

A decorative graphic featuring three blue circles of varying sizes. Two circles are in the upper right quadrant, and one is in the lower right quadrant. Two thin blue lines originate from the top left and extend diagonally towards the circles, one ending near the top-right circle and the other near the middle-right circle.

# Final Documentation

Virtual Eye –Life Guard for Swimming Pools to Detect Active Drowning

**TEAMID: PNT2022TMID36512**

## **Team Members**

Rithick Nathan.R[Team leader]

Ajith Kumar.D

Aswanth.E

Naveen.P

# CONTENT

## **1. INTRODUCTION**

1.1 Project Overview

1.2 Purpose

## **2. LITERATURE SURVEY**

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

## **4. REQUIREMENT ANALYSIS**

4.1 Functional requirement

4.2 Non-Functional requirements

## **5. PROJECT DESIGN**

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

## **6. PROJECT PLANNING & SCHEDULING**

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

## **7. CODING**

## **8. TESTING**

8.1 Test Cases

8.2 User Acceptance Testing

## **9. RESULTS**

9.1 Performance Metrics

## **10. ADVANTAGES & DISADVANTAGES**

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

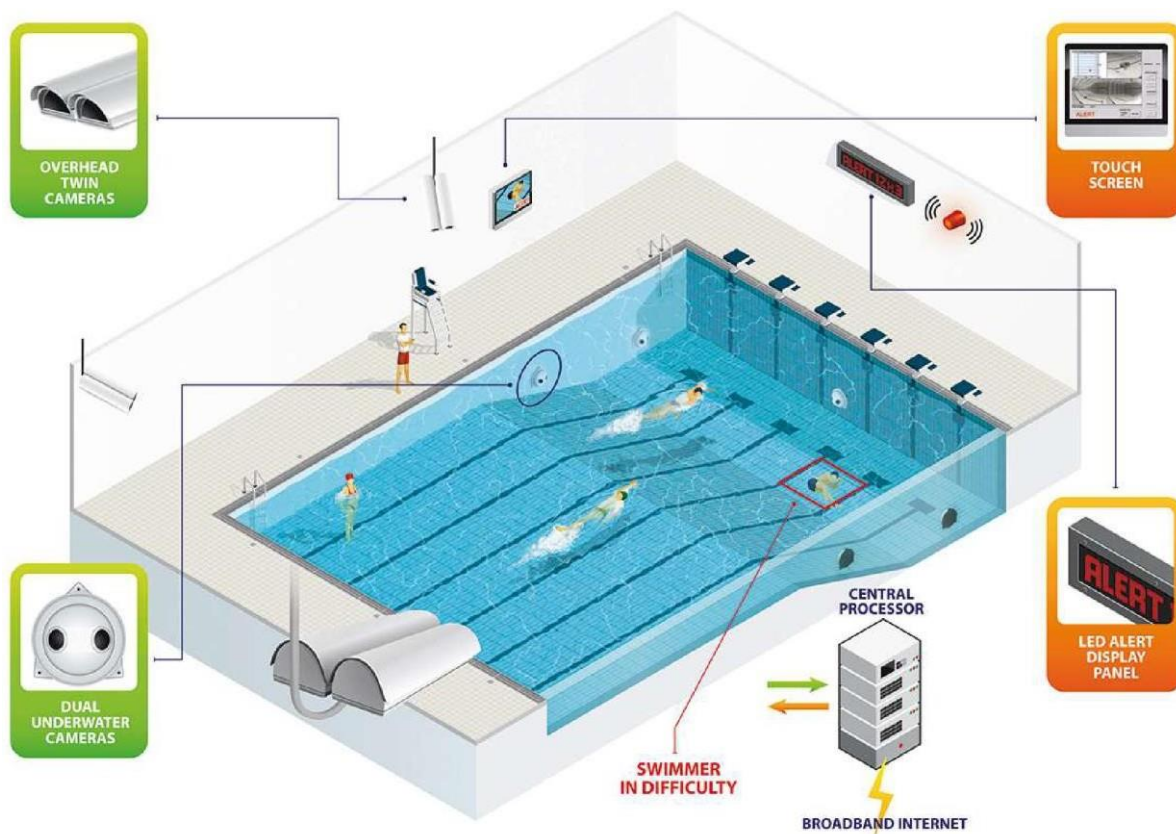
## **13. APPENDIX**

Source Code

GitHub & Project Demo Link

# 1. INTRODUCTION

## 1.1 Project overview:



1.2

## Purpose:

The system is not designed to replace a lifeguard or other human monitor, but to act as an additional tool. “It helps the lifeguard to detect the underwater situation where they can’t easily observe.

## 2. LITERATURE SURVEY

### 2.1 Existing problem

Whilst literature on DDS mostly agrees 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. There is debate around whether DDS can be helpful or harmful towards lifeguarding practices and how DDS may change the landscape of traditional lifeguarding, as well as some disagreement on whether they serve as justification for reducing lifeguard numbers. 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.

