FINAL DELIVERABLE PROJECT DOCUMENTATION

Date	18 November 2022
Team ID	PNT2022TMID00276
Project Name	VirtualEye-Lifeguard for Swimming Pools To Detect the Active Drowning

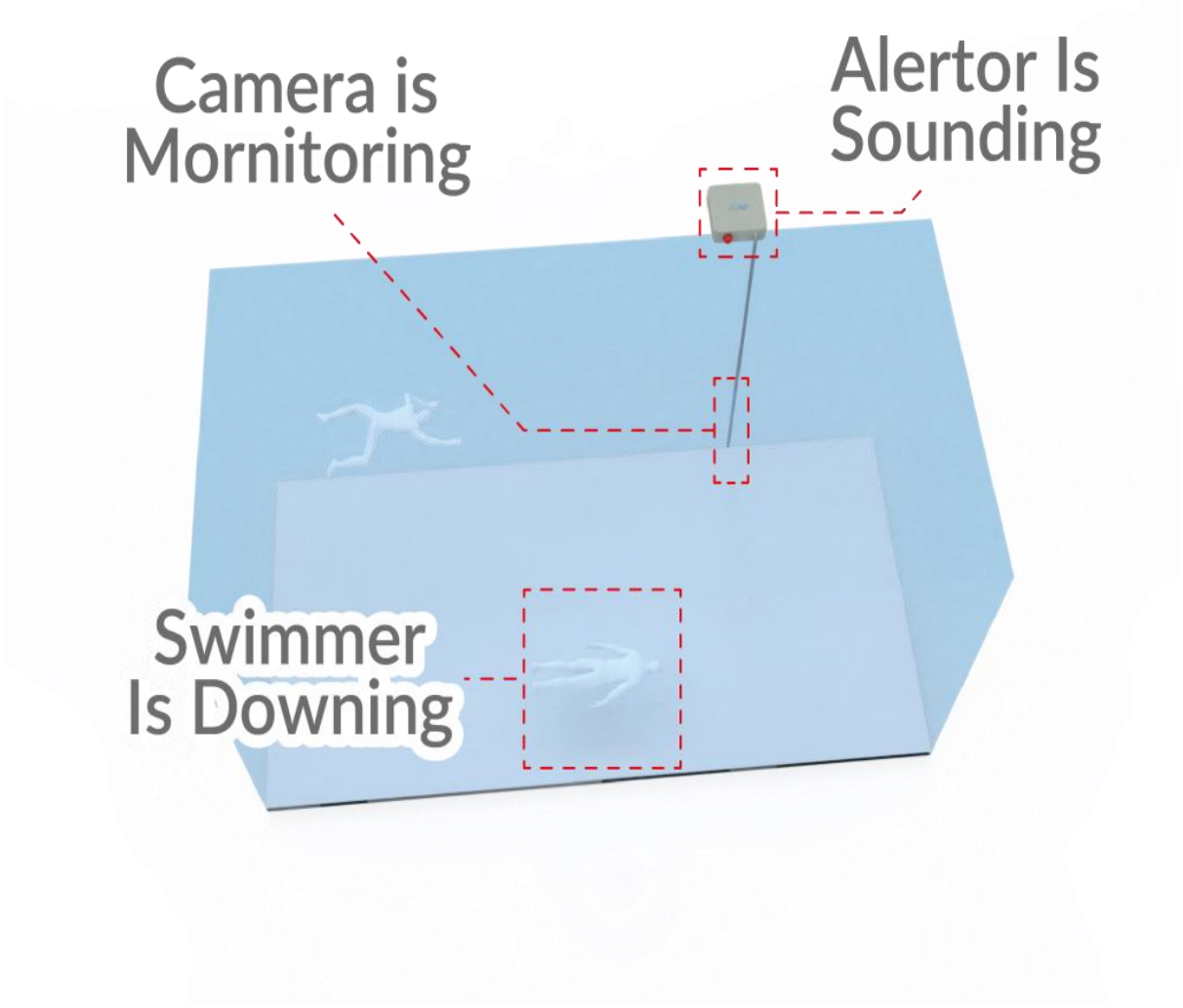
CHAPTER-1

1.INTRODUCTION

Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 1.2 million cases yearly. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analysing the video feeds to detect any anomalies. but AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher than an alert will be generated to attract lifeguards' attention.

1.1 Project Overview



1.2 Purpose

- To advance the techniques used in drowning detection
- To avoid unnecessary deaths in pools due to drowning
- Especially made to decrease the death rate of children under 5.

CHAPTER-2

2. LITERATURE SURVEY

● AUTOMATED DROWNING DETECTION AND SECURITY IN

SWIMMING POOL :

The automated drowning detection system works on the principle of differential pressure. The system contains two fundamental modules to begin with, the wristband consists of pressure sensors on the transmitter side. Second, the receiver module at the swimming pool site. The children entering the pool territory should wear the wristband. The Pressure underwater is different and greater than the pressure at the air - water interface. The pressure at a particular depth is measured and set it as the threshold. Once the child gets into the pool, the pressure is continuously measured and monitored by the microcontroller. When the current value surpasses the threshold limit an alerting signal is sent to the receiver. The wireless transmission and reception of signals is done through an RF module. On receiving the valid signal microcontroller sets the buzzer ON, turns ON the motor driver which in turn lifts the acrylic plate of the multi-floored swimming pool. The kid is brought to the air-water interface, i.e. the top level of the swimming pool by the acrylic plate.

● POSEIDON- VIDEO BASED DROWNING DETECTION SYSTEM IN THE SWIMMING POOL:

Swimming pool drowning monitoring systems based on video technology is mostly reported in the literature. There are three kinds of drowning monitoring systems according to the different positions of the camera. One is that the camera is mounted on the underwater swimming pool wall, then monitors underwater swimmer status. Limitation of this equipment is that if too many swimmers, the occlusion a problem arises. The other is that the camera is mounted upon the water, and monitors the Swimmer posture change. The reflection and refraction of light in air-water interference will affect the image quality,

and drowning man feature this method detected is not easy to distinguish swimmers and divers obviously. The third is a combination of the two, underwater camera and aerial camera matched, monitoring the swimmer's posture. This system needs constant observation which is the main disadvantage.

● DROWNING DETECTION SYSTEM USING CNN:

An alternative for real time downing detection solution for real time developed by using deep learning technology. The method uses convolution neural network object detectors to generate confidence maps of object location in the pool and non maximum suppression to extract head pixel coordinate. Humans detect by various algorithms and detect behaviour of objects that are already set by us.If the swimmer or object gets into difficulty then the system detects it by motion and gets alert and save the swimmer’s life.

● A SMART MULTI -SENSOR DEVICE TO DETECT DISTRESS IN SWIMMERS:

A robust and waterproof sensor-based device to detect distress in swimmers at varying depths and different types of water environments. The proposed device comprises four main components, including heart rate, blood oxygen level, movement, and depth sensors. Although these sensors were designed to work together to boost the system’s capability as an anti-drowning device, each could operate independently. The sensors were able to determine the heart rate to an accuracy of 1 beat per minute (BPM), 1% SpO2, the acceleration with adjustable sensitivities of ±2 g, ±4 g, ±8 g, and ±16 g, and the depth up to 12.8 m. The data obtained from the sensors were sent to a microcontroller that compared the input data to adjustable threshold values to detect dangerous situations. Being in hazardous situations for more than a

specific time activated the alarming system. Based on the comparison made in the program and measuring the time of submersion, a message indicating drowning or safety was sent to a lifeguard to continuously monitor the swimmer' condition via Wi-Fi to an IP address reachable by a mobile phone or laptop.

