Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques

By

Chong Xiao Wei

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques

Academic Session: JANUARY 2024

I CHONG XIAO WEI

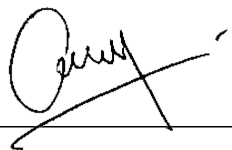
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)



(Supervisor's signature)

Address:
530, Jalan Pahang,
Kampung Chui Chak
36700 Langkap, Perak.

Ts Dr Cheng Wai Khuen
Supervisor's name

Date: 22/04/2024

Date: 22/04/2024

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 22/04/2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***CHONG XIAO WEI*** (ID No: **20ACB03300**) has completed this final year project entitled " Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques " under the supervision of Ts Dr Cheng Wai Khuen (Supervisor) from the Department of Computer and Communication Technology , Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(*CHONG XIAO WEI*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :

Name : Chong Xiao Wei

Date : 22/4/2024

ACKNOWLEDGEMENTS

I would like to express thanks and appreciation to my supervisor, Dr. Cheng Wai Khuen and my moderator, Dr. Tan Joi San who have given me me this opportunity to engage in an Artificial Intelligent field project. I believe that this project will build up my career profile in the future. Again, a million thanks to my supervisor and moderator.

Finally, I must say thanks to my family and my friend for their love, support, and continuous encouragement throughout the course, especially Joseph who helped me a lot.

ABSTRACT

Prawn cultivation is a crucial aquaculture industry, but it faces significant challenges related to inefficient feeding practices and lack of accurate population monitoring. Overfeeding due to imprecise population estimates leads to wasted resources and potential environmental issues. Additionally, traditional methods of monitoring prawn growth and well-being in underwater environments are labor-intensive and prone to inconsistencies, hindering timely decision-making processes.

To address these challenges, this project proposes an innovative solution that leverages computer vision and machine learning techniques. By employing the state-of-the-art You Only Look Once (YOLO) v7 object detection algorithm, the project aims to develop a system capable of accurately detecting and classifying prawns based on their growth stages. The detected prawns are then measured, and their lengths are used to estimate their weights and categorize them into juvenile, subadult, or adult stages. Furthermore, the project automates the estimation of prawn density and population within the aquaculture system, providing farmers with valuable insights into the population dynamics. This automated approach eliminates the need for manual monitoring and enables more efficient resource allocation and management strategies.

By addressing the challenges, this study contributes to the advancement of precision aquaculture operations. The proposed solution offers a viable path towards sustainable and efficient prawn farming practices, optimizing resource utilization, minimizing environmental impact, and ultimately enhancing the profitability and sustainability of the prawn cultivation industry.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AP	Average Precision
BPNN	Back Propagation Neural Network
CSV	Comma-Separated Values
CNN	Convolutional Neural Network
DT	Decision Tree
DCNN	Deep Convolutional Neural Network
EAS	Edge-Assisted Segmentation
ELAN	Efficient Layer Aggregation Networks
FN	False Negatives
FP	False Positives
IoU	Intersection over Union
KNN	K-Nearest Neighbors
MAE	Mean Absolute Error
mAP	Mean Average Precision
MBE	Mean Bias Error
MPAE	Mean Percent Absolute Error
MSE	Mean Squared Error
MLR	Multiple Linear Regression
MCE-CNN	Multi-Scale Context Enhanced CNN
PCR	Principal Component Regression
RF	Random Forest
RELU	Rectified Linear Unit
ResNet	Residual Network
RMSE	Root Mean Square Error
STAC	Sepang Today Aquaculture Centre
SVM	Support Vector Machine
TN	True Negatives
TP	True Positives
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Problem Statement and Motivation

Aquaculture is the fastest growing food production sector in the world, contributing to global food security, poverty alleviation, and economic development. Among the various aquaculture species, prawns are one of the most popular and profitable, with a high demand and a wide range of markets. In the year 2020, exports of prawns in Malaysia were the highest among the fisheries and aquaculture products [1]. However, prawn cultivation also faces many challenges that affect its sustainability and efficiency.

Exports - Top 10 Products (Value)			
			USD
1st	Shrimps, prawns	frozen HS 0306.17	185 805 110
2nd	Fish, other than species in 0303	frozen; excluding fillets, livers and roes HS 0303.89	85 796 580
3rd	Fish	prepared or preserved HS 1604.20	85 544 280
4th	Squid, cuttlefish	frozen HS 0307.43	64 919 750
5th	Flours, meals, pellets. Unfit for human consumption	from fish or aquatic invertebrates HS 2301.20	55 712 710
6th	Fish, other than species in 1604	prepared or preserved; whole or in pieces HS 1604.19	31 563 040
7th	Fish, other than species in 0302	fresh or chilled; excluding fillets, livers and roes HS 0302.89	24 031 350
8th	Fish, other than species in 0304	frozen meat; excluding fillets, livers and roes HS 0304.99	21 967 940
9th	Sea cucumbers	dried, smoked, salted or in brine or smoked HS 0308.19	18 780 990
10th	Squid, cuttlefish	prepared or preserved	16 773 500

Figure 1.1 Top 10 Products for Fisheries and Aquaculture [1]

One of the problems is the overfeeding and underfeeding. Prawn farmers are dealing with inaccurate feeding practices, which is worsened by lack of precise population estimates. This will lead to a potential excess of feed which will then increases operating expenses and pollute the water and underfeed that will affect the growth of the prawns. Another significant problem faced by farmers is the difficulty in obtaining accurate and up-to-date information about the growth and well-being of prawns within the underwater environment. Traditional methods rely heavily on human inspection and observation, which can be time-consuming and prone to inconsistencies. This lack of a systematic and automated approach to monitoring the prawn populations can impede decision-making processes and result in inefficiencies in resource allocation and farm management overall. Hence, an Artificial Intelligence (AI) based system is essential in solving the problems encountered in prawn cultivation, including feeding and real-time monitoring [2].

In conclusion, the problem statement in this paper is overfeeding due to lack of precise population estimates. Besides, prawn farmers difficult in obtaining accurate and up-to-date information about the growth of prawns within the underwater environment.

1.2 Objectives

The main objective of this proposed system is to improve precision aquaculture operations for prawn cultivation by using computer vision techniques to automate density and growth estimation. The proposed system uses computer vision techniques detect the prawns within aquaculture environment. Not only that, but the system is also designed to differentiate between the various growth stages of prawn, including Juvenile, Premature and Mature. This is the critical step for understanding the development dynamics of the prawn population, enabling farmers to make informed decisions about their cultivation strategies. Next it is focusing on employing machine learning techniques to automate the estimation process for both density and population of prawns. By automating these calculations through machine learning, the project aims to reduce the labour force and potential errors, offering a more efficient way to manage prawn cultivation. Finally, the project aimed to integrate the developed computer vision module into an existing prawn farm equipped with a precision aquaculture system. This step is crucial for the practical application of the technology within the existing aquaculture infrastructure.

As a summary for the objectives mentioned above, the project aimed to:

i. To use computer vision techniques, detect the prawn and differentiate between their various growth stages.

ii. Use machine learning techniques to automate the estimation process for density and population.
iii. To integrate the developed computer vision module into an existing prawn farm.

1.3 Project Scope

Based on the problem statement mentioned above, the prawn farming nowadays still utilizing more traditional method to cultivate the prawns. Traditional methods of estimating prawn density and population rely on manual sampling and counting, which are labour-intensive, time-consuming, and prone to errors [3]. In aquaculture industry, the AI technologies have yet been popularized to the existing prawn farmer. The prawn farmers can manage the farms even better by implementing the proposed system. This project will deliver a system which will automatically estimate the density and population of prawn using Computer Vision technique and Machine Learning approaches. This proposed system will only focus on the Giant Freshwater Prawn but not various type of prawns. The dataset of the prawns will be trained using You Only Look Once (YOLO) v7, then it will detect the prawn. The final output of the project will estimate the density and population of the prawns.

1.4 Contributions

The project will be beneficial for the aquaculture industry. First, this project can reduce the workload and improve the efficiency of the prawn farmers, who can use the system to automatically estimate the density and growth of prawns in the ponds, without having to catch the prawns from water or manually sample and count them. This can save time, labour, and resources for the farmers, and also reduce the stress and damage to the prawns. By knowing the density and growth of prawns in real-time, the farmers can make informed decisions about their feeding, harvesting, and marketing strategies, and optimize their production and income. The system can also help the farmers to prevent or mitigate the risks of overfeeding, underfeeding, disease outbreaks, environmental degradation, and market fluctuations, which can affect the quality and quantity of prawns.

Besides, this project can contribute to the advancement of science and technology in the field of aquaculture. The project can demonstrate the potential and applicability of computer vision and machine learning techniques for prawn cultivation, which is a complex and challenging task that requires high accuracy and reliability. The project can also provide a dataset of pond images with annotated prawns and their growth stages, which can be useful for future research and development.

1.5 Report Organization

The report is structured into seven chapters, each serving a specific purpose and providing a comprehensive understanding of the project. Chapter 1 introduces the project, highlighting the problem statement, motivations, objectives, scope, contributions, and the overall organization of the report. Chapter 2 delves into a literature review, exploring relevant technologies, algorithms, and methodologies employed in the project. Chapter 3 focuses on the system methodology, detailing the design, architecture, and evaluation metrics of the selected model, as well as outlining the system requirements. Chapter 4, titled "Preliminary Work," presents the project's workflow and the preparatory steps undertaken. In Chapter 5, the report shifts its focus to the system implementation, describing the hardware and software setup, and discussing the implementation issues and challenges encountered during the project. Chapter 6 is dedicated to system evaluation and discussion, providing an in-depth analysis of the project's performance, results, and insights gained from the evaluation process. Finally, Chapter 7 concludes the report by summarizing the project's findings, drawing conclusions, and offering recommendations for future work and potential improvements.

Chapter 2

Literature Review

2.1 Review of the Technologies

2.1.1 Shrimp Body Weight Estimation in Aquaculture Ponds Using Morphometric Features Based on Underwater Image Analysis and Machine Learning Approach [4]

Data Requirements and Preprocessing: The study delves into the essential data requirements and preprocessing steps necessary for estimating shrimp body weight in aquaculture ponds through underwater image analysis. Data collection involved capturing images of shrimps underwater in aquaculture ponds using a high-resolution camera. The collected video data was subsequently processed to extract individual image frames. To prepare the images for analysis, specific techniques were applied to convert the RGB images into grayscale and binary images. The Otsu algorithm was utilized for the conversion to binary images, ensuring efficient data preprocessing for further analysis.

Algorithm: The research explores the utilization of various algorithms, including Multiple Linear Regression (MLR), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Back Propagation Neural Network (BPNN), K-Nearest Neighbors (KNN), and Principal Component Regression (PCR), in estimating shrimp body weight based on morphometric features extracted from underwater images. Among these algorithms, MLR demonstrated superior performance in terms of accuracy, as evidenced by its ability to produce the lowest Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the highest R-squared value compared to other algorithms.

Performance Metrics: The study conducts an evaluation of algorithm performance using essential metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) to gauge the accuracy of shrimp body weight estimation. These metrics serve as valuable indicators of the predictive capabilities of the machine learning models utilized in the study. Specifically, the study focuses on assessing the effectiveness of the MLR algorithm in accurately estimating underwater shrimp body weight. This estimation is based on morphometric features extracted from the underwater images.

Comparative Analysis: A comparative analysis of the different algorithms used in the study reveals the strengths and weaknesses of each approach in estimating shrimp body weight. By comparing the performance metrics of MLR, DT, SVM, RF, BPNN, KNN, and PCR algorithms, the study identifies MLR as the most accurate algorithm for estimating shrimp body weight using morphometric features derived from underwater images. This comparative analysis sheds light on the effectiveness of machine learning techniques in accurately estimating shrimp body weight in aquaculture ponds.

Summary of the Technologies Review: The study offers a thorough overview of the technologies utilized to estimate shrimp body weight in aquaculture ponds through underwater image analysis and machine learning. It examines data prerequisites, preprocessing methods, algorithm choices, performance metrics, and conducts a comparative analysis of algorithms. The study underscores the importance of MLR in accurately estimating underwater shrimp body weight, highlighting key findings and insights in this context.

2.1.2 Computer Vision Based Estimation of Shrimp Population Density and Size [5]

Data Requirements and Preprocessing: The system leverages a dataset consisting of 100 shrimp images gathered from a controlled laboratory setting. For each image in the training set, binary masks are manually generated to delineate the shrimp from the background. To enhance the training dataset's diversity and improve model generalization, data augmentation techniques are applied.

Algorithm: The computer vision system integrates the U-Net segmentation technique, which is a fully convolutional network capable of assigning a class label to every pixel within an image. The U-Net architecture comprises 23 convolutional layers, encompassing both a contracting path and an expansive path. This architecture is specifically designed to segment individual shrimps from the background in images, making it suitable for precise object detection and classification tasks.

Performance Metrics: The proposed method is evaluated based on its ability to count shrimps and estimate shrimp lengths. The mean absolute error for counting shrimp is reported as 0.093, and the root mean square error for estimating shrimp lengths is 0.293 cm.

Comparative Analysis: The system performs well in counting shrimps when they are separately located, with a mean absolute error of 0.093. However, it faces challenges in accurately counting touching or overlapping shrimps, resulting in a higher mean absolute error of 0.298 in such cases.

Summary of the Technologies Review: The computer vision system presented in the paper utilizes U-Net segmentation and marker-controlled watershed segmentation techniques to estimate shrimp population density and size. While the system demonstrates good performance in counting and estimating lengths of separately located shrimps, improvements are needed for accurately handling touching or overlapping shrimps.

2.1.3 Fish Density Estimation with Multi-Scale Context Enhanced Convolutional Neural Network [6]

Data Requirements and Preprocessing: The research paper highlights the significance of data requirements and preprocessing in fish density estimation, particularly in the context of utilizing a multi-scale context enhanced convolutional neural network. It underscores the necessity of having high-quality labeled images of fish schools for effectively training and testing the algorithm.
The dataset utilized in the study, known as DlouDataset, comprises 300 images depicting high-density fish scenarios. These images are accompanied by 40,000 labeled instances of fish, providing a comprehensive dataset for evaluating the proposed algorithm's performance and effectiveness in fish density estimation tasks.

Algorithm: The paper presents a new convolutional neural network architecture specifically tailored to generate density maps of fish images. The network is designed with the goal of mitigating camera perspective effects and variations in image resolutions, ultimately enhancing the accuracy of fish counting tasks. A key feature of this network is the incorporation of multiple scale filters, which allow it to effectively handle different perspectives and resolutions present in the input images. Moreover, the network integrates a context enhancement module, which plays a crucial role in capturing relevant context information from the images. By doing so, it enhances the quality of the density map output, leading to improved performance in fish counting tasks compared to traditional methods.

Performance Metrics: In evaluating the performance of the proposed method, the paper utilizes performance metrics such as mean square error and mean absolute error. These metrics help assess the accuracy of fish counting based on the density maps generated by the convolutional neural network. The results demonstrate the effectiveness of the algorithm in estimating fish density.

Comparative Analysis: The research paper conducts a comparative analysis of existing methods for fish counting and density estimation. By comparing the proposed multi-scale context enhanced convolutional neural network with other approaches, the study highlights the advantages and improvements achieved in fish density estimation. The analysis provides insights into the strengths of the proposed algorithm in addressing challenges such as occlusion and perspective effects.

Summary of the Technologies Review: In summary, the review of technologies in fish density estimation with a multi-scale context enhanced convolutional neural network emphasizes the significance of data preprocessing, the innovative algorithm design, the use of performance metrics for evaluation, and the comparative analysis with existing methods. The integration of multi-scale filters and a context enhancement module enhances the accuracy of fish counting, as demonstrated by the results on the DlouDataset.

2.1.4 Deep Convolutional Neural Networks for Fish Weight Prediction from Images [7]

Data Requirements and Preprocessing: The authors used a dataset consisting of 259 Australasian snapper fish individuals, with a total of 529 images collected from these fish. The dataset was provided by The New Zealand Institute for Plant and Food Research Limited (PFR). The images were preprocessed by cropping out the ruler, padding to make a square image, resizing to 224x224 pixels, and randomly flipping horizontally and vertically for data augmentation. The images were also normalized using mean and standard deviation values for the RGB channels.

Algorithm: Three distinct deep convolutional neural network architectures to predict fish weight based on image data. These architectures included VGG-11 with Batch Normalization, which is a conventional deep CNN design comprising stacked convolutional layers. They also employed ResNet-18, a CNN architecture that addresses the vanishing gradient issue in deep networks by introducing residual blocks. Additionally, DenseNet-121, an advanced CNN architecture, was utilized, allowing feature maps from earlier layers to be transmitted to deeper layers, thus enhancing feature propagation and reducing model complexity. The models were trained using the Adam optimizer, mean squared error loss function, and a batch size of 32. The training was performed for 100 epochs on each cross-validation fold.

Performance Metrics: Performance evaluation in the study involves metrics such as R^2, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) to assess the accuracy of fish weight predictions. These metrics provide valuable insights into the effectiveness of the proposed algorithm in underwater image processing.

Comparative Analysis: The performance of the three CNN architectures (VGG-11, ResNet-18, and DenseNet-121) are compared on the fish weight prediction task. They used 5-fold cross-validation and reported the best and average performance metrics (MSE, RMSE, and R^2) for each architecture on the test sets. The DenseNet-121 architecture achieved the highest average R^2 of 0.96, outperforming ResNet-18 (0.95) and VGG-11 (0.94).

Summary of the Technologies Review: The dataset of Australasian snapper fish images provided by PFR is used in the paper. The images were preprocessed by cropping, padding, resizing, and data augmentation techniques. Three deep convolutional neural network architectures (VGG-11, ResNet-18, and DenseNet-121) were employed to train regression models for predicting fish weight directly from images. The models were trained using the Adam optimizer, mean squared error loss, and cross-validation. Performance was evaluated using metrics such as R-squared, mean squared error, and root mean squared error. The DenseNet-121 architecture achieved the highest average performance in predicting fish weight from images without the need for manual length measurements or a scale reference.

2.1.5 Applications of Deep Convolutional Neural Networks to Predict Length, Circumference, and Weight from Mostly Dewatered Images of Fish [8]

Data Requirements and Preprocessing: The dataset used in the paper was obtained from the FishL™ Recognition System, which employs six overhead cameras to capture a composite of 18 images (9 color and 9 near-infrared) as fish pass through the system. These composite images were further processed by extracting individual images from them. These extracted images were then resized to dimensions of 75 pixels by 200 pixels. These resized images served as the input data for the deep convolutional neural network (DCNN) regressors employed in the study.

Algorithm: The DCNN regressors were built with three groups of 2D convolutional layers followed by a 2D max pooling layer. The convolutional layers used a 3 by 3 filter size, and the max pool layers used a 2 by 2 filter size. Initially, there were 32 filters in the first convolutional layer, followed by 64 filters in the subsequent layers. The convolution outputs went through a single layer of 256 nodes with a rectified linear unit (relu) activation function, and the final network output was a real value indicating the predicted length, girth, or weight of the fish.

Performance Metrics: The regressors' performance was assessed using three metrics: mean absolute error (MAE), mean bias error (MBE), and mean percent absolute error (MPAE). MAE calculates the average absolute difference between predictions and their actual values. MBE determines the average difference between predictions and actual values. MPAE calculates the average absolute percentage difference between predictions and actual values, offering a more reliable measure of performance across different ranges of actual values.

Comparative Analysis: The study evaluated both single-target and multi-target regressors. The single target regressors were trained to predict only one of length, girth, or weight, while the multi-target regressor was trained to predict all three simultaneously. The ensemble predictions, which averaged the outputs of the nine color images for each fish, generally outperformed the single-image predictions across all metrics and target variables. The multi-target regressor also exhibited lower MBE compared to the single target regressors, indicating less bias in the predictions.

Summary of the Technologies Review: The study demonstrated the potential of DCNN regressors to predict the length, girth, and weight of a diverse set of fish species from dewatered images captured by the FishL™ Recognition System. The ensemble predictions achieved mean percent absolute errors of 7.6%, 16.8%, and 26.9% for length, girth, and weight, respectively, on the test dataset. The multi-target regressor further improved performance, particularly for length and girth. These results suggest that automated image-based biometric data collection could increase the efficiency and accuracy of routine fishery surveys and provide a means for long-term monitoring of fish populations.

