

Automatic Recognition of Fish Diseases in Fish Farms

Ahmed Waleed, Hadeer Medhat, Mariam Esmail, Kareem Osama
Supervised by: DR.Taraggy M Ghanim, Eng.Radwa Samy

February 2020

1 Introduction

1.1 Purpose of this document

The purpose of this Software design document is to describe the architecture and system design for our Automatic recognition of Fish Diseases in Fish Farms system. Our system mainly detects and diagnose fish diseases in fish farms automatically and examine water quality. We provide a full illustration for each stage and development process. This document will explain in details, the components of the system represented in sequence diagram. The implementation of the project and its development will be shown in the class diagram. It illustrates the system components and how they interact with each other.

1.2 Scope of this document

This Software design document targets owners of fish farms and experts in the fish farm domain. Our Application will help them in saving much more time rather than manual detection. Our proposed project is an automated system to diagnose fish diseases. The main objective is detection and improvement of classification accuracy of fish diseases and fish abnormal behaviors. Our system is a recognition system that performs pre-processing, segmentation, feature extraction and classification on image.

1.3 Overview

Our system is created to performs detection and identification of fish diseases and analyze fish abnormal behavior. Its goal is to prevent and control spreading of diseases. Our system will work on three different types of fish diseases. These diseases are Epizootic ulcerative syndrome (EUS), Ichthyophthirius (Ich) and Columnaris. Our system will consist of a mobile application that will read the data that will be collected from sensors (PH and Temperature) and receive

notification if any improper change happened. The farm owner will be provided by an interface through which he could monitor the farm environment.

1.4 Definitions and Acronyms

Term	Definition
UI	User interface is the space where interactions between humans and machines occur
YCBCR	Is a family of color spaces used as a part of the color image in video. Y is the luma component and CB and CR are the blue-difference and red-difference chroma components
GUI	Graphical user interface
Firebase	A mobile and web application platform
GSM	Global System for Mobile
CNN	Convolutional Neural Network
MVC	Model-View- Controller
EUS	Epizootic ulcerative syndrome
Ich	Ichthyophthirius
FC	Fully connected
L*A*B	It is a color space for color measurement. L* for the Lightness, a* for Red/Green Value and b* for Blue/Yellow Value.
HSV	Hue Saturation Value
HOG	Histogram of oriented gradients
ROI	Region of interest

2 System Overview

Our proposed approach aims to detect and diagnose fish diseases in fish farms automatically. Raspberry Pi kit is used and connected to sensors, camera and a personal computer(PC). The proposed system is presented in three consequent stages. During the first stage, water quality is examined by measuring water temperature and rate of pH, while fish captures are acquired by the camera. The kit gets the sensor’s measurements and acquired fish captures. In the second stage, all inputs are passed to the PC for processing. This processing concerns detecting any abnormal behavior in farm environment and fish infections. Infection detection starts by pre-processing, then segmentation of infected areas and finally classification. In pre-processing phase, different color spaces were applied on input images which are RGB, Ycbr and XYZ. For segmentation, we built a Gaussian distribution in the XYZ color space. The XYZ color space was used in the phases of classification by convolution neural networks (CNN). In behavior classification, Histogram of gradient (HOG) feature is applied on videos of fish trajectories to detect fish and displays bounding boxes around the detected fishes. Then, we randomly crop the detected fish and then we applied AlexNet for fish disease classification. Finally in the third Stage, the kit is responsible for sending notification to farm owner’s mobile phones through an application,

In case that mobile is not connected to the internet, GSM800 component, sends message to mobile to notify them of any improper changes in farm environment or detected infections. The overview of our proposed approach is shown in Fig. 1.

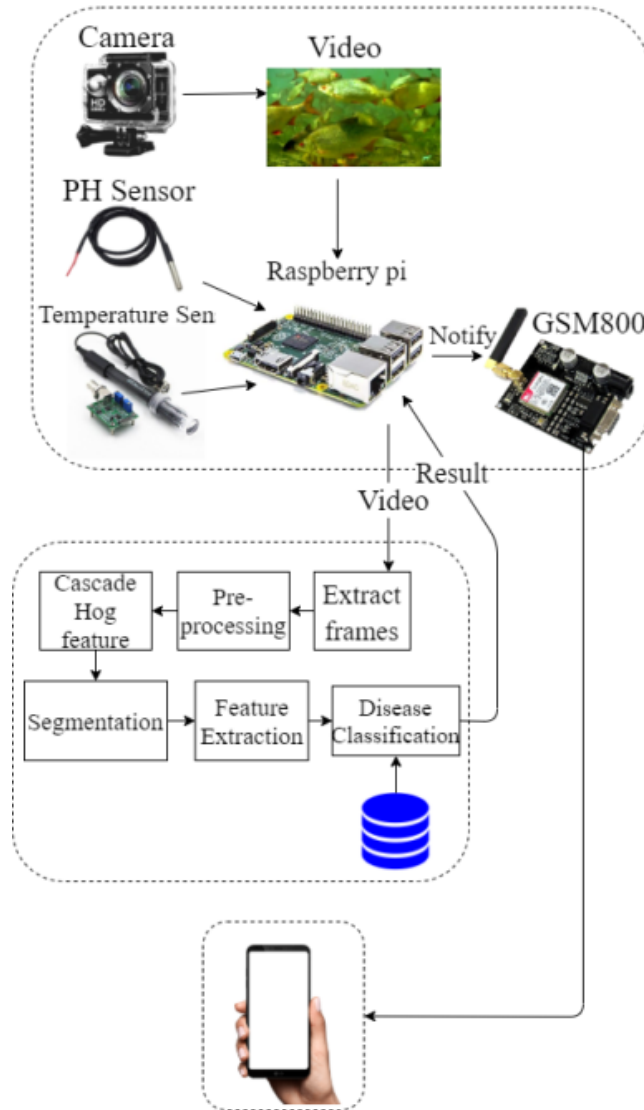


Figure 1: Proposed System Overview

3 System Architecture

3.1 Architectural Design

The System is designed in MVC architecture.

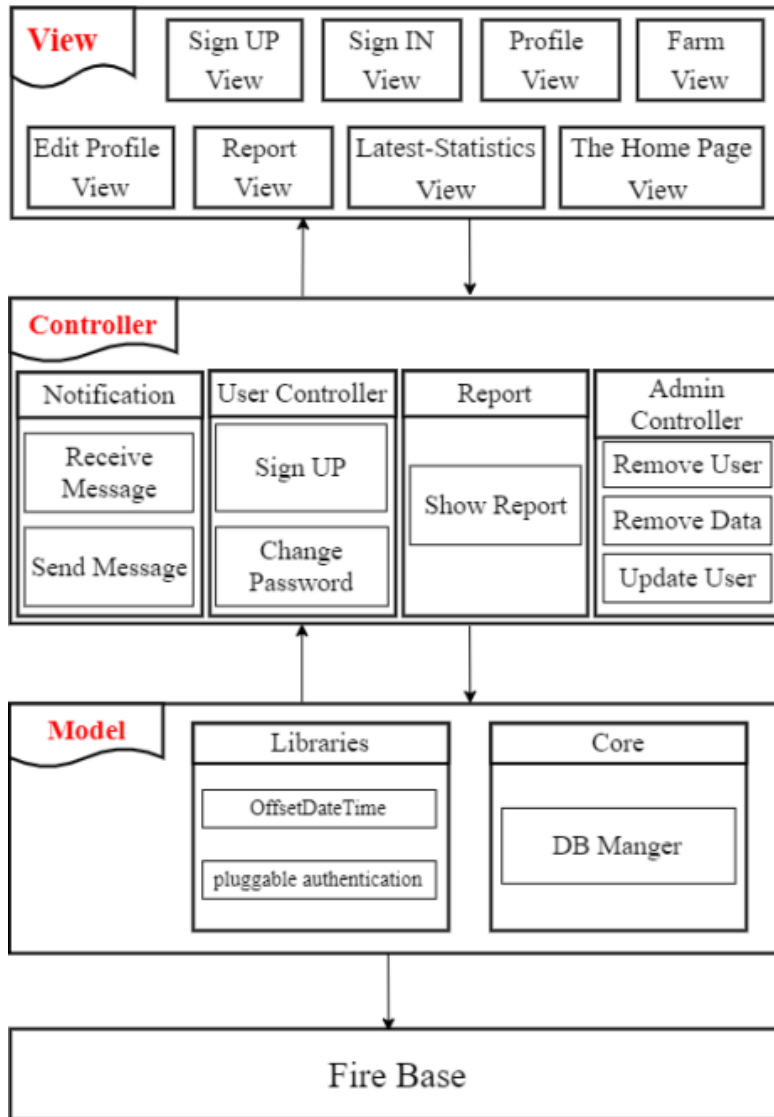


Figure 2: Architectural Design

3.1.1 View

It is responsible for the presentation of data and representing the User Interface(UI). It consists of some forms to help the user to navigate through the system and through the functionalities of the system. such as:

- Sign up: This interface allows new users to sign up to the application, by entering their information for the database to store.
- Sign in: This interface is meant for the users and admins to login the application.
- Home Page : This interface that manages all the modules inside our system. There are 4 modules that the user may interact with.
 - 1) Edit Profile: This interface allow each user to edit his personal data.
 - 2) Profile: This interface allow users to show all his personal information.
 - 3) Report: This interface is for listing all the daily sensor readings.
 - 4) Latest-Statistics: This interface is for listing all latest details for fish farm including sensor readings.