## 2.2 References

- ✓ AngelEye.(2019).AngelEye–Distributors.Retrievedfrom:  
<https://www.angeleye.it/news.php?id=28&newscat=10>
- ✓ 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](https://www.aquaticsintl.com/facilities/traumaticexperiences_o)
- ✓ 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>
- ✓ 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/?pid=000000000030360257>
- ✓ Drowning Prevention. (2017). The Need. Retrieved from:  
<https://www.drowningprevention.com.au/>
- ✓ 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).
- ✓ 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
- ✓ Health and Safety Executive. (2018). HSG179, Health and safety in swimming pools (Fourth edition).

## 2.3 Problem Statement 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 find it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.

3. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

### **3. IDEATION & PROPOSED SOLUTION**

In this paper we have proposed a method for automatic real-time detection of a person drowning in the swimming pools. The overview of the proposed algorithm in this paper is presented.

Our system is based on real time video analysis of the cameras installed around the swimming pool in a way which the entire swimming pool can be covered. Each camera is mounted on pool walls oriented downwards with a sharp angle, so that it can minimize the effect of lightening system which causes occlusions and foreshadowing. In this work, a 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 including the distributing system known as ODROID-XU, and our Logitech HD Pro C920 webcam used to record all the video sequences in this paper is illustrated in. 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 an 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.

# 3.1 Empathy Map Canvas





# 3.2 Ideation & Brainstorming

## Brainstorm & idea prioritization

In this session we aim to achieve a good base for beginning our project. With clear understanding of the task in hand, the next step would be to collectively put in our thoughts. Inspiration and exit with a proper feasibility study.

### Ground Rules

- Be Creative
- Rule out every possible ideas and improvements
- Make your points clear and purposeful
- Don't hesitate. (Every point is noteworthy)
- Arguments are good. All it has to be beneficial
- Have various perspectives towards the problem

### Choose your best 'How Might We?' Questions

Share the top 3 brainstorm questions that you created and either group members who help by voting one question forward with 40 points or who wants to be the most promising/likeable question or whoever you are trying to impact.

1. How might we create a system that can detect and respond to threats in real-time?

2. How might we create a system that can detect and respond to threats in real-time?

3. How might we create a system that can detect and respond to threats in real-time?

### Brainstorm solo

Have each participant begin in the 'Solo Brainstorm' card by clearly understanding the task and writing down the idea. The 'Solo Brainstorm' card is a template for each participant to write down their ideas and vote on the best one. (40 points for the best idea, 20 points for the second best idea, 10 points for the third best idea. Group members to go for the best idea.)

1. How might we create a system that can detect and respond to threats in real-time?

2. How might we create a system that can detect and respond to threats in real-time?

3. How might we create a system that can detect and respond to threats in real-time?

### Brainstorm as a group

Have members share their ideas into the 'Group Brainstorm' card. The 'Group Brainstorm' card is a template for each participant to write down their ideas and vote on the best one. (40 points for the best idea, 20 points for the second best idea, 10 points for the third best idea. Group members to go for the best idea.)

1. How might we create a system that can detect and respond to threats in real-time?

2. How might we create a system that can detect and respond to threats in real-time?

3. How might we create a system that can detect and respond to threats in real-time?

### Priority

Your ideas should all be in the same page about which idea is most important. (40 points for the best idea, 20 points for the second best idea, 10 points for the third best idea. Group members to go for the best idea.)

1. How might we create a system that can detect and respond to threats in real-time?

2. How might we create a system that can detect and respond to threats in real-time?

3. How might we create a system that can detect and respond to threats in real-time?

### Decide your focus

Use each member's votes to vote which idea would you have focus on during the ideation process.

1. How might we create a system that can detect and respond to threats in real-time?

2. How might we create a system that can detect and respond to threats in real-time?

3. How might we create a system that can detect and respond to threats in real-time?

### Team

1. How might we create a system that can detect and respond to threats in real-time?

2. How might we create a system that can detect and respond to threats in real-time?

3. How might we create a system that can detect and respond to threats in real-time?



## 3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Swimming pools are generally places of fun and healthy exercise, but they can be deadly as well. Even with a lifeguard observer on duty, swimmers may still have trouble in underwater or in parts of the pool beyond the lifeguard's field of view.
2.	Idea / Solution description	In this project, we use Artificial Intelligence. We install the cameras in underwater to detect the drowning people. Using deep learning, image can be recognized. If the image is detected, it triggers the alarm to alert the Life Guard who rescue the drowning people.
3.	Novelty / Uniqueness	The uniqueness of our system software to track the position and the location of a drowning person. We use YOLO Algorithm. Because of its high accuracy and fast detection speed. So it helps lifeguard to save people within seconds.

