A Group Project Report

on

DROWSINESS DETECTION SYSTEM

Submitted

In partial fulfillment of the academic requirements for the award of the degree of

BACHLEOR OF TECHNOLOGY

in

Electronics and Communication Engineering [**ECE**]

By

A.YASHWANTH SAI [**RollNo.20311A04L1**]

C.SAI SUKRUTH [**Roll No.20311A04L3**]

UNDER THE GUIDANCE OF

Mrs. N PRATHYUSHA

Associate Professor, Dept. of ECE

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

SREENIDHI INSTITUTE OF SCIENCE & TECHNOLOGY

Yamnampet, Ghatkesar, R.R. District, Hyderabad – 501301

2023

CERTIFICATE

Submitted by

A. YASHWANTH SAI [**RollNo.20311A04L1**]

C. SAI SUKRUTH [**Roll No.20311A04L3**]

in fulfillment for the award of Bachelor **of Technology** in **Electronics and Communication Engineering[ECE] Sreenidhi Institute of Science and Technology,** an autonomous Institute under Jawaharlal Nehru Technological University, is a record of bona fide work carried out by them under our guidance and supervision.

The results embodied in the report have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide	**Group Project Coordinator**	**Head of Department**
Mrs N Prathyusha	Mrs.E Lavanya	Dr. S.P.V Subba Rao
Assistant Professor Department of ECE	Assistant Professor Department of ECE	Professor & HOD Department of ECE

Date

External Examiner

DECLARATION

We hereby declare that the work described in this report, entitled " **DROWSINESS DETECTION SYSTEM** " which is being submitted by me in partial fulfilment for the award of Bachelor of Technology in Electronics and Communication Engineering (ECE), Sreenidhi Institute Of Science & Technology affiliated to Jawaharlal Nehru Technological University Hyderabad, Kukatpally, Hyderabad (Telangana) -500 085 is the result of investigations carried out by us under the Guidance of E. Lavanya, Assistant Professor, ECE Department, Sreenidhi Institute Of Science And Technology, Hyderabad. The work is original and has not been submitted for any Degree/Diploma of this or any other university.

Place : Hyderabad

Signature :

Date :

A.YASHWANTH SAI[RollNo20311A04L1]

C.SAI SUKRUTH [RollNo.20311A04L3]

ABSTRACT

Life is a precious gift but it is full of risk. Therefore, there is a need to take safety precautions in order to avoid accidents. Nowa-days, road accidents have become one of the major cause of insecure life. It is very important to take proper care while driving. Carelessness for a single minute can cause major problems. Most of the road accidents occur due to carelessness and inactiveness of the driver while driving. This is the reason, every year the number of road accidents is increasing especially by cars. Due to drowsiness, drivers become inactive while driving.

Drowsiness detection could have saved number of lives if it could have been detected earlier. Development of drowsiness detection is due to the use/help of machine vision-based concepts. In order to detect fatigue or drowsiness, small camera has been used which points directly towards the driver's face and detects the eye ball movement of the driver. At the very first stage, system detects the face and then detects the eyes and then determines whether the eye detected is open or closed. Changes between the intensity in the eye takes place due to which, it narrows down area of the eye and further gives information to the system. Within a time limit, a system gives information that the driver is falling asleep and there is a need to alert him/her.

Contents

List of Figures

Introduction:

1.1 PURPOSE

1.1.1 HUMAN PSYCHOLOGY WITH CURRENT TECHNOLOGY

Humans have always invented machines and devised techniques to ease and protect their lives, for mundane activities like traveling to work, or for more interesting purposes like aircraft travel. With the advancement in technology, modes of transportation kept on advancing and our dependency on it started increasing exponentially. It has greatly affected our lives as we know it. Now, we can travel to places at a pace that even our grandparents wouldn't have thought possible. In modern times, almosteveryone in this world uses some sort of transportation every day. Some people are rich enough to have their own vehicles while others use public transportation. However, there are some rules and codes of conduct for those who drive irrespective of their social status. One of them is staying alert and active while driving.

