VIRTUAL EYE – LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING

ABSTRACT

In modern metropolitan lifestyle, swimming is one of the best activities for stress reduction. Swimming pools are more prevalent at hotels and weekend tourist destinations, and hardly anyone has one in their backyard. Beginners, in particular, frequently find it challenging to breathe underwater, which leads to respiratory issues, which ultimately lead to a drowning disaster. In the world, children under the age of six are determined to have the highest rates of drowning fatality, with about 1.2 million cases occurring each year. A rigorous mechanism must be put in place beside the swimming pools to save human life in order to resolve this dispute. To tackle this issue, design an underwater pool safety system that lowers the chance of drowning by analysing body movement patterns and integrating cameras with artificial intelligence (AI). Employ a single camera that transmits underwater video while analysing swimmer positioning to determine the probability that they would drown. If the probability is higher than normal, an alarm will be sent to catch lifeguards' attention. The system is intended to be an additional tool rather than to take the place of a lifeguard or other human monitor. The lifeguard can use it to detect situations under water that are difficult for them to notice. Deep Learning Technology used for detecting object in a swimming pool using YOLO algorithm.

Table Of Contents P.NO

1. INTRODUCTION	
1.1 Project Overview	1
1.2 Purpose	2
2.LITERATURE SURVEY	
2.1 Existing Problem	3
2.2 References	8
2.3 Problem Statement Definition	9
3.IDEATION &PROPOSED SOLUTION	
3.1 Empathy Map Canvas	10
3.2 Ideation & Brainstroming	11
3.3 Proposed Solution	12
3.4 Problem Solution Fit	13
4. REQUIREMENT ANALYSIS	
4.1 Functional Requirement	14
4.2 Non- Functional Requirement	15
5. PROJECT DESIGN	
5.1 Data Flow Diagram	16
5.2 Solution & Technical architecture	20
6. TESTING	
6.1 Test Cases	22
6.2 User Acceptance testing	23
7. RESULTS	
7.1 Performance Metrics	24
8. ADVANTAGES & DIASADVANTAGES	25
9. CONCLUSION	26
10. FUTURE SCOPE	27
11.APPENDIX	
Source Code	28
Github & Project Demo link	41

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Swimming pool safety is a critical concern. This research presents a real-time drowning detection approach based on HSV colour space analysis that employs past knowledge of video sequences to choose the appropriate colour channel values. Our approach detects the region of interest in each frame of video sequences by combining a mechanism and contour detection. The provided programme can identify drowning people in indoor swimming pools and sends an alert to lifeguard rescues if the previously recognised individual is gone for an extended period of time. The provided method for this system has been evaluated on various video sequences acquired in real-world swimming pools, and the results are of good accuracy with a high capacity of monitoring persons in real time. According to the assessment results, the system generates few false alarms, and the greatest alert delay provided by the system is 2.6 seconds, which is relatively dependable when compared to the permissible time for rescue and resuscitation.

1.2 PURPOSE

On the basis of an underwater camera, propose a deep learning method for swimmer behaviour detection. Through the real-time gathering of underwater-related picture and data, this technology can properly detect the swimmers' swimming behaviour and drowning behaviour and alert the drowning behaviour. Swimming pools are safe thanks to lifeguards' increased ability to perform rescues.

2. LITERATURE REVIEW

2.1 EXISTING PROBLEM

2.1.1 TITLE: Helping to detect legal swimming pools with Deep Learning and Data Visualization, 2021

AUTHOR NAME: Bruno Lima

One of the major challenges they encounter is dealing with the territory's constant and rapid changes in order to authenticate or licence them. However, much manual intervention is required. This paper describes a prototype system for detecting swimming pools in aerial images. It investigates and integrates artificial intelligence (AI), systems integration, and geographic information systems (GIS), and data visualization. The AI increases objective detection, and middleware facilitates the integration of detection results with other municipal systems for geo-referencing and private property data licensing data crossing. Visualization libraries were investigated in relation to the novel interoperability services, and a sophisticated visualisation and analysis system was built. A dataset of aerial images of swimming pools was created, along with the corresponding classification metadata. To obtain and compare precision results, the model was trained with several convolutional neural networks. The model that is more accurate is described.

2.1.2 TITLE: Construction Method of Swimming Pool Intelligent Assisted Drowning Detection Model Based on Computer Feature Pyramid Networks, 2021

AUTHOR NAME: Keshi Li

Intelligent aided drowning detection in swimming pools is an important research topic in the field of drowning rescue. A significant number of researchers employ an underwater intelligent monitoring system to track drowning targets in real time and use it to develop a dependable swimming pool intelligent aided drowning detection model to lower the danger of drowning. The prior detection methodology was difficult to adapt to the actual need due to the complicated underwater environment of the swimming pool. In this regard, the feature stratification of the swimming pool intelligent assisted drowning detection image is completed based on the summary of the previous swimming pool intelligent assisted drowning detection models and the computer feature pyramid networks, and then the final swimming pool intelligent assisted drowning detection results are obtained using the YOLO principle. Following analysis, it is confirmed that the accuracy rate of this method's swimming pool intelligent assisted drowning detection is significantly improved, which can provide effective data theoretical guidance for swimming pool intelligent assisted drowning rescue and has significant practical benefits.

2.1.3 TITLE: Framework for Intelligent Swimming Analytics with Wearable Sensors for Stroke Classification, 2021
AUTHOR NAME: Joana Costa

An intelligent data analytics system for swimmer performance is suggested in this paper. The system contains I raw signal pre-processing; (ii) wearable sensor and biosensor feature representation; (iii) online detection of swimming style and turns; and (iv) performance post-analysis for coaching decision assistance, including stroke counts and average speed. Wearable inertial (AHRS) and biosensors (heart rate and pulse oximetry) are used to support the system. Radio-frequency connections are used to communicate with the heart rate sensor and the station near the swimming pool where analytics are performed. Experiments were conducted in a real-world training environment with 10 athletes aged 15 to 17 years. This scenario yielded around 8000 samples. Based on real-time data analysis, the experimental findings reveal that the suggested system for intelligent swimming analytics with wearable sensors efficiently provides instant feedback to coaches and athletes. The suggested framework's value was shown by efficiently assisting coaches while monitoring the training of many swimmers.

2.1.4 TITLE: Computer Vision Enabled Drowning Detection System, 2021
AUTHOR NAME: Tereen Prasanga

In all swimming pools, safety is of the utmost importance. Because of technological aspects such as underwater cameras and methodological features such as the requirement for human participation in the rescue operation, the present solutions supposed to handle the challenge of assuring safety at swimming pools have considerable problems. The implementation of an automatic visual-based monitoring system can successfully aid to decrease drowning and ensure pool safety. This study proposes a breakthrough technique that quickly recognises drowning victims and launches an unmanned drone to save them. It can identify a drowning person in three phases using convolutional neural network (CNN) models. When such a scenario is discovered, the inflatable tube-mounted self-driven drone will launch a rescue operation, blasting an alarm to alert surrounding lifeguards. The technology also monitors for potentially harmful acts that might lead to drowning. The capacity of this technology to save a drowning person in less than a minute has been established in prototype experiment performance assessments.

2.1.5 TITILE: A Survey of Deep Learning Techniques for Underwater Image Classification, 2022

AUTHOR NAME: Sparsh Mittal

In recent years, there has been a great deal of interest in utilising deep learning to categorise underwater photos in order to detect diverse items such as fish, plankton, coral reefs, sea grass, submarines, and sea-diver motions. This categorization is critical for monitoring the health and quality of water bodies as well as conserving endangered species. It is also used in oceanography, maritime economics and defence, environmental protection, undersea exploration, and human-robot collaboration work. This study provides an overview of deep learning algorithms for underwater picture categorization. We highlight the similarities and contrasts between various strategies. We believe that underwater image classification is one of the killer applications that will put deep learning techniques to the ultimate test. This survey aims to inform researchers about the state-of-the-art in deep learning on underwater images while also motivating them to push its frontiers forward.

