iFish Farm: Monitoring and Analysis of Fish Anomaly Behavior In Ubiquitous Environment

by

Hussam Eldin, Youssef Wageeh, Omar Anas and Ali Fadl

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Computer Science

in

Department of Computer Science

in the

Faculty of Computer Science
of the
Misr International University University, EGYPT

Thesis advisor:
Dr. Ayman Ezzat, Dr. Ayman Nabil, Eng. Noha ElMasry

(July 2020)

Abstract

The activities of managing fish farms, like fish ponds surveillance , are one of the tough and costly fish farmers' missions. Generally, these activities are done manually, wasting time and money for fish farmers. A method is introduced in this document which improves fish detection, fish trajectories and detects abnormal fish behaviors where the water conditions is challenging. Image Enhancement algorithm is used at first to improve unclear images of water. Object Detection algorithm is then used on the enhanced images to detect fish. Then, features like fish spreadness and speed are calculated from the coordinates of the detected objects. In the end, a classifier is used to detect several abnormal behaviors in fish ponds. Our system aims for better fish surveillance over fish ponds in fish farm.

Acknowledgments

We would like to thank our supervisors Dr. Ayman Ezzat, Dr. Ayman Nabil and Eng. Noha ElMasry for their guidance, encouragement and advice they have provided throughout our project. We have been extremely lucky to have supervisors who cared so much about our work, and who responded to our questions and queries so promptly. Particularly, we would like to thank our university (Misr International University) for supplying us with all help needed. Also, we would like to express our gratitude to the Fish Research Center and the University of Suez Canal especially Dr. Safaa Sharaf and Eng. Mohamed Abdelmohsen for helping us and supplying us with all the assistance required.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

1.1.1 Background

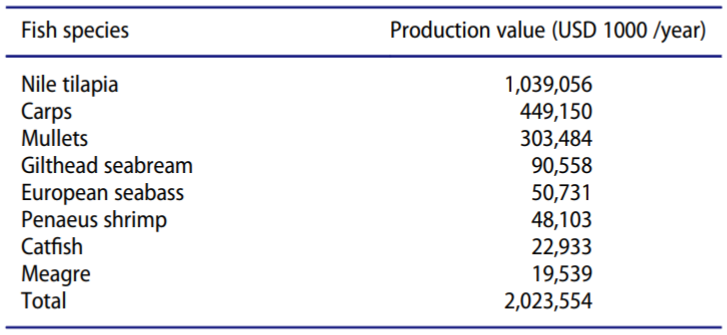
Fishing has evolved in a variety of forms over time, until people actually come to the notion of producing their own fish, which became the start of fish farms [1]. At the present time, fish farms are very relevant as they have a strong effect on the environment by maintaining a steady production and consumption of fish worldwide. Fish farms are known for their value as a consequence of several organisations such as FAO (Food and Agriculture Organization), World Bank and several other organisations have contributed to the development of fish farms by 2030 [2]. Our system aims to aid fish farmers with a solution to their most time/labor intensive tasks to help them focus on their fish production and other more important tasks.

1.1.2 Motivation

1.1.2.1 Market Motivation

Since the country is putting a lot of focus on fish farms in the mean time for economical growth, they suffer from many problems as they have to check the water quality frequently, they need continuous surveillance over the fish ponds to check if the fish are dying due to any cause, while keeping track of fish size. They also need to track fish behaviors so a system is needed to monitor fish health, size, count, feeding times, and check the water quality. Fish farming is a costly and tedious process that requires a lot of labor work, more than

67% of the cost of a fish farm goes to labor work [3]. According to Shaalan et al. [4] the aquaculture business in Egypt provides about 77% of national fish production and provides about 580,000 jobs for workers in this part. In addition, The estimate for aquaculture fish production exceeds USD 2 billion annually, as shown in figure 1. Also, globally, In 2017, the top ten countries produced 71.2 million tonnes of fish, they made up 88.9% of the global fish production [5]. Fish provides more than half the population with at least 15% of their average consumption of animal protein per capita [6]. Also, since 1950, the global fish supply has multiplied 8 times [6].Case studies were done in several countries to calculate the production value of aquaculture industry [7]. The total aquaculture production value of these countries was around USD 71 million farm gate value, which is the difference between the market value and selling costs (transport and marketing costs), which stands for 72% of the global aquaculture production value [7].



Fish species	Production value (USD 1000 /year)
Nile tilapia	1,039,056
Carps	449,150
Mullets	303,484
Gilthead seabream	90,558
European seabass	50,731
Penaeus shrimp	48,103
Catfish	22,933
Meagre	19,539
Total	2,023,554

Figure 1.1: Fish production value in Egypt

1.1.2.2 Academic Motivation

J. Boom, X. Huang et al. developed a tool that analyzes footage that counts the population of the fish underwater by automatic video processing software [8]. Rodriguez et al. proposed a computer vision based system that uses image processing algorithms to study fish behavior and size in the pond [9]. Luo , Li et al. present a precise and automatic algorithm for the recognition and counting of fish in video footage of fisheries activities [10]. Parra et al. propose a system using sensors to control the water quality and fish behavior in fish ponds [11]. Also, Microsoft and Gramener [12] used deep learning AI models to monitor fish species. Moreover, Google X team utilized underwater camera systems in oceans capable of identifying and evaluating various activities of fish that are not noticeable to the human eye [13].

1.1.3 Problem Definitions

Reducing the time and costs needed to maintain a fish farm by offering real-time feedback on water quality, while predicting various causes of fish behavior in the pond at a decent accuracy and giving reports of all fish behaviors and water status.

1.2 Project Description

Providing real-time detection of fish behavior in the fish farm pond while checking water quality for toxicity to reduce fish diseases or deaths in the pond.

1.2.1 Objective

The primary aim of the project is to offer an easy to use application that can help the fish farmers to monitor their farm in more efficient and easy way. The application will notify the farmers of any changes by offering cheap and real-time observation of any anomalies in the fish pond as speed and size while checking water quality for any toxic ammonia that causes fish death.

1.2.2 Scope

The system will cover a number of things within its scope:

1. System will check on ammonia levels using image processing techniques.
2. System will detect any anomalies in fish pond.
3. System will alert farmers after detecting fish abnormal behaviors.
4. Alert to farmer on detecting toxic level of ammonia.

1.2.3 Project Overview

The IFish Farm system aims to ease the monitoring of fish farms to the fish farmers. The system is divided into several phases. Firstly, the data is collected from a camera above the pond to get the video footage of the fish to detect abnormal behavior and another camera underwater that takes a picture periodically of the ammonia paper to detect toxic ammonia levels. Secondly, the preprocessing phase comes where the video footage and images are enhanced and compressed to get a smaller size and better quality of the video/picture on a cloud server to speed up the process. Thirdly, the classification part where we can detect fish using YOLO algorithm and detect the color of ammonia paper using thershold classfier. After that, features are extracted from the detected fish like there speed, direction, skewness, coordinates and many other features. Then, the abnormal behavior classification is applied using Random Forest algorithm to detect the fish behavior and detect any abnormal behavior. Lastly, the farmer can monitor the farm through the web application and generate reports.

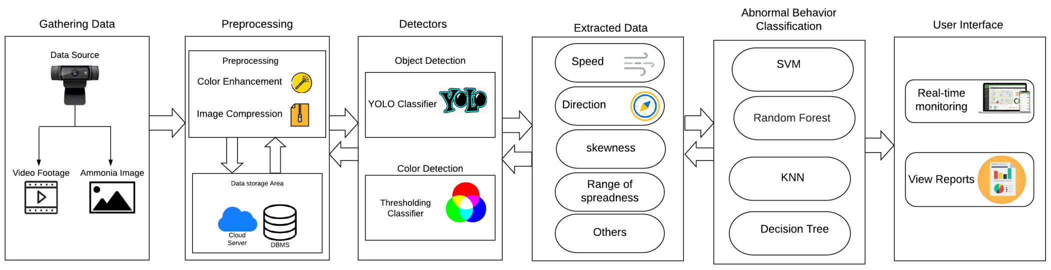


Figure 1.2: System Overview

1.3 Project Management and Deliverable

1.3.1 Tasks and Time Plan

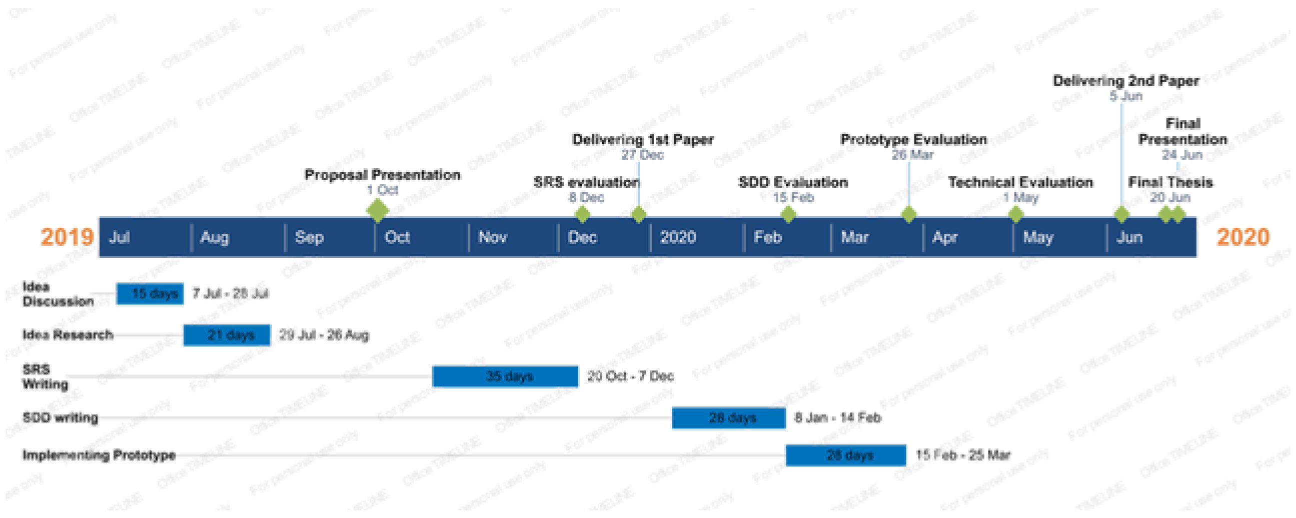


Figure 1.3: Time Plan

1.3.2 Budget and Resource Costs

Items	Price
Seachem Ammonia Paper	25 USD
Wireless webcam	40 USD
Fish pond	25 USD
Fish pond equipment	25 USD

Table 1.1: Items Purchased

Chapter 2

Literature Work

2.1 Related work

2.1.1 Fish detection and tracking

Various methods have been done to detect fish in order to track their count and size. For detection of fish different object detection algorithms have been applied [14], [15]. To track fish size and count image processing and computer vision systems are considered [15], [16], [9], [17]. In order to track fish movement, tracking algorithms like optical flow and frame subtraction are done [18], [19].

Duggal et al. [14] wanted to create a model that automatically describe the video through object detection algorithms. Explanation of a video content is an easy task for a human being to do, but it is a complicated and difficult task for computers. They used the YOLO object detection algorithm as a base for the proposed system. Their proposed model gives better results compared to the other two models as it's faster and got less memory overhead. They used YOLO object detection algorithm which will be used by us to detect and count fish.

Lumauag et al. [15] motivation was to rely on computer vision to count fish as manual counting is a difficult process. The problem with manual fish counting is that it consumes much time and causes eye fatigue. The researchers used image processing techniques (blog analysis and euclidean filtering) to automate the process of counting fish. The system sometimes had issues with over-counting and/or under-counting. Over counting was caused due to lighting conditions. Their stated accuracy was 94% for successful detection and 91%

for successful counting. The paper is useful as a good basis for counting fish from the same camera position that we are going to use.

Toh et al. [16] wanted to automate counting fish in a pond to help giving accurate feeding as counting fish for humans is time-consuming and is subjected to errors. They found an easy method with high accuracy and less computational complexity that count fish. Firstly, they used the background estimation technique to obtain the initial blob. Then, they remove the noise from the image. After that, the remaining blobs are only fish so to detect a single fish they used median area of all blobs. Out of 30 frames, only one frame got an error in counting of 2 excess fish. This paper inspires the idea of fish counting and gives some specific details as background estimation and background subtraction to improve images to get accurate fish count.

Rodriguez et al. [9] have done this paper to study biological changes on fish such as size change based on a stereo system using an image processing algorithm. Their main problem was getting an accurate estimation of fish size in the pond as it may indicate many factors in fish. Firstly, They detected the fish by using the distance map obtained by the stereo-vision system using an image processing algorithm. Then, they estimate the size of the fish by a segmentation technique to detect fish in the region of the RGB space corresponding to the location in the disparity map. They got only 10% error rate in estimating fish size and 90% precision rate. This paper helps us in detecting fish size by providing fish detection techniques based on stereo-vision system and segmentation algorithms so we get an accurate fish size estimation.

Boom et al. [17] aim to study the effects that climate change and pollution has on the environment. Long-term monitoring of the underwater environment is labor-intensive work and other ways of data collection are also labor-intensive. They offered a system that detects and tracks fishes then recognizes the fish using its color and other attributes. Their system is still not fully functional, but so far their system shows a detection and tracking rate of 79.8% with an 11.8% false detection rate. This paper is useful to us as it introduces the idea of covariance based fish tracking, along with multiple background subtraction methods to improve our fish detection.

Chen et al. [18] propose a new method based on optical flow to track any moving object. It's always tough to track an object's contour in complicated scenes. Firstly, they use an algorithm to get the velocity vector. Then, they get the object's contour by getting the position of moving pixels between frames. Finally, they calculate the position of the object and speed by using the position values. Their results showed accurate tracking of objects while the camera is motionless. This paper helps us to track fish movements by providing an optical flow algorithm that is based on calculating position and velocity of the moving object.

Nguyen et al. [19] provide an algorithm to improve the tracking of fish movement. Their problems in tracking fish were showing an illusion of a fish, motionless fish and fish moving at different speeds at different times. They proposed a method that solve all these cases by combining frame difference and Gaussian mixture algorithms. Their proposed algorithm gives better results compared to the other 4 algorithms as it tracks fish in different cases. This paper is helpful to our research as it explores the use of Gaussian Mixture Model in background estimation to detect the fish at a high accuracy in low water quality, while also introducing the use of Kalman Filter to track the detected fish at a high accuracy in difficult conditions.

2.1.2 Abnormal Behavior

Various methods have been done to detect abnormal behaviors in different domains. Detecting fish trajectories and classifying them into normal and abnormal behaviors are considered in [20]. Clustering and feature descriptors algorithms were used to detect abnormal behaviors in human motions [21]. Object Detection algorithm was used as a base for detecting abnormality in different events [22], [23].

Beyan et al. [24] designed a system to analyze fish trajectories and classify them into normal trajectories and abnormal trajectories. The problem with the traditional way of analyzing fish behavior (human visual inspection) makes this task time consuming and limits the number of processed videos. They developed an approach to detect abnormal fish trajectories using an outlier detection method which is based on cluster cardinalities and a distance function. As an average of class accuracies, their system shows 71% accuracy which is as they claim the best in this field. This paper is useful for us as it gives us the first steps to be able to detect abnormal fish behavior in water.

Nady et al. [21] presented this paper as surveillance cameras have become ubiquitous by reason of growing security matters and low costs of equipment. Conventional visual control relies on human analysis of videos which is ineffective with a big number of cameras and causes fatigue due to long monitoring. They proposed a system that depends on Spatio-temporal representation of videos and STACOG descriptors to identify abnormal events. The proposed system processing time is faster than the best competing system by 26%. This paper is helpful as it presents algorithms to detect abnormal behaviors in videos.

Yang et al. [23] developed a framework that detects behavior changes in crowds, on the both local and global level, to increase the safety and security of public places. Crowd monitoring is difficult because of the many variables included, like inconsistent lighting conditions and shadows. Their proposed system firstly detects the people as objects using YOLOv2, clusters them into groups using fixed-width clustering, and then analyzes each group's movement patterns to detect any change in behavior. Their proposed system achieved a higher accuracy than five other methods after testing on 6 different video sequences with an accuracy between 80% and 95.7%. This paper is helpful because depends on uses YOLO for the object detection phase, similar to our system, and it also provides a unique method of detecting behavioral changes compared to traditional methods used before.

Wang et al. [22] aim was to detect abnormal behaviors of hens in different times of day to improve their breeding. It was challenging to differentiate between the collected images by image processing techniques as they were all look alike. They developed a system based on an object detection neural network-based algorithm called YOLOv3. It has some special parameters like batch processing pattern and learning rate which were adjusted to satisfy their needs. The mean accuracy is based on six different behaviors of hens where each behavior has its' own accuracy where the highest accuracy was the mating behavior with 94.72% and the lowest accuracy was the drink behavior with 86.88%. The paper is useful as it detects abnormal behaviors of hens based on YOLO algorithm which is the algorithm used in our system to detect fish.

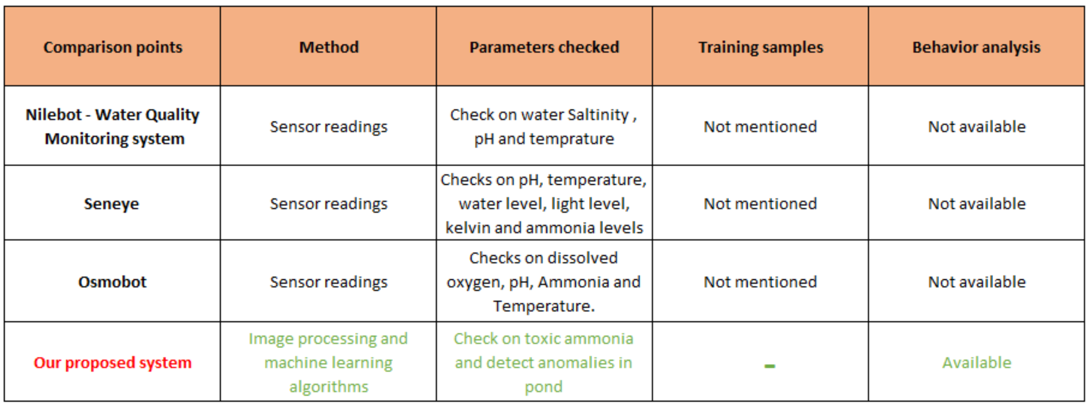
2.1.3 Unclear Water

In order to get better results in tracking and detection of fish, we enhance unclear underwater images in fish ponds [25],[26].

Tang et al. [25] proposed a system that enhances turbid underwater images to get a nearly natural color of the image. Their primary issue was that pictures and videos are generally rather poor in marine settings with a non-uniform illumination, color degradation and low contrast due to the marine environment. They proposed an image enhancement method based on Retinex algorithm which enhances images under different underwater conditions. They compared their algorithms with other 4 enhancing algorithms and found out that their method is better and faster than other algorithms in most of the cases. This paper introduces the Multi-Scale Retinex algorithm which will be used by us to enhance unclear underwater images to get better results in detecting fish.

Lu et al. [26] wanted to create a new and fast algorithm to enhance images underwater by reducing noise level and improving global contrast. Taking images underwater is challenging as it always suffers from light distortion and scattering. They proposed a model consisting of trigonometric bilateral filters which are responsible for noise removal and edge-preserving and ACE-based technique that colors the distorted images. They compared their model with other models and found out that their model gives better results than others with better computational complexity. This paper is useful for our fish detection accuracy as it introduces an enhanced and quick color correction method named , which is an enhanced version of the method based on the ACE model that takes a long time in processing.

2.2 Comparison with Proposed Project



Comparison points	Method	Parameters checked	Training samples	Behavior analysis
Nilebot - Water Quality Monitoring system	Sensor readings	Check on water Saltinity , pH and temprature	Not mentioned	Not available
Seneye	Sensor readings	Checks on pH, temperature, water level, light level, kelvin and ammonia levels	Not mentioned	Not available
Osmobot	Sensor readings	Checks on dissolved oxygen, pH, Ammonia and Temperature.	Not mentioned	Not available
Our proposed system	Image processing and machine learning algorithms	Check on toxic ammonia and detect anomalies in pond	-	Available

Figure 2.1: Comparisons

Chapter 3

System Requirements Specifcations

3.1 Introduction

3.1.1 Purpose of this document

The purpose of this document is to provide a full description of how the iFish Farm monitoring system works. The monitoring system is an online web-based application system to monitor the fish ponds of a farm to ease the maintenance process for farmers to reduce the labor. This System Requirements Specifcations (SRS) will describe the aim of the system and its functionalities. In addition, the document will show all constraints on the system, all interfaces' designs and all diagrams that were needed to build the system.

3.1.2 Scope of this document

The purpose of the IFish Farm system is to ease the monitoring process and to create a convenient and easy-to-use web application for farmers to monitor their fish farms. The system is based on an unsupervised learning methodology to cluster the different fish behaviors occurring in the farm while providing necessary alerts to the farmers based on the events detected and suggesting a solution to the problem of the event. We will have a cloud server that will process the captured footage to be done there to speed up the monitoring process.

3.1.3 Overview

This document describes most of the system diagrams and architectures. It also previews how the system main functionalities work and how the user views and interacts with the software. The sections in this document gives a detailed description for the diagrams that help the developer developing the system. It includes the class diagrams, sequence diagrams and the 4 architectures diagrams.

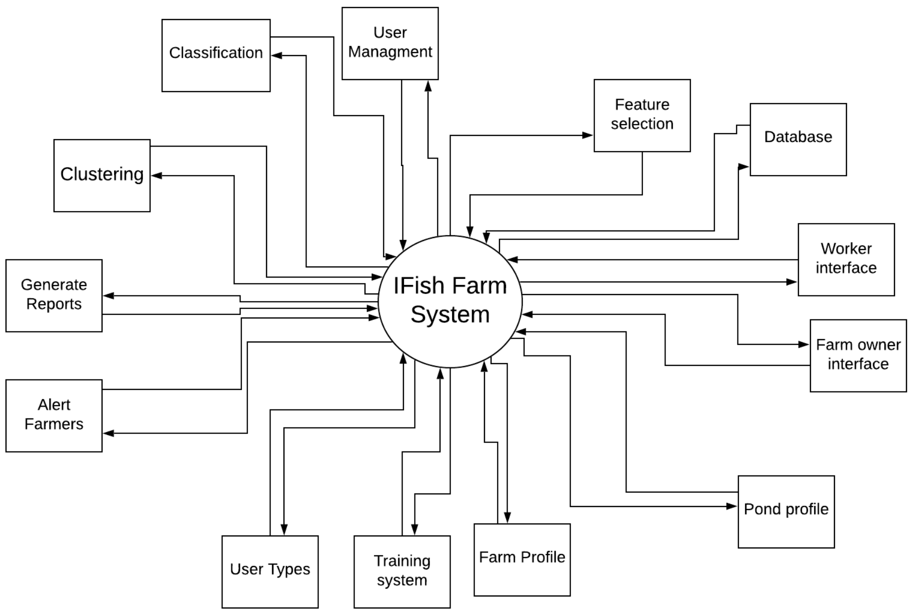


Figure 3.1: Context Diagram

Firstly,two cameras, a camera for detecting toxic ammonia and a camera for detecting anomalies in the pond,They will be connected to a mobile device/ arduino device. The device will send the raw data to the cloud storage. The cloud storage then sends the data to be pre-processed. In the pre-processing, the image will be enhanced to remove any water turbidity. Our code will be written and converted to an API so we can use it on the device to provide notification of pond status. After the API gets the image and video it will be divided into two sections. The first section is for processing and enhancing the colors in the video footage. Firstly, the video will be analyzed then the behavior will be detected after that the behavior will be clustered as normal or abnormal also speed, abnormal size .. etc will be detected. The second section is for processing the image of the paper to detect toxic level of ammonia. The color will be detected and classified as toxic or non toxic ammonia level in water. All of the results of the processed data will be sent to a database. Our application then retrieves this data and shows it to the farmer on the webapp or notify the farmer on his device about any abnormal events that happen in the pond.

