



TECHNISCHE UNIVERSITÄT  
CHEMNITZ

# **Determination of Biomass in Shrimp-Farm using Computer Vision**

## **Master Thesis**

Submitted in Fulfilment of the  
Requirements for the Academic Degree  
M.Sc.

Dept. of Computer Science  
Chair of Computer Engineering

Submitted by: Gowtham Chowdary TAMMINENI

Student ID: 456818

Date: 23/01/2023

University supervisor: Prof. Dr. W. Hardt

Dr. Ariane Heller

External supervisor: M.Sc. Jansen Kariantio Tjong

# Abstract

The automation in the aquaculture is proving to be more and more effective these days. The economic drain on the aquaculture farmers due to the high mortality of the shrimps can be reduced by ensuring the welfare of the animals. The health of shrimps can decline with even barest of changes in the conditions in the farm. This is the result of increase in stress. As shrimps are quite sensitive to the changes, even small changes can increase the stress in the animals which results in the decline of health. This severely dampens the mortality rate in the animals.

Also, human interference while feeding the shrimps severely induces the stress on the shrimps and thereby affecting the shrimp's mortality. So, to ensure the optimum efficiency of the farm, the feeding of the shrimps is made automated. The underfeeding and overfeeding also affects the growth of shrimps. To determine the right amount of food to provide for shrimps, Biomass is a very helpful parameter.

The use of artificial intelligence (AI) to calculate the farm's biomass is the project's primary area of interest. This model uses the cameras mounted on top of the tank at densely populated areas. These cameras monitor the farm, and our model detects the biomass. By doing so, it is possible to estimate how much food should be distributed at that particular area. Biomass of the shrimps can be calculated with the help of the number of shrimps and the average lengths of the shrimps detected. With the reduced human interference in calculating the biomass, the health of the animals improves and thereby making the process sustainable and economical.

Keywords: Biomass estimation, Object detection, YOLOv7, Mask R-CNN, U-Net.

# Acknowledgement

I am incredibly appreciative of all the wonderful people in my life who have helped and supported me. I wouldn't be able to be here without them. For their affection and support, especially during these trying times, I am very grateful.

MonitorFish GmbH has been very supportive in the project, and I would like to thank the team for giving me this opportunity. I am extremely grateful to Mr. Jansen Tjong, our senior manager, for always trusting us and encouraging us in achieving our goals. Mr. Jansen Tjong not only gave us the required help, but he also went out of his way to make us feel comfortable. Additionally, I would like to thank Mr. Jan Apel, our technical lead and Mr. Ralph Apel, our machine learning specialist for their help.

I am extremely grateful to Prof. Dr. Dr. h. c. Wolfram Hardt and the department of computer science for agreeing to supervise my thesis. I am also very grateful for my supervisor Dr. Ariane Heller for giving me a chance. I am very thankful for her time and support.

# About the Company

MonitorFish GmbH is a startup business which focuses mainly on the AI side of the aquafarming. The company is heavily involved in development of projects like ANFISH and OptoFish. These programs are designed to help aqua farmers in maintaining the health of the animals to ensure optimum production.

The company also focuses on underwater cameras to keep track of the animals along with water quality. With ever-increasing global population and the need for protein intake, the consumption of fish is heavily increased. This leads to over-fishing which affects the marine ecosystem and reduces the fish population. One strategy to stop overfishing is to practice sustainable fish farming. But one of the main issues of the fish farming is maintaining the health of the animals. As the establishing of the farm is expensive, the volatile nature of the farming may induce heavy losses. So, keeping the farm sustainable and the animals healthy is very important.



Figure 0-1: MonitorFish logo [1]

MonitorFish was established to help overcome the challenges associated with fish farming. We provide wise and sustainable management to help in dealing the difficulties associated with fish farming. Our digital assistance makes local fish farms sustainable and profitable with the help of computer vision techniques. Thereby giving time for our seas the time to thrive [1]

# Table of content

Abstract .....	1
Acknowledgement .....	2
About the Company.....	3
Table of content.....	4
List of Figures.....	6
List of Tables .....	8
List of Abbreviations .....	9
1 Introduction .....	10
1.1 Motivation.....	11
1.2 Problem Statement.....	12
1.3 Overview of following chapters.....	13
2 Technical background .....	15
2.1 Shrimp.....	15
2.2 Farm setup .....	17
2.3 Basics of image understanding .....	20
2.4 Computer vision .....	22
3 State of the Art .....	31
3.1 Polygon mask detection (Mask R-CNN).....	31
3.2 Key-point detection.....	33
3.3 Bounding box detection.....	35
4 Concept.....	37
4.1 Data processing .....	38
4.2 Number of shrimps.....	38
4.3 Length detection.....	39
4.4 Biomass determination.....	40
5 Implementation and Methodology .....	42
5.1 Data processing .....	42
5.2 Data collection.....	42

5.3	Data preparing .....	48
5.4	Weight estimation.....	48
5.5	Number estimation .....	55
5.6	Biomass estimation .....	62
6	Results and evaluation .....	64
6.1	Segmentations .....	64
6.2	YOLOv7 results.....	66
6.3	Density map results.....	68
6.4	Comparison between Density map and YOLOv7 .....	70
6.5	Biomass determination.....	71
7	Conclusion and Future Scope .....	74
8	References.....	76
	Appendix .....	80

# List of Figures

Figure 0-1: MonitorFish logo [1].....	3
Figure 1-1: Different kinds of Aquafarming .....	10
Figure 2-1: Shrimp Appearance .....	15
Figure 2-2: Top and side view of Shrimp [18].....	16
Figure 2-3: Length measurement of Shrimp .....	17
Figure 2-4: View of the farm .....	18
Figure 2-5: Day/Night situation at farm .....	18
Figure 2-6: View from the installed camera .....	19
Figure 2-7: Pixel details in a grey scale image .....	21
Figure 2-8: Coordinates of a 2D image [19] .....	21
Figure 2-9: CNN architecture example [24] .....	24
Figure 2-10: Convolution layer [25] .....	25
Figure 2-11: Max-pooling with 2 x 2 filter [26].....	26
Figure 2-12:R-CNN architecture [19].....	28
Figure 2-13; Segmentation output [30] .....	28
Figure 2-14: Instance vs Semantic segmentaion [42].....	29
Figure 3-1: Mask R-CNN framework for instance segmentation [32] .....	31
Figure 3-2: Object detection results [32].....	32
Figure 3-3: Mask prediction [32] .....	32
Figure 3-4: Keypoint detection results [32] .....	32
Figure 3-5: U-Net Architecture [40].....	33
Figure 3-6: DeepLabCut workflow [34] .....	35
Figure 3-7: YOLOv7 performance [41] .....	36
Figure 4-1: Flowchart of proposed concept .....	37
Figure 5-1:Proposed test model .....	43
Figure 5-2: Actual test setup.....	44
Figure 5-3: Sample from different channels .....	44
Figure 5-4: Setup with varied intensity and exposure .....	45
Figure 5-5: : Overlapping of shrimps [33] .....	46
Figure 5-6: Image during resting time.....	47
Figure 5-7: Image during feeding time.....	47
Figure 5-8:Annotation .....	50
Figure 5-9: Training loss of Mask R-CNN model .....	51
Figure 5-10: Visualisation on validation images .....	51

Figure 5-11: Sample image output .....	52
Figure 5-12: Grid measurement [33] .....	52
Figure 5-13: Contour of the masks .....	53
Figure 5-14: Contour of single shrimp .....	53
Figure 5-15: Extreme points in the contour .....	54
Figure 5-16: Annotations for YOLO .....	56
Figure 5-17: YOLOv7 output with bounding boxes .....	57
Figure 5-18: YOLO output with coordinates of bounding box .....	58
Figure 5-19: Point annotation for U-Net.....	60
Figure 5-20: Input image and output with density map.....	61
Figure 6-1: Output of the segmentation .....	64
Figure 6-2: Shrimps detected in the images from different camera .....	65
Figure 6-3: Partial detection .....	65
Figure 6-4: length comparision .....	66
Figure 6-5; Second tset length comparision .....	66
Figure 6-6: Input and output of YOLOv7 model.....	67
Figure 6-7: YOLOv7 on test dataset.....	67
Figure 6-8: mAP calculation .....	68
Figure 6-9: Density map .....	69
Figure 6-10: U-Net output.....	70
Figure 6-11: Biomass displayed on the image-1 .....	71
Figure 6-12: Biomass displayed on the image-2 .....	72
Figure 9-0-1: Disease evidence in the shrimp [37] .....	83



# List of Tables

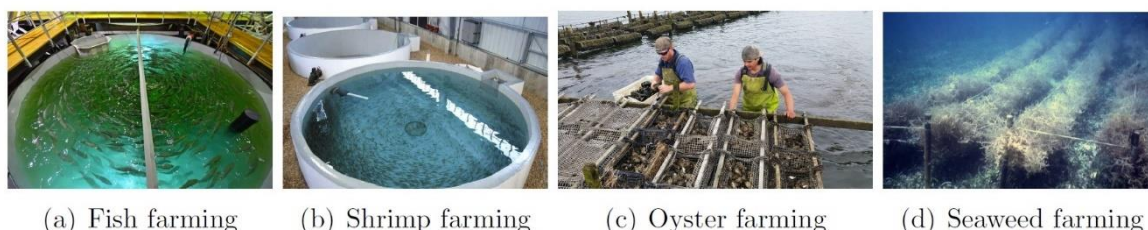
Table 5-1: Environmental setup for Mask R-CNN .....	49
Table 5-2: Environmental setup for YOLOv7 .....	55
Table 5-3: Environmental setup for U-Net .....	59
Table 6-1: Details of YOLOv7 on video .....	68
Table 6-2: Comparison between YOLOv7 and Density map .....	70
Table 6-3: Detection rate .....	72

# List of Abbreviations

- AI** : Artificial Intelligence
- ML** : Machine Learning
- DL** : Deep Learning
- CNN** : Convolution Neural Network
- R-CNN**: Region-based Convolution Neural Network
- ReLU** : Rectified Linear Unit
- RAS** : Recirculating aquaculture system
- ROIs** : Regions of Interest
- NMS** : Non-Maxima suppression
- RPN** : Region Proposal Network
- GPU** : Graphics Processing Unit
- mAP** : Mean Average Precision
- YOLO**: You Only Look Once
- CVAT** : Computer Vision Annotation Tool
- DLC** : DeepLabCut

# 1 Introduction

The role of artificial intelligence in making the farming sustainable and economically feasible is increasing over the last few years. Aquaculture is also known as aquafarming and It is a form of cultivation of aquatic animals in a controlled environment. This is one of the fastest growing food production sectors and it includes breeding, rearing, and harvesting of many types of aquatic animals such as fish (such as salmon, tilapia, trout and several species), mollusks (such as oysters, mussels and clams), crustaceans (lobsters, shrimps, crabs and crawfish) and seaweed farming. [2]



**Figure 1-1: Different kinds of Aquafarming**

Globally aquaculture is responsible for almost half of the seafood consumption, and it only increases from here on. Figure 1.1 shows the different kinds of farming in different kinds of environments. This increase in the farming capacity also helps in decreasing the overfishing and preserving the oceanic life.

Animal welfare deals with the health and well-being of the animals [3] . Humane treatment of the animals leads to healthy growth of the animals. Several factors related to treatment of animals like water quality, human interaction, nutrition, and environment play a significant role in maintaining a healthy aquafarm. In order to meet the ever-increasing requirements for animal protein, animal welfare plays a huge role in making sure of supply meeting the demands.

While taking care of animal welfare is important, it also has its difficulties. As the aquatic animals live under the water, the human interaction is limited compared to land-based animals. This provides a unique challenge in interpreting the facial and body movements. As the human interaction leads to stress which in-turn leads to degradation of health, observing the animals directly should be minimized. Here, cameras and AI is very helpful in monitoring the animals. There are several factors that affect the animal welfare and they are further discussed along with the different types of farming techniques in the appendix.

## 1.1 Motivation

The worldwide population is expected to reach 9.3 billion by 2050. Agriculture alone cannot sustain the population of the world. The major alternative source to plant based for feeding the world population is aquafarming. Aquafarming has been around for four thousand to five thousand years in some parts of the world [4]. However, its been around for fifty to sixty years as a major food industry.

The major natural resources of the world like land and water are already stretched thin. To accommodate the ever-increasing population and providing food sources in spite of looming climate change issues such as rising sea levels, aquafarming is going to be one of the alternative solutions. Given its significance in providing protein source, various technologies are used to increase the food harvested.

70 percent of the planet's surface is covered by water. Given its vast area, the potential of the aquafarming is significant, and humankind is aware of this for a long time. Aquaculture contributed 43 percent of aquatic animal food in 2007 [5] and it increased to more than 50 percent by 2019. This is projected to increase to 62 percent by 2030 [6]. Given the availability of vast amount of sea water, aquafarming is only going to increase.

To help the aquafarming to achieve its potential, technology should go hand in hand. Also, with technology advancing day by day, the need to implement the technologies to help tackle some of the challenges in fish farming also increases. With the progress of artificial intelligence (AI) and computer vision (CV) techniques, the issues associated with the fish farming may be resolved. These techniques also help provide the farmers with fresh perspective by monitoring the animal's movement and health. As the inspected animals are sensitive to any human interaction and any changes in the environment, implementing the computer vision to monitor the subjects and limiting the human interaction to as minimal as possible will minimize the stress [7]. Thereby, it guarantees the health of the animals.

To achieve the best production and sustainability, the feeding of shrimps plays a huge part. When the shrimps are overfed, the sediment formation increases. Thereby affecting the visibility in the water that leads to collision between animals and increase in stress. And when the shrimps are underfed, the growth of the animals is affected.

To avoid these, the animals should be fed appropriately and to know the amount of food needed, the biomass is the key factor.

Biomass is estimated by calculating the average weight of the shrimps and number of shrimps. The existing method of calculating the biomass in the farm is by doing it manually. For calculating the weight of the shrimps, they are taken out of the water and the length of the shrimps is measured individually. By using the length/weight conversion factor, the weight of the shrimps is measured. For calculating the number of shrimps, an image is taken at the shrimp tank and the number of animals in that place is calculated by counting manually.

This process of measuring manually puts a lot of stress on animals which can affect in their growth. It is also a tedious process and requires lot of manual labour. By making this process automotive, the stress on the animals will be very low and thereby not affecting the health of the shrimps. The manual measurement is also prone to mistakes such as the same shrimps being measured more than once affects the average. So, making it automotive is less intrusive and gives better results.

## **1.2 Problem Statement**

With the aquafarms being densely populated, the complexity in monitoring the animals also increases. Even though the traditional bounding box detection work well in detecting the animals, the high density to ensure maximum production becomes an issue. The overlapping of animals with each other, the distinction becomes an issue. The other main issue is the visibility of the animals in the water. With increase in sedimentation and the fast movement of the animals, the visibility goes down.

The task of estimating the biomass as accurate as possible in the shrimp farm have several challenges and this thesis aims to solve those. They are as follows:

- Detecting the shrimps:
  - a) Locating the shrimps in the image.
  - b) Isolating the shrimps from the noisy background.
  - c) Extracting the shrimp individually from the image to calculate the length.

- Density calculation: The number of shrimps in the image is calculated by generating a density map.
- Size estimation: The detected shrimps are separated individually, and the length of the shrimps is calculated. Using the length/weight factor, the average weight is then calculated.
- Biomass Estimation: This is the final step in this thesis. Using both the calculations: number of shrimps and the average weight, the biomass is calculated and then displayed on the image.

### **1.3 Overview of following chapters**

The thesis report has been structured into several chapters: from basics about the biology of shrimps to in-depth analysis of the deep learning algorithms and estimation of biomass of the shrimps. This chapter, Introduction deals with the basics in aquafarming, the motivation for this project and the challenges associated with it. This chapter also discusses about the problem statement for the thesis. The following chapters and their general overview are discussed below.

- Chapter 2- Technical background: As the name suggests, this chapter gives the basic background knowledge needed to understand the project. It gives basic information about shrimp and how the biomass is calculated. It also gives an idea about the farm setup and basics of computer vision.
- Chapter 3- State of the art: This chapter consists of the literature survey and the relevant research area topics related to this thesis. It mainly discusses about the different object detection techniques in depth.
- Chapter 4- Concept: This chapter provides the proposed concept in a step-by-step manner in the form of flowchart to achieve the goal of the thesis. It is a proposed solution to meet the objective of the thesis.
- Chapter 6- Implementation and methodology: This chapter deals with the implementation of the proposed concept. This chapter deals with the process of obtaining the best available parameters for data collection. It determines the parameters by conducting an experiment on a test setup. It also discusses about preparation of dataset with respect to the ML techniques. It also talks about the process of length calculation, number of shrimps estimation and biomass calculation.

- Chapter 7- Result and evaluation: The results from the previous chapter and what those results means are discussed in this chapter. Comparison between the different models is analysed here based on the output obtained by each model.
- Chapter 8- Conclusion and future scope: This is the final chapter and thus it gives the conclusion of the thesis work. It also briefly discusses about the future scope of the project based on the results obtained by this thesis.

### **Summary of the chapter**

This chapter gives an introduction to the aquafarming and the need for the thesis work. The need for improving the efficiency of the aquafarming is also discussed in this chapter. The advantages over the traditional method of biomass estimation are discussed. This chapter also deals with the motivation for this thesis and talks about some of the challenges in the project.

The last section gives an overview of all the chapters to follow in this report. The topics are discussed in depth in their respective chapters.

## 2 Technical background

The chapter discusses about the basic background knowledge needed for this project. The primary goal of this thesis is to determine the biomass of the shrimps in the shrimp-farm in order to make the farm sustainable and to improve the welfare of the shrimps. Reducing the food wastage and improving the mortality rate are the main goals of this project.

### 2.1 Shrimp

Larviculture in hatcheries and captive reproduction were necessary for the development of contemporary shrimp culture [8]. Shrimps are characterized by the semitransparent body flattened from side to side and a flexible abdomen terminating in a fanlike tail. Shrimps have long antennae and elongated body. Shrimps can be found in a variety of aquatic environments, including freshwater lakes, streams, and both shallow and deep marine bodies of water. Their appendages are modified for swimming, and they look like the following figure [9].



Figure 2-1: Shrimp Appearance

Figure 2-1 shows the physical appearance of a commonly found shrimp. These swim backwards by flexing the abdomen and tail rapidly.

#### 2.1.1 Length of the shrimps

The top and side view of a shrimp is shown in the Figure 2-2. While calculating length, there are different parameters to consider.



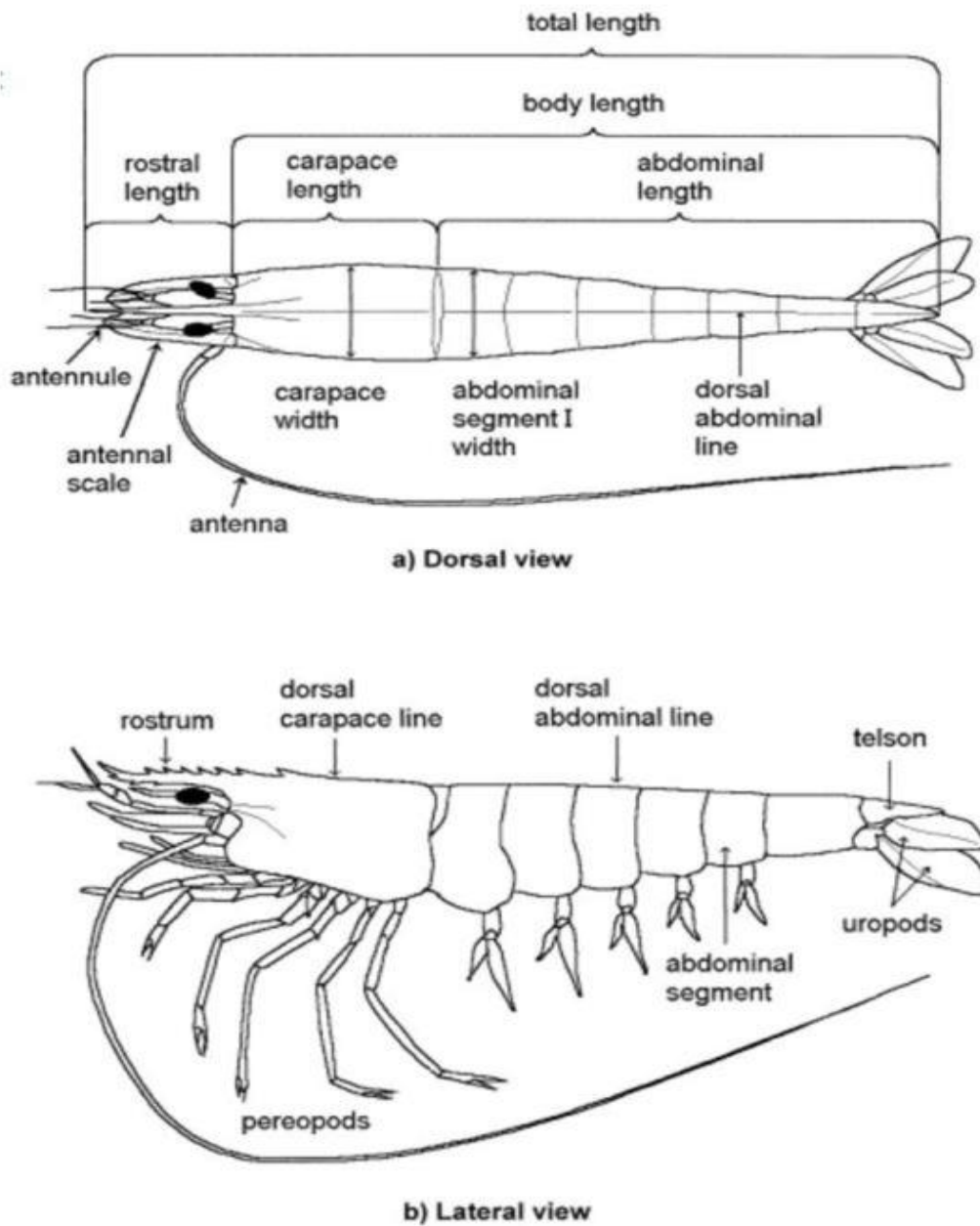


Figure 2-2: Top and side view of Shrimp [10]

While calculating the length of the shrimp, the length from the tip of the rostral length to the end of the tail is considered. The average length of the shrimps varies from region to region. The average European shrimps grow anywhere in between 5 cm to 25 cm [6]. The shrimps in this project varies from 5 cm to 25 cm. We calculate length of the shrimps and with that we calculate weight of the shrimps using length/weight conversion factor.

