image correction library, Lensfun [27], was used. The Lensfun library allowed for camera and lens configurations to be entered and images to be corrected by using pre-calibrated image matrixes, stored in the Lensfun database [27].

2) Image Blur: Gaussian blur was applied to the images to assist in the reduction of minor features during the edge detection processes. Blurring the images causes the net covering each oyster to blend into the oyster shell, thereby assisting the edge detection phases to retrieve the shell outline. Each image was processed by a 5x5 Gaussian blur implementation of OpenCV with the results then passed to the rotation correction sub-routine.

3) Rotation Correction: Rotation correction was used to position the image with the oyster hinge lines horizontal in the image. By orienting the images this way, the measurement algorithm could be designed to make one pass over the pixels vertically and one pass over the pixels horizontally to retrieve the measurements. To orient the image correctly the height and width of the images were retrieved. These values were compared, and if height was larger than width the image was rotated 90 degrees, which oriented the oysters within the image correctly for the algorithm to proceed.

4) Regions of Interest: Region of interest was added to the algorithm to filter out unnecessary data from the images. Due to the static and similar nature of each of the images, region of interest selection was achieved by simply cropping each image appropriately. In addition to this, each image was also cropped to produce individual images of each oyster (again using static cropping) to assist in edge detection.

5) Edge Detection: To determine the most appropriate and accurate method of edge detection the following techniques were used:

- *Canny edge detection*: The first edge detection method implemented was Canny [26]. This was implemented using the OpenCV python library and the thresholds that were used for the detection were a lower bound of 10 and an upper bound of 100.
- *Convolutional layer edge detection*: Convolutional (Conv) layers were also tested as edge detectors for the algorithm via Tensorflow [28]. The decision to use convolution layers was made to reduce computational expense, as opposed to using a fully convolutional neural network. Three kernel sizes were tested in addition to the number of layers and pooling sizes: 3×3 kernel, a 5×5 kernel and a 7×7 kernel.

6) Filtering Edges: Due to the nature of the images, edges within the oyster shells were not as strong as the edges around the oyster shells and the net. Therefore, these 'weaker values' needed to be filtered out. This algorithm took each pixel value from the edges array returned during the convolution process and converted it to either a 1 or 0 if the edge value was less than the specified threshold (0.04 for our purposes). This threshold needed to be chosen carefully to reduce the amount of noise in the edges detected inside the oyster shell area, without compromising on the edges of the outer oyster shells.

7) Pixel Counting: A pixel counting algorithm was developed for this study. The purpose of the algorithm was to measure a number of pixels between edges. The pixel counting algorithm made a horizontal pixel pass and a vertical pixel pass over the input edge array. The horizontal pixel pass processed each pixel value across the horizontal dimension of an image from left to right, moving down to the next row of pixels with each loop. The vertical pixel pass processed each pixel down the vertical dimension of the image from top to bottom, moving across to the next column of pixel values with each loop.

The algorithm was designed to count the distance between edges, the logical structure was as follows. The algorithm processed an array of edge values and sets a starting position for a measurement when an active pixel was found (pixel value $= 1$). The algorithm then reviewed the next pixel in the array for a value. If the pixel was still active, the starting position of the measurement was then moved to that pixel. If the pixel was not active (pixel value $= 0$) the algorithm counted the number of pixels until the next edge was found. The distance between these edges was then returned as a potential measurement.

To remove the oyster net from the potential measurement, a threshold was implemented to distinguish the pixel distances of an oyster shell and the distances between the squares of the net. Each distance between edges found in the pixel counting process was compared directly to this threshold to determine if the potential measurement should be kept or not. The pixel distances between the edges of the square nets in each image were always smaller than the distances between the edges of an oyster, which was why suitably picking the threshold value allows for the easy distinguishing of oyster and net edge distances. Finally, the algorithm then took the maximum distance found in the horizontal and vertical pass and return the number of pixels and the location of the max distance within the image. The maximum distance was returned as this gave the DVM or APM of each oyster.