4.	Social Impact / Customer Satisfaction	Drowning globally has a higher death rate and is also the third leading cause of unexpected deaths worldwide, 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)	We can introduce the software based approach for making a good income. It is extremely useful to lifeguards, swimmers and business operators. The number of features makes it attractive for end users to use our software system.
6	Scalability of the Solution	Our software system can be used by the company driver who manages the pools. We use the IBM cloud server to collect and maintain then data. We will ensure the safety of the swimmers.

## 3.4 Problem Solution fit

<b>CUSTOMER SEGMENT</b> <ul style="list-style-type: none"> <li>Person who swim in the pool are ment to be constantly kept an eye over them by visual based monitoring system.</li> </ul>	<b>CUSTOMER LIMITATIONS</b> <ul style="list-style-type: none"> <li>Constant network connection</li> <li>Camera misunderstanding normal swimming actions to be abnormal.</li> <li>Cost of fitting and maintainance</li> </ul>	<b>AVAILABLE SOLUTIONS</b> <ul style="list-style-type: none"> <li>Setting up of camera and monitoring each and every person swimming in the pool setting an alarm to notify the Lifeguard</li> <li>Detects and prevents active drowning</li> </ul>
<b>JOBS TO BE DONE/PROBLEMS</b> <ul style="list-style-type: none"> <li>People visit the swimming pools to practice or to learn swimming. There is a possibility of someone drowning as they may be new to these activities.</li> <li>Existing visual based monitoring systems are too economical and these are needed to environmnet.</li> </ul>	<b>PROBLEM ROOT / CAUSE</b> <ul style="list-style-type: none"> <li>People think that the camera that is set up to monitor the persons who are swimming are of no proper and accurate use.</li> <li>Anticipation over all the other system happens when one device fails to do its service.</li> </ul>	<b>BEHAVIOUR</b> <ul style="list-style-type: none"> <li>The customer believes more in a manual monitoring system rather than a visual monitoring system</li> <li>He/she want to be always surrounded by a lifeguard rather being monitored by a camera</li> </ul> <p>The customer will exhibit his behaviour until an authenticat-ed application serves its purpose rightly</p>
<b>TRIGGERS TO ACT</b> <ul style="list-style-type: none"> <li>The customer is triggered by their surrounding talking about this approach of detecting and preventing active drowning.</li> <li>Economical installation cost also plays a pivotal role.</li> </ul> <b>EMOTIONS before /after</b> <ul style="list-style-type: none"> <li><b>BEFORE</b> : Fear of unprotected swimming</li> <li><b>AFTER</b> : Fearless and satisfactory swimming experiences</li> </ul>	<b>YOUR SOLUTION</b> <ul style="list-style-type: none"> <li>The proposed system makes a novel attempt to evaluate swimmers condition by analyzing their motion and shape features via visual based monitoring device and an alarm to alert, and provides solution in detecting drowning incidents.</li> <li>While challenging in many aspects, a successful system will bring inestimable value in saving human lives.</li> </ul>	<b>CHANNELS OF BEHAVIOUR</b> <b>ONLINE</b> <ul style="list-style-type: none"> <li>Develop an application and provide all sort of assistance to the users regarding the virtual eye.</li> </ul> <b>OFFLINE</b> <ul style="list-style-type: none"> <li>Provide quality safety wares while swimming</li> </ul>

