FINAL DELIVERABLE PROJECT DOCUMENTATION

Date	19 November 2022
Team ID	PNT2022TMID00309
Project Name	Virtual Eye-Lifeguard for Swimming Pools to Detect the Active Drowning

CHAPTER-1 INTRODUCTION

Video surveillance can be used as a tool for monitoring and security. Observing private as well as public spaces has become a sensitive issue. The visual monitoring capabilities can be employed in many different locations to help people live more safely. Video-based surveillance systems are designed and installed in places such as railway stations, airports, and even dangerous environments. Image processing, pattern recognition and machine-vision based methods are efficient ways for real-time intelligent monitoring of the objects or events of interest.

One important environment that the need for monitoring systems is crucially sensed, is the swimming pool. Each year, many people including children are drowned or at a risk of drowning in the swimming pool as the lifeguards may not always be of help. This raises the need for having a system that will automatically detect the drowning person and alert the lifeguards of such danger. Real-time detection of a drowning person in a swimming pool or any such environment is a challenging task that requires an accurate system. The detection is supposed to have a great deal of accuracy since there can be ripples, shadows and splashes (they can reduce the accuracy).

The Drowning Detection Systems (DDS) is gradually becoming a subject of interest, globally. Whilst there are plenty of academic articles dedicated to the technology and design behind these products in the fields of biometrics, computer science and electronic engineering, there is limited academic research investigating their application to real-world scenarios. Additionally, there is uncertainty around their use alongside traditional lifeguarding; whether international testing standards (ISO standards) are robust enough; and general risks affecting the effectiveness of these products. This includes factors such as water clarity, high pool occupancy, lighting, glare and attractions such as water slides and wave machines. These concerns alongside the lack of research and high installation costs have resulted in a reluctance by some operators to incorporate DDS into their pools. This signifies the importance of independent research into DDS. intends to support the move towards the shared goal of improved pool safety.

This section will start with an outline of the various definitions of DDS, followed by a description of the objectives and methodology of this review. It will then discuss what the current DDS standards are alongside legislation and guidance available around DDS, and provide a summary of the shared tasks towards the effective operation of DDS. Following this, the literature review will examine the co-existence between DDS and traditional lifeguarding, provide an analysis of its impact so far, and conclude with recommendations on the direction of future DDS research.

Project Overview

Purpose

a. Indicate and summarize what is known on Drowning Detection Systems.
b. Assess the current literature on Drowning Detection Systems, including their use in indoor pool environments along with interaction with traditional lifeguarding.
c. Significantly improved comprehension about where DDS are positioned in the health and safety perspective of indoor swimming pools.

The value that can be generated from these objevtives arise from the recognition that currently, there are no published documents drawing together all the current DDS research. The literature review aims to contribute as independent research in this field and hopes to signpost the potential future direction of DDS research.

CHAPTER-2

LITERATURE SURVEY

Of the differing definitions of DDS, most outline three prime factors:

1. surveillance,
2. detection of a pool user in difficulty, and
3. raising an alert or alarm

In swimming pool monitoring intelligent systems, different approaches have been proposed. Most methods tend to perform background processing on input video frames. Some apply background subtraction and image denoising to detect the drowning person. Also, neural networks can be trained to classify near-drowning and normal swimming patterns. However, this requires to have a large dataset of both groups of behaviour. The dataset is obtained by attaching a pressure sensor to a swimmer imitating drowning behaviour and normal swimming. For example, ISO_20380 (the document published by the International Organisation for Standardization (2017) outlining the international safety requirements and test standards for DDS) defines the technology as an ‘Automated system including means for digitizing series of images of people in the pool basin, means for comparing and analysing digitized images and decision means for setting off and sending an alarm to trained staff when a detection occurs’. In comparison, there are broader definitions that are comprehensive of other technologies that focus on the surveillance aspect, for example, ‘DDS is used to explain various electronic systems that are designed to assist with the surveillance of swimmers within the water of a swimming pool’ (Sport England, 2011). This definition would include CCTV that helps give lifeguards an underwater view but does not have the capacity to detect a pool user in difficulty or raise an alarm. For this to be effective, staff would have to make sure the CCTV is being monitored at all times, making the staff experience with this very different to the experience of using a DDS falling under the first definition.

It is important to distinguish what exactly constitutes a DDS as there are different areas of responsibility required from different actors involved in the effective operation of DDS, which will be examined in chapter 4. For this literature review, research has focused on the definition used by the ISO and other sources that incorporate all three elements of surveillance, detection and alarm raising.

Existing Problem

There is discussion around whether DDS can be helpful or harmful towards lifeguarding practices and how DDS may change the view or perspective of traditional lifeguarding, as well as some disputes on whether they serve as justification for reducing lifeguard numbers. Although various literature on DDS mostly agree on areas such as the risks and issues associated with DDS performance, there are other areas where sources offer differing points of view, for example, DDS and their co- existence with lifeguards. The term blended lifeguarding' or 'modern lifeguarding' has been newly coined to describe the concept of traditional lifeguarding practices being blended with technology for drowning detection (Swimming Pool Scene, 2017).

Currently, there is little qualitative or quantitative research analysing the experiences of lifeguards themselves relating to this concept.

References

1. AngelEye. (2019). AngelEye – Distributors. Retrieved from: https://www.angeleye.it/news. php?id=28&newscat=10

2. Aquatics International. (2007). Traumatic Experiences – Should we make our youngest lifeguards come face to face with death? Retrieved from: https://www.aquaticsintl.com/facilities/traumaticexperiences_o

3. British Standards Institution. (2018). BS EN 15288-1, Swimming pools for public use. Safety requirements for design. Retrieved from: https://shop.bsigroup.com/ProductDetail/?pid=000000000030360254

4. British Standards Institution 1. (2018). BS EN 15288-2, Swimming pools for public use. Safety requirements for operation. Retrieved from: https://shop.bsigroup.com/ProductDetail/?p id=000000000030360257

5. Drowning Prevention. (2017).The Need. Retrieved from: https://www.drowningprevention.com.au/

6. German Institute for Standardization. (2019). German national guideline DGfdB R 94.15 “Test methods for camera-based drowning detection systems under operational conditions” (German Association for Public Swimming Pools).

7. Haizhou Li, Haizhou Li, Kar-Ann Toh and Liyuan Li. (2012). Advanced Topics in Biometrics, World Scientific Publishing Co. Pte. Ltd., ISBN-13 978-981-4287-84-5

8. Health and Safety Executive. (2018). HSG179, Health and safety in swimming pools (Fourth edition).

ISO (2017) ISO_20380, First edition, Public swimming pools —
9. Computer vision systems for the detection of drowning accidents in swimming pools — Safety requirements and test methods.

PROBLEM STATEMNET DEFINITION

1. 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.

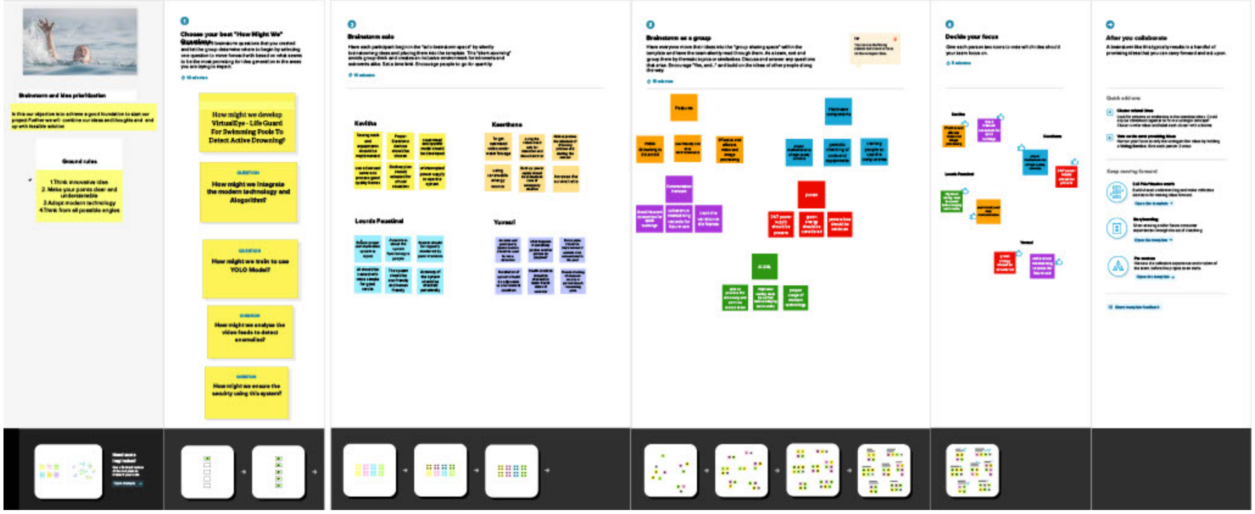
2. Applying the CNN algorithm to the dataset. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.

3. To overcome this dispute, a careful system is to be carried out along the swimming pools to save human life.

CHAPTER-3 IDEATION & PROPOSED SOLUTION

EMPATHY MAP CANVAS

IDEATION & BRAINSTORMING



Brainstorm and idea prioritization

In this our objective is to achieve a good foundation to start our project.Futher we will combine our ideas and thoughts and end up with feasible solution

Ground rules

1.Think innovative idea
2. Make your points clear and understandable
3.Adopt modern technology
4.Think from all possible angles

1

Choose your best "How Might We" Questions

How might we develop VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning?

QUESTION
How might we integrate the modern technology and Alogorithm?

QUESTION
How might we train to use YOLO Model?

QUESTION
How might we analyse the video feeds to detect anomalies?

QUESTION
How might we ensure the security using this system?

2

Brainstorm solo

Kavitha

Keerthana

Lourds Faustinal

Yuvasri

3

Brainstorm as a group

Features

Hardware components

power

4

Decide your focus

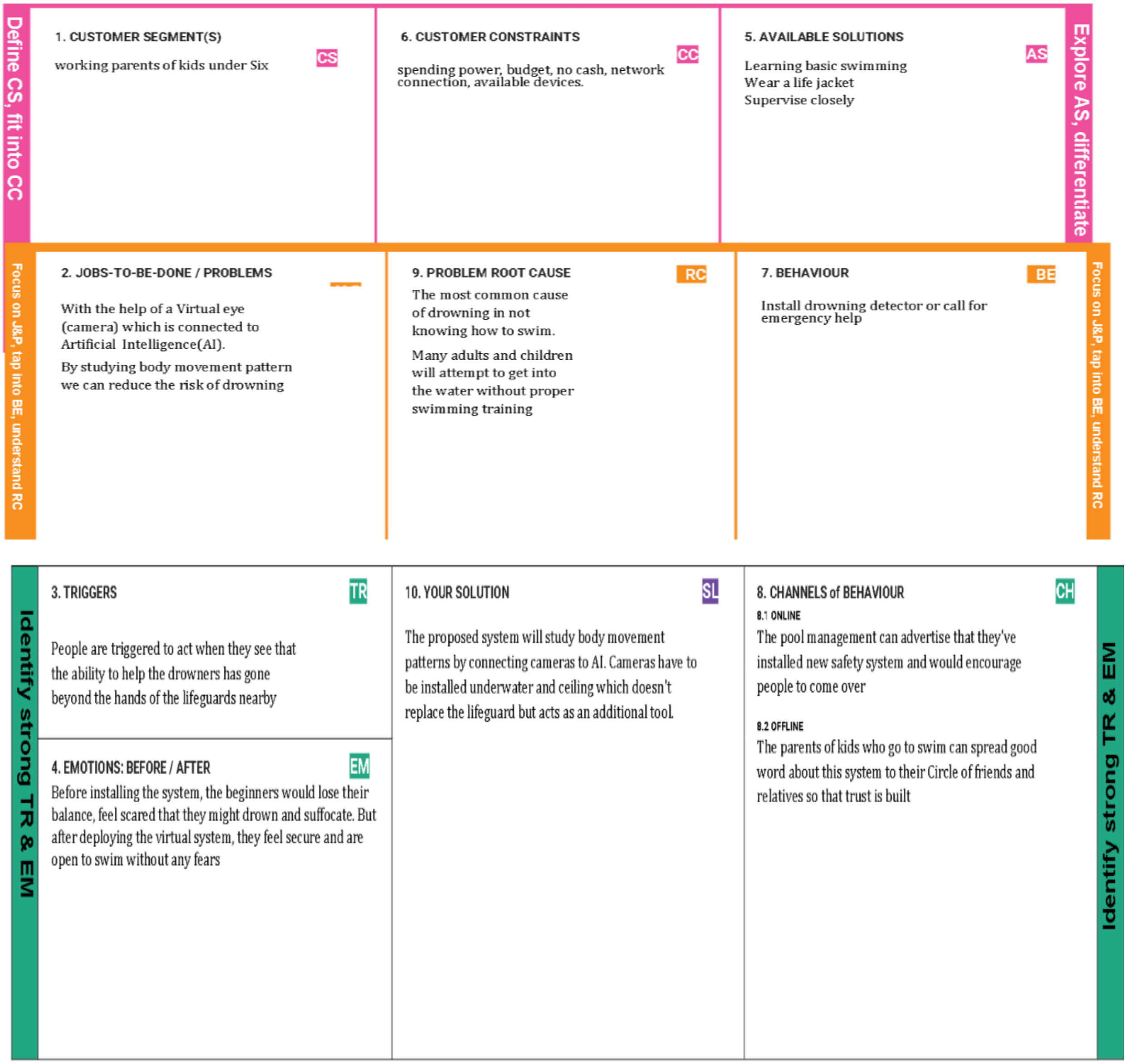
After you collaborate

PROPOSED SOLUTION

Proposed Solution Template:

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Swimming helps people to reduce stress so many people engage in that activity. But for beginners and kids around 6 years of age find it difficult and leads to accidental drowning. In order to reduce the mortality rate, we need to devise a system
2.	**Idea / Solution description**	I Using Artificial intelligence technology, we Install cameras in underwater and ceiling to detect drowning and immediately alert the guards
3.	**Novelty / Uniqueness**	The system effectively tracks the body movement positions with high accuracy. The rate of detection is very quick and will reduce mortality rate
4.	**Social Impact / Customer Satisfaction**	As beginners and kids are very excited to swim, when such a system is deployed their parents will feel safe and relaxed
5.	**Business Model (Revenue Model)**	We can generate direct revenue by tying up with hotels and private pools. Also implementing them in tourist spots will encourage more visitors
6.	**Scalability of the Solution**	By using AI and deep learning, this solution will be effective for now and also future. It will have a solid infrastructure with accurate prediction and increased performance

PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

working parents of kids under Six

6. CUSTOMER CONSTRAINTS CC

spending power, budget, no cash, network connection, available devices.

5. AVAILABLE SOLUTIONS AS

Learning basic swimming
Wear a life jacket
Supervise closely

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

With the help of a Virtual eye (camera) which is connected to Artificial Intelligence(AI).

By studying body movement pattern we can reduce the risk of drowning

9. PROBLEM ROOT CAUSE RC

The most common cause of drowning in not knowing how to swim.

Many adults and children will attempt to get into the water without proper swimming training

7. BEHAVIOUR BE

Install drowning detector or call for emergency help

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

People are triggered to act when they see that the ability to help the drowners has gone beyond the hands of the lifeguards nearby

4. EMOTIONS: BEFORE / AFTER EM

Before installing the system, the beginners would lose their balance, feel scared that they might drown and suffocate. But after deploying the virtual system, they feel secure and are open to swim without any fears

10. YOUR SOLUTION SL

The proposed system will study body movement patterns by connecting cameras to AI. Cameras have to be installed underwater and ceiling which doesn't replace the lifeguard but acts as an additional tool.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

The pool management can advertise that they've installed new safety system and would encourage people to come over

8.2 OFFLINE

The parents of kids who go to swim can spread good word about this system to their Circle of friends and relatives so that trust is built

Identify strong TR & EM

CHAPTER 4 REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	Installation	Should fix the cameras underwater and in the ceilings
FR-2	Detection	Note when there’s terrifying reaction and unconsciousness
FR-3	Alert System	Warn/Intimate the lifeguards immediately
FR-4	Audio/Video needs	Use it for reference data
FR-5	Support	Use it for reference data

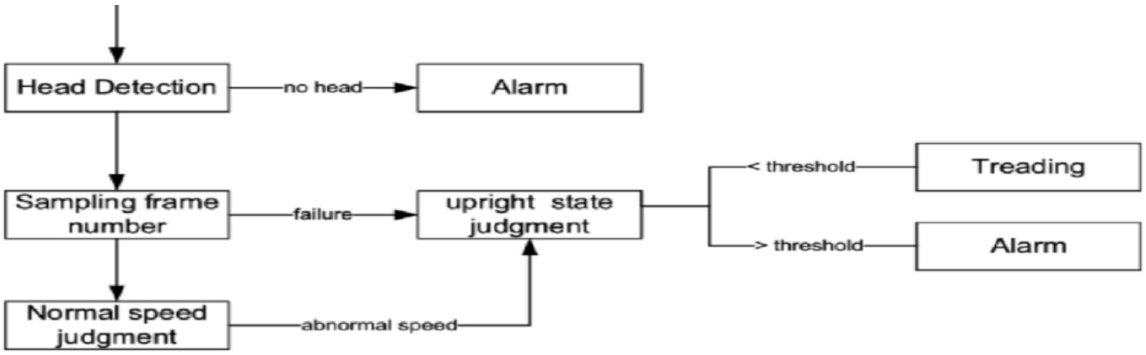
NON-FUNCTIONAL REQUIREMENT

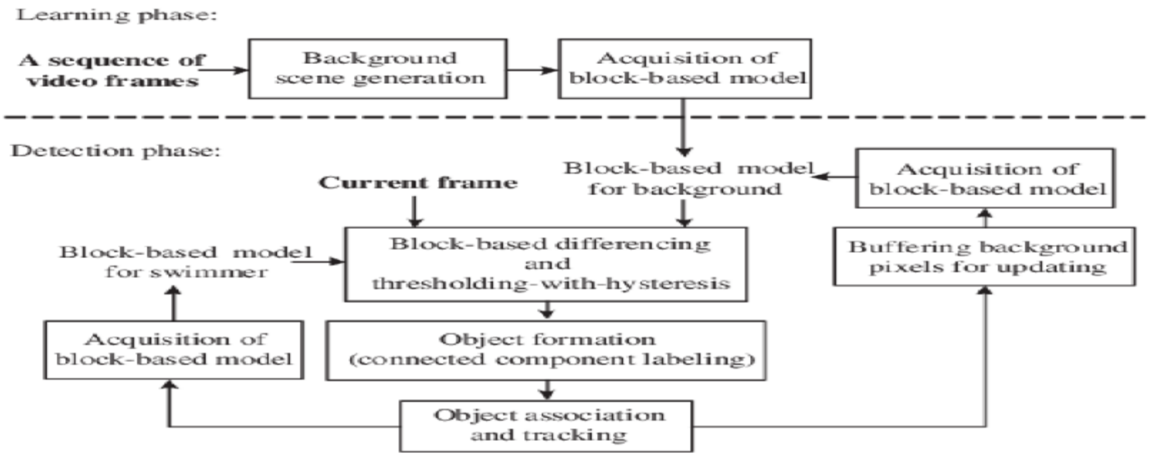
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	**System should be used by knowing all the features appropriately and using it effectively.**
NFR-2	**Security**	**The video feeds should not be used for other purposes**
NFR-3	**Reliability**	**Immediate action should be taken after the trigger of warning**
NFR-4	**Performance**	**Accurate results should be provided**
NFR-5	**Availability**	**The monitoring should be done continuously while the pool is being used anytime. Also vests and tubes for rescue should be there.**
NFR-6	**Scalability**	**The system should be cost effective. It should be practically implementable**

CHAPTER-5 PROJECT DESIGN - DATAFLOW DIAGRAMS

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



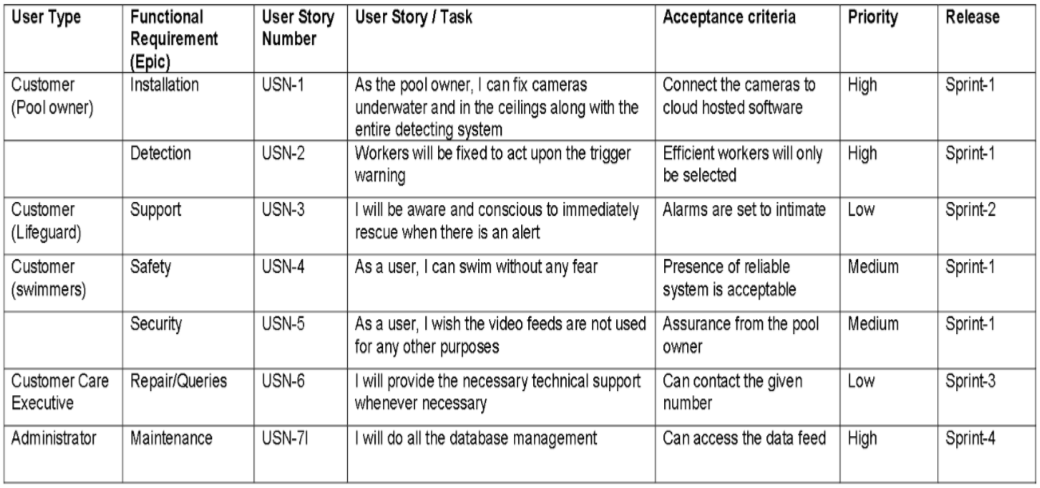

Learning phase:
A sequence of video frames
Background scene generation
Acquisition of block-based model
Detection phase:
Block-based model for background
Acquisition of block-based model
Current frame
Block-based model for swimmer
Block-based differencing and thresholding-with-hysteresis
Buffering background pixels for updating
Acquisition of block-based model
Object formation (connected component labeling)
Object association and tracking

Solution Architecture:

1. 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.
2. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analysing the video feeds to detect any anomalies.
3. 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 then an alert will be generated to attract lifeguards' attention.
4. The system is not designed to replace a lifeguard or other human monitor, but to act as an additional tool.
5. It helps the lifeguardto detect the underwater situation where they can'teasily observe.

USER STORIES

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



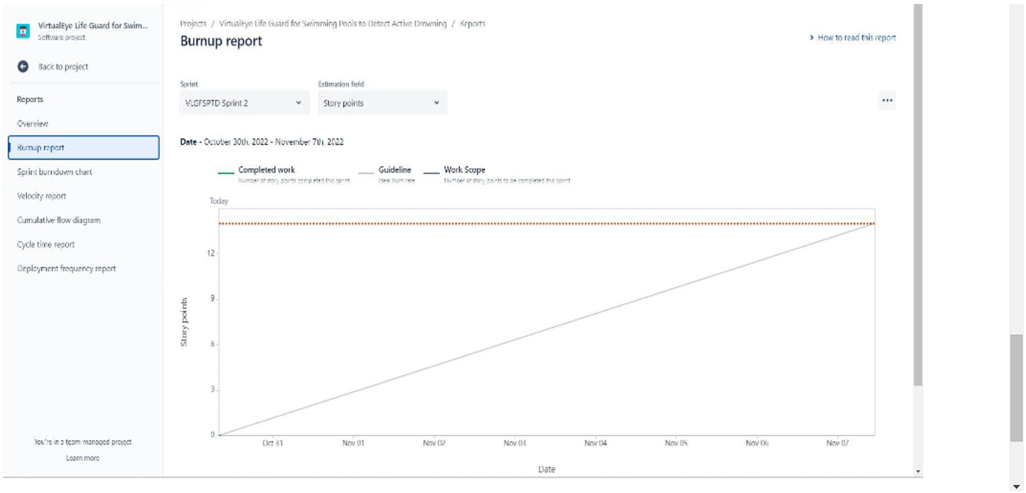
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Pool owner)	Installation	USN-1	As the pool owner, I can fix cameras underwater and in the ceilings along with the entire detecting system	Connect the cameras to cloud hosted software	High	Sprint-1
	Detection	USN-2	Workers will be fixed to act upon the trigger warning	Efficient workers will only be selected	High	Sprint-1
Customer (Lifeguard)	Support	USN-3	I will be aware and conscious to immediately rescue when there is an alert	Alarms are set to intimate	Low	Sprint-2
Customer (swimmers)	Safety	USN-4	As a user, I can swim without any fear	Presence of reliable system is acceptable	Medium	Sprint-1
	Security	USN-5	As a user, I wish the video feeds are not used for any other purposes	Assurance from the pool owner	Medium	Sprint-1
Customer Care Executive	Repair/Queries	USN-6	I will provide the necessary technical support whenever necessary	Can contact the given number	Low	Sprint-3
Administrator	Maintenance	USN-7l	I will do all the database management	Can access the data feed	High	Sprint-4