8) Measurements: The final stage of the algorithm was converting the pixel measurements into quantifiable measurements. To achieve this the pixel count of each detected measurement was multiplied by a pixel size to actual size ratio value. The ratio value for this was determined by running the edge detection algorithm over the length of tape exposed in the validation oyster images.

Code and parameters for the architectures used in this experiment were made available[1].

C. Pilot Study

Pilot testing was completed by capturing an image of the oyster shells provided and running them through the algorithm. The algorithm was run individually using 3×3 Conv, 5×5 Conv, 7×7 Conv and Canny edge detection methods. The sizing algorithm was calibrated manually using a tape measure placed next to the oysters. Calibration was done by measuring 100 pixels along the tape measure on the output image and recording how many centimetres the measurement equated to on the tape measure. Once the algorithm had been calibrated the measurement accuracy was then calculated.

To calculate measurement accuracy, the exact measurement lines from the algorithm's outputs were then directly measured and compared on each oyster shell with a ruler. The accuracy results from the pilot testing were used in combination with the reliability results from algorithm testing on the larger oyster farm dataset.

D. Experiment Design for Testing

After pilot tests were completed, the larger oyster farm dataset was used to determine the reliability of the algorithm with each edge detection method. All images were run through the sizing algorithm using 3x3 Conv, 5x5 Conv and Canny edge detection methods. Due to the lack of access to the physical measurements of each oyster, testing of reliability and accuracy was conducted using a combination of visual sorting and statistical analysis on the outputs.

For visual sorting, each output measurement was reviewed manually on a per oyster basis. The result was sorted into one of four categories, both DVM and APM identified, only DVM identified, only APM identified or neither measurement identified. The threshold for a measurement being classed as identified, was for the measurement to appear over the oyster shell. Finally, the percentage of occurrence for each identification was recorded as the reliability result for each edge detection algorithm run. The measurements from each test were then used in statistical analysis to derive any further correlations and accuracy findings. Regression analysis was the statistical analysis method run to review the data.

[1]https://github.com/Adroso/OysterHonours2018

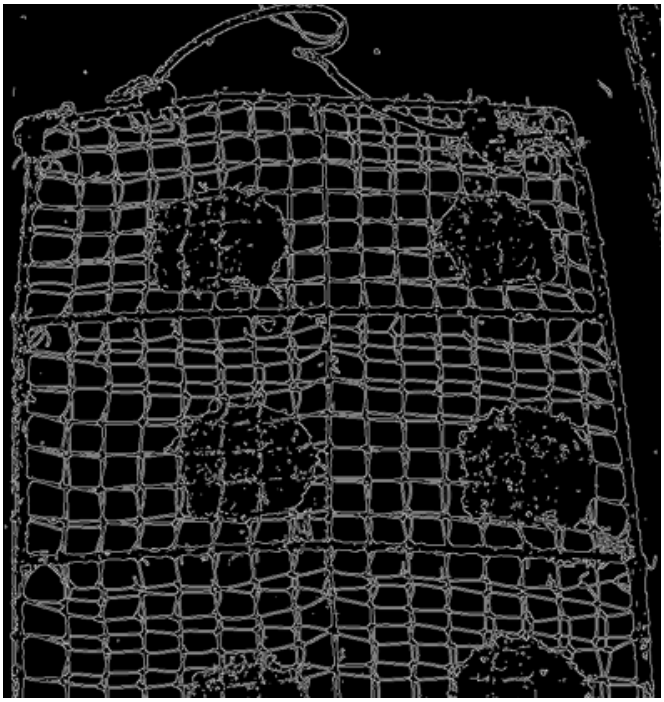

Fig. 3: Canny edge detection.

