A Mini Project Report on

DRIVER DROWSINESS DETECTION

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

In

partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

K. Samuel (17BD1A054R)

Ande Yashwanth (17BD1A0544)

Ketan Jain (17BD1A054T)

K. Anirudh (17BD1A054J)

Under the guidance of

V. Archana Reddy
Assistant Professor
Department of CSE

Department of Computer Science and Engineering

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTUH

3-5-1206, Narayanaguda, Hyderabad – 500029

2020-2021

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(Accredited by NBA & NAAC, Approved By A.I.C.T.E., Reg by Govt of Telangana State & Affiliated to JNTU, Hyderabad)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **DRIVER DROWSINESS DETECTION** being submitted by

K. Samuel	**(17BD1A054R)**
Ande Yashwanth	**(17BD1A0544)**
Ketan Jain	**(17BD1A054T)**
K. Anirudh	**(17BD1A054J)**

In partial fulfillment for the award of the degree of **Bachelor of Technology** in **Computer Science and Technology** affiliated to the **Jawaharlal Nehru Technological University, Hyderabad.**

Internal Guide **Head of the Department**

(V. ARCHANA REDDY) **(DR. S. PADMAJA)**

Submitted for Viva Voce Examination held on ________________________

External Examiner

Vision of KMIT

Producing quality graduates trained in the latest technologies and related tools and striving to make India a world leader in software and hardware products and services. To achieve academic excellence by imparting in depth knowledge to the students, facilitating research activities and catering to the fast growing and ever- changing industrial demands and societal needs.

Mission of KMIT

- To provide a learning environment that helps students to enhance problem solving skills, be successful in their professional lives and to prepare students to be lifelong learners through multi model platforms and educating them about their professional, and ethical responsibilities
- To establish Industry Institute Interaction to make students ready for the industry.
- To provide exposure to students to the latest tools and technologies hardware and software. the area of hardware and software
- To promote research–based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises.
- To induce in the students a spirit of nationalism which will enable the student to develop and understand India's problems and to encourage them to come up with effective solutions for the same.
- To support the faculty in their endeavours to accelerate their learning curve in order to continue to deliver excellent service to students.

Vision & Mission of CSE

Vision of the CSE

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the CSE

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefits the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

PROGRAM OUTCOMES (POs)

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3. Design/development of solutions: Design solutions for complex engineering problem and design system component or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create select, and, apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to societal, health, safety. Legal und cultural issues and the consequent responsibilities relevant to professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Information Technology solutions for social upliftment.

PSO2: Shall have expertise on the evolving technologies like Mobile Apps, CRM, ERP, Big Data, etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

PROJECT OUTCOMES

P1: This project helps in alerting the driver when he/she is in drowsy state.

P2: The project has a email notification system which sends email to legal authority .

P3: Due to less hardware used it makes the system cost effective.

P4: The project is built using emerging technologies.

L – LOW
M – MEDIUM
H – HIGH

PROJECT OUTCOMES MAPPING PROGRAM OUTCOMES

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
P1		M				M	M					
P2		M				M	M					
P3	M		M							M	M	
P4		M			M							

PROJECT OUTCOMES MAPPING PROGRAM SPECIFIC OUTCOMES

PSO	PSO1	PSO2
P1	M	
P2	M	
P3		M
P4		M

PROJECT OUTCOMES MAPPING PROGRAM EDUCATIONAL OBJECTIVES

PEO	**PEO1**	**PEO2**	**PEO3**	**PEO4**
P1		M	L	
P2		M	L	
P3	M	L		
P4	M			M

DECLARATION

We hereby declare that the project report entitled “**DRIVER DROWSINESS DETECTION”** is done in the partial fulfillment for the award of the Degree in Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhere else.

K. SAMUEL (17BD1A054R)

ANDE YASHWANTH (17BD1A0544)

KETAN JAIN (17BD1A054T)

K. ANIRUDH (17BD1A054J)

ACKNOWLEDGMENT

We take this opportunity to thank all the people who have rendered their full support to our project work. We render our thanks to **Dr. Maheshwar Dutta**, B.E., M Tech., Ph.D., Principal who encouraged us to do the Project. We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project. We express our sincere gratitude to **Mr. S. Nitin**, Director and **Mrs. Deepa Ganu**, Dean Academics for providing an excellent environment in the college. We are also thankful to **Dr. S. Padmaja**, Head of the Department for providing us with both time and amenities to make this project a success within the given schedule. We are also thankful to our guide **V. Archana Reddy**, for his/her valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project. We sincerely thank our friends and family for their constant motivation during the project work.

K. SAMUEL (17BD1A054R)

ANDE YASHWANTH (17BD1A0544)

KETAN JAIN (17BD1A054T)

K. ANIRUDH (17BD1A054J)

CONTENT

ABSTRACT

Nowadays, more and more professions require long-term concentration. Drivers must keep a close eye on the road, so they can react to sudden events immediately.Drowsiness and Fatigue of drivers are amongst the significant causes of road accidents. Every year, they increase the amounts of deaths and fatalities injuries globally. Therefore, there is a need to develop the systems that will detect and notify a driver of his bad psycho-physical condition, which could significantly reduce the number of fatigue-related car accidents. Here we design a system that deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence which locate, track, and analyze both the drivers face and eyes .The system measures the drowsiness associated with extended eye closure and alerts the driver with a beep sound if he closes the eyes for a long time and also start recording the video and send it to the an emergency contact as email in order to notify about the situation .

