Table of Contents

CHAPTER 1

INTRODUCTION

INTRODUCTION

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

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

1.1 Project Overview

Lifeguard surveillance is a complex task that is crucial for swimmer safety, though few studies of applied visual search have investigated this domain. This current study compared lifeguard and non-lifeguard search skills using dynamic, naturalistic stimuli (video clips of confederate swimmers) that varied in set size and type of drowning. Lifeguards were more accurate and responded faster to drowning targets. Differences between drowning targets were also found: passive drowning were responded to less often, but more quickly than active drowning, highlighting that passive drowning may be less salient but are highly informative once detected. Set size effects revealed a dip in reaction speeds at an intermediate set-size level, suggesting a possible change in visual search strategies as the array increases in size. Nonetheless, the ability of the test to discriminate between lifeguards and non-lifeguards offers future possibilities for training and assessing lifeguard surveillance skills.

1.2 Purpose

Drowning is the 3rd reason for the highest unintentional deaths, and that‘s why it is necessary to create trustable security mechanisms. This project aims to create a system that will be able to automatically detect drowning incidents in the swimming pool using human action detection. The drowning detection model will be used to process and classify video that will be given to the system which will be recorded using live surveillance cameras. The system will break this video in image frames and apply model over it and if the early actions of drowning like hand waving, water splashing or diving is detected then the system will set the alarm so that the lifeguards can initiate their rescue operations. The classifier model is trained using a Long-term Recurrent Convolution Network which is a combination of convolution neural network and recurrent neural network.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem

Drowning is a silent process where a victim cannot shout for help because of water blocking mouth. As soon as the drowning victim drinks enough water and gains weight and reaches the swimming pool floor, the victim's body will settle on the floor due to gravity. After several hours, microorganisms in the stomach and lungs will produce and the dead body will float on water. This floating may take several hours to several days based on water conditions. If the drowned person is taken out of the surface within five or six minutes, survival chances were more without severe brain or organ damage. Once we have the working drowning detection model we can feed live video footage of the swimming pool to it so that it can keep detecting continuously for any drowning activities. If drowning is detected it will be highlighted on the system screen as well as alarms will be raised to alert security guards so that they can initiate rescue.

2.2 References

**Title : ** A Novel Drowning Detection method for safety of swimmers.

Authors: Ajil Roy, Dr.K.Srinivasan (Department of Instrumentation andControl Engineering).

**Year : ** 2018

Description: 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.

Pros:

- The alarm receivers can be placed at different locations in the water bodies which are having high chance of drowning.
- Another major advantage of this approach unlike other approach is the ease of use in all atmospheric conditions, like rain or wind to day or night.

Cons:

- The average time a child of age between 5-10 years can hold their breath for 10 seconds underwater.
- This feature should be valid only if the GPS connectivity was alive with minimum of 10 minutes before the drowning, as a very old GPS value will give a wrong location itself.

Title **:** Automated Drowning Detection And Security in Swimming Pool.

Authors: Kanchana A, Kavya G.R, Kavitha C, Soumyashree V, Salila Hedge (Department of Electronics and Communication).

Year **:** 2019

Description: Swimming pool surveillance systems plays an essential role in safeguarding the premises. In this project, differential pressure approach is used for detection of drowning incidents in swimming pools at the earliest possible stage. The automated drowning detection system works on the principle of differential pressure. The system contains two fundamental modules: to begin with the wristband consisting of pressure sensors on the transmitter side. Second, the receiver module at the swimming pool territory should wear the wristband. The pressure at underwater is different and greater than the pressure at the air-water interface. The pressure at a particular depth is measured and set as the threshold.

Pros:

- The children's life is saved during drowning incidents in this swimming pool by lifting the acrylic plate.
- The demo system is based on pressure sensor which has an advantage of convenience, cost saving and simple algorithm.

Cons:

- The reflection and refraction of light in air water interference will affect the image quality and drowning man feature in this method does not easily distinguish swimmers and divers obviously.
- This system needs constant observation which is the main disadvantage.

**Title : **Automated and Intelligent System for Monitoring Swimming PoolSafety Based on the IoT and Transfer Learning.

Author: AzizAlotaibi(CollegeofComputerandInformation Technology).

**Year : **2020

Description: Recently, integrating the Internet of Things (IoT) and computer vision has been utilized in swimming pool automated surveillance systems. Several studies have been proposed to overcome off-time surveillance drowning incidents based on using a sequence of videos to track human motion and position. This paper proposes an efficient and reliable detection system that utilizes a single image to detect and classify drowning objects, to prevent drowning incidents. The performance of the specialized model is evaluated by using a prototype experiment that achieves higher accuracy, sensitivity, and precision, as compared to other deep learning algorithms. The collected data from different physical devices were processed by using ML techniques, to generate an action value.

Pros:

- This system utilizes the IoT and transfer learning to provide an intelligent and automated solution for off-time monitoring swimming pool safety.
- A specialized transfer-learning-based model utilizing a model pre-trained on "ImageNet", which can extract the most useful and complex features of the captured image to differentiate between humans, animals, and other objects.

Cons:

- A generative adversarial network should be applied to generate synthesis data, in order to increase the size of the training dataset.
- More classes should be added to explore and investigate the efficiency.

Title **:** Automated Vision Based Swimming Pool Surveillance System.

Authors: Darshan V, Sai Anish R, Shiddaramanaguda T, Achinthya Holla, SwethaT(Department of Computer Science and Engineering).

Year **:** 2021

Description: Automated vision based surveillance for a real time human behaviour analysis provides an efficient way of detecting the occurrence of any abnormal events amid our surroundings. The technical challenges faced encompass the need to reliably detect and track moving targets within possibly dynamic background and inference module that interprets targets behaviour patterns as events with semantic meaning. The drowning person is detected machine learning, using pi camera underwater which is trained to detect these kind of situations, the pi camera is connected to the Raspberry pi, this system is used to monitor the swimming pool, track swimmers in that, if any person is in drowning condition raspberry pi will detect it and it will send command to Arduino Nano board to lift the mesh up. With the help of stepper motors, the mesh will lift up along with the drowning person.

Pros:

- This system doesn't have to wait until life guard comes to rescue because it has uplifting mesh.
- More effective and cost efficient than previous other models.

Cons:

- Internet connection is necessary to use GPS or sending alert messages. Sometimes to send messages SIM balance may be required.
- The system can be improved by attaching an infrared LED to the swimmer's vest.

**Title : ** Drowning Detection Systemusing LRCNApproach.

Authors: Shardul Sanjay Chavan, SanketTukaramDhake, Shubham Virendra Jadhav, Prof. Johnson Mathew (Department of Computer Engineering).

**Year : ** 2022

Description: This project provides the insights of a real-time video surveillance system capable of automatically detecting drowning incidents in a swimming pool. Drowning is the third reason for the highest unintentional deaths, and that's why it is necessary to create trustable security mechanisms. Currently, most of the swimming pools security mechanisms include CCTV surveillance and lifeguards to help in drowning situations. But this method is not enough for huge swimming pools like in amusement parks. Some of security systems are using AI for drowning detection using cameras situated underwater at a fixed location and also by using floating boards having a camera mounted on the bottom side so that underwater view can be captured. But the main problem in this system arises when the pool is crowded and visions of the cameras are blocked by people.