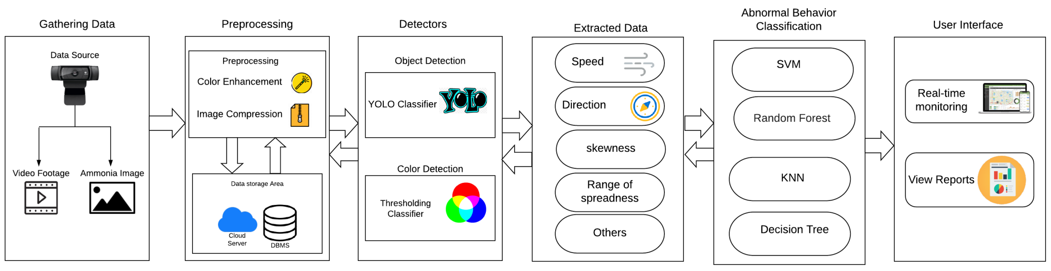
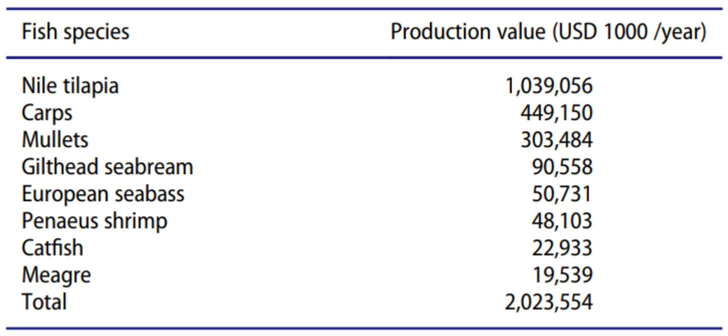


Figure 3.2: Block Diagram

3.1.4 Business Context

Since the country is putting a lot of focus on fish farms in the mean time for economical growth, Fish farms suffer from many problems as farmers have to check the water quality frequently and ammonia levels in water, they need continuous surveillance over the fish ponds to check if the fish are dying to know that the pond is having a problem and then they have to discover the cause, while keeping track of fish size so they can be moved to other ponds. so a system is needed to monitor fish behavior, health, size, count, feeding times, and check the water quality. According to Shaalan et al. [4] the aquaculture business in Egypt provides about 77% of national fish production and provides about 580,000 jobs for workers in this part. In addition, The estimate for aquaculture fish production exceeds USD 2 billion annually, as shown in figure 3.1.4.



Fish species	Production value (USD 1000 /year)
Nile tilapia	1,039,056
Carps	449,150
Mullets	303,484
Gilthead seabream	90,558
European seabass	50,731
Penaeus shrimp	48,103
Catfish	22,933
Meagre	19,539
Total	2,023,554

Figure 3.3: Fish production value in Egypt

3.2 General Description

3.2.1 Product Functions

”IFish Farm” is a system who’s main function is to offer a monitoring system for fish farms to detect any anomaly in the fish ponds and fish behavior also detect any change in the ammonia level in water to alert the workers and ease there jobs

3.2.2 User Characteristics

In this document, we proposed a system that deals with Fish farms issues, there-fore, the system is user friendly enough that any user is able to use it and it can be installed in most conditions.

3.2.3 User Problem Statement

The main problems fish farmers suffer from are the fast and unpredictable change in ammonia level in the water which cause instant death of fish in the fish farm ponds and the need for continues surveillance on the fish ponds for detecting any abnormal behavior in the ponds.

3.2.4 User Objectives

The solution is designed specifically for fish farmers who suffer from difficulties in monitoring their fish farms. The main objective is to be able to detect any anomalies in the fish ponds to prevent any problems before the death of fish. In addition, the system will be able to generate reports and statistics to monitor the farms' productivity rates and profits.

3.2.5 General Constraints

The system needs to be as fast as possible to be able to detect any abnormal behavior in the fish pond at the moment it happens, also setting up the cameras at the appropriate position to get the whole fish ponds is difficult process in the fish farms environment. In addition, the continuous availability of large database and high speed internet would be challenging for the system.

3.3 Functional Requirements

ID	**FR1**
Title	Show Pond
Description	When the user opens the system it starts recording the video footage in the fish pond. It takes the object of type camera
Action	Checks if camera is opened. If it's opened the camera will start recording. Else it will open the camera to collects video footage to be processed.
Input	None.
Output	video footage for fish pond.
Precondition	User Starts the system
Post-condition	None.
Dependencies	FR9
Priority	10/10

ID	**FR2**
Title	Enhance Video
Description	The function will enhance the water colors in order to make fish more visible. It takes video frames to apply on it the MSR algorithm to enhance each video frame.
Action	Checks if the frame is not corrupted or null. If not null or corrupted the function will start enhancing the frame.
Input	Pond Video Footage
Output	Color Enhanced video footage
Precondition	Cameras have captured the video footage of the pond
Post-condition	None.
Dependencies	FR1
Priority	10/10

ID	**FR3**
Title	Enhance Image
Description	The function will enhance the water colors in order to make ammonia paper more visible. It takes the image to apply on it the MSR algorithm to be enhanced.
Action	Checks if the image is not corrupted or null. If not null or corrupted the function will start enhancing the image.
Input	Unclear Image.
Output	Color Enhanced Image.
Precondition	Cameras have captured the Image of the ammonia paper
Post-condition	None.
Dependencies	FR1
Priority	10/10

ID	**FR4**
Title	Detect fish
Description	This function detects the fish in the enhanced video. It takes the enhanced video as a parameter and returns an array with the detected objects' coordinates. The detection of fish is done using YOLO algorithm.
Action	It should draw only the bounding box around the detected fish if the confidence value is higher or equal the desired number.
Input	Enhanced Pond Video Footage
Output	Array of coordinates of detected objects.
Precondition	The video footage should be enhanced
Post-condition	None.
Dependencies	FR2
Priority	10/10

ID	**FR5**
Title	Capture Image
Description	This function connects to the web camera and captures an image of the ammonia paper through it. The captured image is captured periodically and should be enhanced through the enhancement function and then sent to the color detection to detect toxic ammonia levels.
Action	It should check if the image size is appropriate to use for enhancing and if the image is not null.
Input	None.
Output	Captured Image.
Precondition	None.
Post-condition	The image should be enhanced.
Dependencies	None.
Priority	10/10

ID	**FR6**
Title	Clustering
Description	This function takes 2D array of features that contains the selected features needed so we can apply clustering algorithms to split data to different clusters according to fish behaviors.
Action	It takes an 2d array of features to cluster behaviors. The array of features should not be null. If null, the function will break and return nothing. Else, it will apply k-medoids to cluster the data.
Input	2D array of features.
Output	String with the Nearest Cluster.
Precondition	Features should be extracted and put into 2D array .
Post-condition	Alert for abnormal behavior.
Dependencies	FR17
Priority	10/10

ID	**FR7**
Title	Detect colors
Description	The function detects different ammonia level in water by checking the area where the test paper is placed and extracting the color from it where each color indicates the ammonia level.
Action	Checks on the four different colors of the ammonia alert paper. If Blue, then the water is toxic. Else, the water is clear from ammonia.
Input	Image of the ammonia paper.
Output	String WaterStatus
Precondition	The image should be enhanced.
Post-condition	None.
Dependencies	FR3
Priority	10/10

ID	**FR8**
Title	Register
Description	Allows user to make a new account. It takes First Name , Last Name , Email and password of the user.
Action	Checks if the user email does not exist in the database. Also, Checks if any of the data entered is not empty or null. If any of them is empty it alerts the user. Else, the account is made.
Input	FirstName , LastName, Email, Password
Output	Boolean true or false
Precondition	None.
Post-condition	A new account is added to the database.
Dependencies	None.
Priority	10/10

ID	**FR9**
Title	Login
Description	User can not use the system without logging in first. It takes the users' Email and password.
Action	Checks if the user email and password exists in the database. If exists, then the user is logged in. Else, the user is alerted by invalid email or password.
Input	Email and Password of the user
Output	Boolean true or false
Precondition	The user must have an account.
Post-condition	Redirect to homepage.
Dependencies	FR8
Priority	10/10

ID	**FR10**
Title	Logout
Description	Logs out the user from the system. It stops the video footage and redirect user to login page.
Action	Checks if the user is currently logged in to make him logout.
Input	None.
Output	None.
Precondition	The user must be logged in.
Post-condition	Redirect to login page.
Dependencies	FR9
Priority	10/10

ID	**FR11**
Title	User CRUD
Description	Fish Farm Owner has the ability to add,edit,delete and show all users in that are available on the system. Takes the new/edited data with all required information and insert/update it into the database. Deleted users are marked as deleted in the database.
Action	Checks if the edited/new data is valid and not empty. Checks if the deleted data exists in the database.
Input	User first name, last name, email and password
Output	Boolean true or false.
Precondition	Fish Farmer must be logged in with an account that have permissions to manipulate users data.
Post-condition	New user or edited data are added to the database. Deleted user(s) are marked as deleted in the database.
Dependencies	FR9
Priority	6/10

ID	**FR12**
Title	Name behavior
Description	User can name an unknown abnormal behavior that is detected by the system. The name will added in the database and will be sent to the fish farm owner to be approved/declined.
Action	Checks for duplicate names in the database. If the name exists, then notify the user with name already exists. Also, checks if the name is null or empty.
Input	String Behavior Name
Output	None.
Precondition	The name of the behavior was unknown
Post-condition	The name is sent for approval
Dependencies	FR1, FR2, FR4,FR6
Priority	5/10

ID	**FR13**
Title	Approve naming
Description	Fish Farm owner should be able to approve/decline on the naming of the unknown abnormal behavior that is detected by the system and named by other users
Action	An appropriate name for that behavior will be added to the newly detected cluster in the dataset if approved. If declined, the naming will be removed from database.
Input	None.
Output	Boolean.
Precondition	The name of the behavior was on stall.
Post-condition	The Name is either approved and assigned to that behavior or declined and deleted from the database.
Dependencies	FR12
Priority	5/10

ID	**FR14**
Title	Alert users
Description	This function first checks if any variables reflecting the behavior or water status are changed during the system execution and notifies all users on the system when any abnormal event happens in the farm
Action	function receives variable for the water toxicity and current fish behavior Check if the water status and current behavior variables are changed to alarming values and then send a notification to the users.
Input	water toxicy level and current fish behavior.
Output	None.
Precondition	None.
Post-condition	Send a notification to the users.
Dependencies	FR1, FR2, FR4, FR6, FR7
Priority	10/10

ID	**FR15**
Title	Rate Alerted Behavior
Description	This function allows the farmer to rate the behavior classification based on its accuracy after checking it in order to get feedback on the system.
Action	The function is called after notifying the user with abnormal behavior. It also, allows him to put comments.
Input	Integer rating from 0 to 5
Output	None.
Precondition	abnormal behavior detected.
Post-condition	New rating is inserted to the database and is used for further improvements in the system.
Dependencies	FR14
Priority	5/10

ID	**FR16**
Title	Solve Problem
Description	This function starts after an abnormal event is detected and outlines steps to help the farmer deal with the problem in the pond.
Action	The function retrieves steps for the solution of the detected abnormal behavior if exists in the database.
Input	None.
Output	String containing solution steps
Precondition	Detection of Abnormal Behavior
Post-condition	None.
Dependencies	FR14
Priority	5/10

ID	**FR17**
Title	Get features
Description	This function retrieves the speed, position, and direction of objects detected during the yolo model execution and adds those features into a 2-dimensional array to be used for clustering the behaviours.
Action	It checks the box coordinates each given number of frames and calculates the speed and direction based on the values extracted.
Input	Two dimensional array of position X,Y, Integer number of frames to be used in each iteration
Output	Two dimensional array containing speed and direction
Precondition	enhanced video footage.
Post-condition	Cluster features.
Dependencies	FR1, FR2, FR4
Priority	10/10

ID	**FR18**
Title	Upload footage
Description	this function uploads recorded footage to the cloud to be enhanced and processed to extract the needed data from them.
Action	It will be taking an array of frames of pond and upload it to the cloud and it will return true or false if its uploaded or not
Input	array of frames
Output	Boolean
Precondition	video footage.
Post-condition	none.
Dependencies	FR1
Priority	10/10

ID	**FR19**
Title	Generate Reports
Description	This function generates reports for all the behaviors and status the fish pond went though to be checked by the farm owners .
Action	It will automatically generate a report for fish behaviors , count , any problem the fish pond and regarding the farms profit and productivity rates.
Input	None.
Output	Reports PDF.
Precondition	None.
Post-condition	None.
Dependencies	None.
Priority	6/10

ID	**FR20**
Title	Compress Frame
Description	This function takes a frame as input from the input video or livestream and applies a compression algorithm to reduce the size of the frame for faster image processing.
Action	It compresses the given frame losslessly and outputs the same frame with a lower size.
Input	An image (frame) file
Output	An image file (lower in size).
Precondition	None.
Post-condition	None.
Dependencies	FR1.
Priority	7/10

ID	**FR21**
Title	Encrypt
Description	This function takes a user password as input and produces an encrypted string to be stored in the database.
Action	It converts a normal string into an encrypted string.
Input	String (password)
Output	String (encrypted password).
Precondition	User should register.
Post-condition	None.
Dependencies	FR8.
Priority	9/10

ID	**FR22**
Title	Decrypt
Description	This function takes an reads all the encrypted passwords in the database and check if any of them matches the user input password after it is encrypted using the same algorithms.
Action	Checks if the input password matches any encrypted record in the database.
Input	String (password)
Output	Boolean (if password exists in the database or not).
Precondition	User Should Login.
Post-condition	None.
Dependencies	FR9
Priority	9/10

ID	**FR23**
Title	View pond profile
Description	It shows the details of the pond by taking such as fish count, current fish behavior and others. It takes the pond ID as a parameter.
Action	It checks if the pond id exists in the database or if it is not equal null.
Input	Integer pond ID
Output	Array of String containing all the details.
Precondition	None.
Post-condition	Shows all the details of the pond.
Dependencies	FR1
Priority	7/10

ID	**FR24**
Title	Add Pond
Description	This function adds new pond in the database. It takes pond name, fish count, date created.
Action	Checks if the pond name doesn't exist in the database. Also, if the entered data is not null or empty.
Input	Pond name, fish count, date created.
Output	Boolean true or false.
Precondition	Fish Farmer must be logged in with an account that have permissions to add pond data.
Post-condition	New pond(s) are added to the database.
Dependencies	None.
Priority	6/10

ID	**FR25**
Title	List Ponds
Description	This function lists all the ponds that a farmer have in his pond even if its not monitored by the system
Action	Checks if the edited/new data is valid and not empty. Checks if the deleted data exists in the database.
Input	Pond Data.
Output	Boolean true or false.
Precondition	Fish Farmer must be logged in with an account that have permissions to manipulate pond data.
Post-condition	New pond or edited data are added to the database. Deleted pond(s) are marked as deleted in the database.
Dependencies	None.
Priority	6/10