3.1.2 Model

- Libraries
 - 1)Offset-Date-Time is an immutable representation of a date-time with an offset. This class stores all date and time fields, to a precision of nanoseconds, used to model data in simpler applications. This class may be used when modeling date-time concepts in more detail
 - 2)Firebase Authentication: Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.It use to help to Create a new signIn method which takes in an email address and password, validates them, and then signs a user in with the signInWithEmailAndPassword method.
- Core
 - 1) DB Manger: The Database Manger take control by notifying the server with recent update, like a if there is new reading data arriving, the server also uses it when finishing processing by sending notification to the admin if there is any infection in the farm. Or storing the reading data from the sensors .

3.1.3 Controller

It is responsible for binding the view and model. It takes input from the view and sends it to the database and retrieves data from the model and sends it to the view to be previewed there. The handling of all the data coming from the input

hardware devices including PH readings coming from the PH sensor, temperature reading coming from the temperature sensor. Some of the controllers we are having; admin Controller that is responsible for handling interactions made within admin views, user controller that is responsible for handling interactions made by fish farm owner and the notification controller that is responsible for sending notifications to the user when there is any improper change in fish farm environment.

3.2 Decomposition Description

3.2.1 Class Diagram

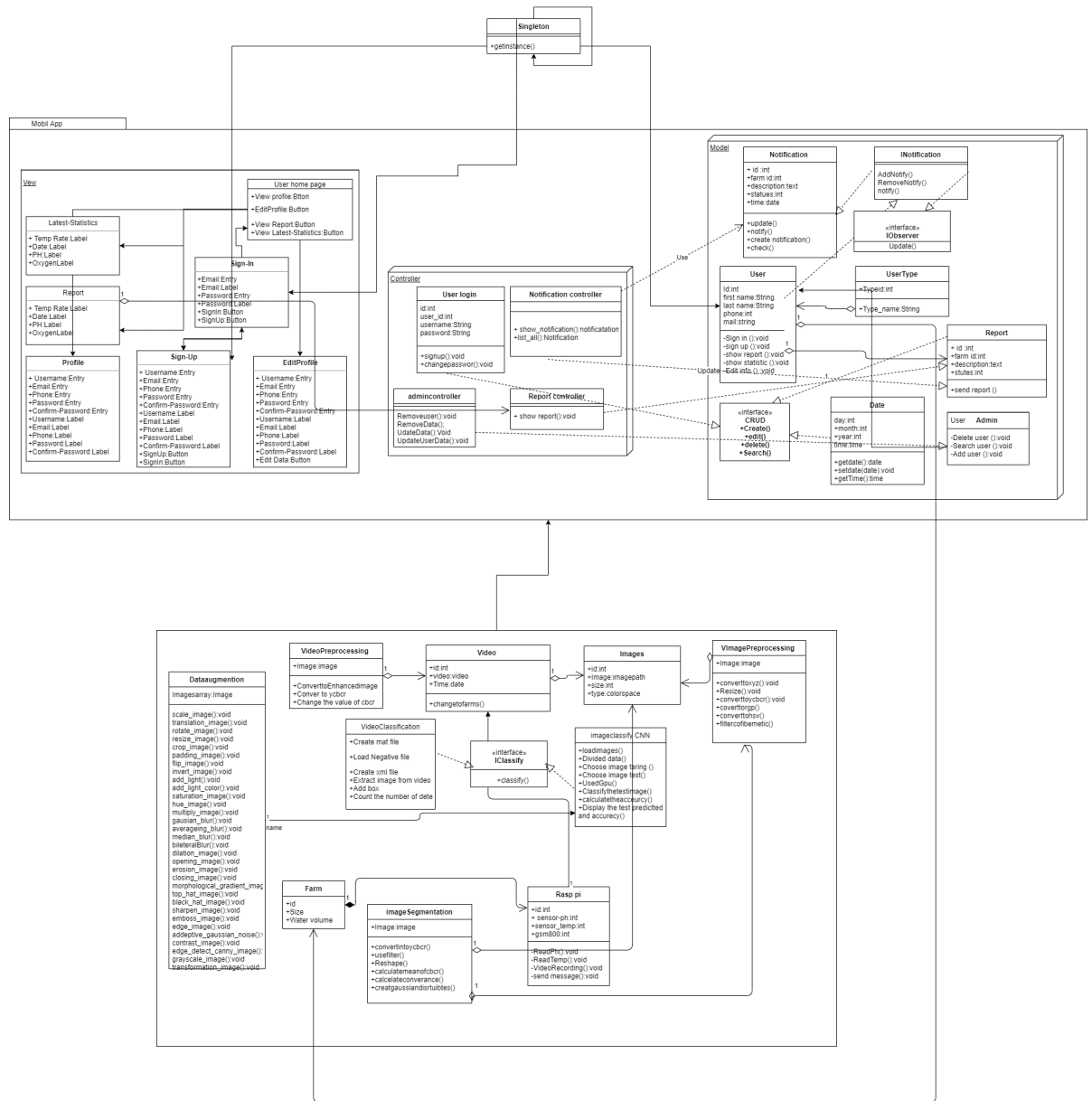


Figure 3: Class diagram

Snaps of our class diagram:

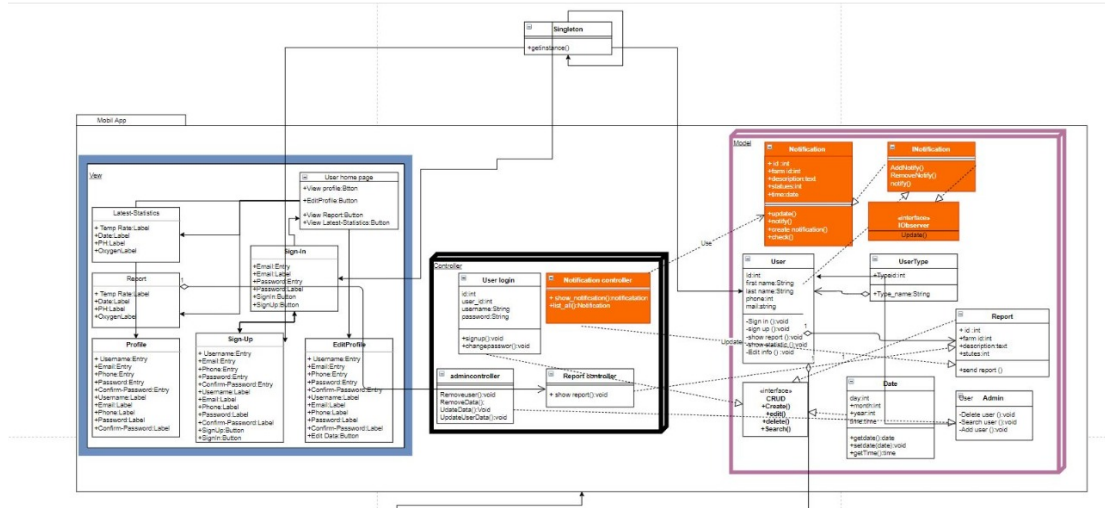


Figure 4: MVC Design Pattern

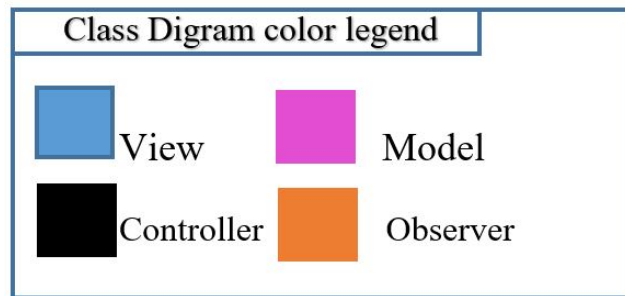


Figure 5: This class diagram implements the MVC design pattern which is used to separate the application's view from it's model by interacting via the controller.

- Model: Represents an object carrying data. It can also have logic to update controller if its data changes.
- View: Represents the visualization of the data that model contains. It is responsible for the presentation of data and representing the user interface(UI).
- Controller: Acts on both model and view. It is responsible for binding the

view and model, it controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

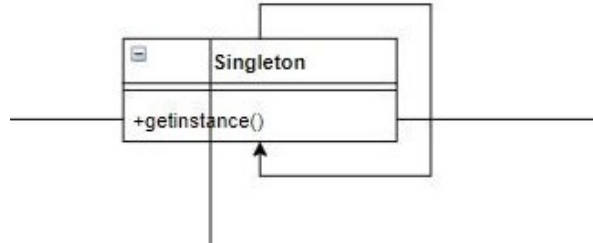


Figure 6: Singleton Design Pattern

In singleton design pattern, the class extends itself. It is one of the simplest design patterns. In our case, it is responsible for the database connection.

- The Observer Design Pattern: Defines a one-to-many relationship so that when one object changes state, the others are notified and updated automatically. Firebase is being used in this case for handling the notifications for the farm owners' mobile application.