2.2 Summary on Related Approaches

Paper	Preprocessing Steps	Model Used	Evaluation Metrics
2.1.1	Conversion of RGB images to grayscale and binary using Otsu algorithm	• Multiple Linear Regression (MLR) • Support Vector Machine (SVM) • Random Forest (RF) • Decision Tree (DT) • K-Nearest Neighbors (KNN) • Back Propagation Neural Network (BPNN) • Principal Component Regression (PCR)	• Root Mean Square Error (RMSE) • Mean Absolute Error (MAE) • R-squared
2.1.2	• Manually created binary masks for training images • Data augmentation techniques	U-Net segmentation method	• Mean absolute error • Root mean square error
2.1.3	High-quality labeled images of fish schools for training and testing	Multi-scale context enhanced convolutional neural network	• Mean square error • Mean absolute error

2.1.4	• Cropping out ruler • Padding to make square image. • Resizing to 224x224 pixels • Random horizontal and vertical flipping for data augmentation • Normalization using mean and standard deviation	• VGG-11 with Batch Normalization • ResNet-18 • DenseNet-121	• R-squared • Mean Squared Error (MSE) • Root Mean Squared Error (RMSE)
2.1.5	• Extraction of individual images from FishL™ Recognition System composites • Rescaling to 75x200 pixels	• Single target regressors • Multi-target regressor	• Mean Absolute Error (MAE) • Mean Bias Error (MBE) • Mean Percent Absolute Error (MPAE)

Table 2.1.1 Summary on Related Approaches

2.3 Algorithm Used in Previous Related Work

2.3.1 MLR

MLR is a statistical approach that models the connection among numerous independent variables and a dependent variable [9]. In the context of the study, MLR was employed to establish a linear relationship between morphometric features extracted from underwater images (such as shrimp body length, width, and segment height) and the corresponding shrimp body weight. The formula for MLR can be represented as: $Y=b_0+b_1X_1+b_2X_2+...+b_n X_n$ where:

- Y is the predicted shrimp body weight,
- b_0 is the intercept,
- $b_1,b_2,...,bn$ are the coefficients,
- $X_1,X_2,...,X_n$ are the morphometric features.

The MLR algorithm in the study involved training a model using the extracted morphometric features as input variables and the corresponding shrimp body weights as the target variable. The model was then used to predict the body weight of shrimp based on new morphometric feature inputs. The layers in the MLR algorithm represent the coefficients assigned to each input feature, which are optimized during the training process to minimize the error between the predicted and actual shrimp body weights. The algorithm's functionality can be visualized as a linear regression line that best fits the relationship between the morphometric features and shrimp body weight, allowing for accurate estimations in aquaculture ponds.

2.3.2 U-Net Segmentation

U-Net is a Convolutional Neural Network (CNN) designed specifically for biomedical image segmentation tasks, particularly in semantic image segmentation or pixel-based classification [10]. Its architecture can be divided into two main parts: the encoder network and the decoder network. The encoder network, often based on a pre-trained classification network like VGG or ResNet, processes the input image using convolution blocks and downsampling through maxpooling. This process extracts features at multiple levels of abstraction. On the other hand, the decoder aims to project the discriminative features learned by the encoder onto the pixel space to achieve dense classification It accomplishes this through upsampling, concatenation, and regular convolution operations [11].

In simpler terms, U-Net includes an expansive path that generates pixel classifications for features or objects identified in the original image. This path expands the output to match a specific image size and forms the latter part of the U shape in the network.

2.3.3 Multi-Scale Context Enhanced CNN

The Multi-Scale Context Enhanced CNN (MCE-CNN) is a specialized type of Convolutional Neural Network (CNN) tailored for intricate image segmentation tasks. It utilizes two weight-shared feature extraction networks, each comprising a ResNet and a Vision Transformer, to extract detailed features from both support and query images [12].

A key component of the MCE-CNN is the Multi-Scale Context Enhancement (MCE) module. This module merges the features extracted by the ResNet and Vision Transformer networks, leveraging cross-scale feature fusion and multi-scale dilated convolutions to capture extensive contextual information within the image. This capability enables the model to grasp long-range pixel relationships, enhancing its performance in semantic segmentation tasks [13].

Additionally, the MCE-CNN incorporates an Edge-Assisted Segmentation (EAS) module. This module combines the shallow ResNet features from the query image with edge features computed using the Sobel operator. This fusion aids in the final segmentation task, leading to improved accuracy and effectiveness in making multi-class predictions.

2.3.4 DenseNet-121

DenseNet-121 is part of the Densely Connected Convolutional Networks (DenseNet) family and is a Convolutional Neural Network (CNN) variant with 121 layers. Its architecture includes several key components [14]:

- Connectivity: DenseNet-121 exhibits dense connectivity, where each layer utilizes the feature maps from all preceding layers as inputs and shares its own feature maps with all subsequent layers.
- DenseBlocks: DenseNets are structured into DenseBlocks, where the feature map dimensions remain constant within the block, but the number of filters between layers varies.
- Transition Layers: Transition layers are inserted between DenseBlocks to reduce the number of channels by half compared to the existing channels.

In summary, DenseNet-121 comprises layers such as 1 7x7 Convolution, 58 3x3 Convolution, 61 1x1 Convolution, 4 AvgPool layers, and 1 Fully Connected Layer. DenseNets like DenseNet-121 are recognized for addressing challenges like the vanishing-gradient issue, promoting feature propagation and reuse, and significantly reducing parameter counts [15].

2.3.5 DCNN Regressors

Deep Convolutional Neural Network (DCNN) regressors are a type of machine learning model that uses DCNNs specifically for regression tasks, which involve predicting continuous numerical values rather than discrete categories [16].

DCNNs are a specialized type of artificial neural network designed for processing complex, multi-dimensional data such as images. They have multiple layers of neurons, each learning different features in the input. The layers are interconnected, passing their outputs to the next layer [17].

In regression, a DCNN regressor is trained to forecast continuous output based on detected input features. This involves adjusting network weights and biases during training to minimize the difference between predicted and actual outputs.

Chapter 3
System Methodology/Approach OR System Model

3.1 System Design

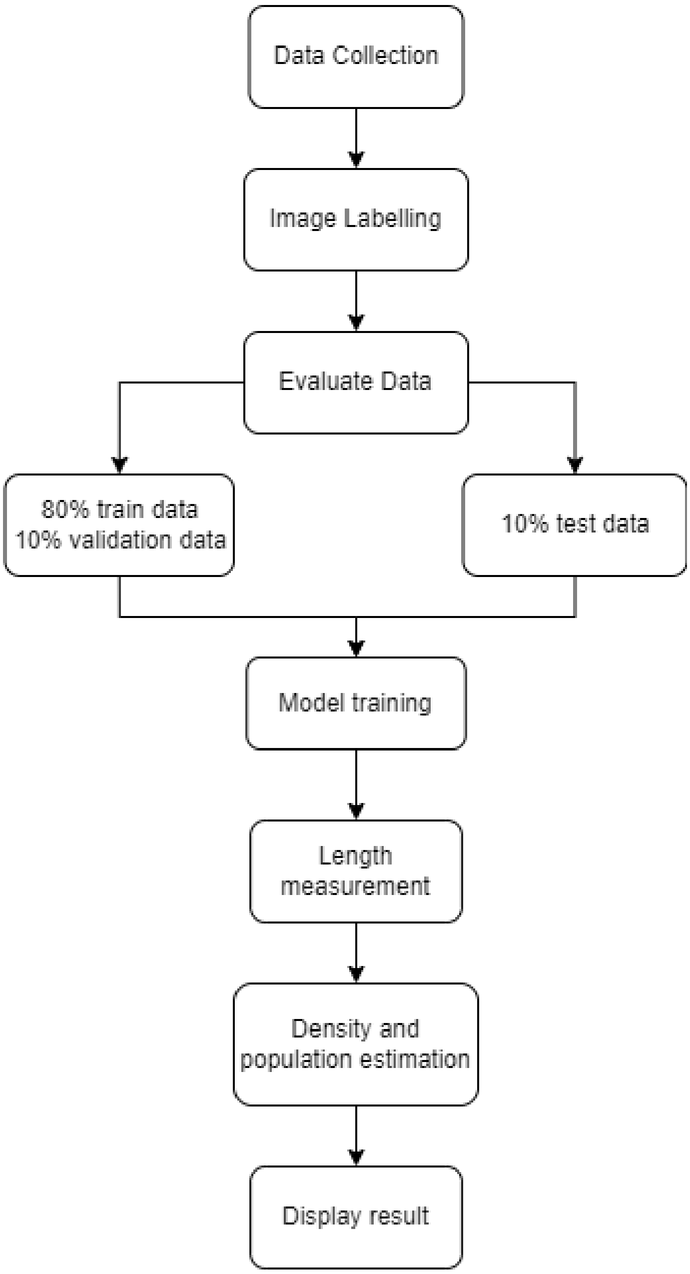


Figure 3.1.1 System Design for the Project

3.1.1 Data Collection

In this project, the image dataset needs to be prepared by researcher. The images were captured in different angle and with different background to stimulate the real-world prawn farms.

3.1.2 Image Labelling

After getting the images, it kept in a folder and then being labeled by “labelimg” which installed by Anaconda. This process is to annotate the images with relevant information, such as object label which will help to enable supervised learning. YOLO format is generated as the selected models required.

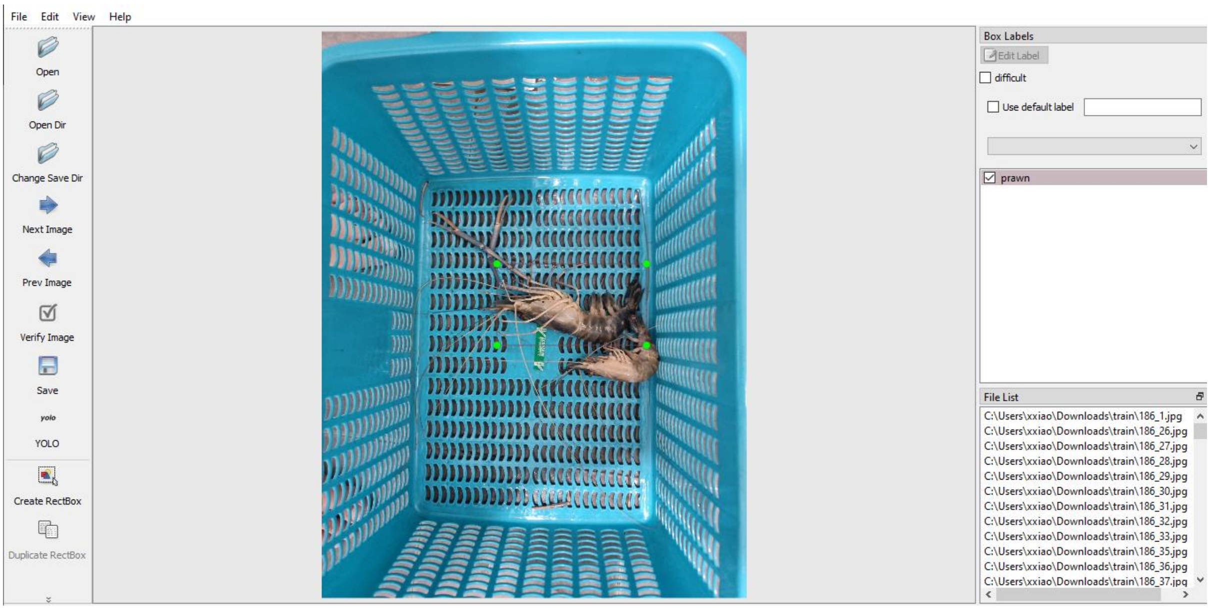


Figure 3.1.2 Label Image

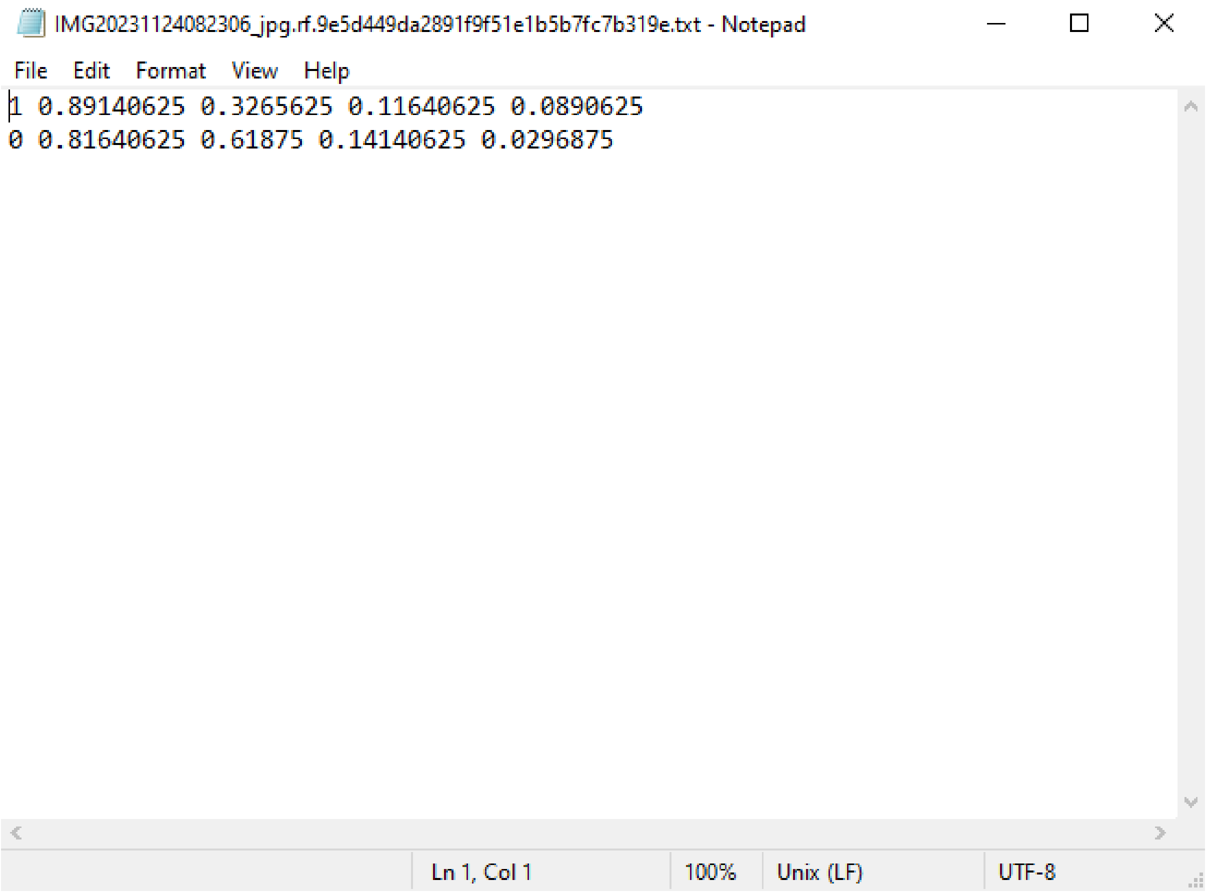


Figure 3.1.3 Annotations of the image

3.1.3 Evaluate Data

The dataset is split into 92% training data, 4% testing data and 4% validation data. The training data is used to train the model, while the test data is used to evaluate the generalizability of the model. This process is important to ensure that the model is reliable and can be used in real-world applications. The dataset is split with the aid of Roboflow.

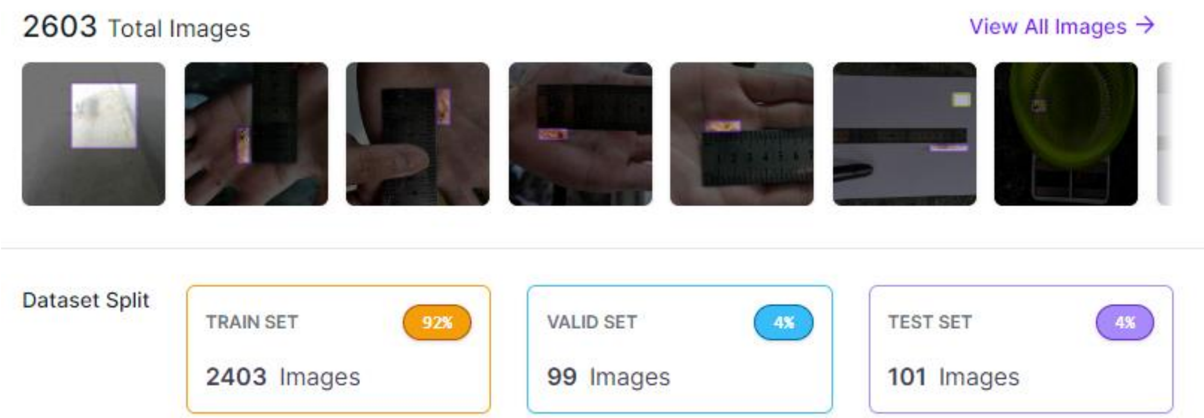


Figure 3.1.4 Splitting Dataset

3.1.4 Model training

You Only Look Once (YOLO)v7 is being chosen to train the model. Before training the model, preprocessing steps are done which auto-orient is applied and the image is resized to 640 x 640. Data augmentation steps such as saturation, brightness, exposure, blur and noise is being adjusted.

3.1.5 Length Measurement

After training the model, it is used to make predictions on the image. Then the bounding box coordinates are being extracted. The next step is to measure the bounding box length to get the length of the prawn. Here, the square is act as the reference object. By knowing the pixels and with the known fixed length of the square, the length of the prawn will be measured. It will display the test images with the predicted bounding box and saved in a folder.

3.1.6 Density and Population Estimation

After getting the length of the prawns and knowing the growth stage for the prawns, it is used to estimate the density and population for the prawn in the tank. For the density estimation, the formula of calculating it is as followed: *prawn counter / total area captured*. The prawn counter for density estimation is according to the growth stage that categorized in the length measurement, and the total area captured is the area of the tank that captured by the camera. The formula for estimation population is *density / total tank* area, and it is calculated based on the growth stage of the prawn.

3.2 Model Architecture

You Only Look Once version 7 (YOLOv7)

YOLOv7 is a cutting-edge object detection methodology known for its superior performance in terms of both speed and accuracy [18]. The architecture of YOLOv7 consists of three primary components:

1. Backbone: A Fully Convolutional Neural Network (FCNN) pulls key features from images.
2. Neck: This component takes the feature maps obtained from the Backbone and generates feature pyramids, aiding in multi-scale feature learning.
3. Head: This contains the output layers responsible for producing final detections.

The YOLOv7 architecture is built upon the E-ELAN design, which is an extension of the Efficient Layer Aggregation Networks (ELAN) architecture [19]. E-ELAN incorporates group convolution to increase the channel and cardinality of computational blocks, then applies a shuffle and merge cardinality operation to enhance feature learning without altering the gradient propagation path.

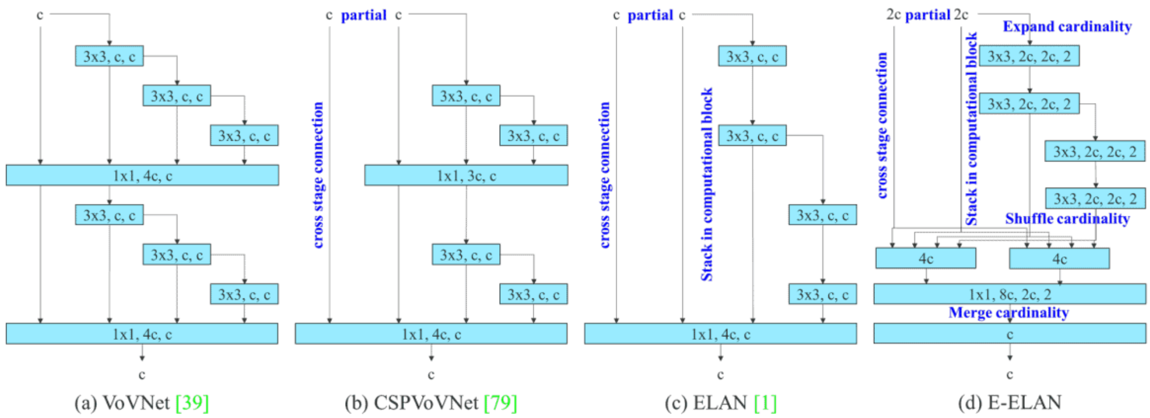


Figure 3.2.1 E-ELAN and previous work on maximal layer efficiency

Model scaling in YOLOv7 adopts a novel compound scaling method tailored for concatenation-based models like E-ELAN. Unlike previous methods that scaled depth and width independently, this approach scales these factors simultaneously within the computational blocks and transition layers.

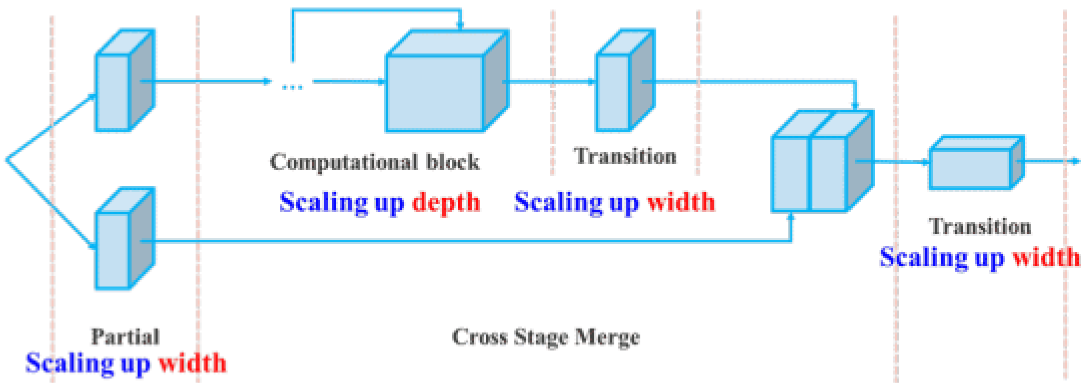


Figure 3.2.2 YOLOv7 compound scaling

The YOLOv7 architecture also incorporates a "trainable bag-of-freebies," incorporating various techniques to further enhance its capabilities. For instance, the planned re-parameterization model optimizes convolutional techniques like RepConv for non-plain architectures like ResNet and DenseNet.

