LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNINGS

A Project Report By

Nandha Kumar S
Dhanush Raj N S
Jayeshwar Karthick V
Musti Ventaka Sai Sravan N
Rajesh S

In fulfillment of project in IBM-NALAYATHIRAN 2022

Team ID : PNT2022TMID20970

PROJECT GUIDES

Industry Mentor : SWATHI

Faculty Mentor : SARAVANAN G

ACKNOWLEDGEMENT

A successful man is one who can lay a firm foundation with the bricks others have thrown at him.—David Brinkley We are indebted to our Head of the Department Dr. G. THAMARAI SELVI for her support during the entire course of this project work. We express our gratitude and sincere thanks to our guide Dr. G. SARAVANAN for her valuable suggestions and constant encouragement for successful completion of this project. Our sincere thanks to our project industrial mentor SWATHI for her kind support in bringing out this project.

ABSTRACT

Safety in swimming pools is a crucial issue. In this paper, a real time drowning detection method based on HSV color space analysis is presented which uses prior knowledge of the video sequences to set the best values for the color channels. Our method uses a HSV thresholding mechanism along with Contour detection to detect the region of interest in each frame of video sequences. The presented software can detect drowning person in indoor swimming pools and sends an alarm to the lifeguard rescues if the previously detected person is missing for a specific amount of time.

The presented algorithm for this system is tested on several video sequences recorded in swimming pools in real conditions and the results are of high accuracy with a high capability of tracking individuals in real time. According to the evaluation results, the number of false alarms generated by the system is minimal and the maximum alarm delay reported by the system is 2.6 sec which can relatively be reliable compared to the acceptable time for rescue and resuscitation.

Project Report Format

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

1.Introduction:

1.1 Project overview:

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.

1.2 Purpose:

By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning.

Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing 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 then an alert will be generated to attract lifeguards' attention.

2. Literature Survey:

2.1 Existing problems:

2.1.1 A Video-Based Drowning Detection System[1]:

This paper provides new insights into robust human tracking and semantic event detection within the context of a novel real-time video surveillance system capable of automatically detecting drowning incidents in a swimming pool. An effective background model that incorporates prior knowledge about swimming pools and aquatic environments enables swimmers to be reliably detected and tracked despite the significant presence of water ripples, splashes and shadows. The technical challenges faced are thus two-fold. First are problems related to object detection and tracking such as dealing with shadows, lighting changes and effects of moving elements of the background. Secondly and no less challenging is the interpretation of the objects' motions into a description of detected actions and interactions. This paper provides fresh insights into these and additional unique problems faced in the development of a novel real-time video surveillance system capable of detecting potential drowning incidents in a swimming pool.

2.1.2 An Automatic Video-based Drowning Detection System for Swimming Pools Using Active Contours[2]

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

2.1.3 A novel drowning detection method for safety of Swimmers[3]:

Effective drowning detection methods are essential for the safety of swimmers. In this paper, a novel type of drowning detection method addressing many limitations of prevailing drowning detectors is proposed. The proposed method ensures detection of drowning and reporting at the earlier stages. The proposed drowning detection method is also a generic solution that suites different water bodies from pools to oceans, and an economically viable method useful for both low and middle income countries. The prototype of the drowning detection method is developed and demonstrated and model of the system is simulated in Proteus design suite. The results of the simulation and hardware experimentation are also reported.

2.1.4 A vision-based approach to early detection of drowning incidents in swimming pools[4]

The proposed approach consists of two main parts: a vision component which can reliably detect and track swimmers in spite of large scene variations of monitored pool areas, and an event-inference module which parses observation sequences of swimmer features for possible drowning behavioral signs. The vision component employs a model-based approach to represent and differentiate the background pool areas and foreground swimmers. The event-inference module is constructed based on a finite state machine, which integrates several reasoning rules formulated from universal motion characteristics of drowning swimmers. Possible

drowning incidents are quickly detected using a sequential change detection algorithm. We have applied the proposed approach to a number of video clips of simulated drowning and obtained promising results as reported in this paper.

2.1.5 Drowning Detection Based on Background Subtraction[5]:

The main research subject in this paper is swimmer detection for visual surveillance of pool. A drowning detection method based on background subtraction is presented in this paper. The consecutive sequence of visual surveillance was obtained by the fixed camera installed in the pool wall. Each pixel is described by Gaussian mixed model, set up self-adapted background model and updating timely. When the foreground objects was separated, for getting good results, the shadows and noises must be removed. The experiments results show that this method is effective to detect the drowners and eliminate the shadows.