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

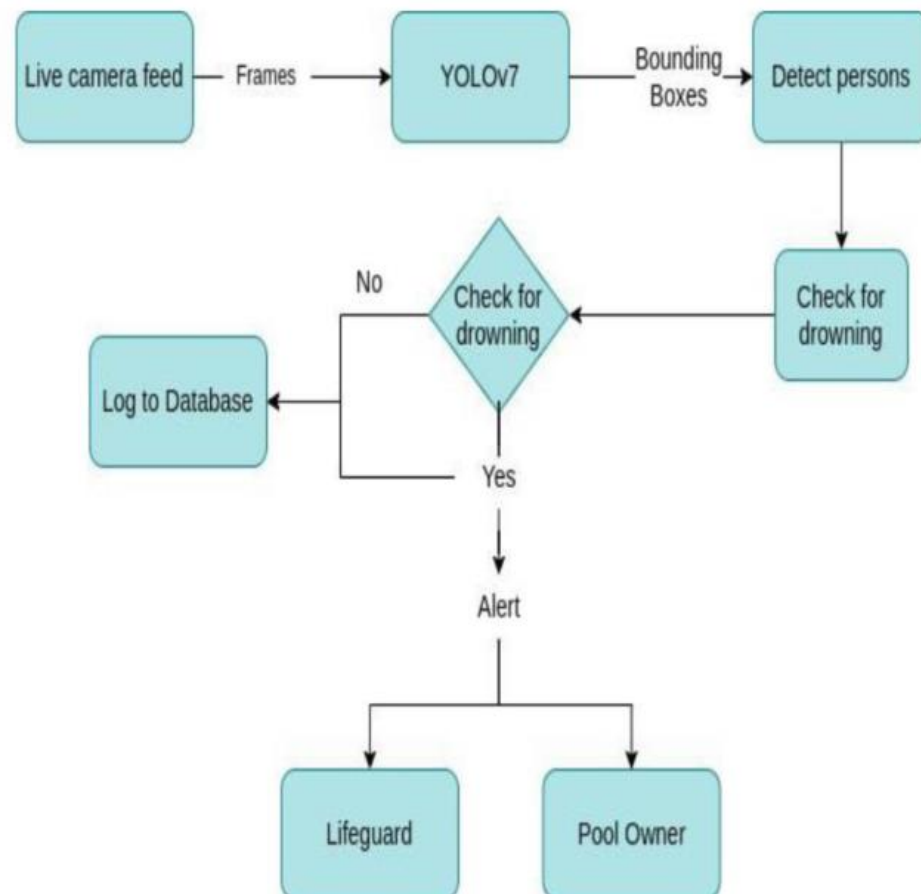
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Installation	Needed to be fixed under the water without creating any disturbance to the people in the swimming pool.
FR-2	Detection	Detecting the human bodies and counting.
FR-3	Audio	Ask for help or stay quiet if the person is unconscious.
FR-4	Support	Take swim tubes or take the help of rescue.
FR-5	Pulse Rate Sensor	A pulse sensor on that <b>measures the changes in light absorption and reflection on the skin to measure blood flow.</b>
FR-6	PriorAlert	Send alert message to the lifeguard.

## 4.2 Non-Functional requirements

<b>FR No.</b>	<b>Non-Functional Requirement</b>	<b>Description</b>
NFR-1	<b>Usability</b>	To ensure the safety of each and every person present in the pool. A Lifeguard should be present all the time in the pool.
NFR-2	<b>Security</b>	Lifeguards should be aware of the alert message to save the life of the swimmer.
NFR-3	<b>Reliability</b>	Virtual eye lifeguard trigger an immediate prior alarm if a swimmer is in peril, helping to avoid panic even in critical situations.
NFR-4	<b>Performance</b>	The alarm is triggered when the swimmer's pulse rate is decreasing.
NFR-5	<b>Availability</b>	Equipment and accessories include lifesaver rings, inflatable vests, a Shepherd's Crook, life hooks, spine boards, rescue tubes, and a first aid kit. Remember to keep them accessible to quickly pull Someone from the water safely.
NFR-6	<b>Scalability</b>	It detects potential drowning and promptly notifies you. It features the latest artificial intelligence technology and adapts the needs of the user.