LIST OF FIGURES

CHAPTER -1

1. INTRODUCTION

1.1 Purpose of Project

Drowsiness is simply defined as “a state of near sleep due to fatigue”. It is technically distinct from fatigue, which has been defined as a “disinclination to continue performing the task at hand”. The effects of sleepiness and fatigue are very much the same. Fatigue affects mental alertness, decreasing an individual’s ability to operate a vehicle safely and increasing the risk of human error that could lead to fatalities and injuries. Sleepiness slows reaction time, decreases awareness, and impairs judgment. Fatigue and sleep deprivation impact all transportation operators (for example: airline pilots, truck drivers, and railroad engineers).In both conditions, driver can’t focus on primary task of driving which may enhance the likelihood of crash occurrence. With the ever-growing traffic conditions, this problem will further deteriorate. For this reason, it is necessary to develop driver alertness system for accident prevention due to Driver Drowsiness.

1.2 Problems with Existing System

Existing detection systems uses iris sensor for the drowsiness of the driving force. A complex distributed sensors model that's particularly suitable for measuring blink of the driving force , accident and hand position on a wheel . These sensors are often utilized in automotive active safety systems that aim at detecting driver’s fatigue, a serious issue to stop road accidents. The key point of this approach is to style a prototype of sensor units, in order that it can function platform for integrating different sorts of sensors into the wheel . The wheel is slowed or stopped counting on the condition. The hardware required here is more. Installation and maintenance is expensive. The system if fails in one way, there is no other alternate to alert the driver and this may worsen the situation even more.

1.3 Proposed System

The proposed system uses deep learning for the analysis of whether eye is open or closed. The algorithm used for the analysis of this is Convolution Neural Network (CNN). The data set is taken for the analysis of whether the eye is closed or open. The dataset are used for the training of the machine. The webcam takes the image of the driver. The taken images is compared with the already trained images for the alerting purpose.If the driver is in a drowsy state a beep sound is a used to alert the driver and also the drowsy state of the driver is recorded and send to an emergency contact and legal authorities through email . Here the hardware required is less and maintenance is not costly. Moreover the notification alert system will help reduce the accident cases that are caused due to drowsiness.

1.4 Flow Diagram

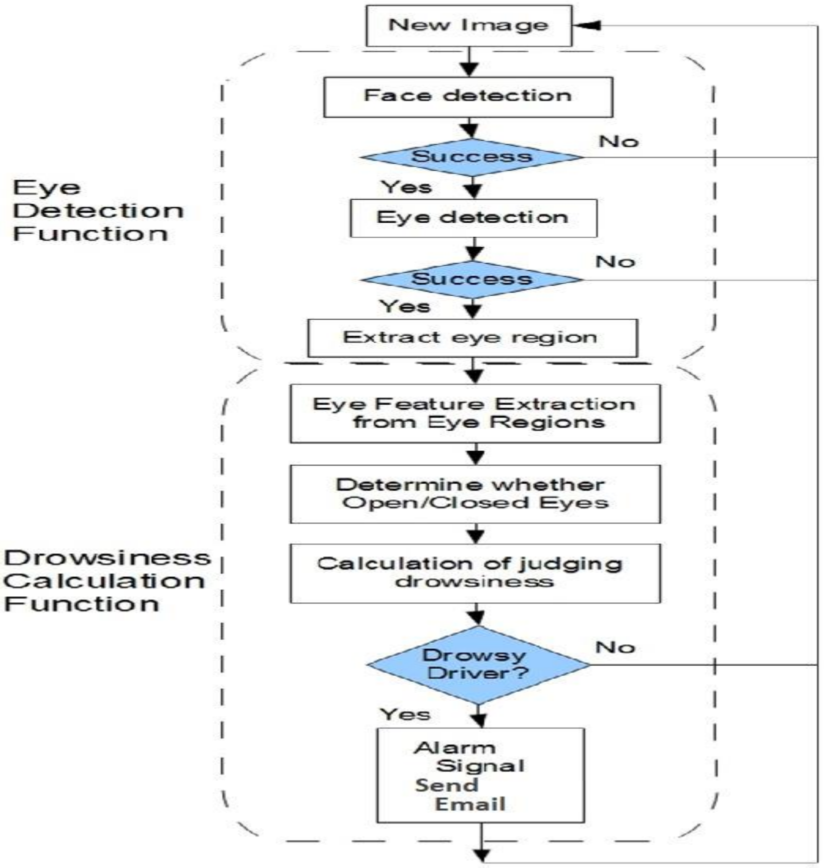


Fig 1.4.1 Flow Diagram

CHAPTER -2

2. SYSTEM RERUIREMENT SPECIFICATIONS

2.1 What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

2.2 Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

2.3 Requirements Specification Document

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS

document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application.

There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and non functional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

The purpose of SRS (Software Requirement Specification) document is to describe the external behaviour of the application developed or software. It defines the operations, performance and interfaces and quality assurance requirement of the application or software. The complete software

requirements for the system are captured by the SRS. This section introduces the requirement specification document for Driver Drowsiness Detection which enlists functional as well as non-functional requirements.

2.4 Functional Requirements

For documenting the functional requirements, the set of functionalities supported by the system are to be specified. A function can be specified by identifying the state at which data is to be input to the system, its input data domain, the output domain, and the type of processing to be carried on the input data to obtain the output data. Functional requirements define specific behaviour or function of the application. Following are the functional requirements:

1. The system captures user’s face.
2. The system detects user’s face and eyes.
3. The system alerts the driver if he is drowsy using an alert sound.