Pros:

- This system aims to create a system that will be able to automatically detect drowning incidents in the swimming pool using human action detection.
- The classifier model is trained using a Long-term Recurrent Convolutional Network which is a combination of convolutional neural network and recurrent neural network which is suitable for large-scale visual understanding tasks such as activity recognition and image captioning.

Cons:

- Availability of better dataset, modern methodologies, and technologies with high computational power accompanied by high-quality surveillance cameras, will help to improve the accuracy of drowning detection and even can be used in adverse conditions.

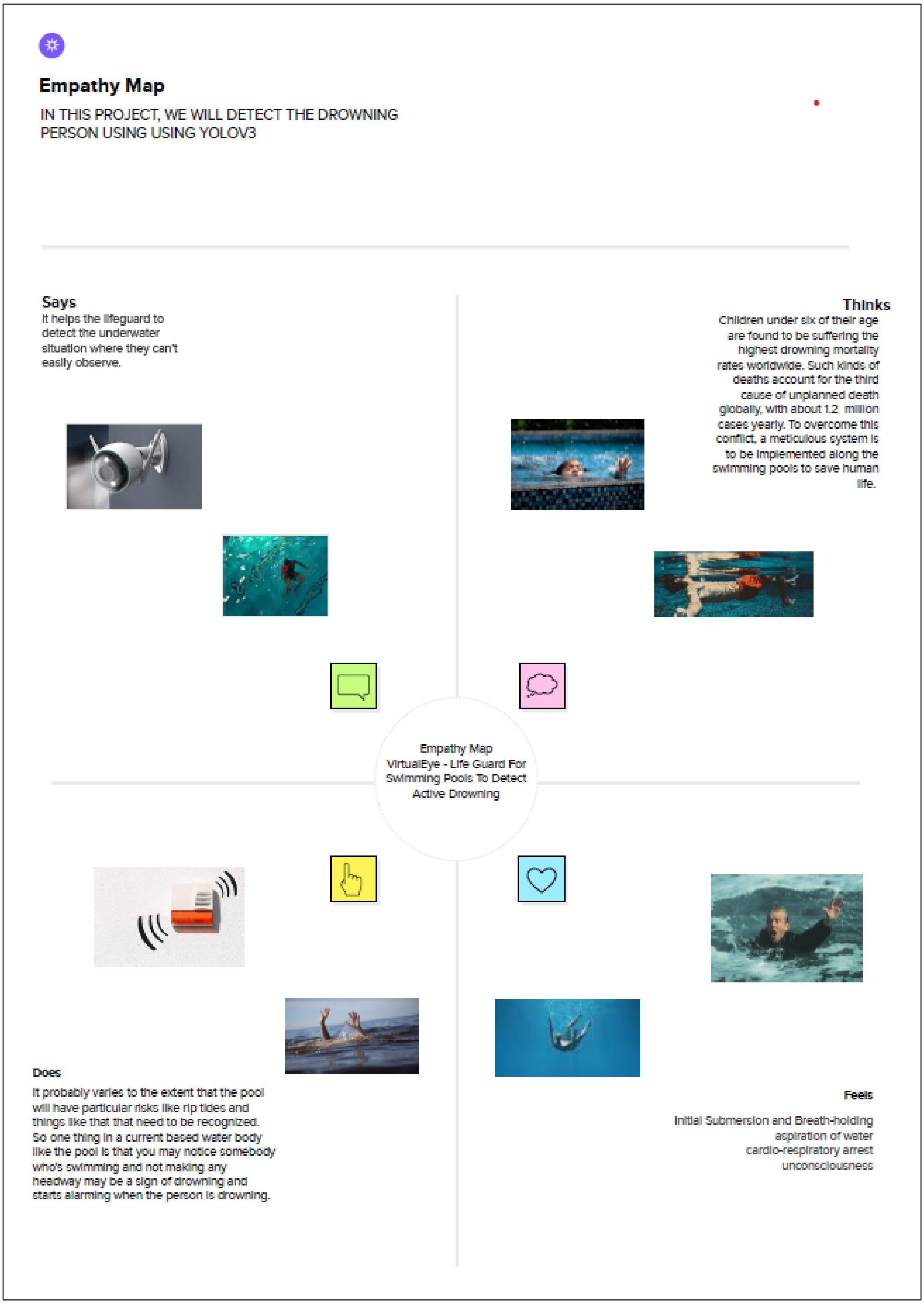
1.3 Problem Statement Definition

Safety is paramount in all swimming pools. The current systems expected to address the problem of ensuring safety at swimming pools have significant problems due to their technical aspects, such as underwater cameras and methodological aspects such as the need for human intervention in the rescue mission. The use of an automated visual-based monitoring system can help to reduce drowning and assure pool safety effectively. This study introduces a revolutionary technology that identifies drowning victims in a minimum amount of time and dispatches an automated drone to save them. Using convolutional neural network (CNN) models, it can detect a drowning person in three stages. Whenever such a situation like this is detected, the inflatable tube-mounted self-driven drone will go on a rescue mission, sounding an alarm to inform the nearby lifeguards. The system also keeps an eye out for potentially dangerous actions that could result in drowning. This system's ability to save a drowning victim in under a minute has been demonstrated in prototype experiments' performance evaluations.

CHAPTER 3

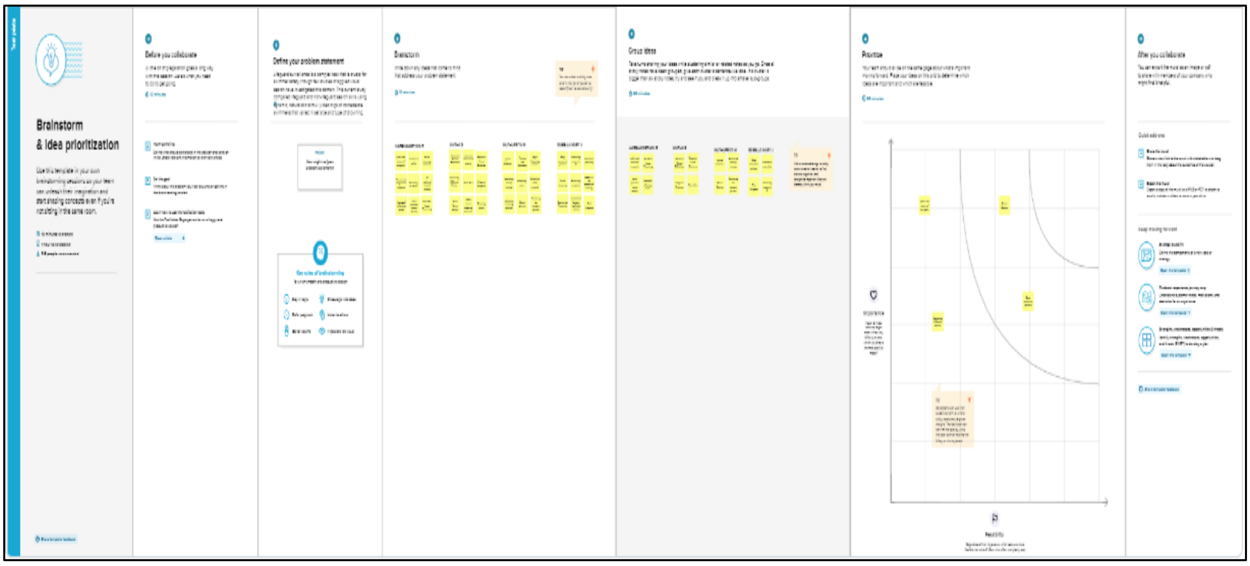
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