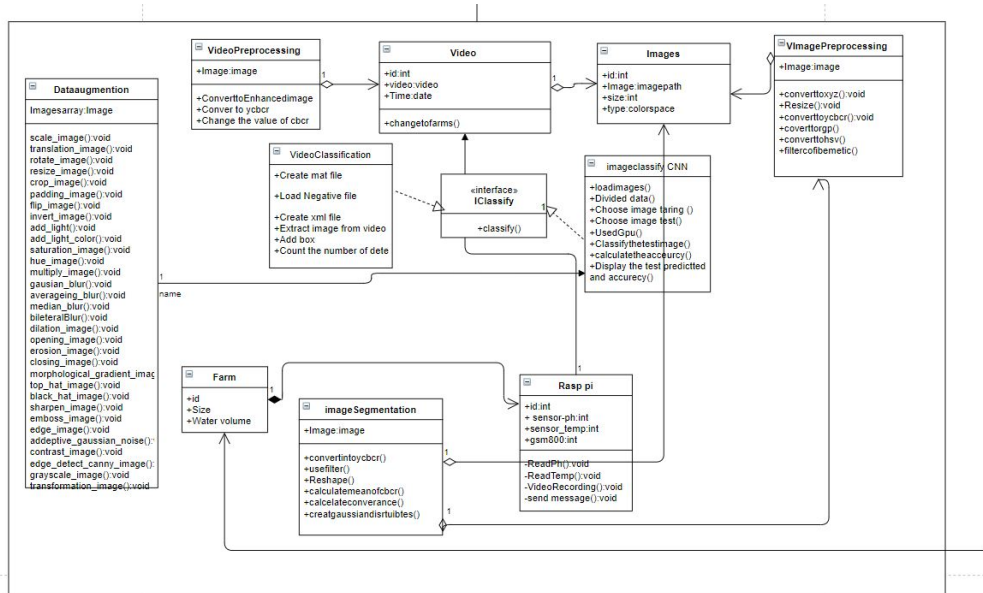


Figure 7: Hardware

- Image Pre-processing: contain ID, images, this table used to apply different color spaces on input images during pre-processing phase

- Image: This table contain ID, Name, path, size to be able to differ between the different images.
- Video Pre-processing: contain ID, images, this table used to apply different color spaces on input images during pre-processing phase
- Video: This table contain ID, Video, date to be able to differ between the different Video.
- Image segmentation: Contain ID, Image.this table used to conversion from RGB color space to Ycbr
- Data augmentation: Contain ID, Image.this table used to increase the number of our dataset
- Image Classification: Contain ID, Image.This table used to detected infections.