2.2 REFERENCES

[1]. Bruno Lima, Helping to detect legal swimming pools with Deep Learning and Data Visualization, 2021

[2]. Keshi Li, Construction Method of Swimming Pool Intelligent Assisted Drowning Detection Model Based on Computer Feature Pyramid Networks, 2021

[3]. Joana Costa, Framework for Intelligent Swimming Analytics with Wearable Sensors for Stroke Classification, 2021

[4]. Tereen Prasanga, Computer Vision Enabled Drowning Detection System, 2021

[5]. Sparsh Mittal, A Survey of Deep Learning Techniques for Underwater Image Classification, 2022

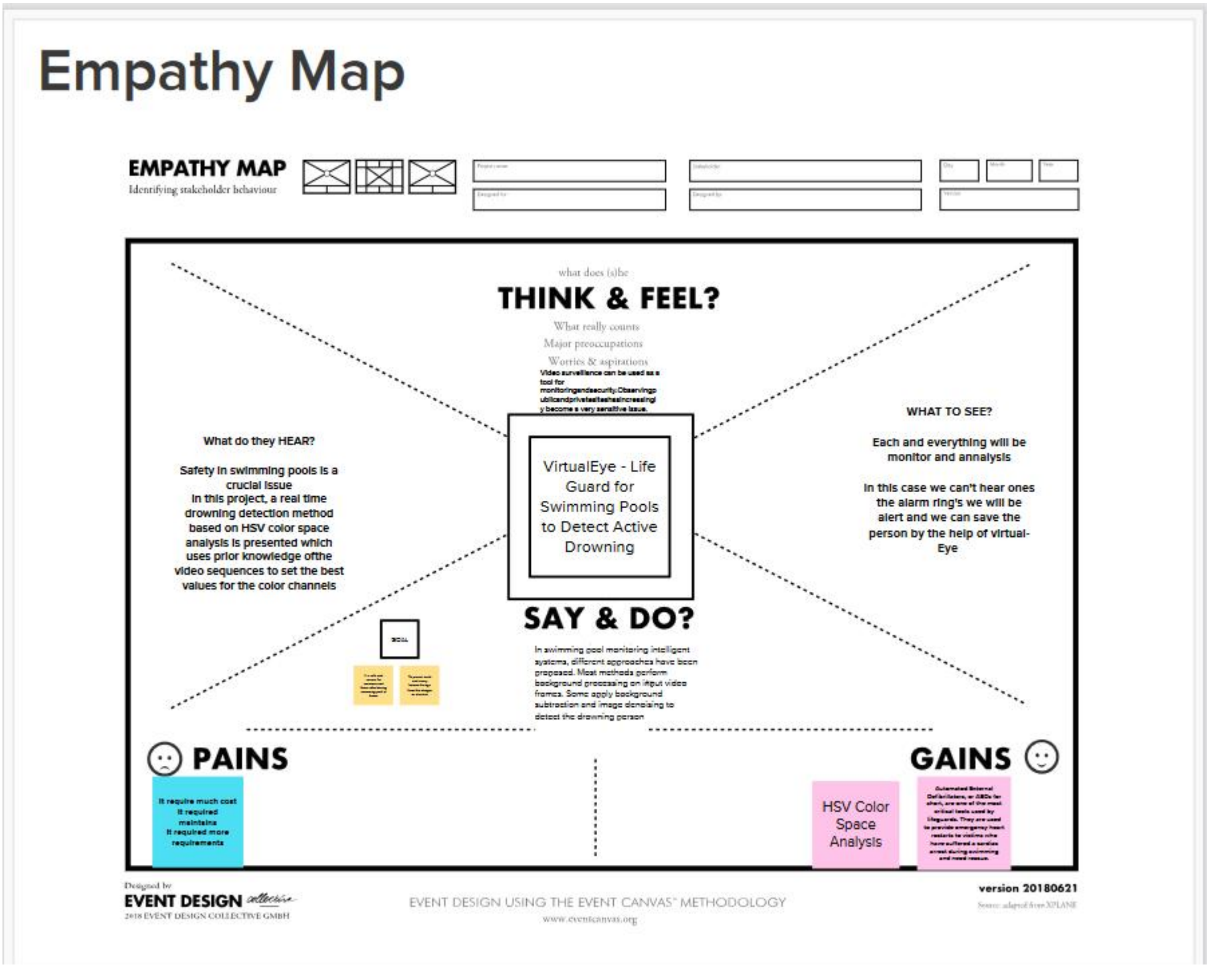
2.3 PROBLEM STATEMENT DEFINITION

In existing system, surveillance systems deliver valued information in monitoring of large areas. Applying intelligence in video surveillance systems allows real-time monitoring of places, people and their activities. The tracking approach can change with varying targets and can change from a single camera to multiple camera configurations. Various factors, including object motion, location, course of movement, and velocity, as well as biometrics like skin colour or clothing colour, are used in tracking techniques in video surveillance. The tracking must be robust and overcome occlusion and noise which are common problems in monitoring.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTORMING



Empathy Map

EMPATHY MAP
Identifying stakeholder behaviour

what does (s)he
THINK & FEEL?
What really counts
Major preoccupations
Worries & aspirations
Video surveillance can be used as a tool for monitoringandsecurity.Observingpublicandprivatefacilitieshasincreasingly become a very sensitive issue.

What do they HEAR?

Safety in swimming pools is a crucial issue
In this project, a real time drowning detection method based on HSV color space analysis is presented which uses prior knowledge ofthe video sequences to set the best values for the color channels

WHAT TO SEE?

Each and everything will be monitor and annalysis

In this case we can't hear ones the alarm ring's we will be alert and we can save the person by the help of virtual-Eye

VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning

SAY & DO?

In swimming pool monitoring intelligent systems, different approaches have been proposed. Most methods perform background processing on input video frames. Some apply background subtraction and image denoising to detect the drowning person

PAINS

It require much cost
It required maintains
It required more requirements

GAINS

HSV Color Space Analysis

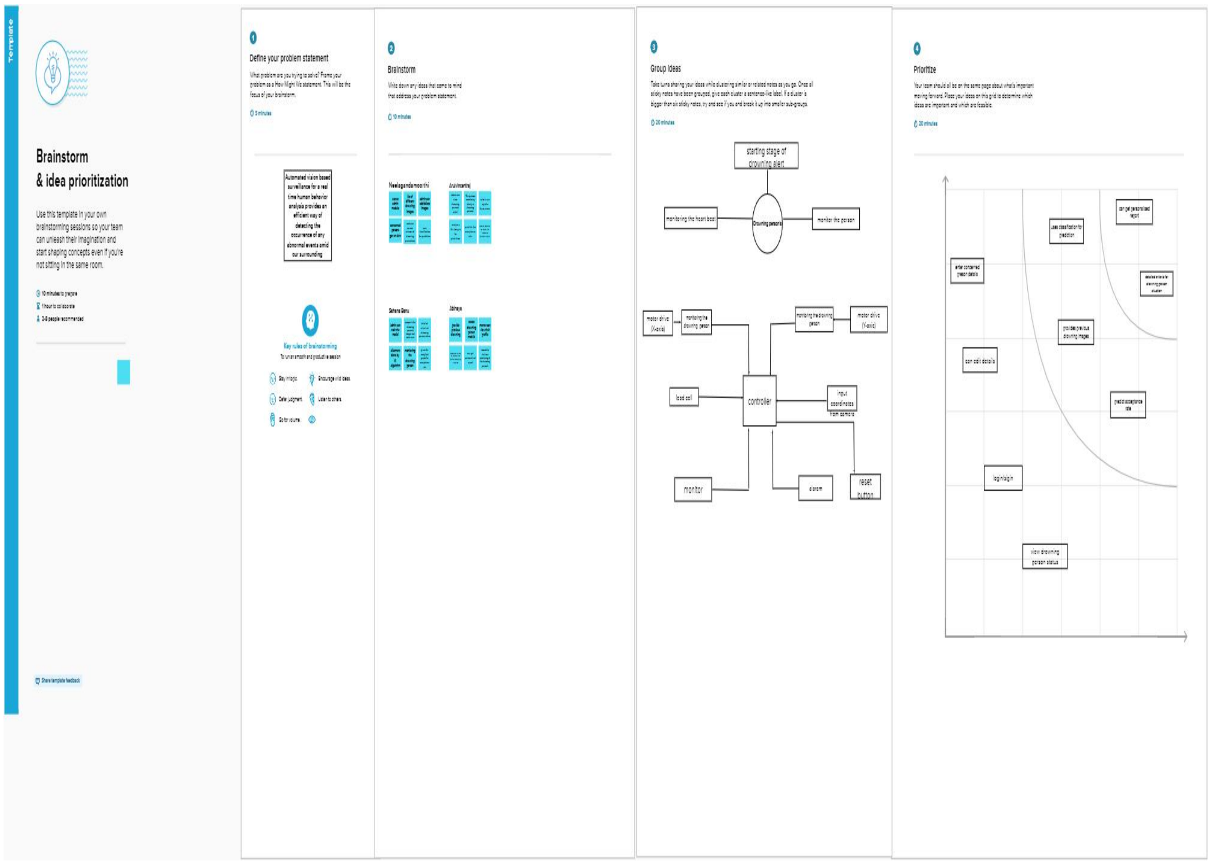
Automated External Defibrillators, or AEDs for short, are one of the most critical tools used by lifeguards. They are used to provide emergency heart restarts to victims who have suffered a cardiac arrest during swimming and need rescue.