2.2 References:

[1] Salehi, Nasrin & Keyvanara, Maryam & Monadjemmi, Seyed. (2016). An Automatic Video-based Drowning Detection System for Swimming Pools Using Active Contours. International Journal of Image, Graphics and Signal Processing. 8. 1-8. 10.5815/ijigsp.2016.08.01.

[2] Wenmiao Lu and Yap-Peng Tan, "A vision-based approach to early detection of drowning incidents in swimming pools," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 2, pp. 159-178, Feb. 2004, doi: 10.1109/TCSVT.2003.821980.

[3] L. Fei, W. Xueli and C. Dongsheng, "Drowning Detection Based on Background Subtraction," 2009 International Conference on Embedded Software and Systems, 2009, pp. 341-343, doi: 10.1109/ICESS.2009.35.

2.3 Problem Statement:

When people swim in the pools, it might not be a mandatory that they should know to swim . Even though they are good at swimming sometimes where they actually loose their control of balance and get themselves drowned. Hence a lifeguard is placed at all times . But it is ot possible for the lifeguard to always be aware of the situations . Hence an AI technology using YOLO model has to built in order to detect the active drowning of the swimmer and give the lifeguard an alarm to alert to the worse situation and help them save their lives.

The model is deployed in the cloud and the end user is the swimming pool owners which will help them improve the safety of the pools and get them advertised with the most safest pool and so on.

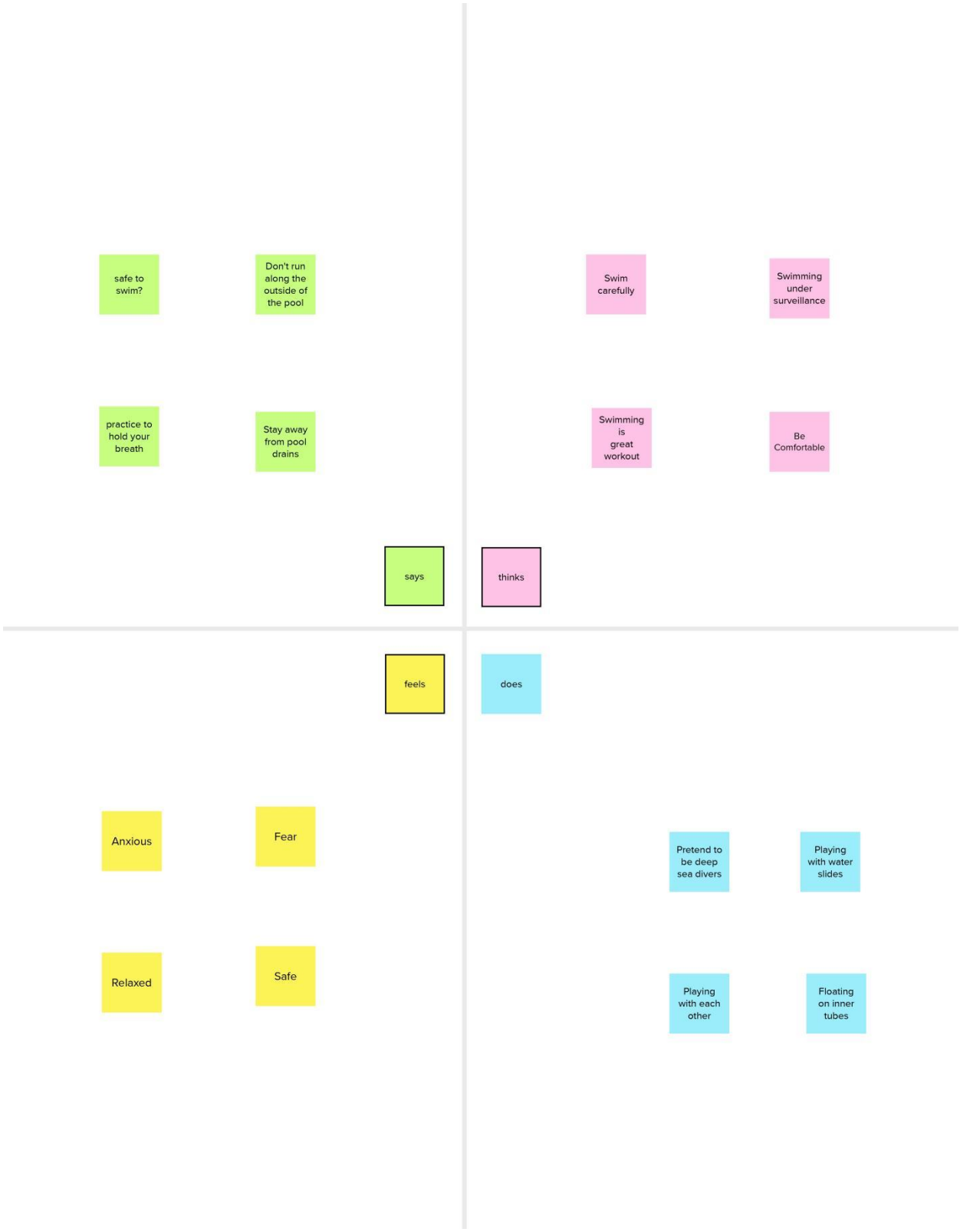
3. Ideation and Problem Solution:

3.1 Empathy Map Canvas:

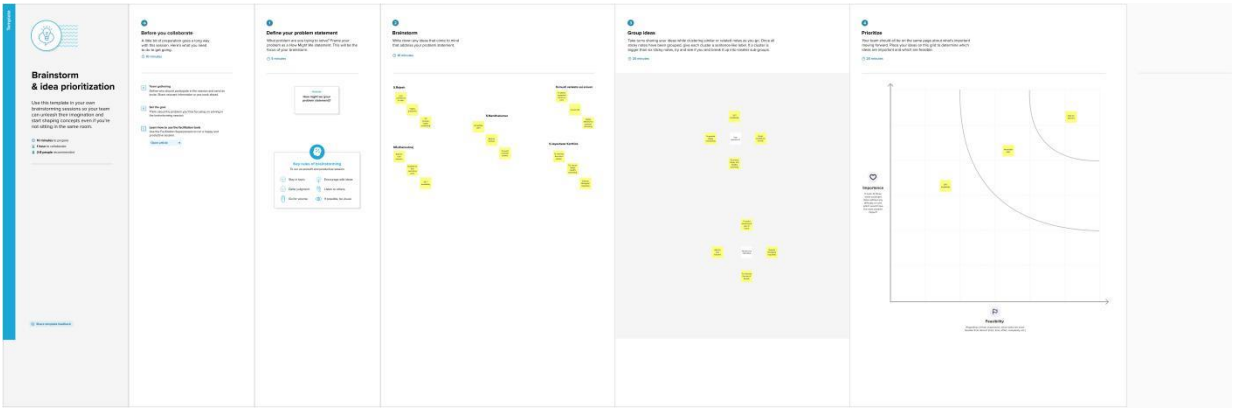
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



3.2 Ideation and Brainstorming:



In this phase, the ideas and the solutions are brainstormed among the team members and the appropriate solution for the problem statement is arrived.