Neglecting our duties towards safer travel has enabled hundreds of thousands of tragedies to get associated with this wonderful invention every year. It may seem like a trivial thing to most folks but following rules and regulations on the road is of utmost importance. While on road, an automobile wields the most power and in irresponsible hands, it can be destructive and sometimes, that carelessness can harm lives even of the people on the road. One kind of carelessness is not admitting when weare too tired to drive. In order to monitor and prevent a destructive outcome from such negligence, many researchers have written research papers on driver drowsiness detection systems. But at times, some of the points and observations made by the system are not accurate enough. Hence, to provide data and another perspective on the problem at hand,in order to improve their implementations and to further optimize thesolution, this project has been done.

1.1.2 FACTS & STATISTICS

Our current statistics reveal that just in 2015 in India alone, 148,707people died due to car related accidents. Of these, at least 21 percentwere caused due to fatigue causing drivers to make mistakes. This can bea relatively smaller number still, as among the multiple causes that canlead to an accident, the involvement of fatigue as a cause is generally grossly underestimated. Fatigue combined with bad infrastructure in developing countries like India is a recipe for disaster. Fatigue, in general,is very difficult to measure or observe unlike alcohol and drugs, whichhave clear key indicators and tests that are available easily. Probably, thebest solutions to this problem are awareness about fatigue-relatedaccidents and promoting drivers to admit fatigue when needed. Theformer is hard and much more expensive to achieve, and the latter is notpossible without the former as driving for long hours is very lucrative.When there is an increased need for a job, the wages associated with itincreases leading to more and more people adopting it. Such is the casefor driving transport vehicles at night. Money motivates drivers to makeunwise decisions like driving all night even with fatigue. This is mainlybecause the drivers are not themselves aware of the huge risk associated with driving when fatigued. Some countries have imposed restrictions onthe number of hours a driver can drive at a stretch, but it is still not enough to solve this problem as its implementation is very difficult andcostly.

1.2 DOCUMENT CONVENTIONS

Main Heading Font size: 24 (bold fonts)Sub-

headings Font size: 16 (bold fonts)

Sub-headings Content Font size: 14 (normal fonts)

1.3 INTENDED AUDIENCE

The intended audience for this document are the development team, the project evaluation jury, and other tech-savvy enthusiasts who wish tofurther work on the project.

1.4 PRODUCT SCOPE

There are many products out there that provide the measure of fatigue level in the drivers which are implemented in many vehicles. The driver drowsiness detection system provides the similar functionality but with better results and additional benefits. Also, it alerts the user on reaching a certain saturation point of the drowsiness measure.

1.5 PROBLEM DEFINITION

Fatigue is a safety problem that has not yet been deeply tackled by any country in the world mainly because of its nature. Fatigue, in general, isvery difficult to measure or observe unlike alcohol and drugs, which haveclear key indicators and tests that are available easily. Probably, the bestsolutions to this problem are awareness about fatigue-related accidents and promoting drivers to admit fatigue when needed. The former is hardand much more expensive to achieve, and the latter is not possiblewithout the former as driving for long hours is very lucrative.

Literature Survey

2.1 SYSTEM REVIEW

This survey is done to comprehend the need and prerequisite of the general population, and to do as such, we went through different sites andapplications and looked for the fundamental data. Based on these data, we made an audit that helped us get new thoughts and make different arrangements for our task. We reached the decision that there is a needof such application and felt that there is a decent extent of progress in thisfield too.

2.2 TECHNOLOGY USED

a. PYTHON - Python is an interpreted, high-level, general-purposeprogramming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed AND supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

b. JUPYTER Lab - Project Jupyter is a nonprofit organization created to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

c. IMAGE PROCESSING - In computer science, digital image processing isthe use of computer algorithms to perform image processing on digital images.

d. MACHINE LEARNING - Machine learning is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data",in order to make predictions or decisions without being explicitly told.