Designed by
EVENT DESIGN collective
2018 EVENT DESIGN COLLECTIVE GMBH

EVENT DESIGN USING THE EVENT CANVAS™ METHODOLOGY
www.eventcanvas.org

version 20180621
Source: adapted from XPLANE

3.1 Empathy map


Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

Automated vision based surveillance for a real time human behavior analysis provides an efficient way of detecting the occurrence of any abnormal events amid our surrounding

Key rules of brainstorming
To run an smooth and productive session

Stay in topic. Encourage wild ideas. Defer judgment. Listen to others. Go for volume.

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

4 Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Share template feedback

Neelagandamoorthi
create admin module
list of different drowning images
admin can add/delete images
concerned persons get an alert
Arulvincentraj
Sahana Banu
admin can visit the model
allotment done by ML algorithm
monitoring the drowning person
Abinaya
provide previous drowning
create drowning person module
manage can view their profile
starting stage of drowning alert
monitoring the heart beat
Drowning person's
monitor the person
motor drive (X-axis)
monitoring the drowning person
monitoring the drowning person
motor drive (Y-axis)
load cell
controller
input coordinates from camera
monitor
alaram
reset button
can get personalised report
uses classification for prediction
enter concerned person details
provides previous drowning images
can edit details
predict acceptance rate
login/sign
view drowning person status

3.2 Brainstroming

3.3 PROPOSED SOLUTION

The use of video surveillance as a monitoring and security tool is possible. Observing both public and private locations has grown to be a highly delicate subject. To make people's lives safer, the visual monitoring capabilities may be used in a variety of settings. There are built and deployed video-based security systems in areas like airports, train stations, and even dangerous surroundings. Real-time intelligent monitoring of the objects or events of interest may be accomplished effectively through the use of image processing, pattern recognition, and machine-vision based approaches. Proposed a method for automatic real-time detection of a person drowning in the swimming pools, this raises the need for having a system that will automatically detect the drowning person and alarm the lifeguards of such danger. Real-time detection of a drowning person in swimming pools is a challenging task that requires an accurate system. The challenge is due to the presence of water ripples, shadows and splashes and therefore detection needs to have high accuracy.

3.4 PROBLEM SOLUTION FIT

There isn't a systematic approach to gather the condition of swimmers and not able to monitor all the surrounding for hours. Analyze the position of swimmers in the images, compute and send alert to the responsive team. This system's effectiveness and accuracy will also be increased by expanding the dataset to cover a larger variety.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Framework Creation

The advancement of technologies for avoidance drowsiness at the wheel is a key dilemma in the field of accident prevention systems. Prevent drowsiness in a swimming pool and send alert to the life savers. Drowsy Detection System has been developed, using a non-intrusive machine vision based concepts. This system offers a method for detecting drowsy person in a pool which could be used for observing a level. In this module, we can detect the swimmers movements from real time camera and it will be registered in admin interface.

Object Detection

Object Detection is an advanced form of image classification where a neural network predicts objects in an image and points them out in the form of bounding boxes. Create an underwater pool safety system that lowers the danger of drowning by analysing body movement patterns and integrating cameras with artificial intelligence (AI) technologies. Object detection thus refers to the detection and localization of objects in an image that belong to a predefined set of classes.

Drowsy Detection

Such systems are often created by mounting a camera that transmits underwater footage while analysing swimmer positioning to determine the likelihood that they would drown; if this probability is high, an alarm will be sent to draw lifeguards' attention. To analyse drowning, the YOLO algorithm is used to find the swimmers' location underwater.

Notification System

In this module send notification to the life savers at the time of drowsy prediction. If the position of swimmers is lesser than 50% means, provide notification alert to the lifeguards.

4.2 NON FUNCTIONAL REQUIREMENTS

Usability

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Scalability

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

Performance

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Reliability

The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

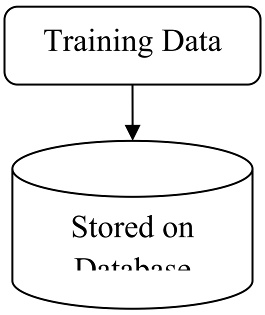
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

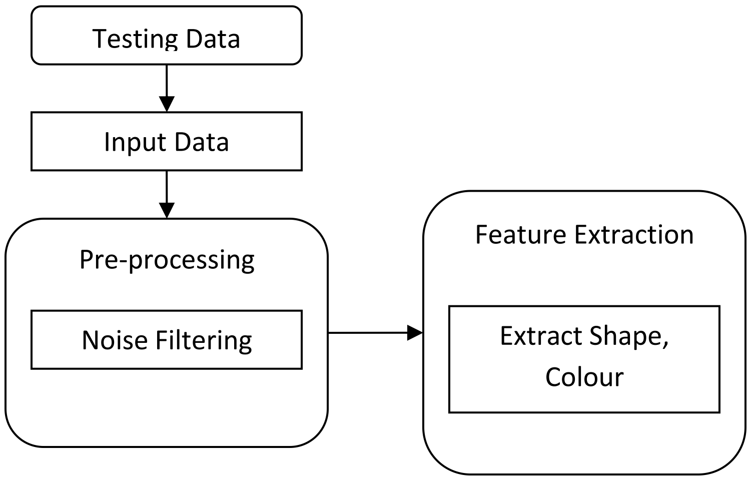
Level 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



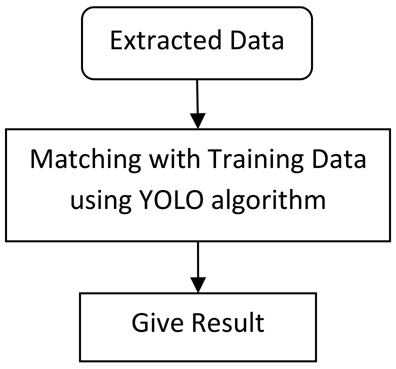
Level 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.