3.4 Interface Requirements

3.4.1 User Interfaces

3.4.1.1 GUI

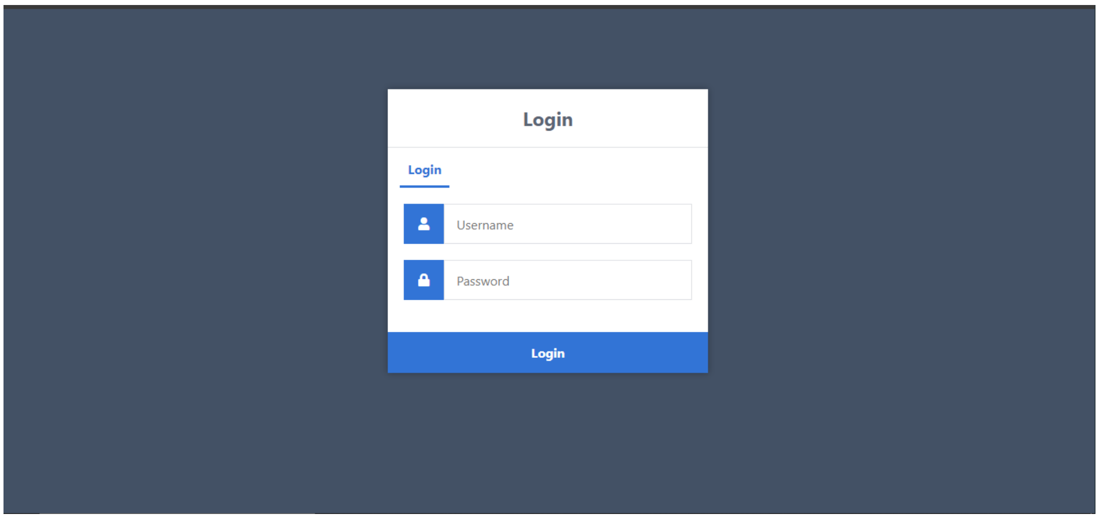


Figure 3.4: Login Screen

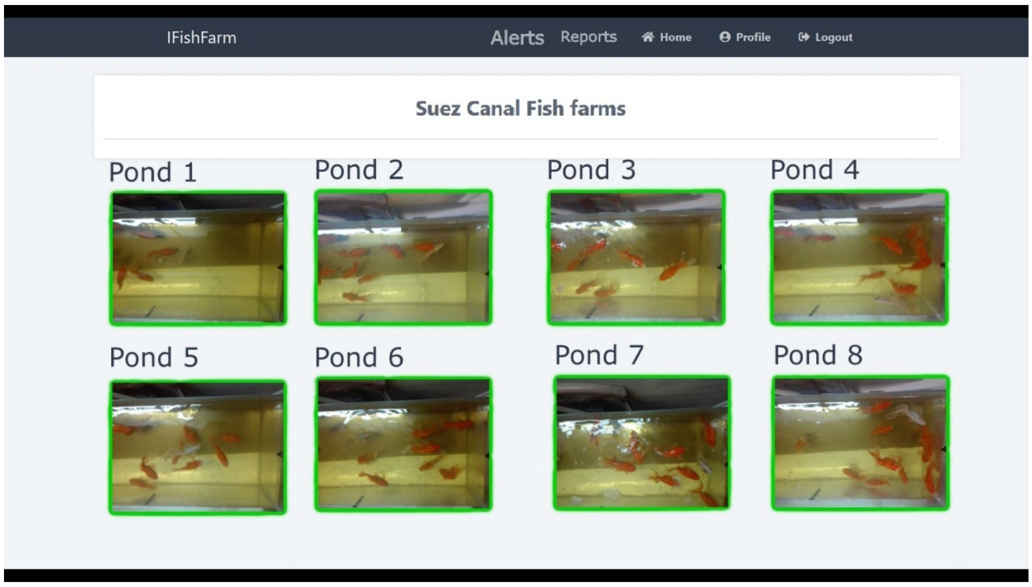


Figure 3.5: home

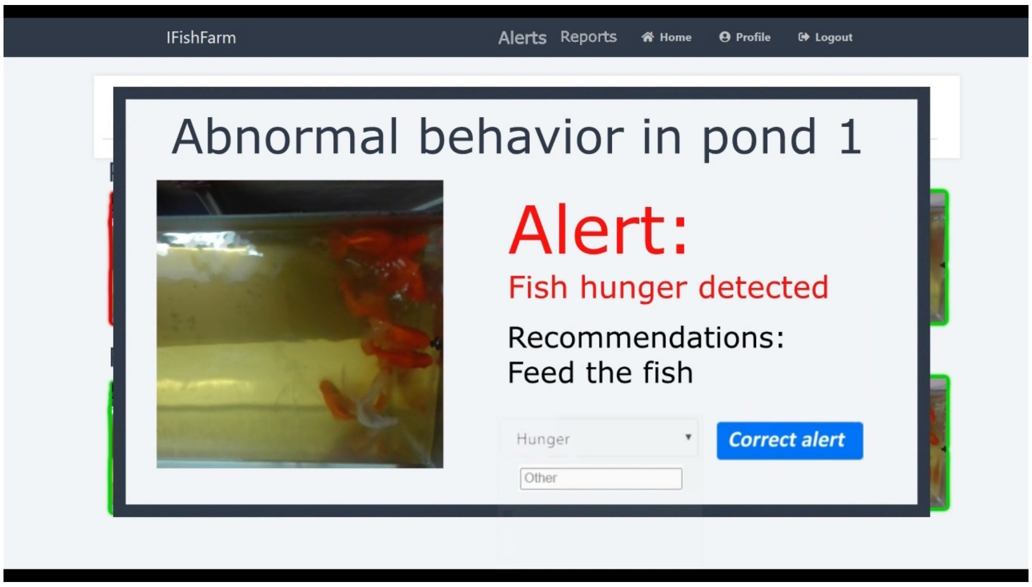


Figure 3.6: Alerts

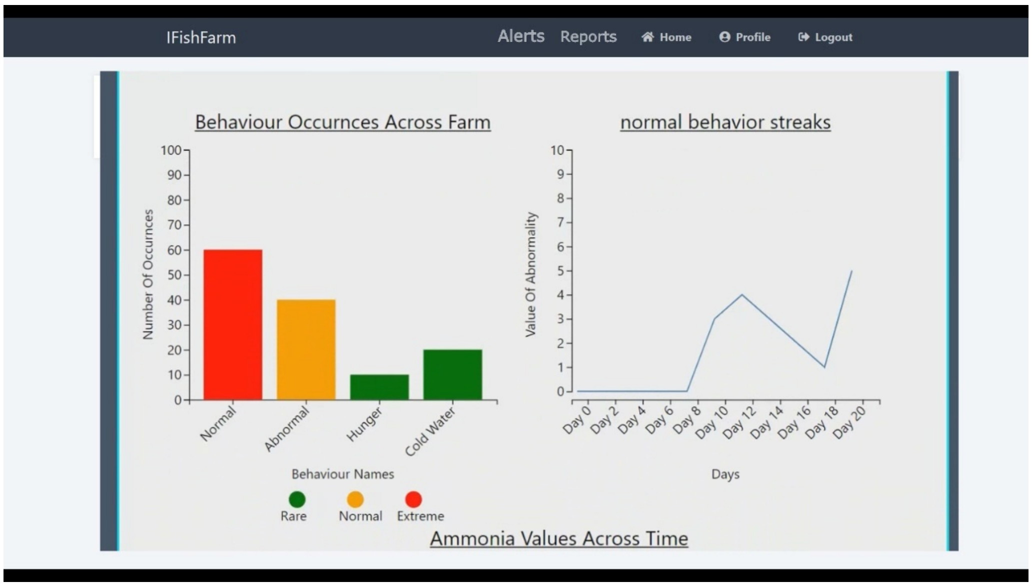


Figure 3.7: Reports 1/2

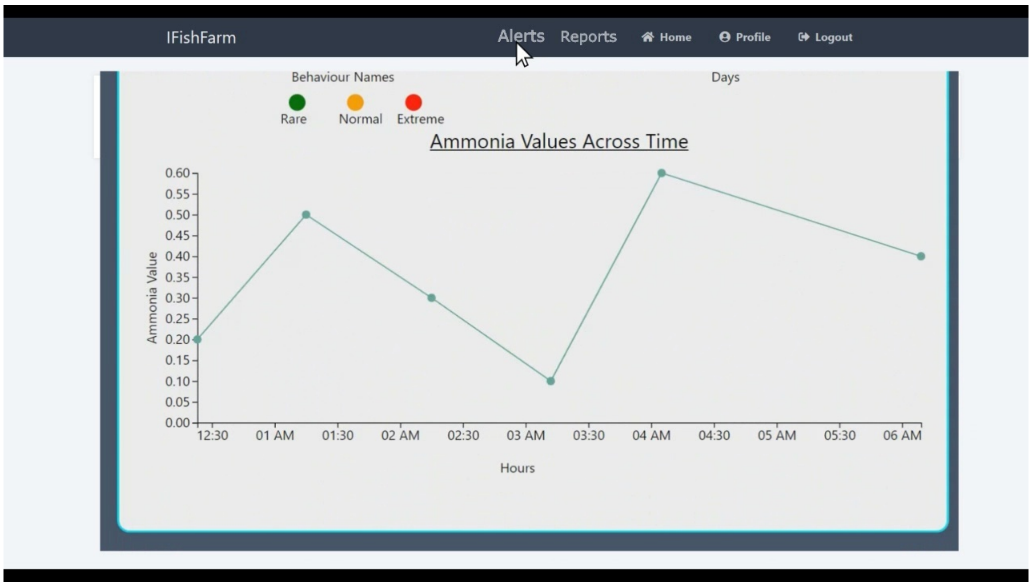
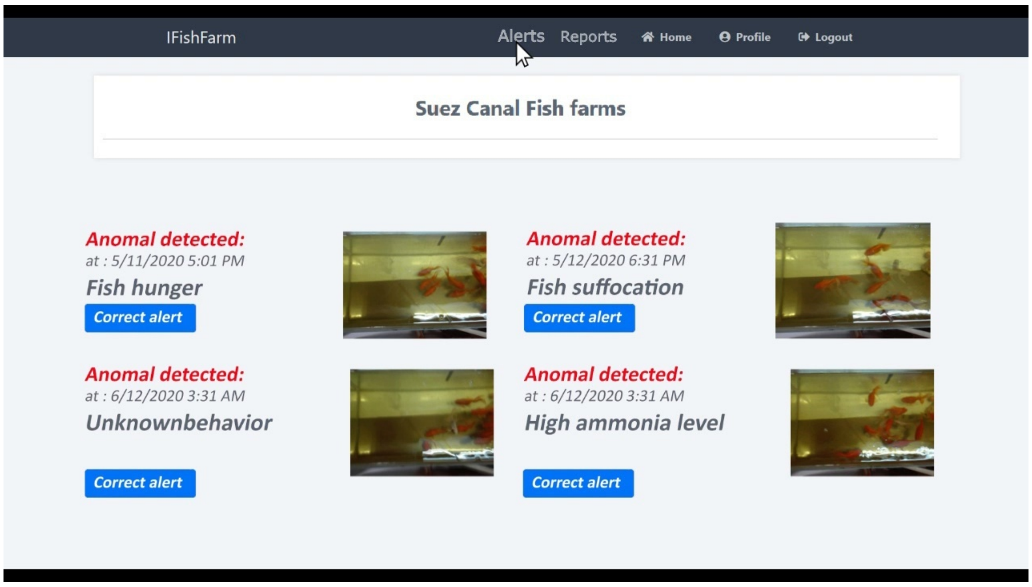


Figure 3.8: Reports 2/2



Suez Canal Fish farms

Anomal detected:
at : 5/11/2020 5:01 PM
Fish hunger
Correct alert

Anomal detected:
at : 5/12/2020 6:31 PM
Fish suffocation
Correct alert

Anomal detected:
at : 6/12/2020 3:31 AM
Unknownbehavior
Correct alert

Anomal detected:
at : 6/12/2020 3:31 AM
High ammonia level
Correct alert

Figure 3.9: old alert

3.4.1.2 API

- openCV
- Flusk

3.5 Performance Requirements

For the IFish Farm, the system that shall be able to process and enhance frames. Also, The system must be able to handle training datasets for model creation in order to ensure model precision.

3.6 Design Constraints

The availability of internet all the time for continuous monitoring.

3.7 Other non-functional attributes

3.7.1 Reliability

- Speed is an important feature in the system as it should provide fish farmers with real-time notifications to notify them on any anomalies in the ponds.
- Accuracy should be nearly 90% in classifying and detecting the type of each anomaly in the fish pond to provide fish farmers with trustworthy feedback and notifications.

3.7.2 Maintainability

- This feature is applied by implementing the Model-View-Controller ”MVC” design pattern and other design patterns which make the system more flexible to be improved or fixed.
- Implementing the Entity-Attribute-Value ”EAV” that allows the developer to add any new requirements dynamically.
- The system can be improved by engaging the user in some actions which can improve the accuracy of the system.

3.7.3 Portability

- This feature is applied by implementing a responsive website that allows any user to use the system on any web browser from any device.

3.7.4 Usability

- Nielsen's heuristics will be applied to ensure a simple and easy interface. A usability study is to be conducted with the client to get feedback and improve the usability.
- The system will be easy to use and learned as fish farmers usually are not familiar with modern technologies.
- The system will be memorized easily as there won't be much tasks for the fish farmer to do.

3.8 Preliminary Object-Oriented Domain Analysis

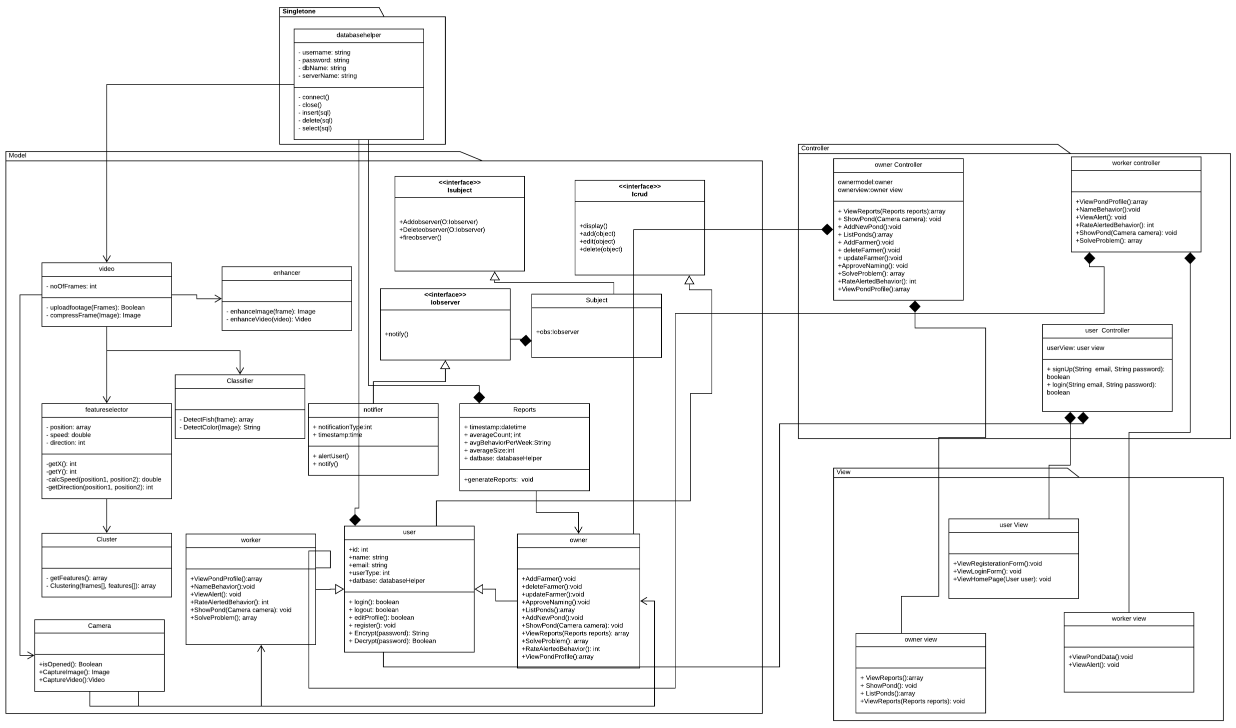


Figure 3.10: Class Diagram

3.9 Operational Scenarios

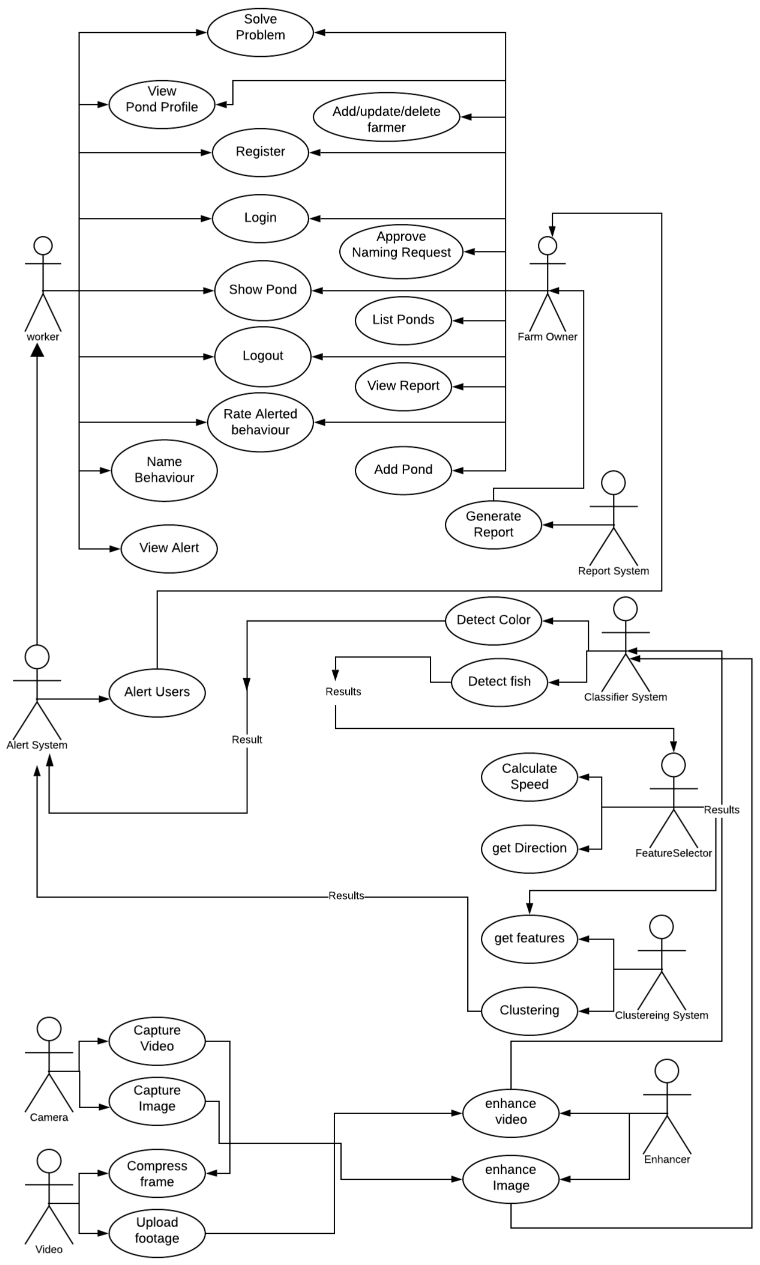


Figure 3.11: Use Case Diagram

Chapter 4

Software Design Document

4.1 Introduction

4.1.1 Purpose

The purpose of this document is to provide a full description of how the iFish Farm monitoring system works. The monitoring system is an online web-based application system to monitor the fish ponds of a farm to ease the maintenance process for farmers to reduce the labor. This software design document (SDD) will describe the aim of the system and its functionalities. In addition, the document will show all constraints on the system, all interfaces' designs and all diagrams that were needed to build the system.

4.1.2 Scope

The purpose of the IFish Farm system is to ease the monitoring process and to create a convenient and easy-to-use web application for farmers to monitor their fish farms. The system is based on an unsupervised learning methodology to cluster the different fish behaviors occurring in the farm while providing necessary alerts to the farmers based on the events detected and suggesting a solution to the problem of the event. We will have a cloud server that will process the captured footage to be done there to speed up the monitoring process.

4.1.3 Overview

This document describes most of the system diagrams and architectures. It also previews how the system main functionalities work and how the user views and interacts with the software. The sections in this document gives a detailed description for the diagrams that help the developer developing the system. It includes the class diagrams, sequence diagrams and the 4 architectures diagrams.

4.1.4 Definitions and Acronyms

Term	Definition
YOLO	You Only Look Once
MSR	Multi-Scale Retinex algorithm.
ROI	Region of Interest.
MVC	Model-View-Controller

Table 4.1: Definition

4.2 System Overview

The IFish Farm system aims to ease the monitoring of fish farms to the fish farmers. The system is divided into several phases. Firstly, the data is collected from a camera above the pond to get the video footage of the fish to detect abnormal behavior and another camera underwater that takes a picture periodically of the ammonia paper to detect toxic ammonia levels. Secondly, the preprocessing phase comes where the video footage and images are enhanced and compressed to get a smaller size and better quality of the video/picture on a cloud server to speed up the process. Thirdly, the classification part where we can detect fish using YOLO algorithm and detect the color of ammonia paper using thershold classifier. After that, features are extracted from the detected fish like there speed, direction, kurtosis, coordinates and many other features. Then, the clustering is applied using K-medoids algorithm to cluster the fish behavior and detect any abnormal behavior. Lastly, the farmer can monitor the farm through the web application and generate reports.

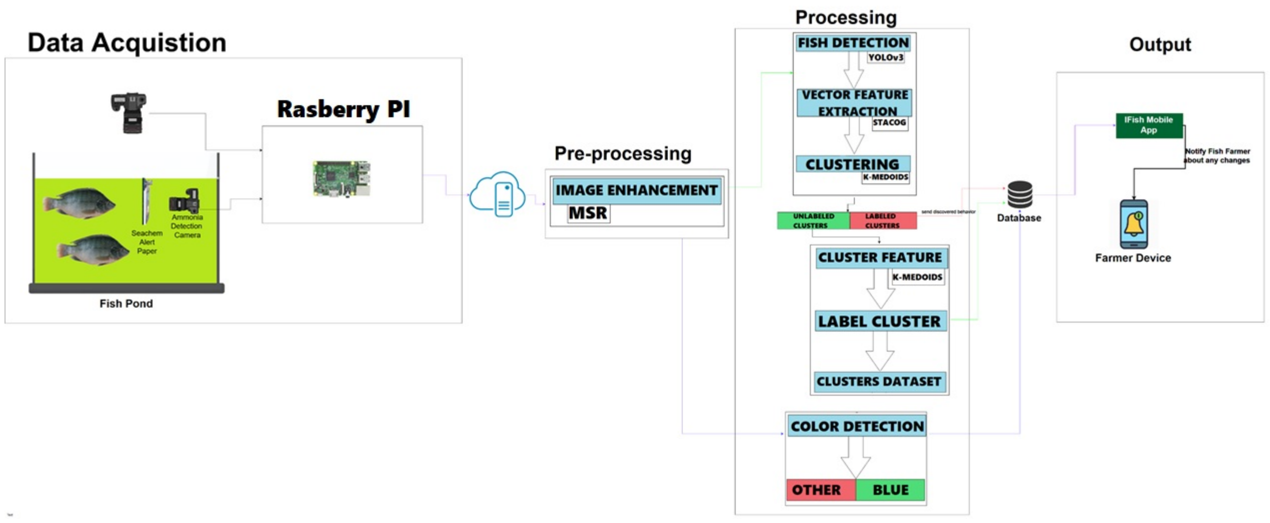


Figure 4.1: System Overview

4.3 System Architecture

4.3.1 Architectural Design

4.3.1.1 Software Diagram

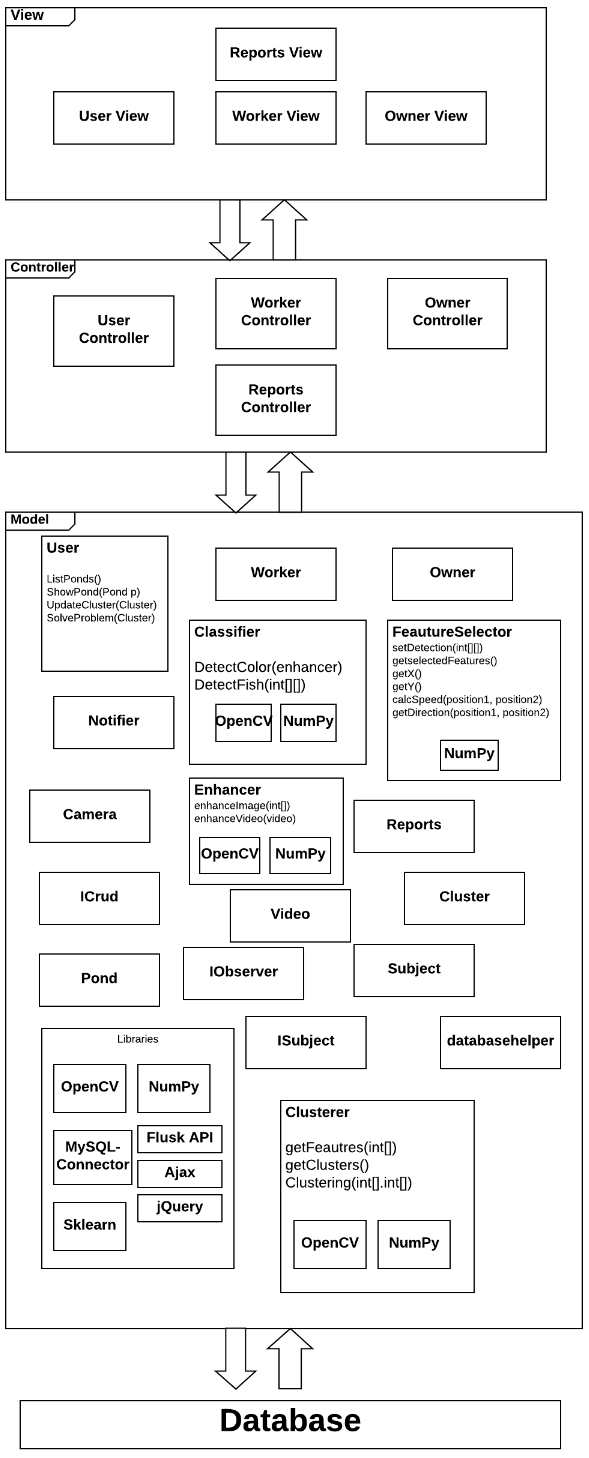


Figure 4.2: Software Diagram

View:
The view is responsible for presenting the data for in the graphical user interface. There are different views in the system such as the owner view , worker view and the reports view. Each one has the responsibility to view different views to the user.

Controller:
The main responsibility for the Controller is playing a role as an intermediate between the View and the model. It takes the data from the model and send this data for the view to display it for the users. For example, User Controller is responsible for handling common data between the owner and the worker. Reports Controller is responsible for the data between reports model to view it in the reports view.

Model:
The main responsibility for the model is dealing with the database to get the data needed for each class. It takes the data from the database and after that it sends the data to the controller. For example, the reports class takes the information required from the database to build the report.

Libraries used:
OpenCV: used in many operations in the system such as image and videos operations.
NumPy: used in many mathematical operations and handling arrays.
Flask API: used to connect the python code with the web application
SkLearn: provides us with many algorithms that is used in our system.
MySQL-Connector: provides the ability to write queries in the python code.
Ajax and JQuery: Used in the web applications as it provides many options regarding the web code.

4.3.1.2 Hardware Diagram

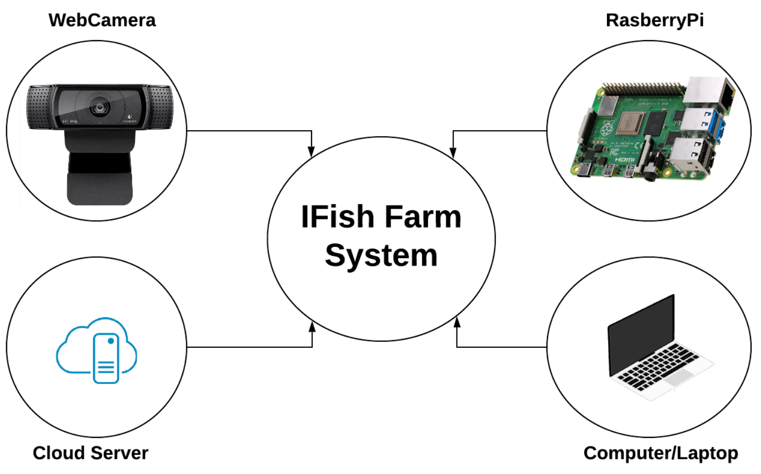


Figure 4.3: Hardware Diagram

This Diagram Shows all hardware components used in our system. They are four components and includes:

Webcamera: Settled above the pond to get the video footage responsible for monitoring the abnormal fish behaviors.

RasberryPi: Takes the video footage and uploads it to cloud server to be processed.

Cloud Server: It is where the data is processed. It is needed to provide a real-time feedback to the farmer.

Laptop/Computer: The owner of the fish farm might need a computer device to monitor the system.

4.3.2 Block Diagram

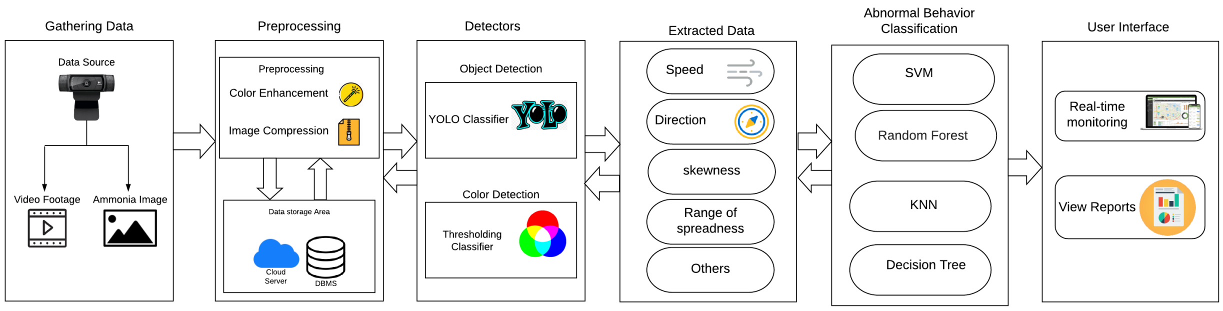


Figure 4.4: Block Diagram

The Diagram shows the system blocks. The system is divided into six different blocks as follows:

Gathering Data: This block is responsible for data acquisition where a camera above the pond gets the video footage for fish behavior and another camera takes a picture of the ammonia paper periodically to detect toxic ammonia.