CHAPTER 6 – SPRINT PLANNING AND ESTIMATION

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	6 Days	24 Oct 2022	29 Oct 2022	6	29 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	12	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

SPRINT DELIVERY SCHEDULE

REPORT FROM JIRA Backlog (scrum)



Projects / VirtualEye Life Guard for Swimming Pools to Detect Active Drowning / Reports

Burnup report

Sprint: VLGFSPTD Sprint 2

Estimation field: Story points

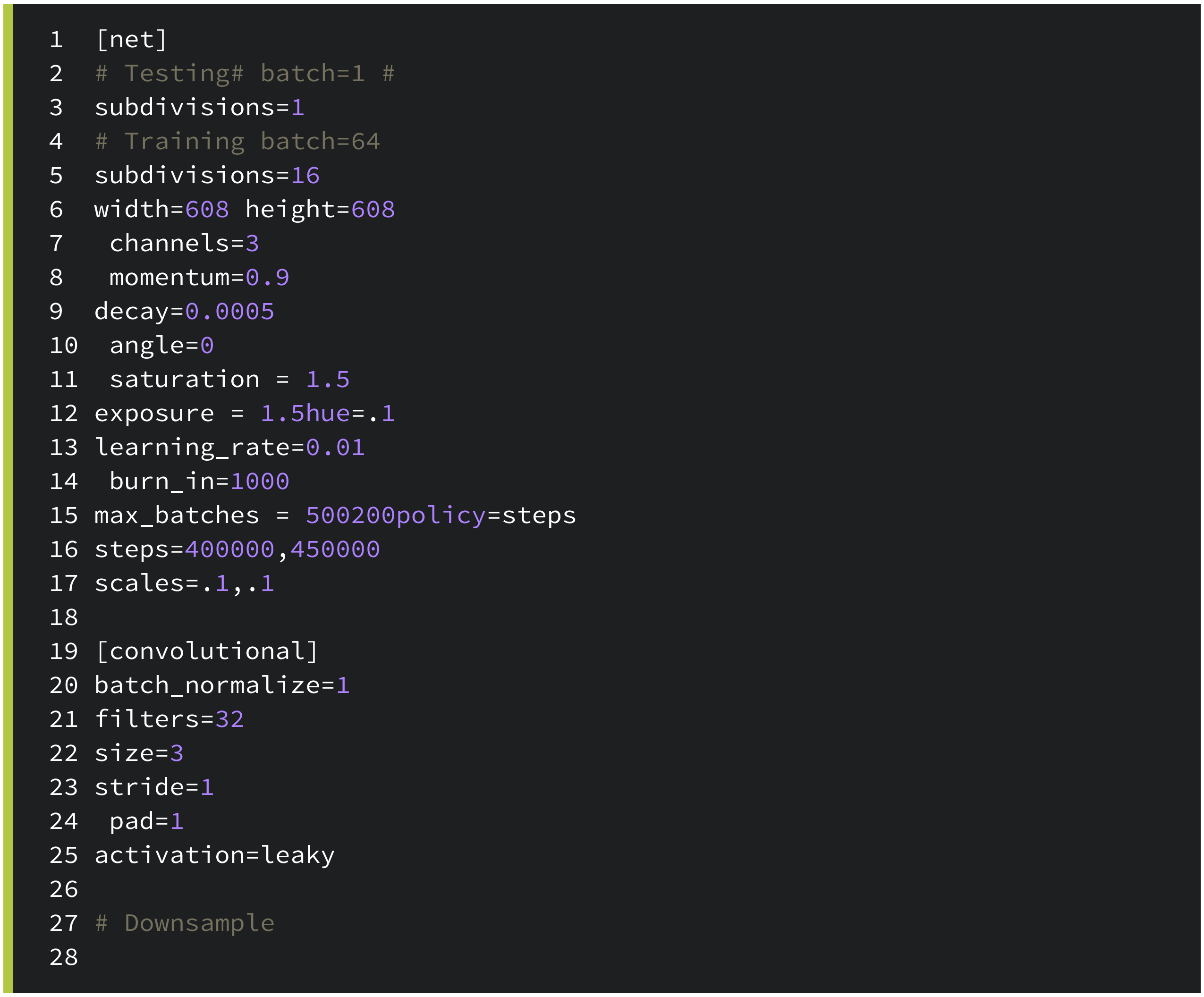
Date - October 30th, 2022 - November 7th, 2022

You're in a team-managed project

Learn more

CHAPTER-7

CODING & SOLUTION 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=400000,450000
scales=.1,.1

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
 pad=1
activation=leaky

# Downsample

```

```
[convolutional]
batch_normalize=1
filters=64
 size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
 filters=32
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky
 [shortcut]
from=- 3 activation=linear#

Downsample

[convolutional] batch_normalize=1 filters=128 size=3 stride=2 pad=1 activation=leaky
[convolutional] batch_normalize=1 filters=64 size=1
stride=1 pad=1
activation=leaky

[convolutional] batch_normalize=1 filters=128 size=3 stride=1 pad=1 activation=leaky

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=64 size=1 stride=1 pad=1 activation=leaky

[convolutional] batch_normalize=1 filters=128 size=3 stride=1 pad=1
activation=leaky

[shortcut]from=- 3 activation=linear
```

```
# Downsample
[convolutional] batch_normalize=1 filters=256size=3 stride=2 pad=1 activation=leaky

[convolutional] batch_normalize=1 filters=128 size=1 stride=1 pad=1
activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=128 size=1 stride=1 pad=1
activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=128 size=1 stride=1 pad=1
activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=128 size=1 stride=1 pad=1
activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=128 size=1 stride=1 pad=1
activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=128 size=1
stride=1 pad=1
```

```
activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=128 size=1 stride=1 pad=1
activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=128 size=1 stride=1 pad=1 activation=leaky

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

[shortcut]from=- 3 activation=linear#

Downsample

[convolutional] batch_normalize=1 filters=512 size=3 stride=2
pad=1 activation=leaky

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

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

[shortcut]from=- 3 activation=linear

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

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

[shortcut]from=- 3 activation=linear

```

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

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

[shortcut]from=- 3 activation=linear

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

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

[shortcut]from=- 3 activation=linear

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

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

[shortcut]from=- 3 activation=linear

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

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

[shortcut]from=- 3 activation=linear

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

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

[shortcut]from=- 3 activation=linear

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

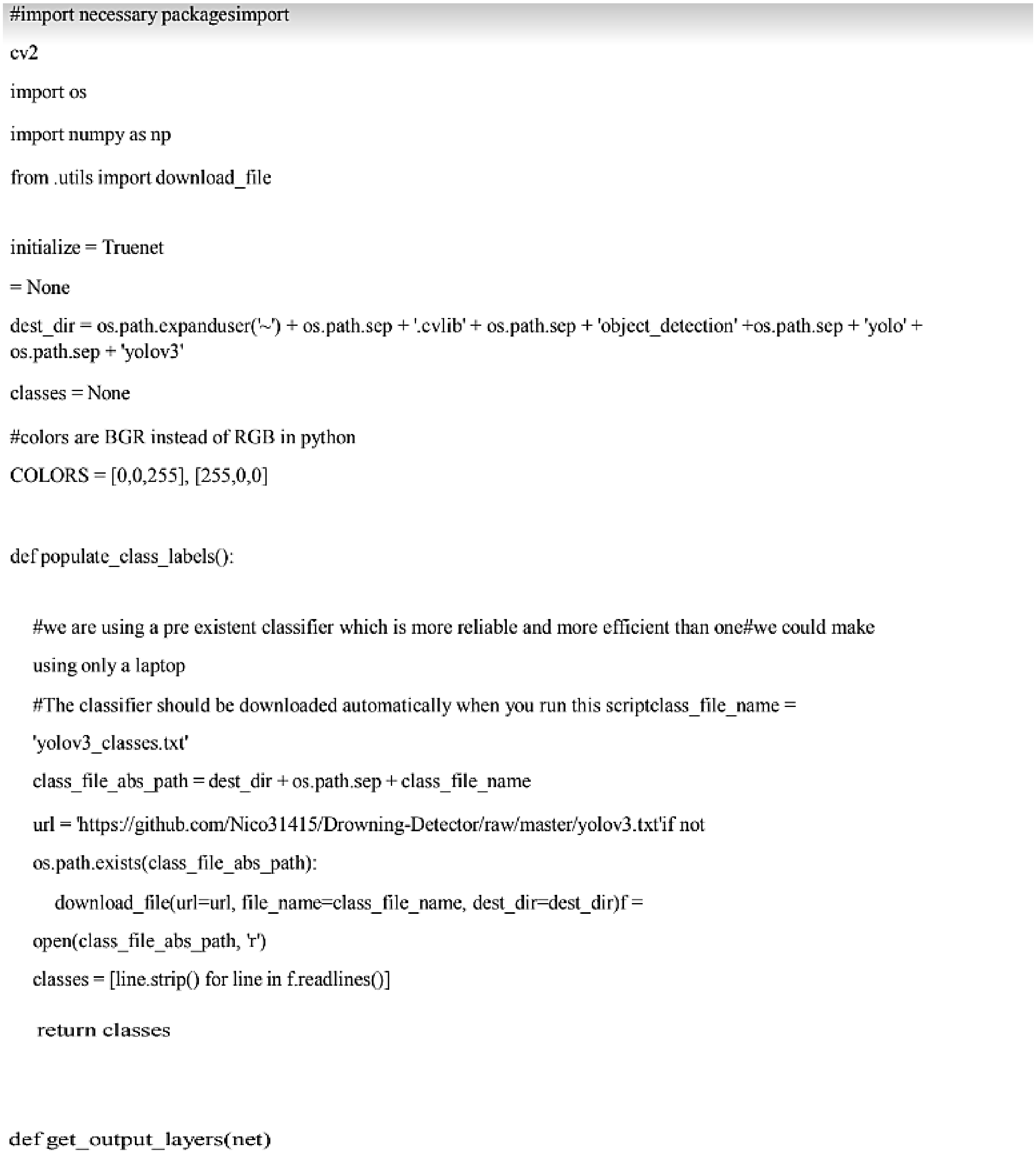
```

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

[shortcut]from=- 3 activation=linear#

Downsample

[convolutional] batch_normalize=1 filters=1024 size=3 stride=2 pad=1 activation=leaky

[convolutional] batch_normalize=1 filters=512 size=1 stride=1 pad=1 activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=512 size=1 stride=1 pad=1 activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=512 size=1 stride=1 pad=1
activation=leaky

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

[shortcut]from=- 3 activation=linear

[convolutional] batch_normalize=1 filters=512 size=1 stride=1 pad=1 activation=leaky

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

[shortcut]from=- 3 activation=linear
######################
[convolutional] batch_normalize=1 filters=512 size=1 stride=1 pad=1 activation=leaky
```

```

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

[convolutional] batch_normalize=1 filters=512 size=1
stride=1 pad=1 activation=leaky

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

[convolutional] batch_normalize=1 filters=512 size=1
stride=1 pad=1
activation=leaky

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

[convolutional]size=1 stride=1 pad=1 filters=255
activation=linear

[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

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

[upsample] stride=2

[route]
layers = -1, 61


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

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

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

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

```
   activation=leaky

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

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

[convolutional]size=1 stride=1 pad=1 filters=255 activation=linear

[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

[convolutional] batch_normalize=1 filters=128 size=1
stride=1 pad=1 activation=leaky

[upsample]
stride=2

[route]
layers = -1, 36


[convolutional] batch_normalize=1 filters=128 size=1
stride=1 pad=1
activation=leaky

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

[convolutional] batch_normalize=1 filters=128 size=1
stride=1 pad=1
activation=leaky

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

[convolutional] batch_normalize=1 filters=128 size=1 stride=1 pad=1
activation=leaky

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