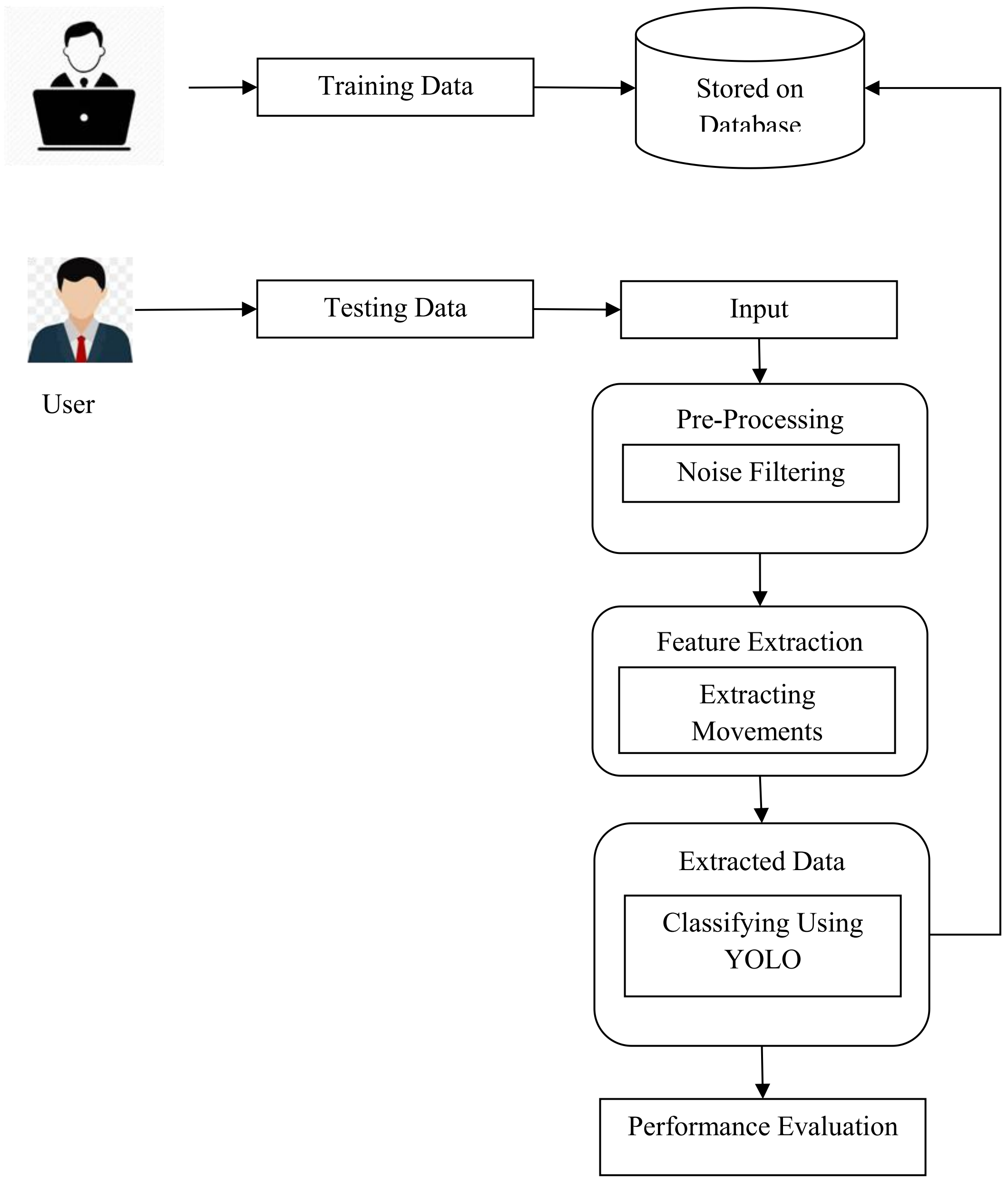
Level 2

A Data Flow Diagram (DFD) tracks processes and their data paths within the business or system boundary under investigation. A DFD defines each domain boundary and illustrates the logical movement and transformation of data within the defined boundary. The diagram shows 'what' input data enters the domain, 'what' logical processes the domain applies to that data, and 'what' output data leaves the domain. Essentially, a DFD is a tool for process modelling and one of the oldest.



5.2 SOLUTION & TECHNICAL ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).


Training Data
Stored on Database
Testing Data
Input
User
Pre-Processing
Noise Filtering
Feature Extraction
Extracting Movements
Extracted Data
Classifying Using YOLO
Performance Evaluation

6. TESTING

6.1 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.
Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

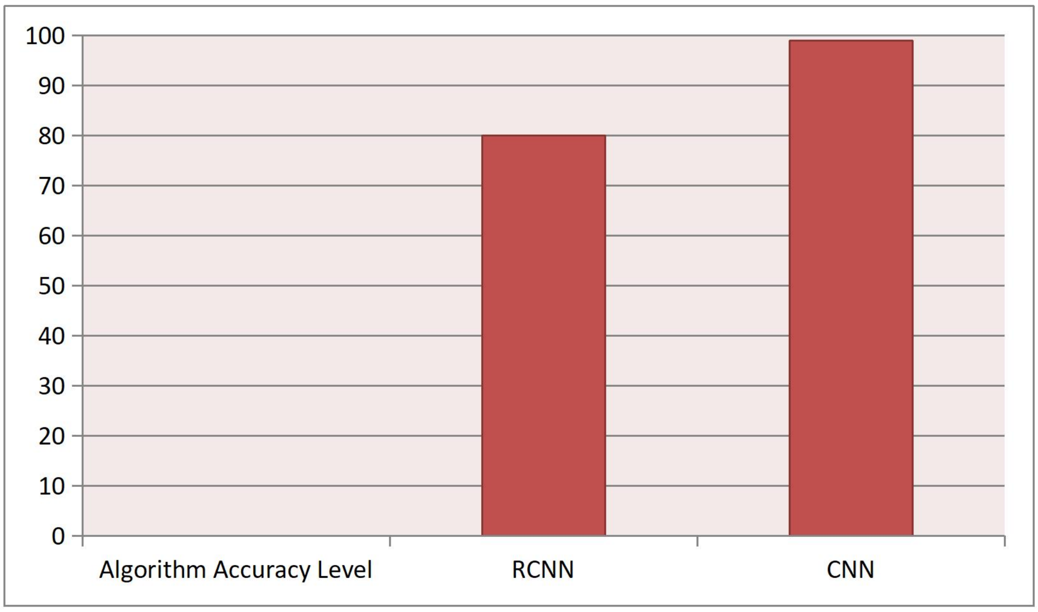
S.NO	**Scenario**	**Input**	**Excepted output**	**Actual output**
1	User login	User name and password	Login	Login success.
2	Upload Image	Upload input image	Detecting swimmers condition	Details are stored in a database.

6.2 USER ACCEPTANCE TESTING

This sort of testing is carried out by users, clients, or other authorised bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programme. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

7. RESULTS

7.1 PERFORMANCE METRICS



8. ADVANTAGES & DISADVANTAGES

ADVANTAGE

- Quickly help life savers
- Detecting the drowsy person accurately
- Provide technical support for reducing drowning accidents

DISADVANTAGE

- Hard for lifeguards to monitor all the surroundings for a long time
- Supervision model is not very reliable
- Consuming more man power

9. CONCLUSION

Provided a method to robust human tracking and semantic event detection within the context of video surveillance system capable of automatically detecting drowning incidents in a swimming pool. In the current work, an effective background detection that incorporates prior knowledge using deep learning algorithm enables swimmers to be reliably detected and tracked despite the significant presence of water ripples. The system has been tested on several instances of simulated water conditions such as water reflection, lightening condition and false alarms. Our algorithm was able to detect all the drowning conditions along with the exact position of the drowning person in the swimming pool and had an average detection delay of 1.53 seconds, which is relatively low compared to the needed rescue time for a lifeguard operation. Our results show that the proposed method can be used as a reliable multimedia video-based surveillance system.

10. FUTURE SCOPE

In the future, a generative adversarial network will be applied to generate synthesis data, in order to increase the size of the training dataset. In addition, more classes will be added to explore and to investigate the efficiency of the proposed system. Thus, the specialized model has outperformed other deep learning algorithms and can achieve impressive results in human drowning incident detection.