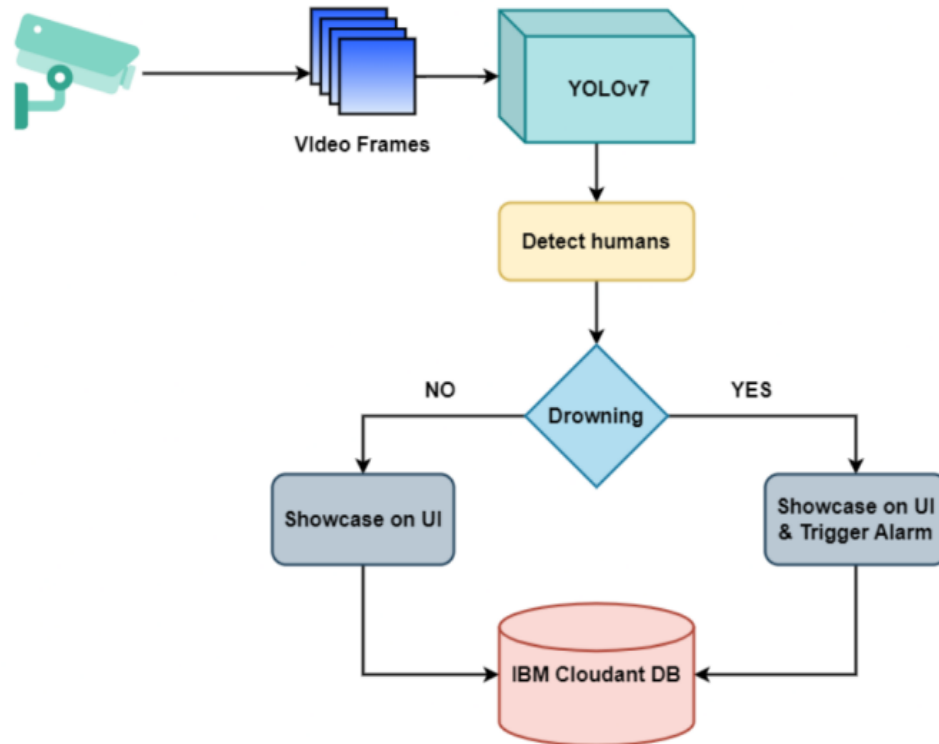
## 5. Project Design

### 5.1 Data Flow Diagrams



## 5.2 Solution & Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2





**Table-1 : Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	How user interact with the application	HTML, JavaScript, CSS
2.	Application Logic-1	Extracting frames from live video feed	Python
3.	Application Logic-2	Person Detection	Python
4.	Application Logic-3	Detect drowning	Python
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM Cloudant DB
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Machine Learning Model	Detect humans	Object Detection Model (YOLOv7)
9.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	Cloud Foundry, Docker.

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Anaconda Navigator, PyTorch, Flask,	Technology of Open-source framework
2.	Security Implementations	Security / access controls	IAM Controls
3.	Scalable Architecture	Whether demand increases gradually or abruptly, scalable web architecture can accommodate any load without compromising the application's integrity.	Microservices, Progressive Web Apps (PWA)
4.	Availability	Availability of applications like load balancers, distributed servers etc.	IBM Cloud

## 5.3 Users Stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	VLGFSP - 1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-1	Registration	VLGFSP - 2	As a user, I will receive confirmation email once I have registered for the application	1	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-1	Registration	VLGFSP - 3	As a user, I can register for the application through Facebook	2	Low	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-1	Registration	VLGFSP - 4	As a user, I can register for the application through Gmail	2	Medium	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-1	Login	VLGFSP - 6	As a user, I can log into the application by entering email & password	1	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-2	Dataset Collect	VLGFSP - 11	Collect number of datasets and get accuracy	2	Medium	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-2	Pre-processing	VLGFSP - 12	The dataset is extracted	2	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-2	Train the Model	VLGFSP - 13	Train the Model	4	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-2	Test the Model	VLGFSP - 14	Test the Model	6	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-3	Detection	VLGFSP - 15	Load the trained model	3	High	Rithick Nathan Ajith Kumar Aswanth Naveen

<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
Sprint-3	Detection	VLGFSP - 16	Identify the person by collecting real-time data through a webcam	5	Medium	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-3	Detection	VLGFSP - 16	Classify it by using a trained model to project the output	8	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-4	Detection	VLGFSP - 17	If person is drowning, the system will ring an alarm to give signal	7	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-4	Detection	VLGFSP - 18	As a user, I can detect the drowning person	3	Medium	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-4	Logout	VLGFSP - 19	As a user, I can logout the application	2	Low	Rithick Nathan Ajith Kumar Aswanth Naveen

## 6.PROJECT PLANNING & SCHEDULING

### 6.1 SPRINT PLANNING & ESTIMATION

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual)</b>
Sprint-1	8	2 Days	31 Oct 2022	02 Nov 2022	2	04 Nov 2022
Sprint-2	14	2 Days	05 Nov 2022	07 Nov 2022	2	09 Nov 2022
Sprint-3	16	2 Days	09 Nov 2022	11 Nov 2022	2	13 Nov 2022
Sprint-4	12	2 Days	14 Nov 2022	16 Nov 2022	2	18 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	VLGFSP - 1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-1	Registration	VLGFSP - 2	As a user, I will receive confirmation email once I have registered for the application	1	High	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-1	Registration	VLGFSP - 3	As a user, I can register for the application through Facebook	2	Low	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-1	Registration	VLGFSP - 4	As a user, I can register for the application through Gmail	2	Medium	Rithick Nathan Ajith Kumar Aswanth Naveen
Sprint-1	Login	VLGFSP - 6	As a user, I can log into the application by entering email & password	1	High	Rithick Nathan Ajith Kumar Aswanth Naveen