IV. Results

In this section we go over the results of our algorithm, we start by briefly summarising the outcome of each step of the algorithm and then move onto the specific performance of our overall process. Steps 1 - 3 (*Lens Correction*, *Image Blur*, *Rotation Correction*) of the algorithm were simple pre-processing steps, of which there were no concrete results. Rather, a simple visual inspection confirmed that each of these processes performed as intended. Step 4, *Regions of Interest*, had a 99.9% reliability of segmenting each oyster into its own region. Only one image failed to have the oysters segmented properly, which was due to the net being misplaced on the table.

In step 5, *Canny Edge Detection* (Fig. 3) performed well but, using this technique resulted in a large number of net edges present over the oyster shell areas. To determine the most effective convolution layer parameters to use for the final experiments, 3×3, 5×5 and 7×7 kernel sizes were tested and max-pooling layers were introduced to further enhance the edges. The size of max-pooling layers was first tested on a 3×3 convolution layer, to determine the most effective max-pooling size in reducing the net edges to use for the algorithm. The results in Fig. 4 show that the 2×2 max-pooling layer was the most effective in reducing the amount of net edges over the oyster shell, with the 3×3 max-pooling layer and no max-pooling layer still identifying net edges over the oyster shells, this is shown in Fig. 4. Thus the optimal parameters were found to be 2×2 max-pooling, with one convolution layer and one max-pooling layer. However, the 7×7 kernel resulted in images without edges and was disregarded from further testing.

TABLE I: Summary of Results

Edge Detector	3×3 **Conv**	5×5 **Conv**	**Canny**
No. of Images	166		
No. of Segmented Images	2656		
No. of Blank Oysters	133		
DVM and APM Identified (%)	50.1%	44.7%	18.6%
DVM or AVM Identified (%)	92.1%	-	-
Regression Formula $D := DVM$	$0.69 \times D + 22.4$	$0.43 \times D + 34.79$	$0.510 \times D + 47.11$
Regression r^2	0.44	0.23	0.24
Regression Gradient	1.45	2.33	1.96

The filtering utilised in step 6, was shown to have a positive effect in reducing the unnecessary features (i.e. the net covering the oyster shells), however fine-tuning was needed to find the best threshold value for the datasets (for our purposes this was a threshold of 0.04). Below we review the performance of the algorithm, in regards to the two datasets used in the experiment.

A. Pilot Results

The sizing algorithm performed well in accuracy and reliability in the pilot test. The highest performing edge detection method was the 5×5 Conv kernel with a measurement accuracy of 99.97%. Both Canny and the 3×3 Conv kernel were tested to have an average measurement accuracy of 99.8% and 98.5%, respectively.

B. Farm Image Test Results

The algorithm results from the oyster farm image set were used to test the reliability of the algorithm, with 3×3 Conv edge detection resulting in the most reliable measurements. The test compared 3x3 Conv, 5x5 Conv and Canny edge detection with the same dataset.

The experiments showed 3×3 Conv edge detection had reliability of 92.1% in identifying at least DVM or APM of an oyster. The 3×3 Conv algorithm was also seen to have the highest reliability in identifying both APM and DVM of an oyster. When only one measurement was identified with 3×3 Conv, DVM was the most reliable with an 80.2% reliability, as opposed to APM identification with a 19.8% reliability. Finally, 3×3 Conv showed the highest r^2 result in the regression analysis. In fact, the other edge detection methods did not perform as well as 3×3 Conv in any evaluation category.

The 5×5 Conv was seen to have a 44.6% reliability in identifying both APM and DVM from an oyster and Canny resulted in an 18.6% reliability. The 5×5 Conv shows to be more reliable than Canny, however, the regression results show that 5×5 may have lower accuracy than Canny. An overall summary of the results can be seen in Table I.