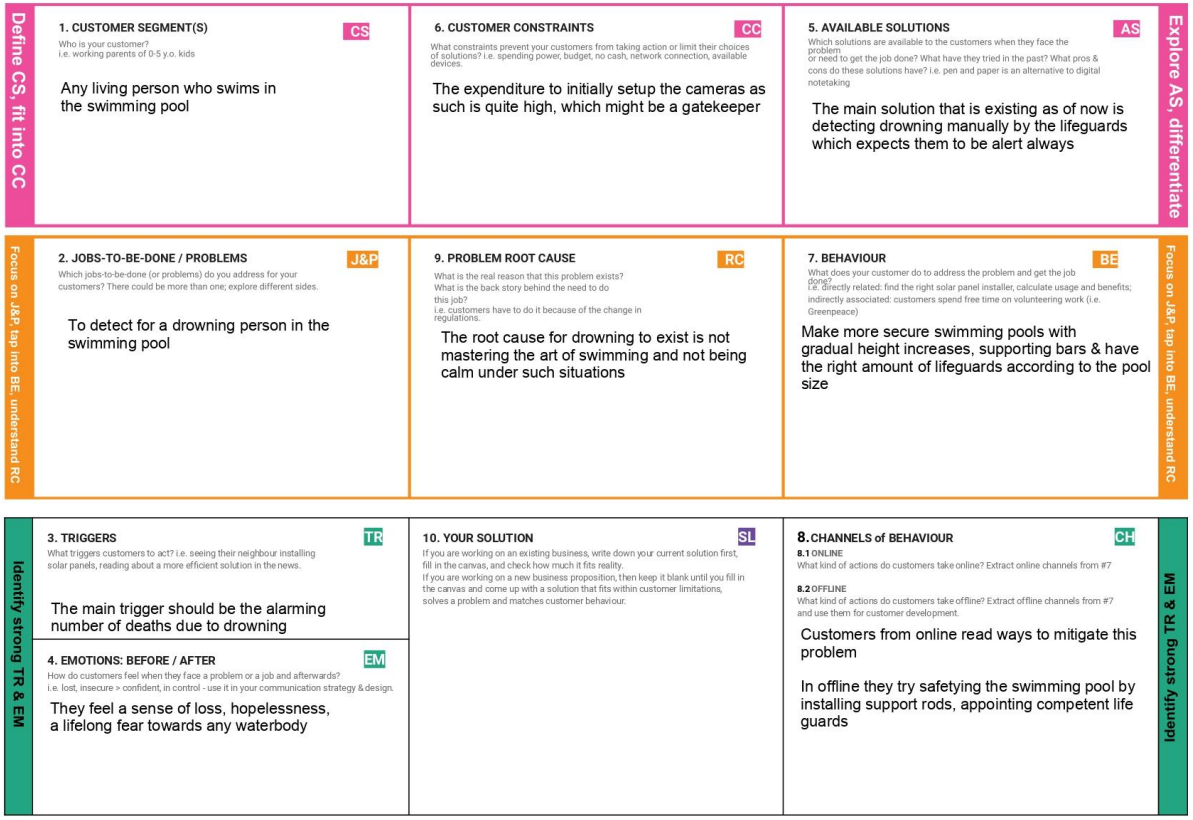
3.3 Proposed Solution:

S. No.	Parameter	Description
1.	Problem Statement (Problemto be solved)	Virtual Eye - Life Guard forSwimming Pools To Detect Active Drowning.

2.	Idea / Solution description	Swimming is one of the quality physical activities that enables humans to reduce strain on this city lifestyle. Swimming swimming pools are located large in wide variety in hotels, and weekend tourist spots, and slightly people have them of their homes and backyards. Beginners, especially, often feel it tough to respire underwater which reasons respiration trouble which in flip reasons a drowning accident In this mission an Accurate Pulse Rate of each man or woman swimmer is likewise detected and sent as a sign to the Life Guard alert message so it enables Life Guard to do in advance prediction of a swimmer pulse charge is decreased or increased By doing this they are able to get alert in advance and may keep extra than one character from Drowning
3.	Novelty / Uniqueness	Accurate pulse rate detection usingDeep learning.

4.	Social Impact / Customer Satisfaction	In the event of an emergency, notonly the video but also the heart rate of the victim can be extractedand saved, which is useful for investigating the cause of drowsiness.
5.	Business Model (Revenue Model)	You can generate revenue fromdirect customers such as lifeguards and work with maritime departments and other pool authorities.
6.	Scalability of the Solution	Heart Rate Detection Deep LearningAlgorithm: Helps lifeguards to predict drowningearly, along with reasons for drowning. Solution Scalability

3.4 Problem solution fit:



4. Requirement Analysis:

4.1 Functional Requirement:

Functional requirements are product features that developers must implement to enable the users to achieve their goals. They define the basic system behavior under specific conditions. **Functional requirements** should not be confused with other types of requirements in product management

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Camera Installation	Cameras should be installed inside water and inthe walls of the building.
FR-2	Sensor Installation	Installed under water without disturbing the people.
FR-3	Deduction	Detected by movements of the swimmer.
FR-4	Alert	Sends an alert message to the lifeguard.
FR-5	Support	Lifeguard help or swim tubes.
FR-6	Alarm	Rings alarm with drowning detected.

In this project , the functional requirements are the required functions that our application has to perform . The application has

User registration- There may be different users who use this application at different locations and they have to be distinguished

User confirmation- Once the user details has been registered , the user is confirmed by login.

Camera Installations- The main source for the detection is the image frames obtained from cameras, which captures live action

Alarms- Once the drowning is detected , the lifeguards are alerted in no delay to save the drowning people

4.2 Non-Functional Requirements:

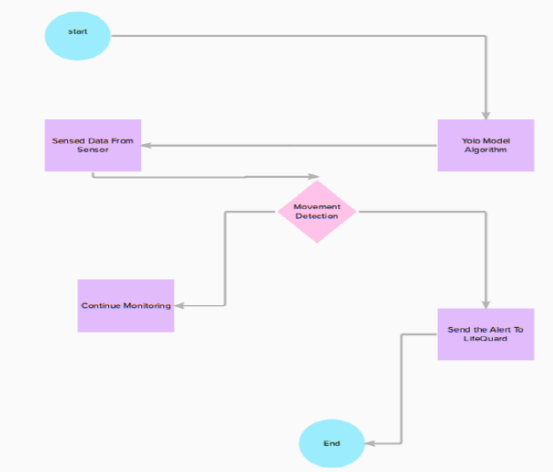
In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviour.