Software Requirements Specification

3.1 Python:

- Python 3

3.2 Libraries

- Numpy
- Scipy
- Playsound
- Dlib
- Imutils
- opencv, etc.

3.3 Operating System

- Windows or Ubuntu

Hardware Requirements Specification

I. Laptop with basic hardware.
II. Webcam

Requirement Analysis

4.1 Python: Python is the basis of the program that we wrote. It utilizes many of the python libraries.

4.2 Libraries:

- Numpy: Pre-requisite for Dlib
- Scipy: Used for calculating Euclidean distance between the eyelids.
- Dlib: This program is used to find the frontal human face andestimate its pose using 68 face landmarks.
- Imutils: Convenient functions written for Opencv.
- Opencv: Used to get the video stream from the webcam, etc.

4.3 OS: Program is tested on Windows 10 build 1903 and PopOS19.04

4.3 Laptop: Used to run our code.

4.4 Webcam: Used to get the video feed.

System Design

5.1 USE CASE DIAGRAM

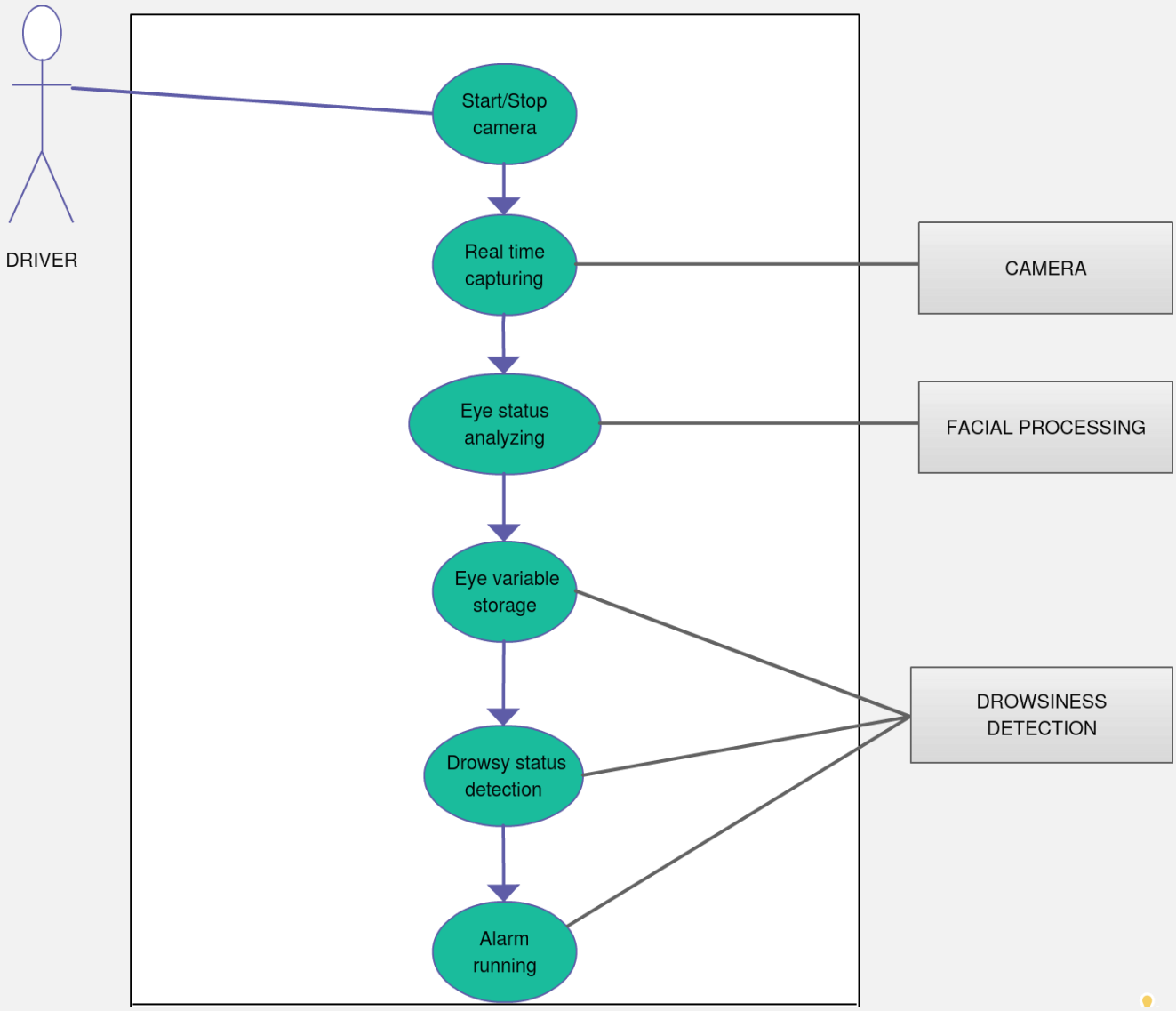


FIGURE 1

5.2 ACTIVITY DIAGRAM

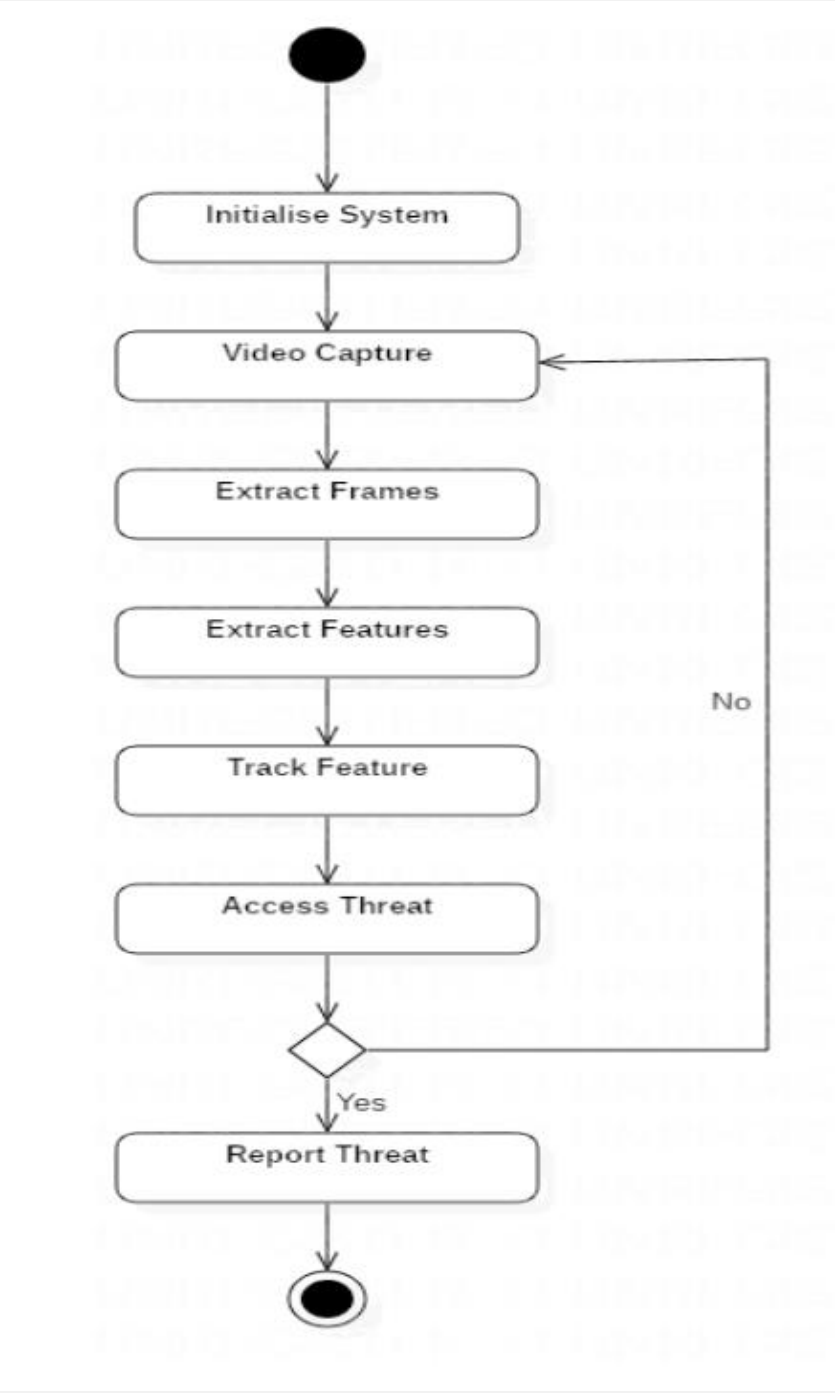


FIGURE 2

5.3 CLASS DIAGRAM

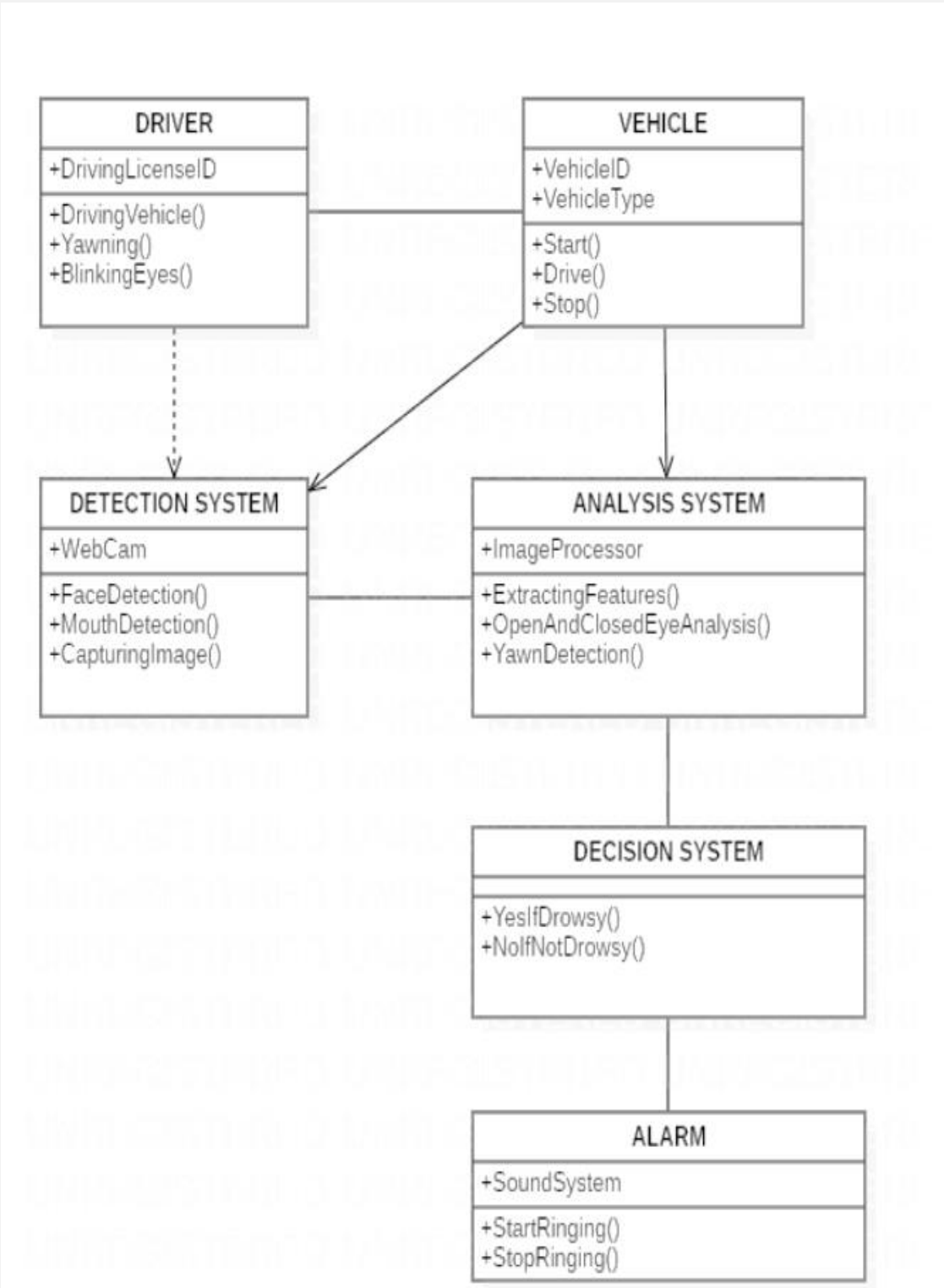