2.5 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Especially these are the constraints the system must work within. Following are the non-functional requirements:

1. The alert sound should start only when the threshold limit is reached.

2.6 Software Requirements

1. Operating System : Windows 10 .
2. Programming Language : Python.
3. Softwares Used :OpenCv , Keras , Pygame.

2.7 Hardware Requirements

- Processor : Intel core i3 and above.
- Hard Disk : 2 GB or above.
- RAM : 2 GB or above.
- Speaker
- Rasberry Pi
- Webcam

CHAPTER -3

3. LITERATURE SURVEY

1. In Jan, 2019, Wisaroot Tipprasert , Theekapun charoenpong, chamaporn sukjamsri , Charoenpong , Chamaporn proposed a complex distributed sensors model that's particularly suitable for measuring blink of the driving force , accident and hand position on a wheel .These sensors are often utilized in automotive active safety systems that aim at detecting driver's fatigue, a serious issue to stop road accidents. The key point of this approach is to style a prototype of sensor units, in order that it can function platform for integrating different sorts of sensors into the wheel . The wheel is slowed or stopped counting on the condition.

.

2. In November, 2019, Anirban Dasgupta , Daleef Rahman , Aurobin da Routray described a smartphone- based drowsiness detection and warning system for automotive drivers. They proposed that measures of the driver's eyes are capable to detect drowsiness under simulator or experiment conditions. Here the problem with this system is that,it is a mobile based application and if the driver doesn't have a mobile or if it is switched off then system will be of no use .

3. In April 2019, Rahul Atul Bhope, proposed a Computer Vision-based system drowsiness system for motorized vehicles with Web Push Notifications to notify the driver before any accident occurs. A real-time video system captures the face of the driver and a pre-trained machine learning model detects the eye boundaries from that real-time video stream. The EAR (Ear Aspect Ratio) is calculated for 20 consecutive frames, which if less than a threshold sounds an alarm and sends an alert on your mobile device through a Web Push Notification. The alert when opened also shows some coffee shops near the driver's location to increase the driver's alertness.

4. In May 2018, Maninder Kahlon ; Subramaniam Ganesan; proposed a driver drowsiness detection algorithm based on the state of eyes of the driver which is

determined by his iris visibility has been implemented. If eyes remain in one state either open or closed longer than expected time as well as if the driver is not looking straight front, it is an indication that driver is drowsy and then the system warns the driver. System is capable of detecting the state of eyes with or without the regular glasses. Matlab with image processing tools has been used to process the image provided by a camera. Matlab creates System Object using Viola_Jones algorithm to detect the objects such as nose, mouth or upper body. The drowsiness detection is done based on the conditions like Black to White pixels ratio, number of pixels in the column greater than the threshold value and eye's shape. Light and position of the driver plays an important role. System can be set to self-learn at startup to setup threshold values.

CHAPTER -4

4. SYSTEM DESIGN

4.1 Introduction to UML

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

User Model View

This view represents the system from the users' perspective. The analysis representation describes a usage scenario from the end- users' perspective.

Structural Model View

In this model, the data and functionality are arrived from inside the system. This model view models the static structures.

Behavioural Model View

It represents the dynamic of behavioural as parts of the system, depicting he interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this view, the structural and behavioural as parts of the system are represented as they are to be built.

Environmental Model View

In this view, the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

4.2 Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running/operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified. In brief, the purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Show the interacting among the requirements are actors.