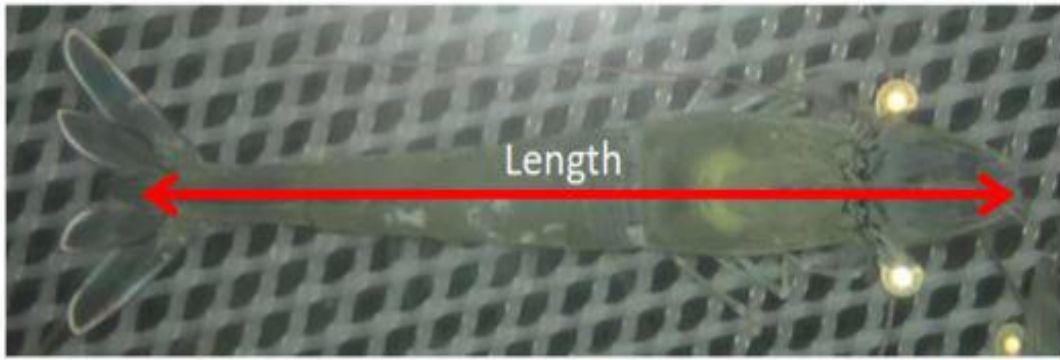


Figure 2-3: Length measurement of Shrimp

### 2.1.2 Biomass calculation

The project makes use of the length/weight conversion factor generated by the biology department employ of the company. The conversion factor is

$$W = a * TL^b \quad (2-1)$$

where W is weight in grams, a= 0.0047, TL is total length in cm and b = 3.1101.

With this relation factor, we can calculate average weight of the shrimps by calculating the average length of the shrimps. Furthermore, the biomass per area can be calculated by multiplying number of shrimps per area with the average weight of the shrimps detected.

$$B = W * N \quad (2-2)$$

Where B is for Biomass in grams, W stands for average weight in grams and N refers to number of shrimps detected.

## 2.2 Farm setup

### 2.2.1 Shrimp-farm

This project is conducted in partnership with Förde Garnelen GmbH & Co. KG. The shrimp-farm is located in the partner company. The shrimps are kept in five different tanks based on the weight range. Sections are divided as per the age class.

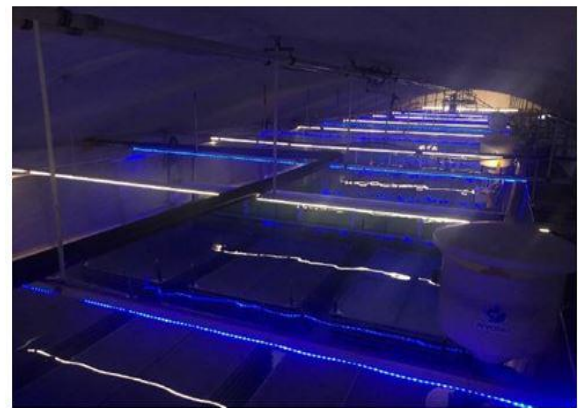
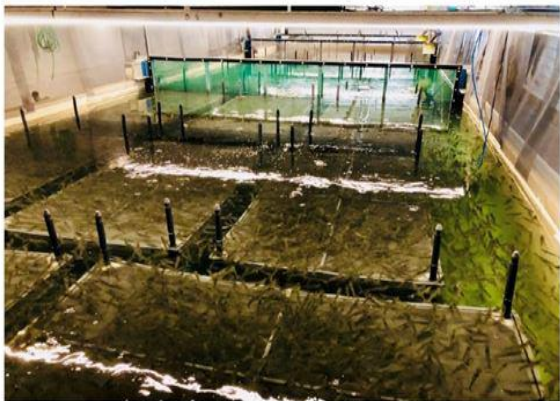
The top view of the farm can be seen in the figure 2-4. Once the shrimps in a tank exceeds the weight range then those are transferred to the next tank.



**Figure 2-4: View of the farm**

The shrimps are fed by calculating the biomass in the tank and dispensing the adequate amount of food. As the farm is in closed environment with high density, maintaining lighting in the farm is very important as it will affect the health of the shrimps.

As there are no windows in the farm, artificial lights are used in the farm to maintain the optimum light conditions. This shows the significant improvement in the growth and health of the shrimps. To create the ambience of day and night, RGBW-LED lights are utilized.



**Figure 2-5: Day/Night situation at farm**

Figure 2-5 shows the view of the farm in day and night conditions. For the project, the lights are in 12h day and 12h night pattern. According to our partner company, this setup yielded the best outcome with higher growth rate. This results in production cycle getting shorter and hence the best output.

The dimensions of the farm at Förde Garnelen: The farm has the pool in  $30m \times 5m = 150m^2$  floor area. This pool is then further divided into 5 sections with flexible partitions. Each section has its age class (cohort). Every month a section is filled and the section

with the oldest shrimps is harvested. This method is called multi-cohort stocking. Every section is flexible and can be expanded if needed. The fattening cycle that is the time from stocking to harvesting lasts from 4 to 5 months. This doesn't include the nursery time of 1 month. That means the entire harvest cycle takes between 150 and 180 days. The wall height is approximately 2.2m and the ceiling height is approximately 3.35m. Figure 2.4 shows the farm setting at Förde Garnelen with no windows and multiple sections having flexible space for expansion.

### 2.2.2 Camera setup

Cameras are setup overlooking the tank from top. Each layer in the tank's sections has a depth of 10 cm and is made up of numerous layers.. The shrimps are distributed evenly throughout the tank. The height of the camera position is 180cm from the water level.

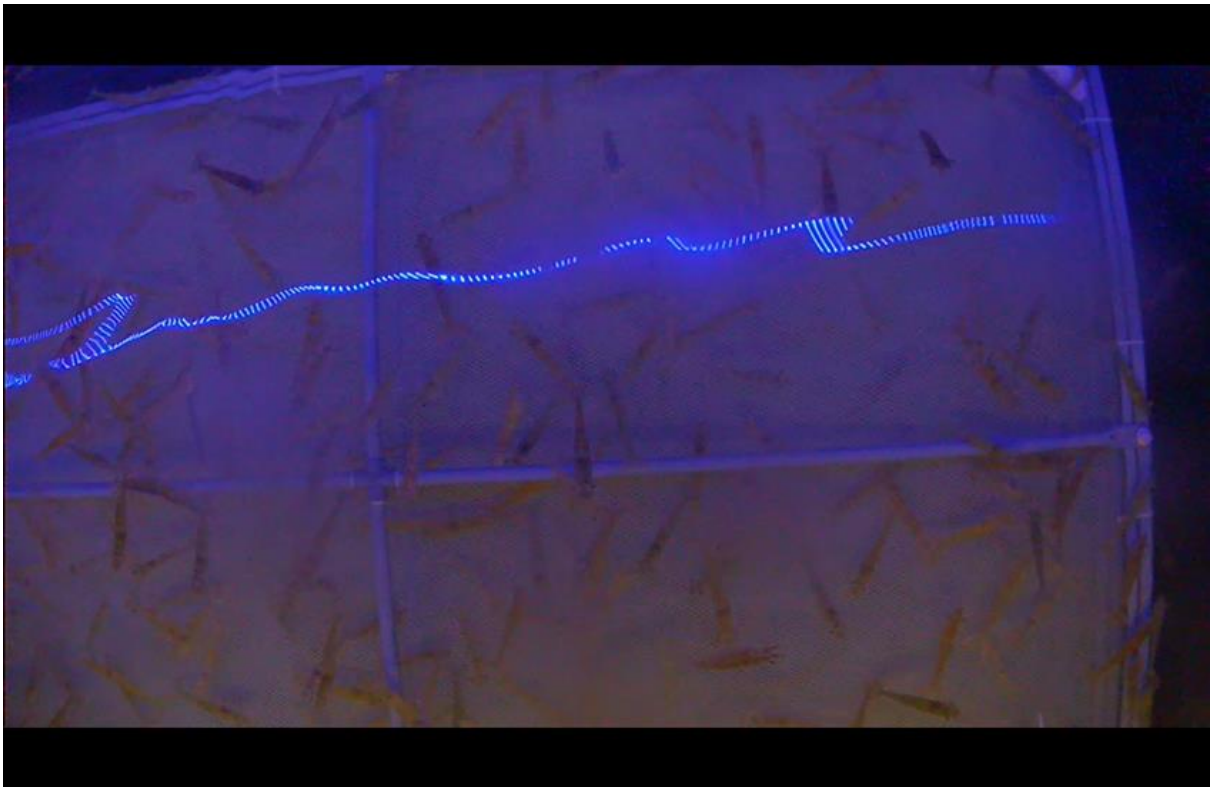


Figure 2-6: View from the installed camera

Figure 2-6 shows the view of the tank from the camera fixed above. Camera are fixed to the pipe going through the tank to which feeding machine is attached to. The height of the camera is fixed and the position of it can be altered as per the need. The cameras can also be placed under the water inside the tank, but the salt aerosols can damage the camera. Also, the damage to the camera when placed under water outweighs to

the benefits of placing the camera there. Also the durability of the camera will plummet by placing it under water without any IP 65 rating. The camera can be moved horizontal in both the direction with this setup to get the result as best as possible.

### *2.2.3 Cameras used*

Two types of cameras were considered in this project for monitoring the shrimps in the tank and they are a monocular camera and a stereo camera. The stereo camera can be used to gather 3D information while the monocular camera cannot. We can use stereo camera to get the depth information. Stereo cameras have two lenses as it tries to imitate the human vision. This gives the camera an ability to capture 3D information. Whereas monocular camera has only one lens and it is preferred when the depth estimation is not needed.

For this project, the depth of the water in each layer in the tank is considerably small than the height of the camera fixed, the need for depth information is considerably less. The stereo cameras can also be used to produce the normal image when we use only one lens. Since, the cameras already fixed in the farm are stereo, we use only one lens from the camera to get the images we need. The camera used in this project is a Nvidia Jetson stereo camera.

## **2.3 Basics of image understanding**

Image processing is a method of performing different operations/functions on an image to change or modify it to get the required information. Modern digital computers can perform wide range of operations on an image. It has a wide range of applications such as object detection, face recognition, optical character recognition, surveillance and so on. In recent days, the applications of this extends to almost all the major fields that include medical science, robotics and autonomous driving.

### **Properties of an Image**

An image is a 2D function  $f(x, y)$ , where  $(x, y)$  represents the spatial coordinates and  $f$  depends on the brightness intensity levels at that point [11]. A digital image is an array of all these, and the components of this array are referred to as pixels. Each pixel is a measure of intensity of light or luminance that falls on a camera sensor. The image coordinates for a 2D image can be seen in the figure 2-7.



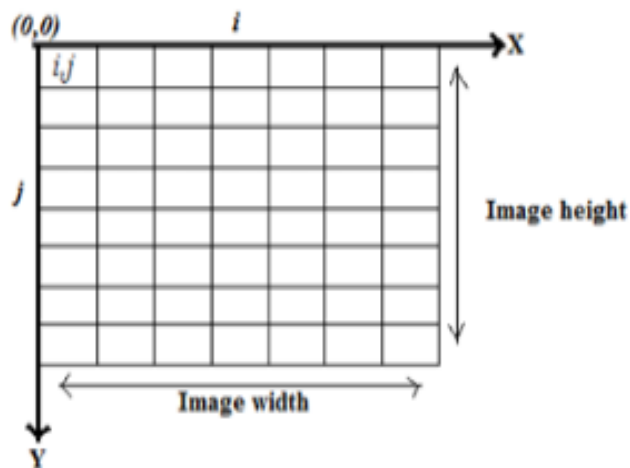


Figure 2-8: Coordinates of a 2D image [11]

An image is a matrix with pixel values as the elements as shown in the following figure 2-8. The pixel value ranges from 0 to 255 for a grey scale image with 0 being complete black to 255 being complete white.

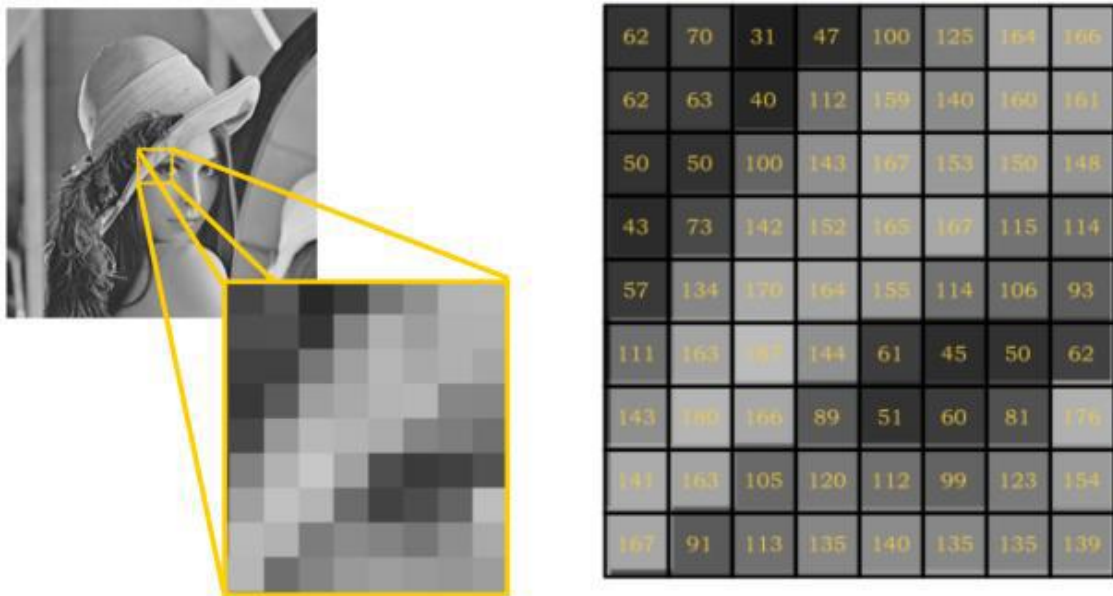


Figure 2-7: Pixel details in a grey scale image

The pixels gain a third dimension thanks to colour. In the case of an RGB image the three coordinates correspond to the intensity of red, green and blue levels at that particular pixel. A digital image can be varied and modified a lot by applying filters or some other functions. Image understanding is the basis for the computer vision field.

An image can have different colour representation and can be stored in different file formats. Those colour representations are binary, grey scale, true colour. The binary images have only two colours and they have to be either black (value: 0) or white (value: 1). Whereas the grey scale image gives only the intensity of light in the image at each pixel. The values range from 0 for no light to 255 for full light intensity. True colour images

provide the natural colour images with most popular colour spaces being RGB which stands for Red, Green and Blue, and CMYK which stands for Cyan, Magenta, Yellow and Black. These gives the intensity of the particular colour intensity.

Images can be manipulated according to the requirement for getting the information needed by applying filters. Edge filters and Smoothing/Blurring filters are the two primary categories of filters. Edge filters are used to detect edges, while Smoothing/Blurring filters are crucial to remove noise in the image.

These filters can be used to get the required information from the images. This helps in understanding the image and the main key for the computer vision is understanding the image.

## **2.4 Computer vision**

The purpose of the computer vision is to enable a computer to understand its environment from the visual information [12]. The computer vision deals with the images and videos from the sensors and process them for the information. This has been one of the corner stones in the development of artificial intelligence (AI). Computer vision helps us in interpreting the images or videos and identify or detect the objects of interest. Thus, it has many applications in the field of automotive, defence, medical and many other fields.

With the advancements in the computer vision, the application of this technique is reaching new fields. The main problem is the complexity of the visual data being processed [13]. Consider the image from Figure 2.6. There are several shrimps in the image and few of the animals even overlap. So, the complexity to differentiate the animals is even more. Since its inception, computer vision algorithms have advanced significantly, and they can now handle more complicated data.

The recent rapid development in the CV has many reasons and the most obvious among them is advancements in processing power, memory and storage capacity of the modern computers [13].

### *2.4.1 Machine learning*

Machine learning is one of the branches of the AI and it uses the data provided and learns from it and improves the accuracy with data like a human would. It is among the crucial areas of data science. It uses algorithms and statistical techniques to draw conclusions from the data. Then it analyses the data and provide the key insights from it.

Deep learning is the sub field of machine learning and can perform a diverse set of impressive tasks. It creates complex neural network models and make accurate data driven decisions [14]. The majority of contemporary technology, including computers, smartphones, and other gadgets, now uses deep learning as standard technology. This is becoming standard technology in recognition software be it a image recognition or a speech recognition. Deep learning algorithms are used in medical field as well to analyse the data and make accurate prediction.

Machine learning is, for the most part, sub- divided into three components based on the approach to learning [15]. They are supervised, unsupervised and reinforcement learning. Supervised learning uses labelled dataset to learn and make the prediction while the unsupervised learning method uses non-labelled dataset. The appropriate prediction is rewarded while the incorrect one is penalized in the machine learning process known as reinforcement learning. We use supervised learning method for this project. We use the labelled data to detect the shrimps in the tank.

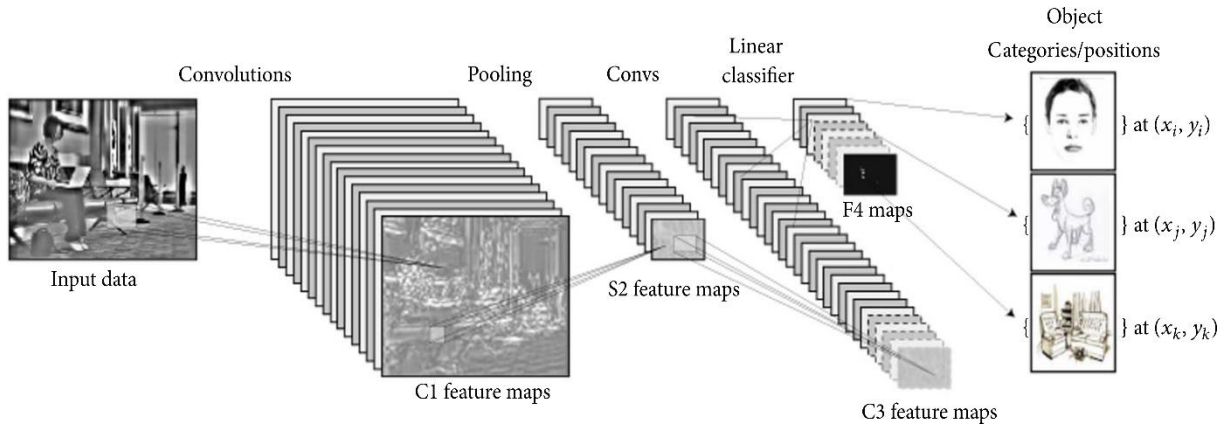
### *2.4.2 Deep learning and Convolution Neural Networks (CNN)*

Deep learning tries to mimic how a human brain perceives and understand the data. To learn and understand, it employs computational models with several processing levels. The need to imitate the human brain led to the development of neural networks. This has fuelled great strides in several computer vision problems such as object detection, motion tracking, human pose estimation and semantic segmentation [16].

The neural networks are heavily inspired by the operation of human nervous system. These are mainly comprised of several interconnected computational nodes, generally referred to as neurons, working collectively to learn from the input and provide the best output.



Convolution neural networks have three main types of neural layers, and they are convolution layer, pooling layer and fully-connected layer. The CNN takes the input image and provide it to convolution layer which in turn will determine the feature maps for the input image.

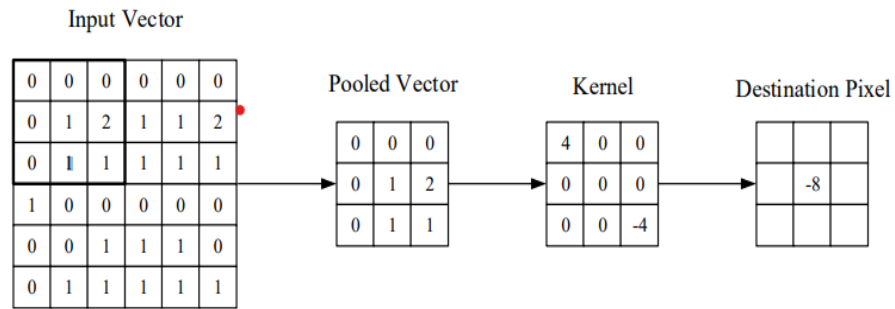


**Figure 2-9: CNN architecture example [16]**

The CNN architecture is illustrated in Figure 2.9, where the convolution layer creates the feature map for the input image before passing it to the pooling layer. Pooling layer performs the down-sampling and up-sampling thereby, reducing the number of parameters in the activation. The fully connected layer will then produce the score to be used for classification.

## Convolution layer

The convolution layer is crucial to how CNNs operate. This layer's parameters focus on the use of learnable kernels. Convolution of the input image using kernel is carried out in this layer. Convolution is a linear operation in which the input is multiplied by the set of weights called kernel. These kernels are usually small in spatial dimensionality, but spreads along the entirety of the depth of the input [17]. The kernel should always be smaller than the input vector. The visual representation of the convolution layer is as shown in the figure 2-10. The input vector is placed over the kernel's central element, and the dot product is then computed. The weighted average of that pixel and any adjacent pixels is then used in its place.



**Figure 2-10: Convolution layer [17]**

This is done on the entire input by gliding the kernel over it and calculating the scalar product. From this the network will learn the activations when they see a feature at any position of the input. The number of times the kernel slid over the image is called a stride. Depending upon the size of the input image, convolution window and the stride applied, the feature map dimension is determined.

### Pooling layer

Downsampling is the pooling layer's primary purpose in order to make the subsequent layers less complicated [18]. The activation vector output from the convolution layer has a considerable size. The size of the activation vector must be decreased in order to minimize the amount of computational resources needed to handle such data.

The input is scaled and operated on by the pooling layer over each activation map. The most common pooling method is max-pooling applied on a kernel of dimensionality 2 x 2 with the stride of 2. This reduces the size of the activation vector to 25% of the original size [17]. Some of the other pooling operations available include L1/L2-normalisation and average pooling.

Figure 2-11 shows the max-pooling operation with 2 x 2 kernel and stride 2. It leads to down-sampling with each 2 x 2 block mapped to a single block and thereby reducing the size of the activation vector.