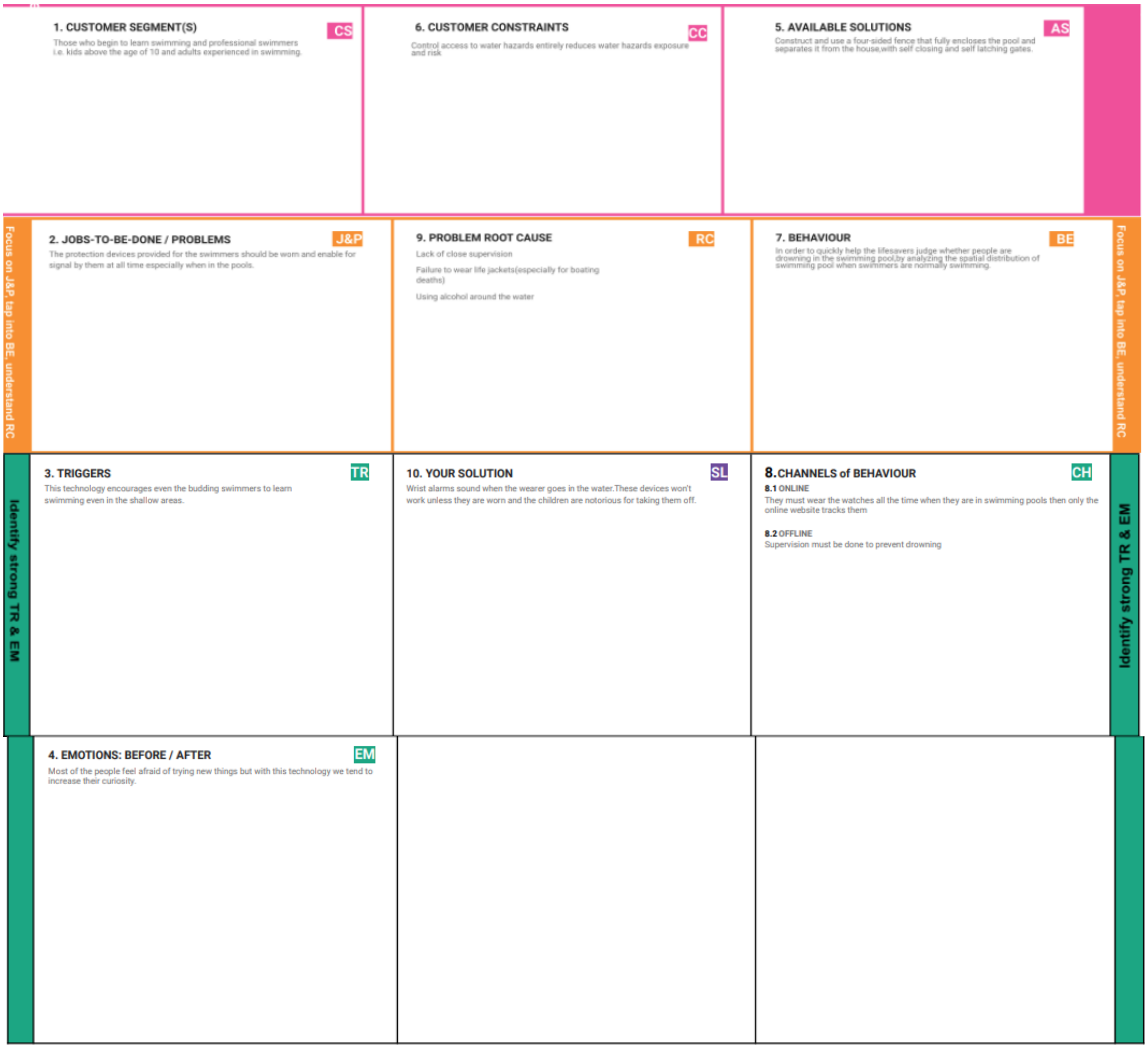
Ideation refers to the whole creative process of coming up with and communicating new ideas. It can take many different forms, from coming up with a totally new idea to combining multiple existing ideas to create a new process or organizational system. Ideation is similar to a practice known as brainstorming.



3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Swimming is one of the best exercises that help 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 1.2millioncasesyearly.
2.	Idea/Solution description	To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life. By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning.
3.	Novelty/Uniqueness	Making use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher than an alert will be generated to attract lifeguards attention.
4.	Social Impact/ Customer Satisfaction	By the use of this technology, customers can feel secure to swim. Using this camera we can monitor the customer body movements so we Can easily get back with them if they drown.
5.	Business Model(Revenue Model)	Swim Eye is a computer vision detection system for the prevention of drowning incidents in swimming pools.
6.	Scalability of the Solution	This is suitable for any swimming pools. New or Existing pools (for all shapes and sizes) and any construction type.

3.4 Problem Solution fit



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Details	Registration through Form Registration through Gmail Registration through LinkedIn
FR-4	Payment Details	Pay through GPay Pay through PhonePe Pay through Debit Card or Cash