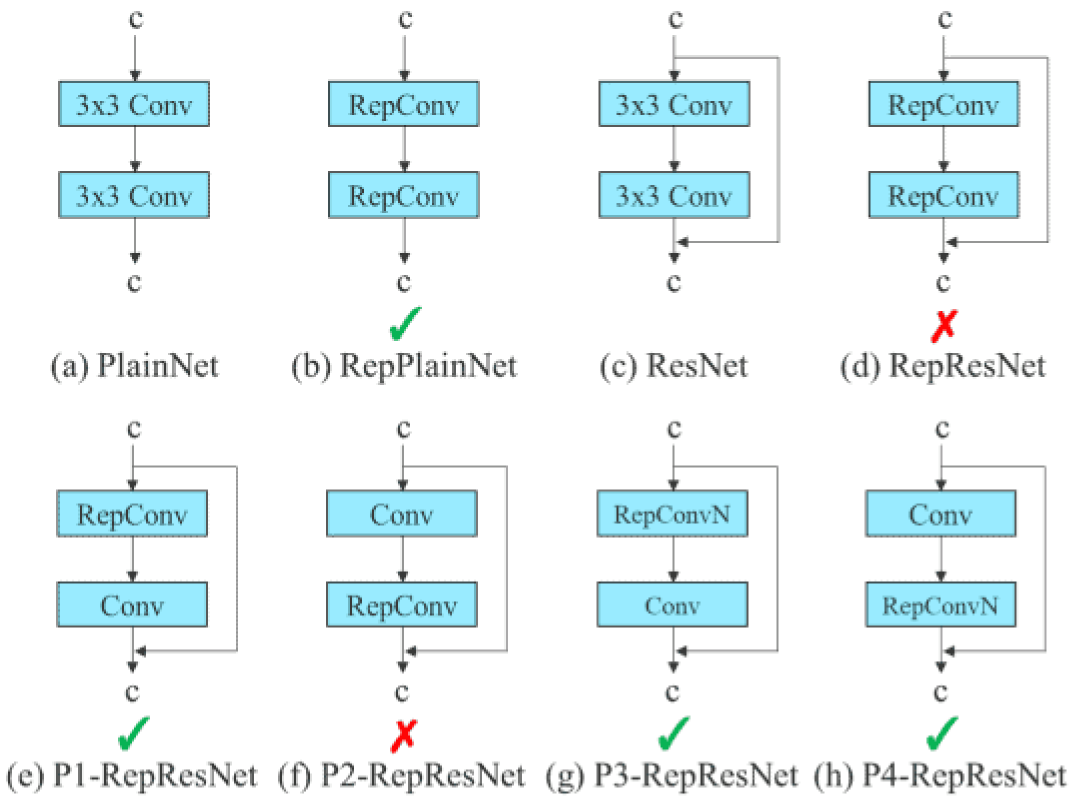


Figure 3.2.3 Re-parameterization trials

Additionally, the use of a coarse-to-fine lead guided label assignment mechanism improves the training of deep supervision mechanisms by leveraging predictions from the lead head to guide label assignments for both lead and auxiliary heads. These innovations collectively contribute to YOLOv7's state-of-the-art performance as a real-time object detector, surpassing previous methods in both speed and accuracy.

3.3 Evaluation Metrics

Assessing the performance of machine learning models is fundamental to their development and deployment in real-world applications. In the realm of object detection, where accuracy and efficiency are paramount, several key evaluation metrics are employed to gauge a model's effectiveness. By analyzing the metrics include Mean Average Precision (mAP), Average Recall and Average Precision, practitioners can gain insights into the model's ability to accurately detect objects, balance precision and recall, handle computational resources efficiently, and ultimately make informed decisions to optimize model performance for specific use cases.

3.3.1 Mean Average Precision (mAP)

Mean Average Precision (mAP) is a crucial metric utilized in object detection tasks to evaluate the performance of a model. It computes the Average Precision (AP) individually for each class within an image and then takes the average of these AP values across all classes. Precision, in this context, gauges the accuracy of the model in correctly identifying positive instances among its predictions. AP is specifically calculated as the area under the precision-recall curve for each class, providing a comprehensive assessment of the model's capability to accurately identify objects (precision) while ensuring it can detect all relevant objects (recall).

3.3.2 Confusion Matrix

A confusion matrix is a tabular representation commonly used to assess the performance of a classification model on a test dataset where the true values are known. It contains four distinct combinations of predicted and actual values: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). TP refers to instances where the model correctly predicts the positive class. TN represents cases where both the model prediction and the actual value are negative. FP are instances that were actually negative but were incorrectly predicted as positive by the model. Conversely, FN are cases where the model predicts a negative outcome, but the actual value is positive.

3.3.3 F1 curve

The F1 score assesses how well a model performs by considering both precision and recall. It's calculated using the formula $2 \times \frac{precision \times recall}{precsion + recall}$. This score is especially important when both false positives and false negatives matter. F1 ranges from 0 to 1, where 1 means perfect precision and recall, and 0 represents the worst performance.

The F1 curve shows how the F1 score changes with different thresholds. In binary classification, the threshold determines when a prediction is positive or negative. Adjusting this threshold helps balance precision and recall.

3.3.4 Recall curve

A recall curve is a plot of the recall (True Positive Rate) against the varying decision threshold of a binary classifier. The recall is the ratio of the number of true positive results divided by the number of all relevant samples (all actual positives). It measures the ability of the classifier to find all the positive instances. For each threshold, the recall is calculated and plotted, which forms the recall curve. The recall curve helps in understanding how the sensitivity of the model changes with different thresholds.

3.3.5 Precision curve

A precision curve, also called a Precision-Recall curve, plots the precision (Positive Predictive Value) against recall. Precision calculates the ratio of true positive results to all positive results (including true positives and false positives), indicating the classifier's capability to avoid labeling negative samples as positive. The Precision-Recall curve displays how precision and recall vary with different thresholds. A large area under the curve signifies high precision and recall, meaning low false positive and false negative rates, respectively.

3.4 System Requirement

3.4.1 Tools and Technologies Involved

The hardware used in this project:

1. Laptop

Description	Specifications
Model	MSI GF65 Thin 10UE
Processor	Intel Core i5-10500H
Operating System	Windows 11
Graphic	NVIDIA GeForce RTX 3060 Laptop GPU
Memory	16GB DDR4 RAM
Storage	500GB NVMe M.2 SSD

Table 3.4.1 Laptop Specifications

2. Smartphone

Description	Specifications
Model	Oppo Reno 8 Pro

Processor	Dimensity 8100-Max Octa-core
Operating System	ColorOS 13.1
GPU	Arm Mali-G610 MC6
Memory	12GB LPDDR5 RAM
Storage	256GB

Table 3.4.2 Smartphone Specifications

The software used in this project:

1. Jupyter Notebook

   Jupyter Notebook is an open-source web application that allows to create and share documents containing live code and visualizations. It supports various programming languages; Python is one of the most popular and will be used in this project. The code can be executed in cells, and the output for each cell can be saved.
2. Tensorflow

   Tensorflow is an open-source machine learning library. It provides a robust ecosystem of tools, libraries and community resources for building and deploying machine learning models. It is widely used for tasks like neural network development and model training.
3. Anaconda

   Anaconda is an open-source distribution of the Python and R programming languages for scientific computing. It is to simplify package management and deployment. Anaconda makes it easier to set up and manage environments with different dependencies.
4. Roboflow

   Roboflow is a platform that empowers developers to build their own computer vision applications, regardless of their skillset or experience. It provides a comprehensive suite of tools for each stage of the computer vision pipeline, including dataset management, labeling, model training, and deployment. Roboflow supports a variety of annotation formats and training frameworks, and it offers both edge and cloud deployment options.
5. Visual Studio Code

   Visual Studio Code is a free, open-source and cross-platform code editor developed by Microsoft. It supports many programming languages and features. It also allows debugging code right from editor. Visual Studio Code is highly extensible and customizable, with support for installing extensions to add new languages, themes and debuggers.

6. Google Colab

   Google Colab is a free cloud service provided by Google that allows users to write and execute Python in browser. It requires zero configuration and provides access to GPUs free of charge. It is integrated with Google Drive which allow for easy sharing and collaboration.

7. Looker Studio

   Looker Studio, formerly known as Google Data Studio, is a powerful data visualization and reporting platform offered by Google Cloud. This comprehensive tool enables users to transform raw data from various sources into visually stunning and interactive dashboards, reports, and data visualizations. The platform seamlessly integrates with a wide range of data sources, including Google's suite of products, databases, and third-party applications, ensuring that users can access and analyze data from multiple sources effortlessly.

Chapter 4
Preliminary Work

4.1 Prawn and Square Detection Model

Before training the prawn and square detection model, it is crucial to have a well-curated image dataset. To begin, the necessary images are captured using appropriate equipment and techniques. The captured images are then labeled using a tool like "LabelImg," which allows for the annotation of objects of interest (in this case, prawns and squares) within each image.

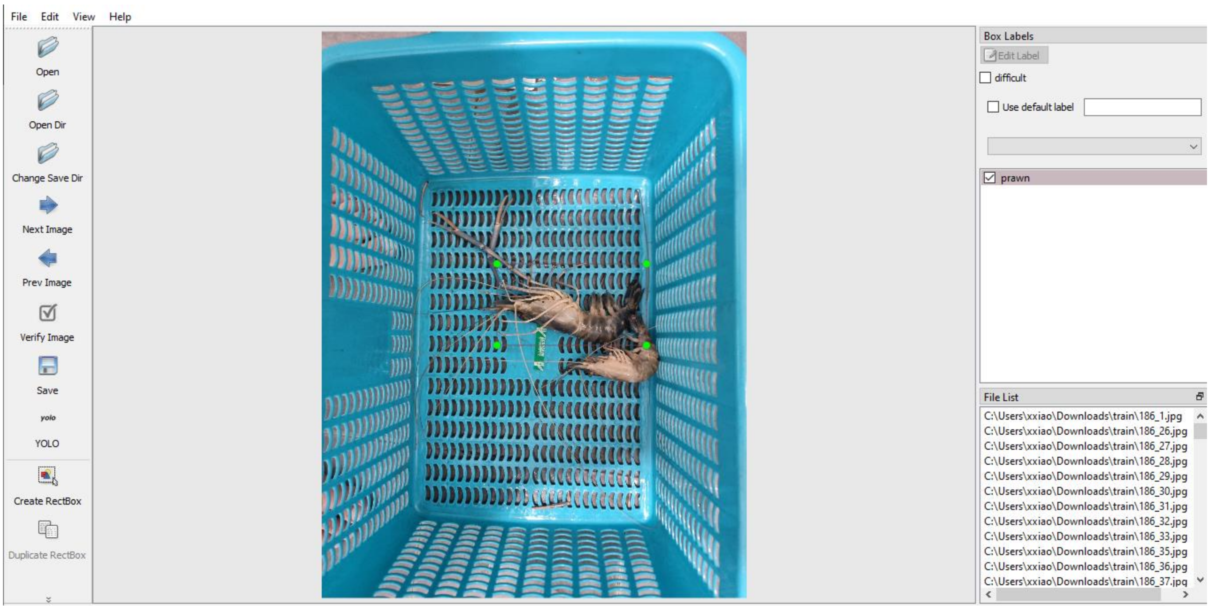
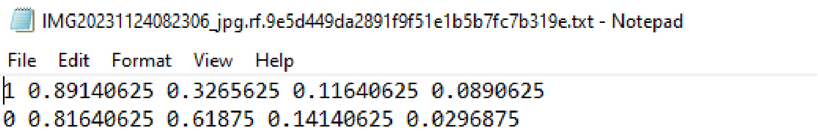


Figure 4.1.1 Labeling an Image


IMG20231124082306_jpg.rf.9e5d449da2891f9f51e1b5b7fc7b319e.txt - Notepad

File Edit Format View Help

1 0.89140625 0.3265625 0.11640625 0.0890625
0 0.81640625 0.61875 0.14140625 0.0296875

Figure 4.1.2 Annotations of the Image

Next, the labeled dataset, consisting of both the images and their associated annotations, is uploaded to Roboflow. Roboflow offers a range of preprocessing and data augmentation capabilities that can significantly improve the model's performance. In this case, the preprocessing step includes auto-orientation of the images and resizing them to a standard size of 640x640 pixels. The data augmentation step applies various transformations to the images, such as adjusting the saturation, brightness, exposure, blur, and adding a controlled amount of noise. These techniques help to increase the diversity of the training data, making the model more robust and less prone to overfitting.

Preprocessing	Auto-Orient: Applied Resize: Stretch to 640x640
Augmentations	Outputs per training example: 3 Saturation: Between -30% and +30% Brightness: Between -25% and +25% Exposure: Between -15% and +15% Blur: Up to 2px Noise: Up to 1.76% of pixels

Figure 4.1.3 Preprocessing and Data Augmentation Step

The project then moves to the Google Colab platform, a popular environment for machine learning model training. After downloading the YOLOv7 repository and installing the required dependencies, the preprocessed and augmented dataset from Roboflow is downloaded and integrated into the training process.

```
%cd yolov7

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="VszbS8cdKqs5K2rtS4tx")
project = rf.workspace("universiti-tunku-abdul-rahman-ox8ko").project("prawn_yolo")
version = project.version(6)
dataset = version.download("yolov7")
```

Figure 4.1.4 Code to Download Dataset from Roboflow

To kickstart the training, a COCO starting checkpoint is used. The COCO dataset is a large-scale object detection dataset that serves as a strong foundation for transfer learning, allowing the model to leverage pre-learned features and accelerate the training process.

```
[ ] # download COCO starting checkpoint
    %cd /content/yolov7
    !wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7_training.pt
```

Figure 4.1.5 Code to Download COCO Starting Checkpoint

Once the training is complete, the model produces several valuable outputs, including the F1 curve, Precision curve, Recall curve, and Confusion Matrix. These metrics provide insights into the model's performance, helping to identify areas for improvement and ensure the model's effectiveness in detecting prawns and squares within the precision aquaculture system. The "best.pt" file, which is the best-performing checkpoint from the training process, is then saved and can be used for further evaluation and deployment of the model. This checkpoint represents the model's optimal state and can be used to make predictions on new, unseen data.

```
%cd /content/yolov7
!python train.py --batch 16 --epochs 150 --data /content/yolov7/prawn_yolo-6/data.yaml --weights 'yolov7_training.pt' --device 0
```

Figure 4.1.6 Code for Training

4.2 Length Measurement

The next step in the project is to run the evaluation of the trained prawn and square detection model. However, before proceeding with the evaluation, it is important to note that some modifications to the existing code will be necessary.

```
# Run evaluation
!python detect.py --weights runs/train/exp2/weights/best.pt --conf 0.1 --source {dataset.location}/test/images
```

Figure 4.2.1 Code to Run Evaluation

The primary focus of the evaluation will be on length measurement, as well as density and population estimation within the precision aquaculture system. To enable these functionalities, changes need to be made to the "detect.py" script, which is part of the original YOLOv7 repository. These modifications involve adding new functionalities and algorithms that can extract the necessary information from the model's predictions.

Length Measurement:

The model detects and identifies the bounding boxes for both the prawns and the square within the image. The bounding box coordinates for each detected prawn and the square are stored in the variables bbox_prawn and bbox_square, respectively. Once both the prawn and the square are detected, the project calculates the size of the square in pixels by taking the difference between the x-coordinates of the square's bounding box corners. Knowing that the actual size of the square is 2.1 cm, the project can then compute the pixel-to-centimeter ratio by dividing the known size (2.1 cm) by the pixel size of the square. With the pixel-to-centimeter ratio established, the project can then calculate the length of the prawn in centimeters.

```
# Initialize bbox_square and bbox_prawn to None
bbox_square = bbox_prawn = None

# Iterate over predictions and print bounding box coordinates
for i, det in enumerate(pred):  # detections per image
    if len(det):
        # Rescale boxes from img_size to im0 size
        det[:, :4] = scale_coords(img.shape[2:], det[:, :4], im0s.shape).round()

        prawns_detected = []  # List to store detected prawns

        for *xyxy, conf, cls in reversed(det):
            if names[int(cls)] == 'square':
                bbox_square = xyxy
            elif names[int(cls)] == 'prawn':
                bbox_prawn = xyxy
                prawns_detected.append(bbox_prawn)


        for bbox_prawn in prawns_detected:
            # Check if both square and prawn are detected
            if bbox_square is not None and bbox_prawn is not None:

                # Calculate the size of the square in pixels
                square_size_pixels = bbox_square[2] - bbox_square[0]  # Assuming the format of bbox is (x1, y1, x2, y2)

                # Calculate the pixel-to-cm ratio
                pixel_to_cm_ratio = 2.1 / square_size_pixels  # The size of the square is known to be 2.1 cm
```

Figure 4.2.2 Code for Measure the Length of Prawn

It does this by taking the difference between the x-coordinates of the prawn's bounding box corners, which gives the length of the prawn in pixels. This pixel-based length is then multiplied by the previously calculated pixel-to-centimeter ratio to obtain the prawn's length in centimeters. The calculated prawn length is then rounded to 2 significant figures to ensure a reasonable level of precision.

```
# Calculate the length of the prawn in pixels
prawn_length_pixels = bbox_prawn[2] - bbox_prawn[0]  # Assuming the prawn's length is along the x-axis

# Convert the length of the prawn from pixels to cm
prawn_length_cm = (prawn_length_pixels * pixel_to_cm_ratio).item()

# Round the length to 2 significant figures
prawn_length_cm = round(prawn_length_cm, 2)

# Categorize the prawn based on its length
prawn_category = categorize_prawn(prawn_length_cm)

# Increment prawn counter based on stage
if prawn_category == "Juvenile":
    prawn_counter_juvenile += 1
elif prawn_category == "Subadult":
    prawn_counter_subadult += 1
else:  # Adult
    prawn_counter_adult += 1
```

Figure 4.2.3 Code for Converting the Length of Prawn

Finally, the prawn is categorized into one of the three growth stages (juvenile, subadult, or adult) based on the measured length.

```
def categorize_prawn(length):
    if length < 6.35:
        return "Juvenile"
    elif 6.35 <= length < 12.7:
        return "Subadult"
    else:
        return "Adult"
```

Figure 4.2.4 Code for Categorize Growth Stage Based on Length

<u>Weight Measurement:</u>

The weight of the prawns is estimated using the measured length of the individual prawns. To facilitate this, the project utilizes a specialized ruler provided by Sepang Today Aquaculture Centre (STAC) This ruler has a predefined relationship between the prawn's length and its corresponding weight. For instance, if a prawn measures 3.2 cm in length, the ruler indicates that its weight is 0.5 g.

To capture this length-weight relationship, the project maintainsh two arrays: “lengths” and “weights”. The lengths array stores the known prawn lengths in centimeters, while the weights array stores the corresponding weights in grams.

```
# Load your length-weight data and fit the model
lengths = np.array([3.2, 4.7, 5.2, 5.9, 6.3, 6.9, 7.3, 7.9, 8.4, 9, 9.6, 10.2, 10.8, 11.5, 12.2, 12.7, 13.2, 13.9, 15])
weights = np.array([0.5, 1, 1.25, 1.75, 2.25, 2.75, 3.38, 4.13, 5, 6, 7.25, 8.5, 10, 12, 14.25, 16.25, 18.38, 20.25, 22.34])
```

Figure 4.2.5 Length and Weight Array

For prawn lengths that are not explicitly stated in the provided arrays, the project employs curve fitting techniques to estimate the weight. Specifically, the curve_fit function from the SciPy library is utilized to fit a mathematical model to the existing length-weight data. This allows for the interpolation of weight values for any given prawn length, even if it falls between the discrete data points in the arrays.

```
def length_weight_model(L, a, b):
    return a * L**b

def estimate_weight(length, a, b):
    return length_weight_model(length, a, b)

# Load your length-weight data and fit the model
lengths = np.array([3.2, 4.7, 5.2, 5.9, 6.3, 6.9, 7.3, 7.9, 8.4, 9, 9.6, 10.2, 10.8, 11.5, 12.2, 12.7, 13.2, 13.9, 15])
weights = np.array([0.5, 1, 1.25, 1.75, 2.25, 2.75, 3.38, 4.13, 5, 6, 7.25, 8.5, 10, 12, 14.25, 16.25, 18.38, 20.25, 22.34])

# Convert NumPy arrays to PyTorch tensors
lengths_tensor = torch.tensor(lengths, dtype=torch.float32)
weights_tensor = torch.tensor(weights, dtype=torch.float32)

# Fit the model to the data
params, params_covariance = curve_fit(length_weight_model, lengths_tensor, weights_tensor, p0=[1.0, 1.0])
a, b = params
```

Figure 4.2.6 Code to Determine Weight

After calculating the prawn's length, determining its growth stage (juvenile, subadult, or adult), and estimating its weight based on the established length-weight relationship, this information is meticulously recorded in a Comma-Separated Values (CSV) file for comprehensive documentation and further analysis.