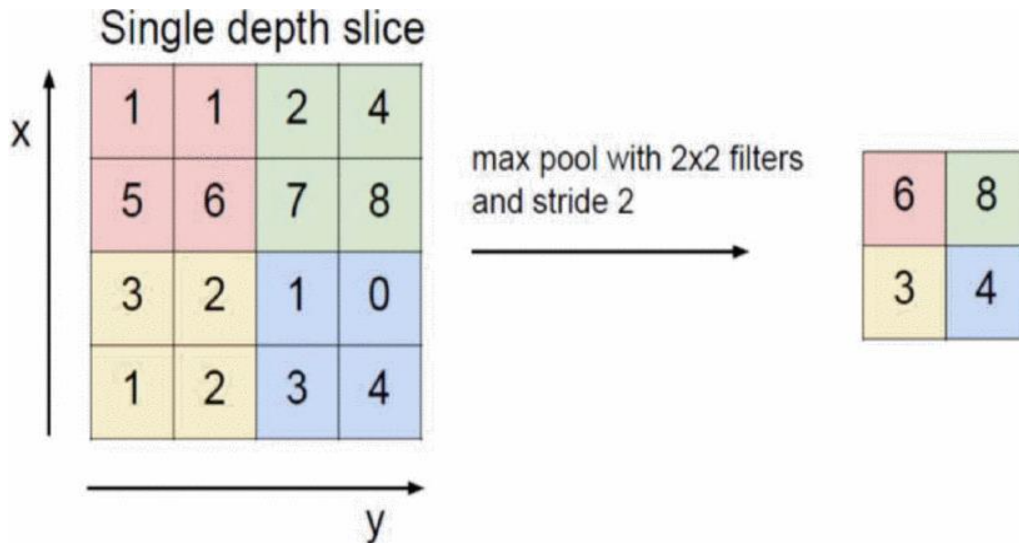


Figure 2-11: Max-pooling with 2 x 2 filter [18]

## Activation Function

An activation function is crucial to neural networks' ability to learn complex data. Depending on how relevant each neuron's input is to the model's prediction, they determine whether a neuron has to be activated or not. This works as a mathematical gate between the input to the neuron and the output going to the following layer.

We generate the sum of products of inputs and their weights in neural networks. The output of that specific layer is then produced by applying the outcome to an activation function. A neural network's performance leads to linear regression without the presence of an activation function. Two of the most used activation functions are ReLU (Rectified Linear Unit) function and Sigmoid function.

## Fully-connected layer

Every node in the fully-connected layer is directly coupled to every other node in the two adjacent layers, much like in conventional neural networks. [19]. It is sometimes referred as densely connected layer. It has all the possible connections between the two adjacent layers. This makes every input influencing every output.

### 2.4.3 Object Detection

Object detection is one of the fundamental concepts in computer vision. These models are used to understand the image and gather the valuable information. It is used in many applications such as face detection, image classification, autonomous driving and so on [20].

The object detection model's approach is classified into two methods: one-stage methods and two-stage methods. In one-stage method, the object recognition and bounding box generation is done in a single stage. While in two-stage model, it is done in two different stages. The object detection is also referred as object recognition and it is a collection of three related computer tasks.

- **Object Localization:** From the input image, locate the region of interest and put a bounding box around it to indicate the location.
- **Feature Extraction:** Predicting the class or classes in case of multiple objects from the input image
- **Classification:** Taking the image with one or more objects, detecting the objects from the region of interest and then identify the class associated with the object. Gives output as image with bounding boxes around objects along with the class label.

The one- stage models are faster while the two-stage models are more accurate. In the one-stage method, the localization and detection are done in a single stage. While the two-stage method finds the region where the objects more likely to appear called regions of interest (ROIs). These ROIs are then further processed and the output with bounding boxes are generated. Finally non-maxima suppression (NMS) is used to eliminate the duplicate or highly overlapping results [21].

You only look once (YOLO) and region-based convolution neural network (R-CNN) models are the best know deep learning models used in object detection. The architecture of the R-CNN model is as shown in the following figure 2-11. The R-CNN method generates the region of interest in the first stage and in the second stage, it focuses on the ROIs to generate the bounding boxes with the object detected.

Figure 2-12 shows the R-CNN architecture with three modules: Region proposal calculates features for each proposal, and after that, using a class specifier, it categorizes each region.

## R-CNN: *Regions with CNN features*

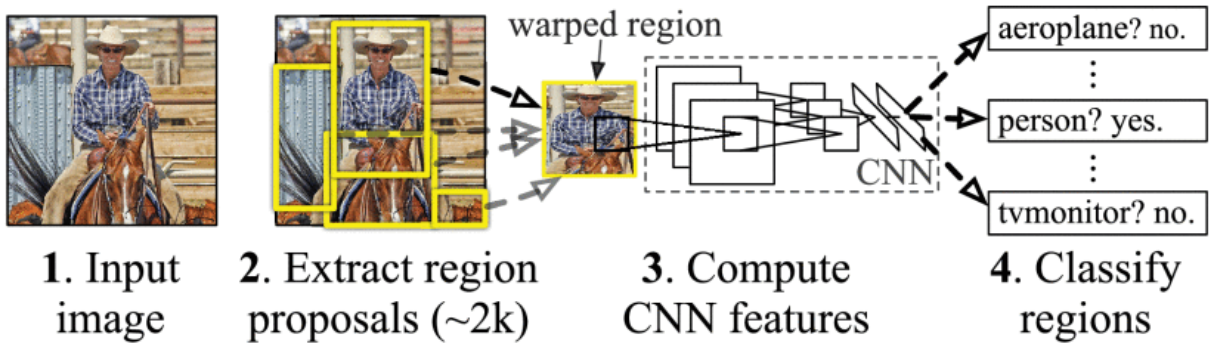


Figure 2-12: R-CNN architecture [20]

The region proposal-based framework's two-step procedure, which delivers a crude scan of the entire scenario before focusing on regions of interest (Rois), somewhat mimics the attentional function of the human brain [20]. The R-CNN is then modified and improved to Fast R-CNN and then to Faster R-CNN.

### 2.4.4 Segmentation

Segmentation is the process of differentiating the regions of an image homogeneously with respect to some characteristics such as grey tone or texture. The adjacent regions should have significantly different values [22].

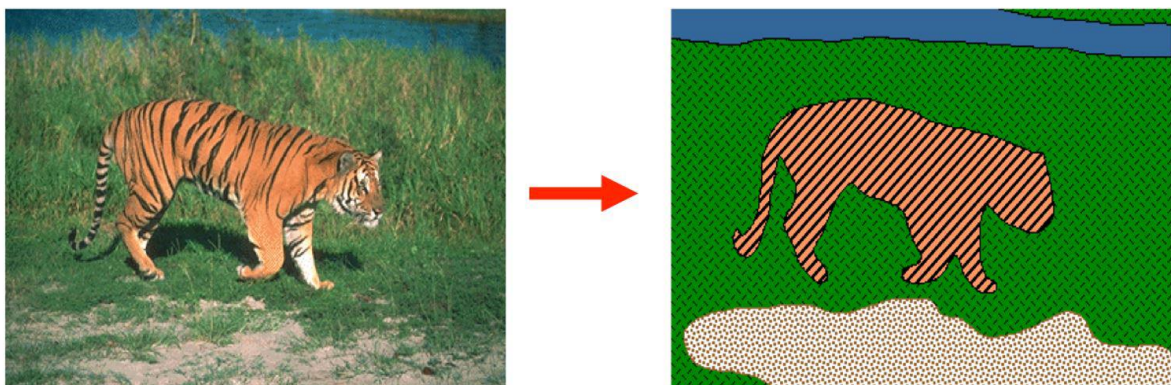


Figure 2-13; Segmentation output [23]

The accuracy of the segmentation depends mainly on how well the objects of interest separate into distinct measurement space cluster [22]. It has major applications in medical image analysis, autonomous vehicles, face recognition, surveillance etc., An example of segmentation is shown in the figure 2-13. In the figure, the tiger is very distinct from the surroundings and hence the accuracy of the segmentation is high for this image.

Depending on how much and what kind of information an image segmentation method conveys, it can be divided into three categories. They are instance, semantic and panoptic segmentations.

Semantic segmentation detects the objects and group them to the class which is labelling every pixel in the image to the associated class label. It is unable to distinguish between two items belonging to the same class. Whereas instance segmentation provides unique label for separate instances of objects belonging to the same class [24]. Panoptic segmentation is the combination of both the instance and semantic segmentation where, it gives two labels to each pixel: one is semantic label, and the other is instance label.

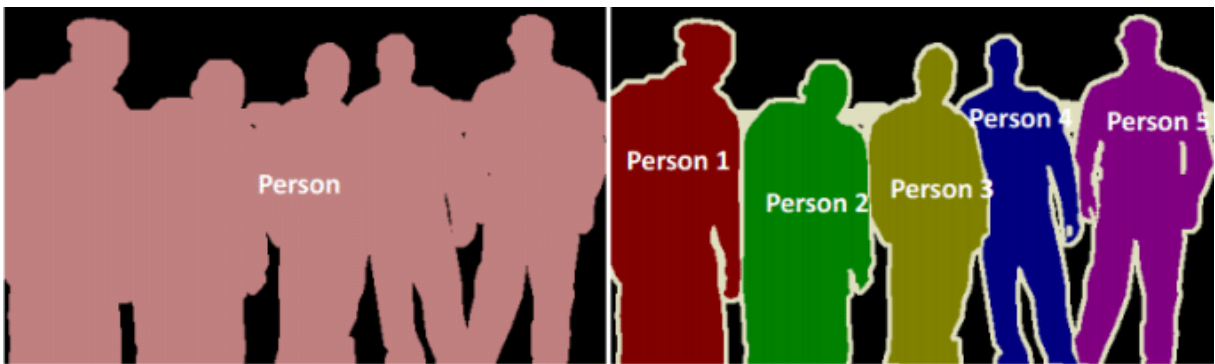


Figure 2-14: Instance vs Semantic segmentation [44]

The difference between the instance and semantic can be observed in the figure 2-14. The semantic segmentation grouped all the pixels associated with the class people a same label while instance segmentation assigned each person a different label.

Instance segmentation is used widely used in various fields such as autonomous driving, medical field, surveillance and robotics. This helps in differentiating the objects of same class, which in turn helps in identifying the boundaries of every object detected.

## **Summary of the chapter**

This chapter focuses on providing the background knowledge needed for this thesis. It gives brief knowledge ranging from shrimp biology for calculating length to image understanding basics.

The anatomy of shrimp and the method used to determine their length are covered in the chapter's opening section. It also provides the length/weight conversion factor needed to calculate weight.

The second section focuses on the farm setup and camera setup. This section discusses about the area of the farm and the details about the camera installed.

The third section deals with the basic image understanding concept. Only the basic digital image structure is described in this section. The fourth section deals with the computer vision techniques and their basics. It started with the machine learning and from there the section goes to deep learning and CNNs. Finishes with object detection and segmentation.



# 3 State of the Art

The calculating of biomass requires the estimation of both the size of the shrimps and number of shrimps in the area. To maintain the health of the animals, the shrimps must be as stress free as possible. Because of its significance, the need for the development of techniques is gaining importance. The use computer vision models in the determination of biomass in aquatic animals is increasing every day. Computer vision techniques such as object detection algorithms have come a long way. In this section, some of the state-of-the-art models on the object detection are discussed.

## 3.1 Polygon mask detection (Mask R-CNN)

The R-CNN architectures are updated and improved to faster and better architectures in the form of R-CNN, Fast R-CNN and Faster R-CNN. The model, known as Mask R-CNN, expands the Faster R-CNN by adding a branch for object mask prediction in addition to the branch already present for bounding box identification [25]. The framework of the model is as shown in the following figure 3-1.

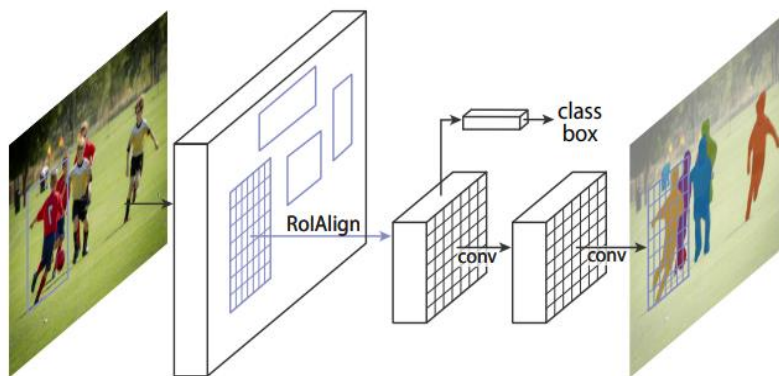


Figure 3-1: Mask R-CNN framework for instance segmentation [25]

The R-CNN models generally have two outputs, a class label and a bounding box. To the already-existing branches that produce output in the Mask R-CNN model, an additional output is introduced. Mask R-CNN model adds third branch that provides mask to the detected object. The Mask R-CNN is basically divided into two parts: region proposal network (RPN) which proposes the bounding box for each object and binary mask classifier, which generates the mask for each object.

The RPN proposes a ROI from the feature map generated when the image runs through the CNN. And in the second stage the, in parallel, this model also generates the binary mask for each ROI.



The results of the model compared to the other CNN models when trained on COCO dataset is as shown in the following figure 3-2.

	backbone	AP <sup>bb</sup>	AP <sup>bb</sup> <sub>50</sub>	AP <sup>bb</sup> <sub>75</sub>	AP <sup>bb</sup> <sub>S</sub>	AP <sup>bb</sup> <sub>M</sub>	AP <sup>bb</sup> <sub>L</sub>
Faster R-CNN+++ [15]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [22]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [32]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [31]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	<b>52.1</b>
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
<b>Mask R-CNN</b>	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
<b>Mask R-CNN</b>	ResNeXt-101-FPN	<b>39.8</b>	<b>62.3</b>	<b>43.4</b>	<b>22.1</b>	<b>43.2</b>	51.2

Figure 3-2: Object detection results [25]

The Mask R-CNN model has a ROIAlign layer that plays a key role in mask prediction. ROI Pool, which is a standard operation for small feature map extraction, has a negative effect on predicting pixel-accurate masks. In order to minimize this, ROIAlign is used. It removes the harsh quantization of RoIPool, properly aligning the extracted features with the input [25]. The mask output of the model in a test image is as shown in the following figure 3-3.



Figure 3-3: Mask prediction [25]

This model has large number of applications. One of them is human pose estimation, which is achieved by key-point detection. The results of the model with key-point detection is as shown in the figure 3-4. The model achieved the average precision (AP) of 63.1 while simultaneously predicting boxes along with the segments and key-point. With the effectiveness of the model in bounding box, segment and key-point extraction, Mask R-CNN is an effective framework for instance segmentation.

	AP <sup>kp</sup>	AP <sup>kp</sup> <sub>50</sub>	AP <sup>kp</sup> <sub>75</sub>	AP <sup>kp</sup> <sub>M</sub>	AP <sup>kp</sup> <sub>L</sub>
CMU-Pose+++ [4]	61.8	84.9	67.5	57.1	68.2
G-RMI [26]†	62.4	84.0	68.5	<b>59.1</b>	68.1
<b>Mask R-CNN, keypoint-only</b>	62.7	87.0	68.4	57.4	71.1
<b>Mask R-CNN, keypoint &amp; mask</b>	<b>63.1</b>	<b>87.3</b>	<b>68.7</b>	57.8	<b>71.4</b>

Figure 3-4: Keypoint detection results [25]

## 3.2 Key-point detection

### 3.2.1 U-Net

U-Net is a convolution network architecture used for fast and precise segmentation of images. It has a U-shaped architecture design and hence the name U-Net. It consists of encoder and decoder blocks as shown in the figure 3-5.

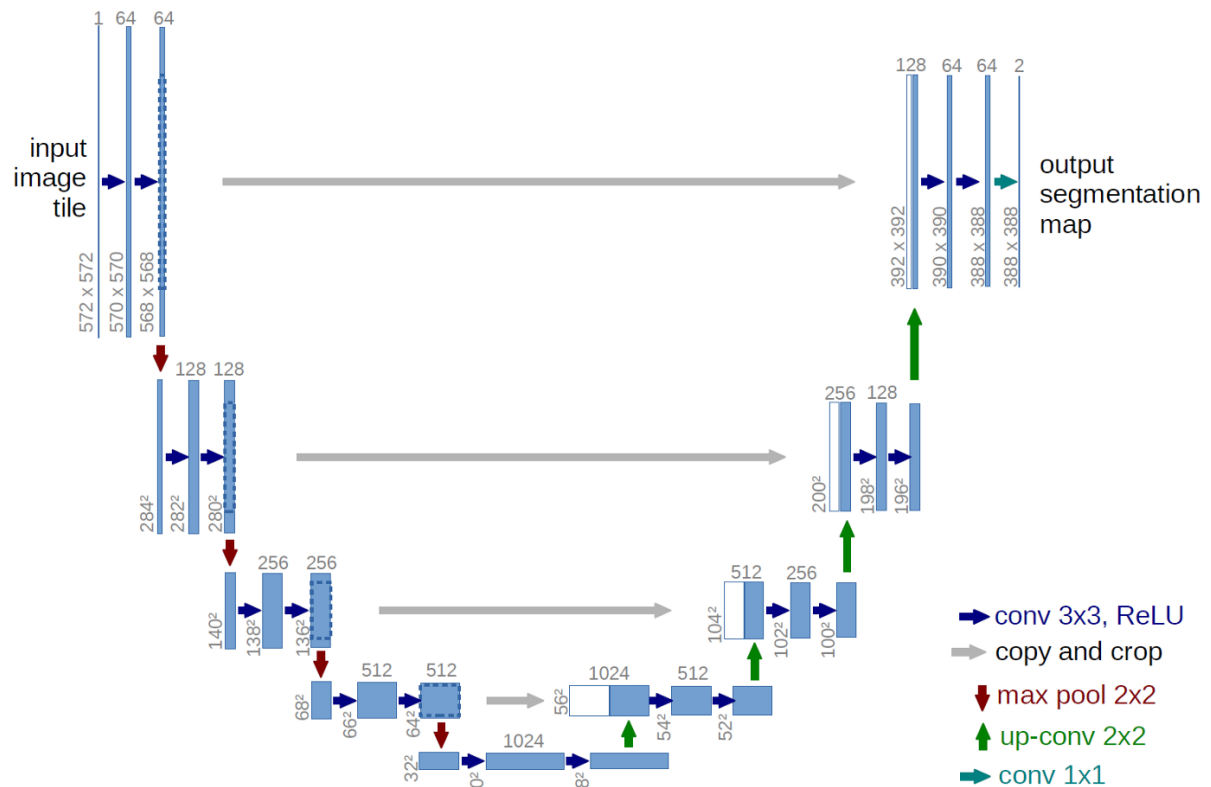


Figure 3-5: U-Net Architecture [43]

It has a path that contracts and one that expands. [26]. The contracting path consists of four encoder blocks and the expansive block has four decoder blocks. There is a bridge connecting the two paths. Every encoder block has twice the number of filters (feature channels) and half the special dimensions as the next encoder block in the network. On the flip side, every decoder block has half the number of filters and double the special dimensions.

### Contracting path

The contracting path is the left-side part in the architecture from figure 3-5. It follows the typical convolution network with repeated application of two 3x3 convolutions

(unpadded convolutions) followed by rectified linear unit (ReLU) and max pooling. The two 3x3 convolutions are used to create each block and a ReLU is applied to each convolution. This is followed by a 2x2 max-pooling with stride 2 for down-sampling. The main role of this block is the feature extraction. Number of feature channels are doubled after each down-sampling step.

### **Expansive path**

The expansive path in the architecture is the right-side part of the figure 3-5. It consists of an up-sampling of the feature map followed by a 2x2 convolution (“up-convolution”). It halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path. The next step consists of two 3x3 convolutions, each followed by a ReLU activation function. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution with sigmoid activation is applied after the final decoder’s output. The segmentation mask for the classification at the pixel level is provided by the sigmoid activation function.

### **Skip connections and bridge**

The skip connections serve as a quick link that provides the uninterrupted flow of gradients to the prior levels. Thereby improving the gradient flow during backpropagation, which helps network acquire better indication. With the help of these, the decoder produces better semantic features. The bridge provides the information flow between the networks of encoders and decoders.

#### *3.2.2 DeepLabCut (DLC)*

DeepLabCut is an open-source python software for the pose estimation of the animals. It uses transfer learning and deep neural networks to get better results with less training data. This has been successfully implemented on rats, various aquatic animals . humans and bacteria. It is a reliable and efficient tool for high-quality video analysis, which requires feature detectors of user-defined body parts to be learned for a specific situation [27].

DLC model is very adaptable and supports the usage of various packages. The workflow of the DLC model is as shown in the figure 3-8.

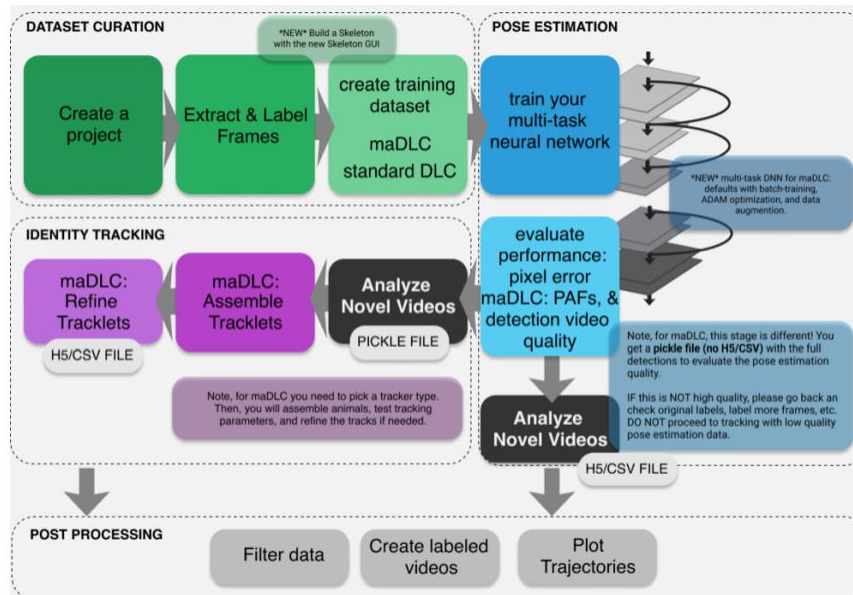


Figure 3-6: DeepLabCut workflow [28]

The DLC model has its own GUI and it is a reliable tool for high-quality video analysis. From the figure 3-6, the DLC workflow can be observed, and it flows in a sequential order from creating a project to extracting and labelling the images. The dataset is created from the labelled images and then this dataset is used to train the model for pose estimation.

### 3.3 Bounding box detection

YOLO stands for you only look once and it is a single step process. It takes input image and provides output with bounding box in a single step unlike R-CNN architecture-based models, which are two steps: classification and detection. It reframes object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using this system, you only look once (YOLO) at an image to determine the classes of the objects present and where they are located [29].