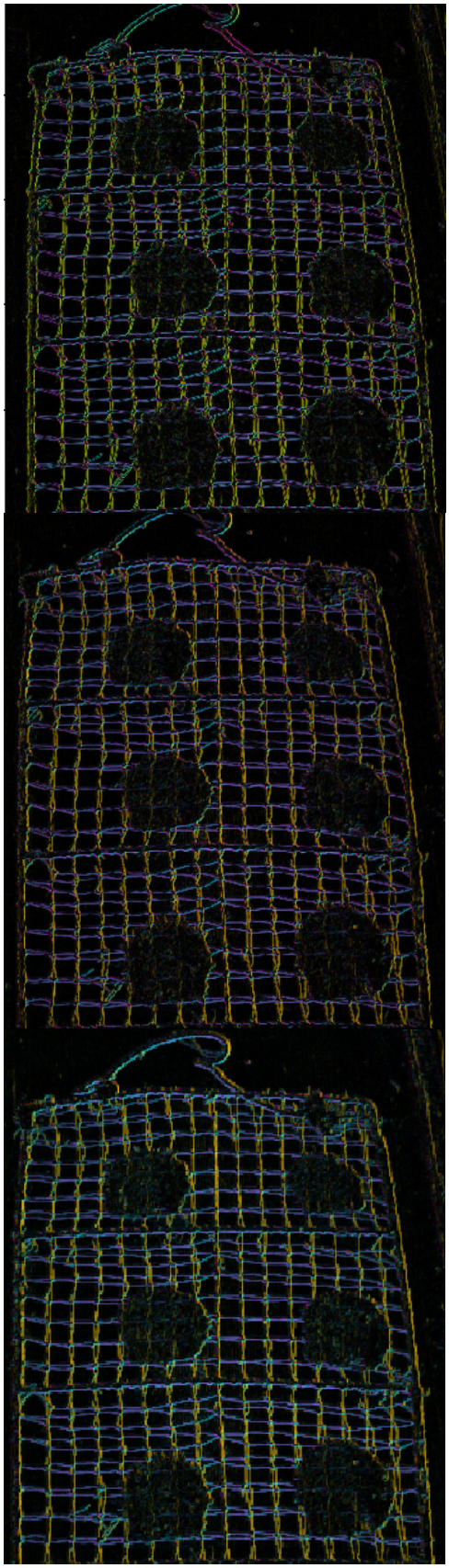

Fig. 4: From top, 3×3 Conv with: no max pool, 2×2 max pool and 3×3 max pool.

V. Discussion

As can be seen from the results, our algorithm was able to identify at least one of either the DVM or APM of an oyster with a reliability of 92.1%. This section discusses the implications of the results and gives a detailed comparison between the edge detection methods used, based on their performance metrics. Additionally, the limitations of the proposed algorithm are discussed. Finally, the overall reliability of the algorithm was investigated, highlighting key benefits for the use of the proposed algorithm to replace the current manual work-flow for measuring pearl oyster growth rate.

A. Edge Detection Method Comparison

The accuracy testing conducted in the pilot test showed that the 5×5 Conv kernel was the most accurate, followed by the 3×3 Conv kernel and Canny edge detection. Furthermore, the 5×5 Conv kernel also identified the most features, followed by the Canny edge detection and the 3×3 Conv kernel. Although this seemed to indicate that the 5×5 kernel was the most optimal choice for our algorithm, we argue that this is not the case.

There are two key points that support this argument. Firstly, the pilot test was conducted on images that did not feature a net. As such, the high feature identification of the 5×5 Conv would actually result in a lot of the net being picked up as features/edges. Secondly, the pixel counting algorithm relied on fewer features being detected within the boundaries of the oyster shell, to count the longest uninterrupted distance between the boundaries. As the 3×3 kernel identifies the least features, this would make it optimal for the pixel counting algorithm. A fact that was reflected in the results of the farm test data.