4.3 Density and Population Estimation

After measuring the length of the detected prawns and categorizing them into their respective growth stages (juvenile, subadult, and adult), the project maintains individual counters to track the number of prawns in each stage. These counters play a crucial role in the subsequent density and population estimation calculations.

```
# Increment prawn counter based on stage
if prawn_category == "Juvenile":
    prawn_counter_juvenile += 1
elif prawn_category == "Subadult":
    prawn_counter_subadult += 1
else:  # Adult
    prawn_counter_adult += 1
```

Figure 4.3.1 Counter to Track Number of Prawns

Before proceeding with the density and population estimation, it is essential to modify the values representing the total area captured by the camera system and the total area of the aquaculture tank. These parameters are critical inputs to the estimation formulas and must be accurately specified to reflect the actual dimensions of the cultivation environment.

```
# Define the total area in square cm
total_area_cm2 = 2100  # real area of camera captured

# Define the total area of the tank in square cm
total_tank_area_cm2 = 17664  # whole tank area
```

Figure 4.3.2 Total Area for Both Captured and Tank

The density estimation is calculated by dividing the prawn counter for each growth stage by the total area captured by the camera system. This provides a measure of the spatial distribution of prawns within the observed area, allowing for insights into the crowding or dispersion of the population.

Building upon the density calculation, the project then estimates the overall population of prawns in the aquaculture tank. This is achieved by dividing the density value by the total area of the tank, effectively scaling the density to the broader cultivation system. This population estimation enables a comprehensive understanding of the prawn population dynamics within the precision aquaculture setting.

```
# Calculate density (prawns per square cm) for each stage
density_juvenile = prawn_counter_juvenile / total_area_cm2
density_subadult = prawn_counter_subadult / total_area_cm2
density_adult = prawn_counter_adult / total_area_cm2

population_juvenile = density_juvenile * total_tank_area_cm2
population_subadult = density_subadult * total_tank_area_cm2
population_adult = density_adult * total_tank_area_cm2
```

Figure 4.3.3 Code for Density and Population

4.4 Data Visualization on Looker Studio

After the automated monitoring system has collected and processed the data on prawn detection, length measurement, and density/population estimation, the project leverages the powerful data visualization capabilities of Looker Studio by Google.

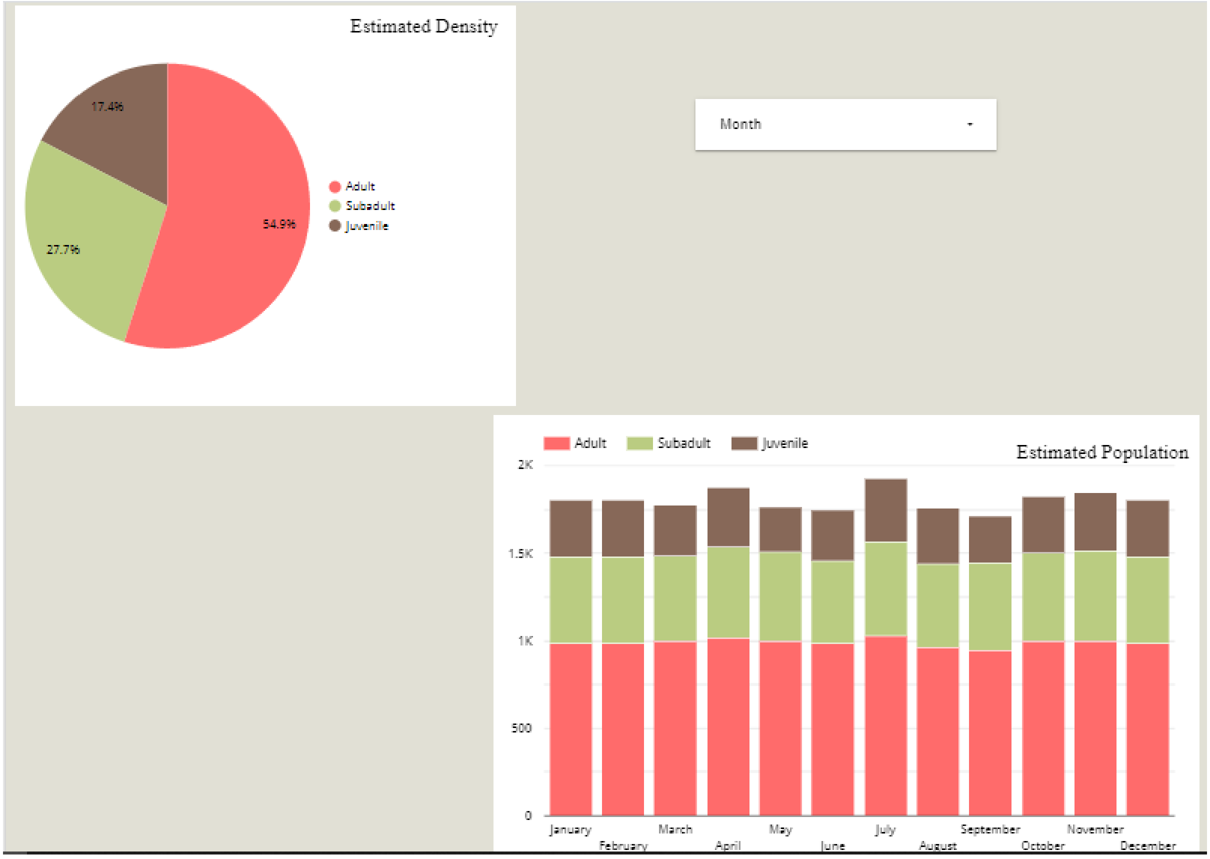


Figure 4.4.1 Visualization Dashboard

The pie chart effectively illustrates the estimated density distribution across the various growth stages (juvenile, subadult, and adult), providing a clear snapshot of the population dynamics within the aquaculture system. Additionally, the stacked column chart offers a temporal perspective, displaying the estimated population for each growth stage over a monthly timeline. To enhance the interactivity and user experience, a dropdown list is incorporated, allowing users to select the specific month they wish to analyze, dynamically updating the chart to display only the relevant data for that chosen period.

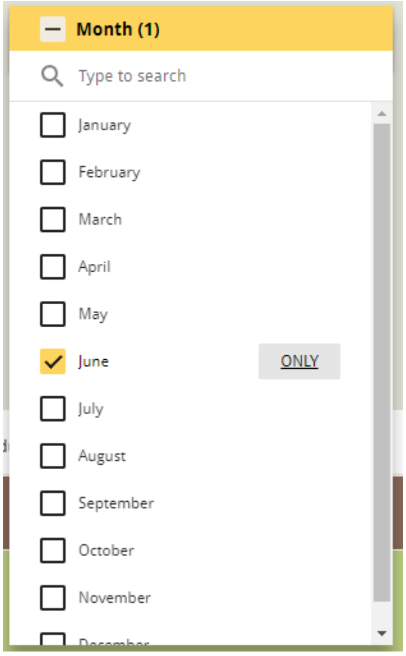


Figure 4.4.2 Dropdown List to Select the Month

After the user select the month that they wish to view, the dashboard will update the chart to display the relevant data.

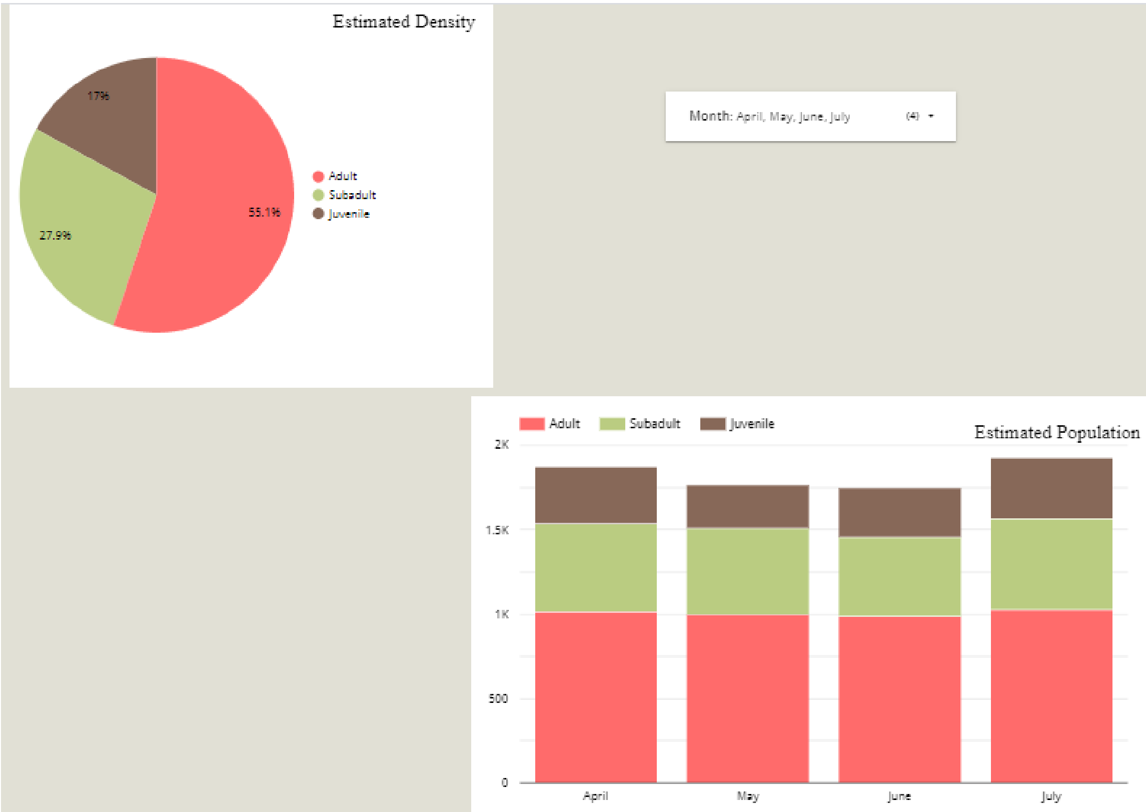


Figure 4.4.3 Updated Dashboard

When the user clicks on the specific stage (like adult), then the chart will only display the data related to the “Adult” stage.

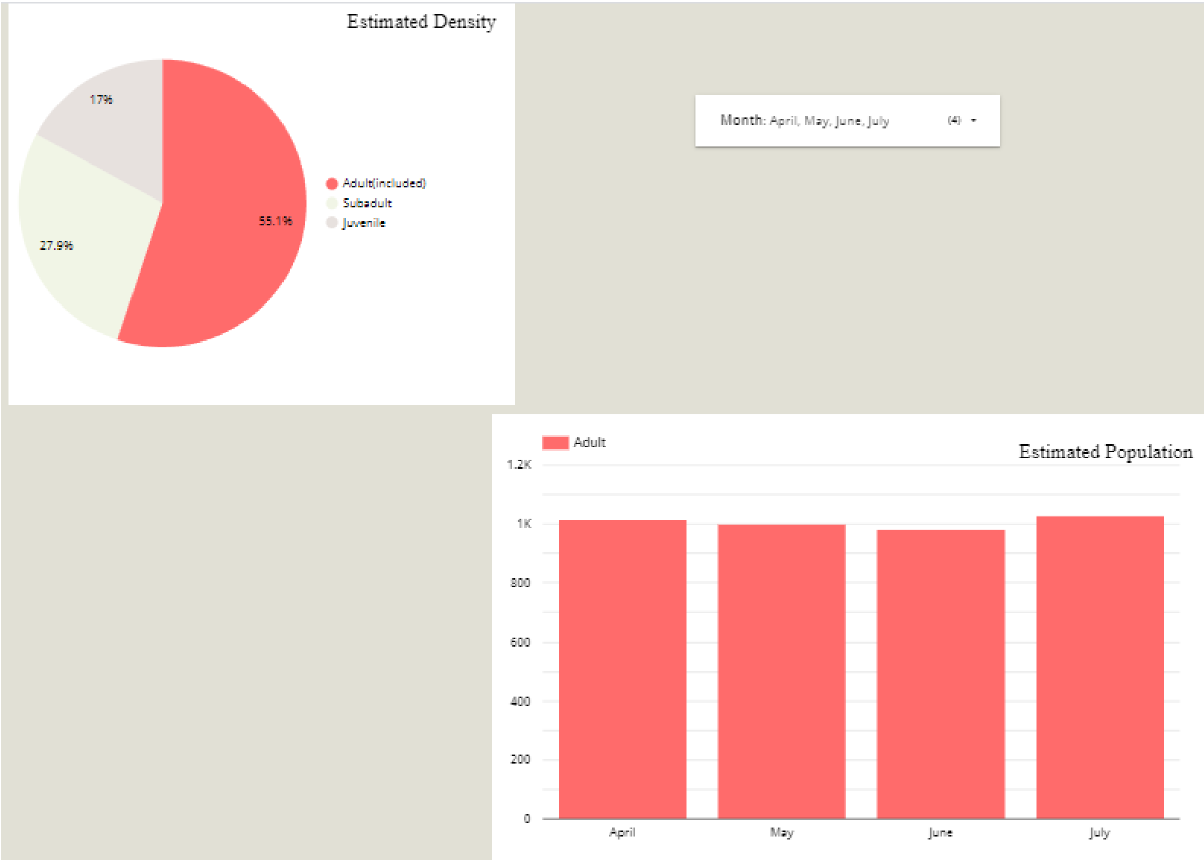


Figure 4.4.4 Display on Specific Stage

Chapter 5
System Implementation

5.1 Hardware Setup

First, the camera should connect with a cable and the power is turned on. Wait until the prompt "Waiting for connection" is heard. Then, open the "Xiaomi Home" app on the phone and tap on the "Add a device" located at the top right corner.

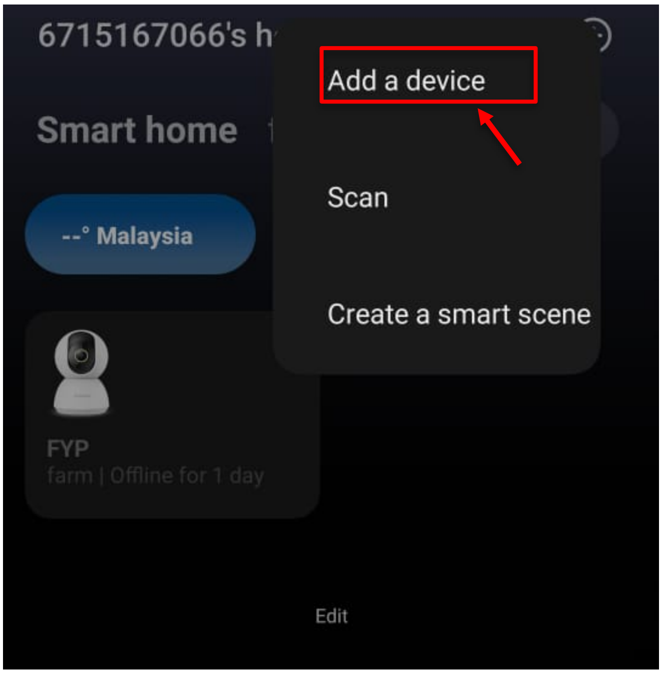


Figure 5.1.1 Add Device

Remember to activate both Wi-Fi and Bluetooth on the phone. Next, select "Scan code to add" and use the phone's camera to scan the QR code located at the bottom of the camera.

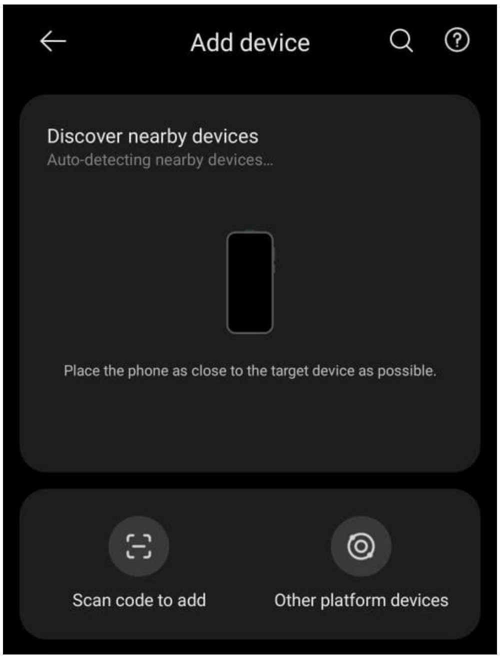


Figure 5.1.2 Scan Code to Add

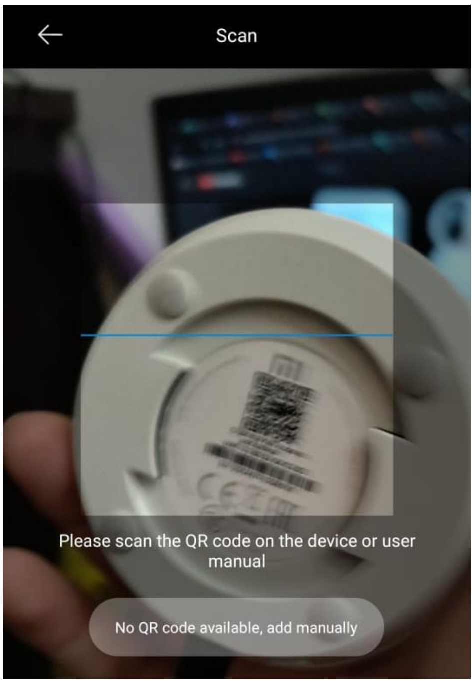


Figure 5.1.3 QR at the Bottom of Camera

After scanning, press and hold the reset button on the camera for about 3 seconds until the notification light turns yellow.

Figure 5.1.4 Reset Button

Figure 5.1.5 Yellow Light

Then proceed with the device reset by pressing the Device reset on the screen. Once done, set up the Wi-Fi connection and wait for phone to detect the device. Follow the on-screen instructions to connect the device to the Wi-Fi network, then press the return button once connected.

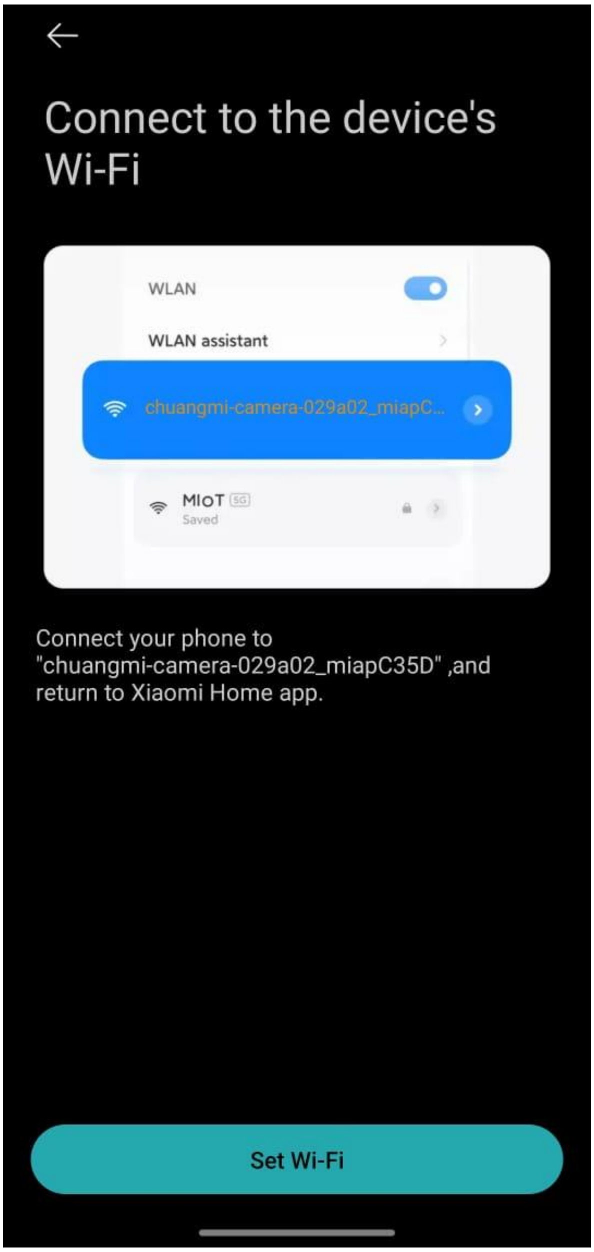


Figure 5.1.6 On-Screen Instruction

A loading page will appear to establish the connection with the nearest router. Once the device is successfully connected, the setup process is finished.

The camera is strategically positioned to capture the view of the tank, enabling comprehensive monitoring and analysis of the prawn.