The YOLO model combines the object detection and object classification in a single convolution network [30]. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection [29]. The latest of the YOLO model is the YOLOv7 which was released in 2022.

The performance of the YOLOv7 model in comparison with other real-time object detectors can be seen in the figure 3-7.

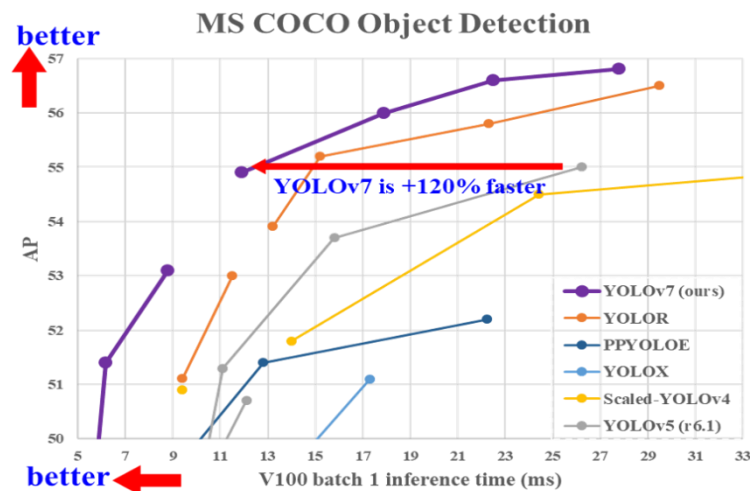


Figure 3-7: YOLOv7 performance [32]

The YOLOv7 model outperformed the others in both the accuracy and speed. “Bag of freebies” are the methods that only changes the training strategy or only increase the training cost [31]. The most often adopted method which can be considered as a bag of freebies is the data augmentation method. YOLOv7 makes use of trainable bag of freebies with a more robust loss function, a more efficient training and label assignment method. All these properties make the YOLOv7, a state of the art real time object detector.

### Summary of the chapter

This chapter discusses the state of the art related to the thesis work. Research areas that are relevant to the object detection were discussed in this chapter. The first section deals with the detection models that work with polygon type annotations. In order to measure the accurate length of the shrimp, polygon type annotation is the better one since the shrimps can be in any direction.

The second section deals with the key-point detection models. This section covers the U-Net and DeepLabCut. The third section focuses on the bounding box detection models especially YOLO model.

## 4 Concept.

The calculation of biomass requires the calculation of average weight of the shrimps detected and determining the number of shrimps in the image. After determining these two values, biomass is calculated by multiplying both. The flow chart of the project is as shown in the figure 4-1. The model we are proposing is the using of three different object detection models: YOLOv7, U-Net and Mask R-CNN to get the best results.

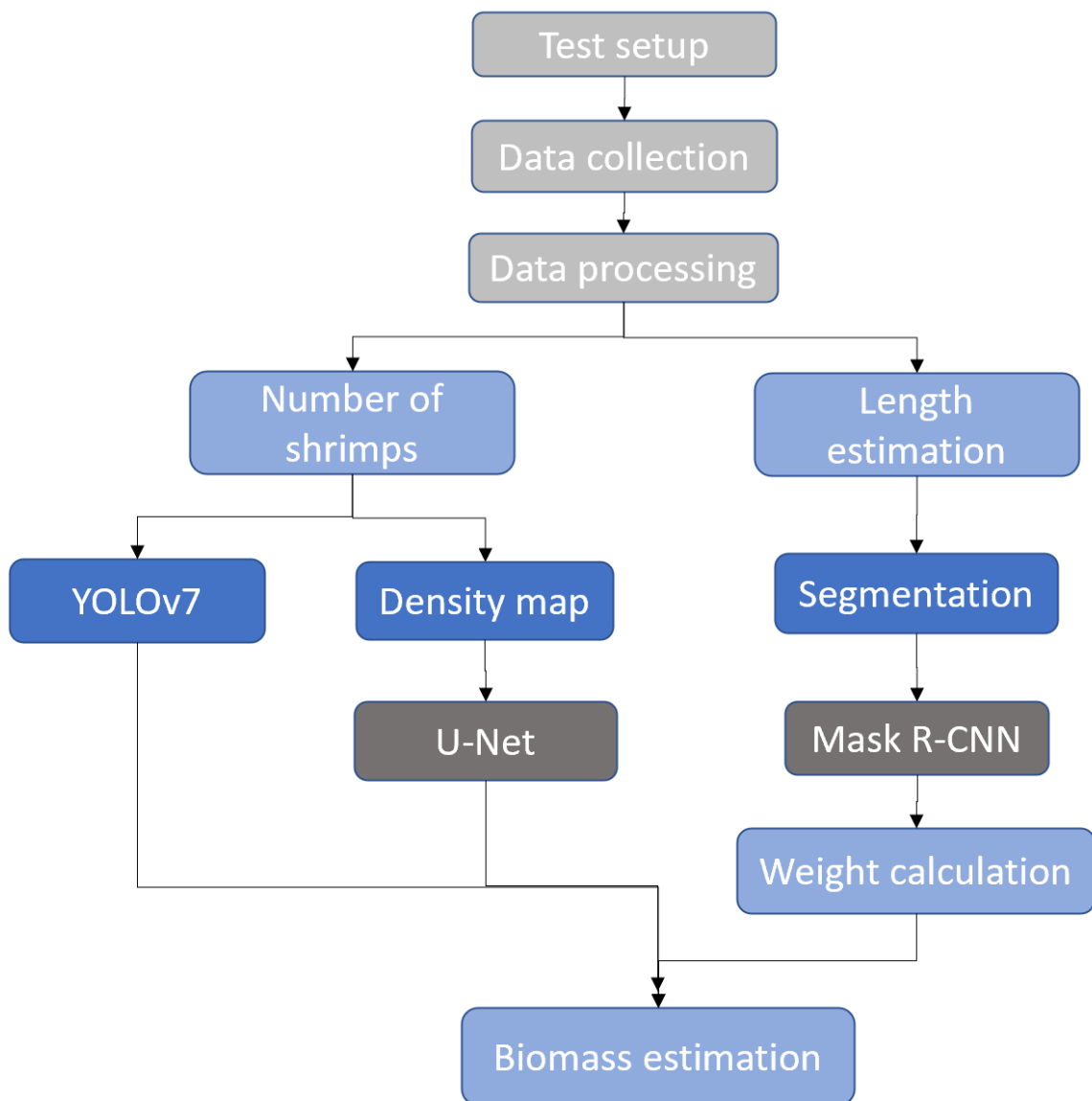


Figure 4-1: Flowchart of proposed concept

## 4.1 Data processing

After deciding the best conditions for the camera settings, and the environment around the tank such as lighting, the data is collected. The data is collected in the form of videos which are then divided into frames. These frames are then annotated as per the requirements. Since we use three different algorithms in this project, the type of annotations varies depending on the model.

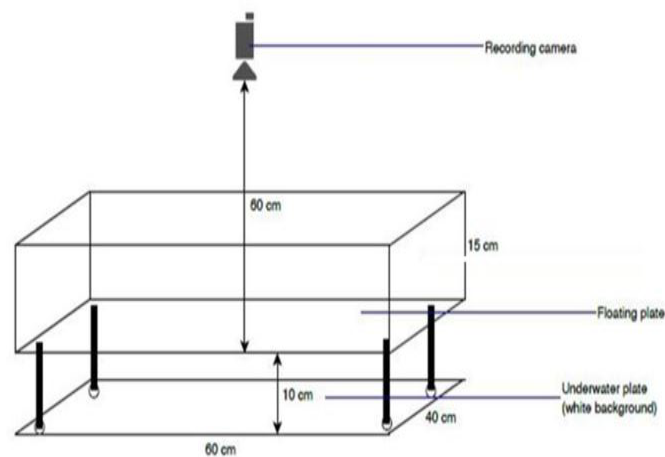


Figure 4-2: Proposed camera setup

Figure 4-2 shows the camera setup at the farm.

## 4.2 Number of shrimps

In order to calculate the number of shrimps in the given image, we use two algorithms: YOLOv7 and density map models. We use both the models and at the end, the accuracy of the models is compared. The results from both the models are evaluated and the model with better accuracy is selected for calculating the biomass.

### Density Map

The Density map model uses U-Net algorithm for segmentation. This model uses point annotations and generate a density map for the image. The density map consists of a map with dots in the place of shrimps detected. The basic idea here is to generate a density map and then count the objects from it. The first step is to create a corresponding density map for every image. The images are annotated at the stomach parts with a point type annotation. Then a density map for each image is generated by applying a convolution with Gaussian kernel. The training dataset consist of images with the density maps of those.

Then the model is trained with the generated dataset to map an image to a density map. The number of shrimps can be calculated from that map.

## **YOLOv7**

The YOLOv7 model is the latest iteration of the YOLO based models. This makes use of bounding box annotations. The backbone of the YOLOv7 computational block is named E-ELAN, which stands for Extended Efficient Layer Aggregation Network. It makes the model better by making it learn by using “expand, shuffle, merge cardinality” [32].

Model scaling is mostly used to modify certain model properties and produce models at various scales to accommodate various inference speeds. Different scaling considerations in conventional techniques with concatenation-based architectures (like ResNet or PlainNet) must be taken into account collectively rather than separately. For instance, increasing model depth will affect the ratio between a transition layer's input and output channels, which may result in less hardware being used by the model. For a concatenation-based model, YOLOv7 introduces compound model scaling. The model's original design qualities can be preserved by using the compound scaling method, keeping the ideal structure intact.

This makes the model faster and more accurate. The images are annotated with rectangular shaped bounding boxes around the shrimps and then the model is trained using the generated dataset.

### **4.3 Length detection**

For calculating the average weight of the shrimps in the given image, the average length of the shrimps is determined. Using length/weight conversion factor, the average weight of the shrimps is calculated.

The output from the density map cannot be used to calculate the length since it uses point type annotations and the shrimps detected are represented by a single dot. Whereas with YOLOv7 model, the shrimps have bounding box around them. Since the shrimps are in different angles and in various positions, the calculation of length with this model is also not possible. That is why, the instance segmentation using Mask R-CNN model is the better alternative as it gives masks as output.



This model makes use of polygon type annotations with each end at the extreme points of the shrimp. Segmentation produces output with three results: masks, bounding boxes and class name. These masks can be used to calculate the length of the shrimps. The instance segmentation gives all the details of the detected shrimps in terms of pixel values. In order to determine the length, we use contours of the masks generated. These contours can be used to get the extreme points of the detected shrimp. The length of the shrimp is calculated by measuring the distance between the extreme points. The length of the shrimp will be the maximum value among the calculated distance between extreme points.

The length of each individual shrimps detected are calculated. The partial detected shrimps are eliminated from these. The average of these values is calculated and by using the average length and length/weight conversion factor, average weight is determined.

#### **4.4 Biomass determination**

With average weight determined using the instance segmentation using Mask R-CNN model and the number of shrimps estimated by YOLOv7 model and density map model, the biomass is calculated by multiplying both. This gives the Biomass in the image in grams.

After training both the models with the dataset generated, the trained weights are then downloaded and used to implement the entire project together. The model is then implemented in Google COLAB which offers GPU. The result is then displayed on the input image. This image is used to determine the amount of food to be dispensed for the shrimps in order to maintain healthy life for the animals and better production for the farmers.

For checking the validity of the data, the images are taken at one set time every day for both manual measurement and the model measurement. The data is collected for several days and compared both the outputs.

## **Summary of the chapter**

This chapter provides the general idea of the proposed concept for the project. The flowchart of the concept is provided in this chapter and this chapter mainly consists of four sections. First section is about data processing. It gives general idea about how the camera is setup at the farm and the generation of dataset needed for training the model.

The second section explains the number of shrimps estimation. It provides the basics of both density map model and YOLOv7 model. It also discusses about how the individual models work and how the output should look like.

Third section is about estimating the average weight. This method uses instance segmentation with Mask R-CNN and this section discusses about why this model is preferred over the other two models.

The fourth section explains the calculation of the biomass which is by using the second and third section. This also discusses about the combining of the models to get the output. The process of verifying the output is also explained in this section.

# 5 Implementation and Methodology

The implementation of this project is done in two parts: weight estimation and number of shrimps estimation. The data is labelled as per the model requirements and the data is stored in Google Drive to make deploying the model easy. Google COLAB is used because of GPU availability.

## Data processing

This chapter deals with the processing of data required for executing the project successfully. Tasks such as data collection and preparation are discussed in this chapter.

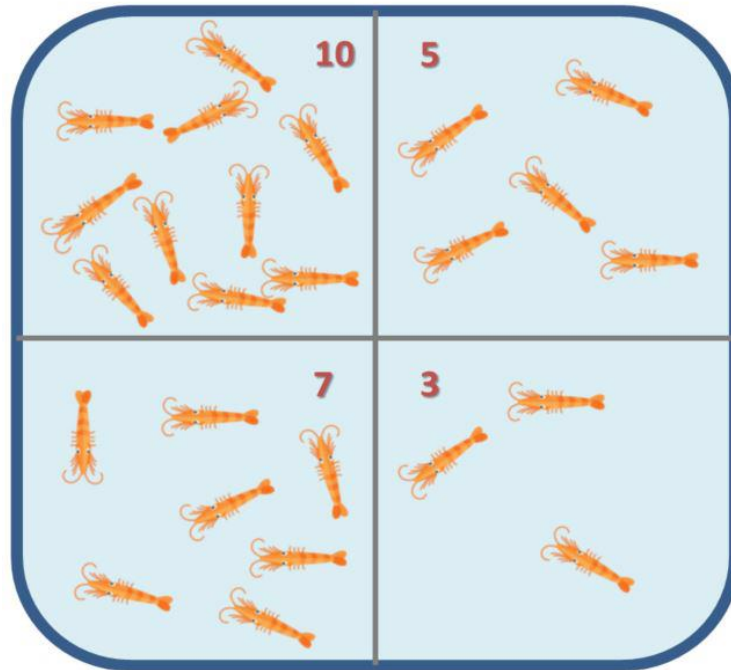
### 5.1 Data collection

Data collection deals with the images to be collected for training the model. For better results, the images of wide variety of settings need to be collected. Several factors such as camera settings, time, light availability in the farm and so on, all can be varied in order to make the data as effective as possible. Camera settings such as white balance, saturation and contrast are varied in this process.

To determine the best conditions from the available, a test model is setup and the model is tested with various factors. Based on the results from the test model, the best camera and light settings which have the best visibility are determined. These settings are then used to collect the data from the actual shrimp farm.

The data is collected in the form of videos from the camera adjusted to the settings determined to be the best by this test model. These videos are then converted into frames for processing the data and create dataset required for each model based on their requirements.

### 5.1.1 Test model



**Figure 5-1:Proposed test model**

A test setup was used to determine the best parameters for the camera. This setup was used only to perform experiments that implement all the varying parameters with both light and camera settings. After determining the best parameters, the actual data is collected from the farm.

Figure 5-1 shows the test model proposed for the experiment. The setup is composed of 4 sections with varying density of animals in each section. This setup is used as it is easier to analyze and determine the best light and camera parameters. Based on this proposed setup, a tub is used in this experiment which is divided into 4 sections as per the proposed model. Figure 5-2 shows the test setup using a tub for the experiment.

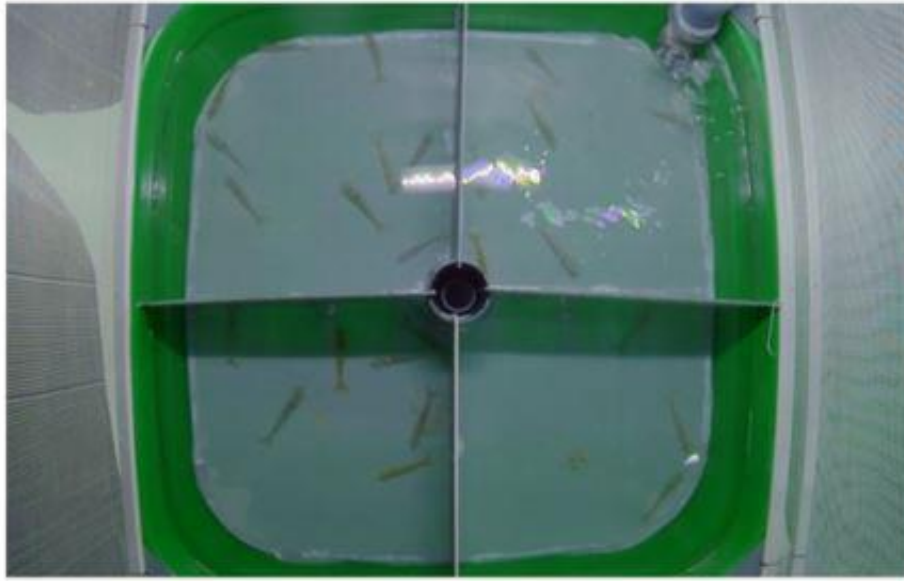


Figure 5-2: Actual test setup

### Light settings

In this experiment, RGBW LED lights are used as artificial light sources. These have four different channels with varying brightness and color. Those are the main factors that impact the visibility of the shrimps. The setup was kept under surveillance with varying light conditions. Channel 1 and channel 4 are nearly indistinguishable and hence the channel 1 is not shown in the following figure 5-3.

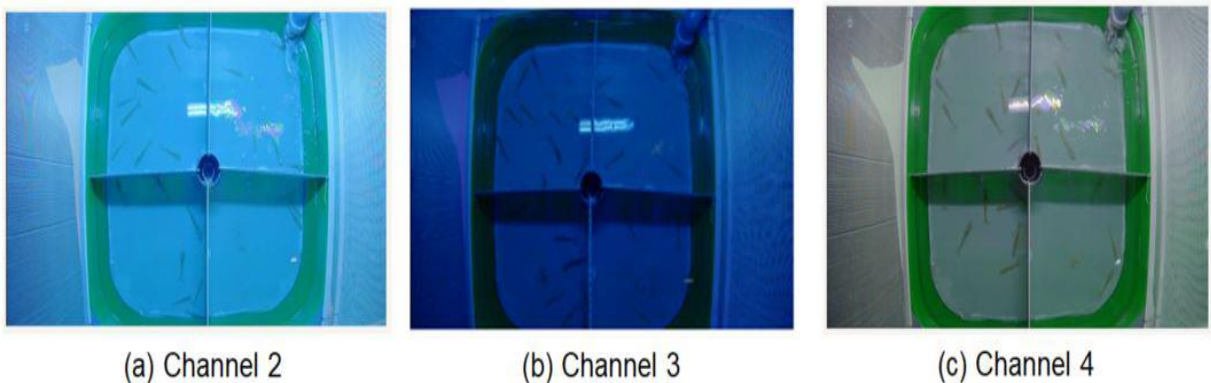


Figure 5-3: Sample from different channels

Here, the setup is kept under different channels for each day and observed. From the figure 5.3, we can see that while the shrimps are visible in channel 2, it gets slightly challenging in annotating them. The visibility of shrimps in channel 3 is quite poor and the annotation of the images will be very hard. Whereas, in channel 4 has the best visibility among the three channels and the ease of annotating is also very much

improved. After the experiment, channel 4 seems to be the best option in the three available channels based on the visibility and hence channel 4 is decided to be used in this project.

### Camera settings

The setup is further used to determine the best camera settings such as intensity and exposure. The tub is kept under camera for 3 days with varying intensity and exposure for every one hour. The intensity with 75 and 100 are observed while the exposure is observed with values of 20, 30, 80, 100 and 130.

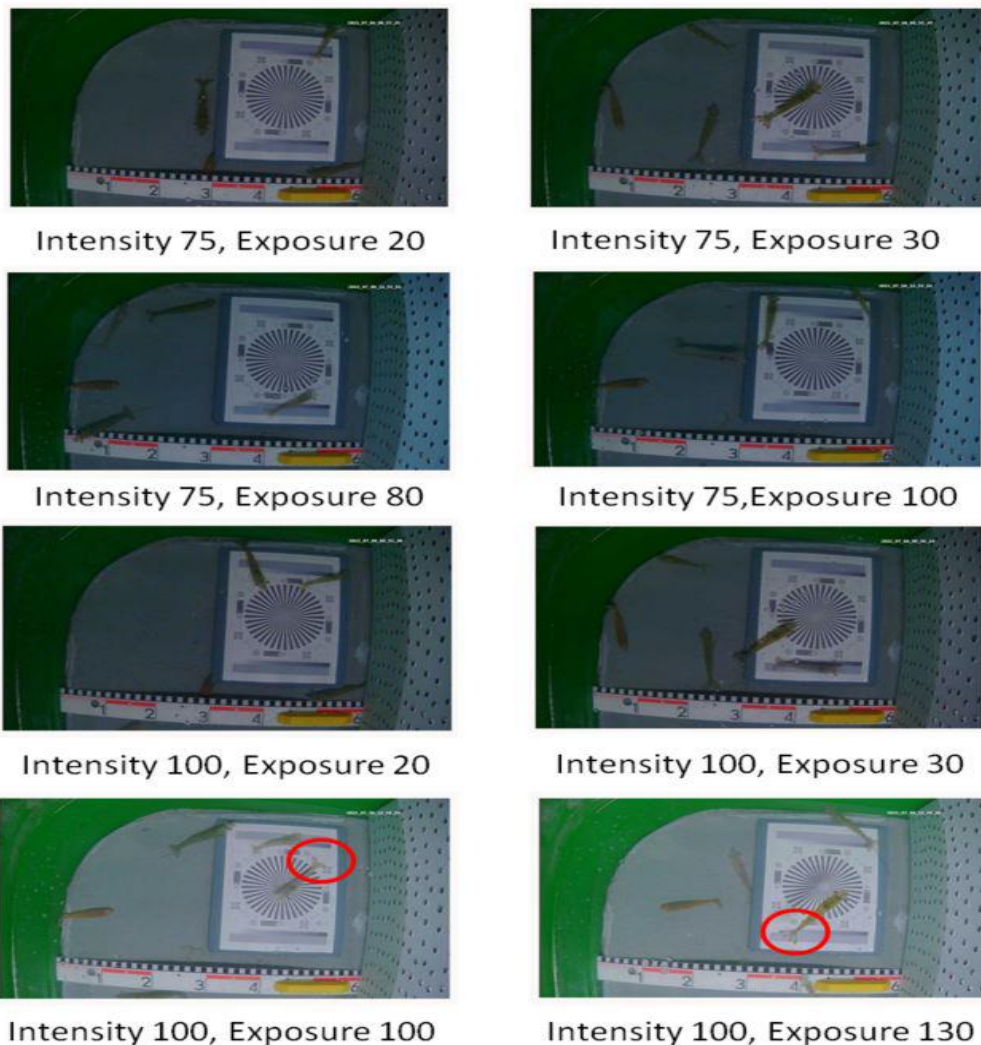


Figure 5-4: Setup with varied intensity and exposure

Figure 5-4 shows the sample images of setup with varying intensity and exposure values. While the animals are visible in all the settings, the color of the shrimps are clearly visible in only few. As the shrimp color is one of the early indicators of the stress, the color visibility is very important for detecting the stressed shrimps. So, the intensity level of 100 with exposure of 100 and 130 are preferred as the camera settings.