Preprocessing: This block is responsible for editing the data to be better classified into the next phase. Firstly, the images and videos are enhanced to get better quality of unclear water. Secondly, the files are compressed to get a smaller size of the images and videos to speed up the process.

Classification: This block is responsible for taking the preprocessed data to classify it. It takes the video footage and detect the fish using YOLO algorithm. Also, It takes the ammonia image and use a color detection algorithm to detect ammonia levels.

Extracted Data: This block is mainly made for extracting the data from videos after fish detection. It takes the video and extract data like speed, direction, kurtosis, coordinated,.. etc. which then helps in detecting abnormal fish behavior.

Clustering: This block is responsible for clustering the fish behaviors into different clusters. It is mainly responsible for detecting any abnormal behavior in fish ponds.

User Interface: This block is responsible for viewing the fish pond in live stream and makes the fish farm owner be able to generate reports.

4.3.3 Process Diagram

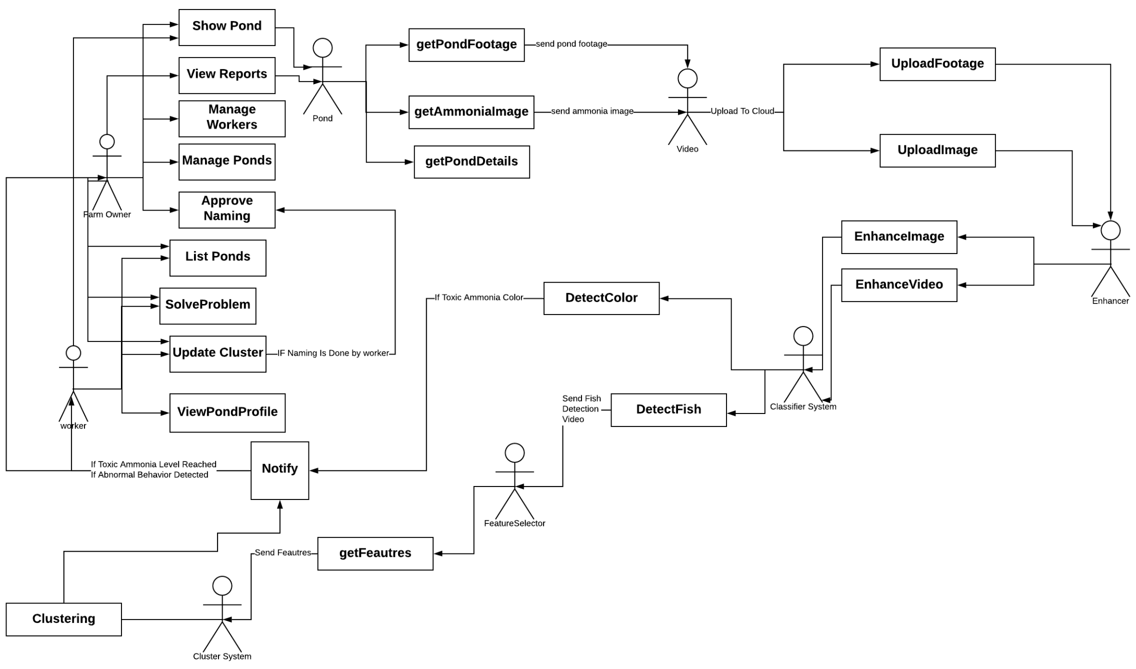


Figure 4.5: Process Diagram

This diagram shows the process flow of the system. The user shows the pond where it goes to the pond class to get the video footage and ammonia image. After that, the image and video are uploaded to the cloud server. In the cloud server the image/video are enhanced then the classifier detects the color in the ammonia image and detects the fish in the video footage. The features are then extracted by the featureSelector class after that it goes to the clusterer class to detect abnormal behavior. The system then notifies about any anomalies in the pond like the fish abnormal behavior and the toxic ammonia. The user has some other actions like listing all ponds, problem solutions, updating cluster names and other actions as shown in the diagram.

4.3.4 Context Diagram

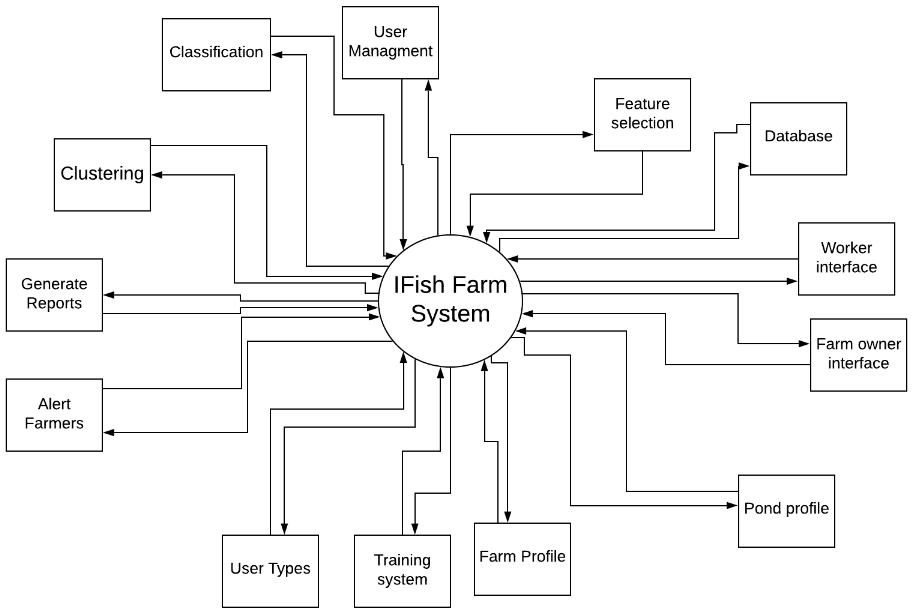


Figure 4.6: Context Diagram

This Diagram shows the relation between the systems' entities and the actual IFish Farm System. The systems' entites in our system is about 13 entity they include training system, classification, feature selection, database and many others.

4.3.5 Decomposition Description

4.3.5.1 Class Diagram

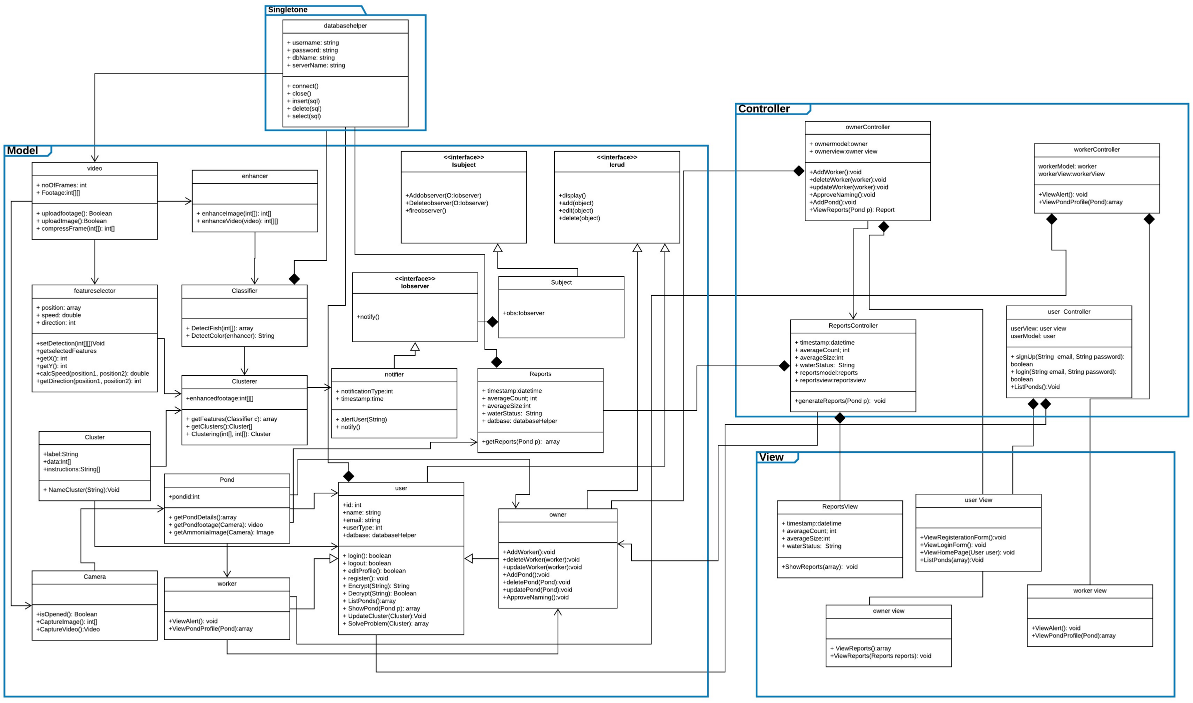


Figure 4.7: Class Diagram

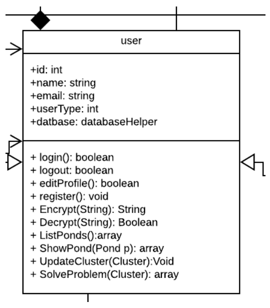


Figure 4.8: User Class

Class name: user

List of super classes: None.

List of sub classes: Owner, Worker.

Purpose: Class to encapsulate different user-types with their common attributes.

Collaboration:

- Aggregates class databaseHelper and UserController.

- Extended by Owner and Worker.

Attributes: Id, name, email, userType, object from databasehelper

Operations: login(): boolean

logout: boolean

editProfile(): boolean

register(): void

Encrypt(String): String

Decrypt(String): Boolean

ListPonds():array

ShowPond(Pond p): array

UpdateCluster(Cluster):Void

SolveProblem(Cluster): array

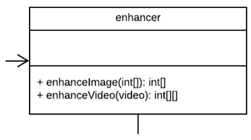


Figure 4.9: Enhancer Class

Class name: enhancer

List of super classes: None.

List of sub classes: None.

Purpose: Class to get images and videos to enhance their colors for better classification.

Collaboration:

- Assisted by video class.

- Assists Classifier Class

. Attributes: None.

Operations: enhanceImage(int[]): int[]

enhanceVideo(video): int[][]

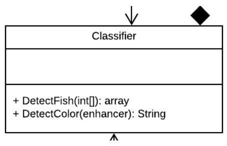


Figure 4.10: Classifier Class

Class name: classifier

List of super classes: None.

List of sub classes: None.

Purpose: Needed to classify objects (fish) and detect ammonia colors for toxic ammonia levels.

Collaboration:

- Aggeregates class databaseHelper.

- Assisted by enhancer class.

- Assists clusterer class.

Attributes: None.

Operations: DetectFish(int[]): array

DetectColor(enhancer): String

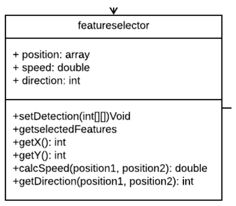


Figure 4.11: Feauture Selector Class

Class name: feautureselector

List of super classes: None.

List of sub classes: None.

Purpose: Class to extract features from the videos to get the needed features.

Collaboration:

- Assisted by video class.
- Assists clusterer class.

Attributes: position, speed, direction

Operations: setDetection(int[][]): void

getselectedFeatures(): void

getX(): int

getY(): int

calcSpeed(position1, position2): double

getDirection(position1, position2): int

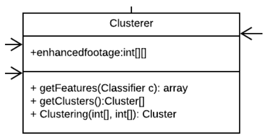

Clusterer
+enhancedfootage:int[][]
+ getFeatures(Classifier c): array
+ getClusters():Cluster[]
+ Clustering(int[], int[]): Cluster

Figure 4.12: Clusterer Class

Class name: clusterer

List of super classes: None.

List of sub classes: None.

Purpose: Class to detect different behaviors of fish in the system by clustering them into different clusters.

- Assisted by feautureselector , cluster and classifier class.

- Assists notifier class.

Attributes: enhancedfootage:int[][]

Operations: getFeatures(Classifier c): array

getClusters():Cluster[]

Clustering(int[], int[]): Cluster

4.3.5.2 Activity Diagram

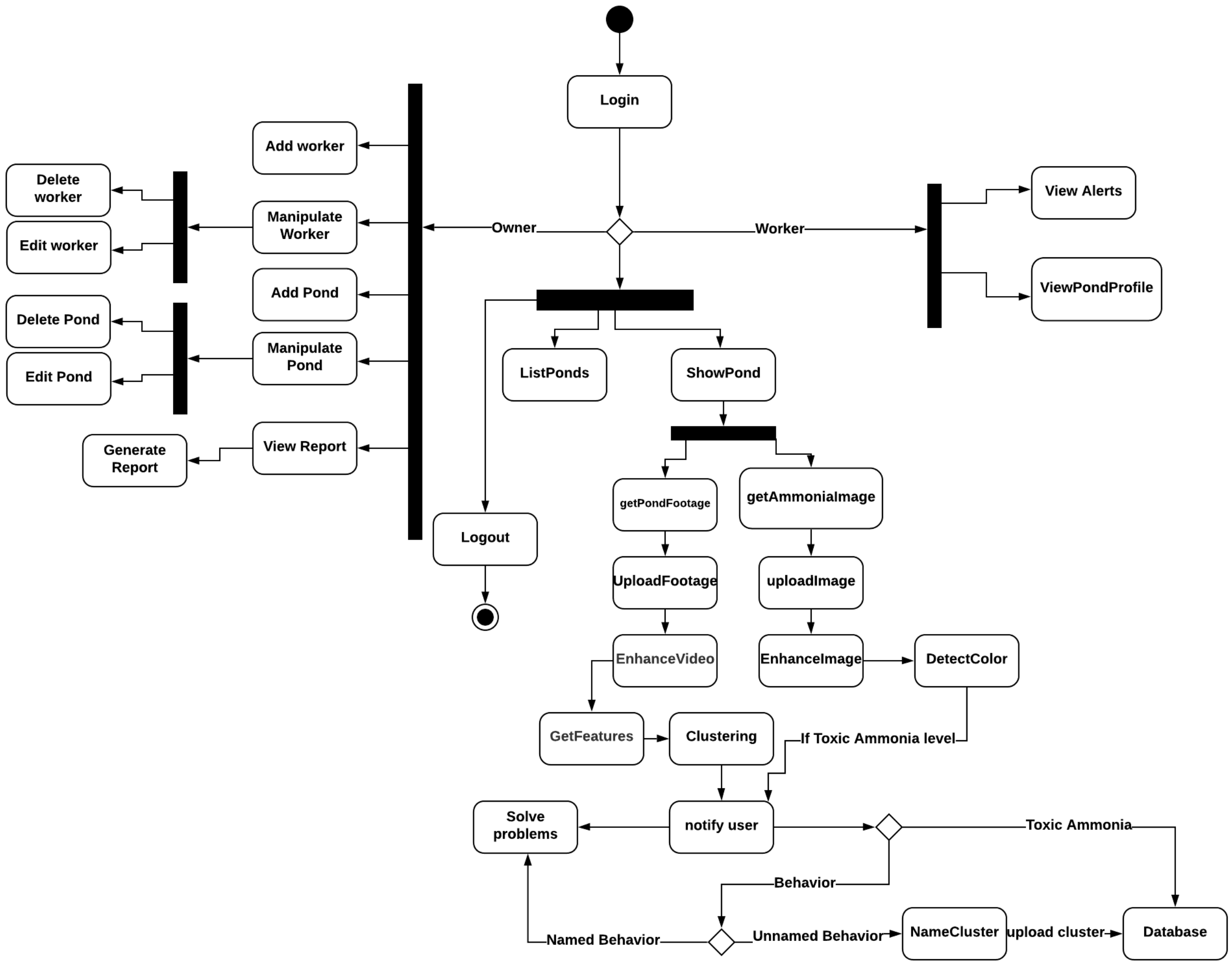


Figure 4.13: Activity Diagram

4.3.6 State Diagram

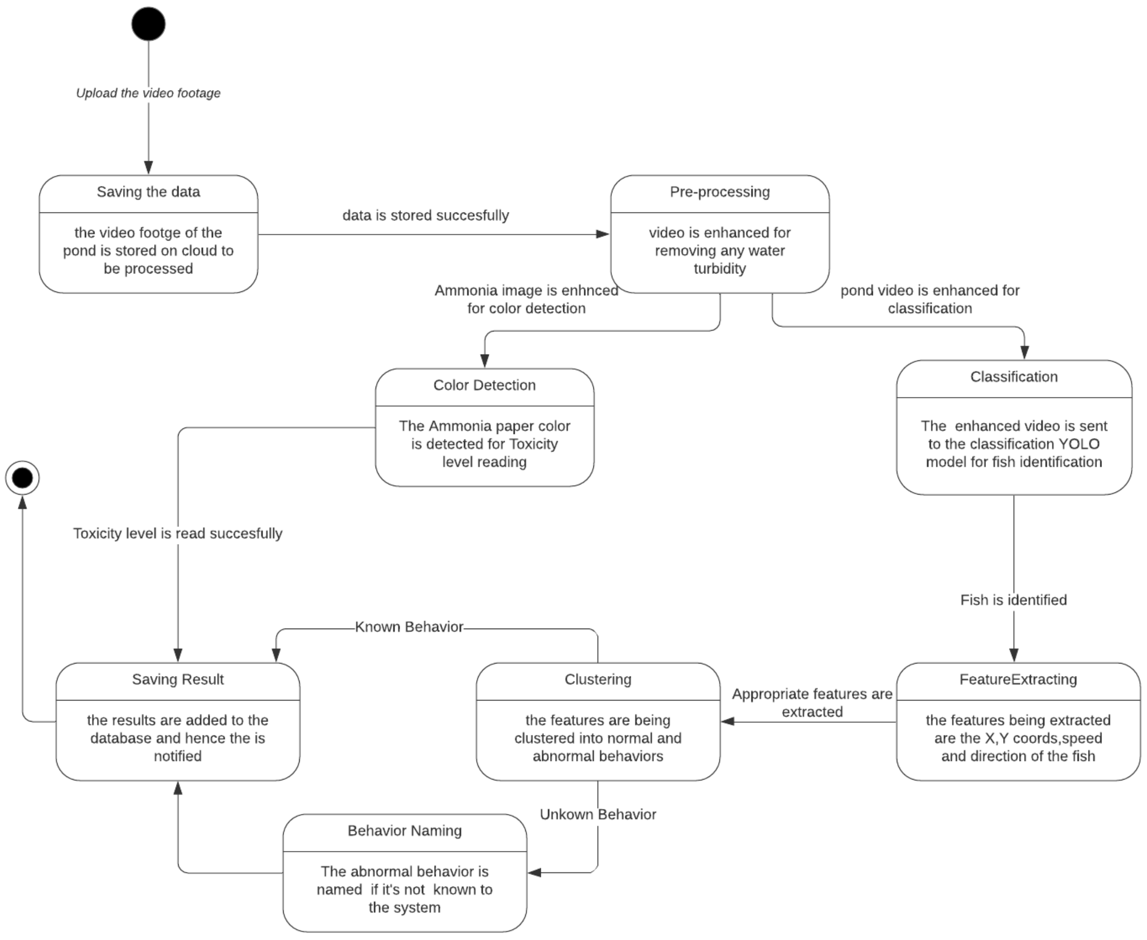

Upload the video footage
Saving the data
the video footge of the pond is stored on cloud to be processed
data is stored succesfully
Pre-processing
video is enhanced for removing any water turbidity
Ammonia image is enhnced for color detection
pond video is enhanced for classification
Color Detection
The Ammonia paper color is detected for Toxicity level reading
Classification
The enhanced video is sent to the classification YOLO model for fish identification
Toxicity level is read succesfully
Fish is identified
Known Behavior
Saving Result
the results are added to the database and hence the is notified
Clustering
the features are being clustered into normal and abnormal behaviors
Appropriate features are extracted
FeatureExtracting
the features being extracted are the X,Y coords,speed and direction of the fish
Unkown Behavior
Behavior Naming
The abnormal behavior is named if it's not known to the system