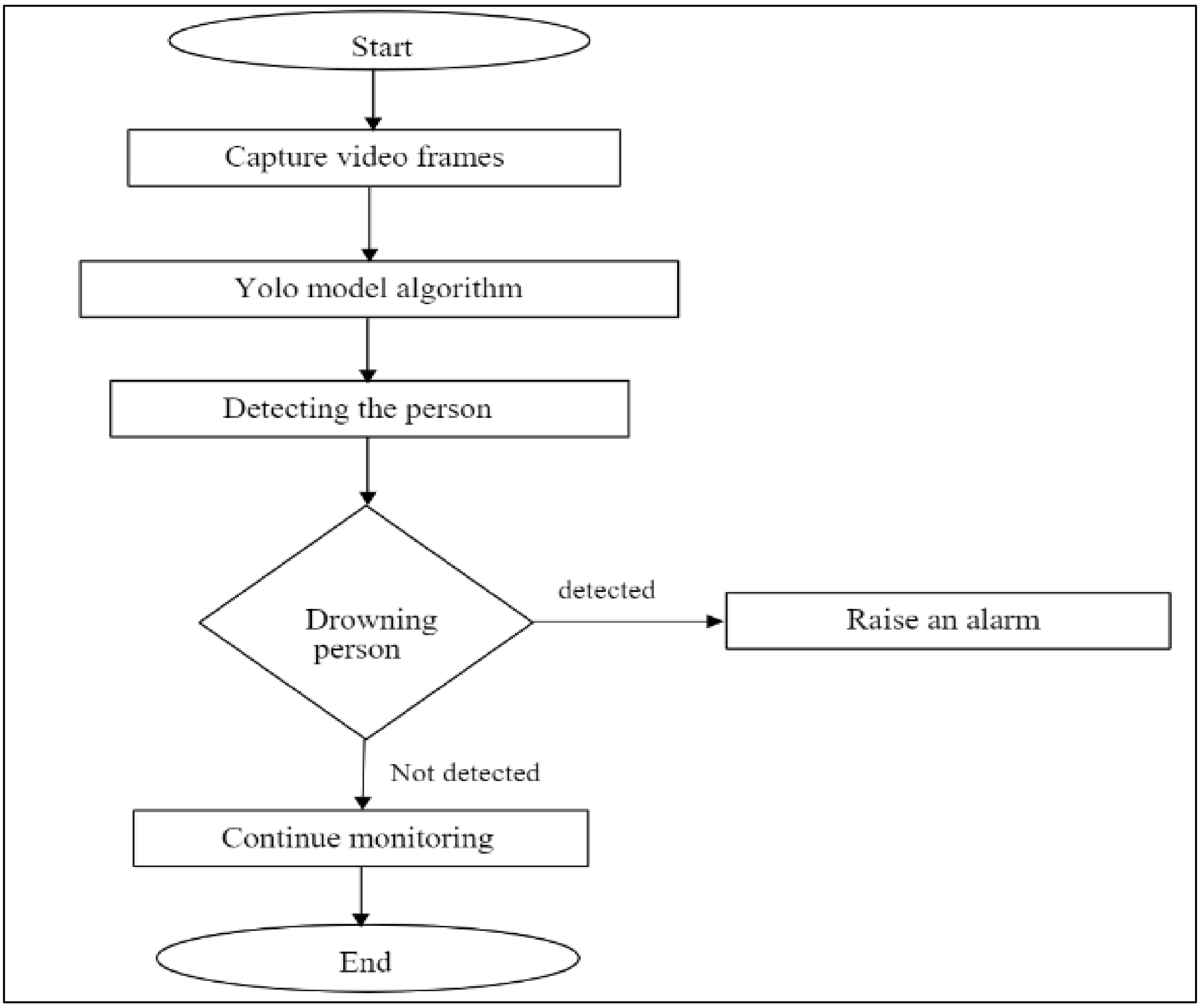
4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Eco-Friendly.
NFR-2	**Security**	Observing each and every body movements of the swimmers.
NFR-3	**Reliability**	Suitable for all the swimming pools.
NFR-4	**Performance**	Life guards can visually assess the developing situation within seconds of the event first occurring, and initiate the rescue procedure when necessary.
NFR-5	**Availability**	24/7 monitoring cameras.
NFR-6	**Scalability**	This is suitable for all swimming pools (new or existing pools) for any shapes, sizes and any construction types.

CHAPTER 5
PROJECT DESIGN

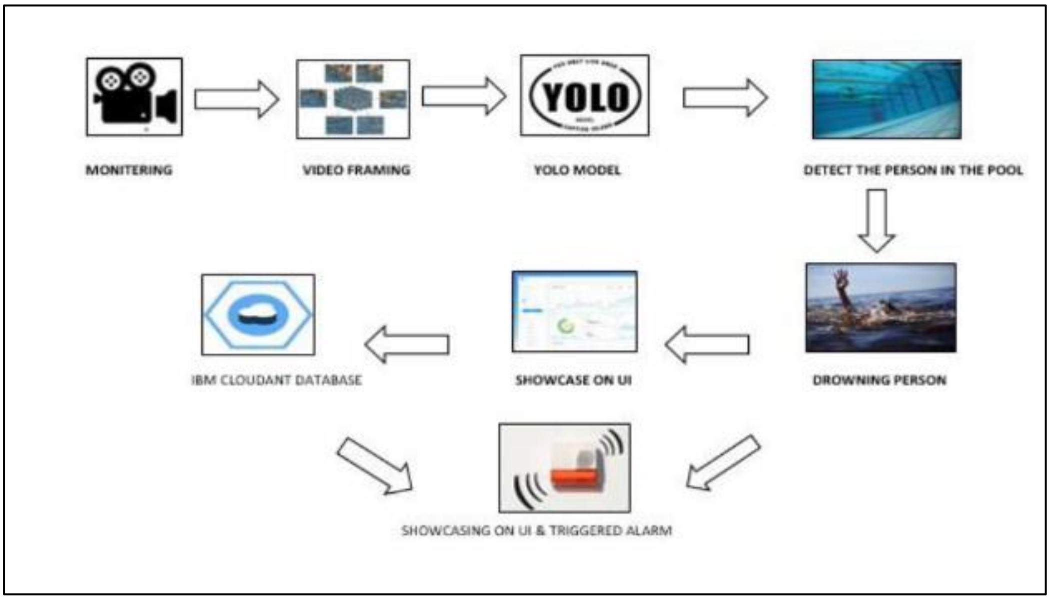
5.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM)..

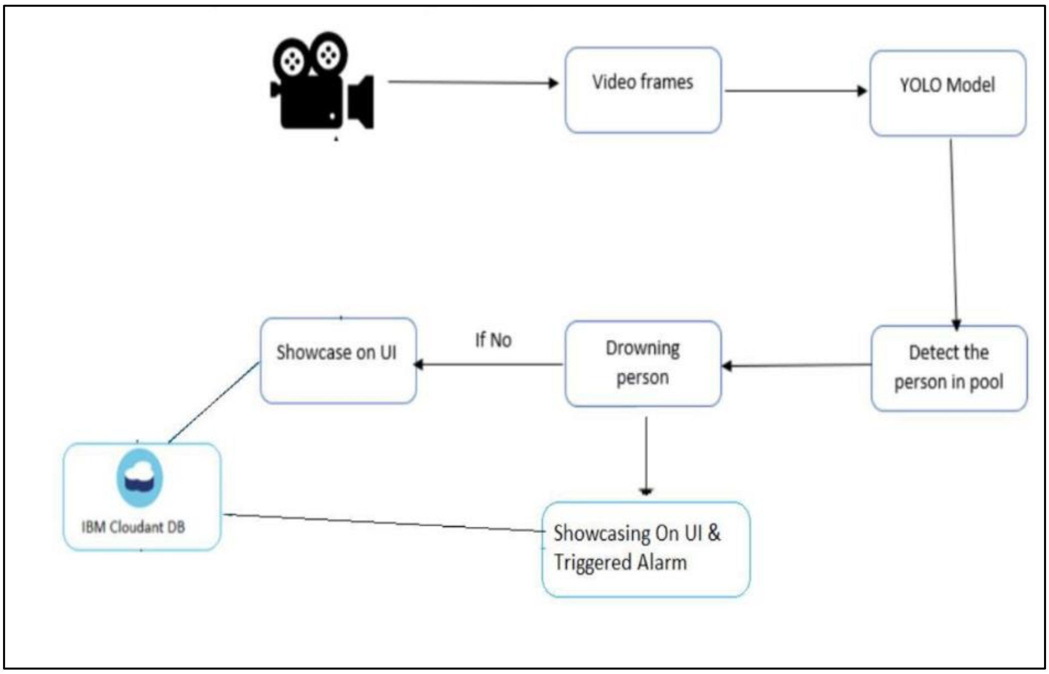


5.3 Solution and Technical Architecture

5.3.1 Solution Architecture



5.3.2 Technical Architecture



CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation

Velocity:

Imagine we have 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

For Sprint-1

The Average Velocity (AV) is: AV = Sprint Duration / velocity = 8 / 6 = 1.3V
For Sprint-2