### 5.1.2 At the farm

With the finalized camera settings, the data from the farm is collected. To increase the variety in the data, the images can be taken with the added artificial light in the night and the daytime. The time of the day to collect the data also helps in this regard. Shrimps tend to move a lot during the feeding time and they tend to be stagnant during the sleeping time. The chances of overlapping shrimps is high during the feeding time as the animals are moving significantly more. The overlapping of shrimps is a big issue when calculating the biomass. The model might not be able to detect the shrimps which are under another shrimp.

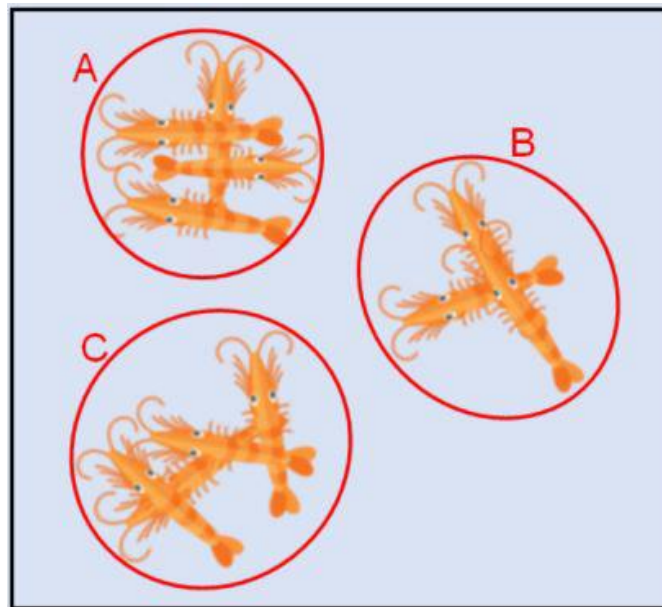
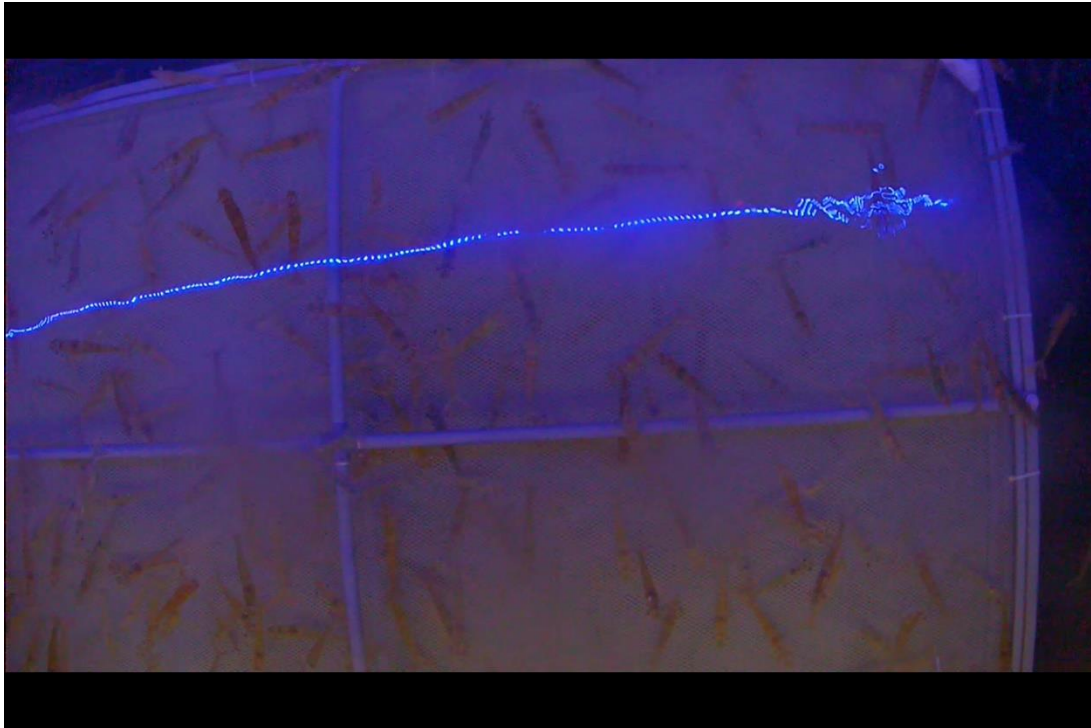


Figure 5-5: : Overlapping of shrimps [27]

Figure 5-5 shows the image with overlapping shrimps. Since the shrimp farms are heavily populated, the accuracy of the shrimps detected may not be very high. One of the reasons for that is overlapping. To improve the accuracy of the model, training the model with and without overlapping is important. Data can therefore be gathered at various times.





**Figure 5-7: Image during feeding time**



**Figure 5-6: Image during resting time**

The figures show the difference between the two images. Figure 5.7 is taken during the feeding time and hence with the heavy movement, the chances of shrimps overlapping is high. Figure 5.6 is taken during the resting time and the chances of overlapping is low.



## 5.2 Data preparing

Once the data required is collected in the form of videos, the images are extracted from the videos in the form of frames. The images collected are then augmented to increase the dataset. Augmentation methods such as scaling, cropping, rotation and so on can be used to increase the size of dataset and thereby improve the detection result of the model.

The data collected is then labelled according to the requirements. Since this project uses three different model, three different methods of annotations are used. For calculating the number of shrimps in the frame, we can use either UNet or YOLOv7. UNet uses dot annotations while YOLOv7 uses bounding boxes. And for determining the length of shrimps, mark R-CNN uses polygon annotations.

The annotated images are then augmented as per the requirement and then split into training dataset and validation dataset. Different models use different format of the labelled datasets. Mask R-CNN uses COCO dataset whereas YOLOv7 uses rectangular bounding box type annotations. YOLO makes use of the labelled dataset in YOLO format i.e. a text file with the coordinates of the bounding box annotations. U-Net uses point format annotations and after augmenting, the dataset is split into training and validation datasets. U-Net model uses the datasets in HDF5 format and hence the datasets are converted.

## 5.3 Weight estimation

Weight can be estimated by using length/weight function after determining the length of the shrimps. The length of the shrimps is estimated by using instance segmentation with Mask R-CNN model. The stepwise algorithm for this process is as follows:

Step1: Labelling the data with polygon type annotations.

Step2: Train the model in the Google COLAB with the labelled dataset.

Step3: Test the model with an image.

Step4: Get the masks of the detected shrimps and divide the image with masks of all the animals into individual masks.

Step5: Get the extreme coordinates of the mask and calculate the distance between them. The maximum value among them gives the length.

Step6: Calculate individual length of all the shrimps detected and then compute the average length.

Step7: Compute the average weight using the conversion factor.

## Environmental setup

Since the training is conducted on Google COLAB, the data is labelled and uploaded to google drive for quick and easy access. The list of packages and libraries need to be installed for the model are listed in the table 5-1. For the estimation of the weight, instance segmentation with mask R-CNN is used.

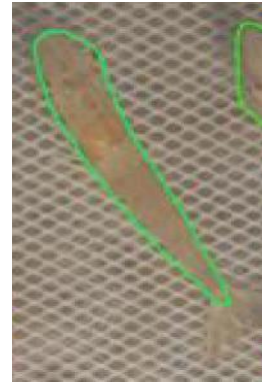
<b>Packages/libraries</b>	<b>Version</b>	<b>Description</b>
Python	3.7.13	Programming language
Pip	21.1.3	Python package installer
TensorFlow	2.5.0	Open-source library for machine learning and artificial intelligence
NumPy	1.19.5	Python library for manipulating multi-dimensional arrays, matrices, and functions
Keras	2.5.0	Open-source library that provides python interface for artificial neural networks
Matplotlib	3.2.2	Python module for visualizing the data.

Table 5-1: Environmental setup for Mask R-CNN

## Labelling the data

For the segmentation process polygon annotations method is used. For labelling the images CVAT, which is a free and open-source interactive video and image annotation tool for computer vision algorithms, is used. It supports various annotation formats including COCO, PASCAL, YOLO, VAL, KITTI and several others. We use COCO format for this project.

The CVAT also supports multiple types of annotations such as line, polygon, point, rectangle and so on. We use polygon type annotation in this project. The annotation of the shrimp is done as shown in the figure 5-9. The annotation starts at the head position and a polygon is drawn around the shrimps till the tail. After finishing the annotations of the images, the dataset is exported in any desired format, which is COCO format for this model.



**Figure 5-8: Annotation**

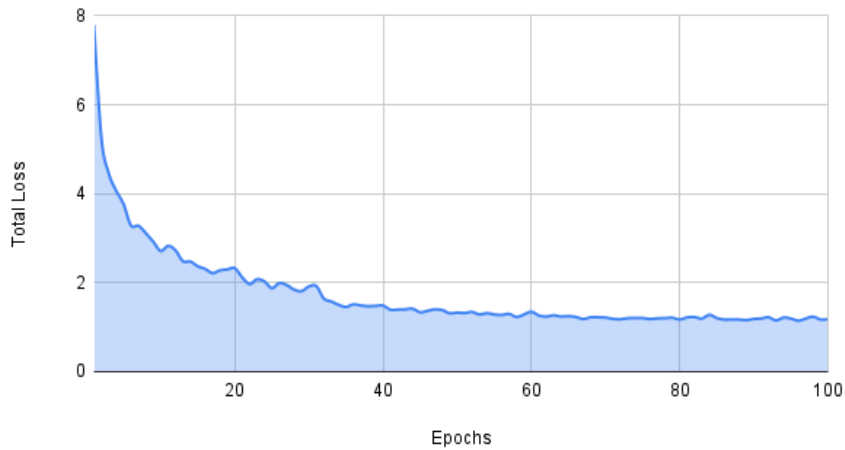
After the annotation of data, the images along with the annotations are augmented to increase size of the training dataset. We annotated 109 images and applied augmentation tools: scaling, blurring, rotate. This increased the number of images to 327.

## **Training**

After preparing the dataset, it is uploaded to google drive for training the model. The model is trained on Google COLAB and the environment setup is done by installing all the dependencies and libraries as mentioned in the table 5-1. Mask R-CNN git repository is then cloned to the environment and the initial pre-trained weights are downloaded.

- Annotated dataset = 109 images
- Augmented dataset = 327 images
- Train dataset = 291 images
- Validation dataset = 36 images
- Batch size = 8
- Iterations = 30
- Epochs = 100

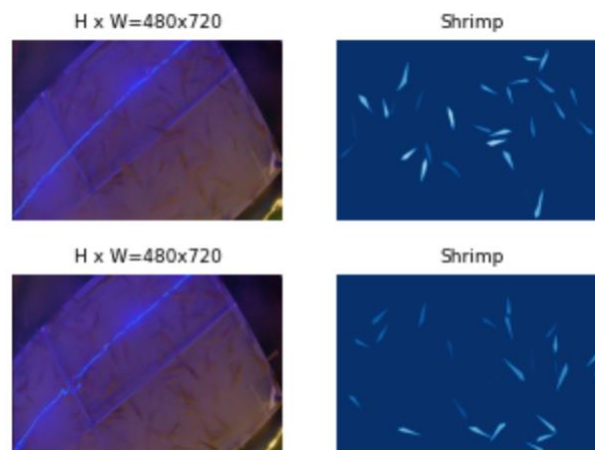
The model is trained with above mentioned parameters and the trained weight is then downloaded. This trained weight is used to test the model. The training loss of the model is as shown in the following figure.



**Figure 5-9: Training loss of Mask R-CNN model**

### Running the model

After training the model, we can cross validate the performance by displaying the trained weights on the validation dataset. Figure 5-10 shows the visualization results on random validation images.



**Figure 5-10: Visualisation on validation images**

The model is then run on a sample image to get the masks with trained weights. Figure 5-11, shows the output of the instance segmentation on an image.

The figure 5-11 shows the output of the model and as we can see from it, every detection has three results: masks, bounding box and the score. For the calculation of length of the shrimps, it is important to further process them.

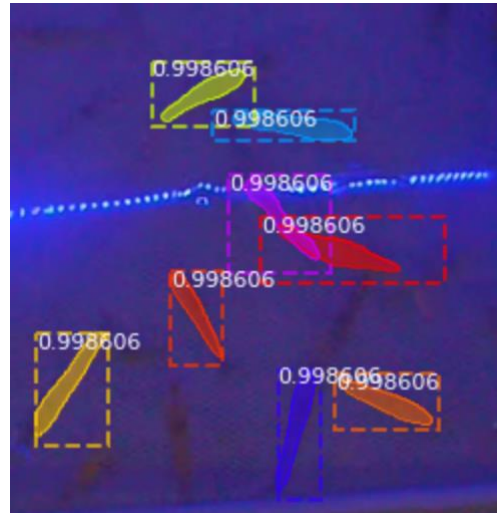


Figure 5-11: Sample image output

The output of the instance segmentation provides all the required information of the shrimps detected and it is easy to calculate the length in terms of pixels. In order to get the length in real terms, a scaling factor is needed. The scaling factor is calculated with the help of the mesh underneath the shrimps.

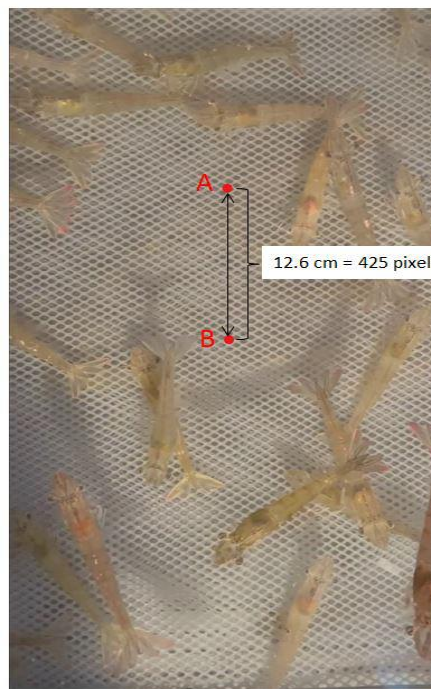


Figure 5-12: Grid measurement [27]

The scaling factor is calculated with the help of grid measurements. Figure 5-12 shows the grid measurements in the farm and since we know the distance between 20 grid holes is 12.6 cm, scaling factor is calculated by finding the pixel length between the same grid holes. The pixel length between the two coordinates is calculated and the distance is 425 pixels. And the scaling factor for each pixel to cm is 0.02965.

### Length calculation

For calculating the length of the shrimps detected, we need to consider only the contour of the mask. First, the output image with masks for each detected shrimp is converted to image with contours. Figure 5-13 shows the contour of the detected shrimps from the figure 5-11.



Figure 5-13: Contour of the masks



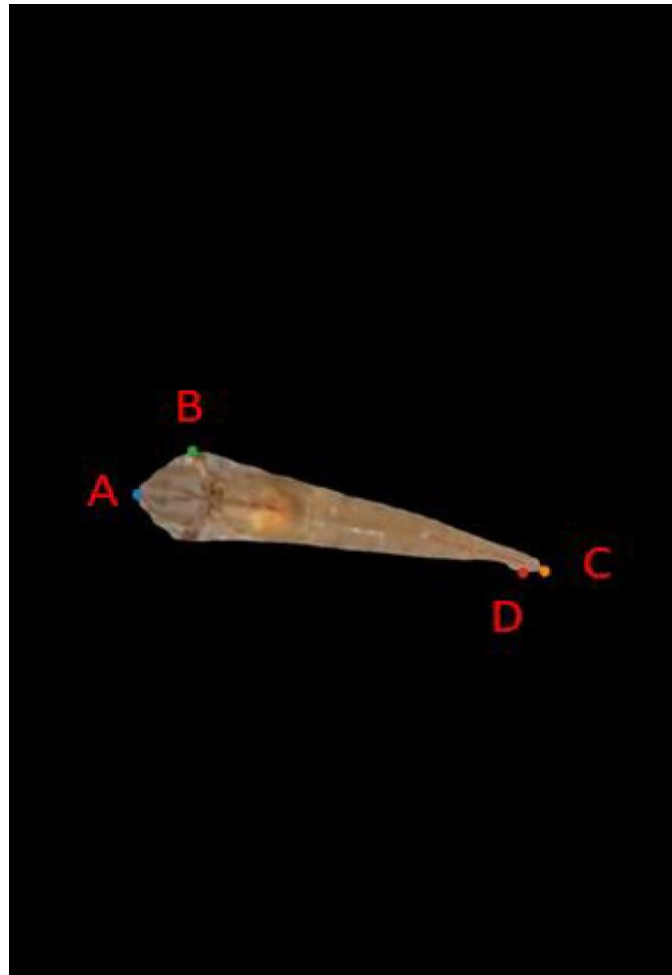
Figure 5-14: Contour of single shrimp

The contour of each shrimp is then extracted to calculate the individual length of the shrimps. The coordinates of the extreme points in the contour is then determined. The distance between all the four coordinates is calculated. Distance between two points can be calculated by using the formula.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5-1)$$

where  $(x_1, y_1)$  and  $(x_2, y_2)$  are the coordinates. For determining the accurate length of the shrimps, only fully visible shrimps are considered. The four extreme points of the fully visible shrimps detected can be seen in the figure 6-8. The extreme points detected here are top-most (B), bottom-most (D), right-most (C) and left-most points (A). The distance between all the points is then computed. There will be six

combination of computed values with distance between every two points i.e. between (A,B), (A;C), (A,D), (B,C), (B,D) and (C,D). The maximum value from the computed values will be between the head and tail point of the shrimps. That is the distance between the point A and point C in the following figure.



**Figure 5-15: Extreme points in the contour**

After extracting the extreme points and calculating the distance between them, the maximum value among the distances between each for the shrimp in the above figure is 590.22 pixels. By using the scaling factor, we calculated from the grid, which is 1 pixel is equal to 0.02965cm, the actual length of the shrimp can be calculated. For the shrimp with the length of 590.22 pixels the length in real time is 17.5cm.

The length of all the shrimps detected in the image is calculated and the average of these values is the average length of the shrimps. Using length/weight conversion factor from the equation 2.1.

## 5.4 Number estimation

For determining the number of shrimps in each image, we use two different algorithms. YOLOv7 and U-Net algorithms are used in this project.

### 5.4.1 YOLOv7

The YOLOv7 model uses the bounding box annotation types. The stepwise algorithm for this process is as follows:

Step1: Labelling the data with bounding box type of annotations.

Step2: Augment the data and prepare the dataset.

Step3: Train the model in Google COLAB with the prepared dataset.

Step4: Run the model on the sample image.

Step4: Determine the number of shrimps detected.

## Environmental setup

The table 5-2 shows the necessary libraries and package that the model needs for this model. This model is also conducted on COLAB and hence the data is uploaded to drive for quick access.

Libraries/Packages	Version	Description
Python	3.7.13	Programming language
NumPy	1.24.0	Python library for large multi-dimensional array, matrices, and function support.
OpenCV- Python	4.1.1	Open-source computer vision and machine learning library.
Pillow	7.1.2	Free and open-source python imaging library
TensorFlow	2.4.1	Open-source library for machine learning and artificial intelligence.
Matplotlib	3.2.2	Python module for visualizing the data.
SciPy	1.4.1	Open-source python library for scientific and technical computing.
PyTorch	1.7.0	Python machine learning framework for torch library.
Scikit-learn	0.19.2	Machine learning library for data analysis

Table 5-2: Environmental setup for YOLOv7



## Data labelling

The YOLO model uses rectangle bounding box type of annotations. The labelling of the images is done on an online tool Makesense.ai. This tool is open-source and free to use under GPLv3 license. This tool doesn't need any installations and can be accessed on the browser. It supports multiple label types including rectangle, polygon, point and line. It also provides the labels in several formats such as YOLO, VOC XML, VGG JSON and CSV.

The annotations of the shrimps is done as shown in the figure 5-16. A rectangle shaped bounding box is placed around the shrimps in the image. The annotation in YOLO format is preferred for this model. After finishing the annotations of all the images, the labels and images can be exported in YOLO format.

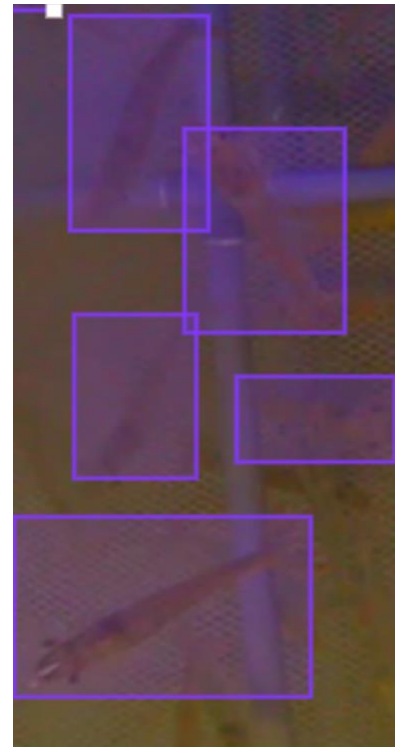


Figure 5-16: Annotations for YOLO

The augmentation is applied to the labelled dataset. 320 images are labelled on Makesense.ai tool and after augmenting the data, the number increased to 1635. This is then split into 1250 images for training and 385 images for validation. The augmentation techniques used to increase the size of the dataset are scaling, rotate, flipping, brightness and blurring.