● DROWNING DETECTION USING AI:

Cameras are placed around the pool, at different points underwater and four in the ceiling, and are connected to a computer and AI server that uses a human pose estimation algorithm and deep learning. The cameras are used firstly to record the coordinates of knees, arms, shoulders and elbows in every swimming position. The pictures are then labelled and fed into the computer to teach it about body parts and movement, a process called annotation. Once the computer has amassed enough data, it can monitor movements in the pool. When the pre-set threshold of a drowning probability is reached – for example, when the computer assesses that the probability of drowning is higher than an adjustable threshold, say 60 percent – an an audio alarm is triggered to attract the lifeguard's attention.

● AUTOMATED VIDEO BASED DROWNING DETECTION USING ACTIVE CONTOURS:

Real time video analysis of the cameras installed around the swimming pool in a way in which the entire swimming The pool can be covered. Each camera is mounted on pool walls oriented downwards with a sharp angle, so that it can minimise the effect of the lighting system which causes occlusions and foreshadowing. In this work, an ODROID-XU as a distributed system is installed in the swimming pool to collect all the video signals collected from cameras and process them using computer vision methods. The

used hardware includes the distributing system known as ODROID-XU.

The system is used to firstly detect the background of the pool and

then decide to send an alarm to rescue team if a previously detected person is

missing in video frames for a specific and defined period of time. In the

next sections of this paper, we try to explain the concepts we used to

detect and track individuals in swimming pools.

2.3 PROBLEM STATEMENT DEFINITION

- A person who cares a lot about privacy tries to continue maintaining their privacy even using a modern swimming pool may worry about drowning detection cameras. Will the person's privacy be guaranteed permanently?
- People looking for children's safe pools may expect added security protection. Can advancement in drowning technique guarantee children's life.

CHAPTER-3

3.IDEATION & PROPOSED SOLUTION

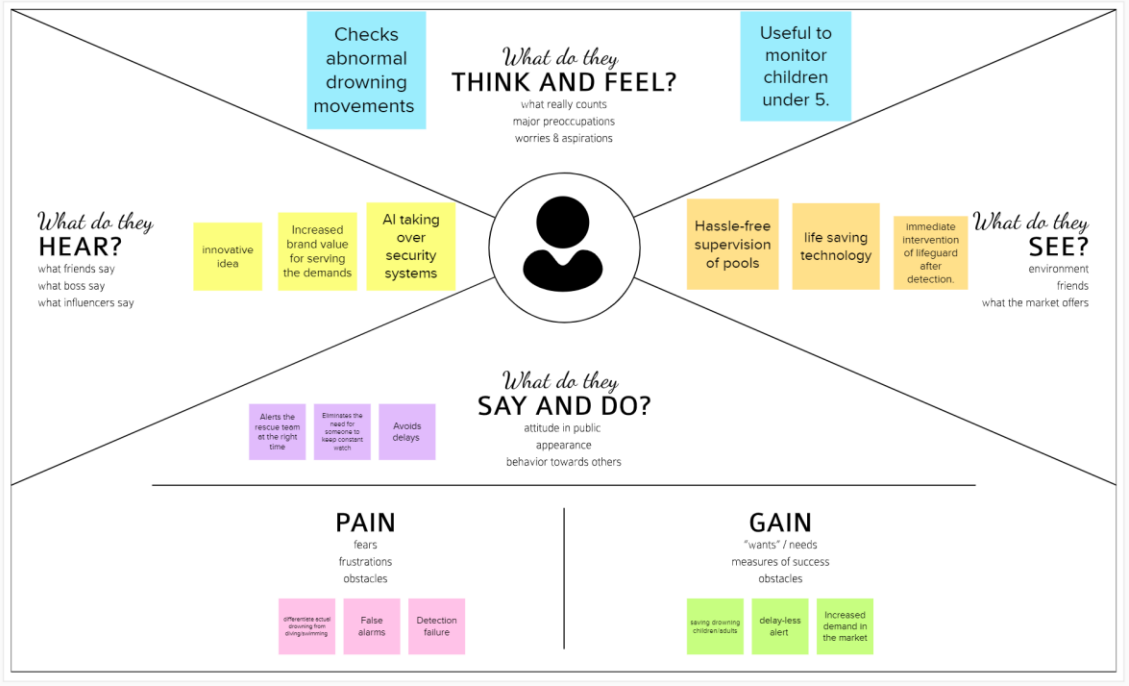
Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

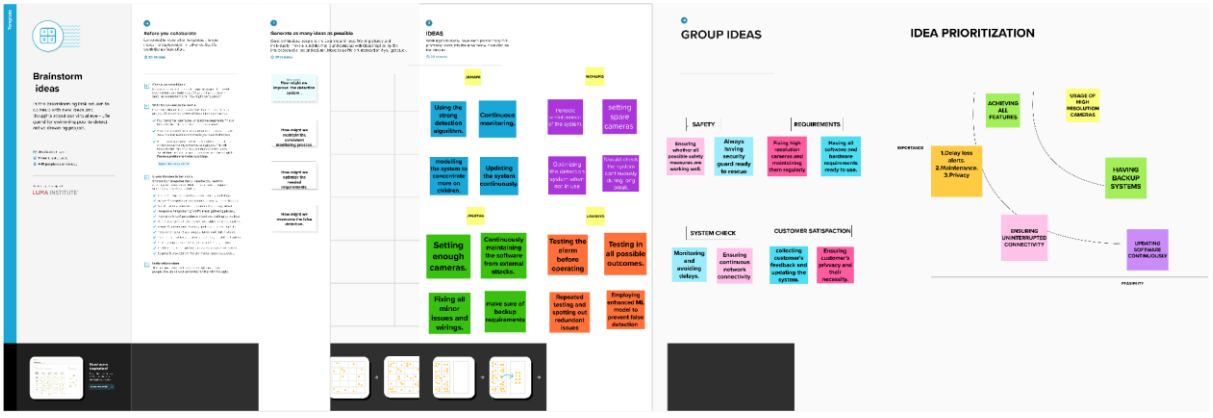
1

Build empathy and keep your focus on the user by putting yourself in their shoes.