The Average Velocity (AV) is: AV = Sprint Duration / velocity = 14 / 6 = 2.3V
For Sprint-3

The Average Velocity (AV) is: AV = Sprint Duration / velocity = 16 / 6 = 2.6V
For Sprint-4

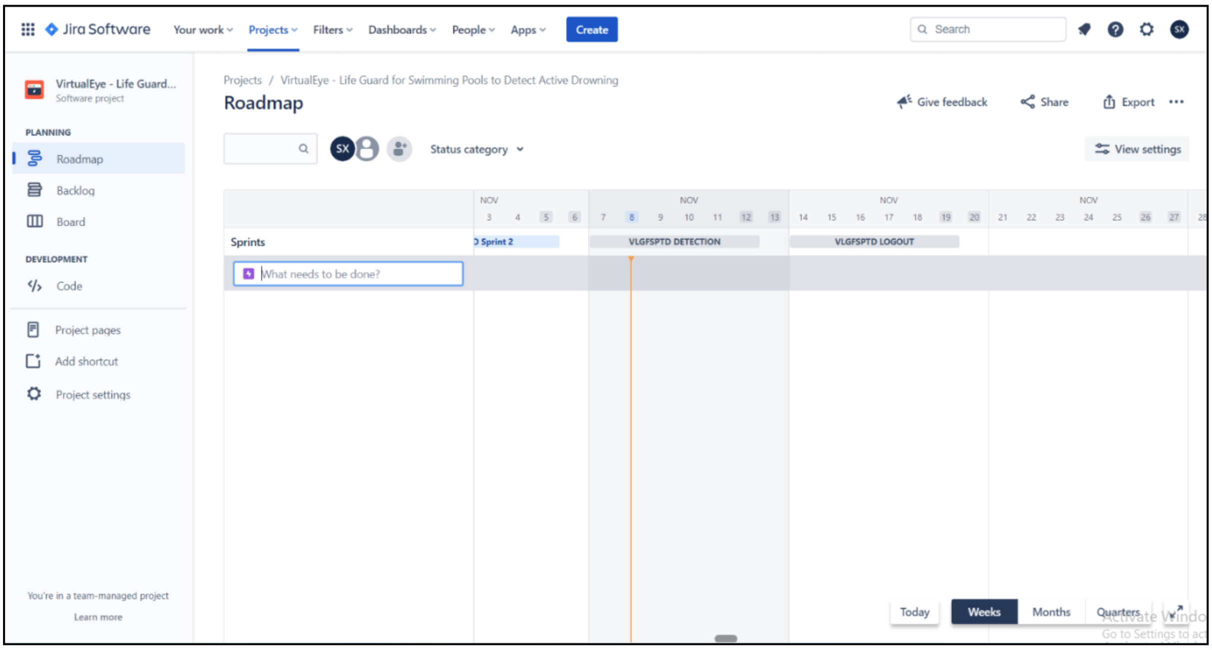
The Average Velocity (AV) is: AV = Sprint Duration / velocity = 12/ 6 = 2.0V
TOTAL TEAM AVERAGE VELOCITY = 2.08

6.2 Sprint Delivery Schedule

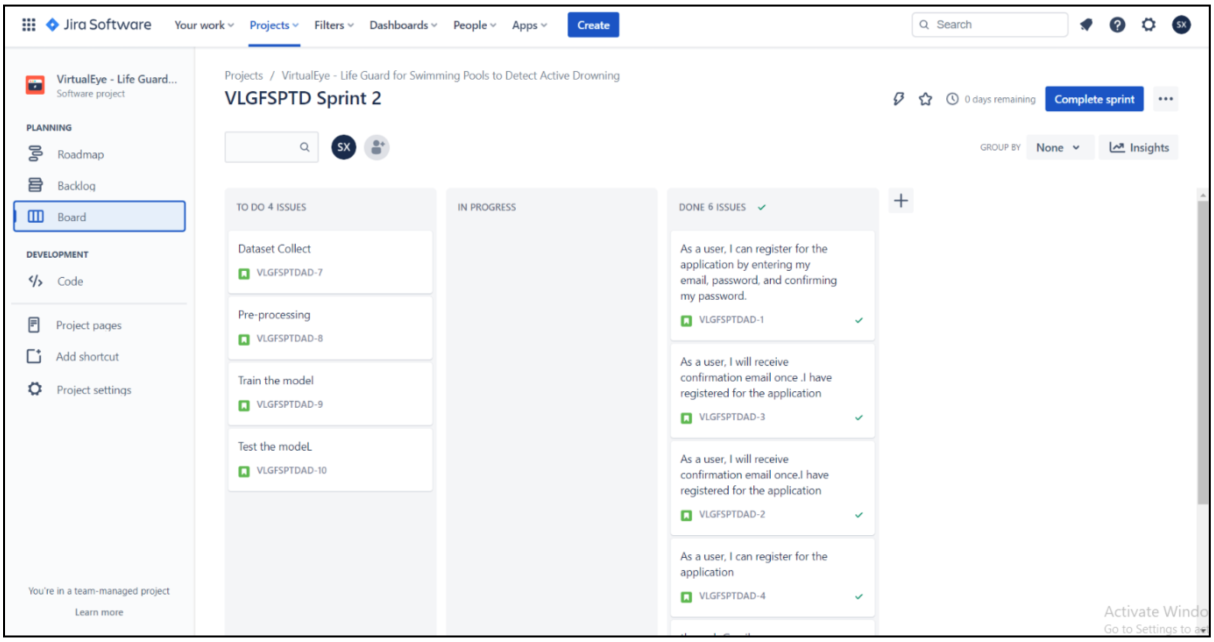
Sprint	Total Story Points	Duration	Sprint End Data (Planned)	Story Points Completed(as on Planned End Date)	Story Points Completed (as on Planned end Date)	Sprint Release Date(Actual)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	6	29 Oct 2022
Sprint-2	14	6 Days	06 Nov 2022	06 Nov2022	12	06 Nov2022
Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	11	12 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