FR No.	Non-Functional Requirement	Description
NFR -1	**Usability**	When someone is drowning, the sensors detect the movements and locate the swimmer who is drowning and alert the people.
NFR -2	**Security**	Lifeguards will be present in the pool and the Cameras are secured by the management and are safe.
NFR -3	**Reliability**	The process will be a reliable multimedia videoBased surveillance system.
NFR -4	**Performance**	When the movements of the swimmer areunusual then the alarm will be triggered.

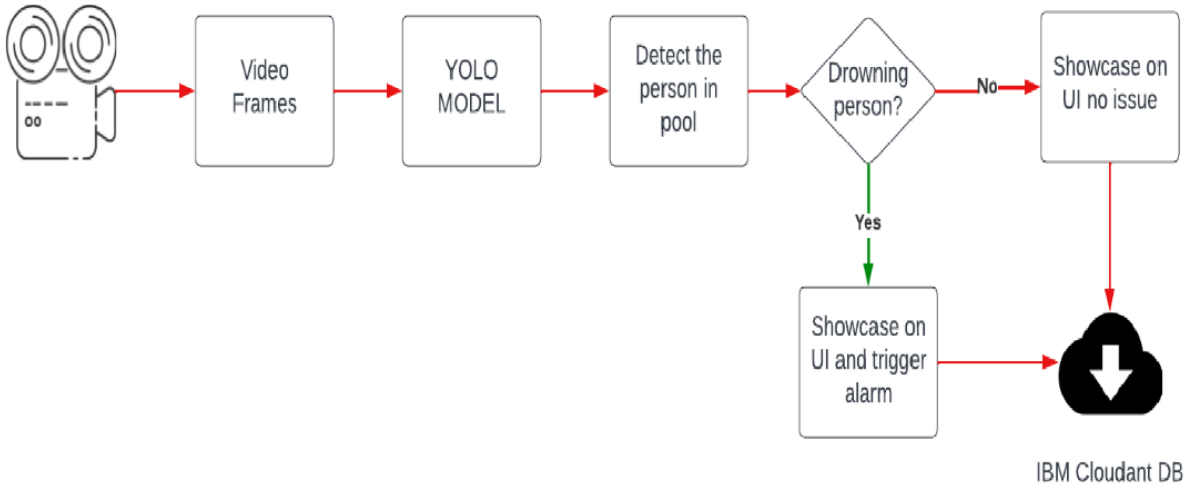
NFR -5	**Availability**	Detection equipment includes safety wheel, poolhook, rescue tubes, first aid box etc.
NFR -6	**Scalability**	Deep learning algorithm for the drowning detection helps the lifeguard for earlierprediction with the reaon behind their drowning.

5. Project Design:

5.1 Dataflow Diagrams:



5.2 Solution and Technical architecture:



6.Project Planning and Scheduling:

6.1 Sprint planning and estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	VIR-11	As a user, I can register for the application byentering my email, password, and confirming my password.	2	High	Nandhakumar S
Sprint-1	Registration	VIR-11	As a user, I will receive confirmation email onceI have registered for the application	2	High	Rajesh S
Sprint-1	Registration	VIR-11	As a user, I can register for the applicationthrough Facebook	1	Low	Dhanush Raj N S
Sprint-1	Registration	VIR-11	As a user, I can register for the applicationthrough Gmail	2	Medium	Jayeshwar KarthickV
Sprint-1	Login	VIR-12	As a user, I can log into the application byentering email & password	3	High	Sai Sravan M V
Sprint-2	Dataset Collect	VIR-13	Collect number of datasets and get accuracy	2	Medium	Rajesh S
Sprint-2	Pre-processing	VIR-14	The dataset is extracted	3	High	Jayeshwar KarthickV
Sprint-2	Train the model	VIR-15	Train the model.	4	High	Nandhakumar S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	S Point
Sprint-2	Test the model	VIR-16	Test the model	
Sprint-3	Detection	VIR-17	Load the trained model.	
Sprint-3	Detection	VIR-18	Identify the person by collecting real-time datathrough a webcam.	
Sprint-3	Detection	VIR-19	classify it by using a trained model to predictthe output	
Sprint-4	Detection	VIR-18	If person is drowning, the system will ring analarm to give signal	
Sprint-4	Detection	VIR-19	As a User,I can detect the drowning person.	
Sprint-4	Logout	VIR-20	As a User,I can logout the application.	