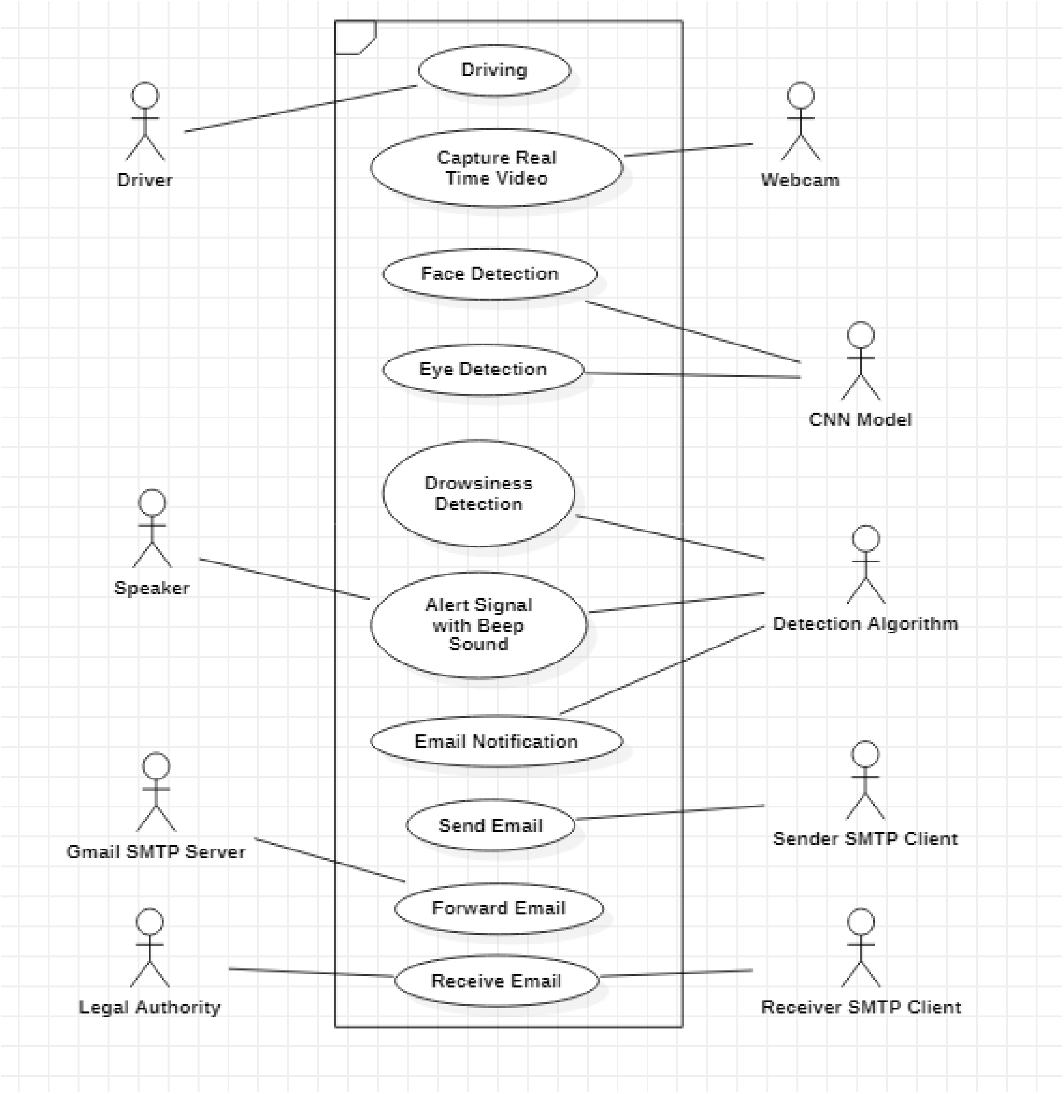


Fig 4.2.1 – Use Case Diagram

4.3 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

The aim of a sequence diagram is to define event sequences, which would have a desired outcome. The focus is more on the order in which messages occur than on the message per se. However, the majority of sequence diagrams will communicate what messages are sent and the order in which they tend to occur.

Basic Sequence Diagram Notations :

Class Roles or Participants

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

Activation or Execution Occurrence

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin grey rectangle placed vertically on its lifeline.

Messages

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages.

Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

Lifelines

Lifelines are vertical dashed lines that indicate the object's presence over time.

Destroying Objects

Objects can be terminated early using an arrow labelled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

Loops

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [].

Guards

When modelling object interactions, there will be times when a condition must be met for a message to be sent to an object. Guards are conditions that need to be used throughout UML diagrams to control flow.

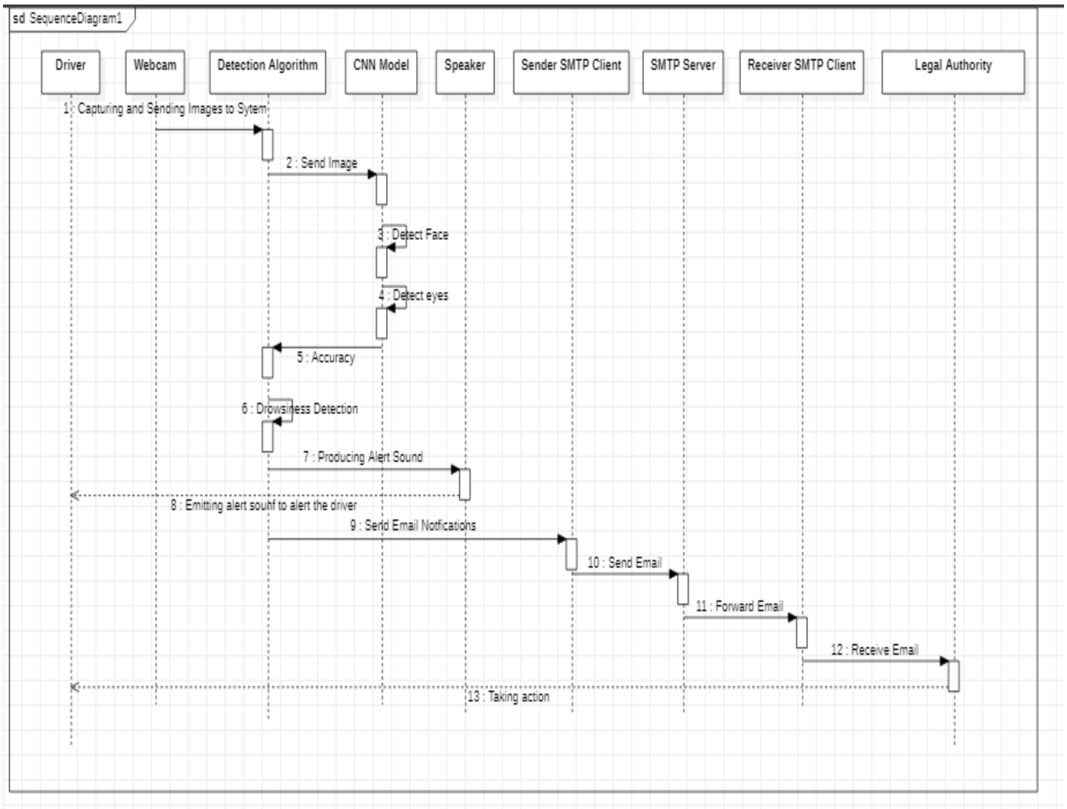


Fig 4.3.1 – Sequence Diagram

4.4 Class Diagram

Class diagrams are the main building blocks of every object oriented methods. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them.

The main purpose to use class diagrams are:

1. This is the only UML which can appropriately depict various aspects of OOP concept.
2. Proper design and analysis of application can be faster and efficient.
3. It is base for deployment and component diagram.
4. Each class is represented by a rectangle having a subdivision of three compartments name,attributes and operation.