## 6.3 Reports from JIRA

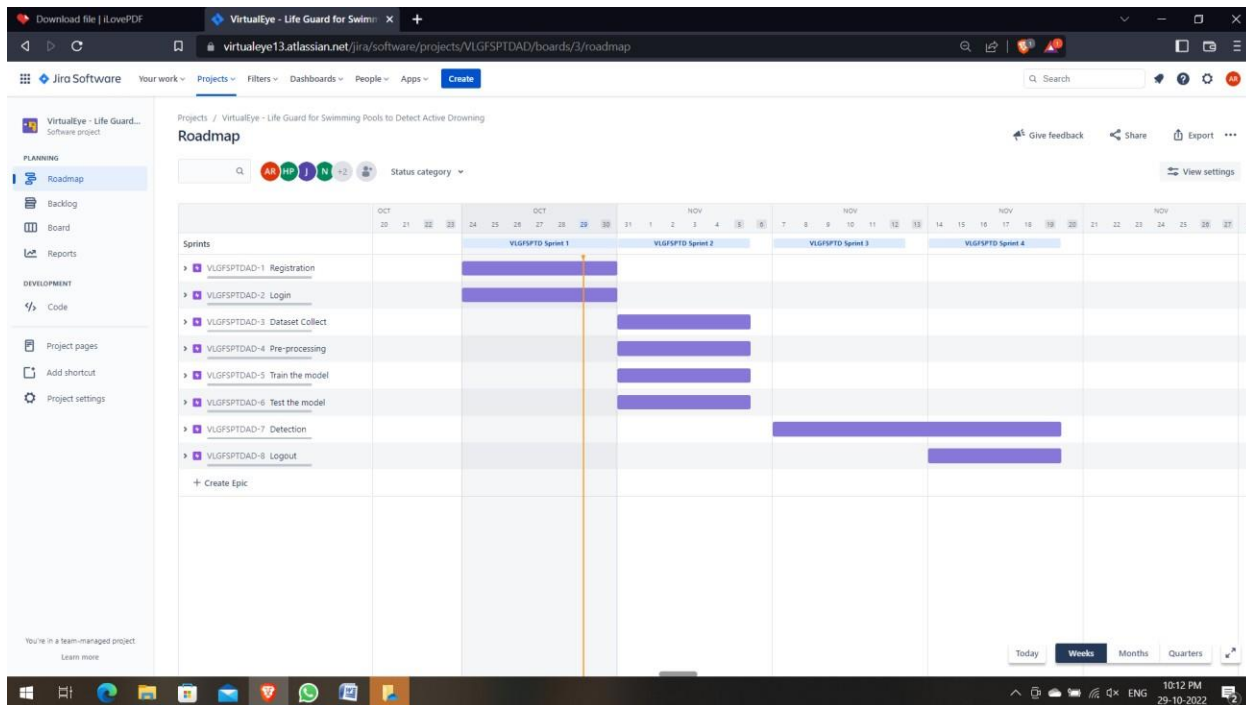
# Backlog

The screenshot displays the JIRA Backlog for the project 'VirtualEye - Life Guard for Swimmers'. The interface includes a left sidebar with navigation options like 'Backlog', 'Board', 'Reports', 'Code', 'Project pages', and 'Project settings'. The main area shows a list of issues organized into four sprints:

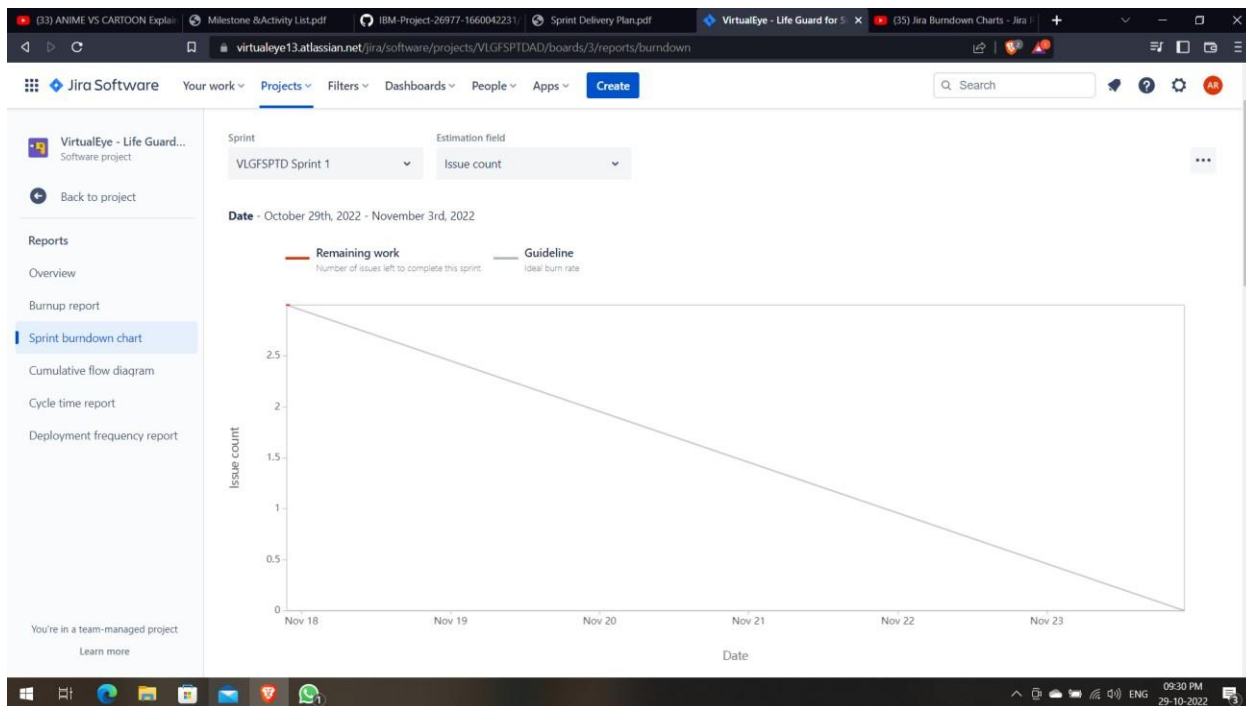
- Sprint 1 (24 Oct - 30 Oct, 3 issues):**
  - Issue 1: As a user, I can register for the application by entering my email, password, and confirming my password. **REGISTER**
  - Issue 2: As a user, I will receive confirmation email once I have registered for the application. **REGISTER**
  - Issue 3: As a user, I can register for the application through Facebook. **REGISTER**
- Sprint 2 (31 Oct - 6 Nov, 6 issues):**
  - Issue 4: Collect number of datasets and get accuracy. **ANALYZE DATASET**
  - Issue 5: The dataset is extracted. **PREPARE DATASET**
  - Issue 6: Train the model. **TRAIN THE MODEL**
  - Issue 7: Test the model. **TEST THE MODEL**
- Sprint 3 (7 Nov - 13 Nov, 3 issues):**
  - Issue 8: Load the trained model. **DETECTION**
  - Issue 9: Identify the person by collecting real-time data through a webcam. **DETECTION**
  - Issue 10: Identify it by using a trained model to predict the output. **DETECTION**
- Sprint 4 (14 Nov - 19 Nov, 3 issues):**
  - Issue 11: If person is drowning, the system will ring an alarm to give signal. **DETECTION**
  - Issue 12: As a user, I can detect the drowning person. **DETECTION**
  - Issue 13: As a user, I can logout the application. **LOGOUT**