6.3 Reports from JIRA

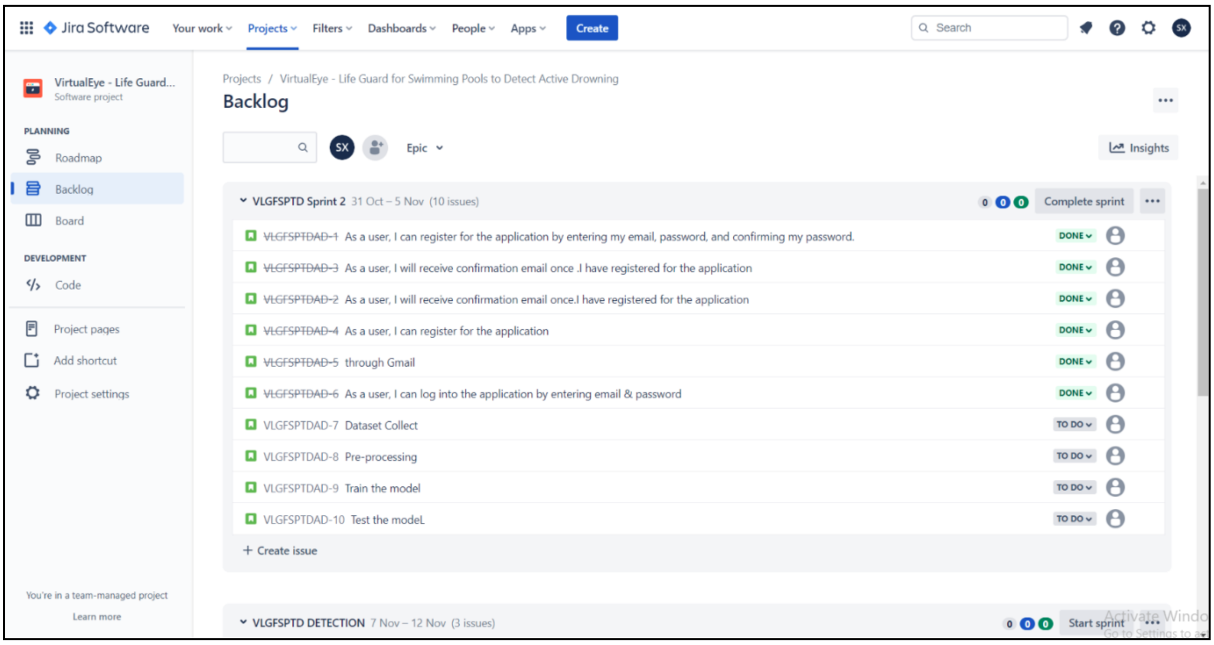
Roadmap



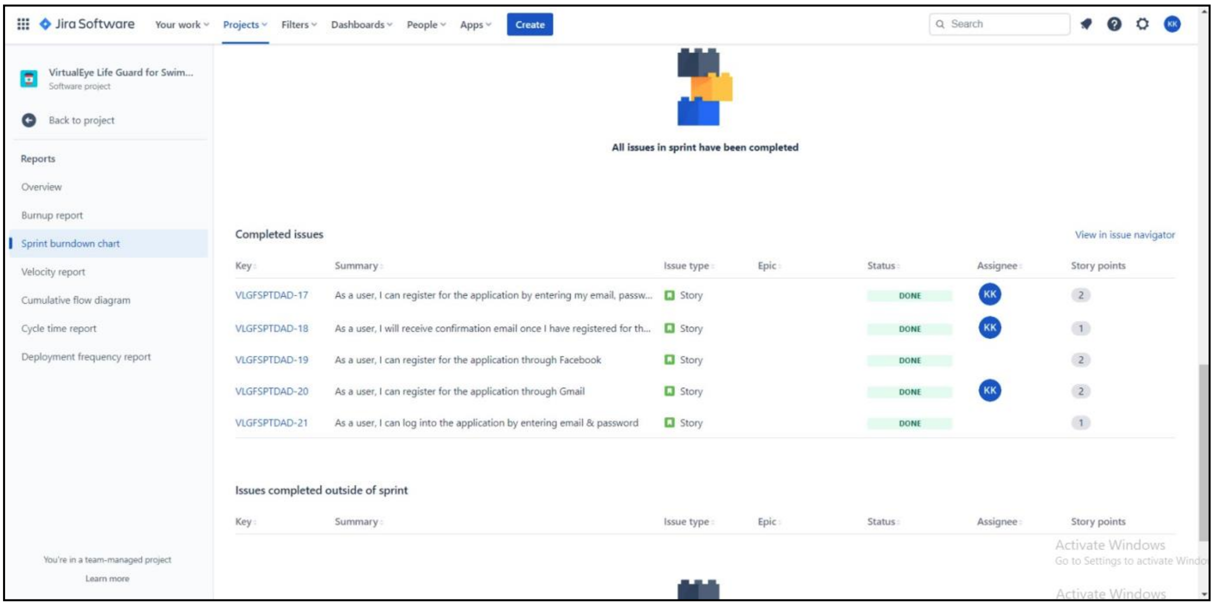
VLGFSPTD Sprint 2



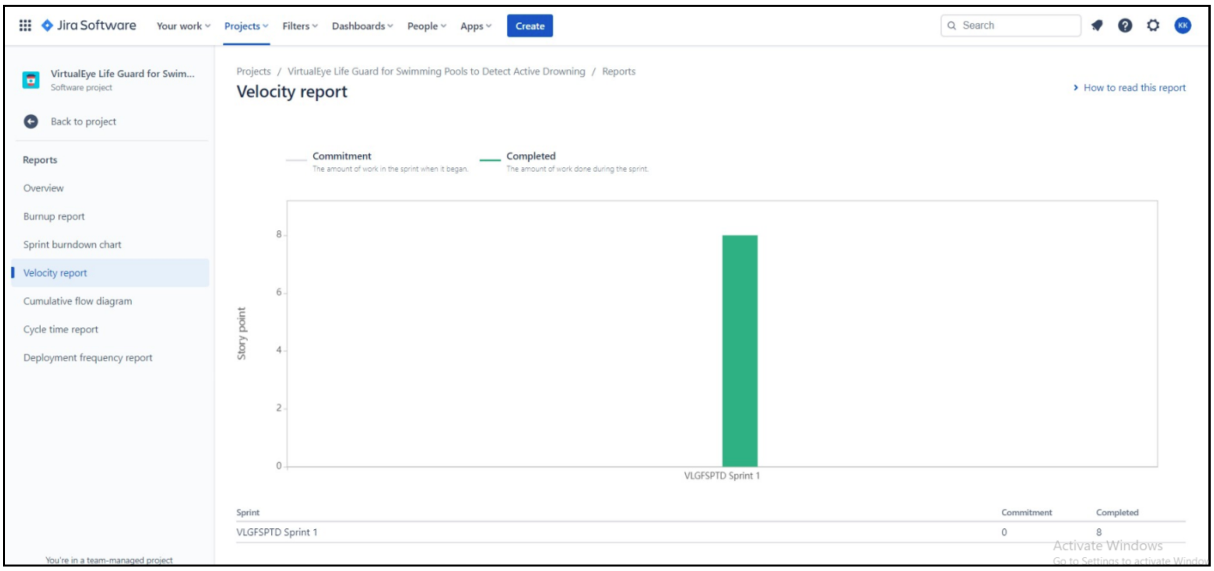
Backlog



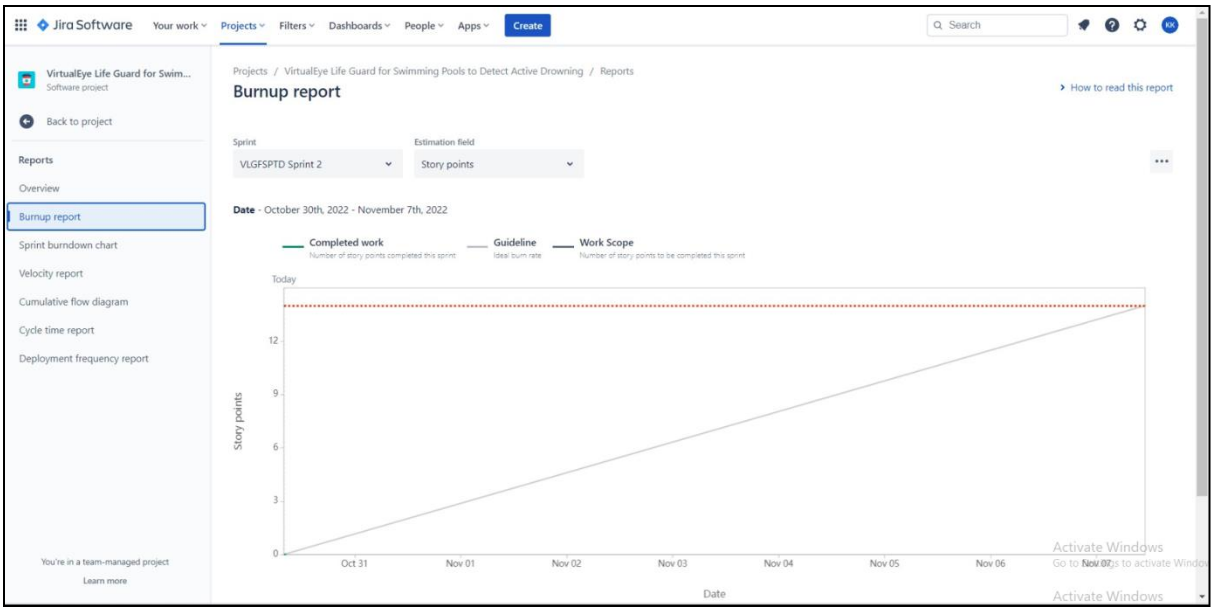
All issues in Sprint have been completed



Velocity Report



Burn up Report



CHAPTER 7

CODING & SOLUTIONING

7.1 Feature 1

Class 1

```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import load_img, img_to_array
import numpy as np
import easygui
import os
import serial
print(tf.__version__)
#feature training
train_datagen = ImageDataGenerator(
       # reducing/normalizing the pixels
        rescale=1./255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)
    #connecting the image augmentation tool to our dataset
        train_set = train_datagen.flow_from_directory(
        'Class1/training_set',
     #final size of the images that will be fed into the ann
        target_size=(64, 64),
     # number of images that we want to have in each batch
        batch_size=32,
     # we have binary classification --> binary class mode
        class_mode='binary')
    #only rescaling but no transformations
     test_datagen = ImageDataGenerator(rescale=1./255)
    #connecting to the test data
        test_set = test_datagen.flow_from_directory(
       'Class1/test_set',
        target_size=(64, 64),
        batch_size=32,
        class_mode='binary')
         print(test_set)

#--------------------- Building CNN --------------------#
# initializing CNN as sequential layers
cnn = tf.keras.models.Sequential()
# Step 1: Convolution to get the Feature Map
cnn.add(tf.keras.layers.Conv2D(filters = 32, kernel_size = 3, activation = 'relu',
```

```
input_shape=[64,64,3]))
# Step 2: Max Pooling
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2 ,strides=2))
#adding a second convolutional layer
cnn.add(tf.keras.layers.Conv2D(filters = 32, kernel_size = 3, activation = 'relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2 ,strides=2))
# Step 3: Flattening
cnn.add(tf.keras.layers.Flatten())
# Step 4: Full Connection
cnn.add(tf.keras.layers.Dense(units = 128, activation = 'relu'))
# Step 5: Output Layer
cnn.add(tf.keras.layers.Dense(units = 1, activation = 'sigmoid'))
#--------------------- Training the CNN --------------------#
#compiling the CNN
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =
['accuracy'])
#training the CNN on the training set and evaluating it on the test set
cnn.fit(x = train_set, validation_data = test_set, epochs = 25)
cnn.save('model/save1',overwrite=True,
include_optimizer=True,
save_format=None,
    signatures=None,
    options=None,
    save_traces=True,)
cnn.save('model/Class1/model_Class1.h5')
# Testing

image11 = easygui.fileopenbox()