Figure 5.1.7 Camera Setup

5.2 Software Setup

Roboflow

1. Go to website https://universe.roboflow.com/ then sign in to the account (If new user then clicks "Create Account").

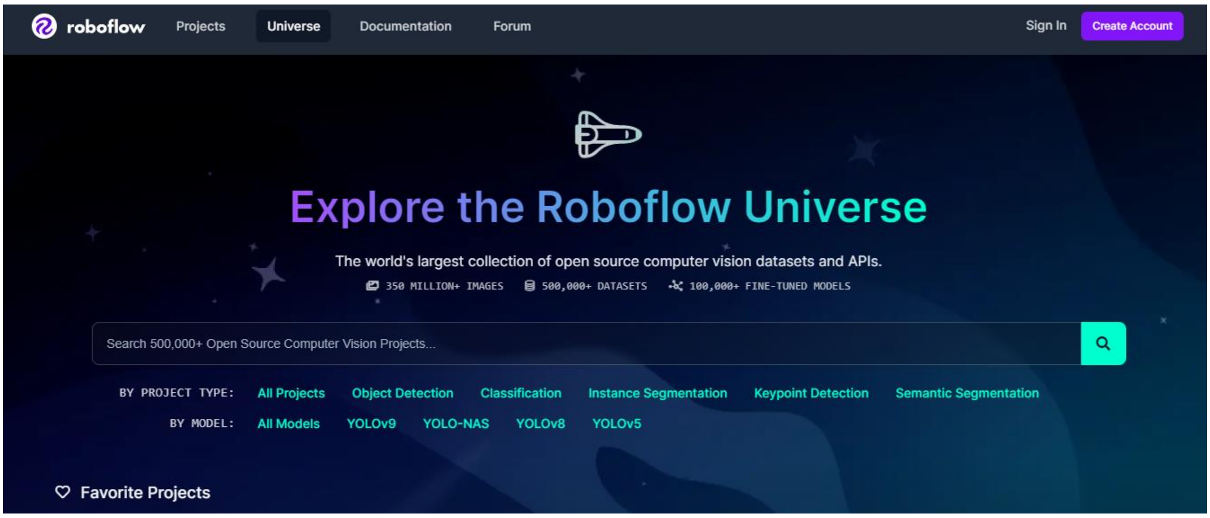


Figure 5.2.1 Roboflow official website

2. Click on the "Project."

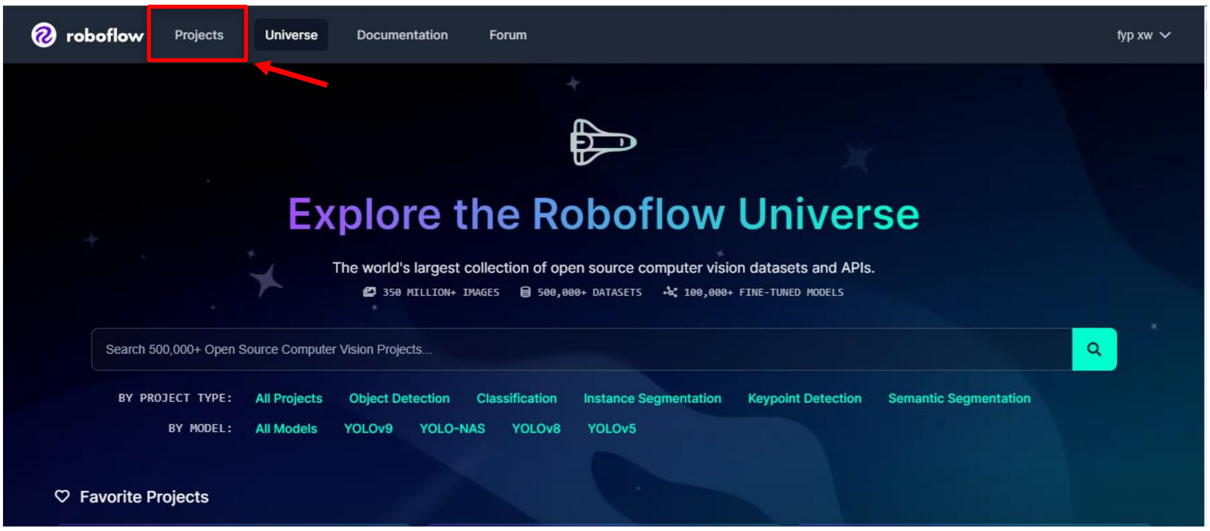


Figure 5.2.2 Roboflow Project

3. Click on the "Create New Project"

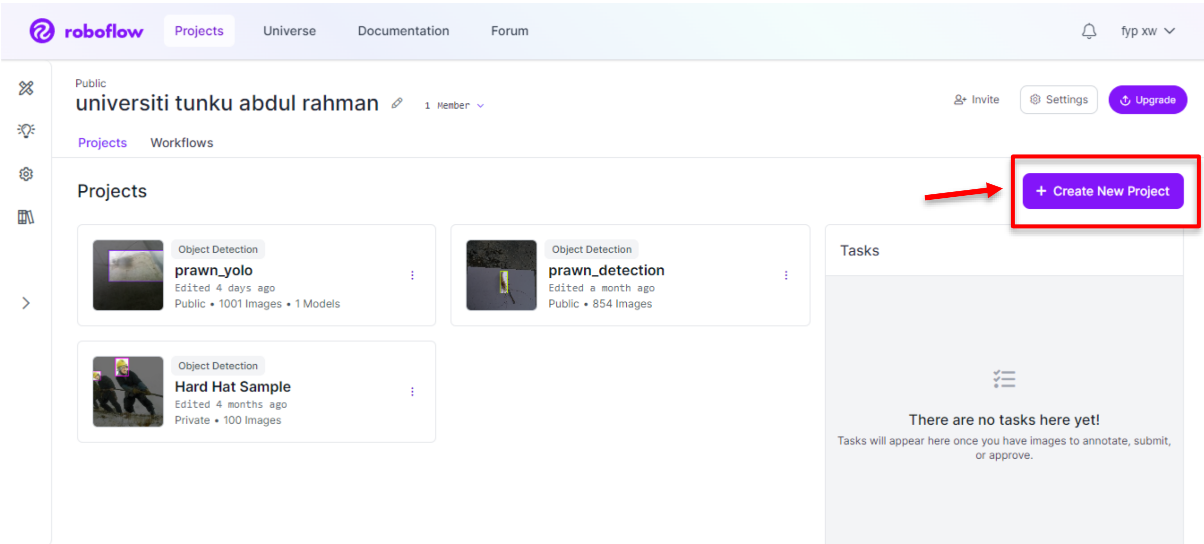


Figure 5.2.3 Project page

4. Create the new project by filling the name and annotation group.

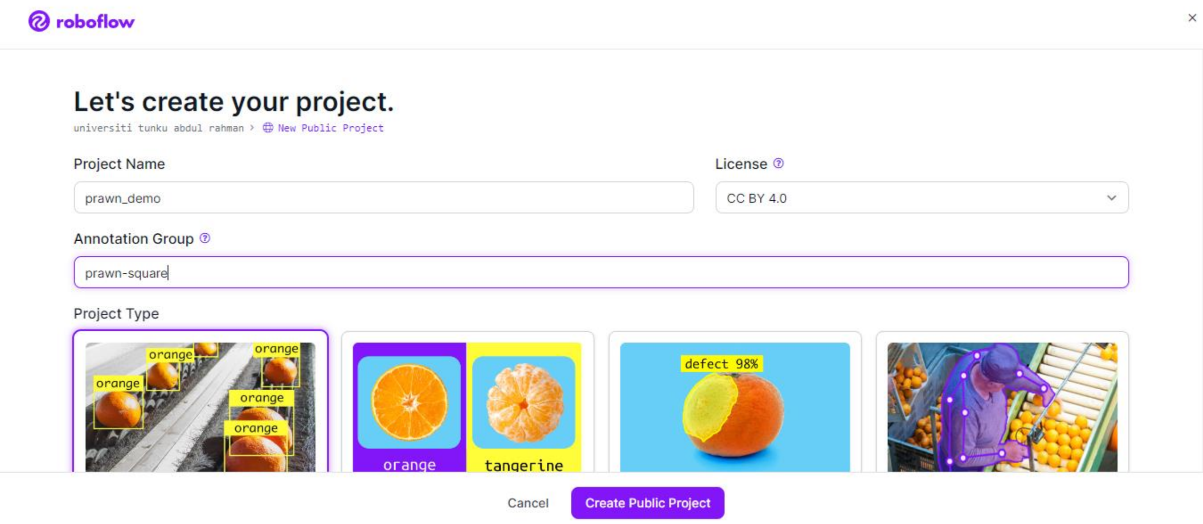


Figure 5.2.4 Information of Project

5. Click "Select Folder" on which the folder containing all the images being captured and labelled.

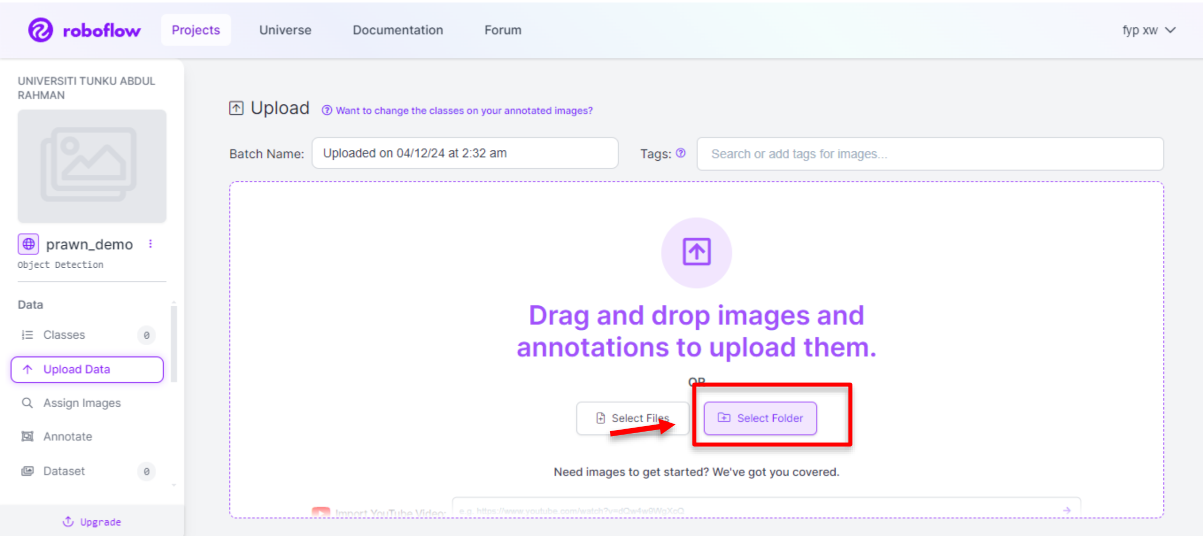


Figure 5.2.5 Select Folder with Images and Annotations.