During the oyster farm image test, the reliability of each edge detection algorithm was evaluated using the reliability of identifying both DVM and APM of each oyster. As seen in the results section, the 3×3 Conv was found to have the highest overall reliability. The OpenCV library executed the Canny edge detection method on a CPU as opposed to the graphical processing unit (GPU). We note, however, that execution time was not a relevant criterion for evaluating the performance of our algorithm. This is because the entire algorithm can be run offline (i.e. not on the actual farm itself) and thus, will not affect the day to day operations of a given pearl farm. Instead, reliability should be the key metric by which we evaluate the algorithm.

Overall, it was very clear that the 3×3 Conv version of the sizing algorithm was the most reliable in the oyster farm image test. This reliability result of 50.1% for identifying both DVM and APM (and roughly 92% for identifying either the DVM or APM) was due to the implementation of the filtering algorithm. The filtering algorithm removed unnecessary edges after the 3x3 Conv edge detection outputs and this, combined with the 3x3 Convs lower edge feature detection further assisted in removing the net edges before the pixel counting algorithm was used.

B. Pixel Counting Algorithm Performance

The pixel counting algorithm was affected by some unexpected features within the image, which lowered the algorithm's reliability. The first unexpected feature was the presence of non-net areas. Specifically, the pixel counting algorithm sometimes identified the DVM to be the length of the table that was present after segmentation. The incorrect identification was not due to an error of the pixel counting algorithm. The algorithm functioned normally to identify the longest distance of missing pixels between edges. This identification issue could be reduced with the use of an object detection algorithm to initially segment the image before continuing to the edge detection and pixel counting algorithms.

The second drawback of the pixel counting algorithm was the incorrect identification of the thicker sections of the net as edges to count. The algorithm often identified the DVM and/or APM as the thicker part of the oyster net. This was due to the thicker sections of the net including an area between its edges were the pixel counting algorithm could identify a long string of empty pixels and identify them as the maximum measurement within the image. The effects of the drawback would again be improved with the implementation of an object detection algorithm during the segmentation process. Despite the drawbacks of the pixel counting algorithm, the overall sizing algorithm achieved a high reliability when identifying at least one measurement from the oyster.

C. Algorithm Reliability

Although the algorithm had a somewhat low probability of identifying both the DVM and APM, it could identify at least one of these features with an much higher probability (over 92%). Identifying both DVM and APM would be ideal, however, we note that the measurement of pearl oyster growth rate is still viable with only one of these features identified. This is because a high correlation was identified between the DVM and APM of *Pinctada maxima* oysters [16]. With the existence of this correlation, an accurate estimate can be made to the length of a missing measurement of an oyster. The estimated measurement can then be used to determine if the oyster is still growing with only one measurement identified. The correlations identified by [16] were also used to validate the accuracy of the 3×3 Conv algorithm.

The accuracy of the 3×3 Conv algorithm was originally not possible to retrieve due to the missing measurement data. However, by using the correlations identified by [16] a regression comparison was completed comparing 3×3 Conv to the regression formula identified by [16]. A regression line and gradient value were calculated for the 3×3 Conv algorithm using a DVM value range of 50mm-190mm. The range was determined from the results of the test as expected sizes of a pearl oyster. The 3×3 Conv algorithm showed a moderate correlation identified by the regression fit:

$$\text{APM} = 0.69 \times \text{DVM} + 22.24, \quad r^2 = 0.44, \tag{3}$$

which was comparable to the Eq. 1 reported by Hart *et al.* [16].

As per Table I, the calculated gradient of the 3×3 Conv was 1.45 and the gradient of the regression identified in [16] was 1.19. This was a comparable result which validated the results and reliability of our algorithm when used with the 3×3 architecture. We note that the gap between the lines can be attributed to the measurement algorithm being incorrectly calibrated for accurate measurements. If the distance between the camera and table would have been known for this research, a more accurate size could be retrieved from the computer vision algorithm and matching the sizes of the regression identified by [16].