Figure 4.14: State Diagram

4.3.6.1 Sequence Diagrams

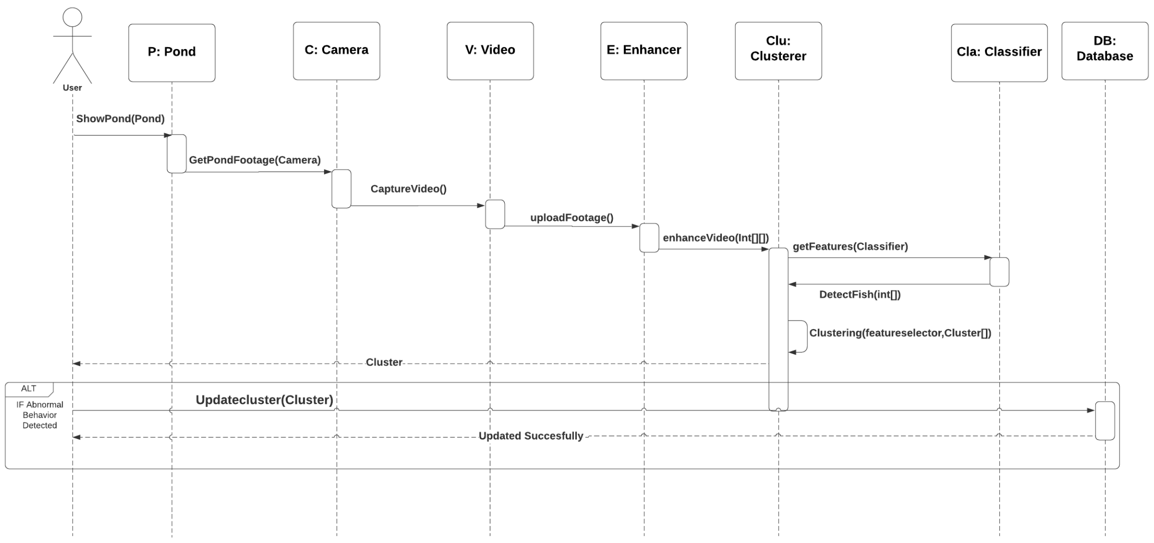


Figure 4.15: Clustering Sequence Diagram

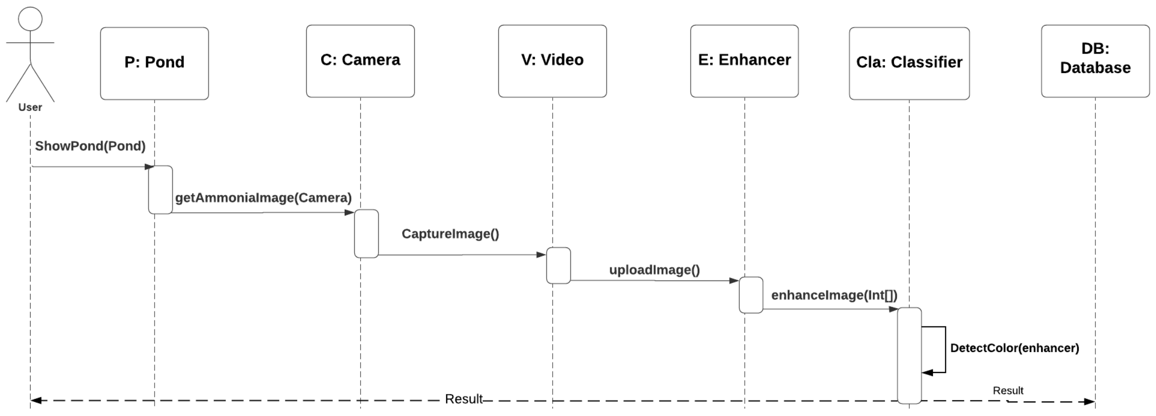


Figure 4.16: Ammonia Detection Sequence Diagram

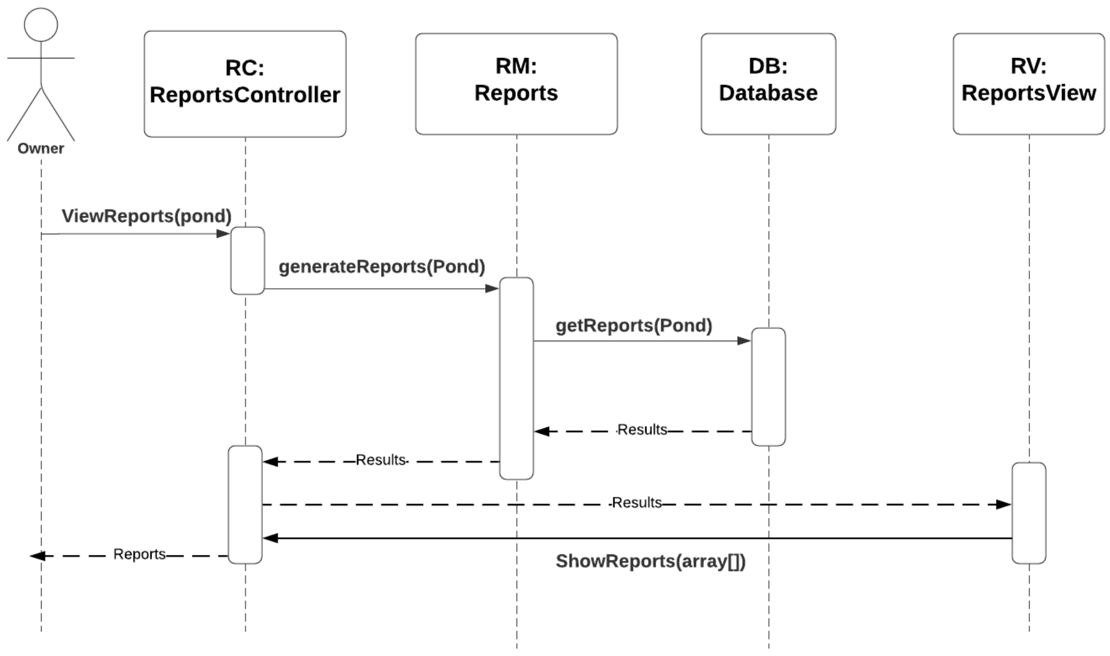


Figure 4.17: Reports Sequence Diagram

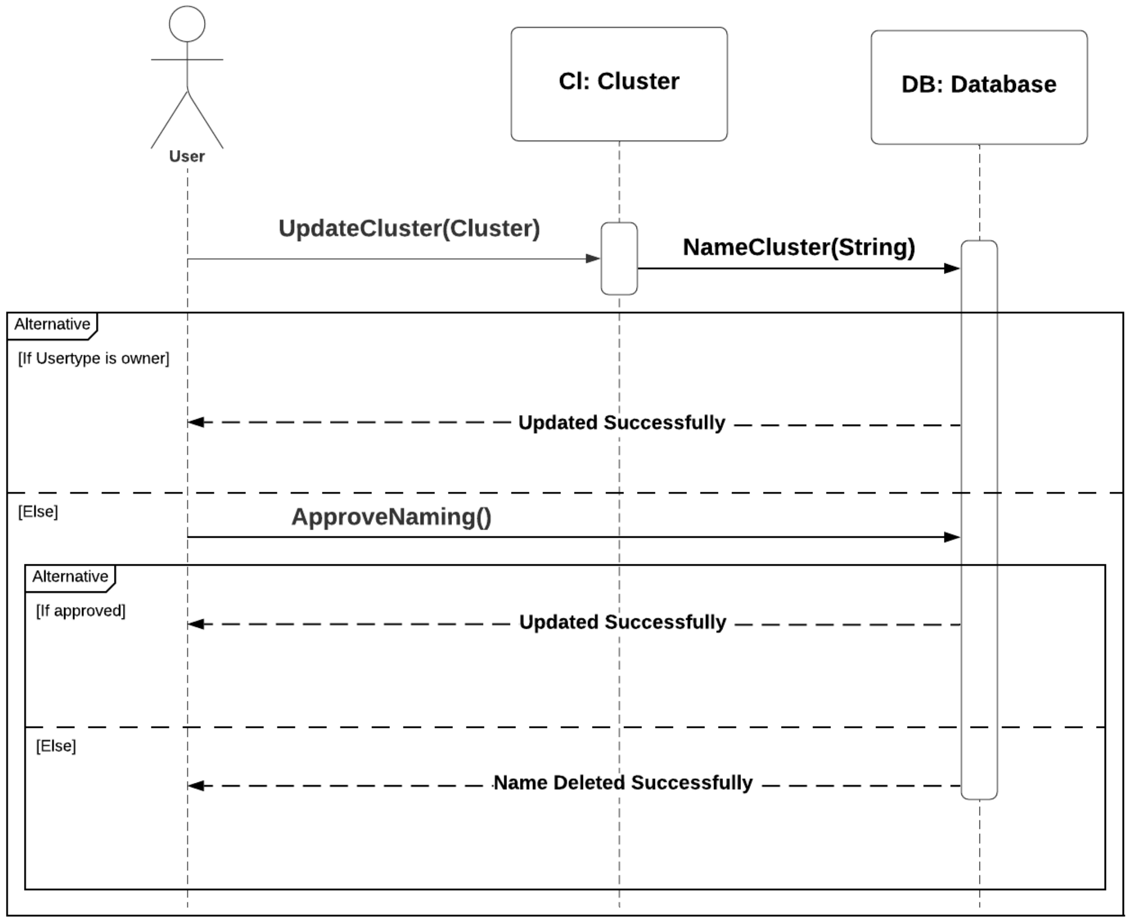


Figure 4.18: Behavior Renaming Diagram

4.3.7 Design Rationale

We use the Model-View-Controller (MVC) architecture as mentioned before. The main reason we used this architecture is that it provides us with flexibility to change in the code with less complexity. Also, the system is developed for fis farms that need accurate and reliable results. We had to choose between different algorithms that are applied in different phases in the system (image enhancement, classification, clustering).

Image Enhancement Algorithms:

- ACE: This algorithm is based on an approach that merges the gray World and white patch mechanisms, while taking into account the spatial distribution of color information.
- MSRCR: The algorithm is based on retinex theory which is greying out the image, either the whole image or in specific regions while applying color restoration mechanism to avoid desaturation of the image.
- MSRCP: This algorithm is also based on retinex but with slight difference in the color preservation techniques which fixes some parameters that needed to be put randomly.

- We chose the MSRCR algorithm to enhance our images after testing with some images with the other algorithms. The MSRCR algorithm was the best enhancement algorithm among the others that gives the better results regarding the enhancement quality of the picture.

Classification Algorithms:

- R-CNN: The main idea of this algorithm is using selective search, it identifies a number of bounding-box objects (region of interest) and then it extracts CNN features from each region independently for classification.
- Fast R-CNN: It's a better version of R-CNN algorithm. The difference is instead of extracting CNN feature independently it combines them into one CNN over the entire image which make it faster than R-CNN.
- YOLO: This algorithm works by applying a single neural network to the image as a whole. Then, the network divides the image into regions and predicts the probabilities for each region.

- The YOLO algorithm was chosen to detect objects (fish) in our system as it is better and faster than competitors like R-CNN or Fast R-CNN as it uses only uses one neural network for its predictions, unlike R-CNN which needs thousands of neural networks for a single image.

Clustering Algorithms:

- Hierarchical clustering: The algorithm starts by treating each observation as a separate cluster. Then, it continuously executes the following two steps: identify the two clusters that are closest together, and merge the two most similar clusters. This continues until all the clusters are merged together.
- K-means: The algorithm is based on calculating the mean of the data to get the center of a cluster.
- K-medoids: The algorithm is based on sorting the data and taking the middle one as the center. It take each data and compute its distance with the other. the we select the diamond with the minimum distance.

- We chose the K-medoids algorithm as it is more robust algorithm than others. Also, it suits our situation in detecting different behaviors of fish as in unsupervised learning.

4.4 Data Design

4.4.1 Data Description

Our Data will be stored in a database using MySQL phpmyadmin.
Some of the database tables are explained as follows :
1.User : Contains all user data required to make the user access the system.
2.Alert : Contains all information about alerts to the farmers.
3.Behavior : Contains all fish behaviors that are detected by the system including their name.

Database Scheme:

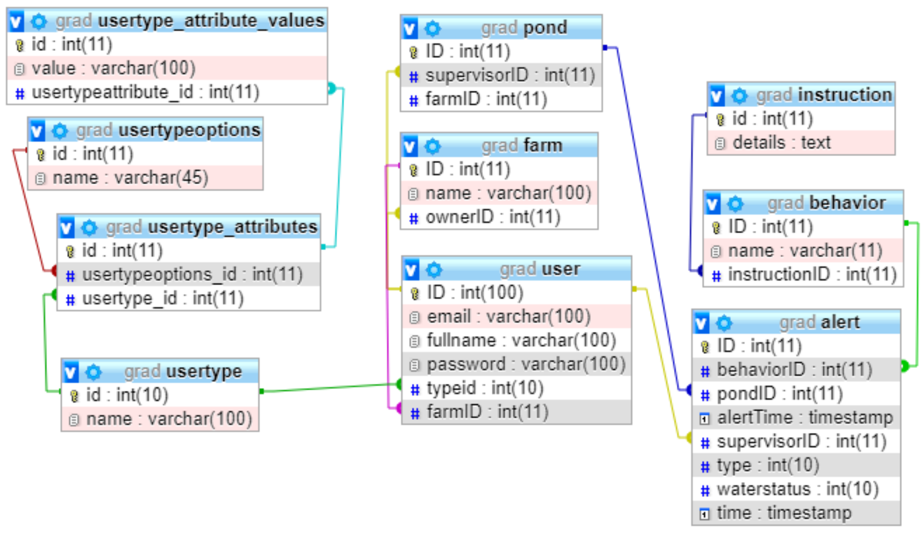


Figure 4.19: Database scheme

4.4.2 Data Dictionary

Table	Column	Type
User	id	int(100)
	email	varchar(100)
	fullname	varchar(100)
	password	varchar(100)
	typeid	int(10)
	farmID	int(11)
Alert	id	int(11)
	behaviorID	int(11)
	pondID	int(11)
	alertTime	timestamp
	supervisorID	int(11)
	type	int(10)
	waterstatus	int(10)
	time	timestamp
Behavior	id	int(11)
	name	varchar(11)
	instructionID	int(11)
Instruction	id	int(11)
	details	text
pond	id	int(11)
	supervisorID	int(11)
	farmID	int(11)
farm	id	int(11)
	name	varchar(100)
	ownerID	int(11)
usertype	id	int(10)
	name	varchar(100)
usertype_attributes	id	int(11)
	userTypeOptions_ID	int(11)
	userType_ID	int(11)
usertype_attribute_values	id	int(11)
	value	varchar(100)
	userTypeAttribute_ID	int(11)
usertypeoptions	id	int(11)
	name	varchar(45)

Table 4.2: Database tables

4.5 Component Design

In this section we describe the blocks of the system in detail. The blocks of the system are data acquisiton, preprocessing, classification and clustering.

4.5.1 Data Acquisition

This is the first phase were we collect our data to be further preprocessed, classified and clustered. The data is collected from a camera placed above the pond which is responsible for visual surveillance of the behaviors of the fish and another camera which is placed underwater and responsible for watching the ammonia paper for any changes in the toxicity levels.

4.5.2 Preprocessing

In this phase preprocessing takes place where we prepare the images and videos to be taken to the next phase which is classification to better classify and detect fish. Specifically, in this phase we use our chosen image enhancement algorithm to enhance videos and images to better detect fish. Image enhancement is important for our system as it provides better visualization for the turbid water images. As shown in the figure below it shows the difference between before enhancement (left image) and after enhancement (right image).

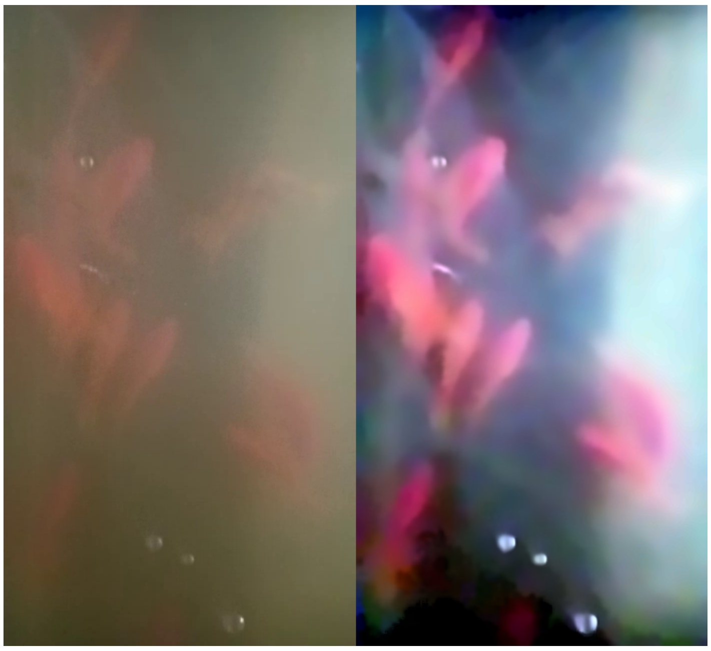


Figure 4.20: Left: Before Enhancement , Right: After Enhancement

4.5.3 Classification

This phase is essential for the system as it acts as an intermediate between preprocessing and clustering phases. The classification phase is divided into two parts. Firstly, the part were we use an object detection algorithm to detect fish which are then used to extract features. Secondly, the part were we use threshold color detection algorithm to detect different colors of ammonia paper to alert farmers if ammonia reaches toxic level.

4.5.3.1 Object Detection

In this section, we describe the object detection algorithm used to detect fish in fish farms. Object detection is important for the system as it detects fish where we can extract features from it and then do the clustering part. For this, we used YOLO to detect objects (fish) which has an acceptable real-time accuracy. The algorithm is one of the regression-based object detection algorithms where it estimates the classes and region of interests for the image in a single run for the algorithm. After detecting fish, we use the bounding box (region of interest) in each single frame of the video to extract from it some features like the coordinates, speed, direction,.. etc. As shown in the figure below an example of fish detection by YOLO.

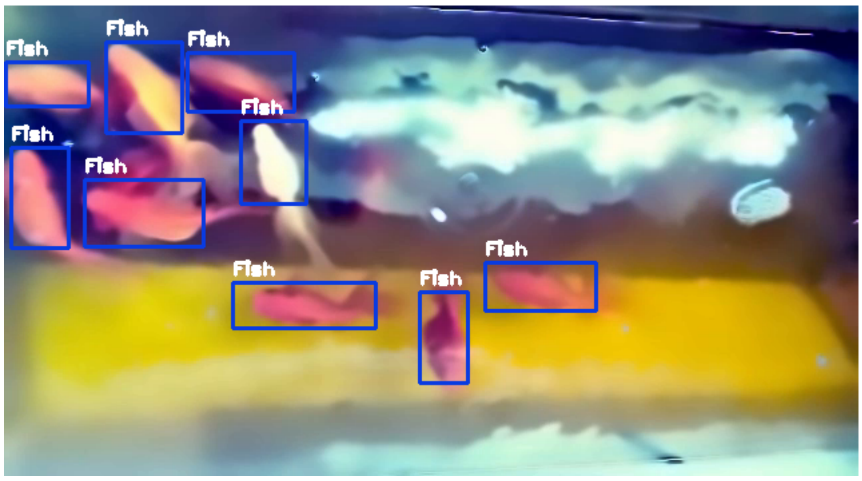


Figure 4.21: Fish Detection

4.5.3.2 Color Detection

In this section, we apply a threshold color detection algorithm to detect different stages of ammonia in the water. Ammonia detection is essential for fish farms as when it reaches toxic levels it leads to fish death. The algorithm works as follows. Firstly, it takes upper and lower BGR pixels to define which region of colors it should take and which it can discard. After that, giving the upper, lower and the image to a function that proceeds to perform a binary mask on the image where the white pixels are the detected region and black pixels are the discarded ones. Finally, the binary image is passed to another function that takes the original input image and the binary image to retain the colors of the detected object. As shown in the figure below this is an example of detecting the color of toxicity in the ammonia paper.

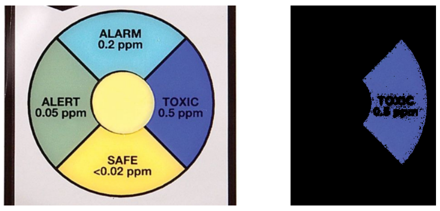


Figure 4.22: Left: Orginal Image , Right: Detected Region

4.5.4 Clustering

In this section, the clustering algorithm we used will be explained in detail. The clustering phase is the main phase in the system as it divides the fish behaviors into different clusters so the fish farmer can be alerted when any abnormal behavior occurs. K-Medoids was chosen to be the algorithm used in our system.

4.5.4.1 K-Medoids Algorithm

K-medoids is a clustering algorithm that partitions the dataset into groups and chooses data points as medoids (centers) for each cluster as its most centrally located point with the least dissimilarity to other objects in the cluster. It can be used in supervised and unsupervised learning techniques.

4.5.5 Advantages of K-Medoids

Compared to K-means, it is more flexible and robust when it comes to noise and outliers because it minimizes the absolute distance between the points and the selected centroid, as opposed to minimizing the square distance in k-means. A medoid has to belong to the set (cluster), while a centroid doesn't. Also, it is fast and executes in fixed number of steps. This makes K-medoids a suitable clustering algorithm to accurately cluster the behaviours in our system.

4.5.6 How K-Medoids Works

The algorithm works in sequence of known steps as follows:

1. Select K random points out of N data points as the medoids.
2. Assign each data point to its closest medoid.
3. Check each cluster for any point that decreases the dissimilarity coefficient, if it does, select the point that decreases it the most as the new medoid for this cluster then repeat step 2.

The following figure shows a flowchart of how the algorithm works.

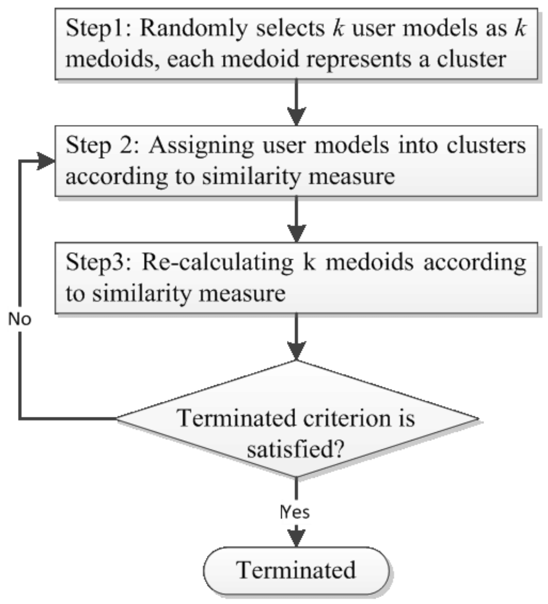

Step1: Randomly selects k user models as k medoids, each medoid represents a cluster
Step 2: Assigning user models into clusters according to similarity measure
Step3: Re-calculating k medoids according to similarity measure
No
Terminated criterion is satisfied?
Yes
Terminated

Figure 4.23: Flow Chart of K-medoids algorithm

4.6 Humnan Interface Design

4.6.1 Overview of User Interface

The IFish Farm user interface (UI) is made to be easy to use as the users will not be familiar with using technologies. The user will be able to login as worker or owner depending on his type. Each user type has different tasks to be done thus different screens are shown to each user. The upcoming sections show the screens in detail.