Share your feedback

3.2 IDEATION & BRAINSTORMING

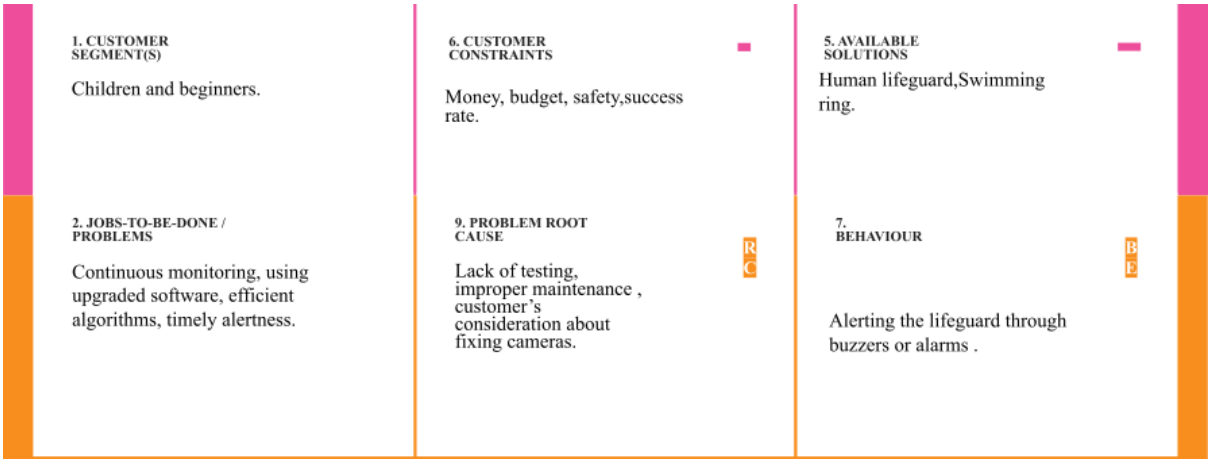


3.3 PROPOSED SOLUTION
Proposed Solution Template:

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	Swimming is one of the best exercises that helps People to reduce stress in this urban lifestyle.Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide
2	Idea / Solution description	By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analysing the video feeds to detect any anomalies. but AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher than an alert will be generated to attract lifeguards' attention.
3	Novelty/Uniqueness	Inorder to detect the drowning person we use a combined software that includes AI,CNN,NLP,YOLO algorithm,Cloudant DB along with some hardware components like high resolution cameras, alarm, triggers etc.
4	**Social Impact / Customer Satisfaction**	Drowning globally has a higher death rate especially among children under the age of six. To overcome this conflict Our drowning detection system will have an impact on society
5	**Business Model (Revenue Model)**	Drowning is an major issue in recent days and it is causing unwanted deaths. So having a fully developed software model for drowning detection will yield high demand and cost.
6	**Scalability of the Solution**	After setting up the software any normal people

		can understand the working procedure. Alarm triggering action is set active all the time and it will be audible to all the people near the pools and private messages are also automatically sent to the lifeguards.

3.4



PROBLEM SOLUTION FIT

CHAPTER-4

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR NO	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR 1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR 2	User Confirmation	Confirmation via Email Confirmation via OTP
FR 3	Payment confirmation	Confirmation via message Confirmation via mail
FR 4	Payment method	Availability of net banking

		Availability of card payment method **Availability of cash payment method** **Allowing on spot payment method**
FR 5	**Software**	**Availability of upgraded software** **Updating software continuously**
FR 6	**Hardware**	**Needed high resolution cameras** **Setting up strong and waterproof wiring**

4.2 NON-FUNCTIONAL REQUIREMENT

NFR NO	Non-Functional Requirement	Description
1	usability	**Usage of understandable methodology .** **Should be useful to all age limits.**
2	security	**Should maintain people's privacy** **Maintaining data carefully**
3	reliability	**Should have a strong enough algorithm to detect unusual movements.**
4	performance	**Should have high speed in alerting the lifeguard.** **Able to send alert messages immediately to the registered workers and relatives.**
5	availability	**Should have an uninterrupted working mechanism.** **Always maintain a backup system,triggers and cameras.**

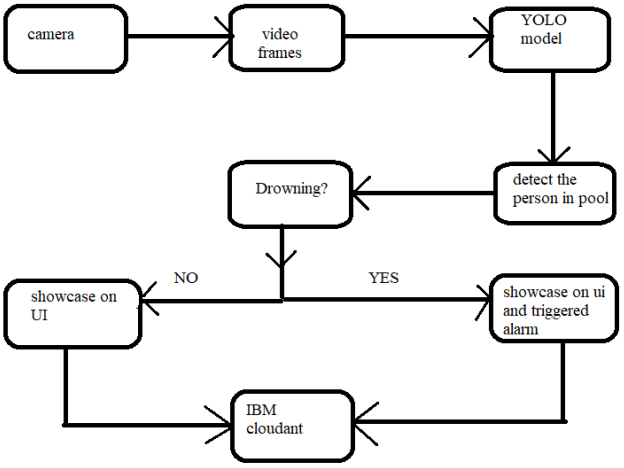
6	scalability	On successful implementation of the software it can be made available in every swimming pool and by making the software more accurate we can also make this as a large scale business model.

CHAPTER-5

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored



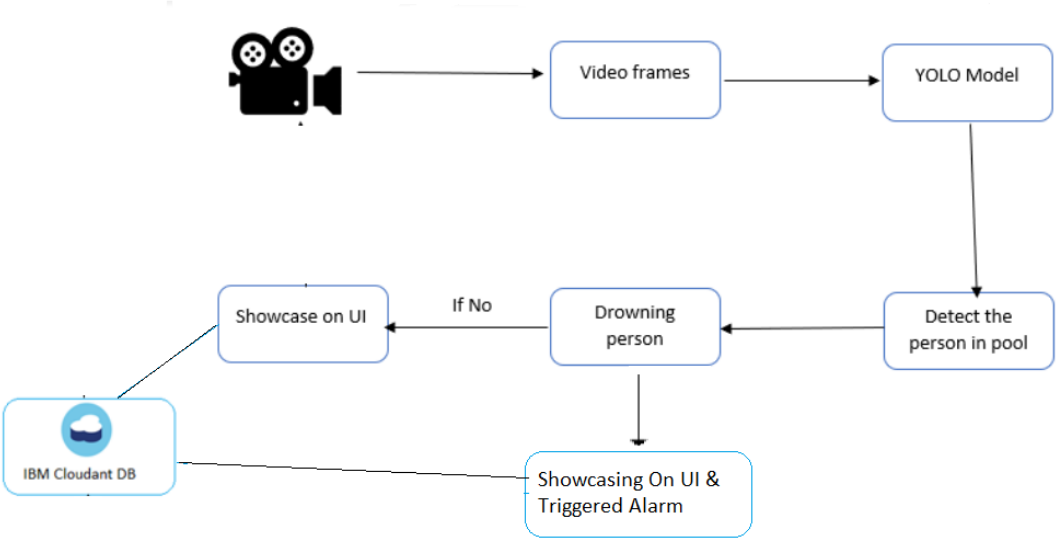
5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture:

To find underwater movement while a person is drowning they have any Problem or anything else we will find the solution using the Artificial Intelligence (AI) detection technology.

Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analysing the video feeds to detect any anomalies.

AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher than an alert will be generated to attract lifeguards' attention



5.3 USER STORIES

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic	User Story Number	Task	Acceptance criteria	Priority	Release
Customer	Installation	USN 1	Install the camera inside the underwater, connect necessary app in the phone or other device	I can cameras to the IBM cloud DB	High	Sprint1
Lifeguard	Secure the people	USN 2	As a user, I can secure the drowning person from the pool	I can save the drowning person	High	Sprint1
swimmers	safety	USN 3	As a user, I can swim inside the underwater without fear of the Drowning	I can swim safely	medium	Sprint2
Executive	Contact	USN 4	As a user, I Can resolve if any problem occurs with any device technically	I can contact the customer care executive to resolve any issues	medium	Sprint3
Administrator	Dashboard	USN 5	Management of the	I can access the system's	High	Sprint4

			drowning detection system and database managemen t	logs and any other data instantly		

CHAPTER-6

6.1 SPRINT PLANNING & ESTIMATION

Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	8	8 Days	24 Oct 2022	31 Oct 2022	6	31 Oct 2022
Sprint-2	14	6 Days	31 Oct 2022	05 Nov 2022	12	05 Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	11	12 Nov 2022
Sprint-4	13	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