6. Wait for the data to upload.

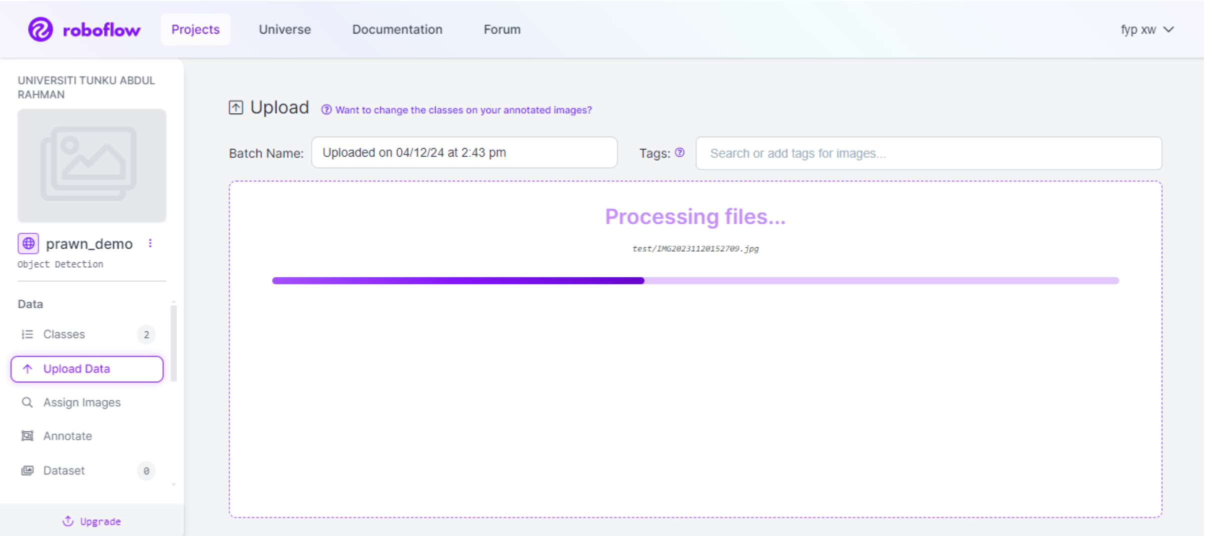


Figure 5.2.6 Uploading data

7. After the data finish uploading, split the data according to the percentage.

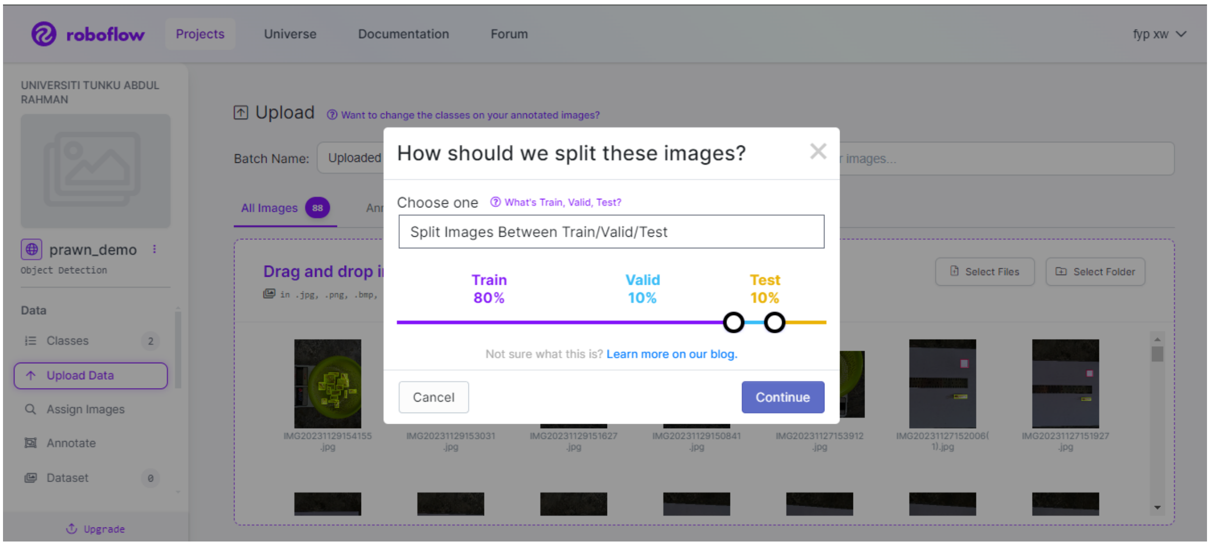


Figure 5.2.7 Splitting data

8. Click “Save and Continue”.

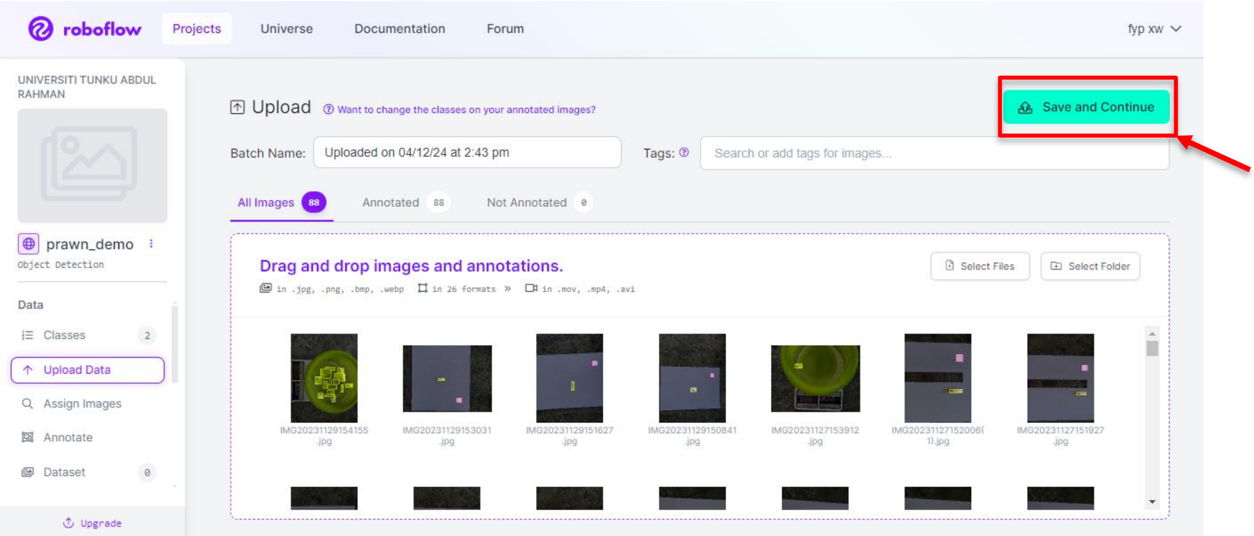


Figure 5.2.8 Save and Continue

9. After the data finish uploaded, click the “Classes” and it will show the classes that labelled.

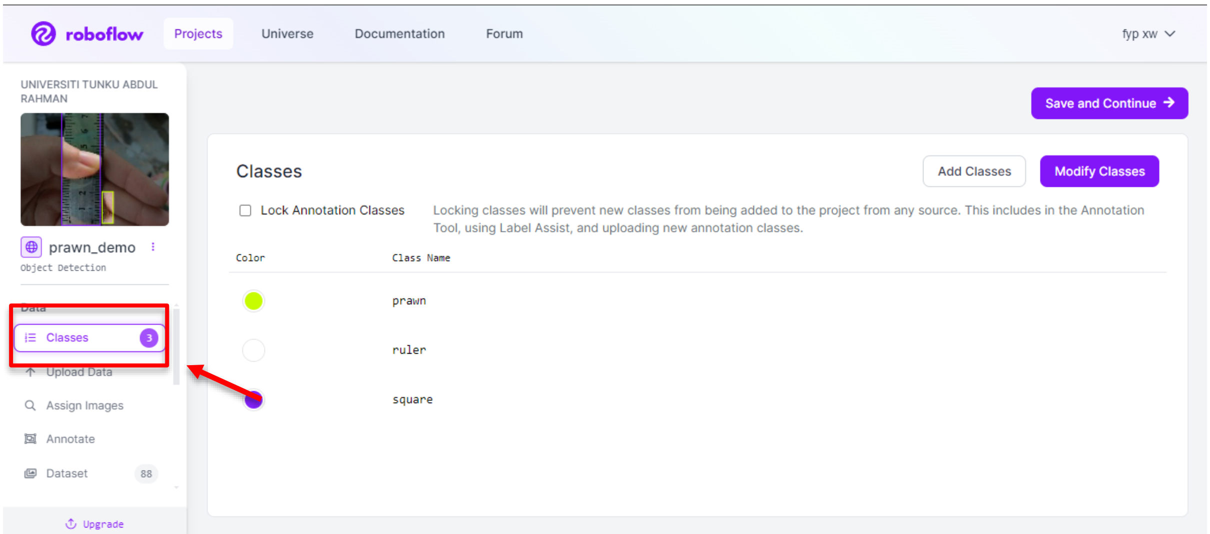


Figure 5.2.9 Classes

10. Then click on the “Generate” on the left of the page to add the preprocessing step ad data augmentation step.

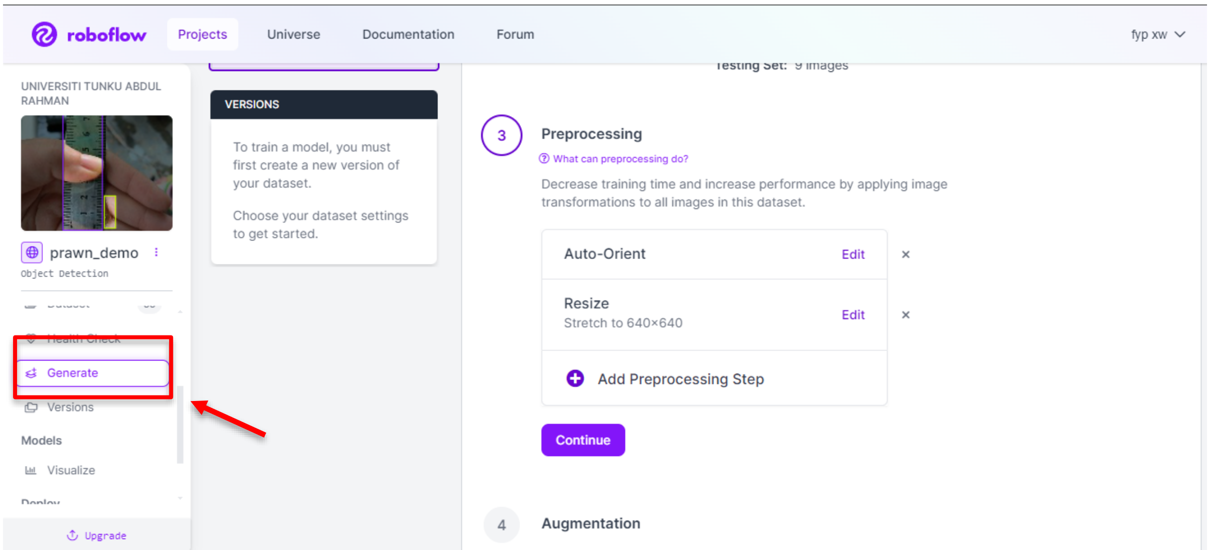


Figure 5.2.10 Add Preprocessing and Data Augmentation Step

11. After adding all the needed preprocessing and data augmentation step, click on the "Create".

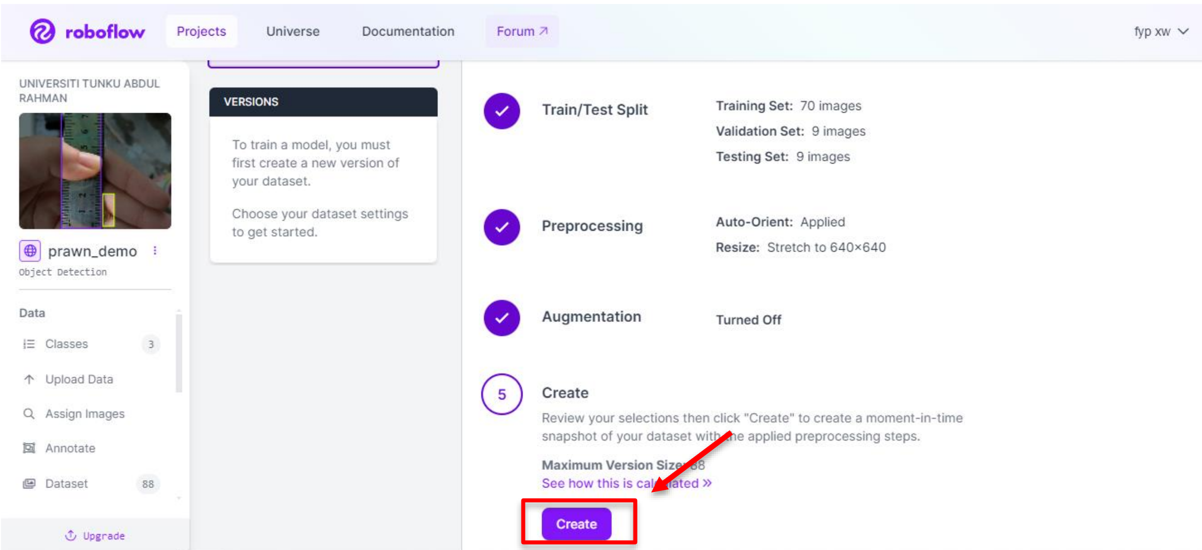


Figure 5.2.11 Creating dataset with the steps needed

12. After it done initializing, click on the "Export Dataset".

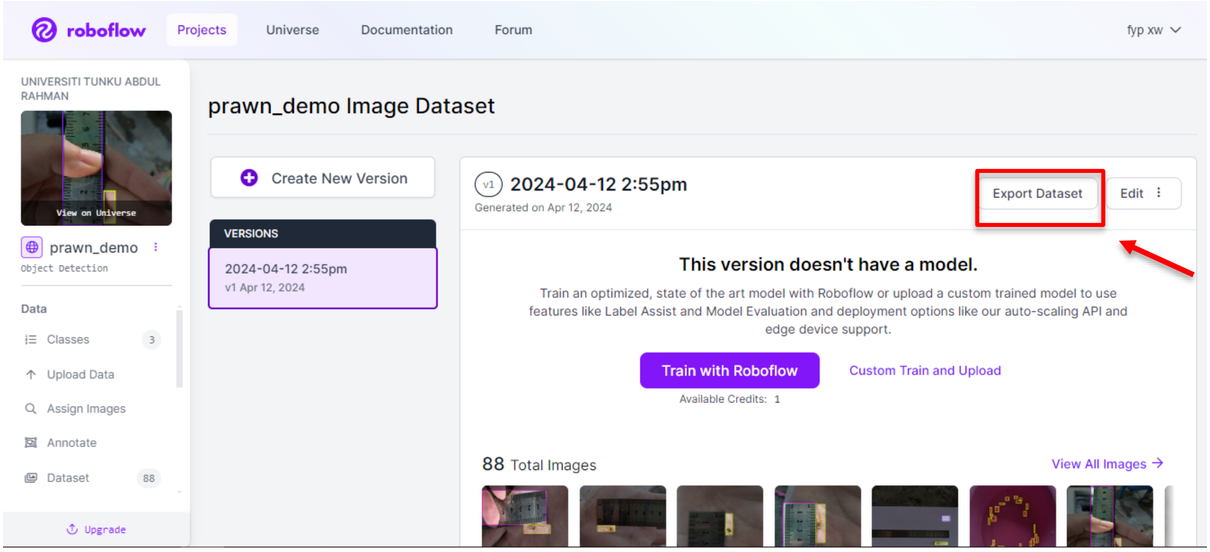


Figure 5.2.12 Export the dataset

13. Select the format wanted, in this case is YOLOv7. Then click "Continue".

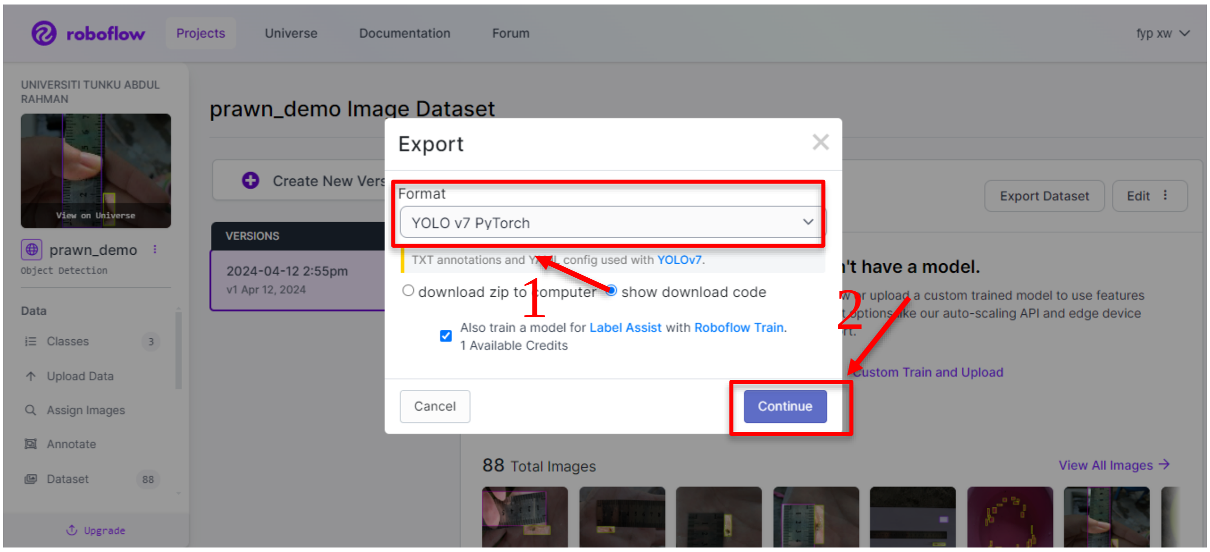


Figure 5.2.13 Format Selection

14. It will generate the download code, copy it and paste it to Google Colab.

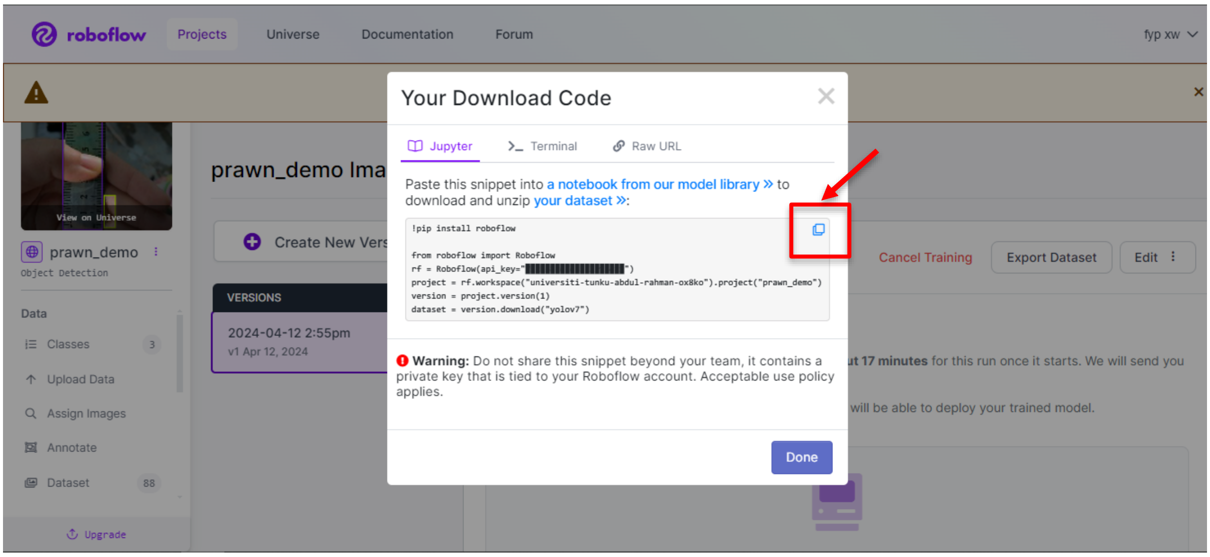


Figure 5.2.14 Download Code from Roboflow

<u>Google Colab</u>

1. Download YOLOv7 repository and install requirements.

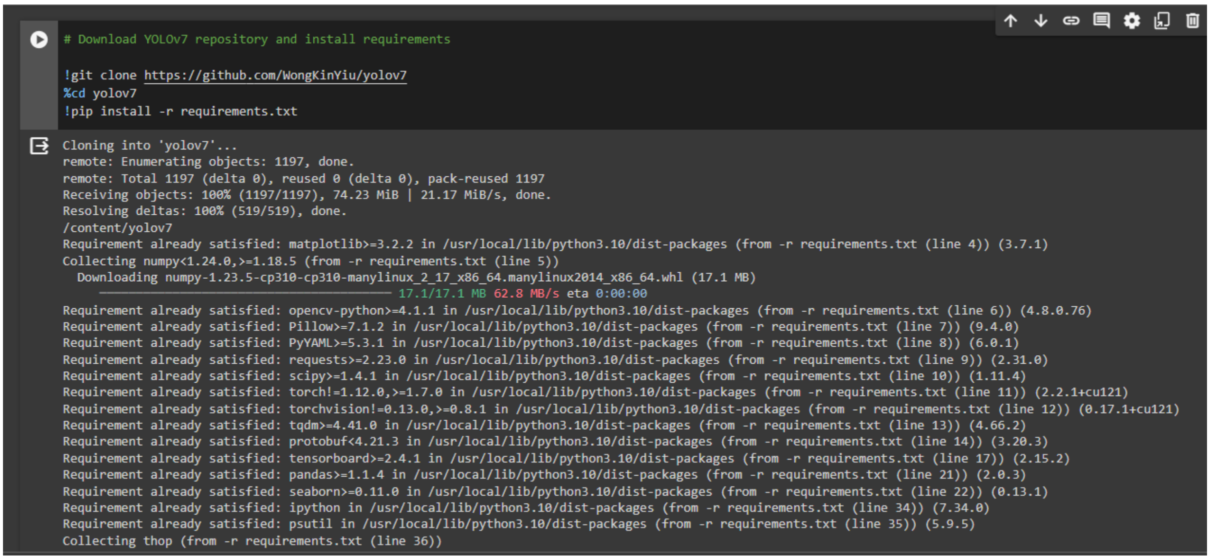


Figure 5.2.15 YOLOv7 installation

2. Paste the code that copied from the Roboflow into the cell, it is to download the dataset that from the Roboflow.

```
%cd yolov7

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="VszbS8cdKqs5K2rtS4tx")
project = rf.workspace("universiti-tunku-abdul-rahman-ox8ko").project("prawn_yolo")
version = project.version(6)
dataset = version.download("yolov7")
```

```
[Errno 2] No such file or directory: 'yolov7'
/content
Collecting roboflow
  Downloading roboflow-1.1.27-py3-none-any.whl (74 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 74.1/74.1 kB 2.0 MB/s eta 0:00:00
Collecting certifi==2023.7.22 (from roboflow)
  Downloading certifi-2023.7.22-py3-none-any.whl (158 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 158.3/158.3 kB 16.2 MB/s eta 0:00:00
Collecting chardet==4.0.0 (from roboflow)
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 178.7/178.7 kB 21.2 MB/s eta 0:00:00
Collecting cycler==0.10.0 (from roboflow)
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting idna==2.10 (from roboflow)
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.8/58.8 kB 9.5 MB/s eta 0:00:00
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.5)
```

Figure 5.2.16 Download code from Roboflow

3. Download the COCO starting checkpoint using the code.

```
[ ] # download COCO starting checkpoint
    %cd /content/yolov7
    !wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7_training.pt

    /content/yolov7
    --2024-04-11 14:42:17--  https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7_training.pt
    Resolving github.com (github.com)... 140.82.121.3
    Connecting to github.com (github.com)|140.82.121.3|:443... connected.
    HTTP request sent, awaiting response... 302 Found
    Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/511187726/13e046d1-f7f0-43ab-910b-480613181b1f?X
    --2024-04-11 14:42:18--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/511187726/13e046d1-f7f0-43ab-910b
    Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
    Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
    HTTP request sent, awaiting response... 200 OK
    Length: 75628875 (72M) [application/octet-stream]
    Saving to: ‘yolov7_training.pt’

    yolov7_training.pt  100%[===================>]  72.12M   329MB/s    in 0.2s

    2024-04-11 14:42:18 (329 MB/s) - ‘yolov7_training.pt’ saved [75628875/75628875]
```

Figure 5.2.17 Download COCO starting checkpoint

5.3 Implementation Issues and Challenges

The biggest implementation issue in this project is lacking enough dataset of Giant Freshwater Prawn online. The dataset that available online is not enough to train a model to be accurately detect the prawn. To handle this issue, the dataset is being obtained by capturing the real prawn with different background and gesture. The dataset is accessible at Roboflow for future purpose: https://app.roboflow.com/universiti-tunku-abdul-rahman-ox8ko/prawn_yolo/6

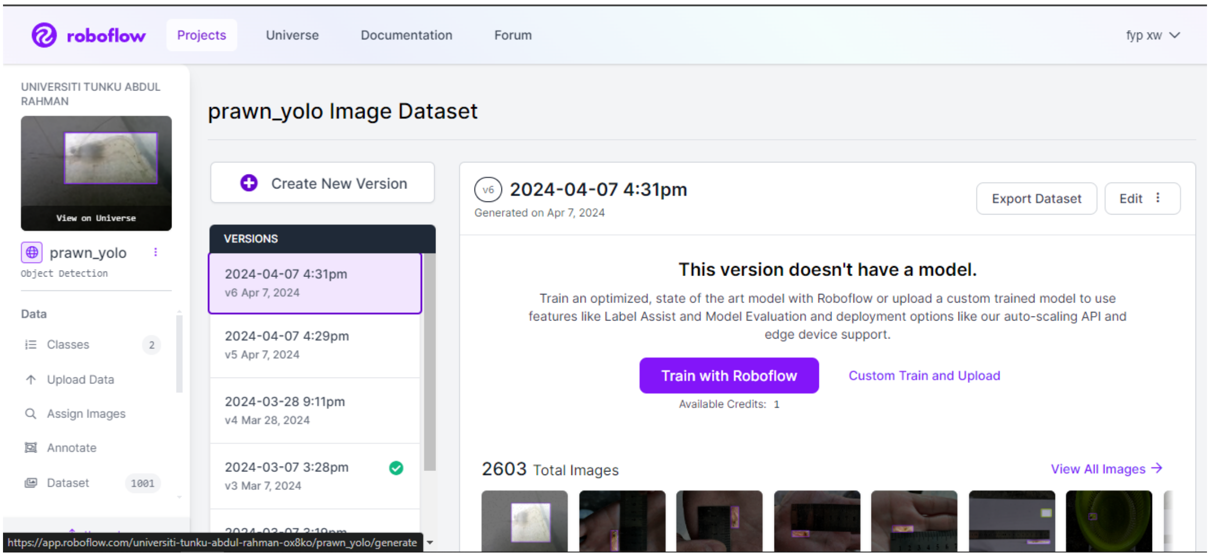


Figure 5.3.1 Dataset uploaded on Roboflow

Another critical implementation challenge for this project is the potential for decreased water visibility within the precision aquaculture system. If the water in the tank becomes cloudy, turbid, or otherwise obstructed, it can significantly impact the ability of the computer vision-based detection and measurement system to accurately identify and analyze the prawns and the reference square. In order to address the challenge, regular monitoring and the use of filtration technology can help ensure consistently high-water clarity, enabling the computer vision system to function effectively.

Chapter 6
System Evaluation and Discussion

6.1 Performance Metrics

6.1.1 mAP

```
     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
   149/149     16.1G   0.03223  0.006868  0.003252   0.04235         9       640: 100% 151/151 [00:52<00:00,  2.89it/s]
               Class     Images     Labels          P          R      mAP@.5  mAP@.5:.95: 100% 4/4 [00:01<00:00,  2.44it/s]
                 all         99        198      0.925      0.949       0.949       0.68
               prawn         99         99      0.952          1       0.995      0.779
              square         99         99      0.899      0.899       0.904       0.58
150 epochs completed in 2.255 hours.

Optimizer stripped from runs/train/exp/weights/last.pt, 74.8MB
Optimizer stripped from runs/train/exp/weights/best.pt, 74.8MB
```

Figure 6.1.1 mAP

The mAP (mean Average Precision) is a popular metric used to measure the accuracy of object detectors like YOLO. The mAP values are provided for each epoch at Intersection over Union (IoU) 0.5 and IoU 0.5:0.95 (averaged over IoU from 0.5 to 0.95 with a step size of 0.05). From the figure, the mAP values for the last epoch are as followed:

- For 'all' classes, the mAP@0.5 is 0.949 and mAP@0.5:.95 is 0.68.
- For the 'prawn' class, the mAP@0.5 is 0.995 and mAP@0.5:.95 is 0.779.
- For the 'square' class, the mAP@0.5 is 0.904 and mAP@0.5:.95 is 0.58.

The mAP@0.5 values are quite high, close to 1, which indicates that the model is performing well at an IoU threshold of 0.5. However, the mAP@0.5:.95 values are lower, indicating that the performance decreases as the IoU threshold increases. This is common as higher IoU thresholds require more precise bounding box predictions.

6.1.2 Confusion Matrix

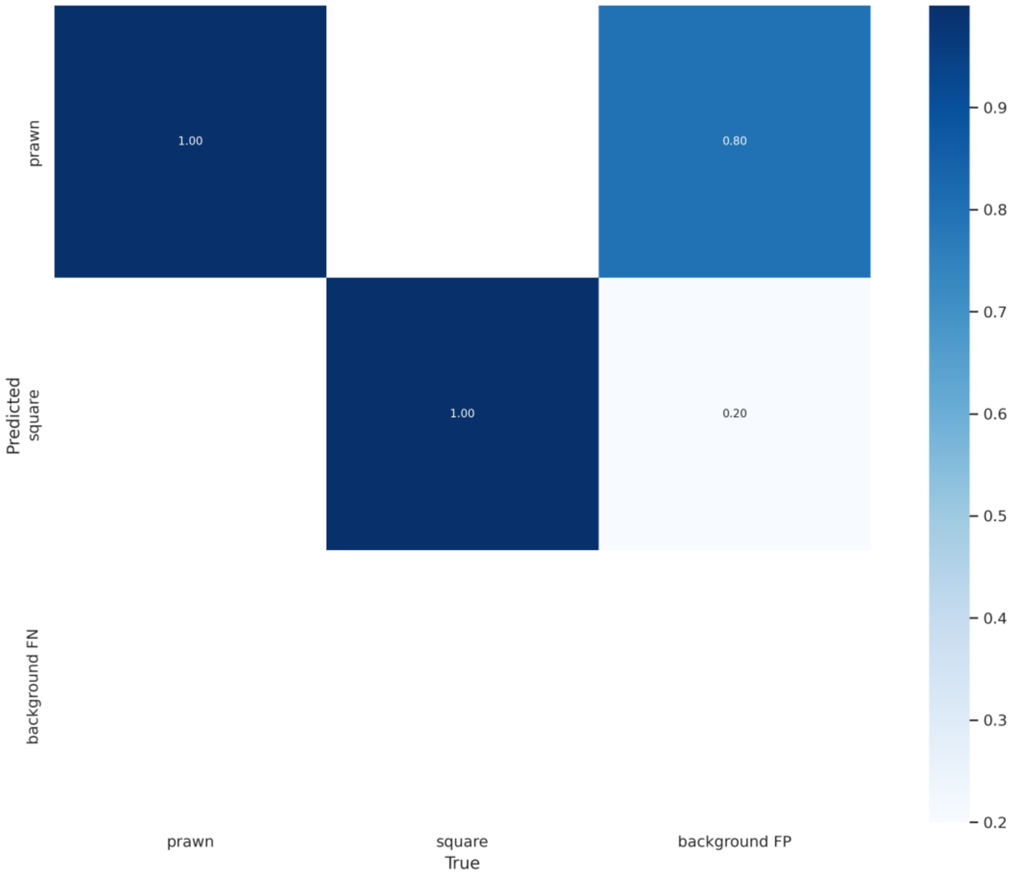


Figure 6.1.2 Confusion Matrix

The Y-axis of the confusion matrix represents the actual classes. It consists of prawn, predicted square and background FN. The X-axis is actually the predicted classes, which consists of prawn, square true and background FP. The explanation is as followed:

Y-axis (Actual Classes):

- Prawn: This is the actual class of prawns in the dataset.
- Predicted Square True: This is where the model predicted squares.
- Background FN (False Negative): This is when the model predicted the background, but it was actually a prawn or square.