```
activation=leaky

[convolutional]size=1 stride=1 pad=1 filters=255 activation=linear

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


```

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_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)
```

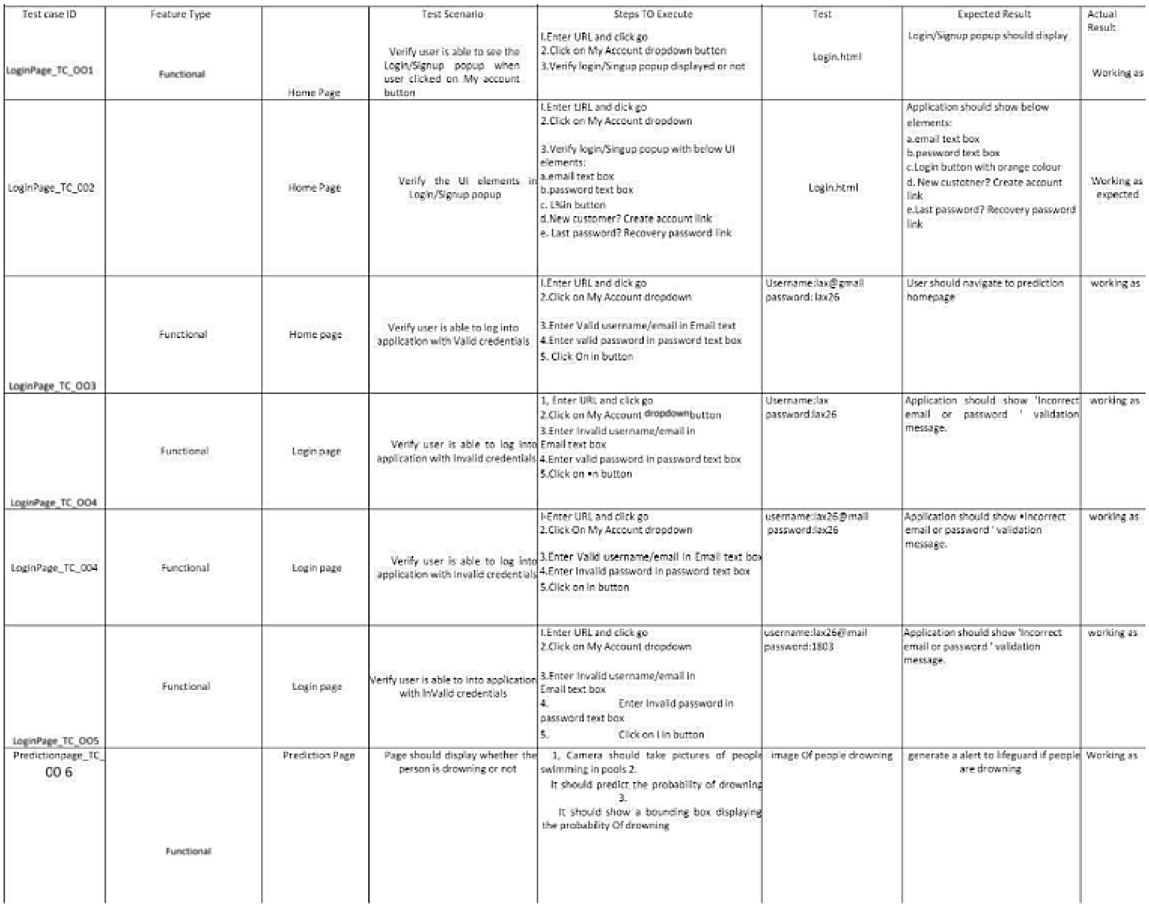
```
url = 'https://pjreddie.com/media/files/yolov3.weights'

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 = []
confidences = []
boxes = []
```

```
for out in outs:
    for detection in out: scores =
        detection[5:]
        class_id = np.argmax(scores)
        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

TESTING

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	l.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	l.Enter URL 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	l.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	l-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	l.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

USER ACCEPTANCE TESTING

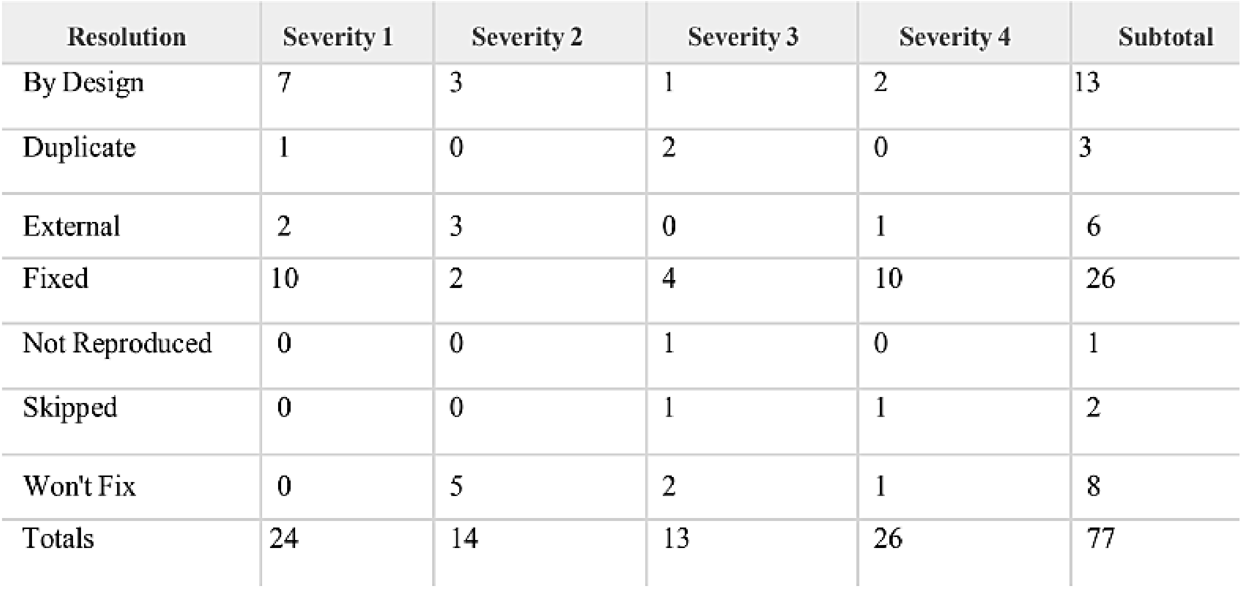
1. Purpose of Document

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

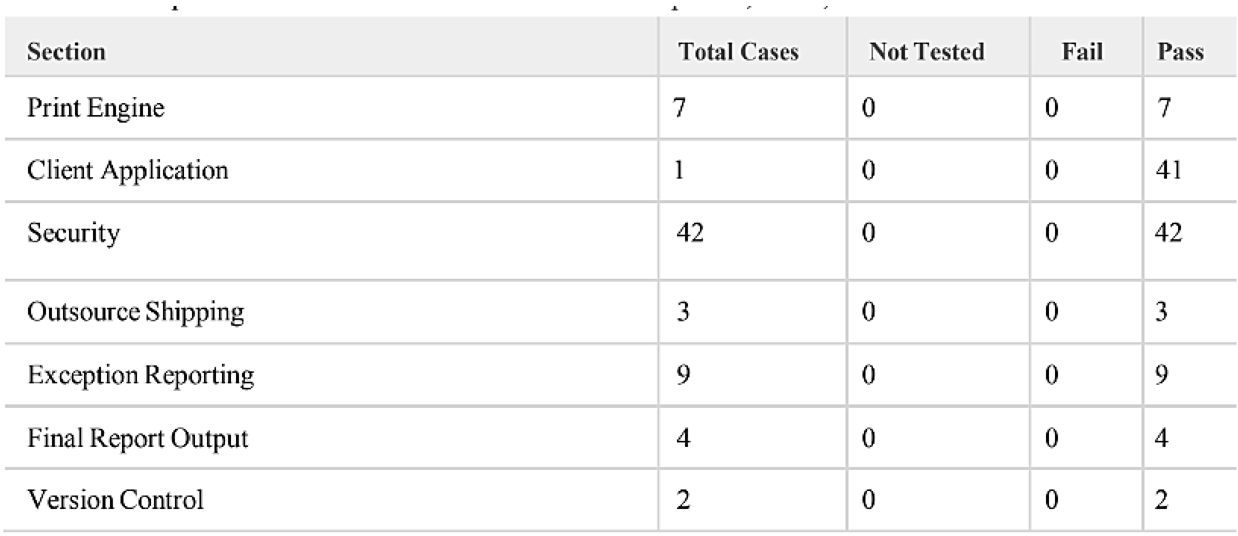
Resolution Severity 1 Severity 2 Severity 3 Severity 4 Subtotal



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

Test Case Analysis

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



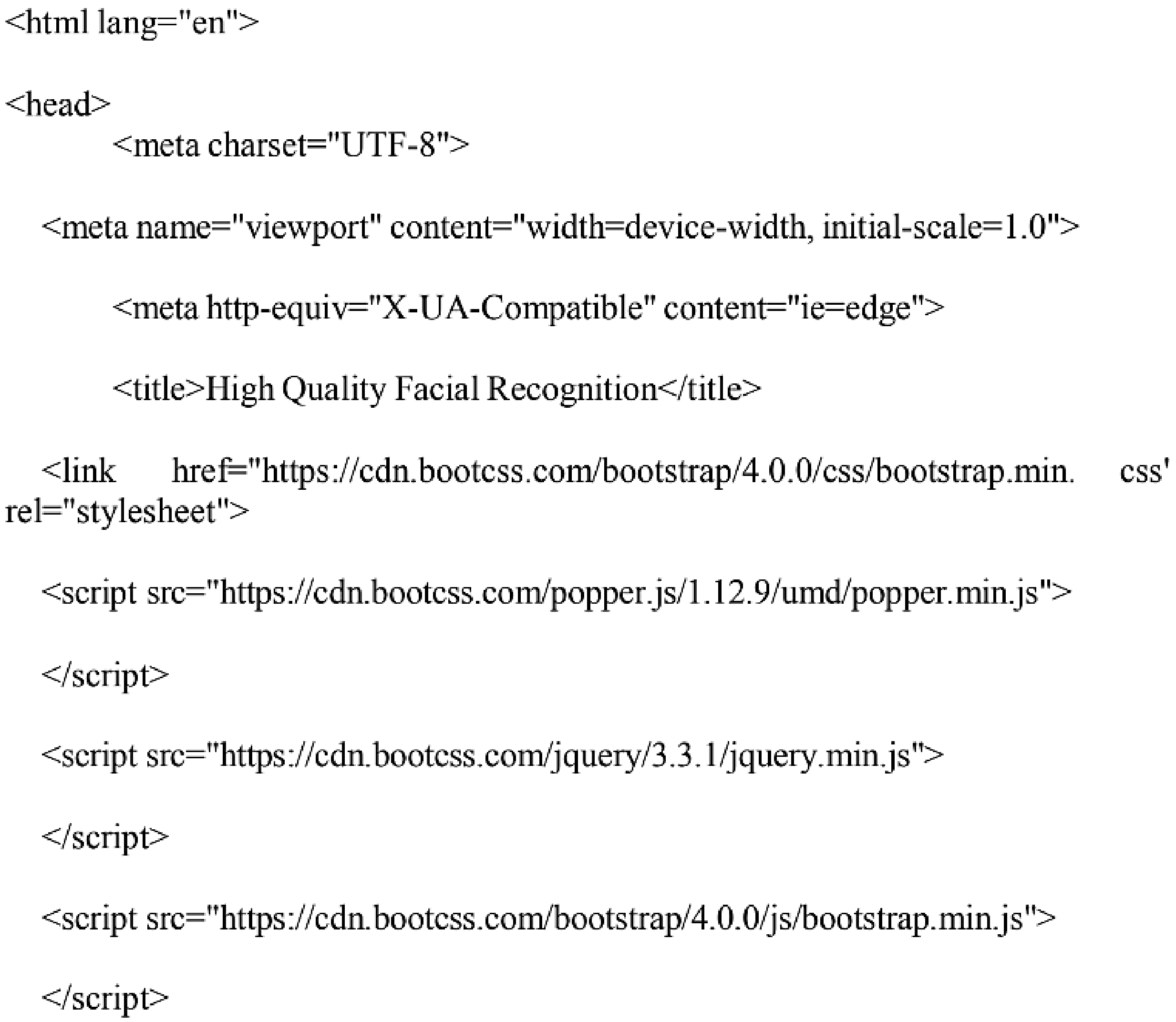
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	1	0	0	41
Security	42	0	0	42
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

Print Engine 7 Client Application 41 Security 42 Outsource Shipping 3 ExceptionReporting 9 Final Report Output 4 Version Control 2

CHAPTER-9

RESULT

PERFORMANCE METRICS

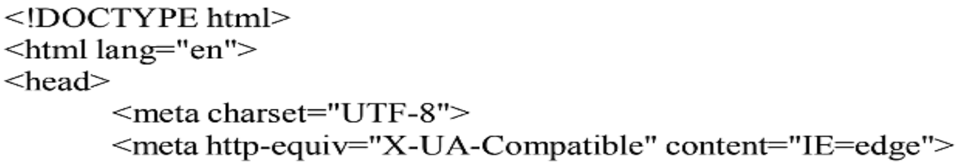