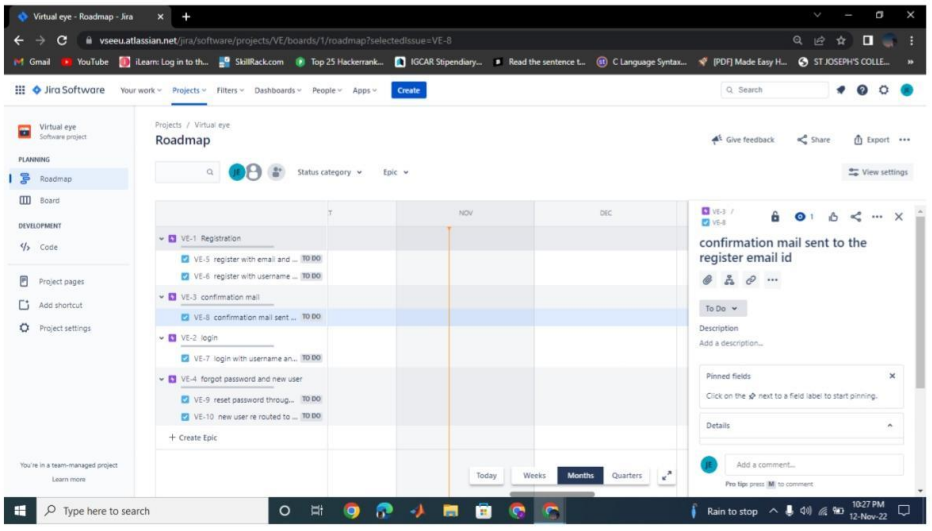
6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	VE-1	As a user, I can register for the application by entering my email password.	2	High	Janani ,Mohanapriya,Jegatha,Lavanya
Sprint-1	Registration	VE-2	As a user, I can register for the application with my username.	1	High	Janani, Mohanapriya,Jegatha,Lavanya
Sprint-1	Confirmation	VE-3	As a user, I will receive confirmation email once I have registered for the application	2	Low	Janani ,Jegatha,Mohanapriya,Lavanya
Sprint-1	Login	VE- 4	As a user, I can log into the application by entering email & password	2	Medium	Janani ,Lavanya,Mohanapriya,Jegatha
Sprint-1	Re registration	VE-6	As a user, I can re- register as new user if i forgot my password	1	High	Janani ,Mohanapriya,Jegatha,Lavanya
Sprint-2	Dataset Collect	VE-11	Collect number of datasets and get accuracy	2	Medium	Janani ,Jegatha,Mohanapriya,Lavanya
Sprint-2	Pre-processing	VE-12	The dataset is extracted	2	High	Janani ,Mohanapriya,Jegatha,Lavanya
Sprint-2	Train the model	VE-13	Train the model.	5	High	Janani ,Lavanya,Mohanapriya,Jegatha

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Test the model	VE-14	Test the model	5	High	Janani ,Mohanapriya,Jegatha,Lavanya
Sprint-3	Detection	VE-15	Load the trained model.	3	High	Janani ,Lavanya , Mohanapriya,Jegatha
Sprint-3	Detection	VE-16	Identify the person by collecting real-time data through a webcam.	5	Medium	Janani ,Mohanapriya,Jegatha,Lavanya
Sprint-3	Detection	VE-16	classify it by using a trained model to predict the output	8	High	Janani ,Jegatha, Mohanapriya,Lavanya
Sprint-4	Detection	VE-17	If person is drowning, the system will ring an alarm to give signal	8	High	Janani ,Jegatha, Mohanapriya,Lavanya
Sprint-4	Detection	VE-18	As a User,I can detect the drowning person.	3	Medium	Janani ,Lavanya,Mohanapriya,Jegatha
Sprint-4	Logout	VE-19	As a User,I can logout the application.	2	Low	Janani ,Mohanapriya,Jegatha,Lavanya

6.3 REPORT FROM JIRA

Roadmap



CHAPTER-7

7. CODING & SOLUTION

7.1 FEATURE 1

```
[net]
# Testing#
batch=1 #
subdivisions=
1# Training
batch=64
```

subdivisions=
16 width=608
height=608
channels=3
momentum=0.
9
decay=0.0005
angle=0
saturation =
1.5
exposure =
1.5hue=.1

learning_rate=0.01
burn_in=1000
max_batches =
500200policy=steps
steps=400000,45000
0 scales=.1,.1

[convolution
al]
batch_norm
alize=1
filters=32
size=3
stride=1
pad=1
activation=le
aky

#
Downsample

[convolution
al]
batch_norm
alize=1
filters=64
size=3
stride=2
pad=1
activation=le
aky

[convolutiona
l]
batch_norma
lize=1
filters=32
size=1
stride=1
pad=1
activation=le
aky

[convolutiona
l]
batch_norma
lize=1
filters=64
size=3
stride=1
pad=1
activation
=leaky
[shortcut]fr
om=- 3
activation
=linear#

Downsample

[convolution
al]
batch_norm
alize=1
filters=128
size=3
stride=2
pad=1
activation=l
eaky

[convolutiona
l]
batch_norma
lize=1
filters=64
size=1

stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=128
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=64
size=1
stride=1
pad=1
activation=le
aky

[convolutiona
l]
batch_norma
lize=1
filters=128
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

#
Downsample

[convolutiona
l]
batch_norma
lize=1
filters=256siz
e=3 stride=2
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

```
[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolutiona
l]
batch_norma
lize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolutiona
l]
batch_norma
lize=1
filters=256
size=3
stride=1
```

pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolutiona
l]
batch_norma
lize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1

activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolutiona
l]
batch_norma
lize=1
filters=128
size=1
stride=1
pad=1
activation
=leaky

[convolutiona
l]
batch_norma
lize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3

activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm

alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near#

Downsample

[convolutiona
l]
batch_norma
lize=1
filters=512
size=3
stride=2
pad=1 activation=leaky

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=l
eaky

[convolutiona
l]
batch_norma
lize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

```
[shortcut]fro
m=- 3
activation=li
near


[convolutiona
l]
batch_norma
lize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutiona
l]
batch_norma
lize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]fro
m=- 3
activation=li
near


[convolutiona
l]
batch_norma
lize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutiona
l]
```

batch_norma
lize=1
filters=512
size=3
stride=1
pad=1 activation=leaky

[shortcut]fro
m=- 3
activation=li
near

[convolutiona
l]
batch_norma
lize=1
filters=256
size=1
stride=1
pad=1
activation
=leaky

[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=l
eaky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=256

size=1
stride=1
pad=1
activation=l
eaky

[convolutiona
l]
batch_norma
lize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]fro
m=- 3
activation=li
near

[convolutiona
l]
batch_norma
lize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512 size=3
stride=1 pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

```
[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=l
eaky

[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolutiona
l]
batch_norma
lize=1
filters=512
size=3
```

stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near#

Downsample

[convolutiona
l]
batch_norma
lize=1
filters=1024
size=3
stride=2
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=1024
size=3
stride=1
pad=1
activation=l
eaky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=1024
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=le
aky

```
[convolution
al]
batch_norm
alize=1
filters=1024
size=3
stride=1
pad=1
activation=l
eaky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=l
eaky

[convolution
al]
batch_norm
alize=1
filters=1024
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

########################
```

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize=
1size=3 stride=1
pad=1
filters=1024
activation=leaky

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=l
eaky

[convolutional]
batch_normalize=
1size=3 stride=1
pad=1
filters=1024
activation=leaky