3.2.2 Activity Diagram

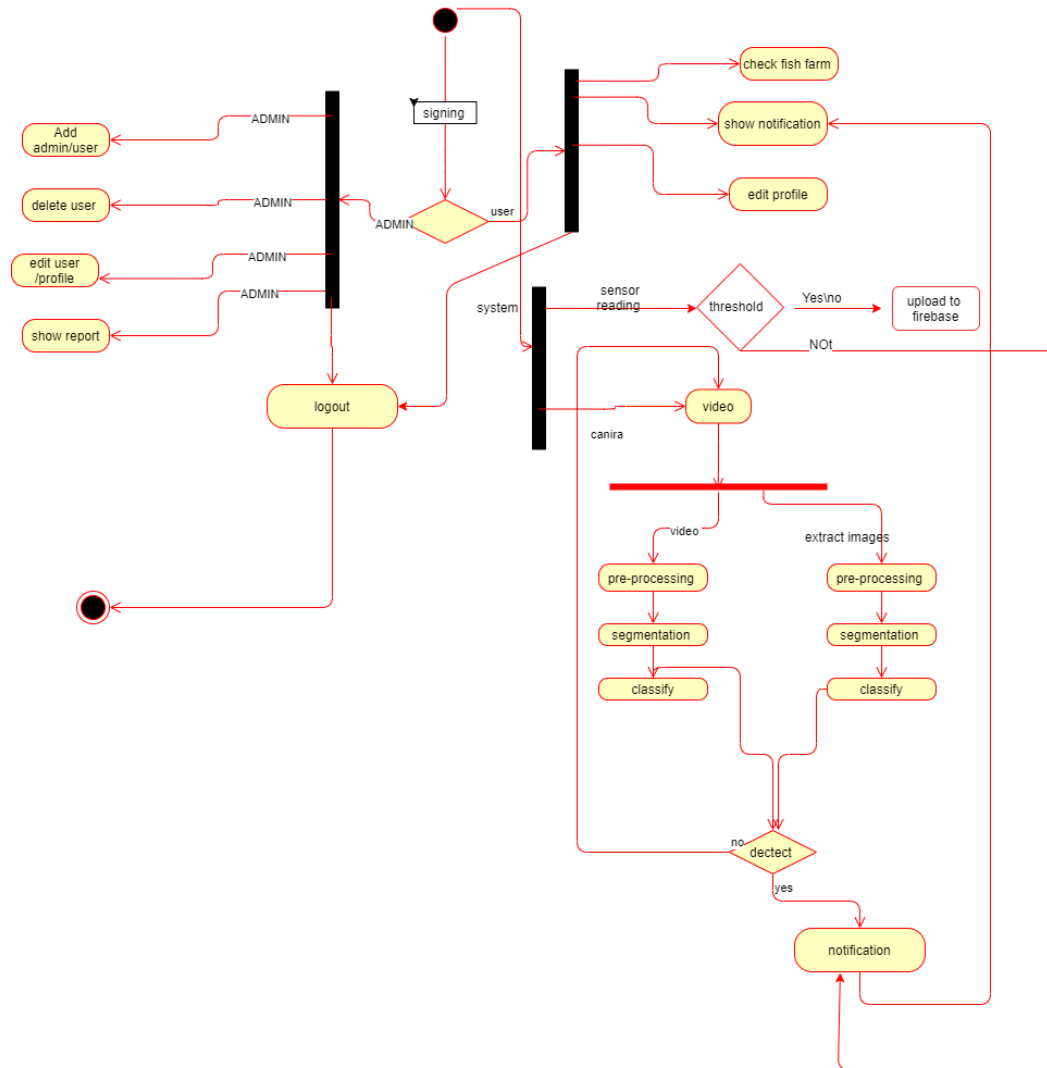


Figure 8: Activity diagram

3.2.3 Sequence Diagrams

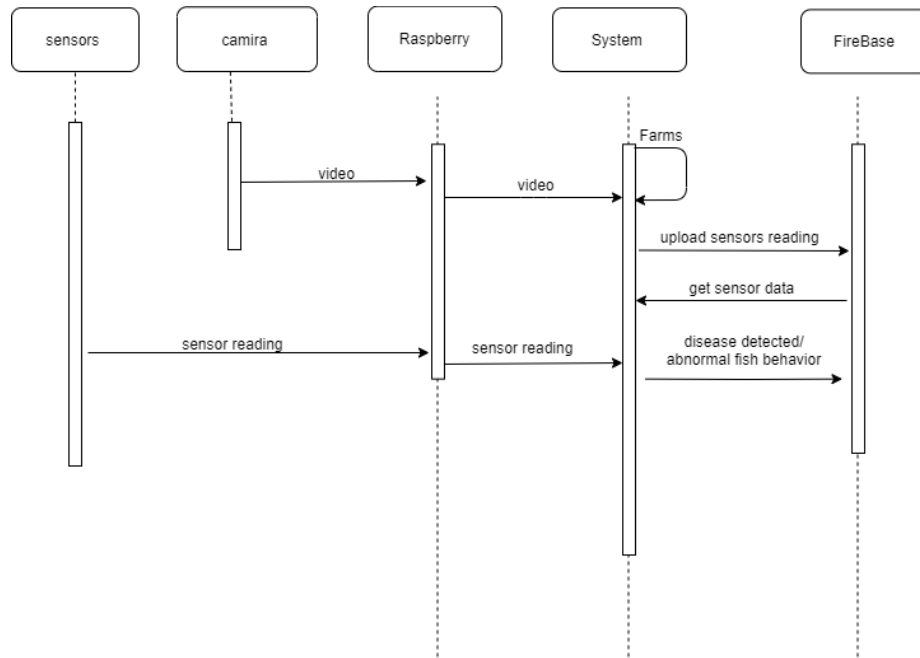


Figure 9: Sequence diagram

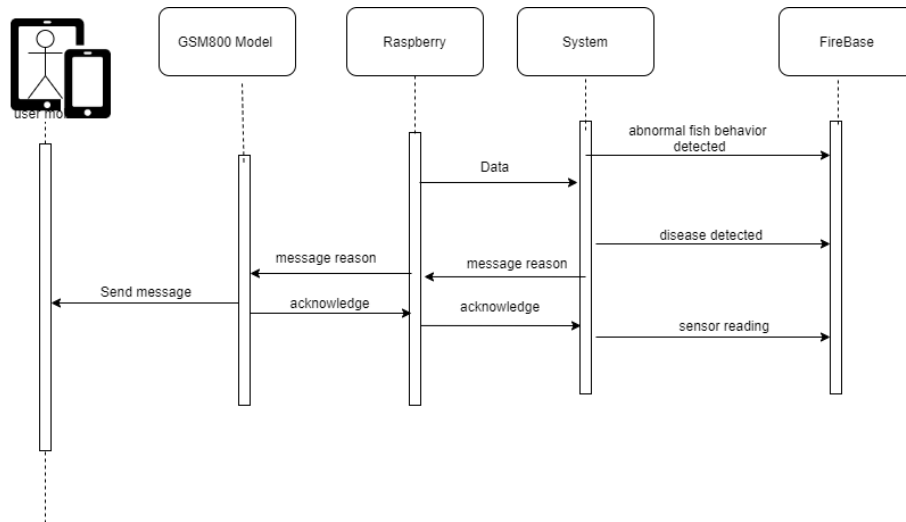


Figure 10: Sequence diagram

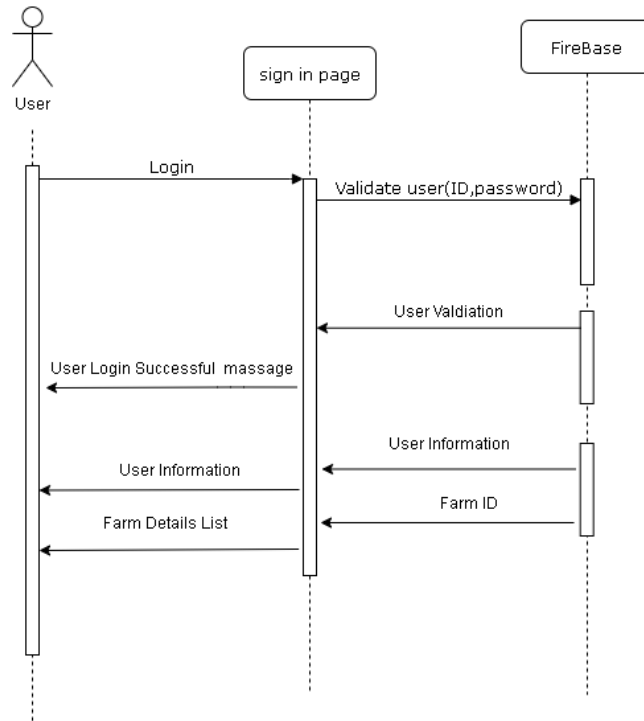


Figure 11: Sequence diagram

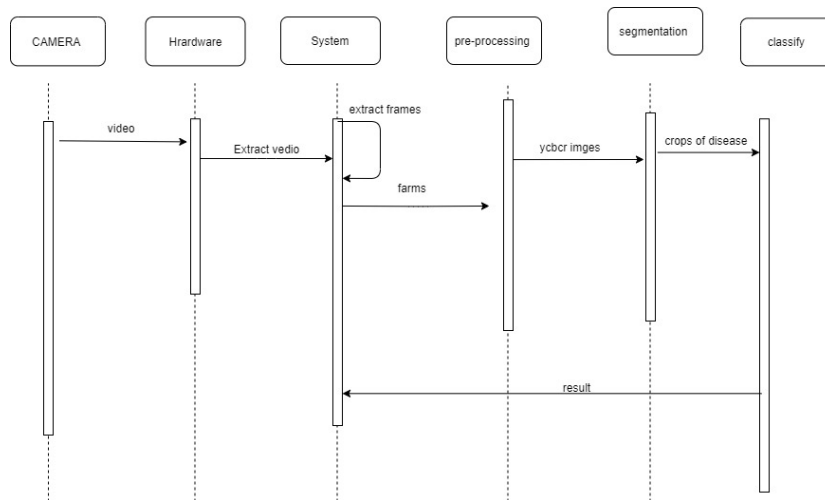


Figure 12: Sequence diagram

3.3 Design Rationale

As mentioned previously, we have used Model-View-Controller (MVC) as our architecture as it differentiating the layers of a project in Model,View and Controller for the Re-usability of code and better maintenance. So, we can easily make modifications, re-use and optimize functionality part as it is our core. Accuracy is a very important aspect of our system so it will be very sensitive with data so this should be developed in a very accurate and efficient way.

4 Data Design

This is a relational preview of the database.

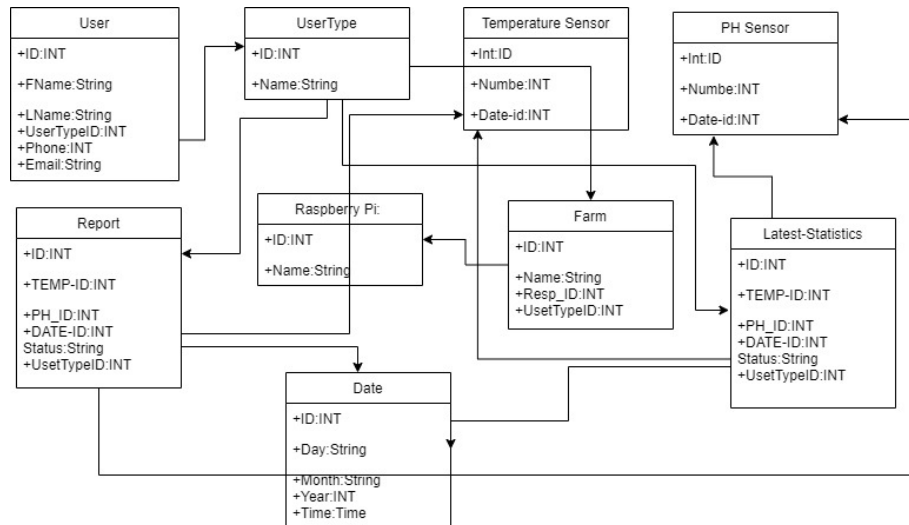


Figure 13: Database

4.1 Data Description

- User: This table contain ID, first-name, last-name, phone, Email. It also contains usertype-ID because we have different user type (Admin, user) and each type has different privileges.
- UserType: This table contain ID, Name to be able to differ between the different users.
- Report: This table contain ID, status, it also contains Farm id because we have different Farms, Date-id to get the current date, temperature-id to get the temperature rate of the current day and PH-id to get the PH rate of the current day .in this table the Admin need to listing all

the sensor daily reading. it contains usertype-ID because each type has different report to view.

- Latest-Statistics: This table contain ID, status. it also contains Farm id because we have different Farms, Date-id to get the date per day, temperature-id to get the temperature rate every day and PH-id to get the PH rate every day. in this table the Admin need to listing all the sensor weakly reading.
- Raspberry Pi: This table contain ID, Name to be able to differ between the different Raspberry Pis
- Farm: This table contain ID, Name to be able to differ between the different Farms. And also contain Raspberry-ID because each farm has its own Raspberry Pi, it also contains usertype-ID because each type has its own Farm.
- Temperature Sensor: This table contain ID, Number, this table contain all the temperature data reading and also contain date-ID because every Temperature rate has its own current date
- PH Sensor: This table contain ID, Number, this table contain all the PH data reading and also contain date-ID because every PH rate has its own current date
- Date: This table contain ID, day, month, year, time. in thus table the Admin need to get the whole data about the sensor reading

5 Component Design

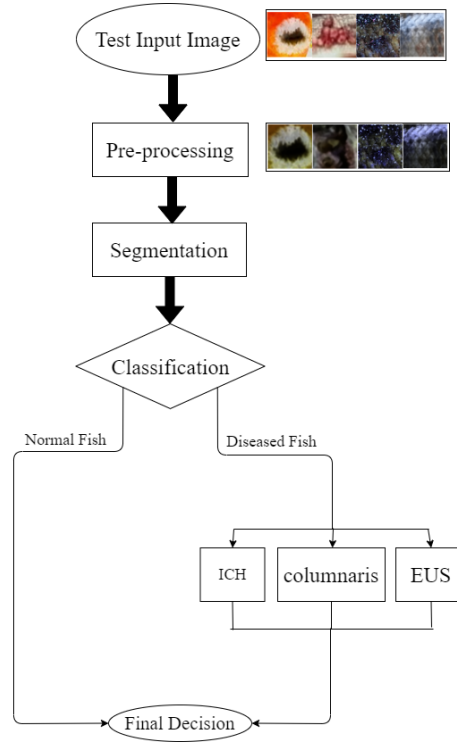


Figure 14: Flowchart of our system approach

5.1 Pre-processing

In this phase we prepare the sample images for the segmentation process, we applied different color spaces on input images during pre-processing phase, as shown in Fig.6. Three different color spaces were applied; RGB as shown in Fig. 15a, Ycber as shown in Fig. 15b and XYZ as shown in Fig. 15c

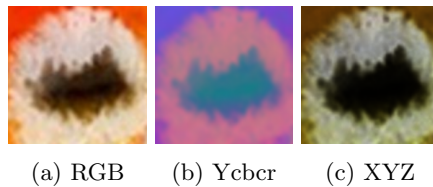


Figure 15: Infected Area in Different Color Spaces

5.1.1 RGB color space

RGB represents color as red, green and blue.

5.1.2 Ycbr color space

YCbCr, is a family of color spaces used as a part of the color image pipeline in video. the Y is the brightness (luma), Cb is blue minus luma (B-Y) and Cr is red minus luma (R-Y). RGB color space is converted to ycbcr color space, After converting to YCbCr space, the extracted Cb and Cr coefficients are used for cell segmentation process. Conversion from RGB color space to Ycbr is defined by equations 1, 2, 3.

$$Y = 16 + \frac{65.738R}{256} + \frac{129.057G}{256} + \frac{25.064B}{256} \quad (1)$$

$$Cb = 128 - \frac{37.945R}{256} - \frac{74.494G}{256} + \frac{112.439B}{256} \quad (2)$$

$$Cr = 128 + \frac{112.439R}{256} - \frac{94.154G}{256} - \frac{18.285B}{256} \quad (3)$$

5.1.3 Xyz color space

The XYZ colour space has the unique property of being able to express every colour that the human eye can see which in turn means that it can express every colour that can be captured by a camera and hence every colour that anyone might ever want to reproduce in video. X, Y and Z are mathematically generated RGB extrapolations to avoid negative numbers and are called Tristimulus values. Y means luminance, Z is somewhat equal to blue And X is a combination of orthogonal and non-negative cone response curves. RGB color space is converted to xyz color space which are used for segmentation process.

To convert from RGB to XYZ, the equations 4, 5 and 6 are applied consequently.