X-axis (Predicted Classes):

- Prawn: This is where the model predicted prawns.
- Square True: This is the actual class of squares in the dataset.
- Background FP (False Positive): This is when the model incorrectly predicted a prawn or square, but actually, it was the background.

The cell at the intersection of “Prawn” (Y-axis) and “Prawn” (X-axis) has a value of 1.00. This means that all prawns were correctly identified as prawns by the model. The cell at the intersection of "Predicted square" (Y-axis) and "Square True" (X-axis) has a value of 1.00. All the squares were correctly identified. And there is a value of 0.80 at the intersection of "Prawn" (Y-axis) and "Background FP" (X-axis), which indicating the model falsely predicted it as prawn, but it actually was the background. Last cell at the intersection of "Predicted square" (Y-axis) and "Background FP" (X-axis) has a value of 0.20, which show that the model incorrectly predicted the background as a square.

6.1.3 F1 Curve

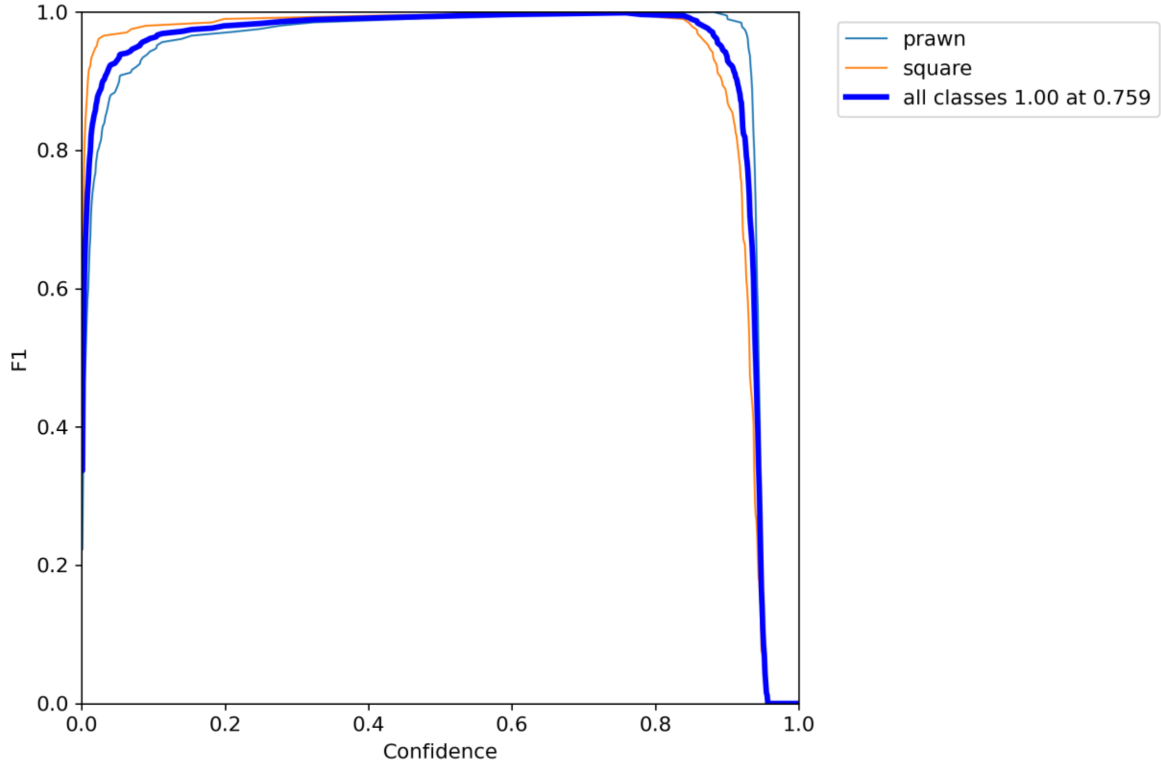


Figure 6.1.3 F1 Curve

- Prawn (Blue Line): The F1 score for the prawn class seems to be high across all confidence levels. This suggests that the model is doing a good job of correctly identifying prawns and not misclassifying other objects as prawns.
- Square (Orange Line): The F1 score for the square class also appears to be high across all confidence levels. This indicates that the model is accurately identifying squares and not confusing them with other objects.
- All Classes (Thick Blue Line): The F1 score for all classes combined reaches 1.00 at a confidence level of 0.759. This is an excellent result, suggesting that the model is performing well overall.

6.1.4 Recall Curve

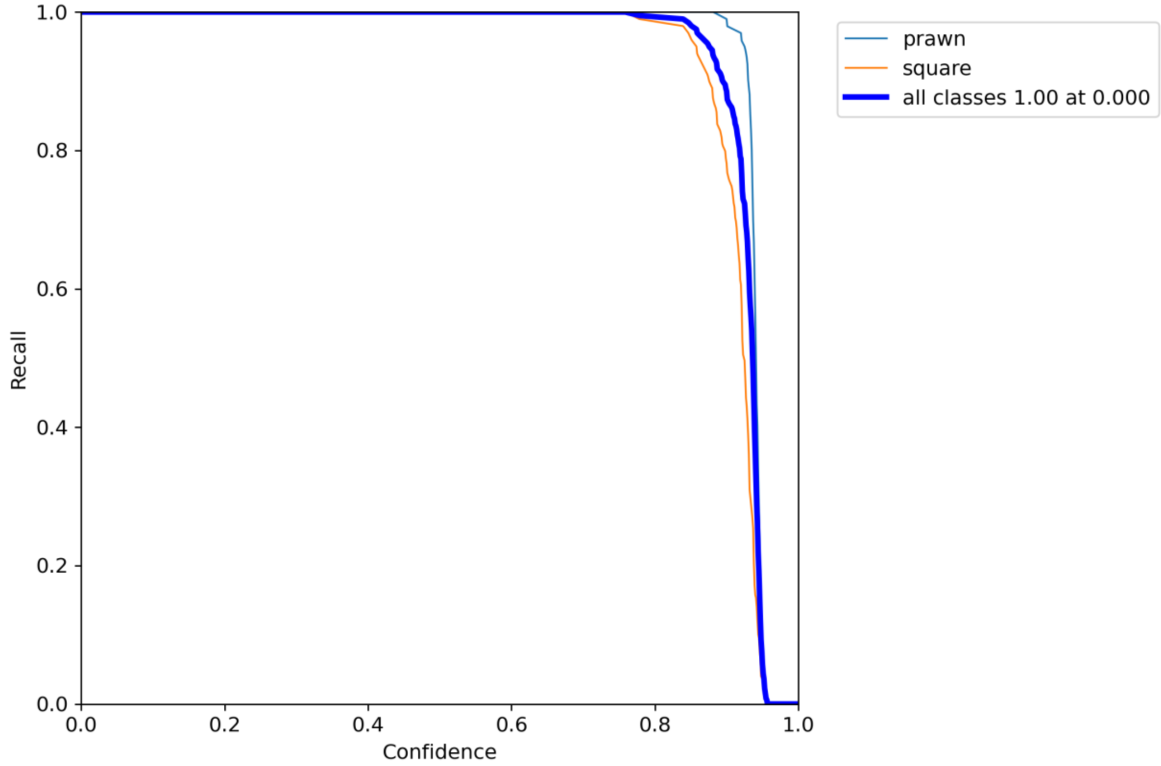


Figure 6.1.4 Recall Curve

- Prawn (Blue Line): This line represents the recall for the prawn class at different confidence levels.
- Square (Orange Line): This line represents the recall for the square class at different confidence levels.
- All Classes (Thick Blue Line): This line represents the recall for all classes combined at different confidence levels. The note on the right side indicates that for "all classes" a recall of 1.00 is achieved at a confidence level of 0.000.

From the graph, it appears that the model has high recall for all classes at certain confidence levels. This means that the model is able to correctly identify a high proportion of prawns and squares from the total actual prawns and squares in the dataset. The recall for all classes combined also reaches 1.00, indicating that the model performs well overall.

6.1.5 Precision Curve

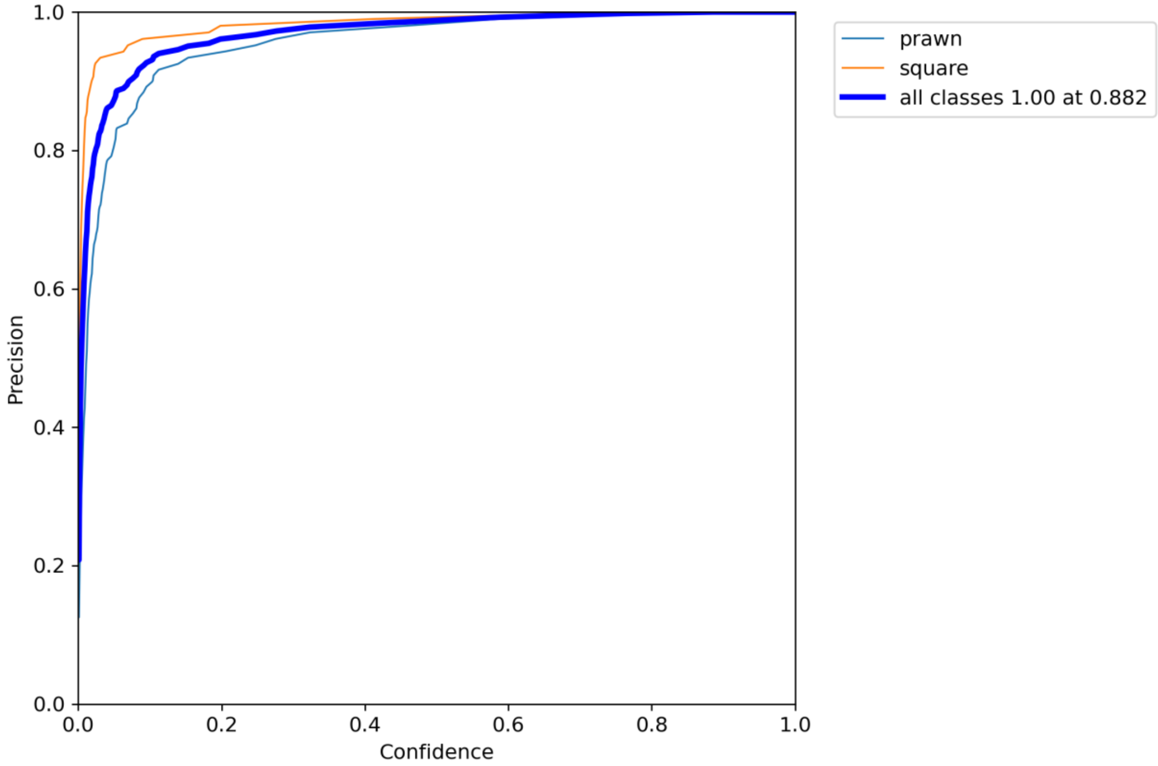


Figure 6.1.5 Precision Curve

- Prawn (Blue Line): This line represents the precision for the prawn class at different confidence levels.
- Square (Orange Line): This line represents the precision for the square class at different confidence levels.
- All Classes (Thick Blue Line): This line represents the precision for all classes combined at different confidence levels. The note on the right side indicates that for "all classes" a precision of 1.00 is achieved at a confidence level of 0.882.

From the graph, it appears that the model has high precision for all classes at certain confidence levels. This means that the model is able to correctly identify a high proportion of prawns and squares from the total predicted prawns and squares. The precision for all classes combined also reaches 1.00, indicating that the model performs well overall.

6.2 Testing Setup and Result

To thoroughly test the developed system, the project team first needs to set up the appropriate testing environment. This involves leveraging the capabilities of the Xiaomi camera viewer, a specialized software tool that enables the live streaming of video from the camera installed within the precision aquaculture system. By opening the Xiaomi camera viewer, the real-time footage of the prawns and the reference square within the aquaculture tank can be captured. This live video stream serves as the primary input for the subsequent testing and evaluation processes. To facilitate the analysis, a script that automatically takes screenshots of the Xiaomi camera viewer window at regular intervals is executed. These screenshots capture the visual information that will be processed by the length measurement and density and population estimation algorithms.

```
import pyautogui
import time
import os

# Function to capture screenshot
def capture_screenshot(filename):
    screenshot = pyautogui.screenshot()
    screenshot.save(filename)

# Function to automate screenshot capture for a duration
def automate_screenshots(duration, interval, output_dir):
    num_images = int(duration * 60 / interval)  # Convert duration to minutes
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)
    for i in range(num_images):
        filename = os.path.join(output_dir, f"screenshot_{i+1}.png")
        capture_screenshot(filename)
        time.sleep(interval)

# Set duration (in minutes) and interval (in seconds) for capturing screenshots
duration_minutes = 5
interval_seconds = 60

# Specify output directory
output_directory = "screenshots_test"

# Automate screenshot capture
automate_screenshots(duration_minutes, interval_seconds, output_directory)
```

Figure 6.2.1 Code to Screenshot

Figure 6.2.2 Example of Screenshot Image

After capturing the necessary screenshots from the Xiaomi camera viewer, these images are stored in a dedicated folder on the local file system. Then the source code to run the evaluation is updated, which it needs to point to the folder containing the captured screenshots. By modifying the file paths and input sources, the team ensures that the automated detection, measurement, and estimation algorithms can now access the relevant data for processing and analysis.

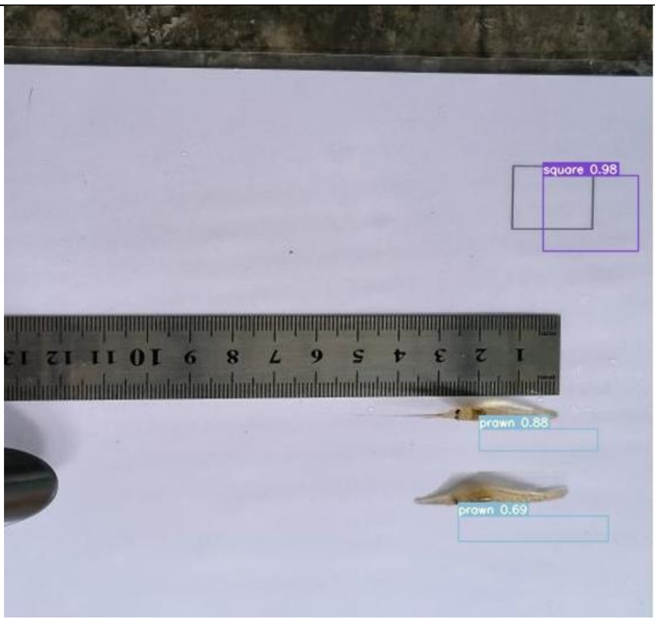
```
%cd C:\Users\xxiao\Desktop\xw\jupyter\yolov7_custom\content\yolov7
# Run evaluation
#!python detect.py --weights runs/train/exp/weights/best.pt --conf 0.1 --source C:\Users\xxiao\Desktop\xw\jupyter\yolov7_custom\content\yolov7\prawn_yolo
!python detect.py --weights runs/train/exp/weights/best.pt --conf 0.1 --source C:\Users\xxiao\Desktop\xw\jupyter\yolov7_custom\screenshots_test
```

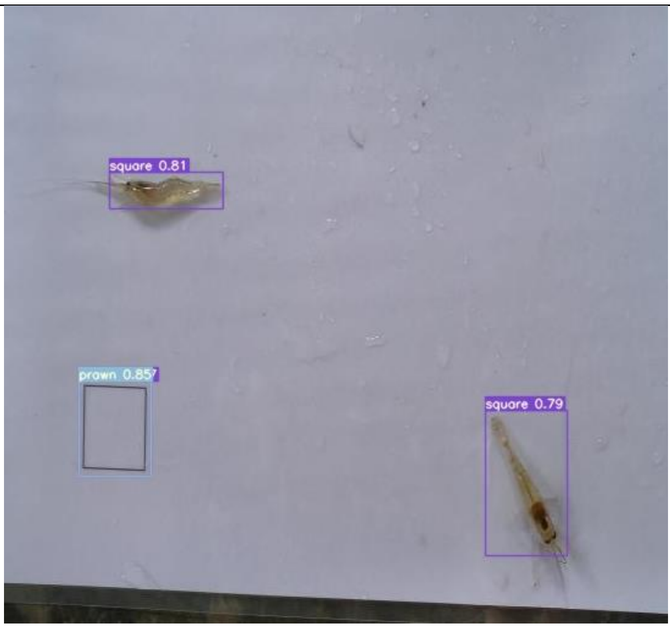
Figure 6.2.3 Code to Evaluate

The following is the result after running the evaluation.

No	Test Case	Remark

1	 **Figure 6.2.4 Test Image I**	All the prawns are predicted correctly with the drawn bounding box.
2	 **Figure 6.2.5 Test Image II**	There is only a prawn and a square, the prawn overlapped with square. However the model predicted wrongly, which the square predicted is actually the prawn and the prawn predicted is actually the square.

3	 **Figure 6.2.6 Test Image III**	This is the perfect case where the square and the prawn are predicted correctly.
4	 **Figure 6.2.7 Test Image IV**	For this case, there is a square and two prawns. The model predicted one square and two prawns, but however the predicted objects is not bounded correctly.

5	 **Figure 6.2.8 Test Image V**	There are two prawns and a square in the image. The model predicted the two prawns as square. And for the square, it actually having two bounding box, indicating the model predict it as square and prawn.
6	 **Figure 6.2.9 Test Image VI**	All the prawns are predicted correctly with correct bounding boxes.

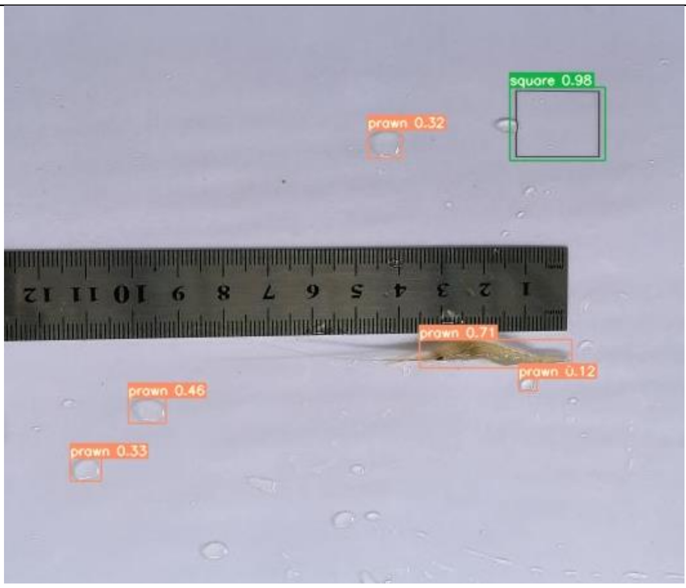
7	 **Figure 6.2.10 Test Image VII**	For this case, there is only one prawn and one square. The model predicted the background as the prawns, which is FP.

Table 6.2.1 Result on Testing

6.3 Project Challenges

One of the primary challenges faced by the project is the lack of reliable Wi-Fi connectivity within the farm area where the precision aquaculture system is situated. This lack of stable internet access poses a significant hurdle, as it can impede the seamless integration and real-time operation of the computer vision-based monitoring system.

To overcome this challenge, this project has devised a pragmatic approach to test the system's functionality and viability without relying on a continuous internet connection. The researcher focuses on verifying the detection capabilities of the system using the captured screenshots from the Xiaomi camera viewer, rather than attempting to deploy the full end-to-end solution in the farm environment immediately. The key to this testing strategy is to first assess the system's ability to accurately detect the reference square within the captured screenshots. If the model can reliably identify the presence and location of the square, it serves as a strong indicator that the overall system is functioning as intended, even in the absence of a live video stream and real-time data processing.

The next challenge is that the aquaculture environment is likely to be populated with various equipment and structures, such as feeding systems, water circulation pumps, and tank dividers, which can potentially interfere with the computer vision system's ability to accurately detect and measure the prawns. These extraneous objects may partially occlude the prawns or introduce visual distractions that can confuse the detection models.

To overcome this challenge, further investigation on techniques for object segmentation, image masking, or the incorporation of context-aware detection algorithms that can distinguish the prawns from the surrounding equipment and structures.

6.4 Objective Evaluation

The project has accomplished three main objectives that significantly improve the precision aquaculture of prawns. Firstly, it has successfully implemented computer vision techniques to accurately identify and differentiate prawns at different growth stages, providing valuable insights into their population dynamics and individual development within the aquaculture system.

Secondly, the project has utilized machine learning algorithms to automate the estimation of prawn density and overall population size, enhancing efficiency and scalability in monitoring compared to manual methods. This automated approach offers aquaculture operators a powerful tool to optimize the cultivation environment effectively.

Lastly, the project has seamlessly integrated the developed computer vision module into existing prawn farm infrastructure, bridging the gap between innovative technologies and practical aquaculture operations. This integration ensures that the enhanced monitoring and estimation capabilities directly contribute to improving management decisions and overall productivity in prawn cultivation efforts.

In summary, the successful completion of these key objectives has established a comprehensive solution that combines advanced computer vision, machine learning, and system integration to transform the way precision aquaculture of prawns is managed and conducted.

Chapter 7
Conclusion and Recommendation

7.1 Conclusion

In conclusion, this comprehensive project has successfully developed and implemented an innovative computer vision-based system to enhance the precision aquaculture of prawns. By leveraging advanced techniques in object detection, length measurement, and density & population estimation, the project has delivered a robust solution that addresses the critical challenges faced by prawn cultivation operators.