[convolutiona
l]
batch_norma
lize=1
filters=512
size=1
stride=1
pad=1

activation=le
aky

[convolutional]
batch_normalize=
1size=3 stride=1
pad=1
filters=1024
activation=le
aky

[convolutional
]size=1
stride=1
pad=1
filters=255
activation=lin
ear

[yolo]
mask =
6,7,8
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198,
373,326
classes=80
num=9 jitter=.3
ignore_thresh =
.7 truth_thresh =
1random=1

[route] layers
= -4

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

```
[upsample]
stride=2

[route]
layers = -1,
61



[convolutiona
l]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize
=1size=3
stride=1 pad=1
filters=512
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize
=1size=3
stride=1 pad=1
filters=512
```

activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize
=1size=3
stride=1 pad=1
filters=512
activation=le
aky

[convolutional
]size=1
stride=1
pad=1
filters=255
activation=lin
ear

[yolo]
mask =
3,4,5
anchors
=
10,13,
16,30,
33,23,
30,61,
62,45,
59,119,
116,90,
156,198,
373,326

classes=80
num=9 jitter=.3
ignore_thresh =
.7 truth_thresh =
1random=1

[route] layers
= -4

[convolutiona
l]
batch_norma
lize=1
filters=128
size=1
stride=1
pad=1
activation
=leaky

[upsample]
stride=2

[route]
layers = -1,
36

[convolutiona
l]
batch_norma
lize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize
=1size=3
stride=1 pad=1

```
filters=256
activation=leaky

[convolutiona
l]
batch_norma
lize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize=1siz
e=3 stride=1
 pad=1
 filters=256
 activation=l
 eaky

 [convolution
 al]
 batch_norm
 alize=1
 filters=128
 size=1
 stride=1
 pad=1
 activation=l
 eaky

 [convolutional]
 batch_normalize
 =1size=3
 stride=1 pad=1
 filters=256
 activation=leaky

 [convolutional
 ]size=1
 stride=1
```

```
pad=1
filters=255
activation=li
near


[yolo]
mask =
0,1,2
anchors = 10,13, 16,30, 33,23,  30,61,  62,45,  59,119,  116,90,
156,198,
373,326
classes=8
0    num=9
jitter=.3
ignore_thr
esh = .7
truth_thresh = 1 random=1
```

7.2 FEATURE 2

```
#import       necessary

packagesimport cv2

import os
import  numpy  as
np

from .utils import download_file



initialize = Truenet

= None

dest_dir = os.path.expanduser('~') + os.path.sep + '.cvlib' + os.path.sep
+ 'object_detection' +os.path.sep + 'yolo' + os.path.sep + 'yolov3'

classes = None

#colors are BGR instead of RGB in python

COLORS = [0,0,255], [255,0,0]


def populate_class_labels():
```

```
    #we are using a pre existent classifier which is more reliable and
    more efficient than one#we could make using only a laptop
    #The classifier should be downloaded automatically when you
    run this scriptclass_file_name =
    'yolov3_classes.txt'
    class_file_abs_path = dest_dir + os.path.sep + class_file_name

    url = 'https://github.com/Nico31415/Drowning-
    Detector/raw/master/yolov3.txt'if not
    os.path.exists(class_file_abs_path):
      download_file(url=url, file_name=class_file_name,
      dest_dir=dest_dir)f =
    open(class_file_abs_path, 'r')
    classes = [line.strip() for line in f.readlines()]


    return classes



  def get_output_layers(net)
 #the number of output layers in a neural network is the
 number of possible#things the network can detect, such
 as a person, a dog, a tie, a phone... layer_names =
 net.getLayerNames()


    output_layers = [layer_names[i[0] - 1] for i in
    net.getUnconnectedOutLayers()]

return output_layers
```

```
        def draw_bbox(img, bbox, labels, confidence, Drowning,
        write_conf=False):


          global
          COLORS
          global
          classes


          if classes is None:
classes                          =
            populate_class_la
            bels()


          for i, label in enumerate(labels):


            #if the person is drowning, the box will be
            drawn red instead of blueif label == 'person'
            and Drowning: color = COLORS[0] label
              = 'DROWNING'
else:
color = COLORS[1]

if write_conf:

              label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'
            #you only need to points (the opposite corners) to draw a
            rectangle. These points#are stored in the variable bbox
            cv2.rectangle(img, (bbox[i][0],bbox[i][1]), (bbox[i][2],bbox[i][3]),
            color, 2)

cv2.putText(img,          label,          (bbox[i][0],bbox[i][1]-10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
```

```
    return img

def detect_common_objects(image, confidence=0.5,
                          nms_thresh=0.3):

    Height, Width = image.shape[:2]scale =
    0.00392

    global
    classes
    global
    dest_dir

    #all the weights and the neural network algorithm are already
    preconfigured#as we are using
    YOLO

    #this part of the script just downloads the
    YOLO files config_file_name = 'yolov3.cfg'
    config_file_abs_path = dest_dir + os.path.sep +
    config_file_name

    weights_file_name = 'yolov3.weights'
                weights_file_abs_path = dest_dir + os.path.sep +
                weights_file_name

    url = 'https://github.com/Nico31415/Drowning-
    Detector/raw/master/yolov3.cfg'
```

```
        if not os.path.exists(config_file_abs_path):
          download_file(url=url,                file_name=config_file_name,
          dest_dir=dest_dir)
        url = 'https://pjreddie.com/media/files/yolov3.weights'

if                                  not
        os.path.exists(weights_fil
        e_abs_path):

          download_file(url=url,              file_name=weights_file_name,
          dest_dir=dest_dir)




        global

        initialize

        global net


        if initialize:
classes                         =
          populate_class_la
          bels()

              net = cv2.dnn.readNet(weights_file_abs_path,
                        config_file_abs_path)initialize = False

        blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0),
        True, crop=False)

net.setInput(blob)

outs                                =
        net.forward(get_output_l
        ayers(net))
```

```
class_ids
=          []
confiden
ces  =  []
boxes  =
[]

for out in outs:
  for detection in out:
    scores              =
    detection[5:]
    class_id            =
    np.argmax(score
    s)   max_conf   =
    scores[class_id] if
    max_conf          >
    confidence:
```

```
center_x = int(detection[0] * Width) center_y = int(detection[1] * Height)w = int(detection[2] * Width) h = int(detection[3] * Height)x = center_x - w / 2 y = center_y - h / 2 class_ids.append(class_id) confidences.append(float(max _conf))boxes.append([x, y, w, h])
```

```
indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence, nms_thresh)
```

```
bbox = []label = []conf = []
```

```
for i in indices:
  i = i[0]
  box = boxes[i]x = box[0] y = box[1] w =
```

box[2]h =

box[3]

bbox.append([round(x), round(y), round(x+w), round(y+h)]) label.append(str(classes[class_ids[i]]))

conf.append(confidences[i])