test_image2 =   load_img(image11, target_size = (64, 64))
test_image2 = img_to_array(test_image2)
test_image2 = np.expand_dims(test_image2, axis = 0)
# cnn prediction on the test image
result2 = cnn.predict(test_image2)
print(result2)
if result2[0][0] == 1:
   prediction2 = 'Present'
else:
   prediction2 = 'Empty'
   print(prediction2)
```

Class 2:

```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
```

```
from tensorflow.keras.utils import load_img,img_to_array
import numpy as np
import easygui
import os
import serial

print(tf.__version__)

#feature training
train_datagen = ImageDataGenerator(
# reducing/normalizing the pixels
  rescale=1./255,
     shear_range=0.2,
     zoom_range=0.2,
     horizontal_flip=True)
#connecting the image augmentation tool to our dataset
train_set = train_datagen.flow_from_directory(
      'Class2/training_set',
      #final size of the images that will be fed into the ann
      target_size=(64, 64),
      # number of images that we want to have in each batch
      batch_size=32,
      # we have binary classification --> binary class mode
      class_mode='binary')
      print(test_set)
#--------------------- Building CNN --------------------#
# initializing CNN as sequential layers
cnn = tf.keras.models.Sequential()
# Step 1: Convolution to get the Feature Map
cnn.add(tf.keras.layers.Conv2D(filters = 32, kernel_size = 3, activation = 'relu',
input_shape=[64,64,3]))
# Step 2: Max Pooling
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2 ,strides=2))
#adding a second convolutional layer
cnn.add(tf.keras.layers.Conv2D(filters = 32, kernel_size = 3, activation = 'relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2 ,strides=2))
# Step 3: Flattening
cnn.add(tf.keras.layers.Flatten())
# Step 4: Full Connection
cnn.add(tf.keras.layers.Dense(units = 128, activation = 'relu'))
# Step 5: Output Layer
cnn.add(tf.keras.layers.Dense(units = 1, activation = 'sigmoid'))
```

```
#--------------------- Training the CNN --------------------#
#compiling the CNN
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =
['accuracy'])
#training the CNN on the training set and evaluating it on the test set
cnn.fit(x = train_set, validation_data = test_set, epochs = 25)
cnn.save('model/save1',overwrite=True,
include_optimizer=True,

save_format=None,
  signatures=None,
  options=None,
  save_traces=True,)
cnn.save('model/Class2/model_Class2.h5')
# Testing
image11 = easygui.fileopenbox()
test_image2 = load_img(image11, target_size = (64, 64))
test_image2 = img_to_array(test_image2)
test_image2 = np.expand_dims(test_image2, axis = 0)
# cnn prediction on the test image
result2 = cnn.predict(test_image2)
print(result2)
if result2[0][0] == 1:
   prediction2 = 'Human'
else:
   prediction2 = 'Animal'
   print('prediction2')
```