```
<html lang="en">

<head>
          <meta charset="UTF-8">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

          <meta http-equiv="X-UA-Compatible" content="ie=edge">

          <title>High Quality Facial Recognition</title>

   <link      href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.     css'
rel="stylesheet">

   <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js">

   </script>

   <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js">

   </script>

   <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js">

   </script>
```

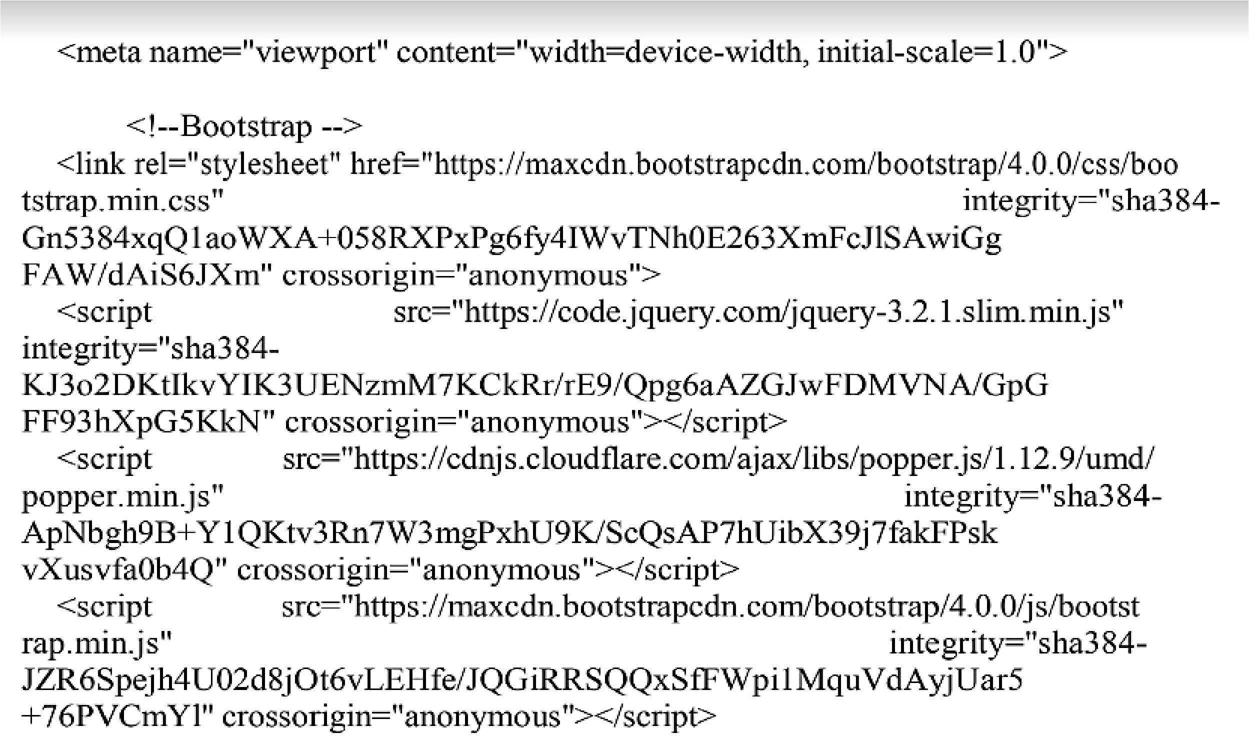
```
<link href="{{ url_for('static', filename='css/main.css') }}"rel="stylesheet">
                <style>
                .bg-dark {
                        background-color: #42678c!important;
                }
                #result {
                        color: #0a1c4ed1;
                }
                </style>
</head>

<body style="background-color:black";>
<header id="head" class="header">
              <section  id="navbar">
                     <h1 class="nav-heading"></i>Virtual Eye</h1>
                  <div class="nav--items">
                     <ul>
                        <li><a  href="{{ url_for('index')}}">Home</a></li>
                                        <li><a
href="{{ url_for('logout')}}">Logout</a></li>
                    <!-- <li><a href="#about">About</a></li>
                    <li><a href="#services">Services</a></li> -->


   </ul>
   </div>
   </section>
       </header>
       <div class="container">
           <div id="content" style="margin-top:2em">
                   <div class="container">
                     <div class="row">
                            <div class="col-sm-6 bd" >
                    <h2><em   style="color:white;">High   Quality Facial
Recognition</em></h2>
                         <br>
                    <p><h5><i style="color:white;">Emotion Detection Through
Facial Feature Recognition</i></h5></p>
                         <img              src="https://130e178e8f8ba617604b-
8aedd782b7d22cfe0d1146da69a52436.ssl.cf1.rackcdn.com/facial-    recognition-
use-triggers-gdpr-fine-showcase_image-10-a-                          12991.jpg"
style="height:240px"class="img-rounded" alt="Gesture">
                        </div>
                        <div class="col-sm-6">
                              <div>
                                    <h4 style="color:white;">Upload
Image Here</h4>
                   <form   action   =   "http://localhost:5000/"   id="upload-file"
method="post" enctype="multipart/form-data">
                              <label for="imageUpload" class="upload-
label">
                                    Choose Image
                         </label>
                         <input type="file" name="image"
id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">
                        </form>

                        <div class="image-section" style="display:none;">
                              <div class="img-preview">
                                    <div id="imagePreview">
                                    </div>
                              </div>
                              <div>


                         </div>
                         <div>
                           <button type="button" class="btn btn-info btn-lg "
id="btn-predict">Analyse</button>
```

```
      </div>
                              </div>

                              <div class="loader" style="display:none;"></div>

                              <h3>
                                      <span id="result"> </span>
                              </h3>


                      </div>
                              </div>

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


<footer>
   <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
</footer>

</html>
```

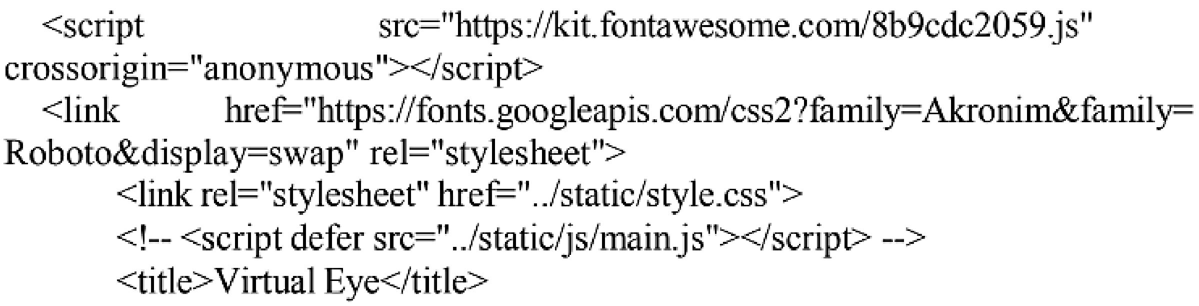
Index.html



```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

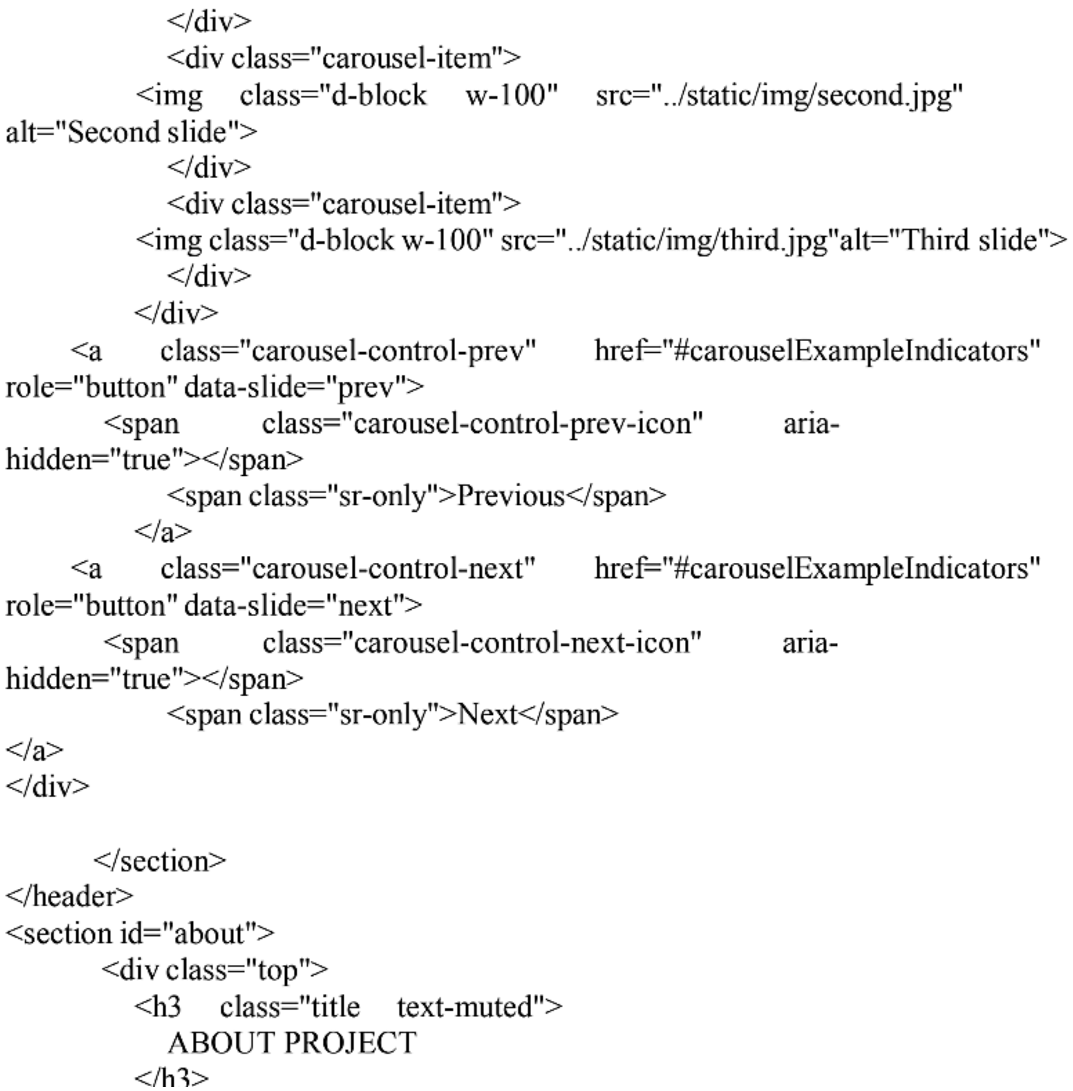


```
   <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <!--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>
```