return bbox, label, conf

CHAPTER-8

8.TESTING

8.1 TEST CASES

Test case ID	Feature Type		Test Scenario	Steps TO Execute	Test	Expected Result	Actual Result
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	I.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	Login.html	Login/Signup popup should display	Working as
LoginPage_TC_002		Home Page	Verify the UI elements in Login/Signup popup	I.Enter tJRL and dick go 2.Click on My Account dropdown 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c. L%in button d.New customer? Create account link e. Last password? Recovery password link	Login.html	Application should show below elements: a.email text box b.password text box c.Login button with orange colour d. New custotner? Create account link e.Last password? Recovery password link	Working as expected
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	I.Enter URL and dick go 2.Click on My Account dropdown 3.Enter Valid username/email in Email text 4.Enter valid password in password text box 5. Click On in button	Username:lax@gmail password: lax26	User should navigate to prediction homepage	working as
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with Invalid credentials	1, Enter URL and click go 2.Click on My Account dropdownbutton 3.Enter Invalid username/email in Email text box 4.Enter valid password in password text box 5.Click on •n button	Username:lax password:lax26	Application should show 'Incorrect email or password ' validation message.	working as
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	I-Enter URL and click go 2.Click On My Account dropdown 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on In button	username:lax26@mail password:lax26	Application should show •Incorrect email or password ' validation message.	working as
LoginPage_TC_OO5	Functional	Login page	Verify user is able to into application with InValid credentials	I.Enter URL and click go 2.Click on My Account dropdown 3.Enter Invalid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on I in button	username:lax26@mail password:1803	Application should show 'Incorrect email or password ' validation message.	working as
Predictionpage_TC_ 00 6	Functional	Prediction Page	Page should display whether the person is drowning or not	1, Camera should take pictures of people swimming in pools 2. It should predict the probability of drowning 3. It should show a bounding box displaying the probability Of drowning	image Of people drowning	generate a alert to lifeguard if people are drowning	Working as

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and howthey were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	3	1	2	13
Duplicate	1	0	2	0	3
External	2	3	0	1	6
Fixed	10	2	4	10	26
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3.Test Case Analysis

This Report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass

Print Engine	7	0	0	7
Client Application	1	0	0	41
Security	42	0	0	42
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

CHAPTER-9

9.RESULT

9.1 PERFORMANCE METRICS

Home.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Home</title>
    <link rel="stylesheet" type="text/css" href="..\static\style.css">
</head>

<body>
    <nav class="navbar">
        <div class="logo">Virtual Eye</div>
        <ul class="nav-links">
            <div class="menu">
                <li><a href="index.html">Home</a></li>
```

```
                <li><a href="register.html" target="blank">Register</a></li>
                <li><a href="login.html" target="blank">Login</a></li>
            </div>
        </ul>
    </nav>
    <div class="row">
        <div class="column">
            <img class="image" src="index.jpeg"><br><br>
        </div>
        <div class="column log">
            <p class="names">TEAM ID : PNT2022TMID00276</p>
            <p class="names">Team Member 1 : Janani T </p>
            <p class="names">Team Member 2 : Mohanapriya D </p>
            <p class="names">Team Member 3 : Jegatha B </p>
            <p class="names">Team Member 4 : Lavanya S </p>
        </div>
    </div>

</body>

</html>
```

Register.html:

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Register Form</title>
    <link rel="stylesheet" type="text/css" href="..\static\style.css">
</head>

<body>
    <nav class="navbar">
        <!-- LOGO -->
        <div class="logo">Virtual Eye</div>
        <!-- NAVIGATION MENU -->
        <ul class="nav-links">
            <!-- USING CHECKBOX HACK -->
            <div class="menu">
                <li><a href="index.html">Home</a></li>
                <li><a href="register.html" target="blank">Register</a></li>
                <li><a href="login.html" target="blank">Login</a></li>
```

```
            </div>
        </ul>
    </nav>
    <div class="row">
        <div class="column">
            <img class="image" src="index.jpeg" <h1>Login</h1>
        </div>
        <div class="registration-form">
            <!-- <h4>Register</h4> -->
            <form action="#" method="post">
                <p class="names">Full Name:</p>
                <input type="text" name="fullname" placeholder="Enter Full
Name" required>
                <p class="names">Email ID:</p>
                <input type="email" name="email" placeholder="Enter Email ID"
required>
                <p class="names">Password:</p>
                <input type="password" name="password" placeholder="Enter
Password" required>
                <button type="submit"> Register </button>

            </form>
        </div>
    </div>
</body>

</html>
```

Login.html:

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Login Form</title>
    <link rel="stylesheet" type="text/css" href="..\static\style.css">
</head>

<body>
    <nav class="navbar">
        <!-- LOGO -->
        <div class="logo">Virtual Eye</div>
        <!-- NAVIGATION MENU -->
```

```
        <ul class="nav-links">
            <!-- USING CHECKBOX HACK -->
            <div class="menu">
                <li><a href="index.html">Home</a></li>
                </li>
                <li><a href="register.html" target="blank">Register</a></li>
                <li><a href="login.html" target="blank">Login</a></li>
                <li><a href="prediction.html"
target="blank">Prediction</a></li>
            </div>
        </ul>
    </nav>
    <div class="row">
        <div class="column">
            <img class="image" src="index.jpeg" <h1>Login</h1>
        </div>
        <div class="column log">
            <form action=" # " method="post ">
                <p class="names">Registered Email ID:</p>
                <input class="input" type="email " placeholder="Enter
registered email ID " name="email " required>
                <p class="names">Password:</p>
                <input class="input" type="password " name="password "
placeholder="Enter Password " required>
                <button class="button-login" type="submit ">Login</button>
                <br>
                <a class="mr" href="register.html">New User?</a>
                <a class="ml" href="">Forgot Password</a>
            </form>
        </div>
    </div>
</body>

</html>
```

Prediction.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial- scale=1.0">
    <!--Bootstrap -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo tstrap.min.css"
```

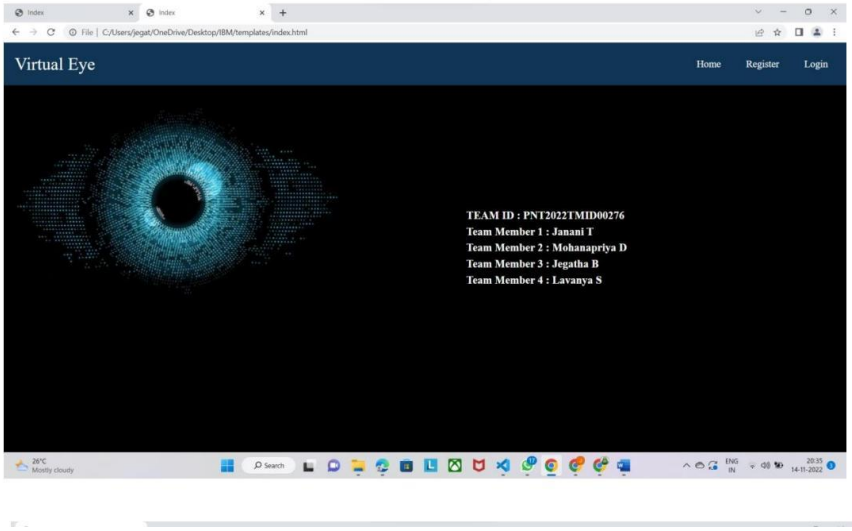
```
integrity="sha384- Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGg
FAW/dAiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpG FF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/
popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk vXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootst
rap.min.js" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5
+76PVCmYl" crossorigin="anonymous"></script>
    <script src="https://kit.fontawesome.com/8b9cdc2059.js"
crossorigin="anonymous"></script>
    <link href="https://fonts.googleapis.com/css2?family=Akronim&family=
Roboto&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="../static/style.css">
    <script defer src="../static/js/JScript.js"></script>
    <title>Prediction</title>
</head>

<body>
    <nav class="navbar">
        <div class="logo">Virtual Eye</div>
        <ul class="nav-links">
            <div class="menu">
                <li><a href="index.html">Home</a></li>
                <li><a href="logout.html" target="blank">Logout</a></li>
            </div>
        </ul>
    </nav>
    <div class="row">
        <div class="column">
            <img class="image" src="index.jpeg"><br><br>
        </div>
        <div class="column log">
            <h2 class="names">Drowning Detection</h2>
            <p class="names">Using Facial Recognition</p>
            <br>
            <br>
            <p class="names">Upload the image below for detection</p>
            <form action="/action_page.php">
                <label for="img">Select image:</label>
```