If $R, G, B \geq 0.04045$,

$$r, g, b = ((R, G, B + 0.055)/1.055)^{(2.4)} \quad (4)$$

otherwise,

$$r, g, b = (R, G, B/12.92) \quad (5)$$

Then, matrix M is used to convert to XYZ.

$$\begin{pmatrix} X \\ Y \end{pmatrix} Z = (M) \begin{pmatrix} r \\ g \end{pmatrix} b \quad (6)$$

5.2 Segmentation

The segmentation process will be divided into two parts:

- Training Model

It passes by 2 stages: Stage one is to convert RGB crops of diseased fish to YCBCR to extract Cb and Cr values to build Gaussian Distribution model. This is defined in equation 7. The values μ and σ are the mean and standard deviation of the *cb* and *cr* values.

$$f(x) = (1/\sigma\sqrt{2\pi})e^{1/2(x - \mu/\sigma)^2} \quad (7)$$

Stage one is shown in fig. 16 However, RGB crops will be converted to gray scale. Gray level different method (GLDM) is then computed. GLDM gets the probability density function for the given image. Then compute PDF1, PDF2, PDF3 and PDF4. Finally compute the mean of PDF1, PDF2, PDF3 and PDF4. Stage 2 is shown in fig. 17

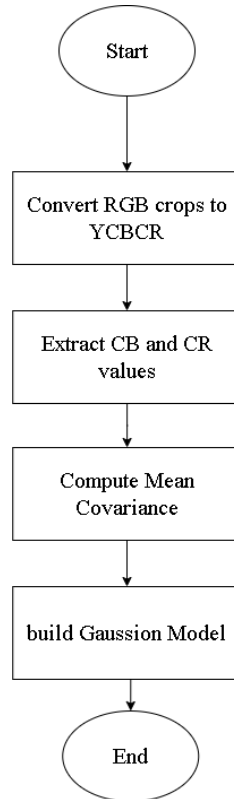


Figure 16: Train Gaussian model based on YCBCR

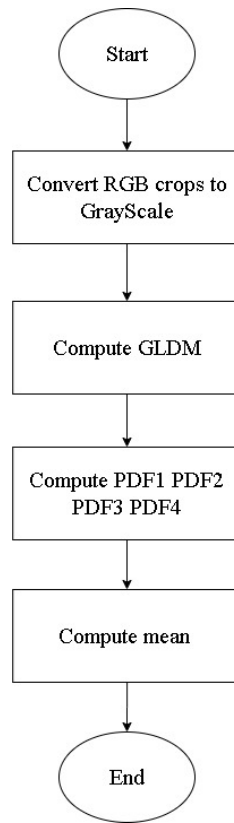


Figure 17: Train PDF computation based on gray level

- Testing model

In testing model, sliding window is applied on RGB images. Then based on stage one in training model. We found that Gaussian model based on XYZ is very effective to segment EUS disease. If this stage ended and the disease is not EUS, then we will start to apply stage 2 after applying sliding window.

Stage 2 passes by applying texture feature based technique. Then based on PDF threshold. We will decide whether the disease is ICH or columnaris. IF yes, It is Ich disease. IF no, it will be columnaris disease. The 2 stages is shown in fig. 19 and fig. 18 respectively.

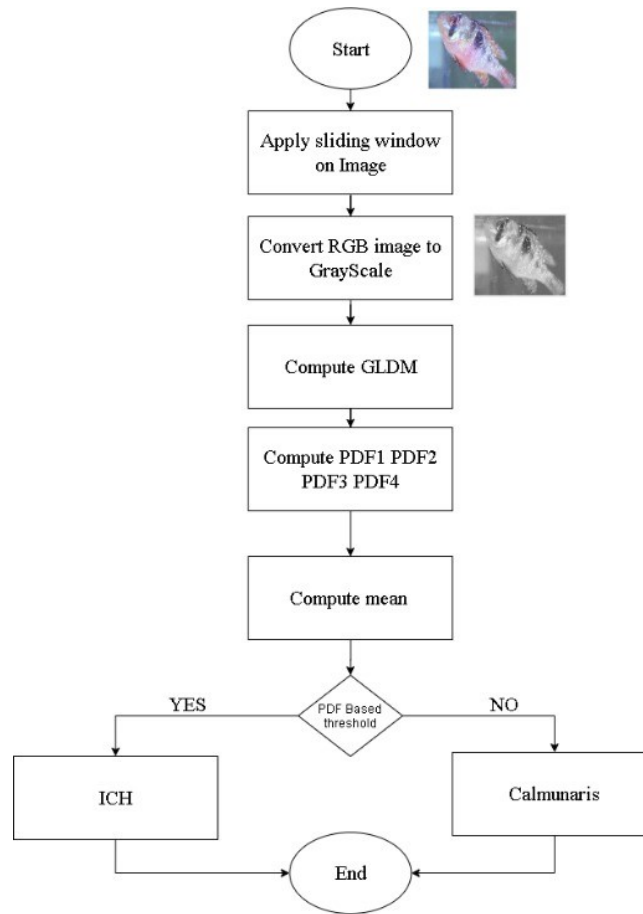


Figure 18: Test PDF computation based on gray level

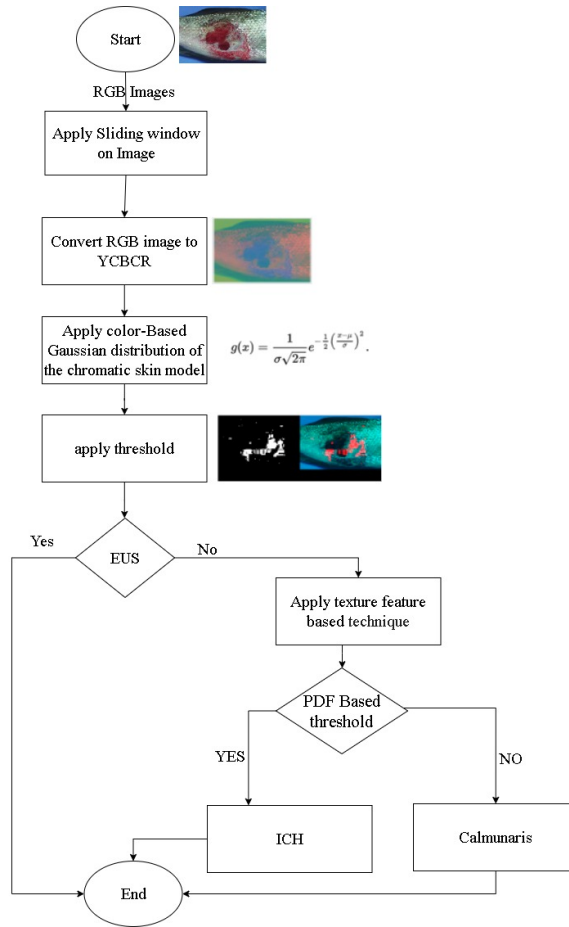


Figure 19: Test Gaussian model based on YCBCR

5.3 CNN Classification

You might know about deep neural network and specially convolution neural network are very effective in building recognition systems [1]. Deep neural network get raw data and features are computed and learned inclusively. CNN can have tens or hundreds of layers that each learn to detect different features of an image. It is composed of an input layer, an output layer, and many hidden layers in between. Three of the most common layers are: convolution, activation and pooling.

5.3.1 Convolutional Layer

Convolutional layers are the major building blocks used in convolutional neural networks. This layer is performed when we extract a segment from the image sized $(n \times n)$ centered at location (x,y) . We then multiply the extracted segment element-by-element with convolution filter also sized $(n \times n)$ and add them all together to create a single output. As shown below in Fig. 20

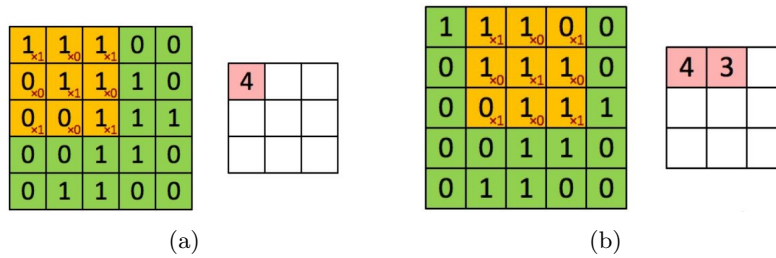


Figure 20: Convolutional Layer

5.3.2 Activation Layer

There are several activation functions each have a different function. Some of popular types of activation functions are Relu , Softmax, linear and Sigmoid. The activation function [1] is a mathematical “gate” in between the input feeding the current neuron and its output going to the next layer as shown in fig 21.