At the bottom, there is a 'Backlog (0 issues)' section with a 'Create issue' button. The Windows taskbar at the bottom shows the time as 10:11 PM on 29-10-2022.

# Roadmap



# Burndown Chart



## 7. CODING

```
import time

import cv2
import numpy as np
from cloudant.client import Cloudant
from flask import Flask, request, render_template, redirect, url_for
from playsound import playsound

import cvlib as cv
from cvlib.object_detection import draw_bbox

# Loading the model

# Authenticate using an IAM API key
client = Cloudant.iam('8780b82a-5a3b-4da0-a180-a0e1516479f9-bluemix',
                      'TzYs8u0Q5eoj204gDo2eOEDAuGRhj0fG_9rlZr5SsJUH',
                      connect=True)

# Create a database using an initialized client
my_database = client.create_database('my_database')

app = Flask(__name__)

# default home page or route
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/index.html')
def home():
    return render_template("index.html")

# registration page
@app.route('/register')
def register():
    return render_template('register.html')

@app.route('/afterreg', methods=['POST'])
def afterreg():
    x = [x for x in request.form.values()]
    print(x)
    data = {
        '_id': x[1], # Setting _id is optional
        'name': x[0],
        'psw': x[2]
    }
    print(data)

    query = {'_id': {'$eq': data['_id']}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))
```



```

        if (len(docs.all()) == 0):
            url = my_database.create_document(data)
            # response = requests.get(url)
            return render_template('register.html', pred="Registration Successful,
please login using your details")
        else:
            return render_template('register.html', pred="You are already a member,
please login using your details")

# login page
@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/afterlogin', methods=['POST'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user, passw)

    query = {'_id': {'$eq': user}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if (len(docs.all()) == 0):
        return render_template('login.html', pred="The username is not found.")
    else:
        if ((user == docs[0][0]['_id'] and passw == docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')

@app.route('/logout')
def logout():
    return render_template('logout.html')

@app.route('/prediction')
def prediction():
    return render_template('prediction.html')

@app.route('/result', methods=["GET", "POST"])
def res():
    webcam = cv2.VideoCapture('drowning.mp4')

    if not webcam.isOpened():
        print("Could not open webcam")
        exit()

    t0 = time.time() # gives time in seconds after 1970

    # variable dcount stands for how many seconds the person has been standing
    still for
    centre0 = np.zeros(2)