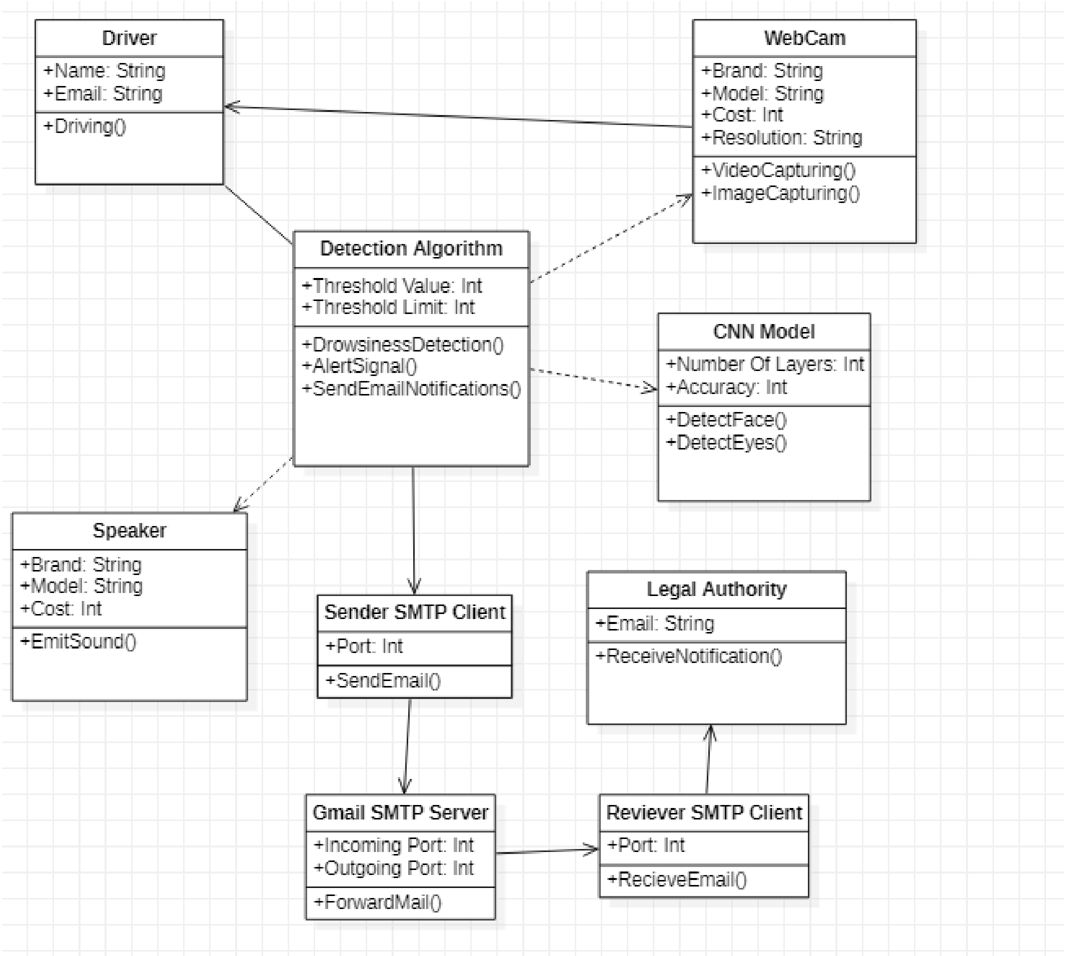


Fig 4.4.1 Class Diagram

CHAPTER -5

5. IMPLEMENTATION

5.1 Code Snippets

- **Model.py:**

```
import os
from keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
from keras.utils.np_utils import to_categorical
import random,shutil
from keras.models import Sequential
from keras.layers import Dropout,Conv2D,Flatten,Dense, MaxPooling2D,
BatchNormalization
from keras.models import load_model



def generator(dir, gen=image.ImageDataGenerator(rescale=1./255),
shuffle=True,batch_size=1,target_size=(24,24),class_mode='categorical' ):

    return
gen.flow_from_directory(dir,batch_size=batch_size,shuffle=shuffle,color_mode='
grayscale',class_mode=class_mode,target_size=target_size)

BS= 32
TS=(24,24)
train_batch= generator('data/train',shuffle=True, batch_size=BS,target_size=TS)
valid_batch= generator('data/valid',shuffle=True, batch_size=BS,target_size=TS)
SPE= len(train_batch.classes)//BS
VS = len(valid_batch.classes)//BS
```

```
print(SPE,VS)



# img,labels= next(train_batch)
# print(img.shape)

model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(24,24,1)),
    MaxPooling2D(pool_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool_size=(1,1)),
#32 convolution filters used each of size 3x3
#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(1,1)),

#64 convolution filters used each of size 3x3
#choose the best features via pooling

#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
#flatten since too many dimensions, we only want a classification output
    Flatten(),
#fully connected to get all relevant data
    Dense(128, activation='relu'),
#one more dropout for convergence' sake :)
    Dropout(0.5),
#output a softmax to squash the matrix into output probabilities
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accura
cy'])
```

```
model.fit_generator(train_batch,
validation_data=valid_batch,epochs=15,steps_per_epoch=SPE
,validation_steps=VS)

model.save('models/cnnCat2.h5', overwrite=True)
```