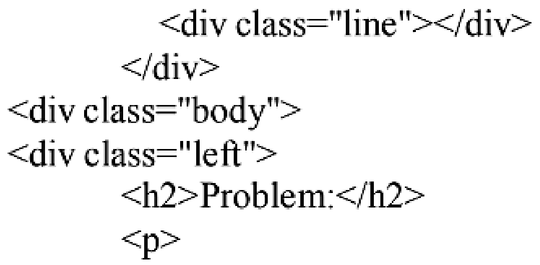


```
   <script                  src="https://kit.fontawesome.com/8b9cdc2059.js"
crossorigin="anonymous"></script>
   <link         href="https://fonts.googleapis.com/css2?family=Akronim&family=
Roboto&display=swap" rel="stylesheet">
        <link rel="stylesheet" href="../static/style.css">
        <!-- <script defer src="../static/js/main.js"></script> -->
        <title>Virtual Eye</title>
```

```
</head>
<body>
          <header id="head" class="header">
         <section  id="navbar">
                <h1 class="nav-heading"></i>Virtual Eye</h1>
             <div class="nav--items">
                <ul>
                                            <li><a
href="{{ url_for('index')}}">Home</a></li>
                                            <li><a
href="{{ url_for('login')}}">Login</a></li>
                                            <li><a
href="{{ url_for('register')}}">Register</a></li>
                    <li><a href="{{ url_for('login')}}">Demo</a></li>
                </ul>
             </div>
         </section>
         <section id="slider">
   <div id="carouselExampleIndicators" class="carousel" data-ride="carousel">
              <ol class="carousel-indicators ">
          <li      data-target="#carouselExampleIndicators"      data-slide-      to="0"
class="active "></li>
          <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
          <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
              </ol>
              <div class="carousel-inner">


      <div class="carousel-item active">
   <img class="d-block w-100" src="../static/img/1.png"alt="First slide">
```

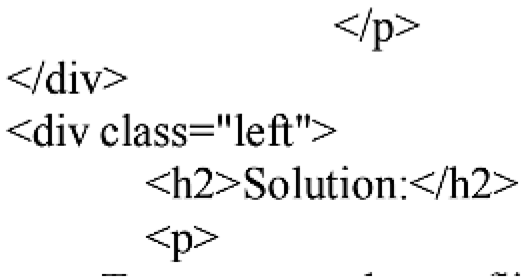


```
                </div>
                <div class="carousel-item">
              <img    class="d-block    w-100"    src="../static/img/second.jpg"
alt="Second slide">
                </div>
                <div class="carousel-item">
              <img class="d-block w-100" src="../static/img/third.jpg"alt="Third slide">
                </div>
              </div>
        <a      class="carousel-control-prev"      href="#carouselExampleIndicators"
role="button" data-slide="prev">
          <span         class="carousel-control-prev-icon"         aria-
hidden="true"></span>
                <span class="sr-only">Previous</span>
              </a>
        <a      class="carousel-control-next"      href="#carouselExampleIndicators"
role="button" data-slide="next">
          <span         class="carousel-control-next-icon"         aria-
hidden="true"></span>
                <span class="sr-only">Next</span>
</a>
</div>

           </section>
</header>
<section id="about">
          <div class="top">
              <h3    class="title    text-muted">
                ABOUT PROJECT
              </h3>
```



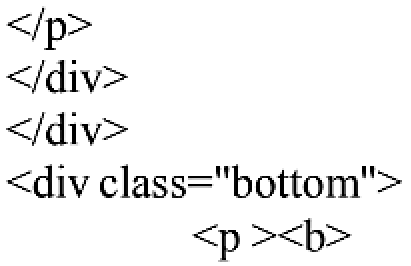
```
                <div class="line"></div>
            </div>
    <div class="body">
    <div class="left">
            <h2>Problem:</h2>
            <p>
```

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 the hotels, weekend tourist spots and barelypeople have in their house backyard. Beginners, especially oftenfeel it difficult to breathe under water and causes breathing trouble which in turn cause 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 besuffering 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.



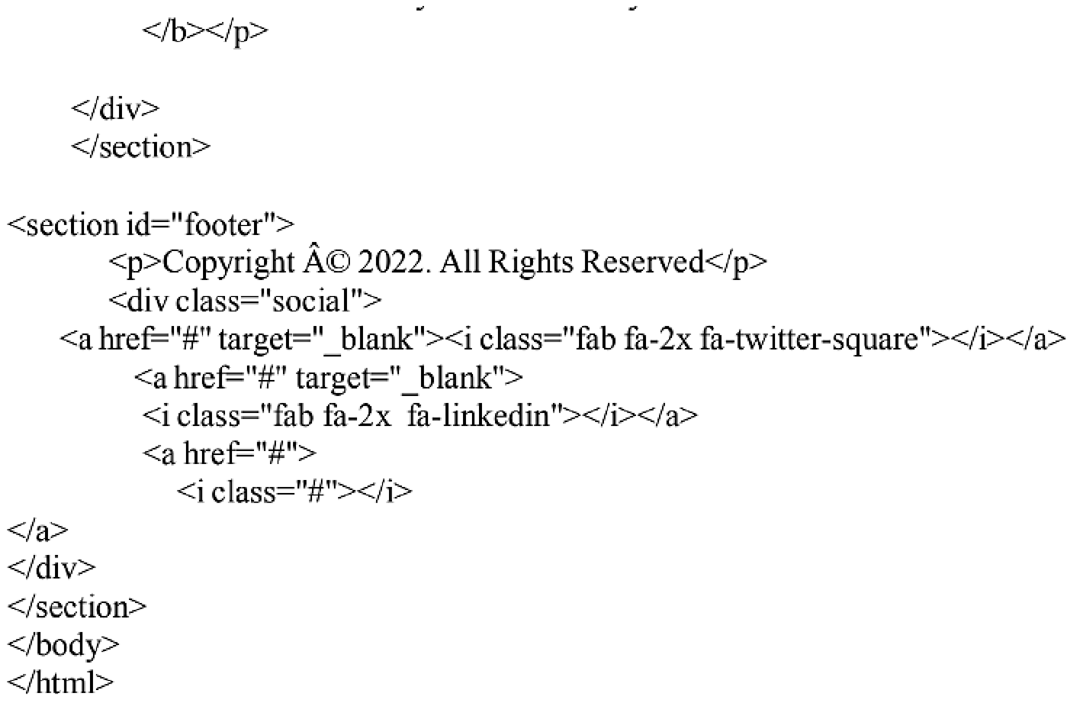
```
                          </p>
</div>
<div class="left">
          <h2>Solution:</h2>
          <p>
```

To overcome the conflict, a meticulous system is to be implemented along the swimming pools to save the human life. Bystudying body movement patterns and connecting cameras to an artificial intelligence (AI)system we can devise an underwater poolsafety 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 thanan alert will be generated to attract lifeguards attention.



```
</p>
</div>
</div>
<div class="bottom">
              <p ><b>
```

Note : The system is not designed to replace a lifeguard or other human monitor, but to act as an additional tool. â€œIt helpsthe lifeguard to detect the underwater situation where they canâ€™t easily observe.



```
                </b></p>

        </div>
        </section>

  <section id="footer">
           <p>Copyright Â© 2022. All Rights Reserved</p>
           <div class="social">
      <a href="#" target="_blank"><i class="fab fa-2x fa-twitter-square"></i></a>
              <a href="#" target="_blank">
               <i class="fab fa-2x  fa-linkedin"></i></a>
               <a href="#">
                  <i class="#"></i>
  </a>
  </div>
  </section>
  </body>
  </html>
```

Logout.html

```
<!DOCTYPE html>
<html >

<head>
        <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Virtual Eye</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>


<link href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>

<style>
.header {
                     top:0; margin:0px;
```

```
                   left:   0px;
                   right: 0px;
                   position: fixed;
                   background-color:     #28272c;
                   color: white;
                   box-shadow: 0px 8px 4px grey;
                   overflow: hidden;
                   padding-left:20px;
                   font-family:  'Josefin  Sans';
                   font-size: 2vw;
                   width:  100%;
                   height:8%;
                   text-align: center;
              }
              .topnav        {
     overflow:           hidden;
     background-color: #333;
}

    .topnav-right  a  {
     float:  left;  color:
     #f2f2f2;
     text-align:          center;
     padding: 14px 16px; text-
     decoration:  none;  font-
     size: 18px;
}
```

```
        .topnav-right         a.active         {
         background-color:        #565961;
         color: white;
}

        .topnav-right    {
         float: right;
         padding-right:100px;
}

.login{
margin-top:-70px;
}
body {

         background-color:#ffffff;
         background-repeat:          no-repeat;
         background-size:cover;
         background-position: 0px 0px;
         }
.main{
                margin-top:100px;
                text-align:center;
}
form { margin-left:400px;margin-right:400px;}

input[type=text], input[type=email],input[type=number],input[type=password] {
width: 100%;
               .topnav-right       a:hover       {
                background-color:       #ddd;
                color: black;
       }
```

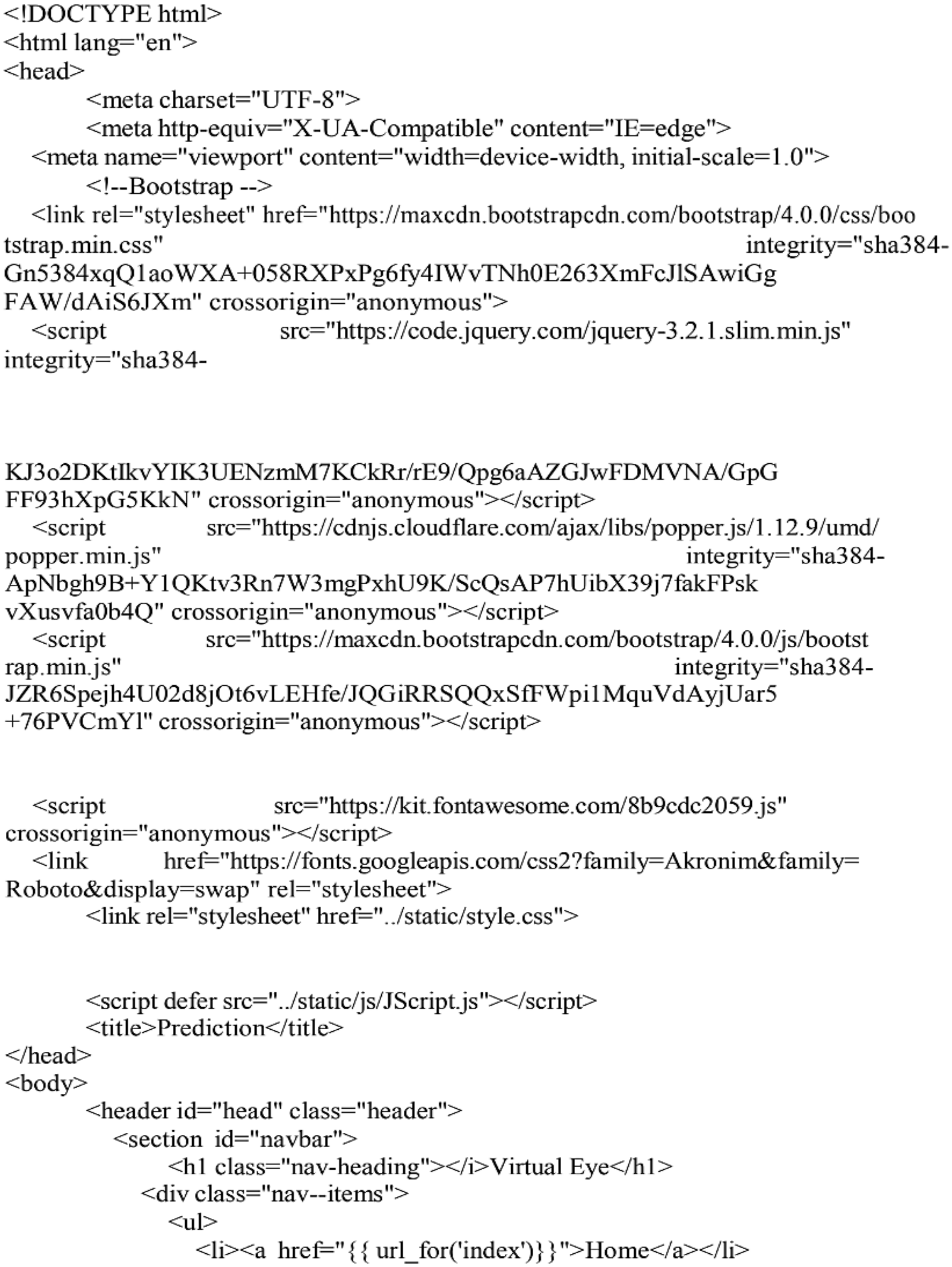
```
    .topnav-right       a.active       {
     background-color:      #565961;
     color: white;
}

    .topnav-right    {
     float: right;
     padding-right:100px;
}

.login{
margin-top:-70px;
}
body {

     background-color:#ffffff;
     background-repeat:         no-repeat;
     background-size:cover;
     background-position: 0px 0px;
     }
.main{
         margin-top:100px;
         text-align:center;
}
form { margin-left:400px;margin-right:400px;}

input[type=text], input[type=email],input[type=number],input[type=password] {
width: 100%;
```

```
        box-sizing: border-box;
}

button {
        background-color:       #28272c;
        color: white;
        padding:   14px   20px;
        margin-bottom:8px;
        border:   none;   cursor:
        pointer; width: 20%;
}

      button:hover     {
        opacity: 0.8;
}

      .cancelbtn           {
        width: auto;
        padding:         10px         18px;
        background-color: #f44336;
}

      .imgcontainer  {  text-
        align: center;
        margin: 24px 0 12px 0;
}

      img.avatar     {
        width: 30%;
        border-radius: 50%;
}
```