```

```

isDrowning = False

# this loop happens approximately every 1 second, so if a person doesn't move,
# or moves very little for 10seconds, we can say they are drowning

# loop through frames
while webcam.isOpened():
    # read frame from webcam
    status, frame = webcam.read()

    if not status:
        print("Could not read frame")
        exit()

    # apply object detection
    bbox, label, conf = cv.detect_common_objects(frame)
    # simplifying for only 1 person

    # s = (len(bbox), 2)
    if (len(bbox) > 0):
        bbox0 = bbox[0]
        # centre = np.zeros(s)
        centre = [0, 0]
        # for i in range(0, len(bbox)):
        # centre[i] = [(bbox[i][0]+bbox[i][2])/2, (bbox[i][1]+bbox[i][3])/2 ]

        centre = [(bbox0[0] + bbox0[2]) / 2, (bbox0[1] + bbox0[3]) / 2]

        # make vertical and horizontal movement variables
        hmov = abs(centre[0] - centre0[0])
        vmov = abs(centre[1] - centre0[1])

        # there is still need to tweek the threshold
        # this threshold is for checking how much the centre has moved

        x = time.time()

        threshold = 10
        if (hmov > threshold or vmov > threshold):
            print(x - t0, 's')
            t0 = time.time()
            isDrowning = False

        else:

            print(x - t0, 's')
            if ((time.time() - t0) > 10):
                isDrowning = True

        # print('bounding box: ', bbox, 'label: ' label , 'confidence: '
        conf[0], 'centre: ', centre)
        # print(bbox, label , conf, centre)
        print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
        print('Is he drowning: ', isDrowning)

        centre0 = centre
        # draw bounding box over detected objects

    out = draw_bbox(frame, bbox, label, conf, isDrowning)

    # print('Seconds since last epoch: ', time.time()-t0)

    # display output

```

```
cv2.imshow("Real-time object detection", out)
if (isDrowning == True):
    playsound('alarm.mp3')
    webcam.release()
    cv2.destroyAllWindows()
    return render_template('prediction.html', prediction="Emergency !!! The
Person is drowning")
    # return render_template('base.html')

# press "Q" to stop
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# release resources
webcam.release()
cv2.destroyAllWindows()
# return render template('prediction.html',)

""" Running our application """
if __name__ == "__main__":
    app.run(debug=True)
```

## 8. TESTING

### 8.1 Test cases

Test case ID	Feature Type		Test Scenario	Steps TO Execute	Test	Expected Result	Actual Result
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup 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	1.Enter URL and click go 2. Click on My Account dropdown 3. Verify login/Signup popup with below UI elements: a.email text box b.password text box c. Login 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 customer? Create account link e. Last password? Recovery password link	Working as expected
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials	1.Enter URL and click 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_004	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 dropdown button 3. Enter Invalid username/email in Email text box 4. Enter valid password in password text box 5. Click on in 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	1-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_005	Functional	Login page	Verify user is able to into application with Invalid credentials	1. 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 in button	username: lax26@mail password: 1803	Application should show 'Incorrect email or password ' validation message.	working as
Predictionpage_TC_006	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

### Defect analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	2	0	0	2
Security	1	0	0	1
Outsource Shipping	1	0	0	1
Exception Reporting	2	0	0	2
Final Report Output	1	0	0	1

### Test analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
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

## 9. RESULTS

### 9.1 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</i></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>
        
    </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>
                    <button type="button" class="btn btn-info btn-lg "
id="btn-predict">Analyse</button>

```





<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/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.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</i></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">
            

```

```
        </div>
        <div class="carousel-item">
          
        </div>
        <div class="carousel-item">
          
        </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 barely people have in their house backyard. Beginners, especially often feel 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 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.

</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. By studying body movement patterns and connecting cameras to an artificial intelligence (AI) system 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 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. It 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: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%;
        padding: 12px 20px;
        display: inline-block;
        margin-bottom: 18px;
        border: 1px solid #ccc;
    }

```

```
        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 screens
*/
        @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-

```

```
KJ3o2DKtIkYIK3UENzmM7KCKRr/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</i></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 barely people have in their house backyard. Beginners, especially often feel 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 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.

```

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

    <div class="prediction-input">
        
        <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

HomeLoginRegister




Enter registered email ID

Enter Password

Login

Virtual Eye

HomeLoginRegisterDemo



ABOUT PROJECT

Problem:

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 barely people have in their house backyard. Beginners, especially often feel 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.

Solution:

To overcome the conflict, a meticulous system is to be implemented along the swimming pools to save the human life. By studying body movement patterns and connecting cameras to an artificial intelligence (AI) system 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.



## Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning

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 barely people have in their house backyard. Beginners, especially often feel 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 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.



[Click Me! For a Demo](#)

## Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning

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 barely people have in their house backyard. Beginners, especially often feel 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 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.



[Click Me! For a Demo](#)

**Emergency !!! The Person is drowning**

## **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

## 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 150m<sup>2</sup>.

- 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.

## 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 at the presence of better-quality hardware, could affect the speed and accuracy of the overall system is becoming a state-of-the-art.

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 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.

## 13. Appendix

### 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=400000,450000

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 = 1 random=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=1 size=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=1 size=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
packagesimportcv2

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
    makeusing 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
    notos.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
networkcan 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

```

Github link

<https://github.com/IBM-EPBL/IBM-Project-54447-1662022472>

Demo link

<https://drive.google.com/drive/folders/1S8QFmMeKThA8H03btPe1umMv5FgzKeq>