## Training

After labelling the dataset and exporting it, it is uploaded into google drive for training the model. The training of the model is done on Google COLAB with GPU. After installing the libraries and packages as shown in the table 5-2, the YOLOv7 repository is cloned along with downloading the pre-trained YOLO weights. The dataset is copied into the COLAB and then split into training dataset and validation dataset. The parameters provided for the model while training are as follows:

- Annotated dataset = 320 images
- Augmented dataset = 1635 images
- Train dataset = 1250 images

- Validation dataset = 385 images
- Batch size = 16
- Epochs = 1000

## Running the model

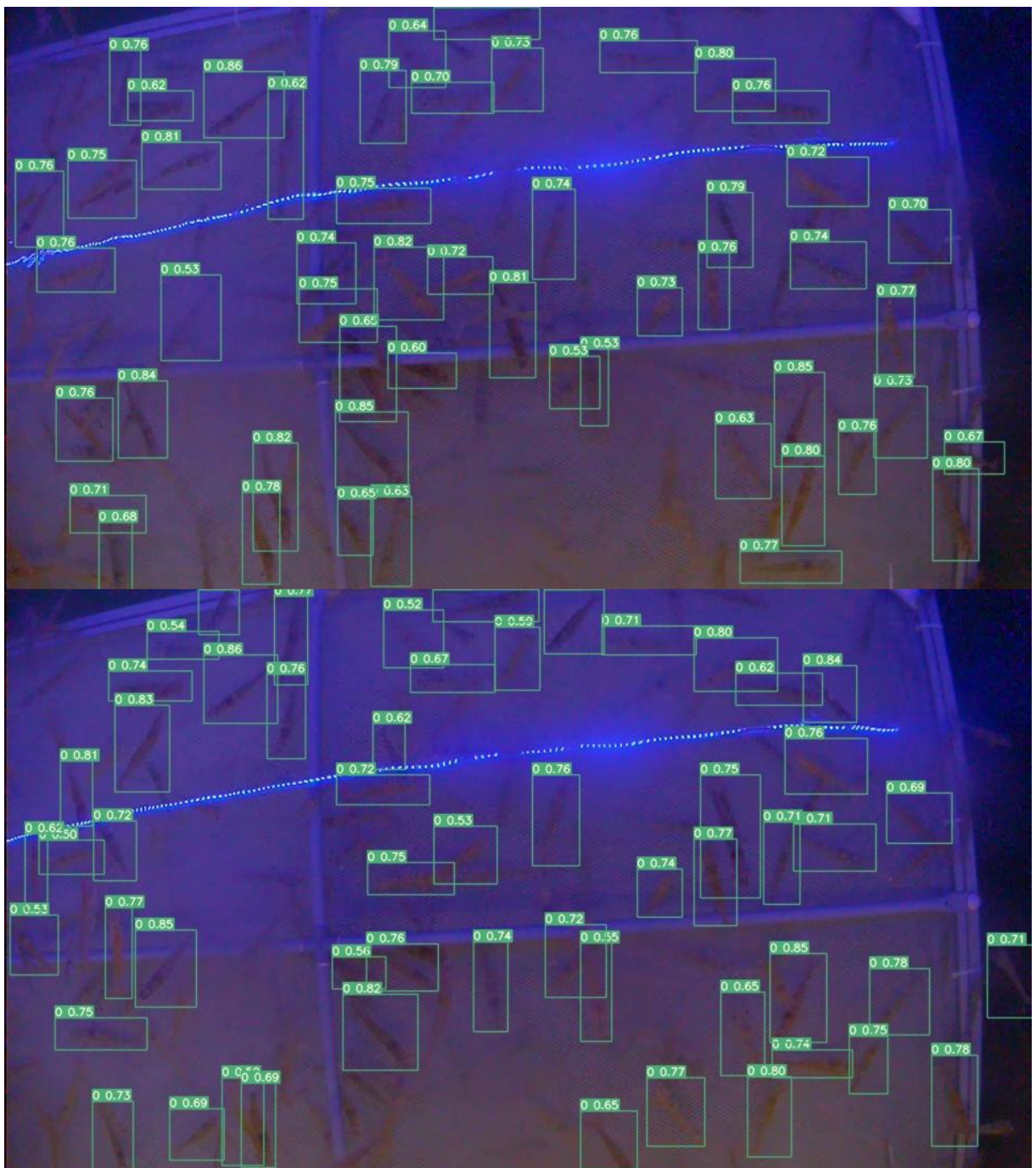


Figure 5-17: YOLOv7 output with bounding boxes

The model is tested on a sample image after training, and the results are displayed in the output in figure 5-17. The model outputs an image with bounding boxes surrounding any detected shrimps as well as a text file containing the bounding box coordinates. The calculating the number of bounding boxes in the text file will give us the number of shrimps detected in the image.

```

0 0.892187 0.465074 0.059375 0.0698529
0 0.117708 0.892463 0.0416667 0.0863971
0 0.0489583 0.875919 0.04375 0.0459559
0 0.752604 0.285846 0.0447917 0.0827206
0 0.0276042 0.926471 0.053125 0.0661765
0 0.296875 0.972426 0.04375 0.0514706
0 0.143229 0.165441 0.0697917 0.0441176
0 0.339063 0.901654 0.0322917 0.101103
0 0.875 0.0450368 0.03125 0.0900735
0 0.722917 0.778493 0.0479167 0.123162
0 0.0921875 0.831801 0.040625 0.0643382
0 0.927083 0.862132 0.0291667 0.154412
0 0.294271 0.902574 0.0510417 0.0661765
0 0.939583 0.704044 0.0291667 0.102941
0 0.433333 0.409926 0.0916667 0.0404412
0 0.934375 0.90625 0.0229167 0.0992647
0 0.442708 0.199449 0.0416667 0.0974265
0 0.257292 0.496324 0.0666667 0.0441176
0 0.122917 0.695772 0.0333333 0.0863971
0 0.317708 0.615809 0.05 0.0808824
0 0.378125 0.91636 0.0291667 0.141544
0 0.0645833 0.957721 0.0291667 0.0772059
0 0.241667 0.0744485 0.0604167 0.0790441
0 0.879167 0.909007 0.0333333 0.123162
0 0.385938 0.221507 0.0239583 0.115809
0 0.394792 0.609375 0.0416667 0.0753676
0 0.528646 0.184743 0.0385417 0.104779
0 0.426562 0.157169 0.065625 0.0533088
0 0.0901042 0.763787 0.059375 0.0533088
0 0.365625 0.345588 0.0875 0.0514706
0 0.657812 0.0496324 0.034375 0.0992647
0 0.100521 0.601103 0.0635417 0.0882353
0 0.148438 0.769301 0.0427083 0.0827206
0 0.756771 0.477022 0.0364583 0.130515
0 0.441667 0.469669 0.0770833 0.0606618
0 0.841146 0.326287 0.0510417 0.0680147
0 0.307292 0.448529 0.05625 0.0919118
0 0.120833 0.487132 0.0291667 0.128676
0 0.271354 0.212316 0.0239583 0.15625
0 0.759375 0.17739 0.075 0.0569853
0 0.689583 0.488971 0.0270833 0.125
0 0.804688 0.452206 0.0677083 0.0625
0 0.635417 0.525735 0.0416667 0.0735294
0 0.169271 0.887868 0.0614583 0.0808824
0 0.436458 0.0670956 0.0520833 0.0680147
0 0.0625 0.45864 0.05625 0.0680147
0 0.360417 0.764706 0.0666667 0.0882353
0 0.809896 0.818015 0.053125 0.143382
0 0.472917 0.0275735 0.102083 0.0551471

```

Figure 5-18: YOLO output with coordinates of bounding box

Figure 6-18 shows the output of a text file with the coordinates of bounding box for all the shrimps detected. The number of shrimps is determined by calculating the number of rows in the text file as each row has one set of coordinates.

### 5.4.2 Density Map

Another model we use in determining the number of shrimps in the image is U-Net. This model uses point annotation type for labelling, and it gives the density map as the output. The stepwise algorithm for this process is as follows:

Step1: Labelling the data with point type of annotations.

Step2: Augment the data and prepare the dataset.

Step3: Split the data and generate train and valid HDF5 files.

Step4: Train the model in Google COLAB with the prepared dataset.

Step5: Execute the model on a sample image.

Step6: Generate the density map for the image.

### Environmental setup

The table 5-3 shows the necessary libraries and package that needed to be downloaded for running this model. This model is also conducted on COLAB and hence the data is uploaded to drive for quick access.

Libraries/packages	Version	Description
Python	3.7.13	Programming language
NumPy	1.16.4	Python library for large multi-dimensional array, matrices, and function support.
H5py	2.9.0	Python package for interpreting HDF5 files
Matplotlib	3.0.3	Python module for visualizing the data.
SciPy	1.2.1	Open-source python library for scientific and technical computing.
Pillow	6.1.0	Free and open-source python imaging library
Torch	1.0.1	Open-source ML library used for creating deep neural networks.
Albumentations	1.3.0	Python library for image augmentation.

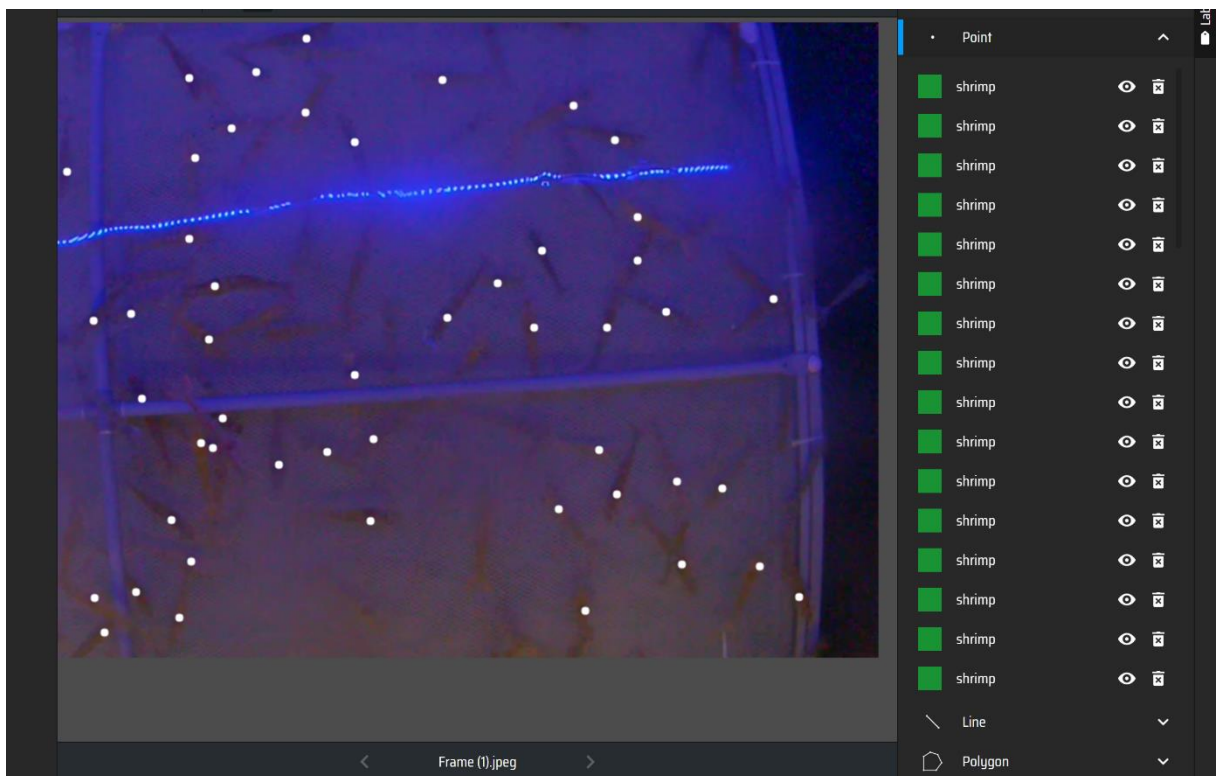
Table 5-3: Environmental setup for U-Net



## Data labelling

For the labelling of the images for the U-Net model, makesense.ai tool is used. This network uses point type of annotations. Since only one key-point is to be labelled on a shrimp, the brightest part or more visibly dynamic part of the shrimp is selected.

From figure 2-1, we can see that the stomach part of the shrimp can be seen clearly because of the contrast in colour. So, decision to annotate the stomach part is taken for this situation. The annotations in the tool makesense.ai looks like the below figure 5-19.



**Figure 5-19: Point annotation for U-Net**

After labelling the images, the annotations are extracted from the project in makesense.ai. In total 109 images were annotated, and the annotations were exported in CSV format. To make the augmentations easy the point annotations are turned to bounding box with coordinates of the point. Then the images with bounding box annotations are augmented and increased the size of the dataset to 763 images. After augmenting the data, the bounding boxes are then turned back into point annotations.

The augmented dataset is then split into 88% for training and 12% for validation. This makes the size of the training dataset to be 671 and validation dataset to be 92 images. The augmentation techniques used in order to increase the size of the dataset for this model are scaling, rotate, flipping and blurring.

## Training

The dataset is split into training and validation datasets, and they are uploaded to google drive for training in Google COLAB. After installing the libraries and packages as shown in the table 5-3, the repository is cloned. The training and validation datasets are then converted to HDF5 format. Then a pickle format, process of converting an object to a byte stream that can be stored as a binary file, is created for training. This pickle file is used for training the model with the parameters provided as follows:

- Annotated dataset = 109 images
- Augmented dataset = 763 images
- Train dataset = 671 images
- Validation dataset = 92 images
- Batch size = 12
- Epochs = 150
- Learning rate = 0.0024

The trained weight of the model is then used to execute the model on an input image to generate the density map and determine the number of shrimps in that image.

## Running the model

After training the model, the generated weights file is used to test the model on a sample image for determining the number. The model gives the output as a number of shrimps detected along with an image with dots where the shrimps are present.

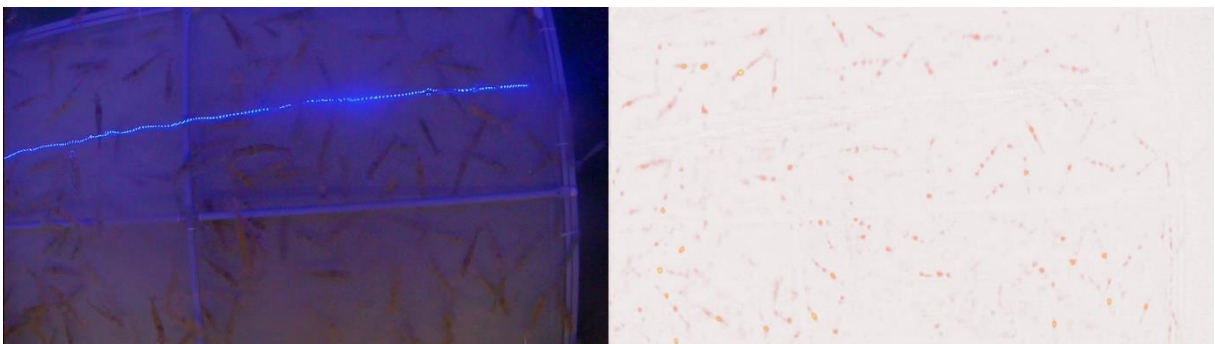


Figure 5-20: Input image and output with density map

Figure 5-20 shows the input image along with the density map generated by the model with dots in the place where shrimps are detected.

## **5.5 Biomass estimation**

With the average weight determined by the instance segmentation with Mask R-CNN and the number of shrimps detected by the density map with U-Net and YOLOv7 models, the biomass is calculated by using the formula  $B = W * N$ , where B is biomass in grams, W is for average weight detected in grams and N is for the number of shrimps detected.

The biomass of the shrimps in the tank is calculated both manually and with the model to verify the results. Since shrimps grow very rapidly, the measurements are done at the same time (12:00) for both manual and automatic. A one-minute video is taken at first and the model works of that. After the video is taken, measuring the biomass manually in the tank is done.

The size of hundred shrimps is measured manually and the average is calculated. While the model calculates the length of the shrimps detected in five frames from the video and the average is calculated. The measurement of the number of shrimps is done by calculating from the image both manually and by using the both the models.

The average length is then used to calculate the average weight by using length/weight factor. This is multiplied with the number of shrimps detected which gives the Biomass in that area. The Biomass is estimated in grams and displayed on the image.

## **Summary of the chapter**

This chapter mainly focuses on the implementation part of the project. This first section focuses on the data processing for the model. It has two subsections: one focuses on data collection and the other focuses on data preparation. In the first section, a test model is developed for experiment and the ideal camera settings and light settings for the better visibility are determined. While the second section gives the basic ideas of data manipulation such as augmentations is discussed along with creating the training and validation dataset.

Later, this chapter focusses on the development of the algorithms discussed in the concept chapter. As discussed in previous chapter, the Biomass calculation requires the estimation of average weight along with the number of shrimps in the image. Each proposed algorithm is discussed in different sections. All the models are trained on Google COLAB because of availability of GPU. Apart from the development, the pre-processing of images such as augmentations, is also discussed.

The first chapter focusses solely on segmentation with Mask R-CNN architecture. The separation of each mask with contour and finding the extreme points from that is the challenging part in this section. The finding of the average weight is the main task for this chapter.

The second chapter deals with the estimation of number of shrimps detected from the image. This section is then split into two subsections. The first subsection focusses on YOLOv7 model while the second one focuses on U-Net. YOLOv7 model gets output with bounding box and a text file with the coordinates of the bounding boxes. U-Net model gets output in the form of density map.

The third section makes use of both sections and calculate the Biomass. In addition to these, the algorithm for each model is discussed with stepwise algorithm. The libraries/packages that needs to be downloaded are covered in this chapter.



# 6 Results and evaluation

This chapter mainly focuses on the results obtained from the previous chapter and their evaluation. While the instance segmentation with Mask R.CNN is used in determining the average weight of the shrimps in the image, the number of shrimps estimation can be done by any of the two models discussed in the previous chapter. Here, the results are compared to determine which model performs better.

## 6.1 Segmentations

### 6.1.1 Outputs

In the previous chapter, we discussed about the proposed algorithm of the instance segmentation for calculating the average weight and its result. In this chapter, we will run the model on more frames and compare the output. This chapter also deals with some of their limitations in different scenarios.

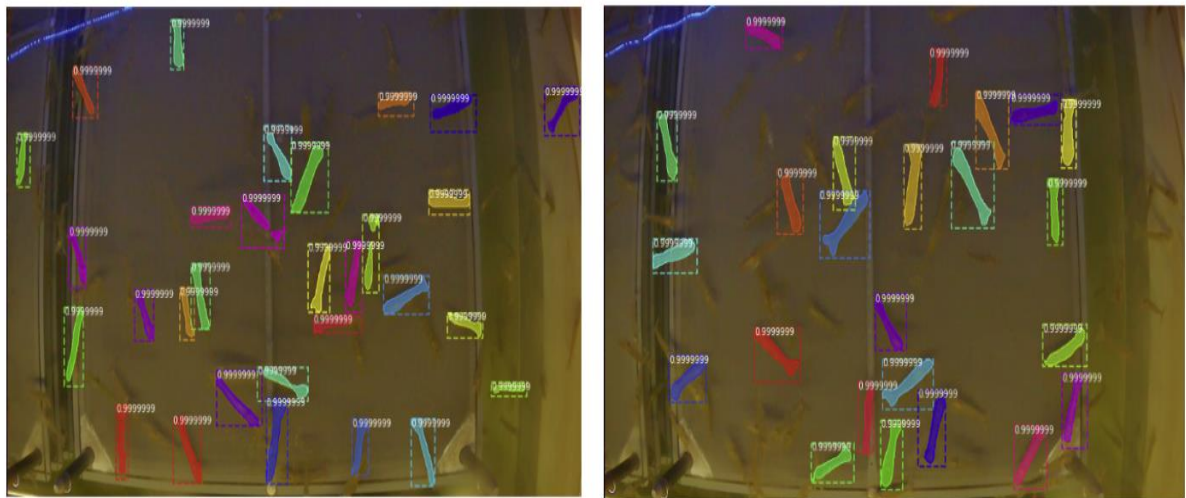
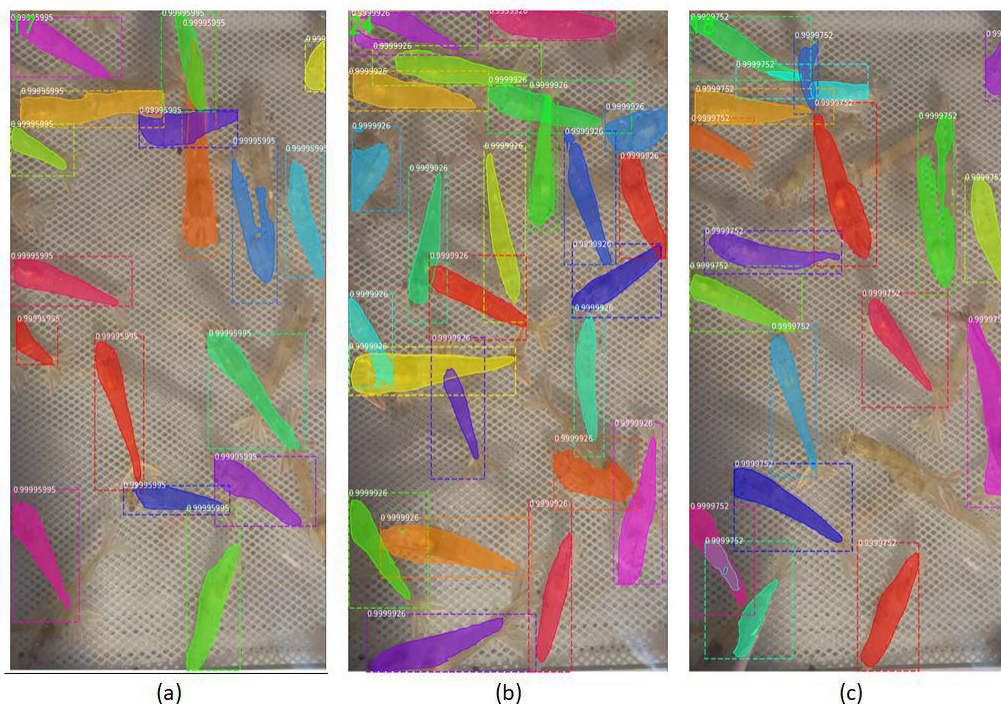


Figure 6-1: Output of the segmentation

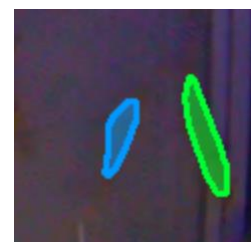
The algorithm detected 32 shrimps in the first image while it only detected 23 in the second image. Since, shrimps of same age are kept together in the tank, the model doesn't need to detect all the shrimps for average weight calculation.

When the sediment formation in the water tank increases, the visibility of the shrimps drastically decreases as evident by the following figure 7-1. When the model is applied on the images with good visibility, the prediction for the shrimps also improves as demonstrated in the figure 7-2.