6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story P Completed Planne Date)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	6
Sprint-2	14	6 Days	31 Oct 2022	05 Nov 2022	12

Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	11
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	12

7. Coding and Solutioning:

7.1 Feature 1:

The application enables user to login and register to access the functionalities . This is achieved through firebase oauth system.

HTML Codes

Index.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 rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
PNT2022TMID38060
<link
href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=JosefinSans' 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:10px;

font-family: 'Josefin Sans';

font-size: 2vw;

width: 100%;

height:8%;

text-align: left;

}

.topnav {

overflow: hidden;

background-color: #333;

}

.topnav-right a {

float: left;

color: #f2f2f2;

text-align: center;

padding: 14px 14px;

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:50px;

}

.login{

margin-top:-70px;

}

body {

background-color:#ffffff;

background-repeat: no-repeat;
```

```
background-size:cover;

background-position: 0px 0px;

}

.login{

margin-top:50px;

}

form {border: 0px solid #f1f1f1; margin-left:400px;margin- right:200px;}

input[type=text],

input[type=email],input[type=number],input[type=password] {

width: 50%;

padding: 12px 12px;

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: 50%;

font-weight:bold;

}

button:hover {

opacity: 0.8;

}

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}

.imgcontainer {

text-align: left;

margin: 24px 0 12px 0;

}
```

```
img.avatar {

width: 30%;

border-radius: 20%;

}

.container {

padding: 14px;

}

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: 50%;
```

```
}

}

</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('index')}}">Home</a>

<a class="active" href="{{ url_for('login')}}">Login</a>

<a href="{{ url_for('register')}}">Register</a>

</div>

</div>

<div id="login" class="login">

<form action="{{url_for('afterlogin')}}" method="post"><div class="imgcontainer">

<img style=""

src="https://i0.wp.com/upload.wikimedia.org/wikipedia/commons/thumb/4/4c/Stylized_eye.svg/1920px-S

tylized_eye.svg.png?resize=565%2C226&ssl=1" alt="Avatar" class="avatar">
```

```
</div>

<div class="container">

<input type="email" placeholder="Enter registered email ID" name="_id" required><br>

<input type="password" placeholder="Enter Password" name="psw" required>

<button type="submit">Login</button><br>

</div>

</form>

</div>

</body>

</html>
```

Register.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 rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

PNT2022TMID38060

<link

href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=JosefinSans' 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:10px;

font-family: 'Josefin Sans';

font-size: 2vw;

width: 100%;

height:8%;

text-align: left;

}

.topnav {

overflow: hidden;

background-color: #333;

}

.topnav-right a {

float: left;

color: #f2f2f2;

text-align: center;

padding: 14px 14px;

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:50px;

}

.login{

margin-top:-70px;

}

body {

background-color:#ffffff;
```

```
background-repeat: no-repeat;

background-size:cover;

background-position: 0px 0px;

}

.login{

margin-top:50px;

}

form {border: 0px solid #f1f1f1; margin-left:400px;margin- right:200px;}

input[type=text],

input[type=email],input[type=number],input[type=password] {

width: 50%;

padding: 12px 12px;

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: 50%;

font-weight:bold;

}

button:hover {

opacity: 0.8;

}

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}

.imgcontainer {

text-align: left;

margin: 24px 0 12px 0;
```

```
}

img.avatar {

width: 30%;

border-radius: 20%;

}

.container {

padding: 14px;

}

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: 50%;
}
}
</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('index')}}">Home</a>
<a class="active" href="{{ url_for('login')}}">Login</a>
<a href="{{ url_for('register')}}">Register</a>
</div>
</div>
<div id="login" class="login">
<form action="{{url_for('afterlogin')}}" method="post"><div class="imgcontainer">
<img style=""
src="https://i0.wp.com/upload.wikimedia.org/wikipedia/commons/thumb/4/4c/Stylized_eye.svg/1920px-S
```

tylized_eye.svg.png?resize=565%2C226&ssl=1" alt="Avatar" class="avatar">

<input type="email" placeholder="Enter registered email ID" name="_id" required>

<input type="password" placeholder="Enter Password" name="psw" required>

<button type="submit">Login</button>

</form>

</body>

</html>

Base.htm

<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.j s"></script>

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

>

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

<link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">

<style>

.bg-dark {

background-color: #42678c!important;