- **Drowsiness_detection.py:**

```
from cv2 import cv2
import os
from keras.models import load_model
import numpy as np
from pygame import mixer
import time
import smtpmail
from threading import Thread

mixer.init()
sound = mixer.Sound('alarm.wav')

face = cv2.CascadeClassifier(
    'haar cascade files\haarcascade_frontalface_alt.xml')
leye = cv2.CascadeClassifier(
    'haar cascade files\haarcascade_lefteye_2splits.xml')
reye = cv2.CascadeClassifier(
    'haar cascade files\haarcascade_righteye_2splits.xml')



lbl = ['Close', 'Open']
```

```
model = load_model('models/cnncat2.h5')
path = os.getcwd()
cap = cv2.VideoCapture(0)
vid_cod = cv2.VideoWriter_fourcc(*'mp4v')
output = None
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count = 0
score = 0
thicc = 2
rpred = [99]
lpred = [99]
record_flag = False
record_length = 10  # in seconds
video_count = 0
mail_frequency_delay = 300
start_time = time.time() - mail_frequency_delay

def play_sound(sound):
    try:
        sound.play()
    except:  # isplaying = False
        pass
if _name_ == '_main_':
    while True:
        ret, frame = cap.read()

        height, width = frame.shape[:2]
        if record_flag:
            print(video_count)
            output.write(frame)
            if time.time() - start_time > record_length:
                print("MAKE FALSE")
                record_flag = False
                output.release()
                print("before send")
                sendEmail = Thread(target=smtpmail.send, args=(video_count-1,))
                sendEmail.start()
                print("after send")

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        faces = face.detectMultiScale(
            gray, minNeighbors=5, scaleFactor=1.1, minSize=(25, 25))
        left_eye = leye.detectMultiScale(gray)
        right_eye = reye.detectMultiScale(gray)

        cv2.rectangle(frame, (0, height-50), (200, height),
                    (0, 0, 0), thickness=cv2.FILLED)

        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y), (x+w, y+h), (100, 100, 100), 1)

        for (x, y, w, h) in right_eye:
            r_eye = frame[y:y+h, x:x+w]
            count = count+1
            r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY)
            r_eye = cv2.resize(r_eye, (24, 24))
            r_eye = r_eye/255
            r_eye = r_eye.reshape(24, 24, -1)
            r_eye = np.expand_dims(r_eye, axis=0)
            rpred = model.predict_classes(r_eye)
            if(rpred[0] == 1):
                lbl = 'Open'
            if(rpred[0] == 0):
                lbl = 'Closed'
            break
```

```
for (x, y, w, h) in left_eye:
    l_eye = frame[y:y+h, x:x+w]
    count = count+1
    l_eye = cv2.cvtColor(l_eye, cv2.COLOR_BGR2GRAY)
    l_eye = cv2.resize(l_eye, (24, 24))
    l_eye = l_eye/255
    l_eye = l_eye.reshape(24, 24, -1)
    l_eye = np.expand_dims(l_eye, axis=0)
    lpred = model.predict_classes(l_eye)
    if(lpred[0] == 1):
        lbl = 'Open'
    if(lpred[0] == 0):
        lbl = 'Closed'
    break

if(rpred[0] == 0 and lpred[0] == 0):
    score = score+1
    cv2.putText(frame, "Closed", (10, height-20), font,
                1, (255, 255, 255), 1, cv2.LINE_AA)
# if(rpred[0]==1 or lpred[0]==1):
else:
    score = score-1
    cv2.putText(frame, "Open", (10, height-20), font,
                1, (255, 255, 255), 1, cv2.LINE_AA)

if(score < 0):
    score = 0
cv2.putText(frame, 'Score:'+str(score), (100, height-20),
            font, 1, (255, 255, 255), 1, cv2.LINE_AA)
if(score > 15):
    # person is feeling sleepy so we beep the alarm
    cv2.imwrite(os.path.join(path, 'image.jpg'), frame)
```

```
            # play_sound(sound)
            if not record_flag and time.time() - start_time > mail_frequency_delay:
                print("FALSE")
                record_flag = True
                output = cv2.VideoWriter("cam_video"+str(video_count)+".mp4",
vid_cod, 10.0, (640,480))
                video_count+=1
                start_time = time.time()

            if(thicc < 16):
                thicc = thicc+2
            else:
                thicc = thicc-2
                if(thicc < 2):
                    thicc = 2
            cv2.rectangle(frame, (0, 0), (width, height), (0, 0, 255), thicc)
        cv2.imshow('frame', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
    if record_flag:
        os.remove("cam_video"+str(video_count-1)+".mp4")
```

- **smtpmail.py**

```
import smtplib
from os.path import basename
from email.mime.application import MIMEApplication
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.utils import COMMASPACE, formatdate
```

```
import os

email_user = ""
email_password = " "
from_email = ""
to_emails = [""]
body = "Please see the attached video for more information about violation"
subject = "Alert for possible driver sleeping"
email_server = 'smtp.gmail.com'
email_server_port = 465

def send_mail(from_email, to_emails, subject, body, file, email_server):
    assert isinstance(to_emails, list)

    msg = MIMEMultipart()
    msg['From'] = from_email
    msg['To'] = COMMASPACE.join(to_emails)
    msg['Date'] = formatdate(localtime=True)
    msg['Subject'] = subject

    msg.attach(MIMEText(body))

    with open(file, "rb") as fil:
        part = MIMEApplication(fil.read(), Name=basename(file))
    # After the file is closed
    part['Content-Disposition'] = 'attachment; filename="%s"' % basename(file)
    msg.attach(part)

    email_server.sendmail(from_email, to_emails, msg.as_string())
    print('Email sent!')


def   email_server_login(email_user,   email_password,   server="smtp.gmail.com",
port=465):
    server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
    server.ehlo()
    server.login(email_user, email_password)
    return server

def send(count):
    file_name = "cam_video"+str(count)+".mp4"
    email_server_connection = email_server_login(email_user, email_password,
email_server, email_server_port)
    send_mail(from_email, to_emails, subject,body, file_name,
email_server_connection)
    email_server_connection.close()
    os.remove(file_name)
```

CHAPTER -6

6. TESTING

6.1 Introduction to Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

Who does Testing?

It depends on the process and the associated stakeholders of the project(s). In the IT industry, large companies have a team with responsibilities to evaluate the developed software in context of the given requirements. Moreover, developers also conduct testing which is called Unit Testing. In most cases, the following professionals are involved in testing a system within their respective capacities:

- Software Tester
- Software Developer
- Project Lead/Manager
- End User

Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

- Functional Testing
- Non-functional Testing

Functional Testing

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

6.2 Software Testing Life Cycle

The process of testing a software in a well planned and systematic way is known as software testing lifecycle (STLC).