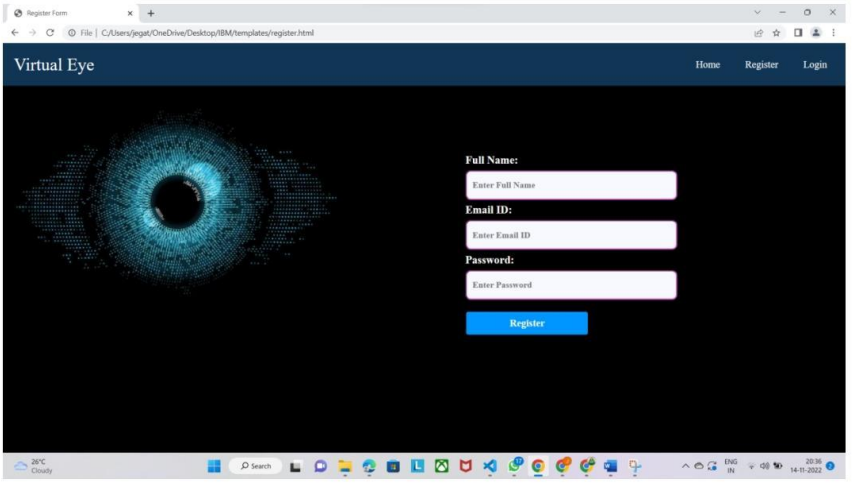
```
                <input type="file" id="img" name="img" accept="image/*">
                <button type="submit"> Submit </button>
            </form>

        </div>
    </div>
</body>

</html>
```

Logout.html:

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial- scale=1">
    <title>Logout</title>
    <link rel="stylesheet" type="text/css" href="..\static\style.css">

</head>

<nav class="navbar">
    <div class="logo">Virtual Eye</div>
    <ul class="nav-links">
        <div class="menu">
            <li><a href="index.html">Home</a></li>
            <li><a href="login.html" target="blank">Login</a></li>
        </div>
    </ul>
</nav>
<div class="row">
    <div class="column">
        <img class="image" src="index.jpeg">
    </div>
    <div class="column logout">
        <h1 class="names">Successfully Logged Out!</h1>
        <br>
        <br>
        <h2 style="color:white">Login again for more information</h2>
        <!-- <a href="login.html"></a><button type="submit"> Register </button>
-->
        </form>
    </div>
```

```
</div>
</body>

</html>
```





CHAPTER-10

10. ADVANTAGES & DISADVANTAGES *
ADVANTAGES:

(i) user feel comfortable and more secure

(ii) Children, adult, pet animal , old age people are used

(iii) spending more time for family, freedom for safety guards near the Swimming pool

(iv) Swimmers, resort are gain in the financial (v) drowning should be monitored

*DISADVANTAGE:

(i) For uneducated people will suffer from this technology

(ii) Electricity will be required

(iii) Software and hardware requirement will need

CHAPTER-11

11. CONCLUSION

CHAPTER-12

FUTURE SCOPE:

This lifeguard system consists of three main components, i.e., the drowning detection, the rescuing drone, and the hazardous activity detection. All three components combined will create a system capable of detecting drowning victims, dispatching an inflatable tube using a drone (as depicted in Fig.9) and detecting hazardous activities—eventually becoming an entity that could assist a lifeguard. The system is accessible to its primary user, presumably a pool owner or a lifeguard, in the form of an interface with a sound alarm and an android mobile service that holds the capabilities of receiving Firebase notifications. Confined with a few of the hardware limitations, such as the use of a single camera and the Jetson Nano in the presence of better-quality hardware, the speed and accuracy of the overall system is becoming a state-of-theart. This limitation could be omitted with the use of multiple cameras that could be placed over the premises in several ground coordinates, increasing the accuracy of the computer vision algorithms. Moreover, due to the inability to fly a drone in extreme weather conditions such as rain, strong winds or lightning, the system is limited to be used under few specifications. As swimming in extreme weather conditions is not preferred either, the system could be further improved to emit a warning signal if a person was to swim in any of the above weather conditions, bypassing the need to fly the drone. Additionally, all the processing is done on the client side of the applications on the Jetson Nano board, preventing any security and privacy issues that might arise due to the sensitive information inputted through the cameras. For future developments convenience wise, the system could benefit by having an additional set of cameras to identify and verify a drowning or a hazardous activity on the premises. Accessibility could also be improved by extending the Android service to be an application both in Android and iOS platforms that could hold the details of each premise individually, making a centralised system that watches over the decentralised pool premises. Both drown and hazardous activity detection could be improved by gathering a night time dataset that increases the accuracy of the data in low light.

CHAPTER-13

13. APPENDIX

(i) SOURCE CODE

```
[net]
# Testing#
batch=1 #
subdivisions=
1# Training
batch=64
subdivisions=
16 width=608
height=608
channels=3
momentum=0.
9
decay=0.0005
angle=0
saturation =
1.5
exposure = 1.5hue=.1

learning_rate=0.01
burn_in=1000
max_batches =
500200policy=steps
steps=400000,45000
0 scales=.1,.1

[convolution
al]
batch_norm
alize=1
filters=32
size=3
stride=1
pad=1
activation=le
aky
```

#
Downsampl
e

[convolution
al]
batch_norm
alize=1
filters=64
size=3
stride=2
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=32
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=64
size=3
stride=1
pad=1
activation=l
eaky
[shortcut]fro
m=- 3

activation=li
near#

Downsampl
e

[convolution
al]
batch_norm
alize=1
filters=128
size=3
stride=2
pad=1
activation=l
eaky

[convolution
al]
batch_norm
alize=1
filters=64
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=128
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=64
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=128
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

#
Downsampl
e

[convolution
al]
batch_norm

alize=1
filters=256si
ze=3
stride=2
pad=1
activation=l
eaky

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1

filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1

filters=256
size=3
stride=1
pad=1
activation=l
eaky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

```
[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation
=leaky
```

```
[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky
```

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near#

Downsampl
e

[convolution
al]
batch_norm

alize=1
filters=512
size=3
stride=2
pad=1 activation=leaky

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=l
eaky

[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1

filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1

filters=512
size=3
stride=1
pad=1 activation=leaky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation
=leaky

[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=l
eaky

[shortcut]fro
m=- 3
activation=li
near

```
[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=l
eaky

[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near


[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky
```

```
[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near


[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=l
eaky

[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=le
aky
```

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=512
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near#

Downsampl
e

[convolution
al]
batch_norm

alize=1
filters=1024
size=3
stride=2
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=1024
size=3
stride=1
pad=1
activation=l
eaky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm

alize=1
filters=512
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=1024
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=le
aky

[convolution
al]
batch_norm

alize=1
filters=1024
size=3
stride=1
pad=1
activation=l
eaky

[shortcut]fro
m=- 3
activation=li
near

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=l
eaky

[convolution
al]
batch_norm
alize=1
filters=1024
size=3
stride=1
pad=1
activation=le
aky

[shortcut]fro
m=- 3

activation=li
near

######################

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize=
1size=3 stride=1
pad=1
filters=1024
activation=leaky

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation=l
eaky

[convolutional]
batch_normalize=
1size=3 stride=1
pad=1