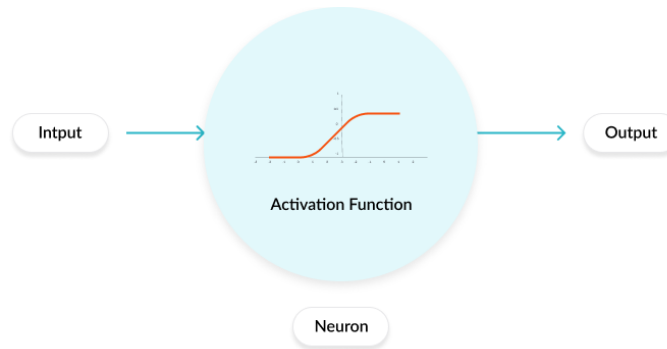


Figure 21: Activation Layer

5.3.3 Max Pooling Layer

The pooling layer is mainly used to reduce the image size (Width and Height) [2] as shown in figure 22

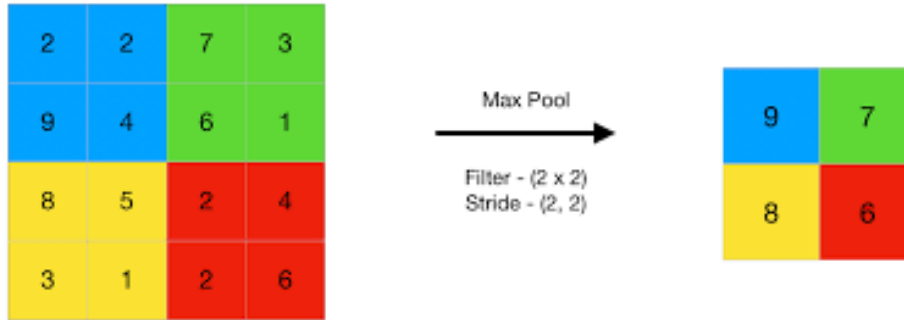


Figure 22: pooling Layer

5.3.4 CNN Architectures

In our experiments, we compared between the performance of the ResNet18, ResNet50, ResNet101, Alex-Net, VGG16, VGG19, mobilenetv2, Xception, Inceptionresnetv2, Shufflenet, Nasnetmobile, Nasnetlarge, Squeezenet, Inceptionv3, Densenet201, Googlenet CNN architectures. Each architecture is trained on 2160 samples, and finally tested on 314 samples. Accuracy is measured at three different learning rates 0.1, 0.01 and 0.001. Experiments were achieved using three different color spaces: RGB, YCBCR and XYZ. Densenet201 and Alexnet achieved highest classification results.

- Densenet201 [3]
DenseNet-201 is a convolutional neural network that is composed of 201 layers deep. The network has an image input size of 224-by-224. DenseNet is composed of Dense blocks. In those blocks, the layers are densely connected together: Each layer receive in input all previous layers output feature maps. A dense block is a group of layers which is connected to all previous layers. Each layer contain: Batch Normalization, ReLU activation and 3x3 Convolution. After a dense block a transition layer is added to proceed to next dense block. Transition layer contain: Batch Normalization, 1x1 Convolution and Average pooling. DenseNet Architecture is shown in Fig. 23

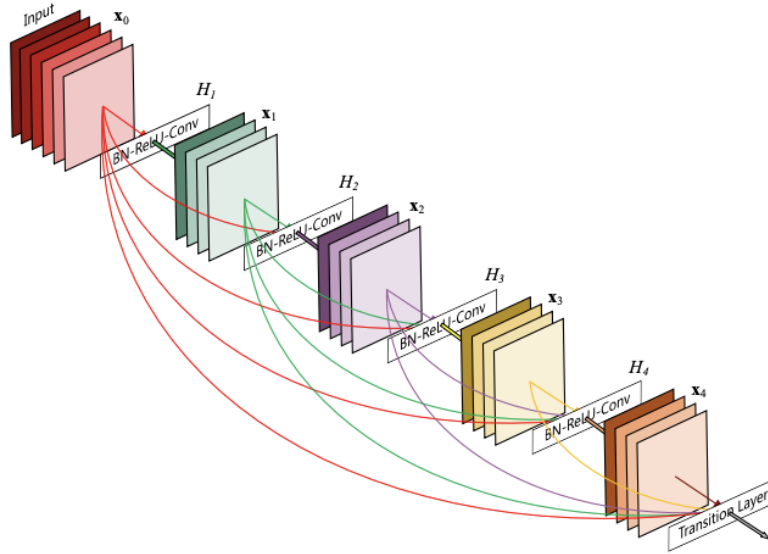


Figure 23: DenseNet CNN Architecture

- Alexnet [4]
 Alexnet CNN is a deep neural network architecture composed of eight layers. The first five layers are convolutional layers (CONV) layers, and the final three layers are fully connected (FC) layers. The input images are normalized to size $227 \times 227 \times 3$, while the output outputs a 3×1 probability vector one corresponding to each disease class. The detailed architecture of the Alexnet CNN is shown in Fig. 24,

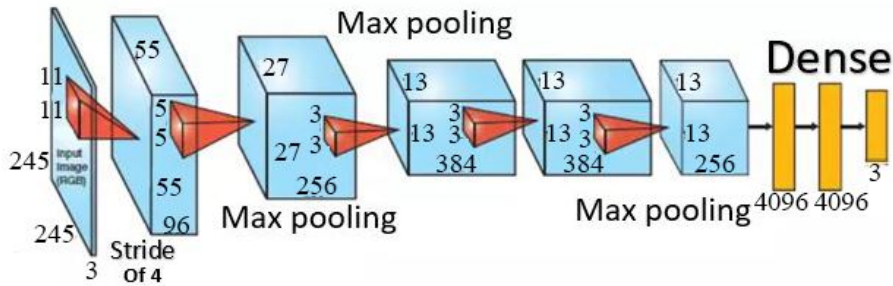


Figure 24: Alex-net CNN Architecture

Table 1: Comparison between different CNN Architectures (ReLU: Rectified linear unit, ILSVRC: ImageNet Large-Scale Visual Recognition Challenge)

Model	Layers	Activation function	Input size	Convolution kernel size	Innovative point
ResNet [5]	18 layers 50 Layers 101 Layers	ReLU	224x224x3	ResNet initial convolution: 7x7. Resnet 50 and 101: 1x1, 3x3 and 1x1. ResNet 18: 3x3	ResNet overcomes degradation and vanishing gradient problems residual blocks that increases the number of hidden layers. The core idea of ResNet is introducing “identity shortcut connection” that skips one or more layers
VGG [6]	16 layers 19 layers	ReLU non-linear	224x224x3	VGG: 3x3	This network uses only 3x3 convolutional layers stacked on top of each other in increasing depth. VGG makes improvement over AlexNet by replacing large kernel-sized
SqueezeNet [7]	18 layers	ReLU	227x227x3	SqueezeNet: 1x1 and 3x3	SqueezeNet goal was to create small neural network with fewer parameters. It was able to achieve a 50X reduction in model size compared to AlexNet. SqueezeNet has advantage of fire module, which uses less filters to decrease number of parameters
DenseNet [3]	201 layers	ReLU	224x224x3	DenseNet: 7x7	In DenseNet, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. DenseNet alleviate the vanishing-gradient problem and reduce the number of parameters
AlexNet [4]	8 layers	ReLU non-linear	227x227x3	Alexnet: 11x11, 5x5 and 1x1	AlexNet has large number of filters to perform the convolution operation of sizes 11x11, 5x5 and 3x3
GoogleNet [8]	22 layers	ReLU	224x224x3 25	GoogleNet: 5X5, 3X3 and 1X1	GoogleNet increases the depth of the network and gain a higher performance level. It is based on the concept of the inception module, it is the collection of convolution and pooling operation performed in a parallel manner so that features can be extracted using different scales

Mobilenetv2 [9]	54 layers	ReLU6	224x224x3	Mobilenetv2: 1x1, 3x3	Mobilenetv2 improves the performance of mobile models on multiple tasks. Mobilenetv2 is based on an inverted residual structure
Xception [10]	71 layers	ReLU	299x299x3	Xception: 1x1, 3x3	Xception involves Depthwise Separable Convolutions, it is supposed to be more efficient than classical convolution in terms of computation time. Xception relies on Shortcuts between Convolution blocks
Inceptionv3 [11]	48 layers	ReLU	299x299x3	Inceptionv3: 3x3	In inceptionv3, computational efficiency and fewer parameters are realized
ShuffleNet [12]		ReLU	224x224x3	Shufflenet: 1x1, 3x3	Shufflenet aim to explore a highly efficient architecture specially designed for limited computing ranges. Shufflenet allows more feature map channels and it is especially critical to the performance of very small networks. ShuffleNet achieves 13x actual speedup over AlexNet while maintaining comparable accuracy

5.4 Video

5.4.1 Detection

- Train a Cascade Object Detector The `vision.CascadeObjectDetector` System object detects objects in images by sliding a window over the image. The detector then uses a cascade classifier to decide whether the window contains the object of interest. Each stage of the classifier labels the region defined by the current location of the sliding window as either positive or negative. Positive indicates that an object was found and negative indicates no objects were found. If the label is negative, the classification of this region is complete, and the detector slides the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive as shown in figure 25.