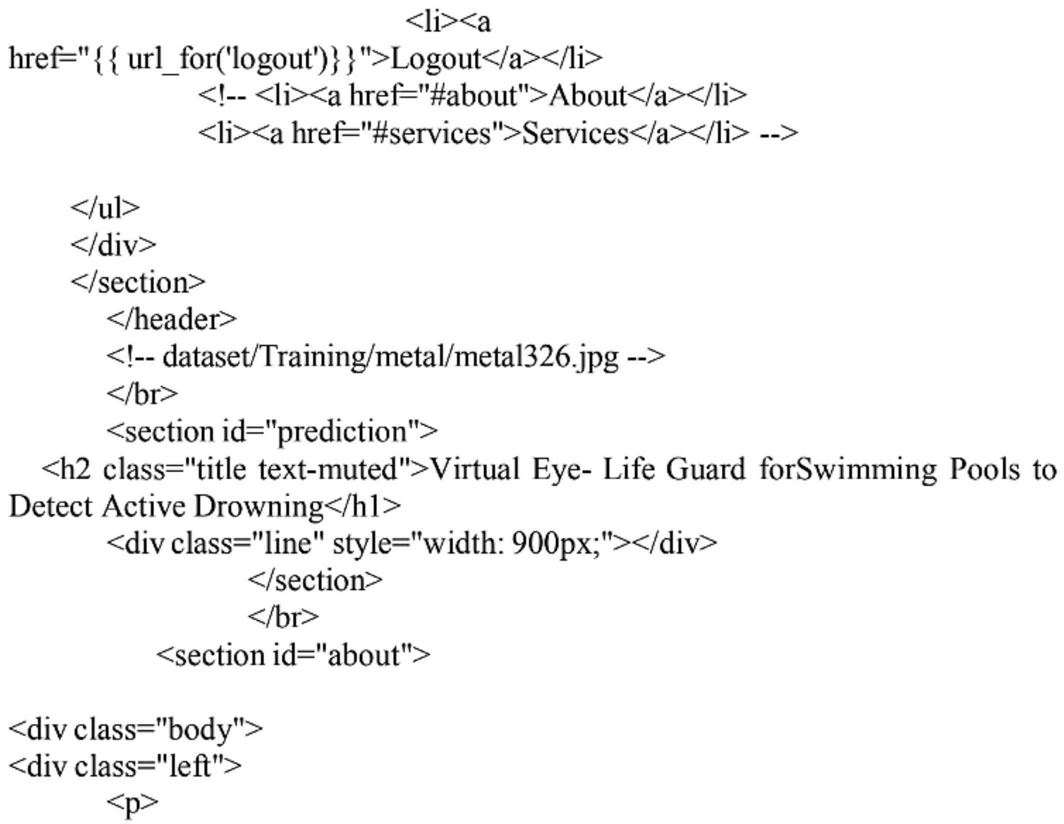
```
        .container          {
          padding: 16px;
}

        span.psw      {
          float: right;
          padding-top: 16px;

}

/* Change styles for span and cancel button on extra small scree
*/
        @media   screen   and   (max-width: 300px) {
          span.psw {
             display:    block;
             float: none;
          }
          .cancelbtn          {
             width: 100%;
          }
}

</style>
</head>

<body style="font-family:Montserrat;">
```

```
<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text- align:left;color:white; padding-
top:1%">Virtual eye</div>
      <div class="topnav-right" style="padding-top:0.5%;">

        <a  href="{{ url_for('home')}}">Home</a>








           <a href="{{ url_for('login')}}">Login</a>
           <a href="{{ url_for('register')}}">Register</a>
         </div>
  </div>
  <div class="main">
  <h1>Successfully Logged Out!</h1>
  <h3 style="color:#4CAF50">Login for more information<h3>

         <a     href="{{     url_for('login')     }}"><button
  type="submit">Login</button></a>
  </form>
  </div>

  </body>
  </html>
```

Prediction.html



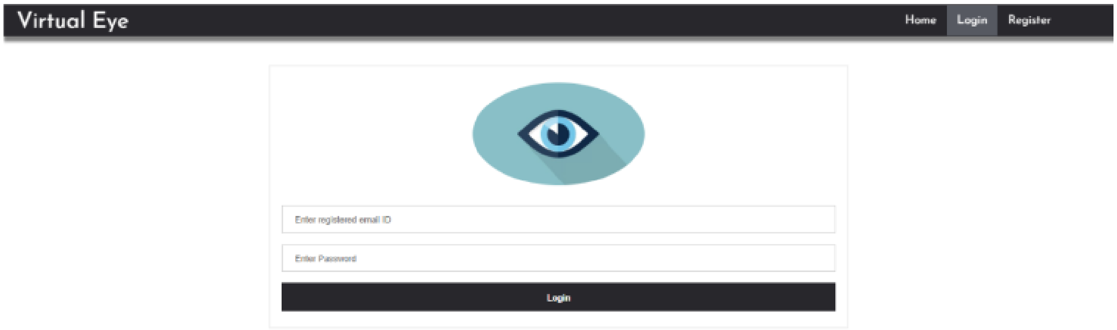
```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <!--Bootstrap -->
   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo
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>
        <header id="head" class="header">
           <section  id="navbar">
                 <h1 class="nav-heading"></i>Virtual Eye</h1>
              <div class="nav--items">
                 <ul>
                    <li><a  href="{{ url_for('index')}}">Home</a></li>
```



```
                                        <li><a
href="{{ url_for('logout')}}">Logout</a></li>
                    <!-- <li><a href="#about">About</a></li>
                    <li><a href="#services">Services</a></li> -->

      </ul>
      </div>
      </section>
          </header>
          <!-- dataset/Training/metal/metal326.jpg -->
          </br>
          <section id="prediction">
   <h2 class="title text-muted">Virtual Eye- Life Guard forSwimming Pools to
Detect Active Drowning</h1>
          <div class="line" style="width: 900px;"></div>
                         </section>
                         </br>
               <section id="about">

<div class="body">
<div class="left">
          <p>
```

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 the hotels, weekend tourist spots and barelypeople have in their house backyard. Beginners, especially oftenfeel it difficult to breathe under water and causes breathing trouble which in turn cause 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 besuffering 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.

```
                            </p>
</div>
<div class="left">

            <div class="prediction-input">
        <img class="d-block w-100" src="../static/img/second.jpg"alt="Second slide">
                </br>
                <form id="form" action="/result" method="post"
enctype="multipart/form-data">

                    <input type="submit" class="submitbtn" value="ClickMe! For a
Demo">
                          </form>
                    </div>
                    <h5 style="text-color:Red">
                    <b style="text-color:Red">{{prediction}}<b>
                  </h5>
</div>
</div>
</section>

              </br></br>



            <section id="footer">
                <p>Copyright Â© 2021. All Rights Reserved</p>

            </section>
</body>

</html>
```


Virtual Eye
Home
Login
Register
Enter registered email ID
Enter Password
Login

CHAPTER-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 financal

(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

CONCLUSION

This section will draw from three core documents: ISO_20380, HSG179, and the recently published German guideline, DGfdB R 94.15. A summary of each is given, outlining the key messages they disseminate and what this means for those involved with DDS.

ISO_20380 This document focuses on the requirements for the installation, operation, maintenance and performance of DDS, the testing methods, and the information required from the supplier in the operating manual. These international standards do not apply to systems used in domestic pools or pools smaller than 150m2 .

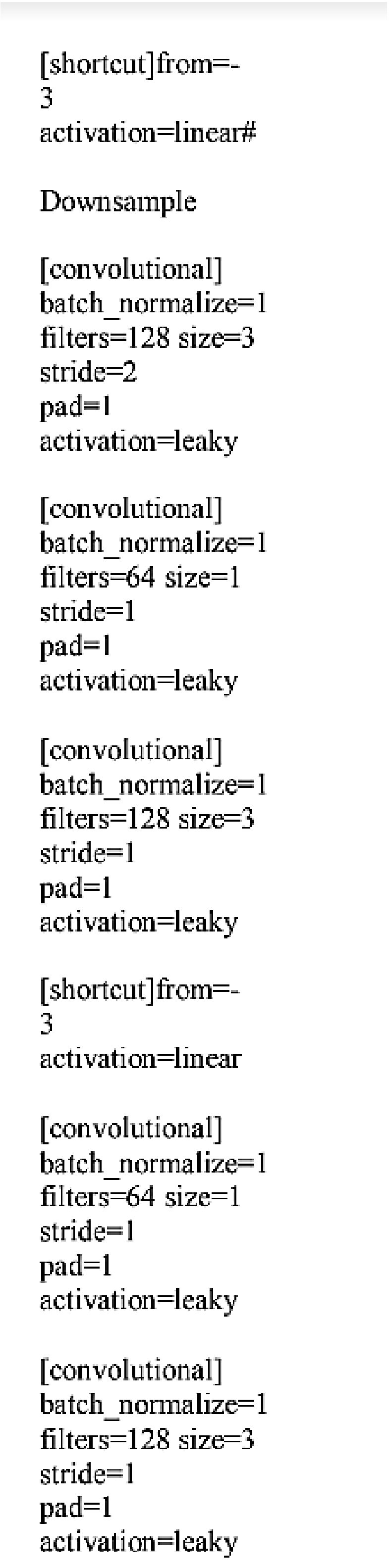
Prior to the installation of any DDS, ‘a technical study shall be carried out by the supplier in consultation with or based on information provided by the swimming pool’s owner/operator’. This is to establish the quantity and positioning of the equipment making up the system such as cameras, central processing unit, alarm tools, and other related equipment. The technical study must also provide a technical drawing of the pool basin, showing areas of ‘coverage’ and ‘non-coverage’, as well as the minimum lighting levels required above and below the water surface for the DDS to operate within performance requirements. To carry out the study, a list of factors to consider are given, outlining the variables that make each pool unique such as the architecture, and alarm reception coverage area of mobile devices to be used with the system. With this information all in one document, the technical study can be used to help optimise performance of the system, and forms part of the contract between the supplier and the pool operator. The next area of the standard is the performance requirements. This outlines the requirements needed to pass the regular maintenance testing and performance requirements for normal operation. This section covers the alarm set off time for operational performance, which is to be 15 seconds or less and displayed on the system interface. It also states that the alarm set off time must be built-in and shall not be changeable by staff. The section also discusses the areas covered by the DDS and highlights that each trained staff member must be aware of these areas. Another coverage-related requirement is that the DDS must be able to temporarily create areas where detection is disabled, to manage specific activities such as rescue drills.