4.6.2 Screen Images

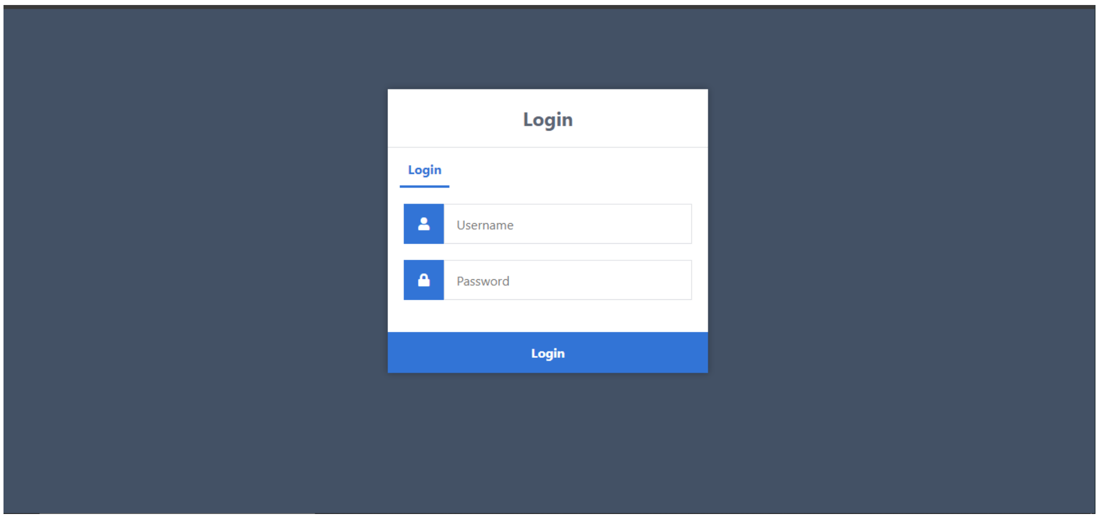


Figure 4.24: Login Screen

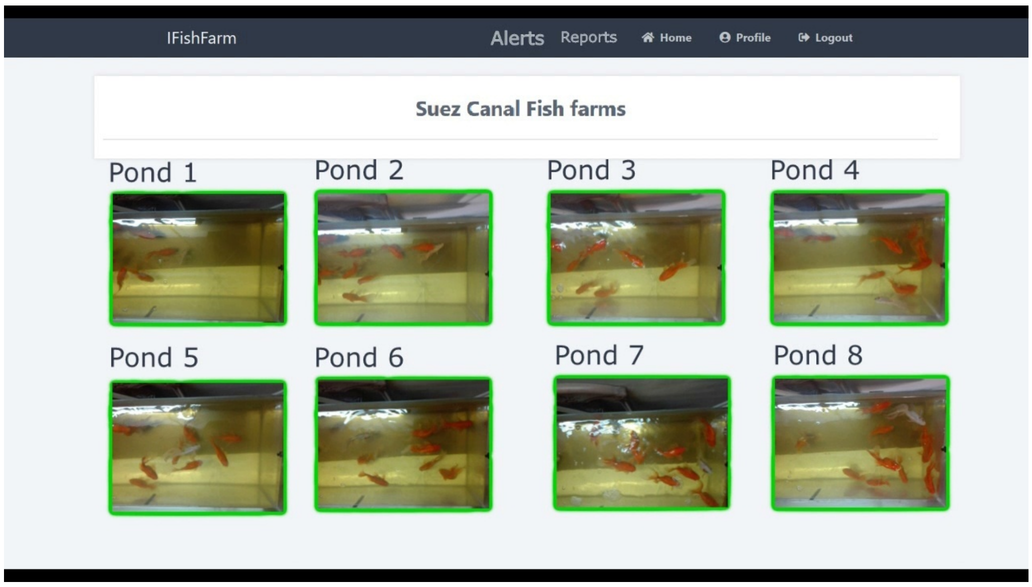
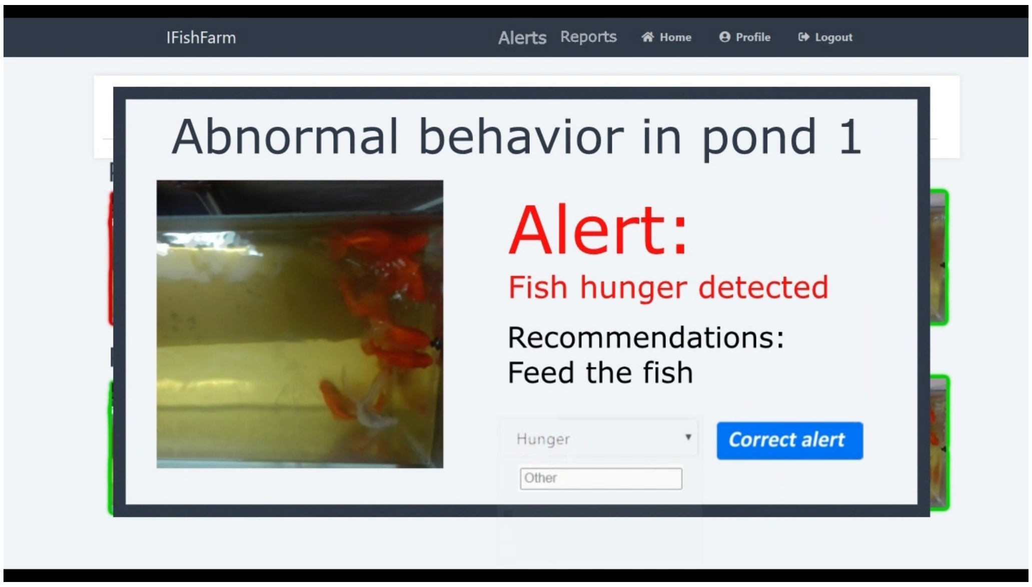


Figure 4.25: home



Abnormal behavior in pond 1

Alert:

Fish hunger detected

Recommendations:
Feed the fish

Figure 4.26: Alerts

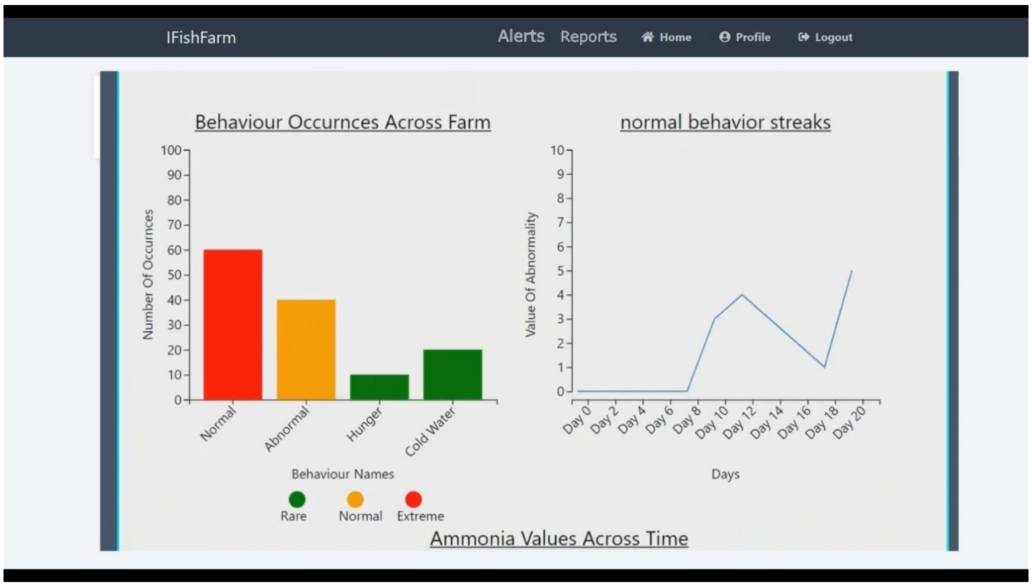


Figure 4.27: Reports 1/2

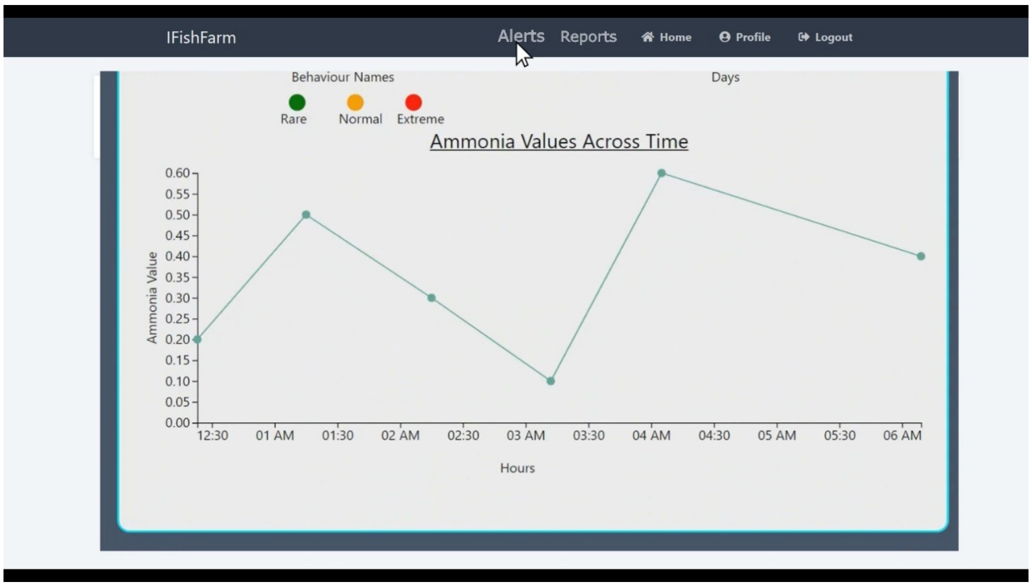
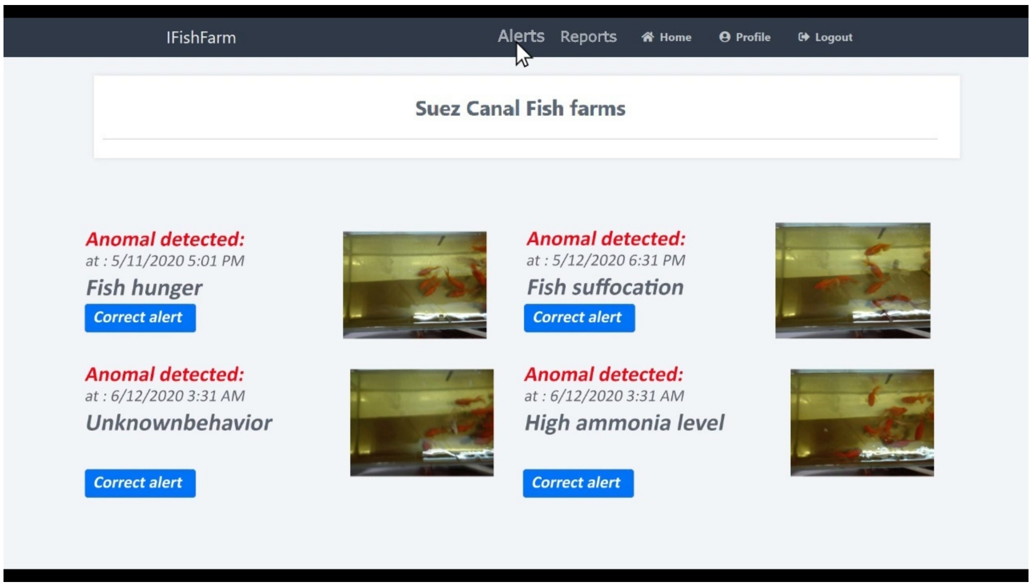


Figure 4.28: Reports 2/2



Suez Canal Fish farms

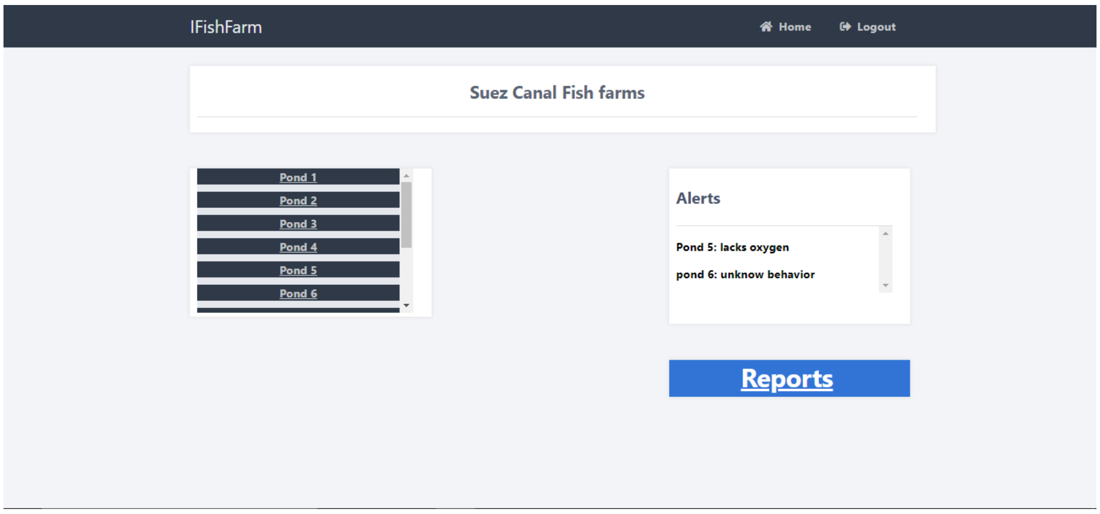
Anomal detected:
at : 5/11/2020 5:01 PM
Fish hunger

Anomal detected:
at : 5/12/2020 6:31 PM
Fish suffocation

Anomal detected:
at : 6/12/2020 3:31 AM
Unknownbehavior

Anomal detected:
at : 6/12/2020 3:31 AM
High ammonia level

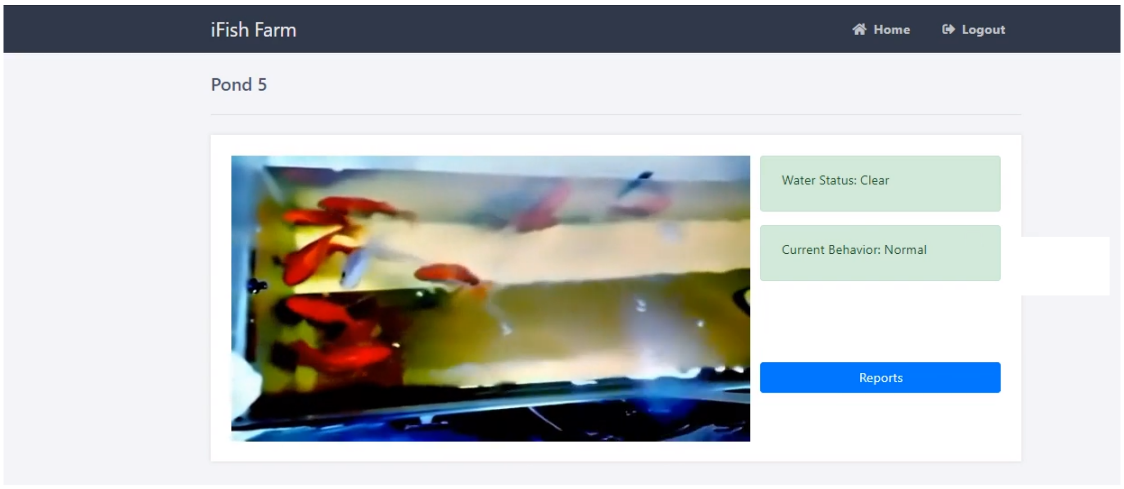
Figure 4.29: old alert



Pond 5: lacks oxygen

pond 6: unknow behavior

Figure 4.30: Main Screen



Water Status: Clear

Current Behavior: Normal

Figure 4.31: Pond Footage Screen Without any alerts happening

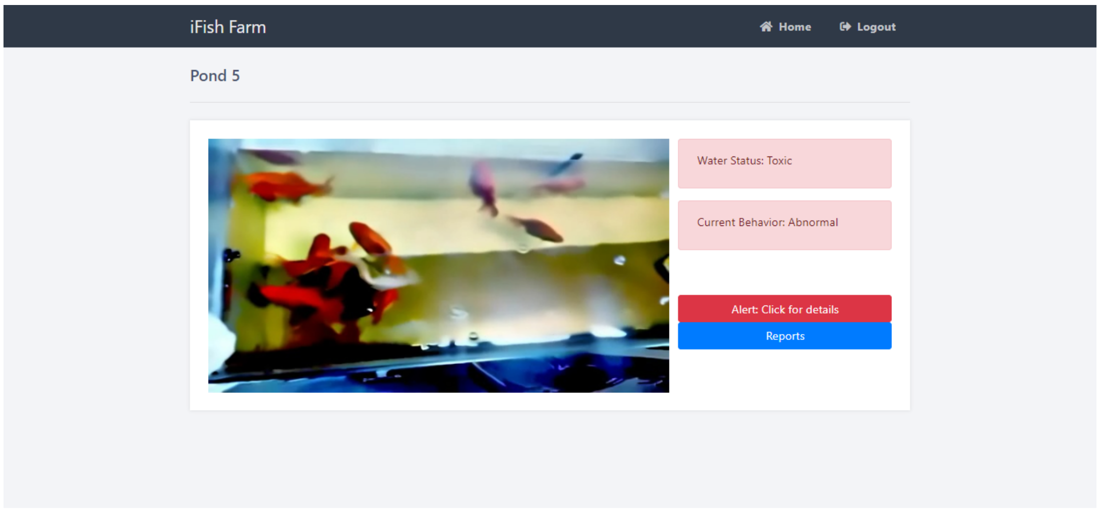


Figure 4.32: Pond Footage Screen with an alert

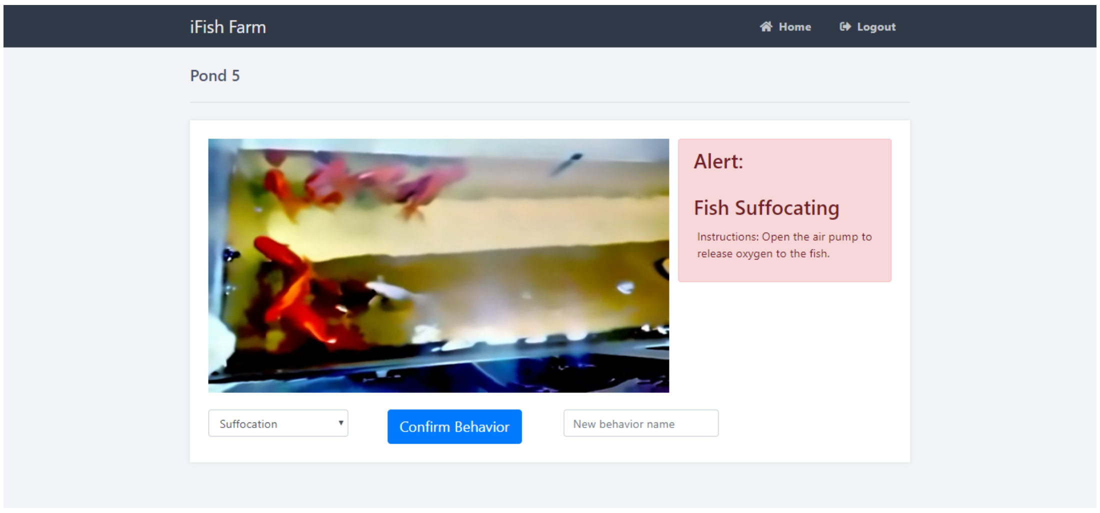


Figure 4.33: Alert Confirmation Screen

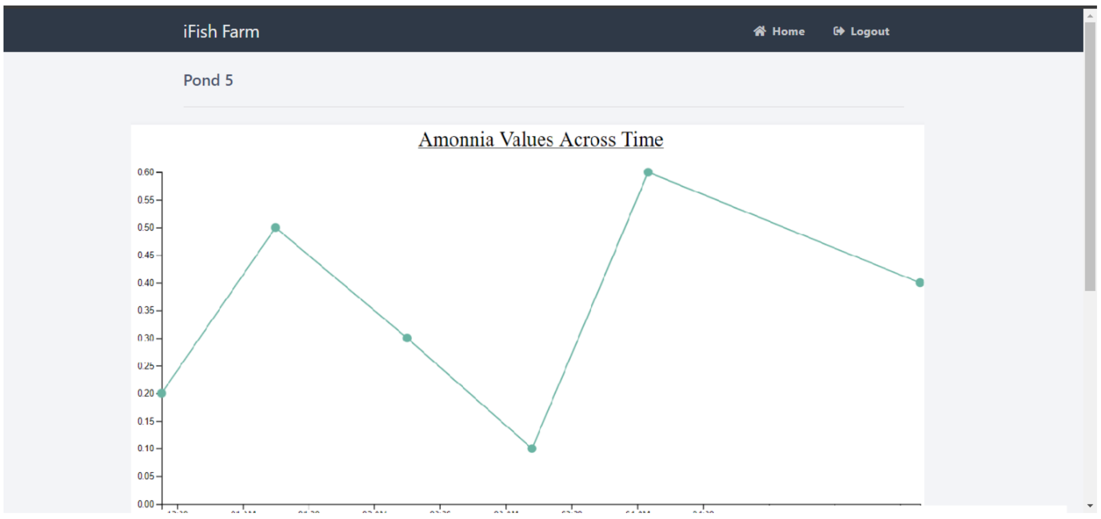
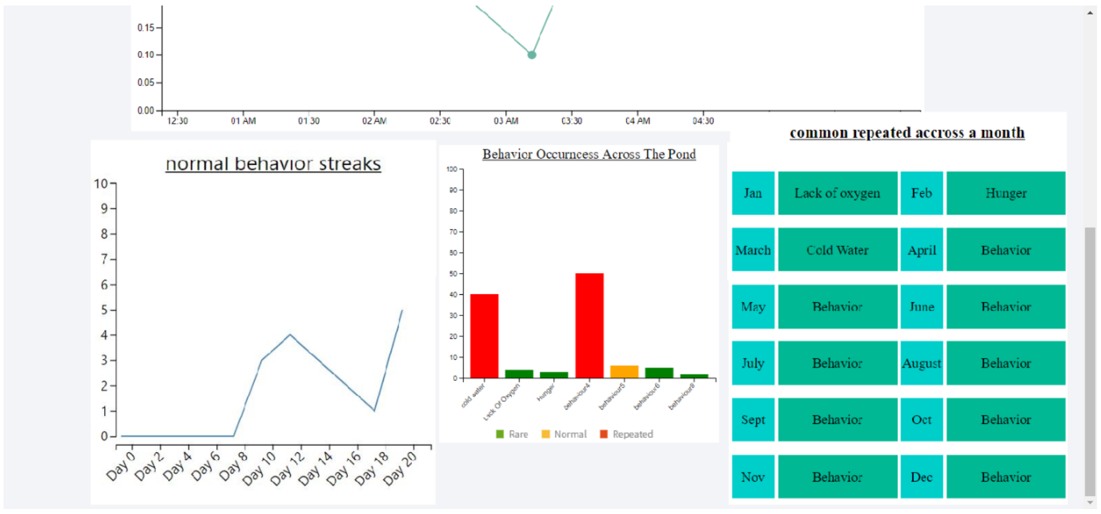


Figure 4.34: Reports Screen



Jan	Lack of oxygen	Feb	Hunger
March	Cold Water	April	Behavior
May	Behavior	June	Behavior
July	Behavior	August	Behavior
Sept	Behavior	Oct	Behavior
Nov	Behavior	Dec	Behavior