}

#result {

color: #0a1c4ed1;

}

</style>

</head>

<body style="background-color:black";>

<header id="head" class="header">

<section id="navbar">

<h1 class="nav-heading"></i>Virtual Eye</h1>

<div class="nav--items">
```

```
<ul>
<li><a href="{{ url_for('index')}}">Home</a></li>
<li><a
href="{{ url_for('logout')}}">Logout</a></li>
<!-- <li><a href="#about">About</a></li>
<li><a href="#services">Services</a></li> -->
</ul>
</div>
</section>
</header>
<div class="container">
<div id="content" style="margin-top:2em">
<div class="container">
<div class="row">
<div class="col-sm-6 bd" >
<h2><em style="color:white;">High Quality Facial Recognition</em></h2>
<br>
<p><h5><i style="color:white;">Emotion Detection Through Facial Feature
Recognition</i></h5></p>
```

<img
src="https://130e178e8f8ba617604b8aedd782b7d22cfe0d1146da69a52436.ssl.cf1.rackcdn.com/facialrecognition-use-triggers-gdpr-fine-showcase_image-10-a- 12991.jpg"
style="height:240px"class="img-rounded" alt="Gesture">

<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="uploadlabel">
Choose Image
</label>
<input type="file" name="image"
id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">
</form>

```
</div>

</div>

<div>

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

</div>

</div>

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

</body>

</div>

</div>

</div>

<footer>

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

</footer>
```

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+Conde nsed: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-

KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpG FF93hXpG5KkN"

crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/ popper.min.js"

integrity="sha384- ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk

vXusvfa0b4Q" crossorigin="anonymous"></script>
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootst rap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5
+76PVCmYl" crossorigin="anonymous"></script>
<script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>
<link href="https://fonts.googleapis.com/css2?family=Akronim&family= Roboto&display=swap"
rel="stylesheet">
<link rel="stylesheet" href="../static/style.css">
<script defer src="../static/js/JScript.js"></script>
<title>Prediction</title>
</head>
<body>
<header id="head" class="header">
<section id="navbar">
<h1 class="nav-heading"></i>Virtual Eye</h1>
<div class="nav--items">
<ul>
<li><a href="{{ url_for('index')}}">Home</a></li>
<li><a
href="{{ url_for('logout')}}">Logout</a></li>
```

<!-- About

Services -->

</section>

</header>

<!-- dataset/Training/metal/metal326.jpg -->

</br>

<section id="prediction">

<h2 class="title text-muted">Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning</h1>

</section>

</br>

<section id="about">

<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>

<img class="d-block w-100" src="../static/img/second.jpg" alt="Second slide">

</br>

<form id="form" action="/result" method="post" enctype="multipart/form-data">

<input type="submit" class="submitbtn" value="Click Me! For a Demo">

</form>

<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>
```

Main.js

```
$(document).ready(function () {

    // Init

    $('.image-section').hide();

    $('.loader').hide();

    $('#result').hide();



    // Upload Preview
```

```
        function readURL(input) {
            if (input.files && input.files[0]) {
                var reader = new FileReader();
                reader.onload = function (e) {
                    $('#imagePreview').css('background-image', 'url(' +
e.target.result + ')');
                    $('#imagePreview').hide();
                    $('#imagePreview').fadeIn(650);
                }
                reader.readAsDataURL(input.files[0]);
            }
        }
        $("#imageUpload").change(function () {
            $('.image-section').show();
            $('#btn-predict').show();
            $('#result').text('');
```

```
    $('#result').hide();
    readURL(this);
});


// Predict
$('#btn-predict').click(function () {
    var form_data = new FormData($('#upload-file')[0]);


    // Show loading animation
    $(this).hide();
    $('.loader').show();


    // Make prediction by calling api /predict
    $.ajax({
        type: 'POST',
        url: '/predict',
```

Python code :

```
# -*- coding: utf-8 -*-

"""pythoncode.ipynb



Automatically generated by Colaboratory.