The ability to accurately detect and differentiate prawns across their various growth stages, from juveniles to adults, provides valuable insights into the population dynamics within the aquaculture system. This detailed understanding of the prawn population enables aquaculture managers to make informed decisions regarding feeding, water quality control, and other operational factors, ultimately optimizing the overall productivity and sustainability of the prawn cultivation efforts.

Furthermore, the integration of machine learning algorithms to automate the density and population estimation processes eliminates the need for manual monitoring, introducing a new level of efficiency and scalability to the aquaculture operations. The seamless deployment of the computer vision module within the existing prawn farm infrastructure demonstrates the project's ability to bridge the gap between innovative technologies and real-world applications, ensuring the widespread adoption and impact of the developed system.

With the successful completion of the project's key objectives, aquaculture operators now have access to a comprehensive, data-driven solution that can revolutionize the way they manage and optimize their prawn cultivation facilities. By enhancing the precision, accuracy, and automation of prawn monitoring and population estimation, this project paves the way for a new era of sustainable and profitable precision aquaculture, benefiting both the industry and the environment.

7.2 Recommendation

One key recommendation is to consider expanding the data collection and model training efforts to encompass a broader range of aquaculture environments and conditions. By gathering images and sensor data from multiple prawn farms, with varying water quality, lighting, and other environmental factors, the computer vision models can be trained to become more robust and adaptable. This increased generalization will ensure the solution can be seamlessly deployed across diverse aquaculture operations, maximizing its impact and scalability.

Another recommendation could be done is the integration of the computer vision module with other aquaculture management systems, such as water quality monitoring, feeding control and environmental regulation. This could create a truly holistic precision aquaculture platform. By establishing seamless data flows and control mechanisms between these various subsystems, the project can discover greater optimization opportunities, enabling aquaculture operators to make informed, data-driven decisions than enhance the overall efficiency and productivity of their prawn cultivation efforts.

REFERENCES

[1] "Fao.org," Countries | GLOBEFISH - Information and Analysis on World Fish Trade | Food and Agriculture Organization of the United Nations | GLOBEFISH | Food and Agriculture Organization of the United Nations, https://www.fao.org/in-action/globefish/countries/en/ (accessed Dec. 1, 2023).

[2] J. Kite-Powell, "Here's How This Company Uses AI To Grow Shrimp Sustainably," Forbes, Oct. 28, 2022. https://www.forbes.com/sites/jenniferhicks/2022/10/28/hereshow-this-company-uses-ai-to-grow-shrimp-sustainably/?sh=5dff926378f

[3] D. Li et al., "Automatic counting methods in aquaculture: A review," Journal of the World Aquaculture Society, vol. 52, no. 2. 2021. doi: 10.1111/jwas.12745.

[4] A. Setiawan, H. Hadiyanto, and C. E. Widodo, "Shrimp Body Weight Estimation in Aquaculture Ponds Using Morphometric Features Based on Underwater Image Analysis and Machine Learning Approach," *Revue d'Intelligence Artificielle*, vol. 36, no. 6, pp. 905–912, Dec. 2022, doi: 10.18280/ria.360611.

[5] T. T. N. Thai, T. S. Nguyen, and V. C. Pham, "Computer Vision Based Estimation of Shrimp Population Density and Size," in *Proceedings - 2021 International Symposium on Electrical and Electronics Engineering, ISEE 2021*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021, pp. 145–148. doi: 10.1109/ISEE51682.2021.9418638.

[6] Y. Zhou *et al.*, "Institute of Marine In-dustry Technology of Universities in Liaoning Province (2018-CY-34), Na-tional Natural Science Foundation of China (31972846), and China Post-doctoral Science Foundation (2018M640239)," 2019. [Online]. Available: http://odc.dlou.edu.cn/

[7] Y. Yang, B. Xue, L. Jesson, M. Wylie, M. Zhang, and M. Wellenreuther, "Deep Convolutional Neural Networks for Fish Weight Prediction from Images," in *International Conference Image and Vision Computing New Zealand*, IEEE Computer Society, 2021. doi: 10.1109/IVCNZ54163.2021.9653412.

[8] N. Bravata, D. Kelly, J. Eickholt, J. Bryan, S. Miehls, and D. Zielinski, "Applications of deep convolutional neural networks to predict length, circumference, and weight from mostly dewatered images of fish," *Ecol Evol*, vol. 10, no. 17, pp. 9313–9325, Sep. 2020, doi: 10.1002/ece3.6618.

[9] R. Bevans, "Multiple Linear Regression | A Quick Guide (Examples)," Scribbr, Jun. 22, 2023. Available: https://www.scribbr.com/statistics/multiple-linear-regression/. [Accessed: Apr. 7, 2024].

[10] M. Isaksson, "A Guide to Using U-Nets for Image Segmentation," Medium, Oct. 8, 2021. Available: https://martinisaksson.medium.com/a-guide-to-using-u-nets-for-image-segmentation-4799410c8aef [Accessed: Apr.7,2024]

[11] ArcGIS API for Python, “How U-net works?” Available: https://developers.arcgis.com/python/guide/how-unet-works/ [Accessed: Apr.7,2024]

[12] H. Zhou, R. Zhang, X. He, N. Li, Y. Wang, and S. Shen, “MCEENet: Multi-Scale Context Enhancement and Edge-Assisted Network for Few-Shot Semantic Segmentation,” Sensors, vol. 23, no. 6, 2023. [Online]. https://www.mdpi.com/1424-8220/23/6/2922

[13] T. Li, Z. Cui, Y. Han, G. Li, M. Li, and D. Wei, “Enhanced multi-scale networks for semantic segmentation,” Complex & Intelligent Systems, 2023. [Online] https://link.springer.com/article/10.1007/s40747-023-01279-x

[14] OpenGenus IQ, “Architecture of DenseNet-121,” [Online]. https://iq.opengenus.org/architecture-of-densenet121/

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708, 2017. [Online]. https://arxiv.org/abs/1608.06993

[16] MATLAB, “Train Convolutional Neural Network for Regression,” [Online]. https://www.mathworks.com/help/deeplearning/ug/train-a-convolutional-neural-network-for-regression.html

[17] Run.ai, “Understanding Deep Convolutional Neural Networks,” [Online]. https://www.run.ai/guides/deep-learning-for-computer-vision/deep-convolutional-neural-networks

[18] G. Boesch, “YOLOv7: The Most Powerful Object Detection Algorithm (2022 Guide),” *viso.ai*, Aug. 11, 2022. https://viso.ai/deep-learning/yolov7-guide/

[19] “YOLOv7 Paper Explanation: Object Detection and YOLOv7 Pose,” Aug. 02, 2022. https://learnopencv.com/yolov7-object-detection-paper-explanation-and-inference/

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 2**
Student Name & ID: Chong Xiao Wei & 20ACB03300	
Supervisor: Ts Dr Cheng Wai Khuen	
Project Title: Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Review the project according to last trimester achievement.
- Plan what tasks to be done.

2. WORK TO BE DONE

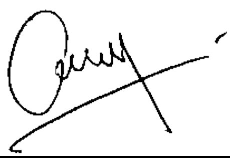
- Label newly acquired images.

3. PROBLEMS ENCOUNTERED

- -

4. SELF EVALUATION OF THE PROGRESS

- Slightly behind the progress.



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 4**
Student Name & ID: Chong Xiao Wei & 20ACB03300	
Supervisor: Ts Dr Cheng Wai Khuen	
Project Title: Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Label on the new images.
- Build train environment.

2. WORK TO BE DONE

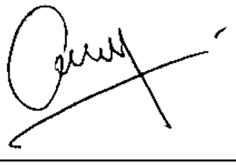
- Train the model.
- Code for the length measurement.

3. PROBLEMS ENCOUNTERED

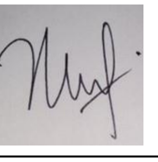
- The environment is slightly hard to build.

4. SELF EVALUATION OF THE PROGRESS

- Shall put more effort on development.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 6**
Student Name & ID: Chong Xiao Wei & 20ACB03300	
Supervisor: Ts Dr Cheng Wai Khuen	
Project Title: Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Trained the model.
- Code for length measurement is done.

2. WORK TO BE DONE

- Modify the code to measure multiple prawns.
- Code for density and population estimation.

3. PROBLEMS ENCOUNTERED

- The length measured is only for a prawn, not multiple prawns.

4. SELF EVALUATION OF THE PROGRESS

- Slightly behind the track, need to put more efforts.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 8**
Student Name & ID: Chong Xiao Wei & 20ACB03300	
Supervisor: Ts Dr Cheng Wai Khuen	
Project Title: Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques	

1. WORK DONE [Please write the details of the work done in the last fortnight.] • Solved the issue to measure multiple prawns. • 30% coding for density and population estimation.
2. WORK TO BE DONE • Continue the density and population estimation part. • Test in the real farm.
3. PROBLEMS ENCOUNTERED • -
4. SELF EVALUATION OF THE PROGRESS • On track, still need to put more efforts.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 10**
Student Name & ID: Chong Xiao Wei & 20ACB03300	
Supervisor: Ts Dr Cheng Wai Khuen	
Project Title: Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Done density and population estimation part.
- Tested the program is workable.

2. WORK TO BE DONE

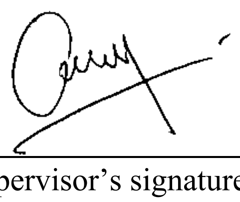
- Try install the camera in the place with Wi-Fi.
- FYP Report

3. PROBLEMS ENCOUNTERED

- The real farm area does not have Wi-Fi connection.

4. SELF EVALUATION OF THE PROGRESS

- Progress is on track.



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 12**
Student Name & ID: Chong Xiao Wei & 20ACB03300	
Supervisor: Ts Dr Cheng Wai Khuen	
Project Title: Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Installed camera in Wi-Fi area and tested.
- 60% of FYP report.

2. WORK TO BE DONE

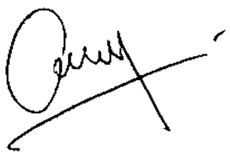
- Continue doing the FYP report.

3. PROBLEMS ENCOUNTERED

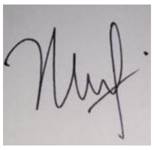
- -

4. SELF EVALUATION OF THE PROGRESS

- Progress is on track, able to submit FYP report on time.

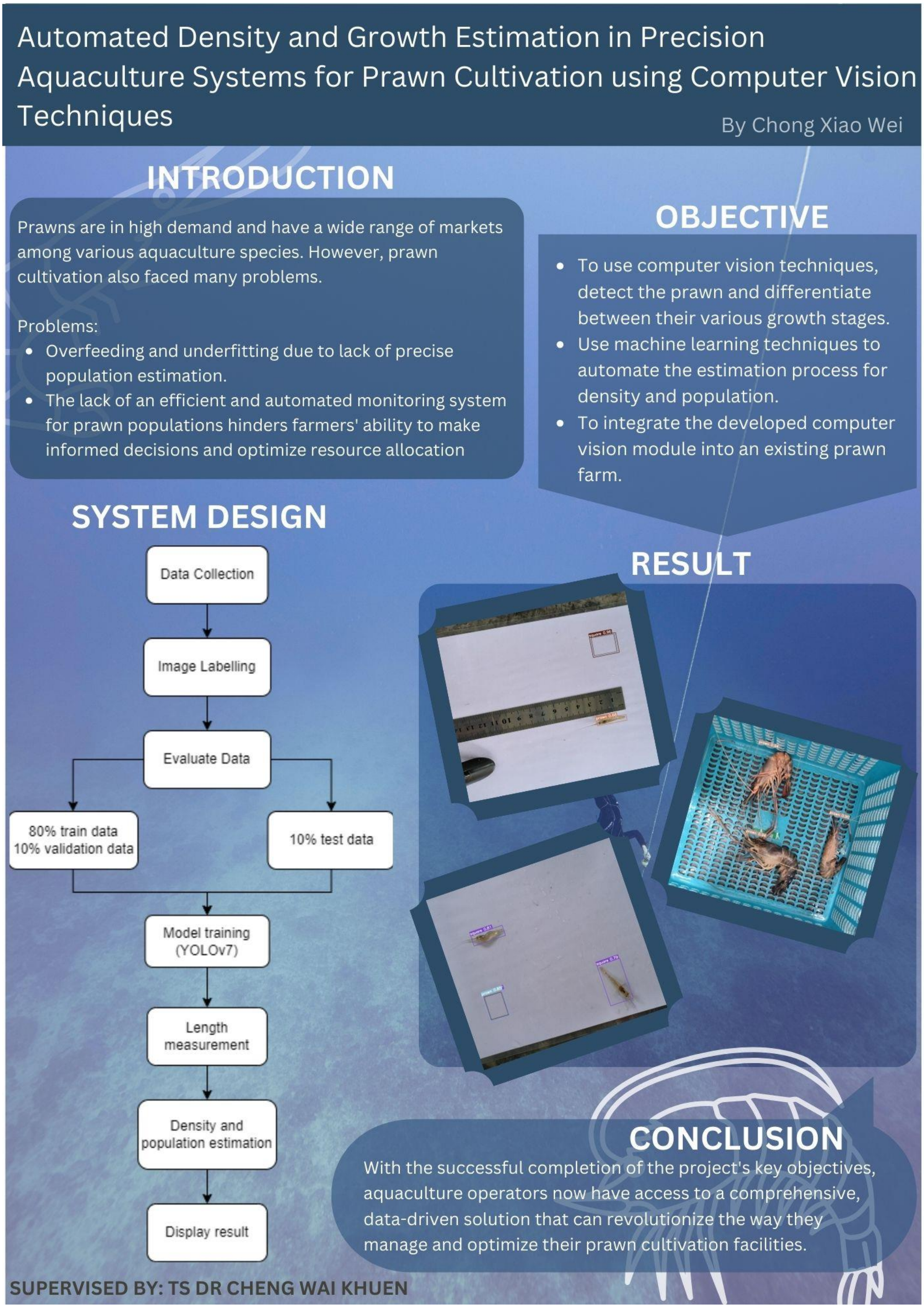


Supervisor's signature



Student's signature

POSTER


Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer Vision Techniques
By Chong Xiao Wei
INTRODUCTION
Prawns are in high demand and have a wide range of markets among various aquaculture species. However, prawn cultivation also faced many problems.
Problems:
• Overfeeding and underfitting due to lack of precise population estimation.
• The lack of an efficient and automated monitoring system for prawn populations hinders farmers' ability to make informed decisions and optimize resource allocation
OBJECTIVE
• To use computer vision techniques, detect the prawn and differentiate between their various growth stages.
• Use machine learning techniques to automate the estimation process for density and population.
• To integrate the developed computer vision module into an existing prawn farm.
SYSTEM DESIGN
Data Collection
Image Labelling
Evaluate Data
80% train data
10% validation data
10% test data
Model training
(YOLOv7)
Length
measurement
Density and
population estimation
Display result
RESULT
CONCLUSION
With the successful completion of the project's key objectives, aquaculture operators now have access to a comprehensive, data-driven solution that can revolutionize the way they manage and optimize their prawn cultivation facilities.
SUPERVISED BY: TS DR CHENG WAI KHUEN

PLAGIARISM CHECK RESULT

ORIGINALITY REPORT

14%	12%	9%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	1%
2	www.ncbi.nlm.nih.gov Internet Source	1%
3	doctorpenguin.com Internet Source	1%
4	fastercapital.com Internet Source	1%
5	www.iieta.org Internet Source	1%
6	Trung Thanh N. Thai, Thanh Sang Nguyen, Viet Cuong Pham. "Computer Vision Based Estimation of Shrimp Population Density and Size", 2021 International Symposium on Electrical and Electronics Engineering (ISEE), 2021 Publication	1%
7	Arif Setiawan, Hadiyanto Hadiyanto, Catur E. Widodo. "Shrimp Body Weight Estimation in Aquaculture Ponds Using Morphometric	<1%

PLAGIARISM CHECK RESULT

	Features Based on Underwater Image Analysis and Machine Learning Approach", Revue d'Intelligence Artificielle, 2022 Publication	
8	Submitted to Cerritos College Student Paper	<1 %
9	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1 %
10	www.nature.com Internet Source	<1 %
11	open-innovation-projects.org Internet Source	<1 %
12	Yizhi Zhou, Hong Yu, Junfeng Wu, Zhen Cui, Hongshuai Pang, Fangyan Zhang. "Fish Density Estimation with Multi-Scale Context Enhanced Convolutional Neural Network", Journal of Communications and Information Networks, 2019 Publication	<1 %
13	www.vernon.eu Internet Source	<1 %
14	Submitted to University of Hertfordshire Student Paper	<1 %
15	scholarworks.aub.edu.lb Internet Source	<1 %

PLAGIARISM CHECK RESULT

16	Submitted to October University for Modern Sciences and Arts (MSA) Student Paper	<1 %
17	Yunhan Yang, Bing Xue, Linley Jesson, Matthew Wylie, Mengjie Zhang, Maren Wellenreuther. "Deep Convolutional Neural Networks for Fish Weight Prediction from Images", 2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ), 2021 Publication	<1 %
18	Submitted to University of East London Student Paper	<1 %
19	www.infocomm-journal.com Internet Source	<1 %
20	hsr-journal.com Internet Source	<1 %
21	crimsonpublishers.com Internet Source	<1 %
22	arxiv.org Internet Source	<1 %
23	escholarship.org Internet Source	<1 %
24	Submitted to University of Hull Student Paper	<1 %

PLAGIARISM CHECK RESULT

25	discovery.researcher.life Internet Source	<1 %
26	www.medrxiv.org Internet Source	<1 %
27	Nagaraju Y, Venkatesh, Rajani G, Sathish Basapur. "BacilliFinder: Revolutionizing tuberculosis detection with computer vision", Indian Journal of Tuberculosis, 2023 Publication	<1 %
28	ebin.pub Internet Source	<1 %
29	kruthikareddymth522.sites.umassd.edu Internet Source	<1 %
30	Submitted to Bournemouth University Student Paper	<1 %
31	Princy K. Saji, K. C. Krishnachalitha, M. Kannan. "chapter 5 Multi-Modal MRI Images Analysis for Improved Herniated Disc Diagnosis Using Deep Learning", IGI Global, 2024 Publication	<1 %
32	Submitted to The University of Memphis Student Paper	<1 %
33	mobt3ath.com Internet Source	<1 %

PLAGIARISM CHECK RESULT

34	onlinelibrary.wiley.com Internet Source	<1 %
35	pdfs.semanticscholar.org Internet Source	<1 %
36	scholar.ppu.edu Internet Source	<1 %
37	www.researchsquare.com Internet Source	<1 %
38	Submitted to KIET Group of Institutions, Ghaziabad Student Paper	<1 %
39	Submitted to viit Student Paper	<1 %
40	"Soft Computing for Problem Solving", Springer Science and Business Media LLC, 2019 Publication	<1 %
41	in.pcmag.com Internet Source	<1 %
42	www.chromecasthelpline.com Internet Source	<1 %
43	Submitted to University of Carthage Student Paper	<1 %
44	Submitted to University of Essex Student Paper	<1 %

PLAGIARISM CHECK RESULT

45	agrifutures.com.au Internet Source	<1 %
46	www2.mdpi.com Internet Source	<1 %
47	Submitted to ICL Education Group Student Paper	<1 %
48	repository.essex.ac.uk Internet Source	<1 %
49	www.speech.kth.se Internet Source	<1 %
50	conf.uni-obuda.hu Internet Source	<1 %
51	downloads.hindawi.com Internet Source	<1 %
52	repository.uantwerpen.be Internet Source	<1 %
53	Submitted to University of Birmingham Student Paper	<1 %
54	dspace.cvut.cz Internet Source	<1 %
55	studymoose.com Internet Source	<1 %
56	repositorium.sdum.uminho.pt Internet Source	<1 %

PLAGIARISM CHECK RESULT

57	"Advances in Intelligent Systems and Computing V", Springer Science and Business Media LLC, 2021 Publication	<1 %
58	Submitted to Letterkenny Institute of Technology Student Paper	<1 %
59	deepai.org Internet Source	<1 %
60	hdl.handle.net Internet Source	<1 %
61	journals.ametsoc.org Internet Source	<1 %
62	researchonline.jcu.edu.au Internet Source	<1 %
63	www.diva-portal.org Internet Source	<1 %
64	www.ijees.net Internet Source	<1 %
65	www.scilit.net Internet Source	<1 %

PLAGIARISM CHECK RESULT

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	**CHONG XIAO WEI**
ID Number(s)	**20ACB03300**
Programme / Course	**Bachelor of Computer Science (Honours)**
Title of Final Year Project	**Automated Density and Growth Estimation in Precision Aquaculture Systems for Prawn Cultivation using Computer**

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 14 **%** **Similarity by source** Internet Sources: 12 % Publications: 9 % Student Papers: 5 %	OK
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: Ts Dr Cheng Wai Khuen

Date: 22/04/2024

____
Signature of Co-Supervisor

Name: ______________________

Date: ______________________

PLAGIARISM CHECK RESULT

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB03300
Student Name	Chong Xiao Wei
Supervisor Name	Ts Dr Cheng Wai Khuen

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____
(Signature of Student)
Date: 22/4/2024