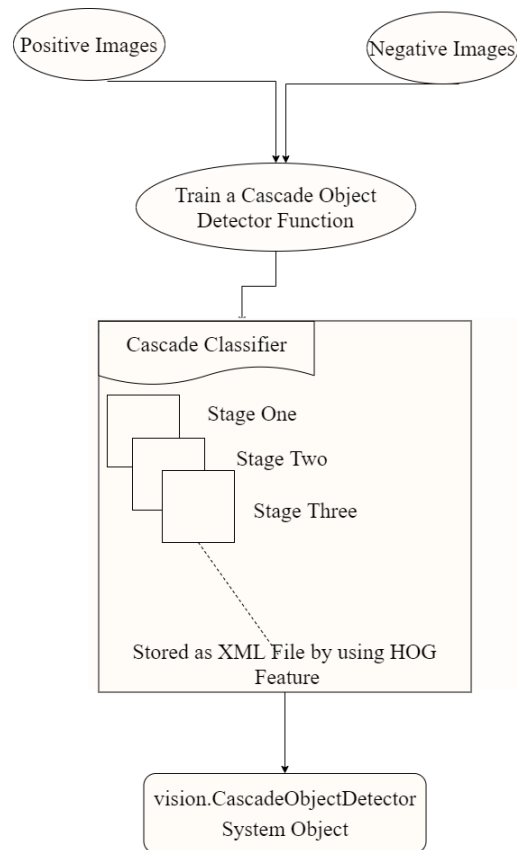


Figure 25: Cascade Object Detector

- Load the positive samples data from file. The file contains a table specifying bounding boxes for several object categories. Select the bounding boxes for Object from the table. Create an imageDatastore object containing negative images. Choose the feature that suits the type of object detection, The HOG features are often used to detect objects. Then run trainCascadeObjectDetector to create xml file as shown in figure 26

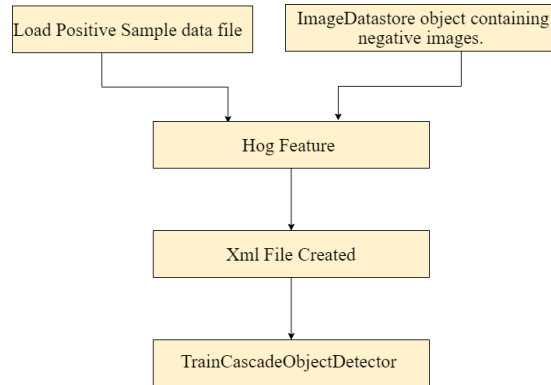


Figure 26: XML Creation

The trainCascadeObjectDetector supports two types of features: Haar, and histograms of oriented gradients (HOG). In our experiment we used HOG features to create our own XML file.

- Histogram of oriented gradients (HOG) is a feature descriptor used to detect objects in computer vision and image processing. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image - detection window, or region of interest (ROI).

The algorithm implementation scheme:

- Divide the image into small connected regions called cells, and for each cell compute a histogram of gradient directions or edge orientations for the pixels within the cell.
- Discretize each cell into angular bins according to the gradient orientation.
- Each cell's pixel contributes weighted gradient to its corresponding angular bin.
- Groups of adjacent cells are considered as spatial regions called blocks. The grouping of cells into a block is the basis for grouping and normalization of histograms.
- Normalized group of histograms represents the block histogram. The set of these block histograms represents the descriptor.

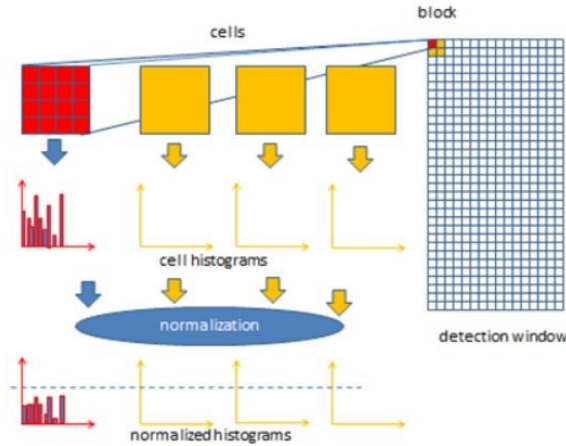


Figure 27: HOG Scheme

Table 2: Different between Feature

Feature	Difference
HOG features	are often used to detect objects such as people and cars. They are useful for capturing the overall shape of an object.
Haar features	are often used to detect faces because they work well for representing fine-scale textures.

5.4.2 Video Tracking scenario

- Calculate Matrix

First, find the coordinates for each bounding box of the image 4 points, starting position(x,y) , Height and width. Second, get the midpoint of the Height and width $x=x+1/2w$ $y=y+1/2H$ Third, get the centroid of the Object. Finally, save the result in Matrix this in every box in frame it saves in matrix .

- Platform

Platform defines a platform object belonging to a tracking scenario. Platforms represent the moving objects in a scenario and are modeled as points or cuboids with aspect-dependent properties. Platform motion, specified as a waypointTrajectory object. The motion defines the time evolution of the position and velocity of the platform frame origin.

- Time corresponding to arrival at each waypoint in seconds, The first

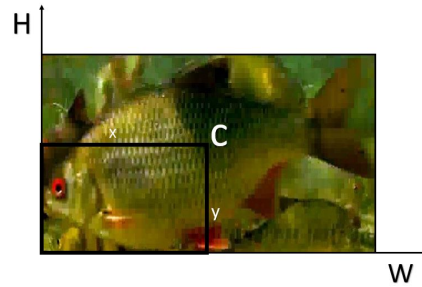


Figure 28: centroid graph

matrix2			
181x2 double			
	1	2	3
1	372	102.5000	
2	1.0745e+03	94	
3	670	199	
4	687	458	
5	458.5000	104.5000	
6	682.5000	197	
7	578	526	
8	288	331.5000	
9	428.5000	105.5000	
10	279	138.5000	
11	719	191	

Figure 29: centroid points

element of TimeOfArrival must be 0.

- Waypoints: Positions in the navigation coordinate system in meters, specified as an N-by-3 matrix.
- Velocity in the navigation coordinate system at each way point in meters per second, specified as an N-by-3 matrix.

1	2		3			
Time	Position		Velocity			
0	-10.0800	202.6500	33.9418	-11.2800	8.8100	-0.0588
1.9608	-21.1200	211.2700	33.8846	-11.2500	8.7900	-0.0577
2.9412	-32.1300	219.8800	33.8287	-11.2100	8.7800	-0.0565
3.9216	-43.1000	228.4900	33.7738	-11.1600	8.7800	-0.0554
4.9020	-54	237.1000	33.7201	-11.0900	8.7900	-0.0543
5.8824	-64.8500	245.7300	33.6674	-11.0300	8.8100	-0.0531
6.8627	-75.6200	254.3800	33.6158	-10.9500	8.8400	-0.0520
7.8431	-86.3100	263.0700	33.5654	-10.8600	8.8800	-0.0510
8.8235	-96.9100	271.8000	33.5159	-10.7600	8.9300	-0.0499
9.8039	-107.4100	280.5800	33.4676	-10.6600	8.9900	-0.0488
10.7843	-117.8100	289.4300	33.4202	-10.5500	9.0500	-0.0478
11.7647	-128.0900	298.3400	33.3739	-10.4200	9.1300	-0.0467
12.7451	-138.2500	307.3200	33.3286	-10.2900	9.2100	-0.0457
13.7255	-148.2700	316.3900	33.2843	-10.1500	9.2900	-0.0447
14.7059	-158.1500	325.5400	33.2410	-10	9.3800	-0.0437
15.6863	-167.8800	334.7900	33.1987	-9.8400	9.4800	-0.0427
16.6667	-177.4500	344.1400	33.1574	-9.6700	9.5900	-0.0417
17.6471	-186.8500	353.5900	33.1170	-9.5000	9.6900	-0.0407
18.6275	-196.0600	363.1500	33.0776	-9.3100	9.8100	-0.0397
19.6078	-205.1000	372.8200	33.0391	-9.1100	9.9200	-0.0388
20.5882	-213.9300	382.6100	33.0016	-8.9100	10.0400	-0.0378
21.5686	-222.5500	392.5100	32.9649	-8.6900	10.1600	-0.0369
22.5490	-230.9600	402.5400	32.9292	-8.4600	10.2800	-0.0360
23.5294	-239.1400	412.6800	32.8944	-8.2300	10.4100	-0.0351
24.5098	-247.0900	422.9400	32.8604	-7.9800	10.5300	-0.0342

Figure 30: Result

5.5 Hardware

This section introduces the interface circuit that connects the reality and computer recognition system as shown in figure 31.

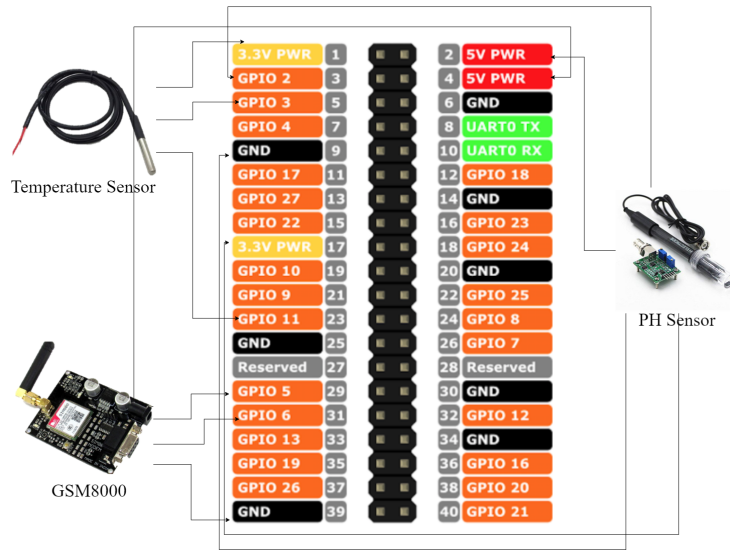


Figure 31: Interface circuit

Interface circuit consists of:

5.5.1 Raspberry pi 3 kit

Raspberry Pi kit is used to get the sensor's measurements and acquired fish captures. It is one full function UART port, and can be configured to two independent serial ports. One SIM card interface. Support Bluetooth function. Raspberry Pi 3 has built in (on board) WiFi and Bluetooth. It is shown in figure 32



Figure 32: Raspberry pi 3

Some Features of Raspberry pi 3:

- CPU: 4× ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
- Ports: HDMI, 3.5mm analogue audio-video jack
- RAM: 1GB LPDDR2 (900 MHz)