Figure 4.35: Reports Screen

4.6.3 Screen Objects and Actions

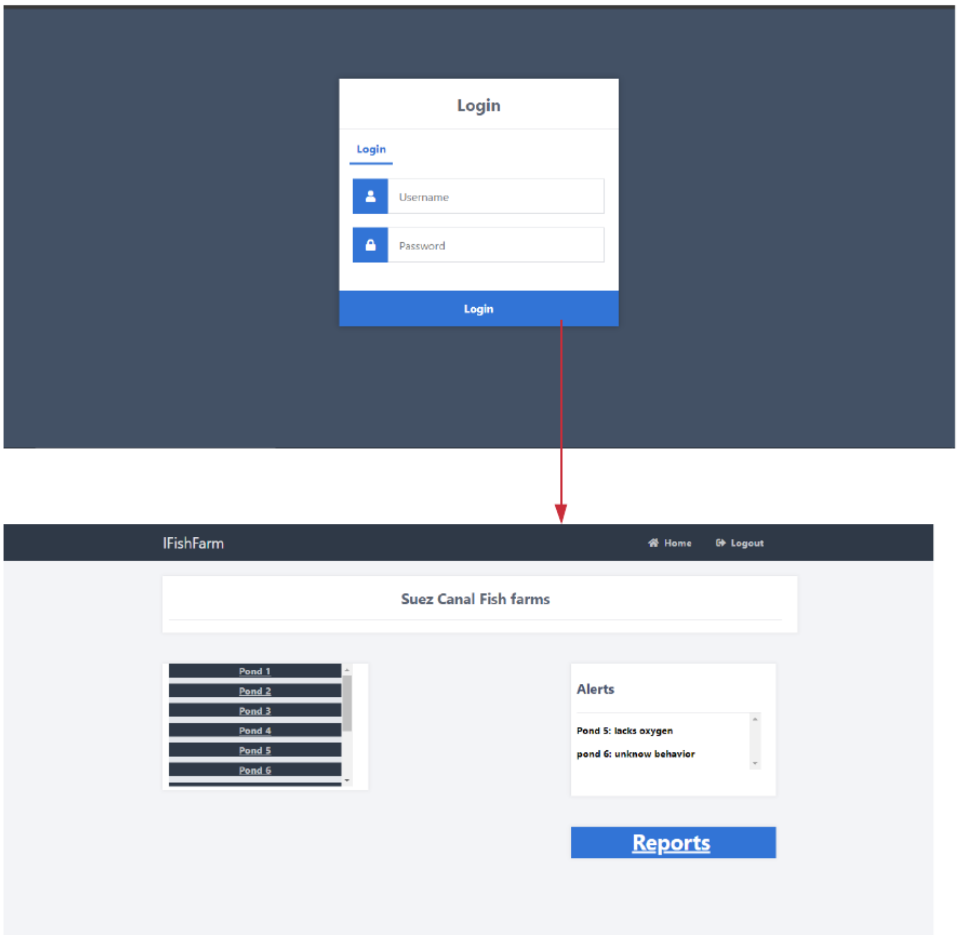
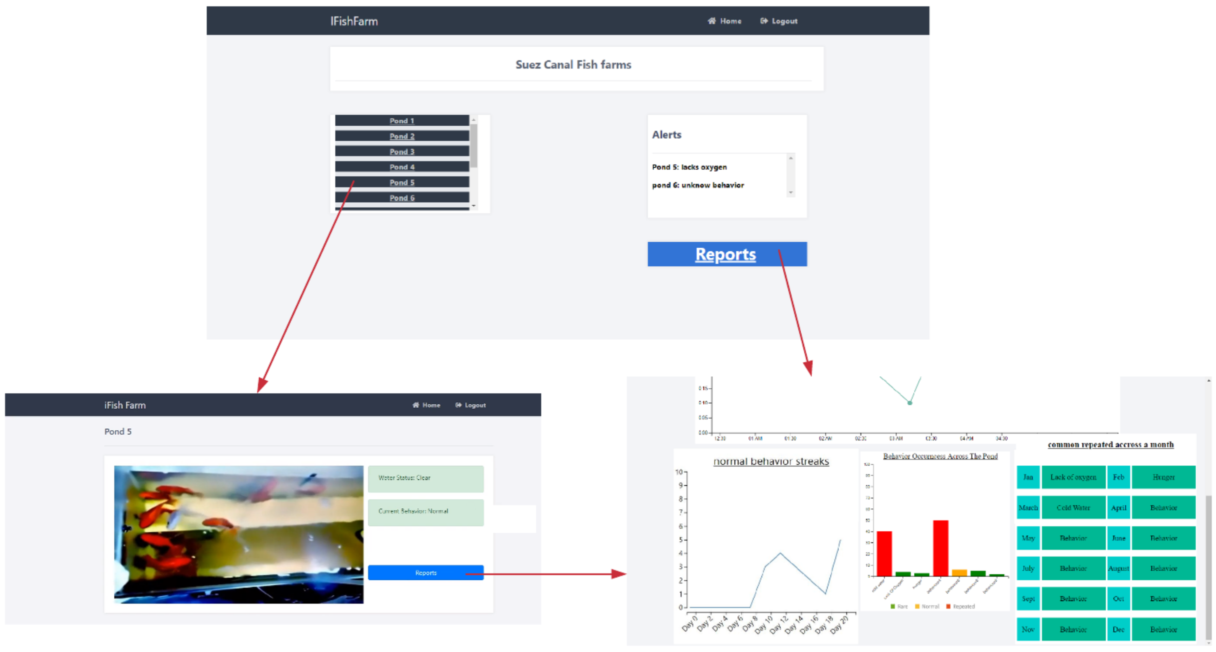


Figure 4.36: Login Action



Alerts

Pond 5: lacks oxygen

pond 6: unknow behavior

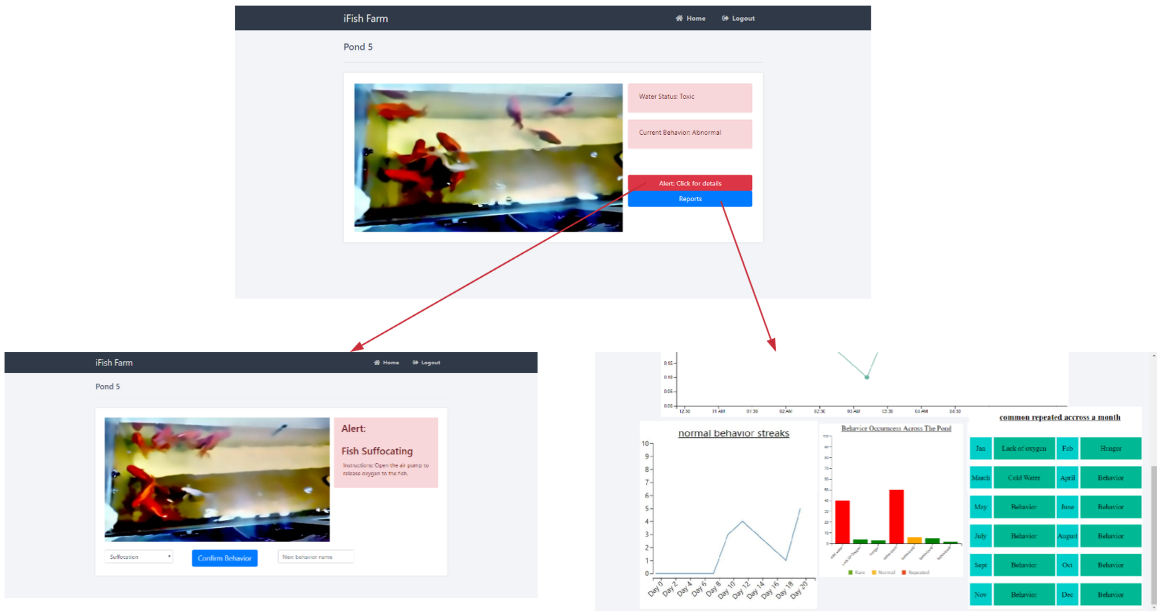
Water Status: Clear

Current Behavior: Normal

common repeated accross a month

Jan	Lack of oxygen	Feb	Hunger
March	Cold Water	April	Behavior
May	Behavior	June	Behavior
July	Behavior	August	Behavior
Sept	Behavior	Oct	Behavior
Nov	Behavior	Dec	Behavior

Figure 4.37: Main Page Actions

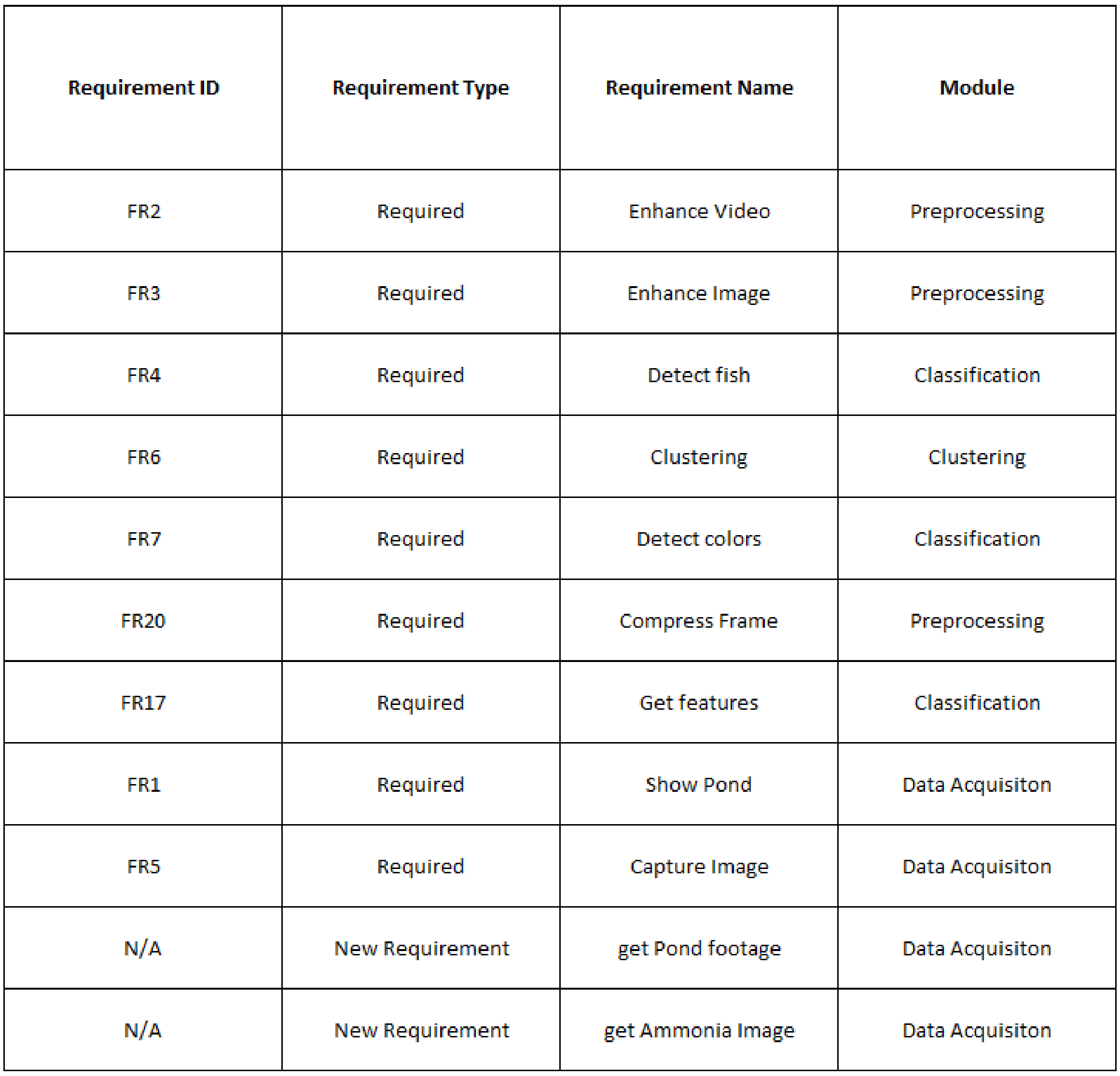


common repeated accross a month

Jan	Lack of oxygen	Feb	Hunger
March	Cold Water	April	Behavior
May	Behavior	June	Behavior
July	Behavior	August	Behavior
Sept	Behavior	Oct	Behavior
Nov	Behavior	Dec	Behavior

Figure 4.38: Pond Footage Page actions when there is an alert

4.7 Requirements Matrix



Requirement ID	Requirement Type	Requirement Name	Module
FR2	Required	Enhance Video	Preprocessing
FR3	Required	Enhance Image	Preprocessing
FR4	Required	Detect fish	Classification
FR6	Required	Clustering	Clustering
FR7	Required	Detect colors	Classification
FR20	Required	Compress Frame	Preprocessing
FR17	Required	Get features	Classification
FR1	Required	Show Pond	Data Acquisiton
FR5	Required	Capture Image	Data Acquisiton
N/A	New Requirement	get Pond footage	Data Acquisiton
N/A	New Requirement	get Ammonia Image	Data Acquisiton

Figure 4.39: Requirement Matrix Table

Chapter 5

Evaluation

5.1 Introduction

Throughout the system different phases it is evaluated through each phase. So, many experiments were done on the system to evaluate its performance. There are five main experiments done on the system. Firstly, an experiment that measures the fish detection accuracy before and after the image enhancement. Secondly, measuring the performance of drawing fish trajectories using optical flow and yolo. Thirdly, this experiment was an extension of the previous one to get better results regarding the fish trajectories. Fourthly, an experiment is done in order to detect the normal and abnormal behavior of fish through fish trajectories. Finally, detecting 3 types of fish behavior through extracted data using a classifier algorithm.

5.2 Experiments Setup

We created an experimental fish tank under the guidance of Fish Research Center of Suez Canal University to conduct tests on our system in a controlled environment. A 60-liter fish aquarium (100x40x35) was brought to home in a testing environment, where exposed to normal sunlight in the morning, and average room lighting at night. The aquarium was kept at room temperature. Also, 15 golden fish were bought to do our experiments. For taking images and videos a web camera was placed above the pond. In processing, a laptop of specs: Intel core i7-6700HQ CPU 2.60 GHz and 16 GB RAM was used with the aid of

Google Colab GPU to provide faster performance in training our model. Our experiment setup is shown in figure 5.1.

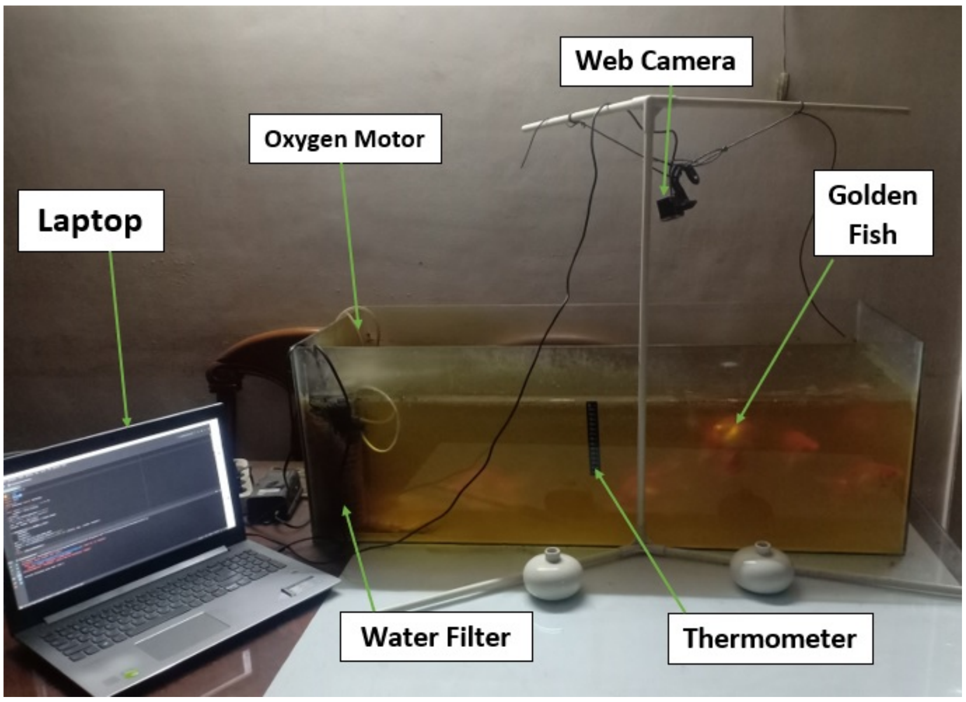


Figure 5.1: Our Experiment Setup

5.3 Datasets

5.3.1 Object Detection Dataset

The needed dataset was obtained through our fish tank, where 2000 photos of goldfish were captured. In conjunction with the fish research center, 400 photos of tilapia fish were obtained from their fish farm ponds to test them on their ponds.

5.3.2 Behaviors Dataset

The dataset for behaviors classification is divided into 3 classes. Normal behavior, hunger behavior, and obstacle induced behavior.

There is a total of 133 behavior samples in the dataset divided into 55 normal behavior samples, 43 hunger behavior samples, and 40 obstacle induced behavior samples.

5.4 Experiment 1

5.4.1 Objective

The main idea of this experiment is to enhance unclear images of water. The water was healthy and clean but was unclear. This experiment also allows us to decide the best location to put the camera according to the pond. The accuracy was measured before and after the enhancement algorithm was applied. 30 images were tested from the two different situations applied (underwater or above) so we can get the average detection of our model.

5.4.2 Results

As mentioned previously, two kinds of images were taken to run tests on the enhancement algorithm. Images where the camera was settled above pond as in figure 5.3 and images where the camera was settled underwater as shown in 5.2.

The model showed that 3 fish can be detected from above and 1 fish can be detected from underwater on average while the images were before enhancement. These numbers are shown in the graph in 5.4 A.

The model showed better outcomes after enhancement as it detected 11 fish from above while, it detected 2 fish from underwater on average. These numbers are shown in the graph in 5.4 B.

So, Firstly, there was no significant change in the underwater images detection before or after enhancement. So, the camera was settled above the pond. Secondly, images after enhancement are better in both situations which solidifies the idea of using the enhancement algorithm to enhance detection accuracy.

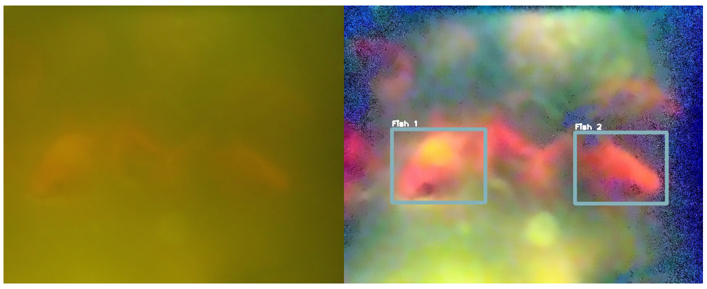


Figure 5.2: Underwater Image, Left: Before Enhancement , Right: After Enhancement

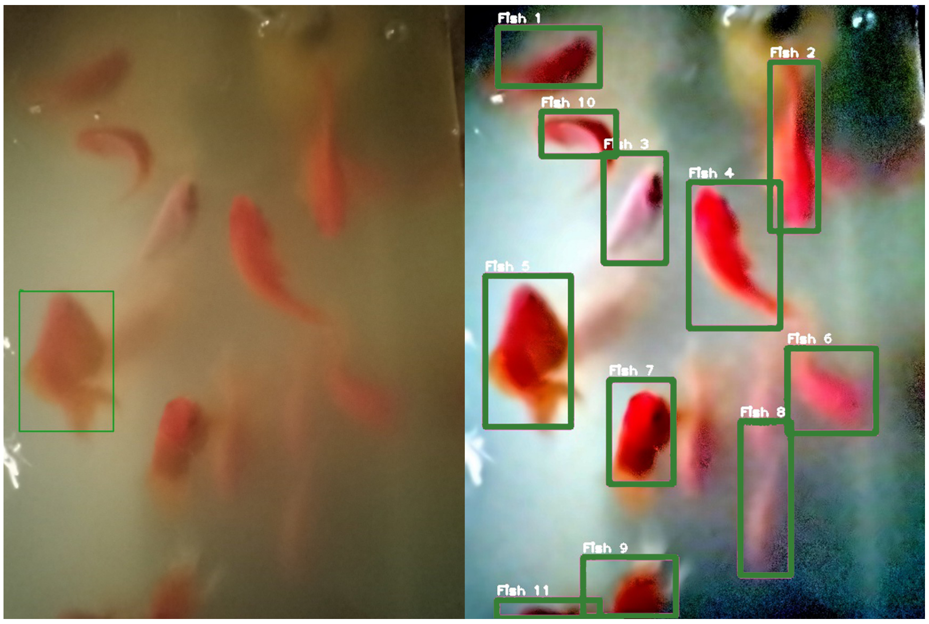


Figure 5.3: Above the pond Image, Left: Before Enhancement , Right: After Enhancement

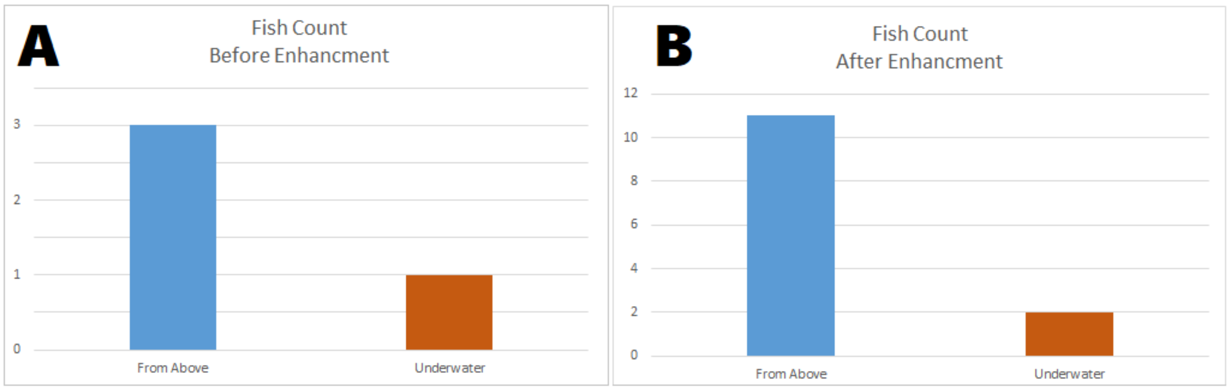


Figure 5.4: Fish Count Average Graph (based on 30 test images) A: Before enhancement, B: After enhancement

5.5 Experiment 2

5.5.1 Objective

This experiment was made to check the improvement of the detection and tracking after using tracking algorithm to track fish frame by frame as YOLO object detection cannot link between fish in each video frame. Also, linear regression was used in order to predict the coordinates of misclassified fish by YOLO. The dataset was also expanded to 2000 images.

5.5.2 Results

As shown in the graph in figure 5.5 there was improvement in the number of fish detected after applying the new model based on several test cases.

The old model using dataset of 400 image was only able to detect 6 fish while the new model using dataset of 2000 image was able to detect 11 fish as shown in figure 5.5. Also, to elaborate visually the difference between fish detection by new model and old model is shown in 5.6.

The model also showed better results in the extracted coordinates and tracking of the fish as the empty frames have been eliminated and all the detected fish are tracked as long as possible. The graph in 5.7 shows that the number of misclassified fish decreased per frame when applying the linear regression prediction and fish tracking algorithm. Also, a snapshot of our dataset that contains fish coordinates is shown in 5.8 where the number of zeros represents fish that has been misclassified (Left image) while in the (Right Image) the number of zeros has significantly decreased.