**Figure 6-2: Shrimps detected in the images from different camera**

The bad visibility of the shrimps in the image leads to the partial detection of shrimps and this affects the average length calculations. The partial detection of a shrimp can be seen in the figure 7-3. This leads to the estimated length of that individual shrimps to be less than the actual length. This affects the average length of the shrimps detected.



**Figure 6-3: Partial detection**

To eliminate this, we eliminated these types of partial detections before the calculation of average length. For this, the partner business established a formula that states that the shrimp's length must be at least eight times their width. Using this, the partially detected shrimps are eliminated.

### 6.1.2 Evaluation

The lengths of the shrimps were calculated manually in the shrimp farm before the introduction of the ML techniques. To check the validity of the data predicted, the lengths of 100 shrimps are calculated and compared the average length. Since shrimps grow very fast, the calculation of the length both manually and using the model are done on the same day. The video was taken from the installed camera right before the manual calculation.

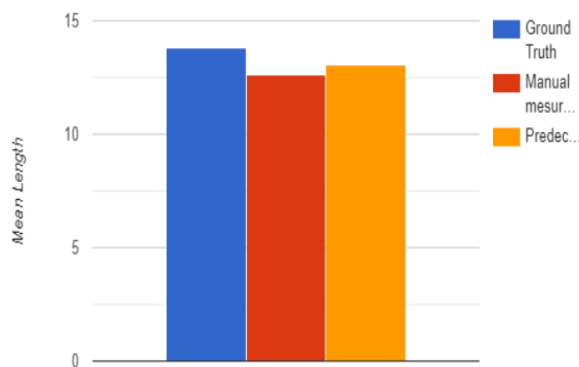


Figure 6-4: length comparison

- Ground truth = 13.8cm
- Manual measurement = 12.6cm
- Predicted length = 13.07cm

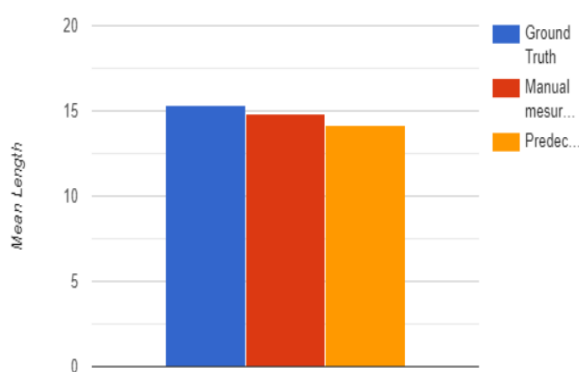


Figure 6-5; Second tset length comparison

- Ground truth = 15.3cm
- Manual measurement = 14.8cm
- Predicted length = 14.2cm

The average length of a total of 100 shrimps is calculated at two different time both manually and using the model. The comparison is as shown in the above figure 7-4 and 7-5. Ground truth is the average length measured manually at the farm while the manual measurement is the average length manually measured in the computer using scaling factor. The predicted length is the average length predicted by the model.

## 6.2 YOLOv7 results

### 6.2.1 Outputs

The outputs of the YOLOv7 model would be an image with bounding box around the shrimps and a text file with the coordinates of every bounding box. The coordinates of a single bounding box is in one row of the text file and therefore, when we calculate the number of rows in the text file, we get number of shrimps detected.

The output of the YOLOv7 for a sample image is shown along with the input image in figure 7-6. The image comes along with a text file with the coordinates.



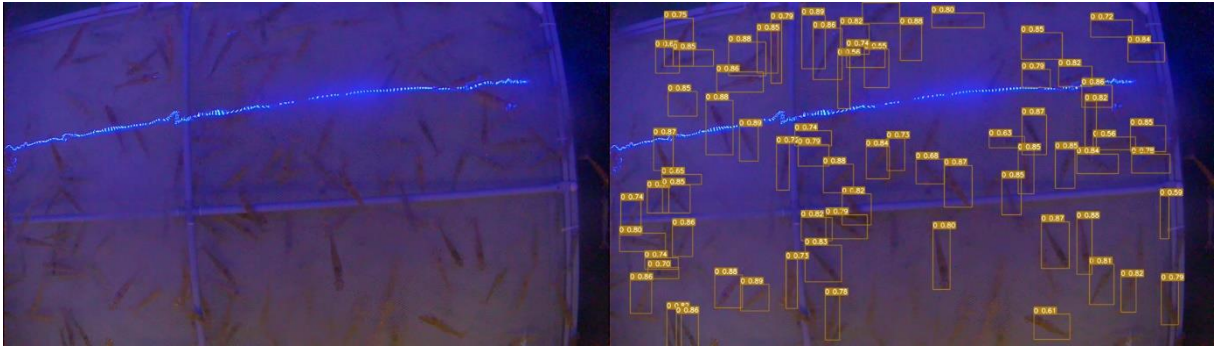


Figure 6-6: Input and output of YOLOv7 model

## 6.2.2 Evaluation

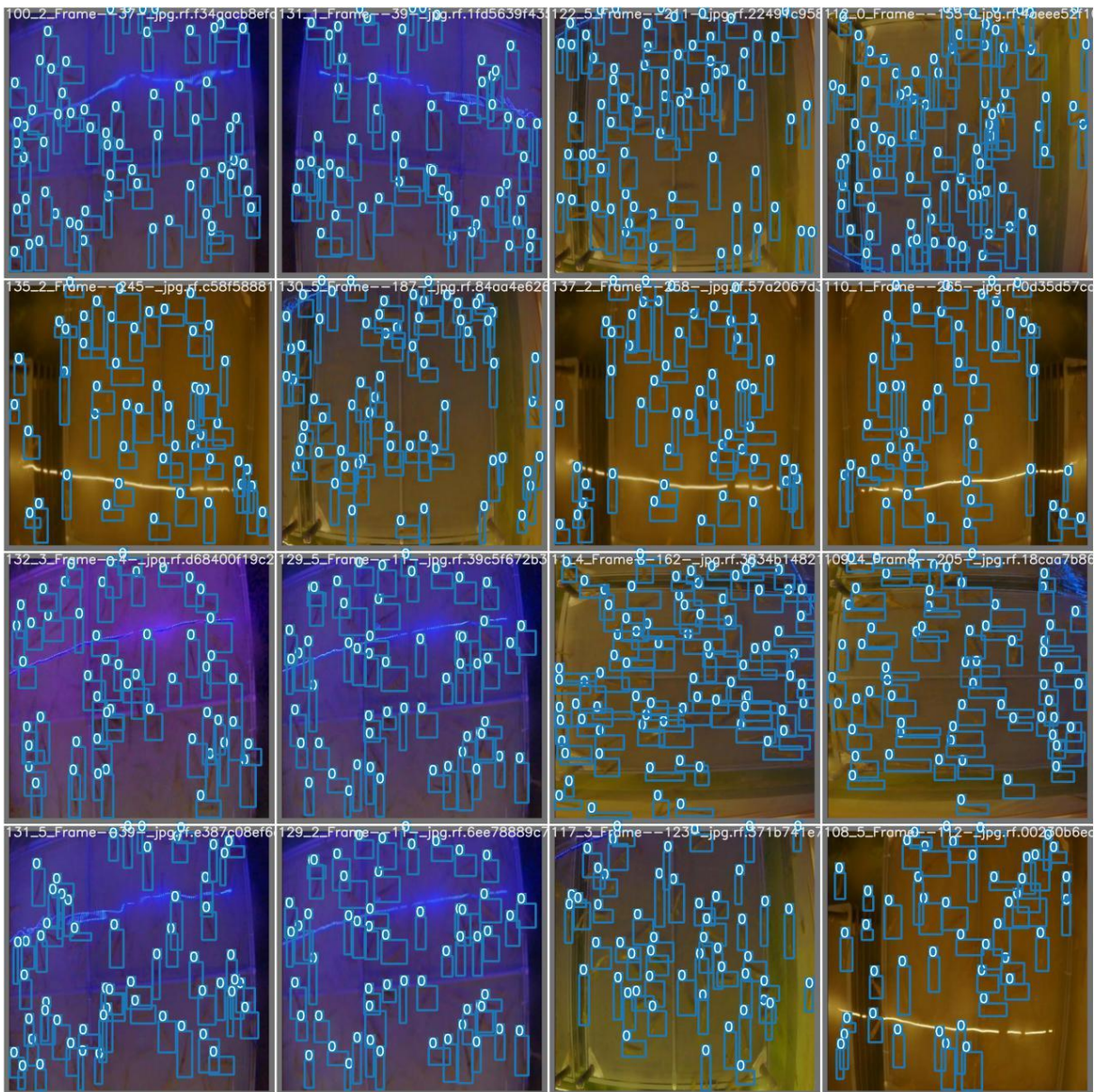


Figure 6-7: YOLOv7 on test dataset

When the model is run on a test dataset of 21 images, to evaluate the model, the output of the model is as shown in the figure 7-7. The model is then evaluated, and the mean average precision (mAP) is calculated. The mAP of 0.969 is achieved with this model. The output for the evaluation step in the model is as shown in the figure 6-10.

```

!python /content/yolov7/test.py --weights /content/YOLO_weights.pt --data /content/data.yaml --task test --name yolo_det
Namespace(augment=False, batch_size=32, conf_thres=0.001, data='/content/data.yaml', device='', exist_ok=False, img_size=640, iou_thres=0.65)
YOLOv7 v0.1-121-g2fdc7f1 torch 1.13.0+cu116 CUDA:0 (Tesla T4, 15109.75MB)

Fusing layers...
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
IDetect.fuse
/usr/local/lib/python3.8/dist-packages/torch/functional.py:504: UserWarning: torch.meshgrid: in an upcoming release, it will be required to
return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Model Summary: 314 layers, 36481772 parameters, 6194944 gradients, 103.2 GFLOPS
Convert model to Traced-model...
traced_script_module saved!
model is traced!

test: Scanning '/content/yolov7/Shrimp-yolo-1/test/labels' images and labels... 21 found, 0 missing, 0 empty, 0 corrupted: 100% 21/21 [00:00:00.00]
test: New cache created: /content/yolov7/Shrimp-yolo-1/test/labels.cache
          Class      Images      Labels      P      R      mAP@.5      mAP@.5:.95: 100% 1/1 [00:03:00:00, 3.15s/it]
          all         21         1249      0.957      0.938      0.969      0.629
Speed: 17.6/2.5/20.1 ms inference/NMS/total per 640x640 image at batch-size 32
Results saved to runs/test/yolo_det3

```

Figure 6-8: mAP calculation

YOLOv7 can also detect the shrimps in a video and when a video is given as a input the model divides the video to frames and detect the number of shrimps in each frame. A video of 10 seconds is given as input to the model and the statistics of it are shown in the table:

Parameters	Values
Duration of the video	10 seconds
Frames per second (FPS)	30
Total frames	300
Average number of shrimps per frame detected/predicted by the model	65
Average number of shrimps per frame (manually counted)	68
Precision	0.9558

Table 6-1: Details of YOLOv7 on video

## 6.3 Density map results

### 6.3.1 Outputs

The U-Net gets the output in the form of a density map. The figure 7-9 shows exactly how the density map looks like.

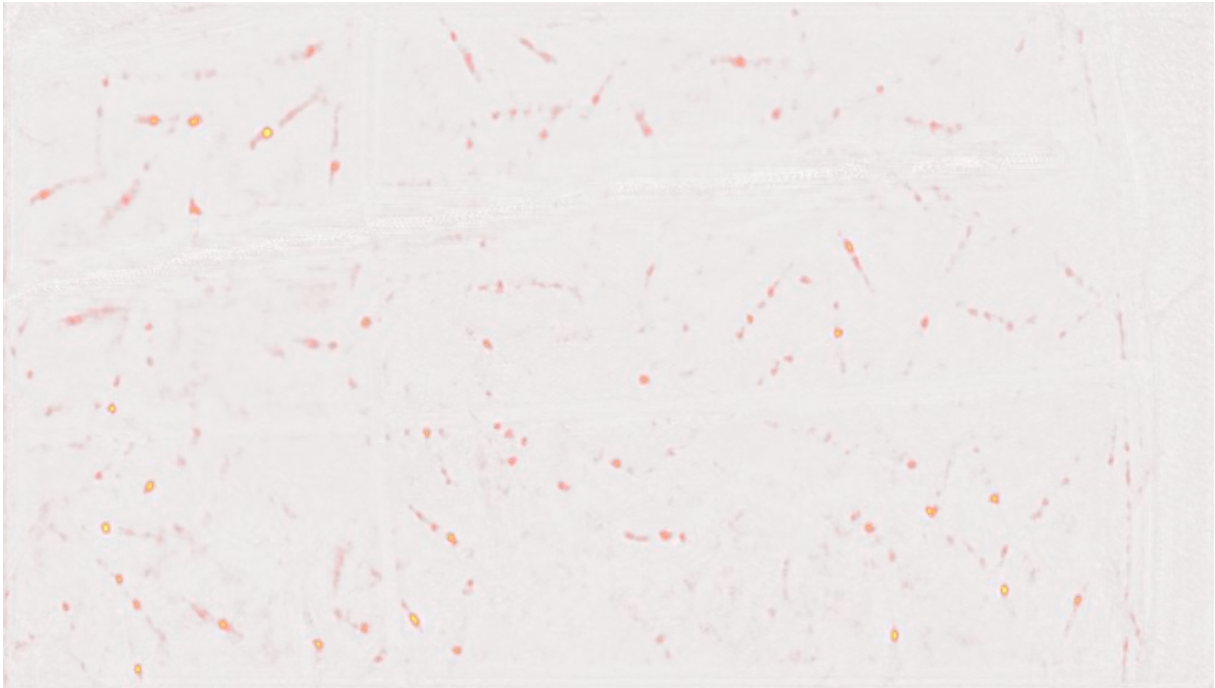


Figure 6-9: Density map

The model also gives the number of objects detected as the output as shown in the figure 7-10. Here the model detected 84 shrimps from the sample image.

## 5. Testing our model

For testing the model, create input and output folders and add the image to input folder for testing.

```
[ ] !python /content/infer.py -o /content/output -n UNet -f /content/input -c /content/shrimp_UNet.pth --visualize
```

```
The number of objects found: 84.178857421875
```

Figure 6-10: Number of shrimps using U-Net

```
-----  
Epoch 150
```

```
Train:
```

```
Average loss: 0.1102  
Mean error: 0.276  
Mean absolute error: 7.290  
Error deviation: 9.671
```

```
Valid:
```

```
Average loss: 0.1472  
Mean error: 5.884  
Mean absolute error: 7.167  
Error deviation: 7.327  
-----
```

Figure 6-11: Average loss values for U-Net



The average training loss of 0.1102 is obtained by the U-Net model. The training losses and valid losses at the end of the training can be seen in the figure 7-11. When the input image is superimposed on the density map, we will get the output as shown in figure 7-12.

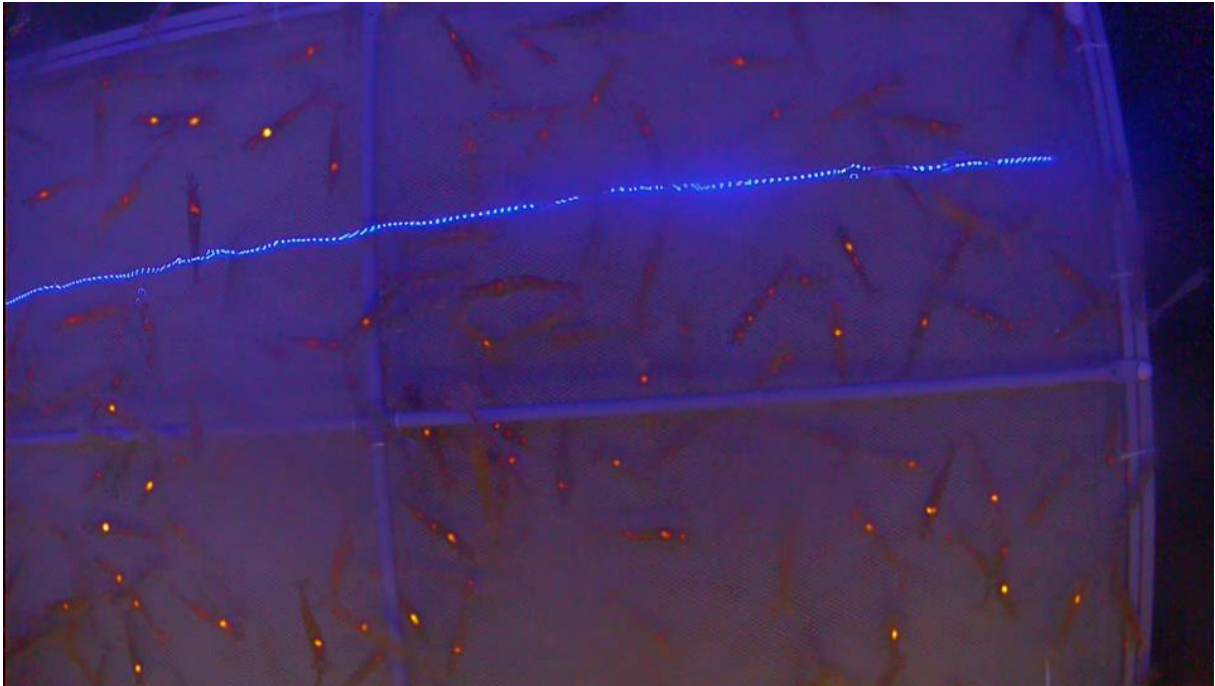


Figure 6-10: Density map with U-Net output

#### 6.4 Comparison between Density map and YOLOv7

Either of the models can be used for the calculation of number of shrimps. Both the models are tried for 5 frames to compare, and the results are shown in the table 7-2.

	YOLOv7	U-Net	Manual
Frame 1	78	83	85
Frame 2	77	79	86
Frame 3	84	88	90
Frame 4	88	87	91
Frame 5	76	79	84
Average	80.6	83.2	87.2
Error	7.59	4.6	

Table 6-2: Comparison between YOLOv7 and Density map

The YOLOv7 has an error rate of 7.59 while U-Net has an error rate of 4.6. The U-Net performed better than YOLOv7 for the images. But, YOLOv7 can work on videos while U-Net model does not. So, for calculating the Biomass in an image, U-Net performs better and hence it is preferred.

## 6.5 Biomass determination

The average weight calculated from the segmentation and the number of shrimps calculated with U-Net model, biomass can be determined by multiplying them. The calculated biomass is then displayed on the image as shown in the figures below.

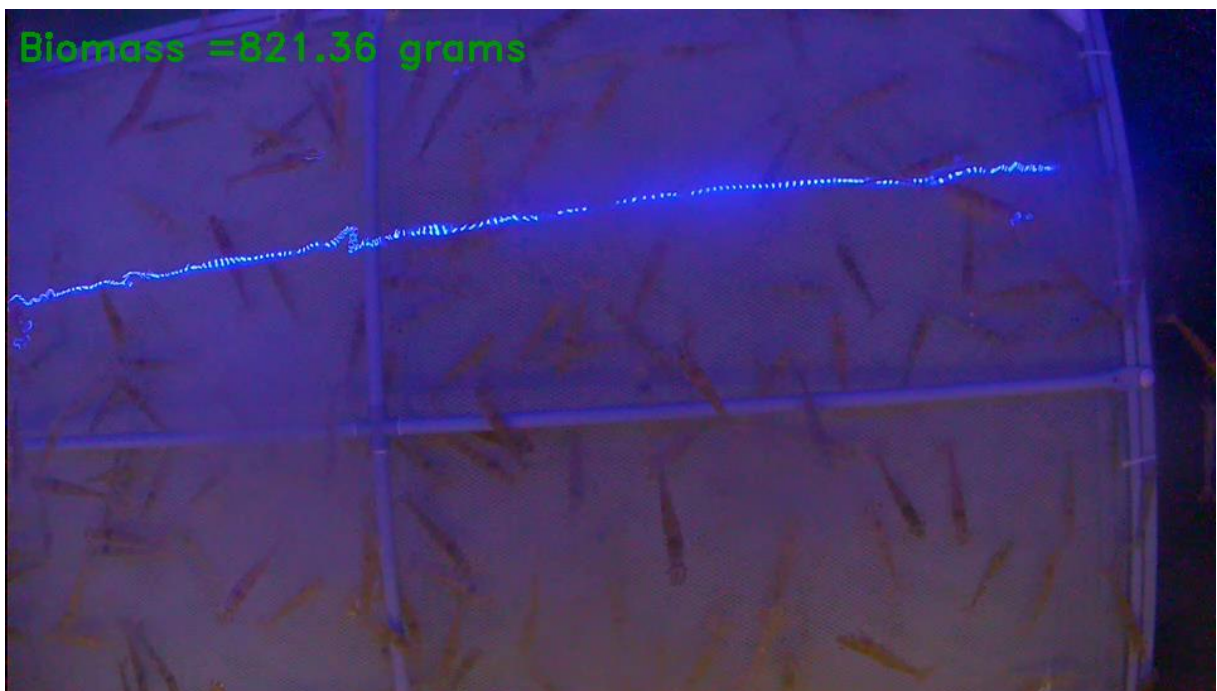


Figure 6-11: Biomass displayed on the image-1

Figure 7-11 shows the image with biomass of 821.36 grams while the image in figure 7-12 have the biomass of 737.22 grams.

The biomass calculated here is for the entire image area and for validation the biomass calculated manually in the farm is then compared with these values.

For this as we know the area of the entire farm and the area of the pool, the camera covers, we calculated the error rate. The model has an error rate of 7%. This can still be reduced with the better cameras and clear water.