7.2 Feature 2

Testing Serial.py

```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import load_img,img_to_array
import numpy as np

import easygui
from keras.models import load_model
import os
import serial
from serial import Serial
import time
print(tf.__version__)
model1 = load_model('model/Class1/model_Class1.h5')
model2 = load_model('model/Class2/model_Class2.h5')
# Testing
image11 = easygui.fileopenbox()
test_image2 = load_img(image11, target_size = (64, 64))
test_image2 = img_to_array(test_image2)
test_image2 = np.expand_dims(test_image2, axis = 0)
# cnn prediction on the test image
result2 = model1.predict(test_image2)
print(result2)
if result2[0][0] == 1:
  result3 = model2.predict(test_image2)
  if result3[0][0] == 1:
    prediction2 = 'Human Drowning Detected'
    SerialObj =Serial('COM7')
    SerialObj.baudrate = 9600
    SerialObj.bytesize = 8
    SerialObj.parity   ='N'
    SerialObj.stopbits = 1
  time.sleep(3)
    SerialObj.write(b'a')
    SerialObj.close()
   ser = serial.Serial("COM7", 9600)
   data = "X61"  # a  -> Human
   ser.write(data)
   s = ser.read(9)
  else:
```

```
        prediction2 = 'AnimalDrowning Detected'
        SerialObj = serial.Serial('COM7')
        SerialObj.baudrate = 9600
        SerialObj.bytesize = 8
        SerialObj.parity   ='N'
        SerialObj.stopbits = 1
        time.sleep(3)
        SerialObj.write(b'a')
        SerialObj.close()
        ser = serial.Serial("COM7", 9600)
       data = "X62"  # b -> Animal
       ser.write(data)
       s = ser.read(9)
else:
   prediction2 = 'Normal '
   SerialObj = Serial('COM6')
   SerialObj.baudrate = 9600
   SerialObj.bytesize = 8
   SerialObj.parity   ='N'
   SerialObj.stopbits = 1
   time.sleep(3)
   SerialObj.write(b'a')
   SerialObj.close()
ser = serial.Serial("COM7", 9600)
  data = "X63"  # c -> Empty
  ser.write(data)
  s = ser.read(9)
  print(prediction2)
```

Testing WithoutSerial.py

```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import load_img,img_to_array
import numpy as np

import easygui
from keras.models import load_model
import scipy.integrate as integrate
import scipy
import os
import serial
print(tf.__version__)
```

```
model1 = load_model('model/Class1/model_Class1.h5')
model2 = load_model('model/Class2/model_Class2.h5')
# Testing
image11 = easygui.fileopenbox()
test_image2 = load_img(image11, target_size = (64, 64))
test_image2 = img_to_array(test_image2)
test_image2 = np.expand_dims(test_image2, axis = 0)
# cnn prediction on the test image
result2 = model1.predict(test_image2)
print(result2)
if result2[0][0] == 1:
  result3 = model2.predict(test_image2)
  if result3[0][0] == 1:
    prediction2 = 'Drowning#human
  else:
    prediction2 = 'Drowning#Animal'
else:
  prediction2 = 'Empty'
  print(prediction2)
```

CHAPTER 8
TESTING

8.1 TESTCASES

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

8.2 USER ACCEPTANCE TESTING

1. Defect Analysis

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

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
Version Control	1	0	0	1

2. Test Case Analysis

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

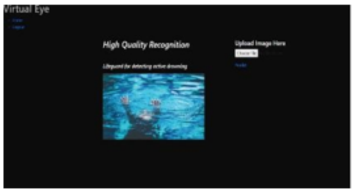
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	

2.	Accuracy	Training Accuracy - 28 Validation Accuracy -44	

CHAPTER 10
ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

- It represents an additional level of safety and protection for swimmers.
- It ensures effective and reliable drowning detection by limiting the number of alarms generated by disturbance factors.
- Prevents drowning accidents by improving the rescue time of the lifeguards.
- It helps the lifeguard to detect the underwater situation where they can't easily observe.
- It identifies both active and passive drowning targets more frequently and more quickly than control participants.

10.2 DISADVANTAGES

- Glare and specular reflection3 should be avoided and points to particular visibility problems for pools that are more than 16 m wide.
- Even when glare is not an issue, it is likely to be difficult fora lifeguard on one side of the pool to see objects beneath the water on the far side.
- There can also be visibility issues on the near side of the pool unless the lifeguard is positioned close to the pool edge.
- Drowning claims the lives of more children under five than any other cause of accidental death.
- And the vast majority of these tragedies happen in private swimming pools, under the parents’ supervision.

CHAPTER 11

CONCLUSION

CONCLUSION

The outputs achieved will predict the class names for a batch of frames of the videos given as input. The predicted class name having the highest probabilities will be detected as the action being performed in the swimming pool. The predicted class name having maximum probability can be displayed as the confidence variable. After the successful completion of the project, one can observe the video surveillance and rely on the drowning detection system. An alarm will be raised if someone is detected as drowning. Drowning preventive measures can be performed due to early alerts raised by the system. Once we have the working drowning detection model we can feed live video footage of the swimming pool to it so that it can keep detecting continuously for any drowning activities. If drowning is detected it will be highlighted on the system screen as well as alarms will be raised to alert security guards so that they can initiate rescue.

CHAPTER 12

FUTURE SCOPE

FUTURE SCOPE

Availability of better dataset, modern methodologies, and technologies with high computational power accompanied by high-quality surveillance cameras, will help to improve the accuracy of drowning detection & even can be used in adverse conditions. After the implementation of all these essentials, this system also can be used on sea beaches for drowning detection

CHAPTER 13

APPENDIX

13.1 SOURCE CODE

Index page

```
<!DOCTYPE html>
<html>
<head>
   <link rel="stylesheet" href="home.css">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>
      VirtualEye-Life Guard for Swimming Pools to detect Active Drowning
   </title>
</head>
<body>
   <div class="container">
      <div class="topnav">

       <a href="#demo">Demo</a>
        <a href="C:\Users\muthu\Downloads\IBM
PROJECT\register.html">Register</a>
        <a href="C:\Users\muthu\Downloads\IBM
PROJECT\login.html">Login</a>
        <a class="active" href="C:\Users\muthu\Downloads\IBM
PROJECT\home.html">Home</a>
   </div>
        <h1>VIRTUAL EYE-LIFEGUARD FOR SWIMMING POOLS TO
DETECT ACTIVE DROWNING</h1>
<p>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. </p>
<img src="C:\Users\muthu\Downloads\IBM PROJECT\im 1.png" alt="">
<img src="C:\Users\muthu\Downloads\IBM PROJECT\img 3a.png" alt=""
height="415px">
</body> </html>
```

Login page

```
<!DOCTYPE html>
<html>
<head>
   <link rel="stylesheet" href="login.css">
   <script src="login.js"></script>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>
      VirtualEye-Life Guard for Swimming Pools to detect Active Drowning
   </title>
</head>
<body>
   <div class="container">
  <div class="topnav">
   <a href="#demo">Demo</a>
   <a href="C:\Users\muthu\Downloads\IBM
PROJECT\register.html">Register</a>
   <a class="active" href="C:\Users\muthu\Downloads\IBM
PROJECT\login.html">Login</a>
   <a href="C:\Users\muthu\Downloads\IBM
PROJECT\home.html">Home</a>
  </div>
  </div>
  <form class="box" action="login.html" method="POST" name="form">
   <div class="div1">
   <h1>LOGIN</h1>
   <label>Enter required email id:</label>
   <input id="email" type="alphanumeric" size="35">
   <label>Enter Password:</label>
   <input id="pass" type="password" size="35">
   <button type="submit" class="block" id="pass" value="Login"
onclick="validate()">Login</button>

    <br>
   <a href="">Forget Password?</a>
  </form>
</div>
</body>
</html>
```

13.2 GitHub & Project Demo Link

GitHub link

https://github.com/IBM-EPBL/IBM-Project-3738-1658594240

Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-3738-1658594240/blob/main/DEMO%20VIDEO/Demo.mp4