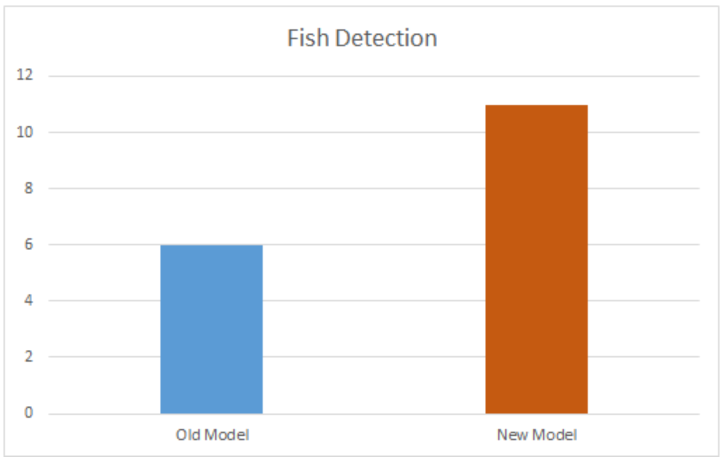


Figure 5.5: Graph showing the difference in fish detection between the new model and old model

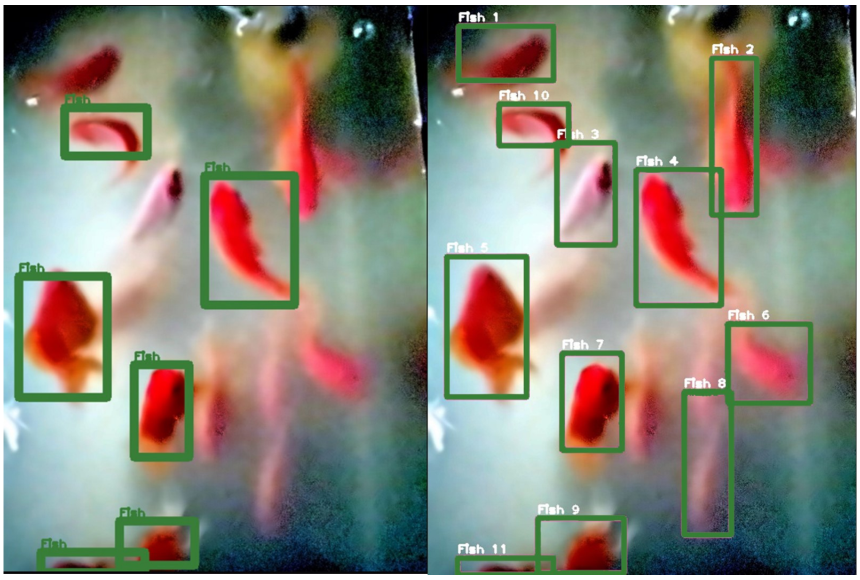


Figure 5.6: Above the pond Image, Left: old model , Right: new model

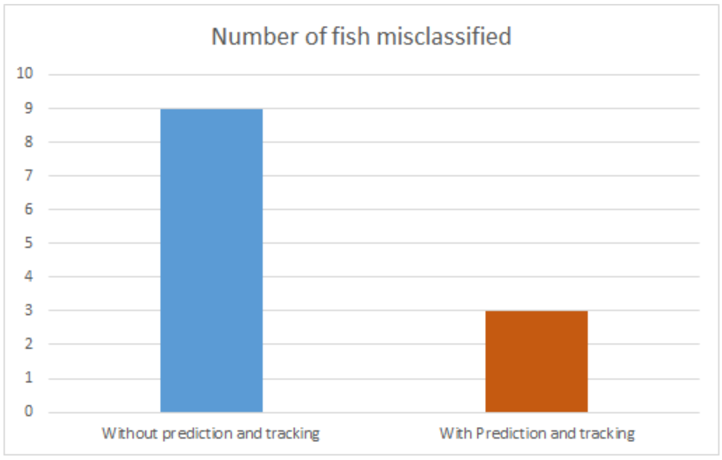
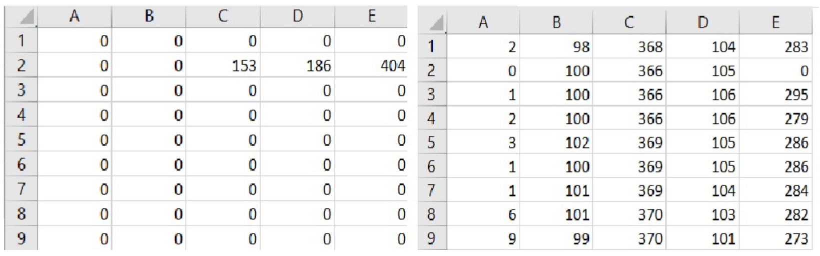


Figure 5.7: Graph showing the number of fish misclassified with and without fish tracking and prediction



	A	B	C	D	E
1	0	0	0	0	0
2	0	0	153	186	404
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0

	A	B	C	D	E
1	2	98	368	104	283
2	0	100	366	105	0
3	1	100	366	106	295
4	2	100	366	106	279
5	3	102	369	105	286
6	1	100	369	105	286
7	1	101	369	104	284
8	6	101	370	103	282
9	9	99	370	101	273

Figure 5.8: Fish coordinates data, Left: without prediction and tracking , Right: with prediction and tracking

5.6 Experiment 3

5.6.1 Objective

This experiment was conducted to measure the performance of fish tracking using the optical flow algorithm with and without the combination with YOLO box coordinates as a tracking point.

5.6.2 Results

As shown in the graph figure 5.9, the optical flow algorithm was able to successfully track the movement of 4 out of the 4 fish detected with YOLO. On the other hand, the optical flow algorithm was only able to detect the trajectory of 1 of the fish without using YOLO object detection.

Also, to elaborate visually, figure 5.10(A) and 5.10(B) shows the movement trajectories which demonstrates the tracking of fish in both test cases (with and without YOLO). Also, the trajectories extracted from the test footage in this experiment is helpful for our further experiments regarding behavior analysis where it is possible to attribute movement patterns to certain behaviors.

As shown in figure 5.10(B) the results from tracking frame by frame and tracking every 6 frames (figure 5.10(C)) is similar so tracking every 6 frames is a better options due to low processing cost.

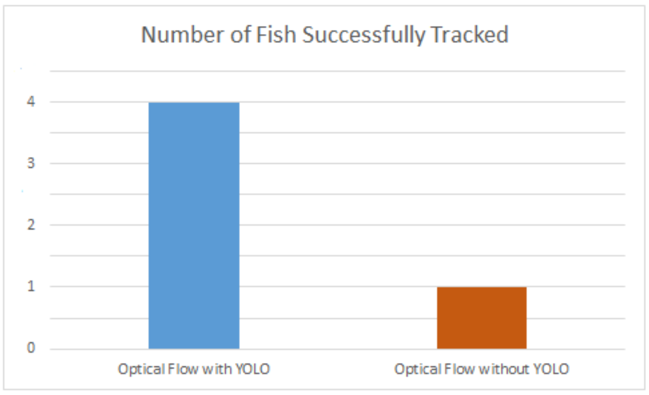


Figure 5.9: Fish Trajectories with and without YOLO.

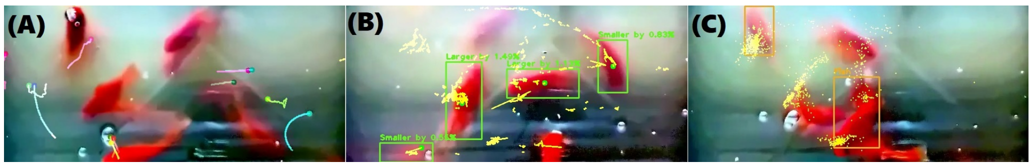


Figure 5.10: A: Trajectories of the fish with only optical flow , B: Trajectories of the fish with optical flow and yolo , C: Trajectories of the fish with optical flow and yolo each 6 frames

5.7 Experiment 4

5.7.1 Objective

The aim of this experiment is to compare between the performance of two methods that are used to draw fish trajectories and track their movements. Firstly, we used combination of YOLO and optical flow, which is used in experiment 2, and compared it with the trajectories and tracking method combination with YOLO used in our system.

5.7.2 Results

The two methods tracked all fish detected by YOLO due to the combination done with it. The trajectories extraction method shown to be better than optical flow method as it produced more accurate trajectories.

To illustrate visually, the trajectory lines drawn in figure 5.11 A shows the optical flow drawn trajectories which lacks accuracy due to the scattered lines and wrong drawn lines of fish movements. While, figure 5.11 B shows the trajectories drawn using the tracking method and yolo gives better results as it draws each line without scattering and tracks every detected fish. Moreover, the trajectories extraction method reduces the overhead that was done by using the optical flow method.

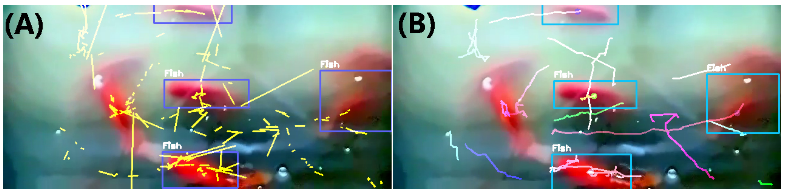


Figure 5.11: A: optical flow and YOLO combination B: Trajectories extracting and YOLO combination

5.8 Experiment 5

5.8.1 Objective

This experiment is carried out to firstly measure the classification accuracy of normal and abnormal fish trajectories, and secondly, to identify a suitable segmentation time to draw trajectories and classify them.

The experiment was conducted in our fish tank, where the water was purified and healthy but visually vague. Some chaotic events may happen around the fish pond to let fish get excited so we can test our system.

5.8.2 Results

Four algorithms were tested on four kinds of trajectories images. The four algorithms are Naive Bayes, KNN, Random forest and Linear Regression. Trajectories images where captured every 5, 10, 15 and 20 seconds to test them. The data was split into 80% training and 20% testing. Every algorithms' result is explained as follows.

Analysis of variance (ANOVA) test was conducted to decide if there is a significant difference between the results of the algorithms. The test returned a p-value = 0.0344, which means that there is a significant difference between our tested algorithm as the p-value is less than 0.05. This validates importance of our selection of our chosen algorithm based on the accuracy.

Random Forest was tested where it gave better results than linear regression. It reached a highest accuracy of 86% in the 10 seconds images. while the rest was kind of lower accuracy compared to this as shown in table 5.14.

KNN was tested with different number of K to determine if it can reach a high accuracy. It was tested by K=1, K=3 and K=5. Overall, the highest accuracy was 89% in the 10 seconds images and the best option for KNN was when the K=3 as shown in the table 5.14 below.

Finally, Naive Bayes was tested and it showed good results as it reached the highest accuracy of all algorithms by reaching 90% in the 10 seconds images. While, on other kind of images it reached a good accuracy but not better compared to the 10-seconds image as shown in table 5.14.

Naive Bayes achieved higher accuracy over all other algorithms in the classification of normal/abnormal behaviors as shown in table 5.14. So, this algorithm was chosen to detect abnormal behavior.

10 seconds was identified as a suitable segmentation time to draw trajectories and classify them as it got the highest average accuracy over all algorithms as shown in the graph in figure 5.12. Also, the difference of the trajectories drawn between each time-frame is shown in figure 5.13 as an explanation that the amount of trajectories drawn may affect the accuracy of the algorithms.

Algorithm	5 seconds	10 seconds	15 seconds	20 seconds
NB	84%	90%	88%	87%
KNN (k=1)	82%	86%	80%	78%
KNN (k=3)	85%	89%	82%	87%
KNN (k=5)	86%	88%	84%	85%
RF	82%	86%	80%	78%

Table 5.1: Algorithms Comparison

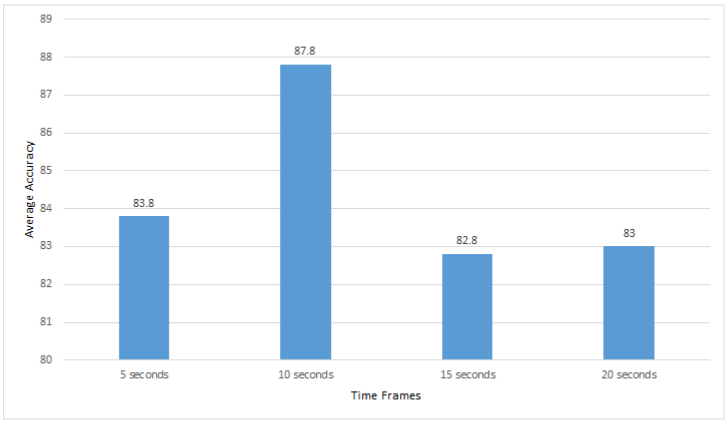


Figure 5.12: Average accuracy on each time-frame

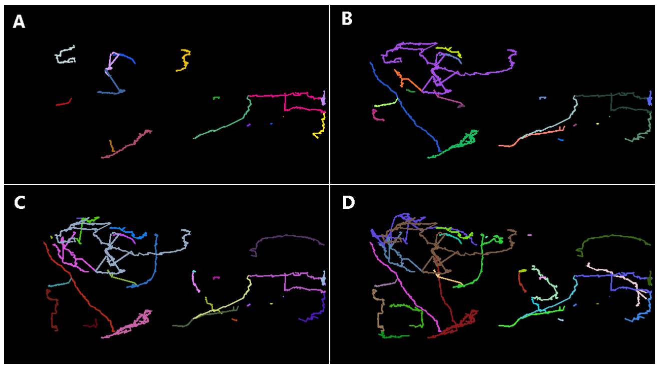


Figure 5.13: (A) Fish trajectories every 5 seconds — (B) Fish Trajectories every 10 seconds — (C) Fish Trajectories every 15 seconds — (D) Fish trajectories every 20 seconds

5.9 Experiment 6

5.9.1 Objective

This experiment is carried out to firstly measure the classification accuracy of classifying fish behavior as normal, hunger, or obstacle induced. And secondly, to determine if the normalization of features using Min-Max scaling improves the classification accuracy.

5.9.2 Results

The experiment tested four classification algorithms on three different behaviors which are normal,hunger and obstacles using the previously mentioned features/data as the dataset for the classifiers. The four algorithms are SVM (Support Vector Machine), Random forest, KNN (K-Nearest Neighbour) and Decision tree. For the dataset splitting, and in order to achieve more accuracy, iFish applied five fold cross validation.

The classification algorithm Random Forest was applied where it performed better than the other classifiers. In the normalized data,it achieved a top accuracy of 90%. Whilst the others is far less reliable than this in performance as shown in figure 5.14.

KNN has been applied with different K numbers to determine whether it can achieve a higher accuracy. The K values were as follows: K=1, K=3 and K=5. In all K values mentioned, the highest accuracy was 81% with the normalized data and when K was set to 3 as shown in the figure 5.14 below. Decison tree was tested where it got the lowest accuracy results between other algorithms. It reached 78% in the normalization and without the normalization as shown in figure 5.14.

Finally, SVM has been applied and it got 86% accuracy with the normalized the data. While, without normalization it reached 85% as shown in figure 5.14.

Since random forest achieved the highest accuracy over all other classifiers in classifying of normal,hunger and obstacles behaviors. So, it was chosen to detect abnormal behavior. Also, the data is chosen to be normalized as normalization using the Min-Max Scaler is better with all algorithms tested.

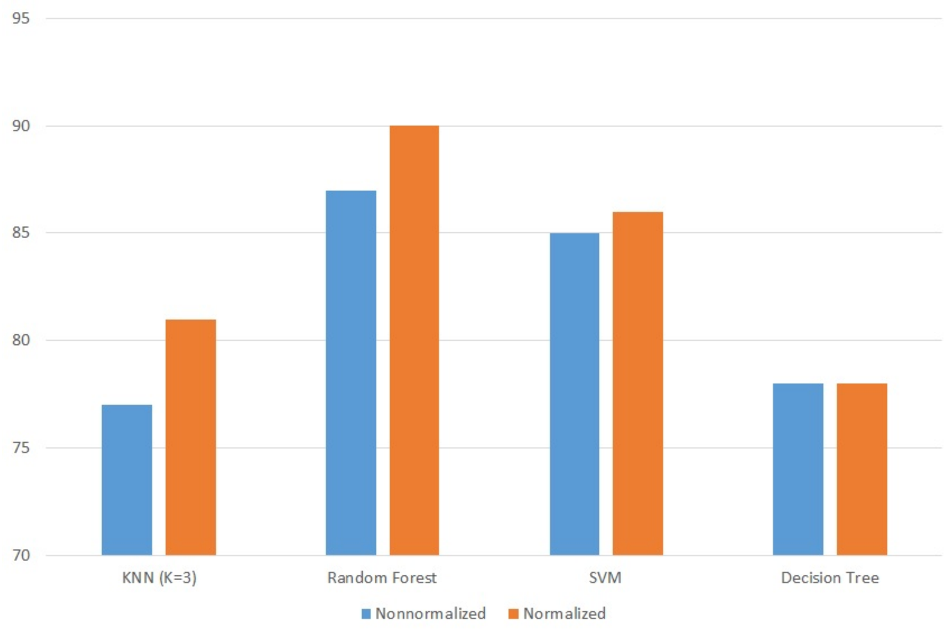


Figure 5.14: Algorithms comparison

Chapter 6

Conclusion

Throughout this document, we presented the design, development and evaluation phases for our system. The system introduces a new approach to manage fish farm ponds and also increase their management efficiency. The system proposes a cost-efficient setup to monitor and analyze ponds for change in fish behavior, and also monitor the changes in water quality. iFish mainly detects fish hunger, chaotic behavior due to obstacles, and normal behavior. Also, the system monitors the ammonia levels in water for timely alerts. The system functionality is achieved through a combination of the image enhancement algorithm, YOLO object detection algorithm and classifiers to generate successful methods to identify abnormal fish behaviors in fish farm ponds.

6.1 Future directions

In the future, we suggest collecting a dataset with a huge variety of fish abnormal behaviours. Also using the STACOG algorithm [27] to extract more features from different scenes in a frame might increase the accuracy.

Bibliography

[1] "Milestones in aquaculture development," available at http://www.fao.org/3/ag158e/AG158E02.htm.

[2] FAO, *FAO Yearbook Fishery and Aquaculture Statistics 2017.* Food and Agriculture organization, 2019.

[3] "Monitoring, record keeping, accounting and marketing," available at http://www.fao.org/fishery/static/FAO$_Training/FAO_Training/General/x6709e/x6709e16.html$.

[4] M. Shaalan, M. El-Mahdy, M. Saleh, and M. El-Matbouli, "Aquaculture in egypt: Insights on the current trends and future perspectives for sustainable development," *Reviews in Fisheries Science & Aquaculture*, vol. 26, no. 1, pp. 99–110, 2018. [Online]. Available: https://doi.org/10.1080/23308249.2017.1358696

[5] W. Bank, *Fish to 2030: Prospects for Fisheries and Aquaculture.* World Bank, 2014.

[6] C. Béné, M. Barange, R. Subasinghe, P. Pinstrup-Andersen, G. Merino, G.-I. Hemre, and M. Williams, "Feeding 9 billion by 2050–putting fish back on the menu," *Food Security*, vol. 7, no. 2, pp. 261–274, 2015.

[7] M. Phillips, N. Tran, L. Kassam, C. Y. Chan, and R. P. Subasinghe, *Aquaculture Big Numbers.* FAO Fisheries and Aquaculture Technical Paper - T601, 2016.

[8] B. J. Boom, P. X. Huang, C. Beyan, C. Spampinato, S. Palazzo, J. He, E. Beauxis-Aussalet, S.-I. Lin, H.-M. Chou, G. Nadarajan, J. Chen-Burger, J. van Ossenbruggen, D. Giordano, L. Hardman, F.-P. Lin, and B. Fisher, "Long-term underwater camera surveillance for monitoring and analysis of fish populations," 2012.

[9] A. Rodriguez, A. J. Rico-Diaz, J. R. Rabuñal, J. Puertas, and L. Pena, "Fish monitoring and sizing using computer vision," in *International Work-Conference on the Interplay Between Natural and Artificial Computation.* Springer, 2015, pp. 419–428.

[10] S. Luo, X. Li, D. Wang, J. Li, and C. Sun, "Automatic fish recognition and counting in video footage of fishery operations," 12 2015, pp. 296–299.

[11] L. Parra, S. Sendra, L. García, and J. Lloret, "Design and deployment of low-cost sensors for monitoring the water quality and fish behavior in aquaculture tanks during the feeding process," *Sensors*, vol. 18, p. 750, 03 2018.

[12] "Gramener and microsoft ai for earth help nisqually river foundation automate identification of fish species," available at https://partner.microsoft.com/en-us/case-studies/gramener. [Online]. Available: https://partner.microsoft.com/en-us/case-studies/gramener

[13] "Introducing tidal," available at blog.x.company/introducing-tidal-1914257962c3., journal=Medium, X, The Moonshot Factory, publisher=Davé, Neil.

[14] S. Duggal, S. Manik, and M. Ghai, "Amalgamation of video description and multiple object localization using single deep learning model," in *Proceedings of the 9th International Conference on Signal Processing Systems*, ser. ICSPS 2017. New York, NY, USA: ACM, 2017, pp. 109–115. [Online]. Available: http://doi.acm.org/10.1145/3163080.3163108

[15] R. Lumauag and M. Nava, "Fish tracking and counting using image processing," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM).* IEEE, pp. 1–4.

[16] Y. Toh, T. Ng, and B. Liew, "Automated fish counting using image processing," in *2009 International Conference on Computational Intelligence and Software Engineering.* IEEE, 2009, pp. 1–5.

[17] B. J. Boom, P. X. Huang, C. Beyan, C. Spampinato, S. Palazzo, J. He, E. Beauxis-Aussalet, S.-I. Lin, H.-M. Chou, G. Nadarajan *et al.*, "Long-term underwater camera surveillance for monitoring and analysis of fish populations," *VAIB12*, 2012.

[18] Z. Chen, J. Cao, Y. Tang, and L. Tang, "Tracking of moving object based on optical flow detection," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. 2. IEEE, 2011, pp. 1096–1099.

[19] N. D. Nguyen, K. N. Huynh, N. N. Vo, and T. Van Pham, "Fish detection and movement tracking," in *2015 International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 2015, pp. 484–489.

[20] C. Beyan and R. B. Fisher, "A filtering mechanism for normal fish trajectories," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 2286–2289.

[21] A. Nady, A. Atia, and A. Abutabl, "Real-time abnormal event detection in crowded scenes," *Journal of Theoretical and Applied Information Technology*, vol. 96, pp. 6064–6075, 09 2018.

[22] J. Wang, N. Wang, L. Li, and Z. Ren, "Real-time behavior detection and judgment of egg breeders based on yolo v3," *Neural Computing and Applications*, pp. 1–11, 2019.

[23] M. Yang, L. Rashidi, A. S. Rao, S. Rajasegarar, M. Ganji, M. Palaniswami, and C. Leckie, "Cluster-based crowd movement behavior detection," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2018, pp. 1–8.

[24] C. Beyan and R. B. Fisher, "Detecting abnormal fish trajectories using clustered and labeled data," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 1476–1480.

[25] C. Tang, U. F. von Lukas, M. Vahl, S. Wang, Y. Wang, and M. Tan, "Efficient underwater image and video enhancement based on retinex," *Signal, Image and Video Processing*, pp. 1–8, 2019.

[26] H. Lu, Y. Li, and S. Serikawa, "Underwater image enhancement using guided trigonometric bilateral filter and fast automatic color correction," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 3412–3416.

[27] T. Kobayashi and N. Otsu, "Motion recognition using local auto-correlation of space–time gradients," *Pattern Recognition Letters*, vol. 33, no. 9, pp. 1188–1195, 2012.