Different organizations have different phases in STLC however generic Software Test Life Cycle (STLC) for waterfall development model consists of the following phases.

- Requirements Analysis
- Test Planning
- Test Analysis
- Test Design

1. Requirements Analysis

In this phase testers analyze the customer requirements and work with developers during the design phase to see which requirements are testable and how they are going to test those requirements.

It is very important to start testing activities from the requirements phase itself because the cost of fixing defect is very less if it is found in requirements phase rather than in future phases.

2. Test Planning

In this phase all the planning about testing is done like what needs to be tested, how the testing will be done, test strategy to be followed, what will be the test environment, what test methodologies will be followed, hardware and software availability, resources, risks etc. A high level test plan document is created which includes all the planning inputs mentioned above and circulated to the stakeholders.

3. Test Analysis

After test planning phase is over test analysis phase starts, in this phase we need to dig deeper into project and figure out what testing needs to be carried out in each SDLC phase. Automation activities are also decided in this phase, ifautomation needs to be done for software product, how will the automation be

done, how much time will it take to automate and which features need to be automated. Non functional testing areas(Stress and performance testing) are also analyzed and defined in this phase.

4. Test Design

In this phase various black-box and white-box test design techniques are used to design the test cases for testing, testers start writing test cases by following those design techniques, if automation testing needs to be done then automation scripts also needs to written in this phase.

6.3 Test Cases

- The system detects the normal state of the driver.
- The system detects the drowsy state of the driver.

CHAPTER -7

7. SCREENSHOTS

7.1 Normal State

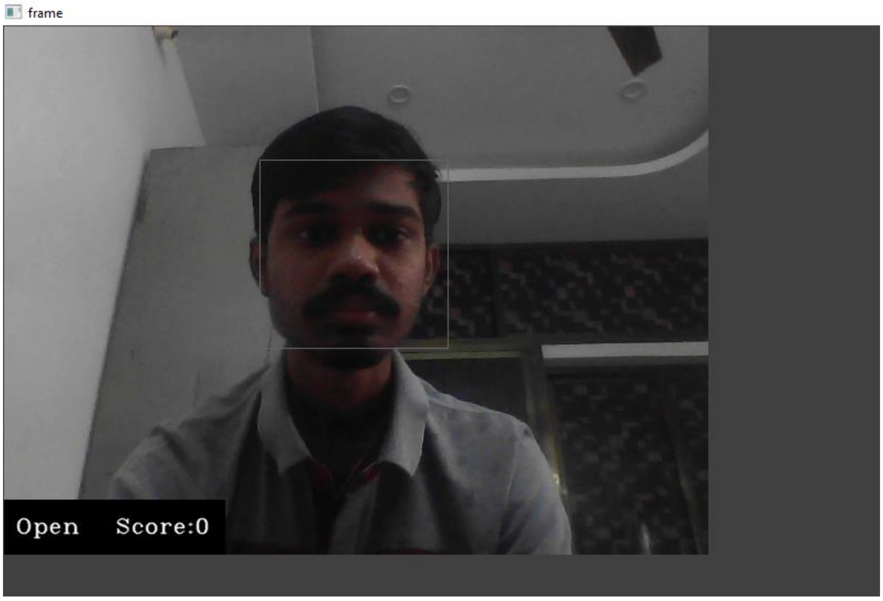


Fig 7.1.1 Normal State

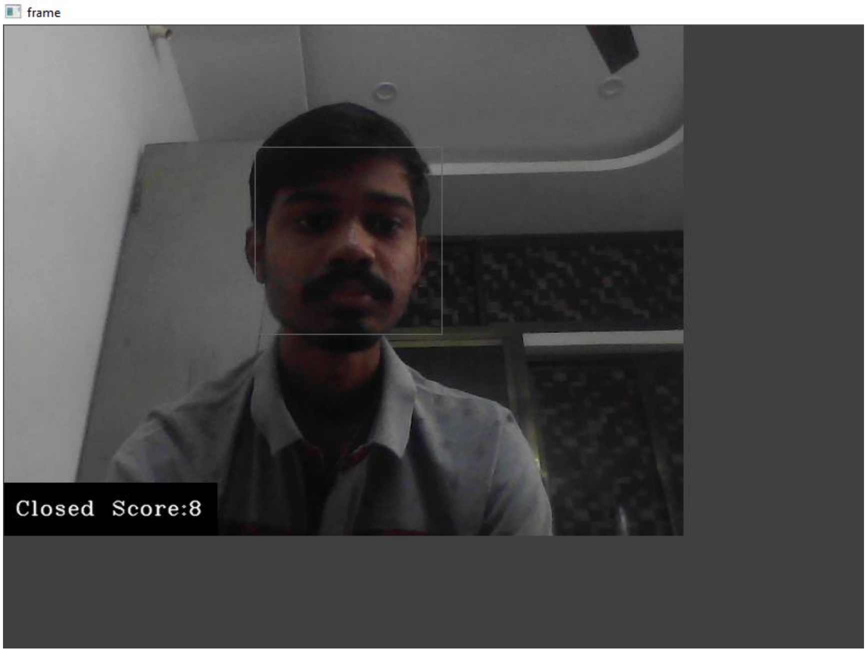


Fig 7.1.2 Normal State (Cont.)