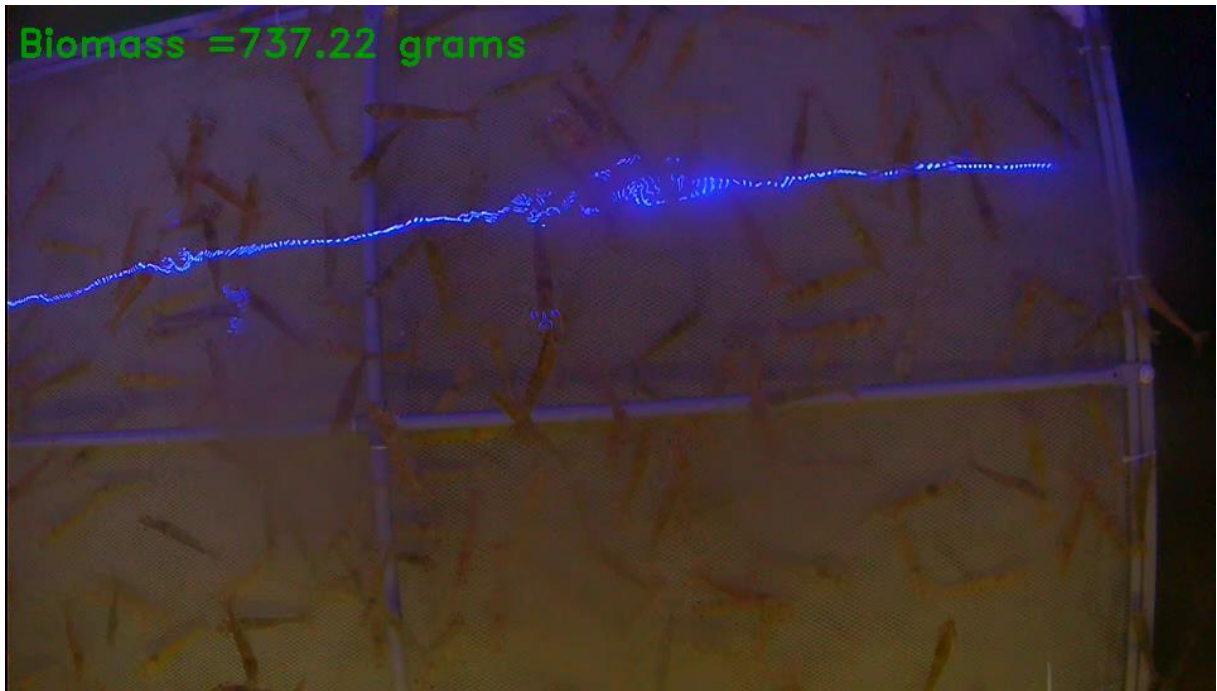


Figure 6-12: Biomass displayed on the image-2

**Evaluation:**

The model is then tested on 100 images and the output of the model is then compared with the manually measured data. The output is as shown in the following table.

	Segmentation	Density map	YOLOv7
Manually calculated	14.2	84.3	80.2
Model detected	14.8	88.6	88.6
Detection rate (%)	95.9	95.1	90.5

Table 6-3: Detection rate

The rate of detection for the model using the segmentation and density map is

$$\text{Detection rate} = 0.959 \times 0.951 = 0.912$$

$$\text{Detection rate (\%)} = 91.2\%$$

The detection rate for the model using segmentation along with YOLOv7 is

$$\text{Detection rate} = 0.959 \times 0.905 = 0.867$$

$$\text{Detection rate (\%)} = 86.7\%$$

## **Summary of the chapter:**

This chapter solely focusses on the results and evaluation of those results from the previous chapter. It have five sections with each section focussing on different models and their evaluations.

The first section mainly discusses about the results for size estimation of the detected shrimps. It has two subsections, one for results and the other for evaluation. First subsection shows the output for the size estimation by instance segmentation using Mask R-CNN on sample images and partial detection is also discussed in this. The second subsection focusses on evaluation process of the model. The results with the manual measurement are compared here.

The second section Focusses on the number of shrimps estimation with YOLOv7 model. Similar to the previous section, this section also has two subsections: one for results and the other for evaluation. In the same way, third section discusses about the results and evaluation of the density map model with U-Net.

The fourth section compares the results from the second and third sections to determine the better model to detect the number of shrimps in the image. The fifth section focusses on the Biomass calculation by combining the models for number of shrimps and size estimation. In the evaluation part of this section, detection rate is calculated for the model by using the manually measured data.

# 7 Conclusion and Future Scope

In this chapter, after seeing the results in the previous chapter, we can summarize the overall performance of the proposed algorithms to be very good. The first task in this project is to calculate the weight from instance segmentation with Mask R-CNN architecture, In the second task of estimating the number of shrimps in the image, Density map with U-Net model performed better than the YOLOv7 model.

Based on the previous chapters and the results obtained in chapter 7, these are some of the advantages and limitations of the models used in the thesis.

Advantages:

- The shrimps which are fully visible are detected and the lengths of these shrimps are detected with a good accuracy.
- The YOLOv7 model achieved 0.969mAP with the test dataset while density map with U-Net got the mean absolute error of just 7.29 and 7.167 for training and validation respectively.
- Achieved the precision of 93 for calculating the biomass of the shrimps when compared to the manual measurements.

Limitations

- The model suffered when the visibility of the shrimps in the water is low and due to the poor visibility, the tails of the shrimps are not detected properly.
- Overlapped shrimps are hard to detect completely and these partially detected shrimps can bring down the average length.
- Labelling of the images with low visibility by using polygon type annotations is challenging.

## **Future scope**

In the future, to enhance the scope of the project, improving the water and camera quality will drastically improve the predictions. Making the feeding completely automatic will improve the mental state of the animals. Due to the limitations in the camera quality, identifying the color of the shrimps is nearly impossible. But if we can identify the color, we can identify the early signs of stress in the animals and we can even know, what's causing the animals' stress.

With improvements in both the water quality and camera quality, the visibility of the shrimps improves which makes surveillance of the animals better. Tracking of the animals and their movements can provide very insightful data in assessing the health and the mood of the animals. This also helps in identifying the aggressive animals and the damage of they cause.

With the development of underwater cameras, the tracking of the shrimps can be done even closely. The stereo cameras can be used to gather the data in 3D and it gives the size of the shrimps precisely. Since the layers of the tank in this situation is quite low, usage of stereo camera is not priority. But, with the tanks with more depth needs stereo camera for the accurate measurement of the size of the shrimps.

In the future, the model can be trained with even more data to improve the prediction. Also, re-training the model with several different network backbones and comparing the result would make this better.

## 8 References

- [1] J. Tjong. "MonitorFish Home". Internet: <https://monitorfish.com/en/company/>, Dec. 12, 2022 [Jan. 2, 2023], [Online].
- [2] T. Pillay. *Aquaculture and the Environment*. John Wiley & Sons, 2008.
- [3] T. Hastein, A.D. Scarfe & V.L. Lund. "Science-based assessment of welfare: aquatic animals". *Revue Scientifique et Technique-Office International des Epizooties*, Vol. 24.2 (2005): 529., Aug 2005.
- [4] J.M. Ebeling, M.B. Timmons. "Recirculating aquaculture systems.". *Aquaculture production systems*, Vol. 1 pp: 245-277., Feb 2012 .
- [5] J. Bostock, B. McAndrew, R. Richards, K. Jauncey, T. Telfer, K. Lorenzen, D. Little, L. Ross, N. Handisyde, I. Gatward, R. Corner. "Aquaculture: global status and trends.". *Philosophical Transactions of the Royal Society B: Biological Sciences*, Vol. 365.(1554), pp: 2897-2912, Sep 2010.
- [6] "Globalsea food," [Online] Internet: <https://www.globalseafood.org/blog/what-is-aquaculture-why-do-we-need-it/>, [Accessed: Dec. 20, 2022]
- [7] B. Zion. "The use of computer vision technologies in aquaculture—a review,“. *Computers and electronics in agriculture* Vol. 88, pp:125-132, Oct 2012
- [8] A. W. Fast, L.J. Lester. *Marine shrimp culture: principles and practices*. Elsevier, Oct 2013..
- [9] T.E. Britannica, "Shrimp,“ Encyclopedia Britannica, [Online] Internet: <https://www.britannica.com/animal/shrimp-crustacean>, [Accessed: Jan. 2, 2023].
- [10] "Dr. Eckel" [Online] Internet: <https://www.dr-eckel.de/en/strengthen-shrimp-against-salinity->, [Accessed Dec.20, 2022]
- [11] Abhik Banerjee, "Insulator Fault Detection using Image Processing“, Master Thesis, Department of computer science, TU Chemnitz, Aug. 2018.
- [12] Y. Shirai. *Three-dimensional computer vision*. Springer Science & Business Media, 2012.
- [13] S. J. Prince. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012..
- [14] J. D. Kelleher. *Deep learning*. MIT press, 2019..
- [15] E. Bisong. *Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners*. Apress, 2019..

- [16] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis. "Deep learning for computer vision: A brief review". *Computational intelligence and neuroscience*, Feb. 2018.
- [17] K. O'Shea, R. Nash. "An introduction to convolutional neural networks". *arXiv preprint*, Vol. arXiv:1511.08458, Nov. 2015.
- [18] S. Albawi, T.A. Mohammed, S. Al-Zawi. "Understanding of a convolutional neural network". In *2017 international conference on engineering and technology (ICET)*, IEEE, 2017.
- [19] S. Albawi, T.A. Mohammed, S. Al-Zawi. "Understanding of a convolutional neural network," In *2017 international conference on engineering and technology (ICET)*, IEEE, 2017.
- [20] Z.Q. Zhao, P. Zheng, S.T. Xu, X. Wu. "Object detection with deep learning: A review". *IEEE transactions on neural networks and learning systems* Vol. 30.11, pp: 3212-3232, Jan. 2019.
- [21] K. Oksuz, B.C. Cam, S. Kalkan, E. Akbas. "Imbalance problems in object detection: A review". *IEEE transactions on pattern analysis and machine intelligence* Vol. 43.10, pp: 3388-3415, Mar. 2020.
- [22] R.M. Haralick, L.G. Shapiro. "Image segmentation techniques". *Computer vision, graphics, and image processing*, Vol. 29.1, pp:100-132, Jan. 1985.
- [23] "stanford.edu", Internet: <https://ai.stanford.edu/~syyeung/cvweb/tutorial3.html>, [Online], [Accessed Dec 20, 2022].
- [24] A.M. Hafiz, G.M. Bhat "A survey on instance segmentation: state of the art". *International journal of multimedia information retrieval*, Vol. 9.3, pp: 171-189, Sep. 2020.
- [25] K. He, G. Gkioxari, P. Dollär, R. Girschik. "Mask r-cnn". *Proceedings of the IEEE international conference on computer vision*, pp: 2961-2969, 2017.
- [26] R. Ke, A. Bugeau, N. Papadakis, M. Kirkland, P. Schuetz, C.B. Schönlieb. "Multi-task deep learning for image segmentation using recursive approximation tasks". *IEEE Transactions on Image Processing*, Vol. 30, pp: 3555-3567, Mar. 2021.
- [27] A. M. Jois. "Determining the Size of Shrimp Using Computer Vision," Master Thesis, Department of Electrical Engineering and Information Technology, TU Chemnitz, 2022.
- [28] J. Lauer, M. Zhou, S. Ye, W. Menegas, S. Schneider, T. Nath, M.M. Rahman, V. Di Santo, D. Soberanes, G. Feng, V.N. Murthy. "Multi-animal pose estimation, identification and tracking with DeepLabCut". *Nature Methods*, Vol. 19(4). pp: 496-504, 2022.

- [29] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. "You only look once: Unified, real-time object detection". *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 779-788, 2016.
- [30] S. Saleh, S.A. Khwandah, A. Heller, A. Mumtaz, W. Hardt. "Traffic signs recognition and distance estimation using a monocular camera". In 6th International Conference Actual Problems of System and Software Engineering.[online], Moscow, 2019, IEEE (pp. 407-418).
- [31] A. Bochkovskiy, C.Y. Wang, H.Y.Liao. "Yolov4: Optimal speed and accuracy of object detection". *arXiv preprint*, Vol. arXiv:2004.10934 (2020).
- [32] C.Y. Wang, A. Bochkovskiy, H.Y. Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". *arXiv preprint*, Vol. arXiv:2207.02696 (2022).
- [33] N. Kautsky, P. Rönnbäck, M. Tedengren, M. Troell. "Ecosystem perspectives on management of disease in shrimp pond farming". *Aquaculture*, Vol. 191.1-3, 2000, pp:145-161.
- [34] W.P. Chen, L.K. Wang, T.T. Wang, Y.T. Chen " An Intelligent Management System for Aquaculture's Environmental Monitoring and Energy Conservation". *2013 International Workshop on Computer Science in Sports*, Atlantis Press, 2013, pp: 194-198..
- [35] U. C. Barg. *Guidelines for the promotion of environmental management of coastal aquaculture development*. Food & Agriculture Org, 1992.
- [36] M.J. Allison, R.W. Dougherty, J.A. Bucklin, E.E. Snyder. "Ethanol accumulation in the rumen after overfeeding with readily fermentable carbohydrate". *Science*, Vol, 144.3614, pp: 54-55.
- [37] A. Martínez, Y. Romero, T. Castillo, M. Mascaró, I. López-Rull, N. Simóes, F. Arcega-Cabrera, G. Gaxiola, A. Barbosa. "The effect of copper on the color of shrimps: redder is not always healthier". *PloS one*, Vol. 9.9, p.e107673.
- [38] G. Picone, S.B. Engelsen, F. Savorani, S. Testi, A. Badiani, F. Cappozi. "Metabolomics as a powerful tool for molecular quality assessment of the fish *Sparus aurata*". *Nutrients*, Vol. 3(2), 2011, pp: 212-227.
- [39] "Agrifarming". [Online], Internet: <https://www.agrifarming.in/ras-fish-farming>, [Accessed on Jan.3, 2023].
- [40] R. Crab, T. Defoirdt, P. Bossier, W. Verstraete. "Biofloc technology in aquaculture: beneficial effects and future challenges". *Aquaculture*, Vol. 356, 2012, pp: 351-356.

- [41] "The fishsite," [Online]. Internet: <https://thefishsite.com/articles/why-small-scale-tank-based-shrimp-production-is-on-the-rise>, [Accessed on Jan. 3, 2023].
- [42] "Edureka," [Online]. Internet: <https://www.edureka.co/blog/what-is-deep-learning>, [Accessed on Jan. 10, 2023].
- [43] O. Ronneberger, P. Fischer, T. Brox. "U-net: Convolutional networks for biomedical image segmentation". *International Conference on Medical image computing and computer-assisted intervention*, Springer, Cham, 2015, pp: 234-241.
- [44] V. Varatharasan, H.S. Shin, A. Tsourdos, N. Colosimo. "Improving learning effectiveness for object detection and classification in cluttered backgrounds". *Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, IEEE, 2019, pp: 78-85.
- [45] Uranchimeg Tudevdagva, Batbayar Battseren, Wolfram Hardt, Galina V. Troshina. "Image Processing Based Insulator Fault Detection Method". *Technische Hochschule Ingolstadt Zentrum für Angewandte Forschung (ZAF)*, 2018, pp: 579-583.
- [46] Shadi Saleh, Shanmugapriyan Manoharan, Wolfram Hardt. "Real-time 3D Perception of Scenes with Monocular Camera". *Embedded Selforganising Systems*, Vol. 7(2), pp: 4-7.



# Appendix

## Animal welfare

Factors affecting the welfare of the animals are as follows:

### Environment

#### I. Water Quality

The most crucial element to take into account when maintaining a healthy aquaculture is the farm's environment. Quality of water is one of the main issues in assuring the health of the animals. There is a direct relation between quality of water and the disease control. Changes in normal environmental conditions such as oxygen levels, water temperature and salinity of water have a significant effect on the health of the animals [33].

#### II. Density

With highly intensive farming, the density of the shrimps in the ponds tends to be very high. Many shrimps living in a single close quarter leads to animals accidentally hitting by swimming into each other. Sometimes this may cause severe injury but, most of the times it increases the stress on the animals. So, maintaining the density of the animals to a safe number is very important.

Shrimps usually grow very fast and when the different sizes of the shrimps live in a small pond together, the effect on small shrimps is huge. To prevent this, the shrimps must be transferred to different ponds based on the size in order to reduce the density in the ponds [34]. Also, with increase in density, the risk of disease transmission increases significantly.

#### III. Sediment Management

The bottom of the tank will always become clogged with waste from aquafarms, including uneaten food and feces. [35]. High amount of sediment formation in the tank results in decrease in availability of oxygen. The sediments also release toxic amount of ammonia, nitrate and phosphate. All these factors influence the health of animals.

When the farms are located in a naturally available water bodies, then the sediment increases the nutrients in the nearby farms. But, in the case of man-made tanks, the sediment causes only harm to the animals. To keep the water quality high, the sediment in the tank needs to be cleaned at the conclusion of each crop cycle.

#### **IV. Overfeeding**

Overfeeding is one of the significant factors in the formation of sediment at the bottom of the tank. The uneaten food usually ends up at the bottom of the tank. This uneaten food breaks down and produce toxic byproducts which effects the water quality and thereby animal health.

When the uneaten food decays it produces carbon dioxide while consuming oxygen. This results in decreasing levels of oxygen availability for the animals. This also results in decrease of the pH value of the water making it acidic and not safe for animals. The decaying of food also produces high amounts of ammonia and nitrate which are considered dangerous for the aquatic animals [35].

Overfeeding also causes liver issues in the animals. This leads to a disease called fatty liver which is highly common in aquatic animals [36]. With increase in uneaten food, the water quality and the visibility in water decreases. With low water visibility, the animals swim into each other more often and resulting in the increase of stress.

#### **V. Stress**

One of the key elements affecting an animal's health is stress. Shrimps are usually highly sensitive and stress easily. Stress is a psychological response to any harmful stimuli. Stress even for a short time might affect the health of the shrimps and long term may result in death of the shrimps. Shrimps stress for even smallest of changes in the environment. This affects their immunity and make them vulnerable to diseases. There are several stressors for shrimps in a tank.

##### **i. Physical stressors**

Any small changes in the water or the surroundings near tank results in stress. Some of the physical stressors related to water quality are changes in water temperature, oxygen levels and pH value. Sudden change in the light intensity also stresses the

animals. Any sudden sounds in the surrounding area of the tank is also one of the stressors.

## **ii. Biological stressors**

Biological stressors include density of the shrimps in the tank and presence of parasitic macro-organisms. With the high density in the tank, shrimps have less space to swim around and this will lead to unwanted collisions. Hence, the non-availability of lateral swimming space increases the stress in the shrimps. Also, the presence of pathogenic or parasitic macro-organisms also increases the stress.







## **iii. Chemical stressors**

Usage of chemicals in the cleaning of tank also causes stress in shrimps. Chemicals used in water treatment for any insects or spills is one of the stressors. Sediment in the tank and release of toxic waste results in changes in pH.

## **iv. Procedural stressors**

The common procedures in maintaining the tank such as cleaning of tanks or changing of tanks, water changing, shipping and disease control are procedural stressors.

Shrimps have several physical attributes to indicate that they are stressed such as erratic movement, lethargy, loss of appetite, loss of color, decreased reproduction rate and molting.

	<p>Necrotic tissue (white discoloration) in abdominal segment also called as white tail disease. This is caused by muscle spasms caused by stress</p>
	<p>Small but not yet completely melanised (blackened) lesion on the abdominal segment</p>
	<p>Small but completely melanised(blackened) lesion on the abdominal segment</p>
	<p>Cataracts visible as white spots on the eyes</p>
	<p>Short antennae are a sign of bacterial contamination</p>
	<p>Red spot on the tail is also a sign of stress</p>

**Figure 9-0-1: Disease evidence in the shrimp [37]**

The color of the shrimps as shown in the Figure 9-1 is a good indicator for the stress or any diseases in the shrimps.

- Erratic movement: One of the simplest signs that the shrimp are anxious is their irregular movement in the aquarium. Stressed shrimps tend to swim erratically, and this leads to bumping and scrapping.
- Lethargy: Lethargy in shrimps can be identified easily as the shrimps generally are active animals. Loss of enthusiasm or energy generally indicates the shrimps are unhealthy.
- Loss of appetite: Shrimps generally eat any organic matter in the tank such as algae, dead plants or animal matter. Hence, the loss of appetite is an indicator for compromised immunity.

- Loss of color: When agitated or ill, shrimp frequently lose color. Some of the issues and what they indicate is shown in Figure 9-1.
- Decreased reproduction rate: Generally female shrimps can carry up to 50 eggs depending on the size. Stress in shrimps can affect the fertility rate.
- Molting: A new exoskeleton is developed during molting in preparation for growth. Any problem in this process can lead to loss of life and stressed shrimps usually have a problem at this stage.

Stress can be easily identified by change in color and early symptoms are as shown in the Figure 9-0-1. Appearance of red dots on the tail and shrimps turning to pale are symptoms for terminal diseases.

## **Shrimp farms**

Shrimp farms grew in scale significantly to meet the global demand. These exist in both fresh water or salt water environments.

### Recirculating aquaculture systems (RAS)

RAS is a technology used to maintain water quality and oxygen levels to achieve desired environment for fish growth. In comparison outdoor pond system cannot sustain in long term as maintaining the ideal environment for the growth is not possible. The RAS system is a closed-circuit system with continual cleaning of water, removal of waste and maintaining the optimum pH value of water [9].

These systems are sustainable as they use significantly less water than the conventional methods. Most of the water is recycled and thereby reducing the water wastage. These recirculating systems are non-dependent on the external environmental factors and the internal environment is completely controlled for optimum growth. Factors like feeding, disinfecting, temperature of water and so on are controlled in this method. To produce and maintain the system properly for producing excellent outcome, deployment requires knowledge, experience and perseverance [10].

The typical RAS system involves tanks, filters, UV disinfectors and oxygen enrichment units. It can be further modified based on the customization needed in that moment such as pH regulator, denitrification unit or heat exchanger. The water passes from

shrimp tank through the various filters both biological and mechanical, getting rid of carbon dioxide before returning to the tanks.

There are several advantages in using this system and some of them are listed below

- Low water usage
- Efficient usage of resources such as land and energy
- Maximum production
- Ease of growing and harvesting
- Production throughout the year
- Complete control of environment resulting in less stress for shrimps
- Less water pollution

There are also few constraints in using this method such as requirement of good water source, skilled staff and uninterrupted electricity.

### Biofloc technology

Biofloc technology makes use of waste and uses it as fish food. The pond's heterotrophic bacterial growth is accelerated by the addition of carbohydrates, and nitrogen production rises as a result of the creation of microbial proteins. [11]. In laymen's terms, this system produces nutritious fish food from the toxic waste by making use of added microorganisms.

Biofloc technology also has its own advantages. Some of them are

- Minimizes water exchange
- Improved water quality
- More economical alternative

### Tank-based systems

This system is gaining popularity in recent years. The major reason for this is its proximity to the market. This type of technology can be used in any environment and is simple to maintain. The floor and the top must be insulated properly.

To assure the health of the shrimps, adequate ventilation is important. This method has 2000-3000 post-larvae per m<sup>3</sup> harvest rate [12].

Name: <input type="text" value="Tammineni"/>	<b>Bitte beachten:</b>
Vorname: <input type="text" value="Gowtham Chowdary"/>	1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.
geb. am: <input type="text" value="03.05.1993"/>	
Matr.-Nr.: <input type="text" value="456818"/>	

Selbstständigkeitserklärung\*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende **Masterarbeit**  selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum:

Unterschrift: *T. Gowtham* .....

\* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.