11. APPENDIX

SOURCE CODE

```
import Flask, render_template, flash, request,session
from cloudant.client import  Cloudant



client = Cloudant.iam("710709cf-9751-479d-8368-f7286e38749d-
bluemix","h7TBqBQPUFRXQpxRRSchhmLXakubgKUOGL5jEJDTPYrs",connect=
True)
my_database = client.create_database("database-dharan")


app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'



@app.route("/")
def homepage():

    return render_template('index.html')




@app.route("/userhome")
def userhome():

    return render_template('userhome.html')
@app.route("/addamount")
```

```
@app.route("/NewUser")
def NewUser():

    return render_template('NewUser.html')







@app.route("/user")
def user():

    return render_template('user.html')


@app.route("/newuse",methods=['GET','POST'])
def newuse():
    if request.method == 'POST':#

        x = [x for x in request.form.values()]
        print(x)
        data = {
            '_id': x[1],
            'name': x[0],
            'psw': x[2]
        }
        print(data)
        query = {'_id': {'Seq': data['_id']}}
        docs = my_database.get_query_result(query)
        print(docs)
        print(len(docs.all()))
        if (len(docs.all()) == 0):
```

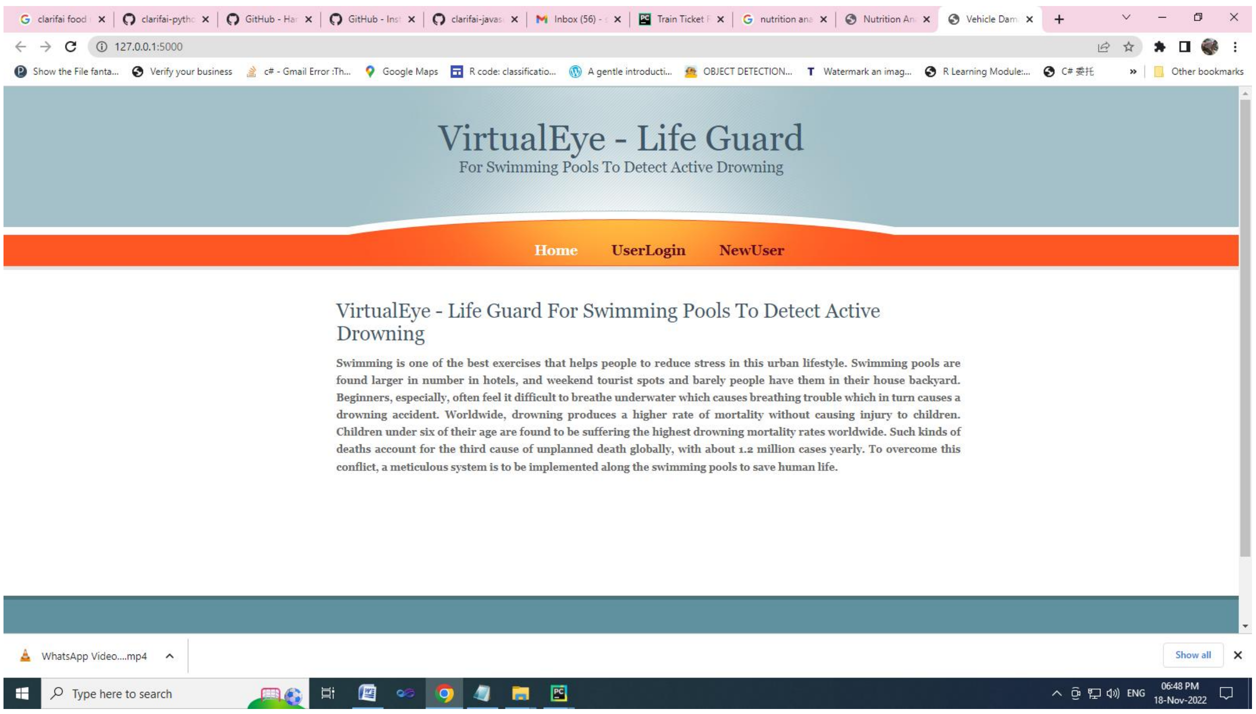
```
        url = my_database.create_document(data)
        return render_template('goback.html', data="Register, please login using your details")
    else:
        return render_template('goback.html', data="You are already a member, please login using your details")

@app.route("/userlog", methods=['GET', 'POST'])
def userlog():
    if request.method == 'POST':

        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('goback.html', pred="The username is not found.")
        else:
            if ((user == docs[0][0]['_id'] and passw == docs[0][0]['psw'])):

                return render_template("userhome.html")
            else:
                return render_template('goback.html',data="user name and password incorrect")
```

```
@app.route("/vvideo", methods=['GET', 'POST'])
def vvideo():
    if request.method == 'POST':

        outttt = " No Drowing"
        ss ="person"



        file = request.files['fileupload']
        file.save('static/Out/Test.avi')

        import warnings
        warnings.filterwarnings('ignore')

        import time
        import cv2
        import os
        import numpy as np

        args = {"confidence": 0.5, "threshold": 0.3}
        flag = False

        labelsPath = "./yolo-coco/coco.names"
        LABELS = open(labelsPath).read().strip().split("\n")
        final_classes = ['person']

        np.random.seed(42)
        COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),
                                   dtype="uint8")

        weightsPath = os.path.abspath("./yolo-coco/yolov3-tiny.weights")
```

```
configPath = os.path.abspath("./yolo-coco/yolov3-tiny.cfg")

# print(configPath, "\n", weightsPath)

net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

vs = cv2.VideoCapture('./static/Out/Test.avi')
writer = None
(W, H) = (None, None)

flag = True

flagg = 0

while True:
    # read the next frame from the file
    (grabbed, frame) = vs.read()

    # if the frame was not grabbed, then we have reached the end
    # of the stream
    if not grabbed:
        break

    # if the frame dimensions are empty, grab them
    if W is None or H is None:
        (H, W) = frame.shape[:2]

    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),
                                 swapRB=True, crop=False)
    net.setInput(blob)
    start = time.time()
    layerOutputs = net.forward(ln)
```