7.2 Drowsy State

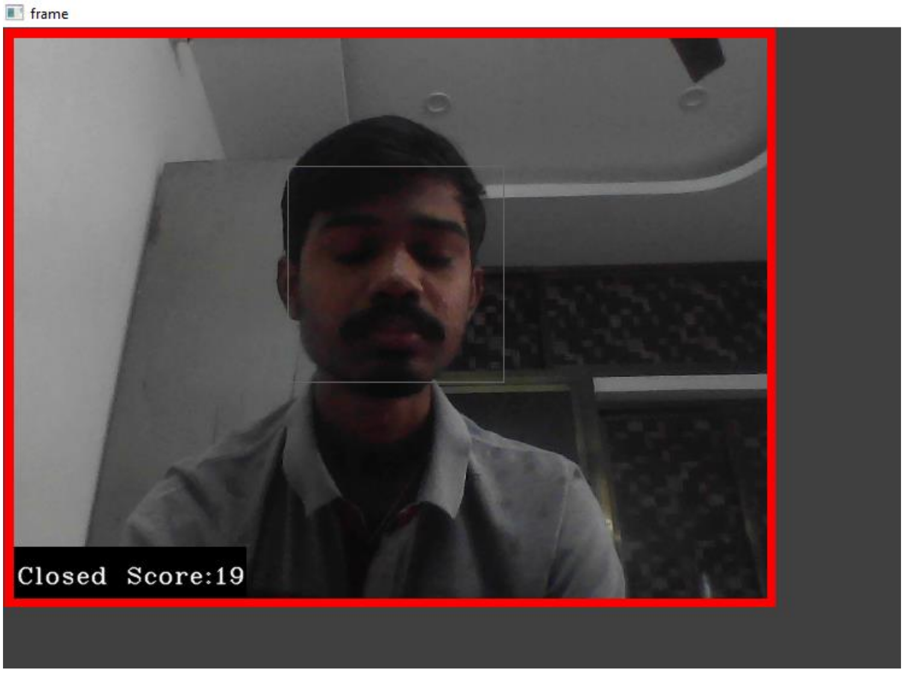
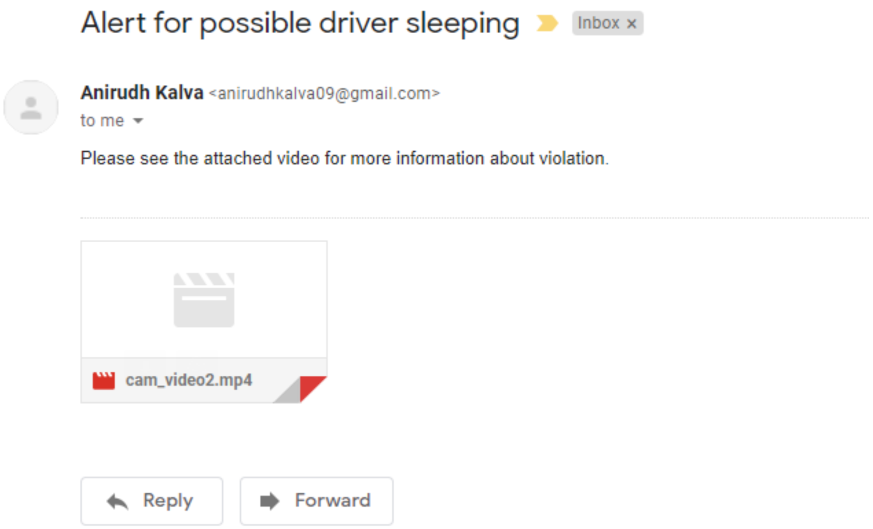


Fig 7.2.1 Drowsy State

7.3 Email Notification


Alert for possible driver sleeping

Anirudh Kalva <anirudhkalva09@gmail.com>

to me

Please see the attached video for more information about violation.

cam_video2.mp4

Fig 7.3.1 Email Notification

CHAPTER -8

8. FUTURE ENHANCEMENTS

The system at this stage is a “Proof of Concept” for a much substantial endeavor. This will serve as a first step towards a distinguished technology that can bring about an evolution aimed at ace development. The developed system has special emphasis on real-time monitoring with flexibility, adaptability and enhancements as the foremost requirements.

Future enhancements are always meant to be items that require more planning, budget and staffing to have them implemented. There following are couple of recommended areas for future enhancements:

- **Standalone product:** It can be implemented as a standalone product, which can be installed in an automobile for monitoring the automobile driver.
- **Smart phone application:** It can be implemented as a smart phone application, which can be installed on smart phones. And the automobile driver can start the application after placing it at a position where the camera is focused on the driver.

CHAPTER -9

9. CONCLUSION

Previous studies have proposed a number of methods to detect drowsiness. After doing literature survey, different techniques has been found for detecting driver drowsiness and they use different types of data as input for their algorithm. After the survey of different types of methods, it is found that using camera is the best method which can be easily applied and appropriate in all conditions. We decide to explore this method of computer vision and proposed a noble method to detect driver drowsiness based on detecting eyelid closing and opening using convolutional neural networks. This is useful in detecting drowsiness of the driver and intimating the drivers about it using deep learning technique.

CHAPTER- 10

10. REFERENCES

1. Wisaroot Tipprasert ; Theekapun charoenpong ; chamaporn sukjamsri ; Charoenpong ; Chamaporn "**A method of driver eyes closure and yawning detection for drowsiness analysis using infrared camera".**

2. "**A Smartphone- Based Drowsiness Detection and Warning System for Automotive Drivers**", Anirban Dasgupta ; Daleef Rahman ; Aurobin da Routray IEEE Transactions on Intelligent Transportation Systems, Year: 2019 | Volume: 20, Issue: 11 | Journal Article.

3. "**Computer Vision based Drowsiness Detection for motorized vehicles with Web Push Notifications**", Rahul Atul Bhope, 2019 4th a International Conference on Internet of Things: Smart Innovation and Usages (IOT-SIU).

4. "**Driver Drowsiness Detection System Based on Binary Eyes Image Data**" Maninder Kahlon ; Subramaniam Ganesan, 2018 IEEE International Conference on Electro/Information Technology (EIT).