```
filters=1024
activation=leaky

[convolution
al]
batch_norm
alize=1
filters=512
size=1
stride=1
pad=1
activation
=leaky

[convolutional]
batch_normalize=
1size=3 stride=1
pad=1
filters=1024
activation=le
aky

[convolutional
]size=1
stride=1
pad=1
filters=255
activation=li
near


[yolo]
mask =
6,7,8
anchors = 10,13, 16,30, 33,23,  30,61,  62,45,  59,119,  116,90,
156,198,
373,326
classes=80
num=9    jitter=.3
```

ignore_thresh =
.7 truth_thresh =
1random=1

[route]
layers = -4

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=l
eaky

[upsample]
stride=2

[route]
layers =
-1, 61

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize
=1size=3
stride=1 pad=1
filters=512
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize
=1size=3
stride=1 pad=1
filters=512
activation=le
aky

[convolution
al]
batch_norm
alize=1
filters=256
size=1
stride=1
pad=1
activation=le
aky

```
[convolutional]
batch_normalize
=1size=3
stride=1 pad=1
filters=512
activation=le
aky

[convolutional
]size=1
stride=1
pad=1
filters=255
activation=li
near


[yolo]
mask =
3,4,5
anchors
= 10,13,
16,30,
33,23,
30,61,
62,45,
59,119,
116,90,
156,198,
373,326
classes=80
num=9    jitter=.3
ignore_thresh   =
.7 truth_thresh =
1random=1


[route]
layers = -4
```

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[upsample]
stride=2

[route]
layers =
-1, 36

[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation
=leaky

[convolutional]
batch_normalize
=1size=3
stride=1 pad=1
filters=256
activation=leaky

```
[convolution
al]
batch_norm
alize=1
filters=128
size=1
stride=1
pad=1
activation=le
aky

[convolutional]
batch_normalize=1siz
e=3 stride=1
 pad=1
 filters=256
 activation=l
 eaky

 [convolution
 al]
 batch_norm
 alize=1
 filters=128
 size=1
 stride=1
 pad=1
 activation=l
 eaky

 [convolutional]
 batch_normalize
 =1size=3
 stride=1 pad=1
 filters=256
 activation=leaky
```

[convolutional
]size=1
stride=1
pad=1
filters=255
activation=l
inear

[yolo]
mask =
0,1,2
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198,
373,326
classes=8
0 num=9
jitter=.3
ignore_thr
esh = .7
truth_thres
h = 1
random=1

Source code(ii)

```
#import       necessary

packagesimport cv2

import os

import numpy as np

from .utils import download_file
```

```
initialize    =

Truenet    =

None

dest_dir = os.path.expanduser('~') + os.path.sep + '.cvlib' + os.path.sep
+ 'object_detection' +os.path.sep + 'yolo' + os.path.sep + 'yolov3'

classes = None

#colors are BGR instead of RGB in python

COLORS = [0,0,255], [255,0,0]


def populate_class_labels():


  #we are using a pre existent classifier which is more reliable

  and more efficient than one#we could make using only a laptop

  #The classifier should be downloaded automatically when you
  run this scriptclass_file_name =

  'yolov3_classes.txt'

  class_file_abs_path = dest_dir + os.path.sep +
  class_file_name


  url = 'https://github.com/Nico31415/Drowning-

  Detector/raw/master/yolov3.txt'if not

  os.path.exists(class_file_abs_path):

    download_file(url=url, file_name=class_file_name,
    dest_dir=dest_dir)f =

  open(class_file_abs_path, 'r')

  classes = [line.strip() for line in f.readlines()]
```

```
    return classes


 def get_output_layers(net)
#the number of output layers in a neural network is the
number of possible#things the network can detect, such
as a person, a dog, a tie, a phone... layer_names =
net.getLayerNames()

    output_layers = [layer_names[i[0] - 1] for i in
    net.getUnconnectedOutLayers()]


    return output_layers



def draw_bbox(img, bbox, labels, confidence, Drowning,
write_conf=False):


    global
    COLORS
    global
    classes


    if classes is None:

      classes = populate_class_labels()
```

```
    for i, label in enumerate(labels):


        #if the person is drowning, the box will be
        drawn red instead of blueif label == 'person'
        and Drowning: color = COLORS[0] label
          = 'DROWNING'
else:
color                =
        COLORS[1
        ]

if write_conf:

          label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'
        #you only need to points (the opposite corners) to draw a
        rectangle. These points#are stored in the variable bbox
        cv2.rectangle(img,                    (bbox[i][0],bbox[i][1]),
        (bbox[i][2],bbox[i][3]), color, 2)


                    cv2.putText(img, label, (bbox[i][0],bbox[i][1]-10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)


    return img

def  detect_common_objects(image,  confidence=0.5,
                nms_thresh=0.3):


    Height, Width = image.shape[:2]scale =
```

0.00392

global

classes

global

dest_dir

#all the weights and the neural network algorithm are already preconfigured#as we are using

YOLO

#this part of the script just downloads the

YOLO files config_file_name = 'yolov3.cfg'

config_file_abs_path = dest_dir + os.path.sep + config_file_name

weights_file_name = 'yolov3.weights'

weights_file_abs_path = dest_dir + os.path.sep + weights_file_name

url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.cfg'

if not os.path.exists(config_file_abs_path):

download_file(url=url, file_name=config_file_name, dest_dir=dest_dir)

url = 'https://pjreddie.com/media/files/yolov3.weights'

```
    if not os.path.exists(weights_file_abs_path):

download_file(url=url, file_name=weights_file_name,
                        dest_dir=dest_dir)




global
    initialize
    global
    net


    if initialize:

      classes = populate_class_labels()

          net = cv2.dnn.readNet(weights_file_abs_path,
                    config_file_abs_path)initialize = False


    blob = cv2.dnn.blobFromImage(image, scale, (416,416),
    (0,0,0), True, crop=False)


    net.setInput(blob)


    outs = net.forward(get_output_layers(net))
```

```
class_ids
=          []
confiden
ces  =  []
boxes   =
[]

for out in outs:
  for detection in out: scores =
    detection[5:]
    class_id            =
    np.argmax(score
    s)   max_conf   =
    scores[class_id] if
    max_conf          >
    confidence:
```

```
        center_x = int(detection[0] *
        Width) center_y =
        int(detection[1] * Height)w =
        int(detection[2] * Width)
        h   =   int(detection[3]   *
        Height)x = center_x - w
        / 2
        y    =    center_y    -    h    /    2
        class_ids.append(class_id)
        confidences.append(float(max_
        conf))boxes.append([x, y, w, h])


          indices = cv2.dnn.NMSBoxes(boxes,
        confidences, confidence, nms_thresh)


bbox = []
label = []
conf = []


for i in indices:
  i = i[0]
   box =
   boxes[i]x
```

```
 = box[0] y
 = box[1]
 w =
 box[2]
  h       =
box[3]
bbox.append([round(x),  round(y),  round(x+w),  round(y+h)])
label.append(str(classes[class_ids[i]]))
conf.append(confidences[i])

return bbox, label, conf
```

Github Link:

https://github.com/IBM-EPBL/IBM-Project-13209-1659514405