CHAPTER-12

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 at the presence of better-quality hardware, could affect 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 clientside 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 centralized system that watches over the decentralized 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



```
[shortcut]from=-
3
activation=linear#

Downsample

[convolutional]
batch_normalize=1
filters=128 size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky
```

```
activation=linear

# Downsample

[convolutional]
batch_normalize=1
```

```
filters=128 size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky

[shortcut]from=-
3
activation=linear
```

```
filters=128 size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky

[shortcut]from=-
3
activation=linear
```

```
[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky

[shortcut]from=-
3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
```

```
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256 size=3
```

```
stride=1
pad=1
activation=leaky

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

[shortcut]from=-
3
activation=linear


[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

[shortcut]from=-
3
activation=linear
```

```
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

[convolutional]
```

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

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

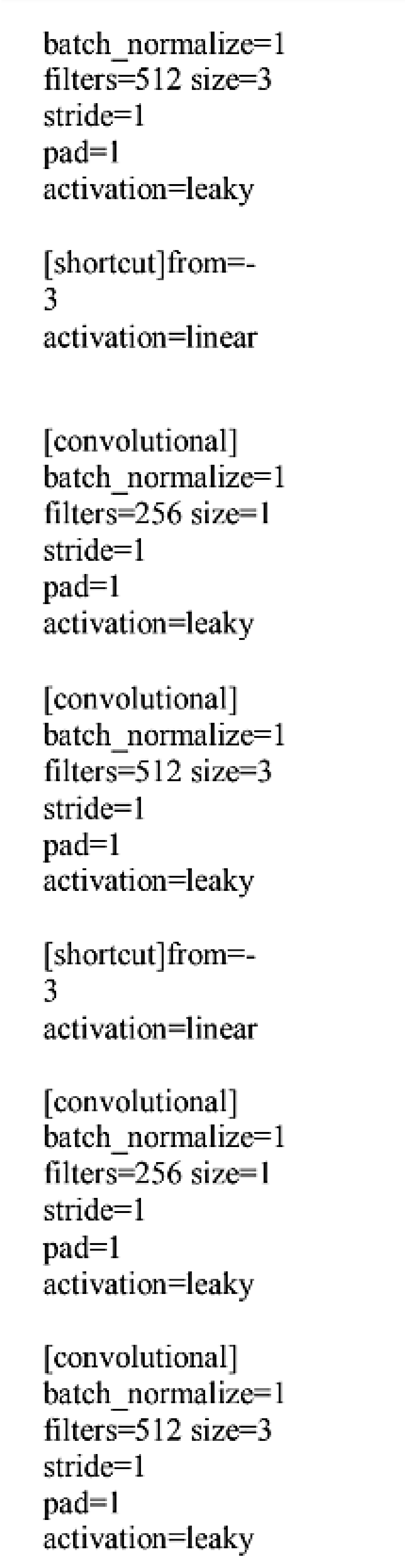
```
[shortcut]from=-
3
activation=linear#

Downsample

[convolutional]
batch_normalize=1
filters=512 size=3
stride=2
```

```
pad=1 activation=leaky

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

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

[shortcut]from=-
3
activation=linear


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

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

[shortcut]from=-
3
activation=linear
```

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

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

```
pad=1 activation=leaky

[shortcut]from=-
3
activation=linear


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

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

[shortcut]from=-
3
activation=linear

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

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

[shortcut]from=-
3
activation=linear
```

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

[convolutional]
```



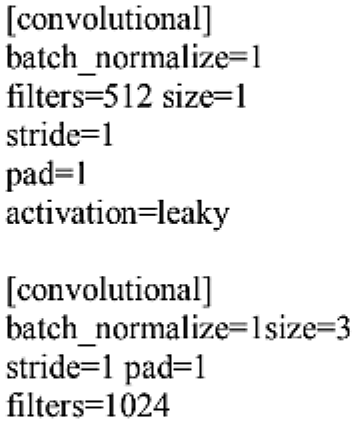
```
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky

[shortcut]from=-
3
activation=linear


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

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

[shortcut]from=-
3
activation=linear

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

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

```
[shortcut]from=-
3
activation=linear#

Downsample

[convolutional]
batch_normalize=1
filters=1024 size=3
```

```
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky

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

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky

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

[shortcut]from=-
3
activation=linear
```

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

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

```
activation=leaky

[shortcut]from=-
3
activation=linear

[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky

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

[shortcut]from=-
3
activation=linear

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

[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky

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



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

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

```
activation=leaky

[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky

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

[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear


[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

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

[upsample]
stride=2
```

```
[route]
layers = -1, 61


[convolutional]
```

```
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky

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

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

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

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

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

[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
```

```
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky

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

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

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

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

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

[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
```

```
[yolo]
mask = 3,4,5
```

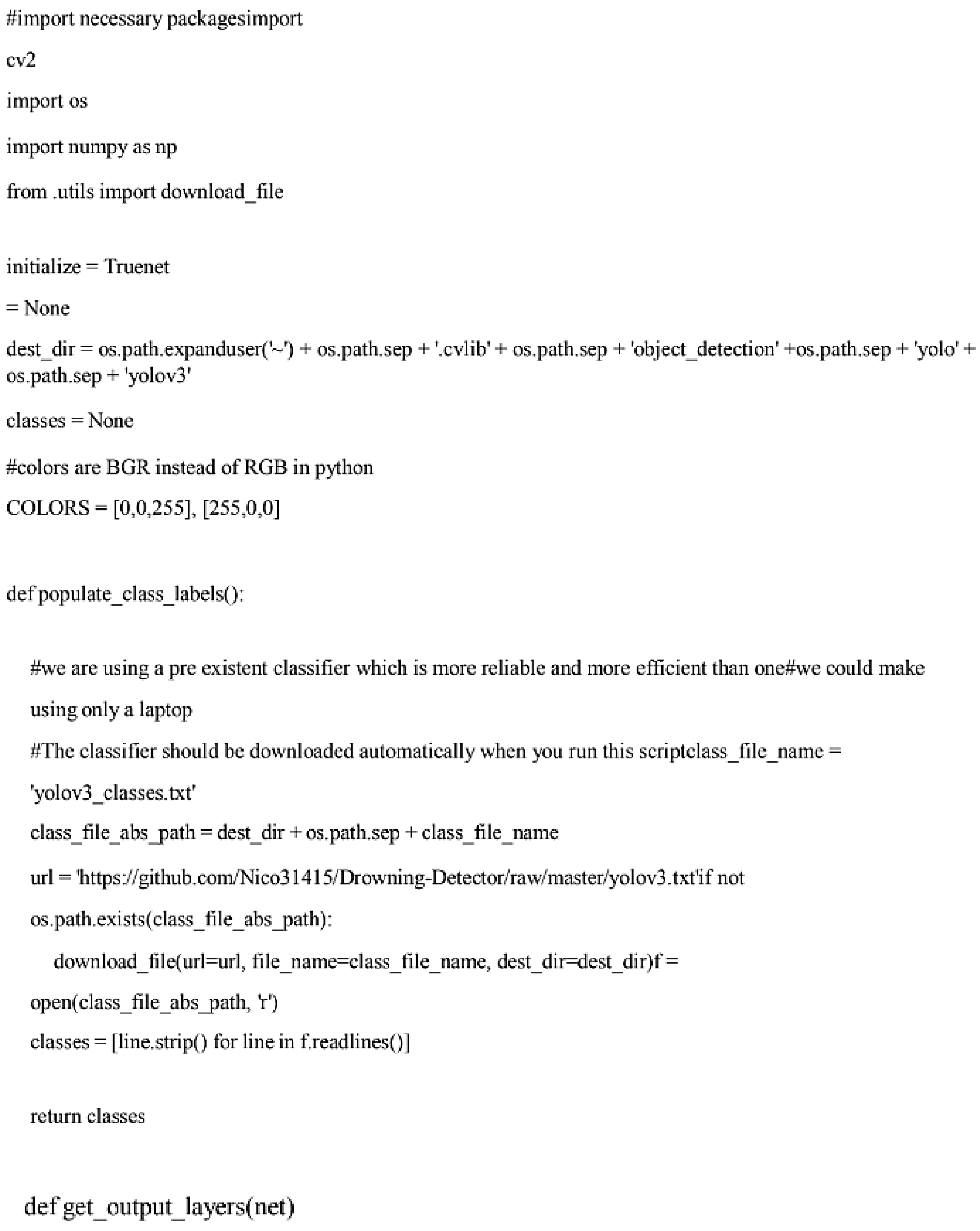
```
anchors = 10,13,  16,30,  33,23,  30,61,   62,45,   59,119,   116,90,
156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1random=1


[route] layers = -4

[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 36


[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

```
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

[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 36



[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

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

[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

```
pad=1 filters=256
activation=leaky

[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

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

[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear


[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=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 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)
```

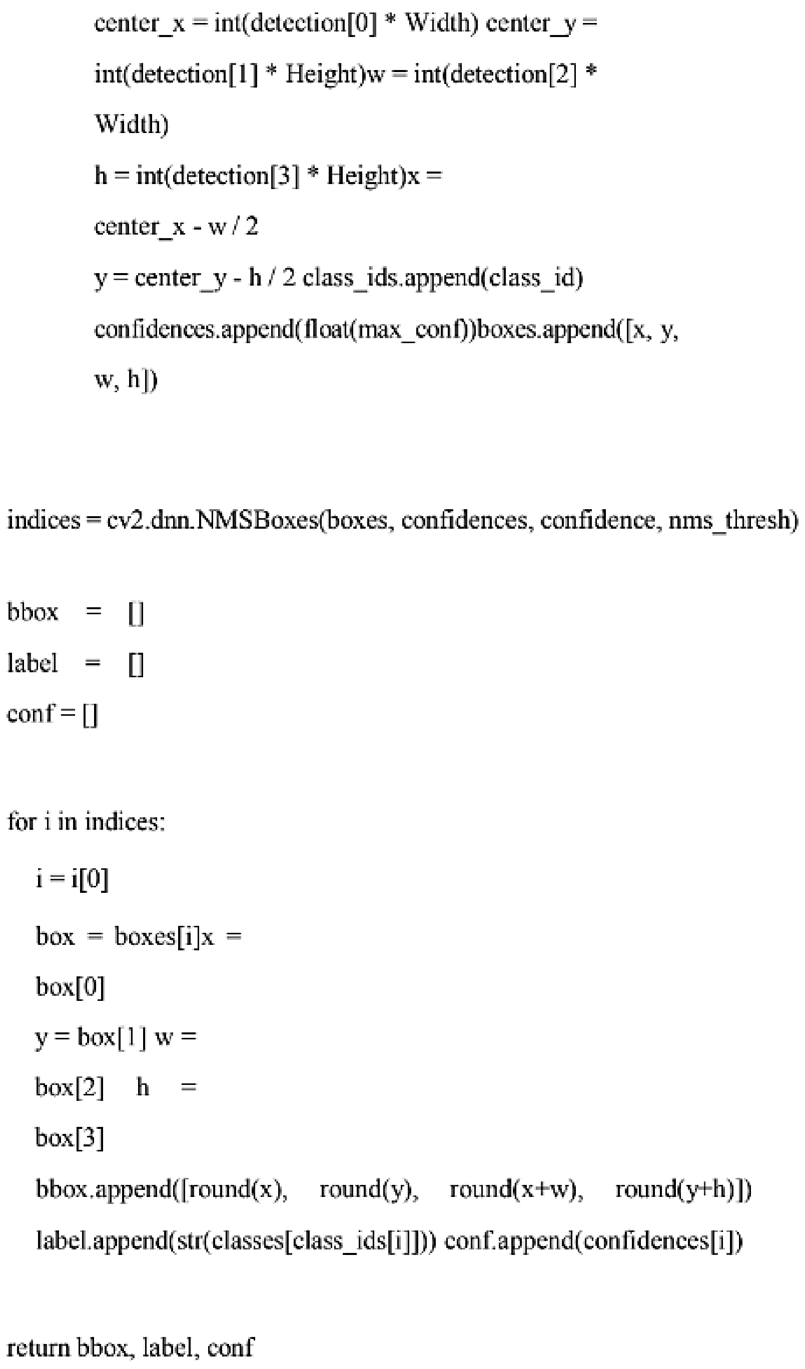
```
#the number of output layers in a neural network is the number of possible#things the network
can detect, such as a person, a dog, a tie, a phone... layer_names = net.getLayerNames()

   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')) + '%'
```

```
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 = []
confidences = []
boxes = []
```

```
for out in outs:
    for detection in out: scores =
        detection[5:]
        class_id = np.argmax(scores)
        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
```



```
          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:

Link

Demo Link