Original file is located at

    https://colab.research.google.com/drive/1gqbVQbV1MiAgD2AZjlTe9fOvE2FpUjRL

"""



import requests

import progressbar as pb

import os

import cv2

import numpy as np

import time
```

```
def download_file(url, file_name, dest_dir):


    if not os.path.exists(dest_dir):

        os.makedirs(dest_dir)


    full_path_to_file = dest_dir + os.path.sep + file_name


    if os.path.exists(dest_dir + os.path.sep + file_name):

        return full_path_to_file


    print("Downloading " + file_name + " from " + url)


    try:

        r = requests.get(url, allow_redirects=True, stream=True)

    except:

        print("Could not establish connection. Download failed")

        return None
```

```
file_size = int(r.headers['Content-Length'])

chunk_size = 1024

num_bars = round(file_size / chunk_size)



bar = pb.ProgressBar(maxval=num_bars).start()



if r.status_code != requests.codes.ok:

   print("Error occurred while downloading file")

   return None



count = 0



with open(full_path_to_file, 'wb') as file:

   for chunk in  r.iter_content(chunk_size=chunk_size):

      file.write(chunk)

      bar.update(count)
```

```
            count +=1


        return full_path_to_file


    initialize = True

    net = None

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

    classes = None

    #colors are BGR instead of RGB in python

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


    def populate_class_labels():


        #we are using a pre existent classifier which is more reliable and more efficient than one

        #we could make using only a laptop

        #The classifier should be downloaded automatically when you run this script

        class_file_name = 'yolov3_classes.txt'
```

```
    class_file_abs_path = dest_dir + os.path.sep + class_file_name

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

    if not os.path.exists(class_file_abs_path):

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

    f = open(class_file_abs_path, 'r')

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



    return classes



def get_output_layers(net):



    #the number of output layers in a neural network is the number of possible

    #things the network can detect, such as a person, a dog, a tie, a phone...

    layer_names = net.getLayerNames()



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

```
    return output_layers


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


    global COLORS

    global classes


    if classes is None:

        classes = populate_class_labels()


    for i, label in enumerate(labels):


        #if the person is drowning, the box will be drawn red instead of blue

        if 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

#for PiCamera
#from picamera Import PiCamera
#camera = PiCamera
#camera.start_preview()
```

```
# open webcam

webcam = cv2.VideoCapture(0)



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)



# press "Q" to stop

if cv2.waitKey(1) & 0xFF == ord('q'):

    break
```

```
# release resources

webcam.release()

cv2.destroyAllWindows()
```

```
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text('Prediction: '+data);
                console.log('Success!');
            },
        });
    });


});
```

8. Test Cases:

Drowning detected:

User is safe:

9. Results:

Drowning Detection and Tracking Results The YOLO detection algorithm uses 416X416 as input size. Drowning victims are detected in three stages using a YOLO-based detection technique. Even if the swimmer stayed underwater for a long time, the Deep SORT algorithm was able to track him.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

TP - True Positives

FP - False Positives

TN - True Negatives

FN - False Negatives

ACCURACY VARIABLES

COUNT

TP

220

TN

208

FP

42

FN

30

TOTAL ACCURACY

85.6%

Hazardous Activities Because of the noise removal thru image masking, the posture estimate accuracy became significantly improved. To permit the pose estimation set of rules to make greater radical judgments, the default threshold fee for the OpenPose frame components warmth map became modified from 0.2 to 0.1. Although body-by-body identifications have been simplest recognized with a possibility of 53% because of the brink adjustment, the overall system, which tested a body in real-time, became capable of discover a dangerous hobby with a great deal extra ease inside 60 seconds, with an average accuracy of 91.four percent, after the brink became modified. A near exam of the misclassified postures the various checking out units discovered that a posture became greater liable to misclassification because it approached the some distance stop of the digital digicam, suggest the want for a secondary digital digicam to enhance accuracy and verify the actual positives from the Primary digital digicam, as proven in Table II. Although using a better first-class digital digicam to restoration this hassle is a great idea, the considered necessary hardware and the near-real-time CNN strategies used to come across similarly items might not be up to conventional at present.

10. Conclusion:

Large numbers of people, including children, are constantly choking or nearly choking in the depths of swimming pools, and lifeguards are not everywhere prepared to deal with these problems. This calls for a framework to detect choking persons and alert lifeguards to such hazards. It can be set up in an international standard school where classes are held for the education of children.

11. Future Work:

As one of the most promising solutions to active drowning detection life-saving mechanisms, all licensed swimming pools can be mandated to install this drowning detection system. This model is broadly applicable to social use as it can be modified for use in other bodies of water.