During the experiments, further benefits for using a computer vision workflow to replace the current manual workflow for measuring pearl oyster sizes were identified. The most obvious benefit was that processing speed of the computer vision algorithm was faster than a manual workflow. The use of a computer vision algorithm would also increase the consistency of the correct measurement being identified.

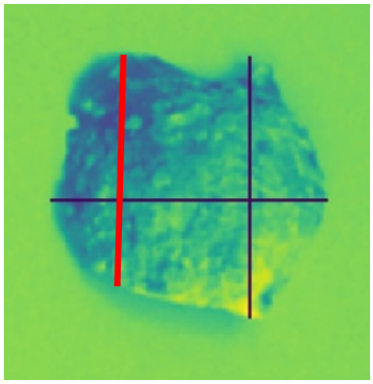

Fig. 5: True maximum distance in oyster measurement.

To calculate the DVM or APM of a pearl oyster, the maximum distance between the respective edges of the shell was calculated. In some cases, the true maximum distance was difficult to identify by a human consistently and accurately. This was in addition to intra-observer variability, where a human measuring the same object repetitively will have slightly different measurements each time. The automatic sizing algorithm created for this research overcomes the issues with identifying the true maximum distance between shell edges.

An example is illustrated in Fig. 5, where measuring the DVM of the oyster, a person may gravitate to measuring the distance from the protrusion on the left side of the shell (highlighted in red). The automatic sizing algorithm disagrees with this and identifies the true maximum distance as shown. The one example given here will happen multiple times when reviewing millions of oysters annually. If a manual process is kept, the accuracy and reliability of the measurements will not be consistent. The factors of intra-observer variability and human fatigue will result in a lower consistency in measurement, which can be avoided with the implementation of this automatic sizing algorithm.

VI. Conclusion

A new automatic algorithm was created and tested on two data sets of pearl oyster images to determine the most reliable and accurate technique to measure each oyster from the images. The new algorithm produced test-reliability and pilot-accuracy results of 92.1% and 98.5% respectively. The algorithm was made freely available to be modified in future to improve the accuracy and reliability results further.

Specifically, the accuracy and reliability of the pixel counting algorithm can be improved by implementing an object detection algorithm to assist in the image segmentation process. The mis-identification of the DVM and APM measurements by the pixel counting algorithm was due to the inclusions of non-oyster net areas and the thicker areas of the oyster net. If a segmentation technique could be implemented to accurately identify an oyster within the scene and then process the segmented oyster through the sizing algorithm, the accuracy and reliability of identifying both DVM and APM should increase.

References

[1] FIAS, "FAO year book: fishery and aquaculture statistics 2015," *FAO, Rome*, 2017.

[2] M. Vandeputte, "Selective breeding – lessons learned from terrestrial animals and status of aquaculture implementation," in *Aquaculture Europe 2011 – Mediterranean Aquaculture 2020*. European Aquaculture Society, 2011.

[3] I. Atkinson, "Hi-tech revolution for livestock," Mar 2018. [Online]. Available: https://bit.ly/2I3FKbn

[4] J. Salau, J. H. Haas, W. Junge, U. Bauer, J. Harms, and S. Bieletzki, "Feasibility of automated body trait determination using the sr4k time-of-flight camera in cow barns," *SpringerPlus*, vol. 3, p. 225, 2014.

[5] E. Davies, "The application of machine vision to food and agriculture: a review," *The Imaging Science Journal*, vol. 57, pp. 197–217, 2009.

[6] D. Konovalov, J. Domingos, R. White, and D. Jerry, "Automatic scaling of fish images," in *Proceedings of the 2nd International Conference on Advances in Image Processing*. ACM, 2018, pp. 48–53.