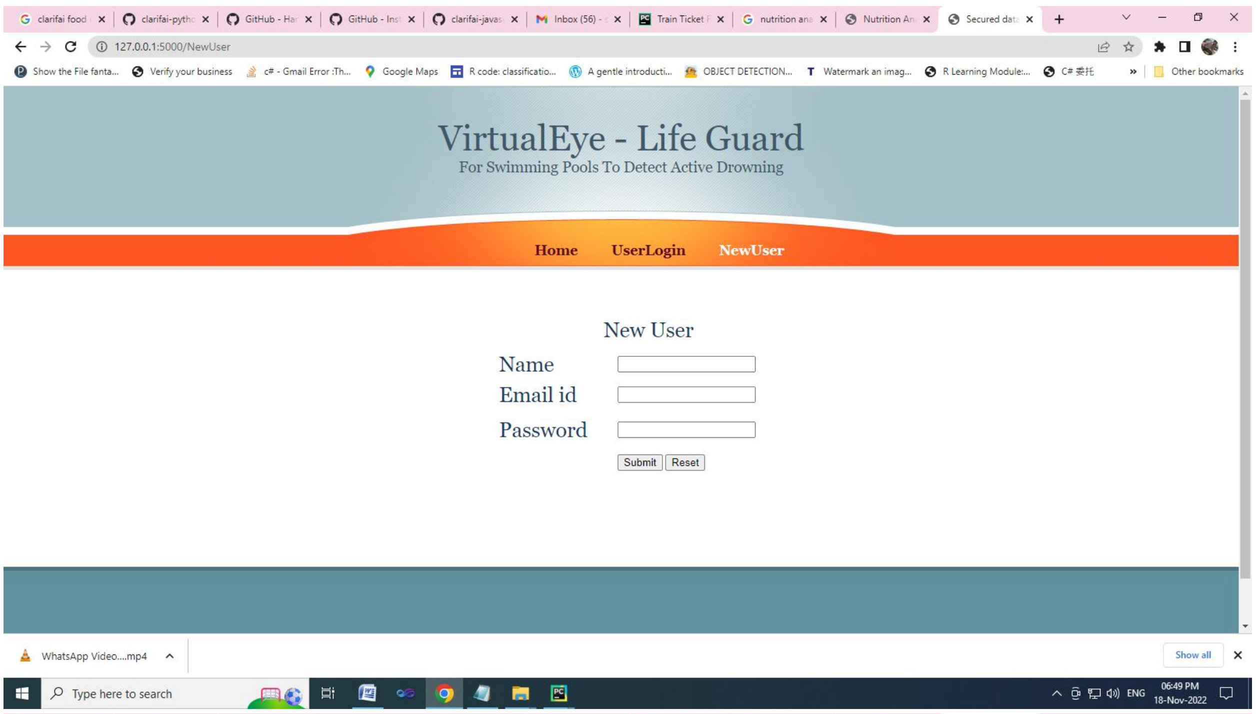
```
end = time.time()

# initialize our lists of detected bounding boxes, confidences,
# and class IDs, respectively
boxes = []
confidences = []
classIDs = []

# loop over each of the layer outputs
for output in layerOutputs:
    # loop over each of the detections
    for detection in output:
        # extract the class ID and confidence (i.e., probability)
        # of the current object detection
        scores = detection[5:]
        classID = np.argmax(scores)
        confidence = scores[classID]

        # filter out weak predictions by ensuring the detected
        # probability is greater than the minimum probability
        if confidence > args["confidence"]:
            # scale the bounding box coordinates back relative to
            # the size of the image, keeping in mind that YOLO
            # actually returns the center (x, y)-coordinates of
            # the bounding box followed by the boxes' width and
            # height
            box = detection[0:4] * np.array([W, H, W, H])
            (centerX, centerY, width, height) = box.astype("int")

            # use the center (x, y)-coordinates to derive the top
            # and and left corner of the bounding box
            x = int(centerX - (width / 2))
            y = int(centerY - (height / 2))
            # update our list of bounding box coordinates,
            # confidences, and class IDs
            boxes.append([x, y, int(width), int(height)])
            confidences.append(float(confidence))
            classIDs.append(classID)

# apply non-maxima suppression to suppress weak, overlapping
# bounding boxes
idxs = cv2.dnn.NMSBoxes(boxes, confidences, args["confidence"],
            args["threshold"])

# ensure at least one detection exists
if len(idxs) > 0:
    # loop over the indexes we are keeping
    for i in idxs.flatten():
        # extract the bounding box coordinates
        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])

        if (ss=="person"):



            flagg += 1
            # print(flag)

            if (flagg == 10):
                outttt = "Drowing"
                flagg = 0
                import winsound

                filename = 'alert.wav'
                winsound.PlaySound(filename, winsound.SND_FILENAME)
                sendmsg("9600713957","Drowing")
```

```
                        cv2.imwrite("alert.jpg", frame)
                    #color = [int(c) for c in COLORS[classIDs[0]]]
                    color ="red"
                    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
                    text = "{}: {:.4f}".format("Person",
                                               confidences[i])
                    cv2.putText(frame, text, (x, y - 5),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)

            else:
                flag = True

            cv2.imshow("Output", frame)

            if cv2.waitKey(1) == ord('q'):
                break

        # release the webcam and destroy all active windows
        vs.release()
        cv2.destroyAllWindows()

        return render_template('userhome.html', prediction=outttt)

def sendmsg(targetno,message):
    import requests

requests.post("http://smsserver9.creativepoint.in/api.php?username=fantasy&passwor
d=596692&to=" + targetno + "&from=FSSMSS&message=Dear user  your msg is " +
message + " Sent By FSMSG
FSSMSS&PEID=1501563800000030506&templateid=1507162882948811640")

if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```



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