5.5.2 Raspberry pi camera

Raspberry pi camera is used to capture fish movements under water. It is a professional camera with high resolution day/ night vision, with fish eye lens for wider field of view and adjustable focal lens. It produced high definition (HD) videos. The camera module is good for both professional and beginner users. It can be used for security applications, slow motion, image recognition and many other. It is shown in figure 33

Some Features of Raspberry pi camera:

- Omnivision OV5647 sensor
- 5MPixel sensor
- Still picture resolution : 2592 x 1944
- Max video resolution : 1080p

- Max frame rate: 30fps



Figure 33: Raspberry pi camera

5.5.3 PH Sensor kit

PH sensor is used to examine water quality in fish farms. PH is a measure of acidity or alkalinity of a solution. The pH scale ranges from 0 to 14. The Ph sensor kit include sensor probe and signal conditioning module. They are connected to each other with BNC connector. PH sensor is shown in figure 34
Some Features of PH sensor:

- Measuring Range: 0 -14
- Measuring Temperature :0-60 °C
- Accuracy : 0.1 pH (25 °C)
- Response Time : 5 sec



Figure 34: PH Sensor

5.5.4 Waterproof Temperature Sensor

Temperature sensors are used to examine water in fish farms. All weather temperature probe, specifically designed to be used in the field. This sealed digital temperature probe lets you precisely measure temperatures in wet environments with an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. It is shown in figure 35

Some Features of Temperature sensor:

- Usable temperature range: -55 to 125°C (-67°F to +257°F)
- $\pm 0.5^\circ\text{C}$ Accuracy from -10°C to $+85^\circ\text{C}$
- Temperature-limit alarm system



Figure 35: Temperature Sensor

5.5.5 GSM-GPRS Module

GSM-GPRS is used to notify owner of fish farm of any improper changes in farm environment or detected infections. Frequency GPRS/GSM Module is an ultra compact and reliable wireless module. It is a breakout board and minimum system of SIM900 Quad-band GSM/GPRS module. It can communicate with controllers via AT commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands). This module supports software power on and reset. The GPRS is configured and controlled via its UART using simple AT commands. Just connect on the Arduino/PIC/AVR/ARM/FPGA/CPLD board, you could easily use AT command control it. The module can be connected to PC via FTDI. It is shown in figure 36

Some Features of GSM-GPRS Module:

- Fully compatible with AVR/PIC/Arduino/ARM/FPGA
- Super capacitor power supply for the RTC

- Not only can use the button for power on , but also can use the digital pin of Arduino to power on and reset the SIM900 module
- Control via AT commands
- Supply voltage range : 3.1 ... 4.8V



Figure 36: GSM-GPRS

5.6 Dataset

Our data-set includes fish photos with the three types of fish diseases, shown in Fig. 37. Our data were collected from different internet resources. Due to limitation of these resources and lack of diseased images, we applied image data augmentation [13] to increase our samples. Augmentation was based on zooming, shearing, rotating, and applying preprocessing function. Our collected data-set include 15 images per disease. After applying data augmentation, samples are increased to 800 images per disease. 550 images were used as training samples per disease, and the remaining samples were used for testing.

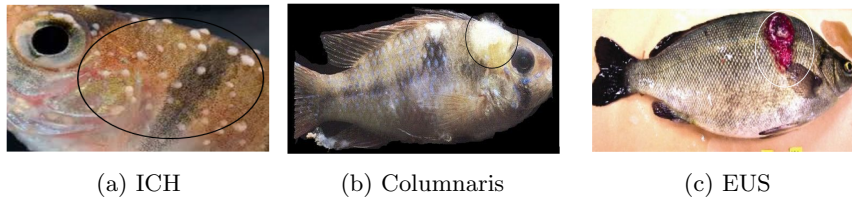


Figure 37: Infectious diseases

6 Human Interface Design

6.1 Overview of User Interface

Our system user interface is very usable and clear. You can login whether you are an admin or user. Once logged in, He/She will be directed to the homepage menu.

- Sign up view: This interface allows new users access the application
- Sign in: This interface is meant for the users to login the application.
- The Home Page is an interface that manages all the menus inside our system.

The Home Page will provide 4 modules which are:

- Edit Profile: This interface allows each user to edit his personal data
- Profile: This interface allows users to show all his personal data
- Report: This interface listing all the sensor daily reading
- Latest-Statistics: This interface listing all the sensor reading

6.2 Screen Images

The image displays two side-by-side screenshots of user interface forms. The left form is titled "Sign Up form" and features a blue background with white text. It includes input fields for "UserName:", "Email:", "Phone:", "Password:", and "Confirm Password:". Below these fields is a white "SUBMIT" button and a link that says "Already have an account? Sign in". The right form is titled "Sign In form" and also has a blue background with white text. It includes input fields for "Email:" and "Password:". Below these fields is a white "SUBMIT" button, a link for "Forget password?", and a link for "Don't have an account? SignUp".

Figure 38: Sign In/Up Screens



Figure 39: User Screen

7 Requirements Matrix

Table 3: Requirements Matrix

ReqID	Reqtype	Requirement Name	Requirement Description	Status	Implemented In
3.1	Required	Register	The user registers with his/her information to create an account	Completed	Android App
3.2	Required	Login	user/Admin can login with his/her username and password into his/her account	Completed	Android App
3.3	Required	Crud User	Cread/Read/Update/Delete user	Completed	Android App
3.4	Required	View Report	User can view report for fish farm details	In progress	Android App
3.5	Required	View statistics	User can view report of latest details for fish farm	In progress	Android App
3.6	Additions	Adding fish farm	The admin adds farm to specific user	In progress	Android App
3.7	Required	Take video	This function is used to take video from camera	Completed	Hardware
3.8	Required	Get PH sensor reading	Get Reading from PH sensor and pass it in Raspberry pi	In progress	Hardware
3.9	Required	Get Temperature sensor reading	Get Reading from Temperature sensor and pass it in Raspberry pi	In progress	Hardware
3.10	Required	Check PH value	This function is for checking PH value is within the normal range for PH or not	In progress	Hardware
3.11	Required	Check temperature value	This function is to check temperature value is within the normal range for temperature or not	In progress	Hardware
3.12	Required	Send message from GSM800	Message will be sent from GSM800 to the user mobile to notify him/her that he/she has improper changes in farm environment or detected infections	In progress	Hardware
3.13	Required	Send Notification	Notification will be sent to the user through mobile application to notify him/her that he/she has improper changes in farm environment or detected infections	In progress	Hardware
3.14	Required	Upload PH sensor readings to firebase	This function is for collecting PH sensor readings from rasp-berry pi and uploading it to firebase	In progress	Hardware

Table 4: Requirements Matrix

ReqID	Reqtype	Requirement Name	Requirement Description	Status	Implemented In
3.15	Required	Upload temperature sensor to firebase	This function is for collecting temperature sensor readings and uploading it to firebase	In progress	Hardware
3.16	Required	Data augmentation	This function is used to increase the diversity of images that is available in data-set for training models	Completed	Pre-processing
3.17	Required	Create XML file This XML	file contain data which is extracted from video in the dataset to be compared with frames taken from video	Completed	Video Detect
3.18	Required	Draw border	This function is used to add border on detected fish	Completed	Video Detect
3.19	Required	Count Number of detected fish	This function is used to count borders of detected fish	Completed	Video Detect
3.20	Required	Video segmentation	This function is used divide video into frames then Convert frames to YCBCR then change values of YCBCR	Completed	segmentation
3.21	Required	Trajectories	This function is used divide videos into two classes normal/Abnormal	Completed	Trajectories
3.22	Required	Velocity and acceleration	This function is used to calculate velocity and acceleration	Completed	Trajectories
3.23	Required	Centroid distance function	This function is used to calculate distance between segmented fish	Completed	Trajectories
3.24	Required	Convert to YCBCR	Converts RGB image to YCBCR	Completed	Segmentation
3.25	Required	Convert to XYZ	Converts RGB image to XYZ	Completed	Segmentation
3.26	Required	Resize image	Image will be resized to cover the part that will be classified	Completed	Pre-processing
3.27	Required	Apply Gaussian distribution	It applies the predefined Gaussian filter on the image	Completed	Segmentation

Table 5: Requirements Matrix

ReqID	Reqtype	Requirement Name	Requirement Description	Status	Implemented In
3.28	Required	Calculate Mean of CB/CR	It Calculates the Mean of CB/CR	Completed	Segmentation
3.29	Required	Calculate Co-variance	It Calculates the co-variance.	Completed	Segmentation
3.30	Required	CNN Classifier	this function is used to train data and integrate with it with features to classify new inputs of images	Completed	Classification
3.31	Required	graycoprops(glcm)	this function exhibits how the pixel brightness in an image contain info of the portions of pixels having similar value	Completed	Texture Feature
3.32	Required	GLDM(inImg, d)	this function that gets probability density function for the given image	Completed	Texture Feature

References

- [1] U. Karn, An intuitive explanation of convolutional neural networks, The data science blog (2016).
- [2] G. Tolas, R. Sire, H. Jégou, Particular object retrieval with integral max-pooling of cnn activations, arXiv preprint arXiv:1511.05879 (2015).
- [3] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
- [4] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.
- [5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
- [6] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
- [7] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, Squeezenet: Alexnet-level accuracy with 50x fewer parameters and; 0.5 mb model size, arXiv preprint arXiv:1602.07360 (2016).

- [8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
- [9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
- [10] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.
- [11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
- [12] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.
- [13] A. Mikołajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in: 2018 international interdisciplinary PhD workshop (IIPhDW), IEEE, 2018, pp. 117–122.