[7] J. J. U. Taylor, "Juvenile production and culture of the silver-lip pearl oyster, Pinctada maxima (Jameson)," Ph.D. dissertation, James Cook University, 1999. [Online]. Available: https://bit.ly/2m1zD1z

[8] M. Albuquerque, R. Alves, A. Zanandrea, J. Ferreira, C. Melo, and A. Magalhães, "Growth and survival of the pearl oyster pteria hirundo (l., 1758) in an intermediate stage of culture in santa catarina, brazil," *Brazilian Journal of Biology*, vol. 72, pp. 175–180, 2012.

[9] "White south sea pearl information," 2018. [Online]. Available: https://bit.ly/32ft8J3

[10] "Oyster grading technology," 2018. [Online]. Available: https://bit.ly/2m0CszP

[11] D.-J. Lee, X. Xu, R. M. Lane, and P. Zhan, "Shape analysis for an automatic oyster grading system," in *Two-and Three-Dimensional Vision Systems for Inspection, Control, and Metrology II*, vol. 5606. International Society for Optics and Photonics, 2004, pp. 27–37.

[12] J. Li and F. W. Wheaton, "Image processing and pattern recognition for oyster hinge line detection," *Aquacultural engineering*, vol. 11, pp. 231–250, 1992.

[13] D.-J. Lee, R. M. Lane, and G.-H. Chang, "Three-dimensional reconstruction for high-speed volume measurement," in *Machine Vision and Three-Dimensional Imaging Systems for Inspection and Metrology*, vol. 4189. International Society for Optics and Photonics, 2001, pp. 258–268.

[14] "NIR and UV cameras," 2018. [Online]. Available: https://bit.ly/2G7gIcP

[15] "Hero4 specifications," 2018. [Online]. Available: https://bit.ly/2xITNzQ

[16] A. Hart, K. L. Travaille, R. Jones, S. Brand-Gardner, F. Webster, A. Irving, and A. V. Harry, *Western Australian silver-lipped pearl oyster (pinctada maxima) industry*. Goverment of Western Australia, Department of Fisheries, Perth, 2016. [Online]. Available: https://bit.ly/2YNWyeX

[17] 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, 2014.

[18] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International journal of image processing*, vol. 3, pp. 1–12, 2008.

[19] J. Beyerer, F. P. León, and C. Frese, *Machine vision: Automated visual inspection: Theory, practice and applications*. Springer, 2015.

[20] T. Breuer, M. M. Robbins, and C. Boesch, "Using photogrammetry and color scoring to assess sexual dimorphism in wild western gorillas (gorilla gorilla)," *American Journal of Physical Anthropology*, vol. 134, no. 3, pp. 369–382, 2007.

[21] P. Drap and J. Lefèvre, "An exact formula for calculating inverse radial lens distortions," *Sensors*, vol. 16, no. 6, p. 807, 2016.

[22] D. Konovalov, J. Domingos, C. Bajema, R. White, and D. Jerry, "Ruler detection for automatic scaling of fish images," in *Proceedings of the International Conference on Advances in Image Processing*. ACM, 2017, pp. 90–95.

[23] F. Odone, E. Trucco, and A. Verri, "A trainable system for grading fish from images," *Applied Artificial Intelligence*, vol. 15, no. 8, pp. 735–745, 2001.

[24] B. Zion, V. Alchanatis, V. Ostrovsky, A. Barki, and I. Karplus, "Real-time underwater sorting of edible fish species," *Computers and Electronics in Agriculture*, vol. 56, no. 1, pp. 34–45, 2007.

[25] M. R. M. Shafry, A. Rehman, R. Kumoi, N. Abdullah, and T. Saba, "Filedi framework for measuring fish length from digital images," *International Journal of Physical Sciences*, vol. 7, no. 4, pp. 607–618, 2012.

[26] A. Rosebrock, "Measuring size of objects in an image with OpenCV." [Online]. Available: https://bit.ly/2H2Lvdc

[27] M. Riechert, "Lensfunpy," 2018. [Online]. Available: https://github.com/letmaik/lensfunpy

[28] "Tensorflow," 2018. [Online]. Available: https://www.tensorflow.org/