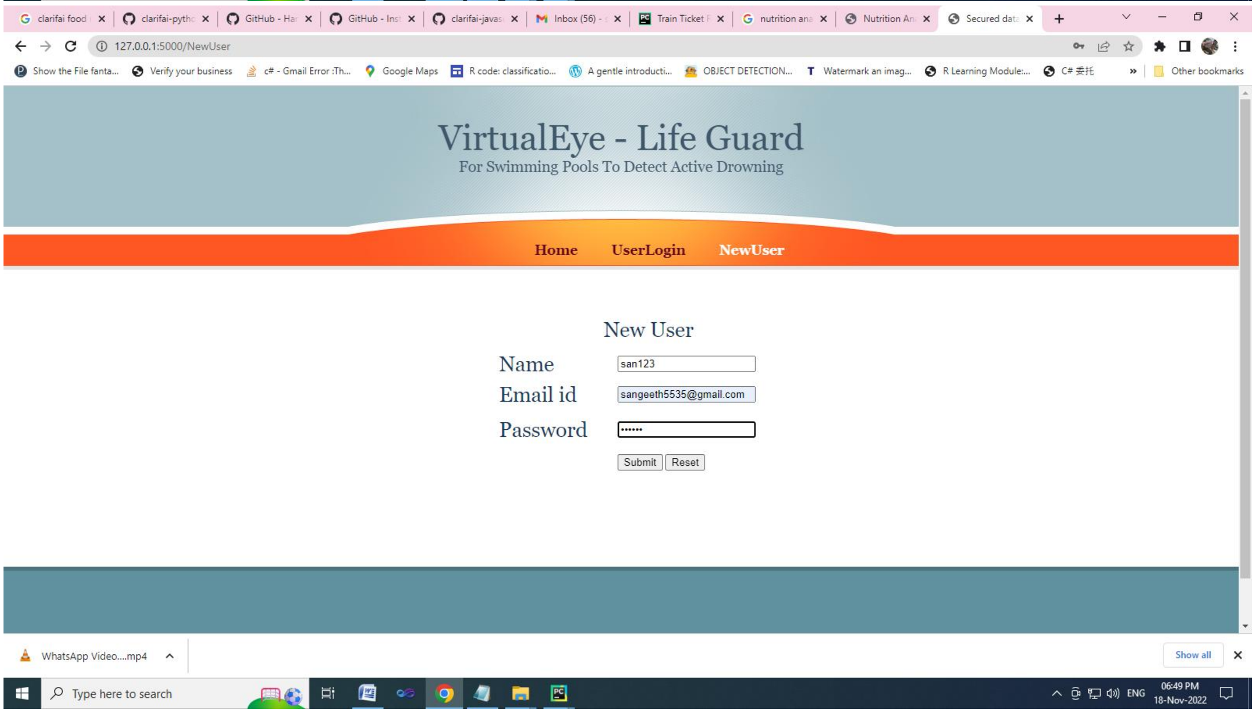


VirtualEye - Life Guard

For Swimming Pools To Detect Active Drowning

New User

Name	
Email id	
Password	

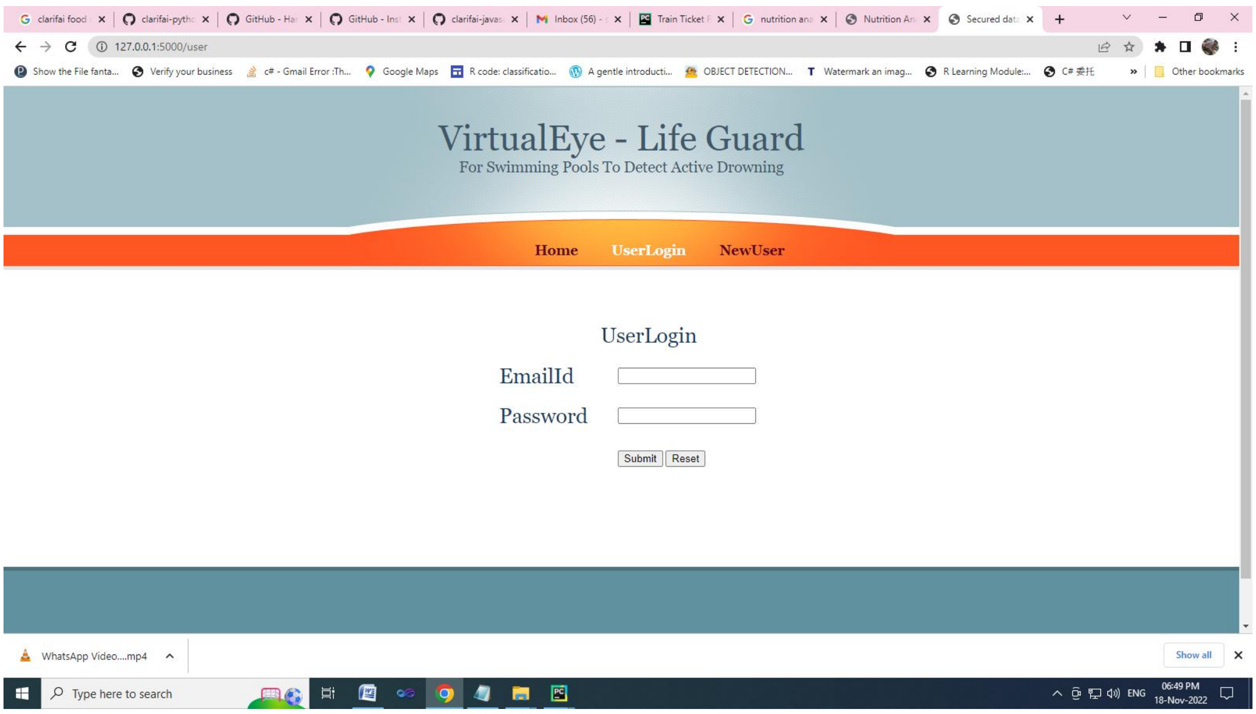


VirtualEye - Life Guard

For Swimming Pools To Detect Active Drowning

New User

Name	san123
Email id	sangeeth5535@gmail.com
Password	••••••



127.0.0.1:5000/user

VirtualEye - Life Guard

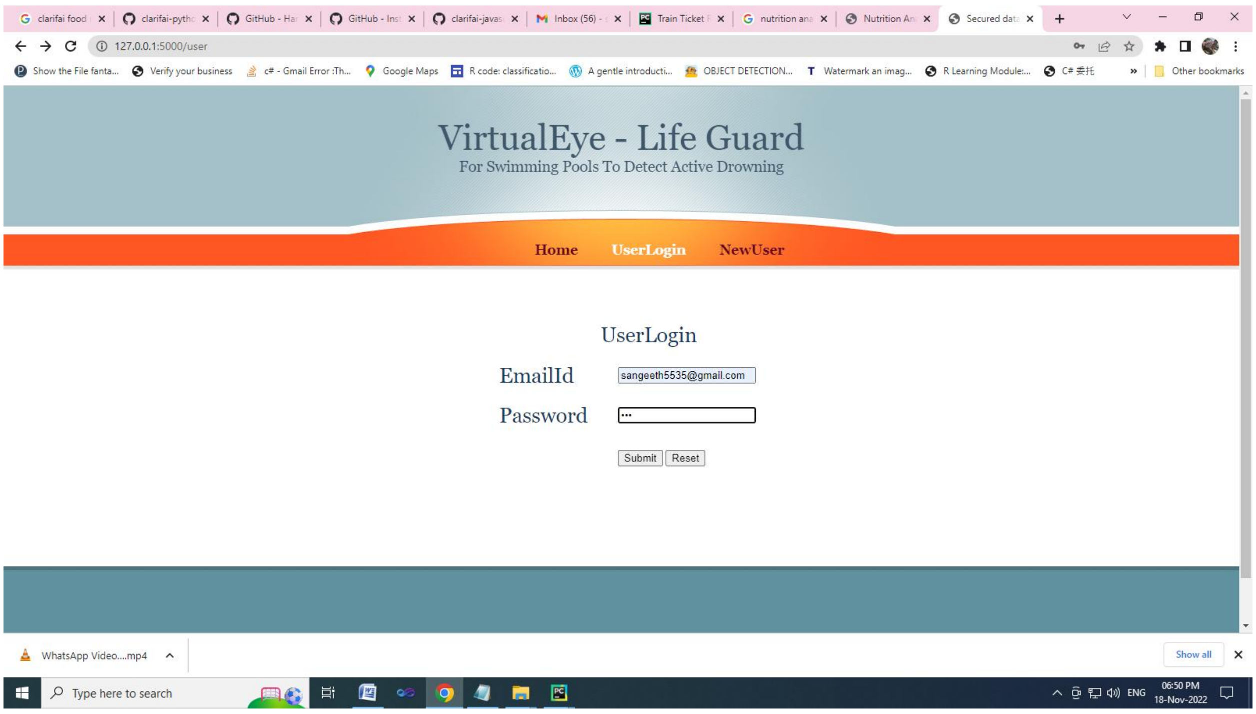
For Swimming Pools To Detect Active Drowning

Home UserLogin NewUser

UserLogin

EmailId	
Password	

Submit Reset



127.0.0.1:5000/user

VirtualEye - Life Guard

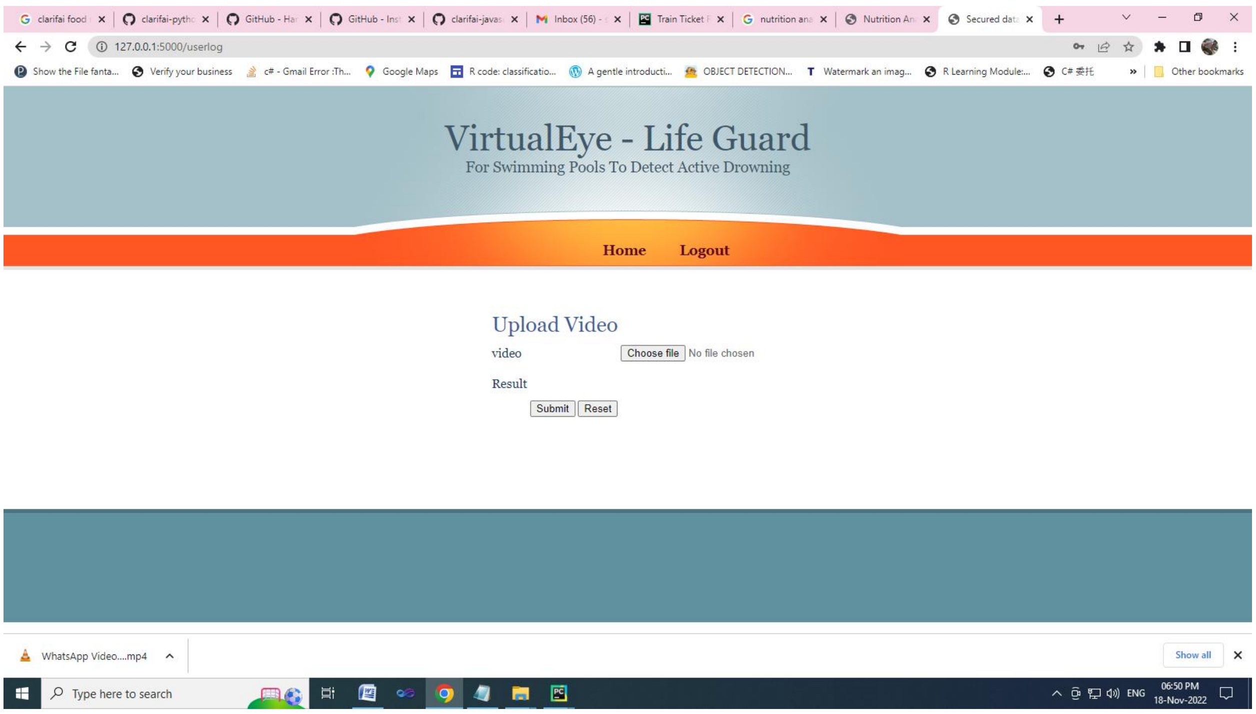
For Swimming Pools To Detect Active Drowning

Home UserLogin NewUser

UserLogin

EmailId	sangeeth5535@gmail.com
Password	•••

Submit Reset

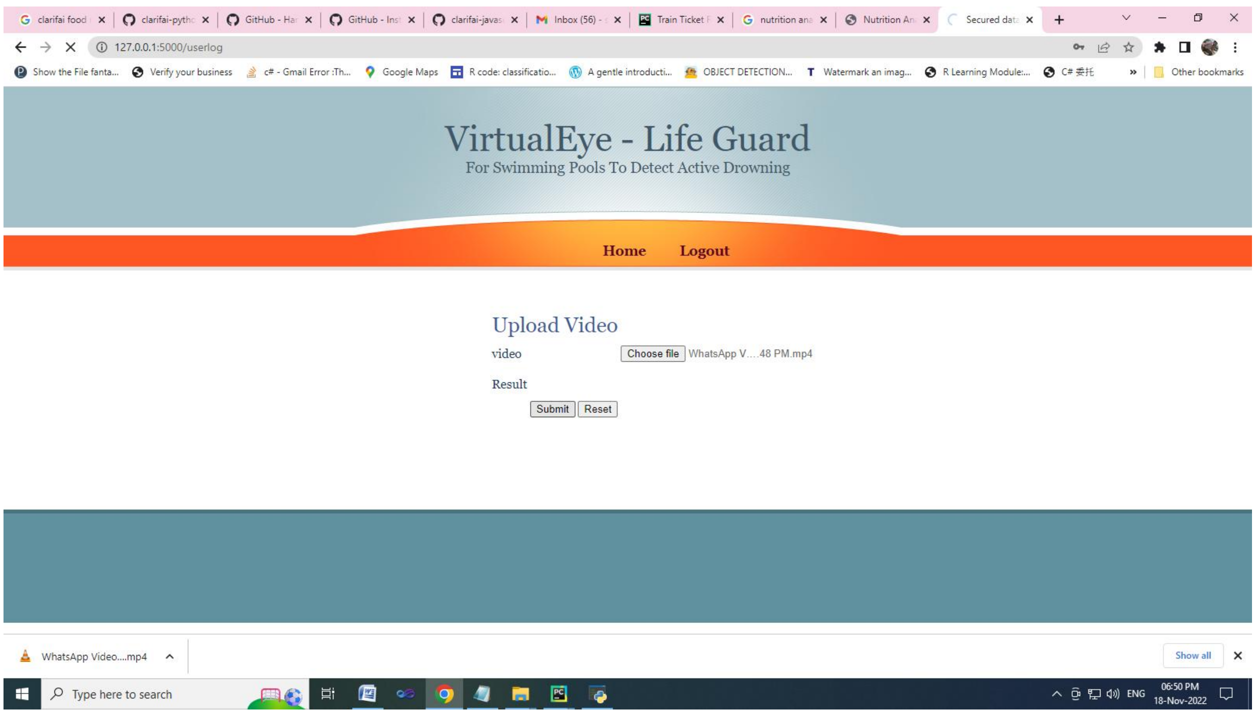

VirtualEye - Life Guard

For Swimming Pools To Detect Active Drowning

Upload Video

video Choose file No file chosen

Result

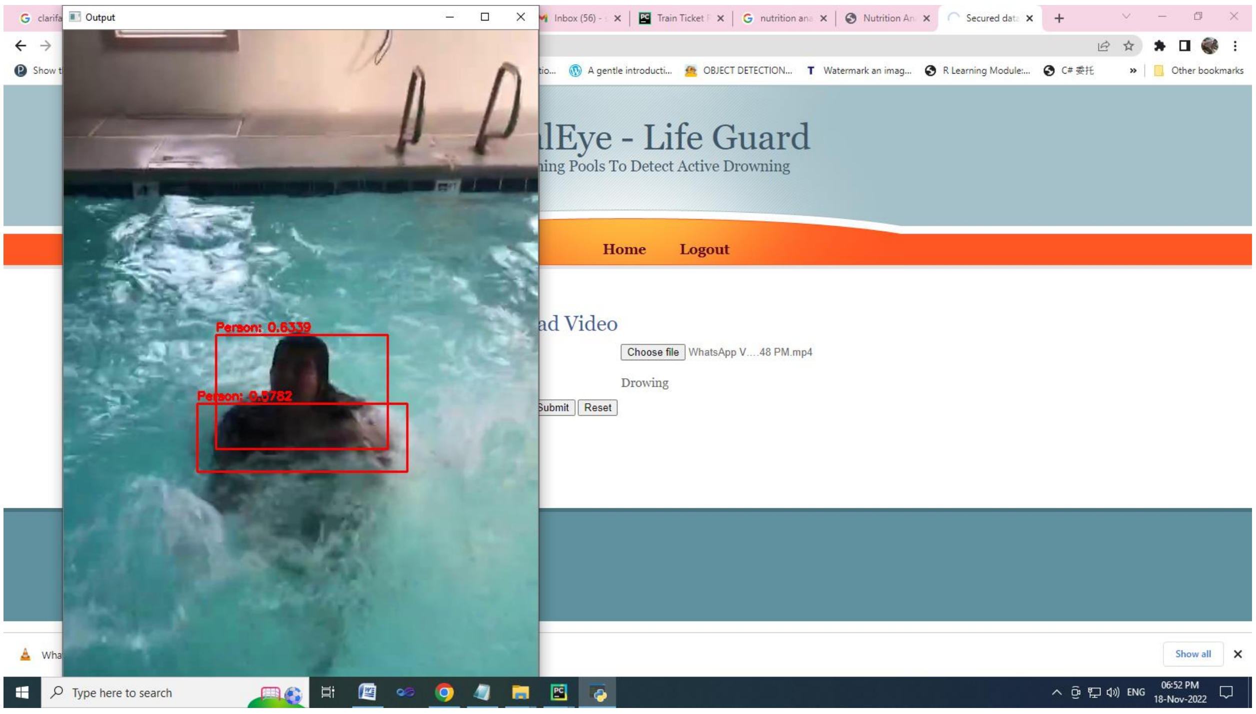

VirtualEye - Life Guard

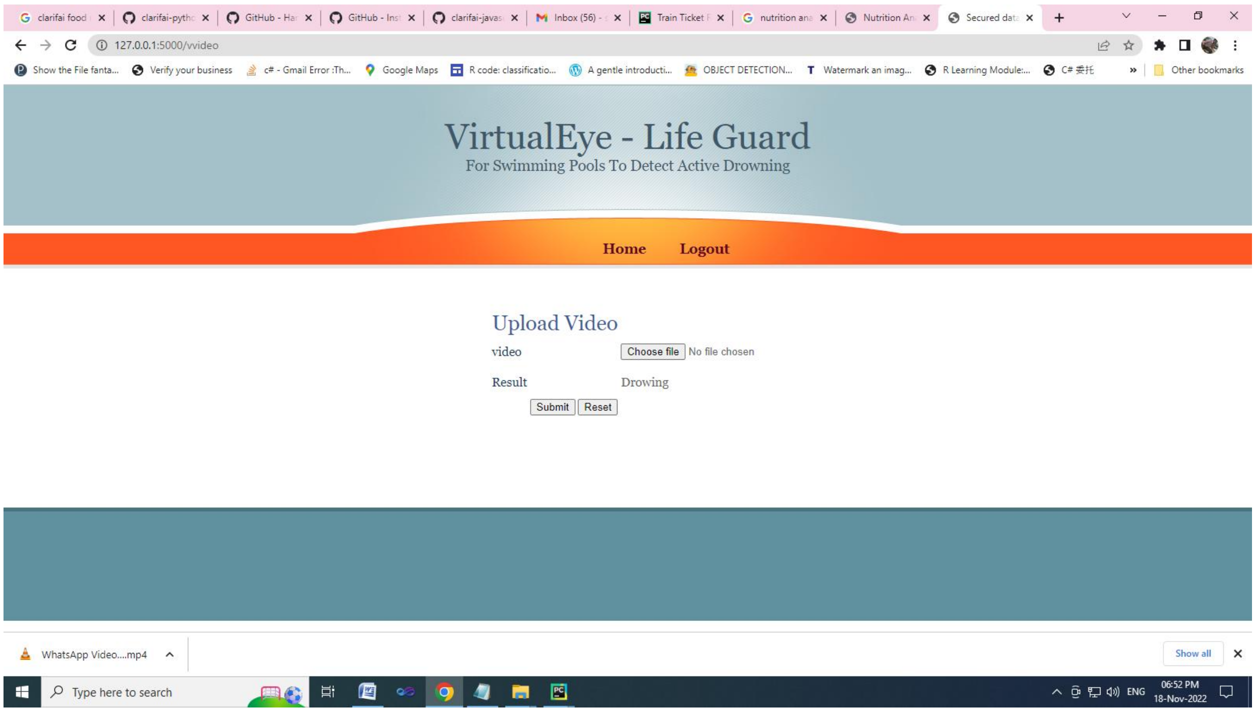
For Swimming Pools To Detect Active Drowning

Upload Video

video Choose file WhatsApp V....48 PM.mp4

Result


Output
Person: 0.6339
Person: 0.5782
lEye - Life Guard
ing Pools To Detect Active Drowning
Home
Logout
ad Video
Choose file
WhatsApp V....48 PM.mp4
Drowing
Submit
Reset
Type here to search
06:52 PM
18-Nov-2022


127.0.0.1:5000/video
VirtualEye - Life Guard
For Swimming Pools To Detect Active Drowning
Home
Logout
Upload Video
video
Choose file
No file chosen
Result
Drowing
Submit
Reset
WhatsApp Video....mp4
Type here to search
06:52 PM
18-Nov-2022

GITHUB & PROJECT DEMO LINK

1. https://github.com/IBM-EPBL/IBM-Project-26063-1659982902

2. https://drive.google.com/file/d/1Qm-n2iqh5-wF3pqnsXGbORuEVH7ct4Eq/view?usp=drivesdk