FIGURE 3

CODE:

```
#python drowniness_yawn.py --webcam webcam_index

from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import os

def alarm(msg):
global alarm_status
global alarm_status2
global saying

while alarm_status:
print('call')
s = 'espeak "'+msg+'"'
os.system(s)

if alarm_status2:
print('call')
saying = True
s = 'espeak "' + msg + '"'
os.system(s)
saying = False

def eye_aspect_ratio(eye):
A = dist.euclidean(eye[1], eye[5])
B = dist.euclidean(eye[2], eye[4])

C = dist.euclidean(eye[0], eye[3])

ear = (A + B) / (2.0 * C)

return ear

def final_ear(shape):
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]

leftEAR = eye_aspect_ratio(leftEye)
rightEAR = eye_aspect_ratio(rightEye)
```

```
ear = (leftEAR + rightEAR) / 2.0
return (ear, leftEye, rightEye)

def lip_distance(shape):
top_lip = shape[50:53]
top_lip = np.concatenate((top_lip, shape[61:64]))

low_lip = shape[56:59]
low_lip = np.concatenate((low_lip, shape[65:68]))

top_mean = np.mean(top_lip, axis=0)
low_mean = np.mean(low_lip, axis=0)

distance = abs(top_mean[1] - low_mean[1])
return distance


ap = argparse.ArgumentParser()
ap.add_argument("-w", "--webcam", type=int, default=0,
help="index of webcam on system")
args = vars(ap.parse_args())

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 20
alarm_status = False
alarm_status2 = False
saying = False
COUNTER = 0

print("-> Loading the predictor and detector...")
#detector = dlib.get_frontal_face_detector()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")    #Faster but less accurate
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')


print("-> Starting Video Stream")
vs = VideoStream(src=args["webcam"]).start()
#vs= VideoStream(usePiCamera=True).start()       //For Raspberry Pi
time.sleep(1.0)

while True:

frame = vs.read()
frame = imutils.resize(frame, width=450)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

#rects = detector(gray, 0)
rects = detector.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30),
flags=cv2.CASCADE_SCALE_IMAGE)

#for rect in rects:
for (x, y, w, h) in rects:
rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))
```

```
shape = predictor(gray, rect)
shape = face_utils.shape_to_np(shape)

eye = final_ear(shape)
ear = eye[0]
leftEye = eye [1]
rightEye = eye[2]

distance = lip_distance(shape)

leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

lip = shape[48:60]
cv2.drawContours(frame, [lip], -1, (0, 255, 0), 1)

if ear < EYE_AR_THRESH:
COUNTER += 1

if COUNTER >= EYE_AR_CONSEC_FRAMES:
if alarm_status == False:
alarm_status = True
t = Thread(target=alarm, args=('wake up sir',))
t.deamon = True
t.start()

cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

else:
COUNTER = 0
alarm_status = False

if (distance > YAWN_THRESH):
cv2.putText(frame, "Yawn Alert", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
if alarm_status2 == False and saying == False:
alarm_status2 = True
t = Thread(target=alarm, args=('take some fresh air sir',))
t.deamon = True
t.start()
else:
alarm_status2 = False

cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
cv2.putText(frame, "YAWN: {:.2f}".format(distance), (300, 60),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
```

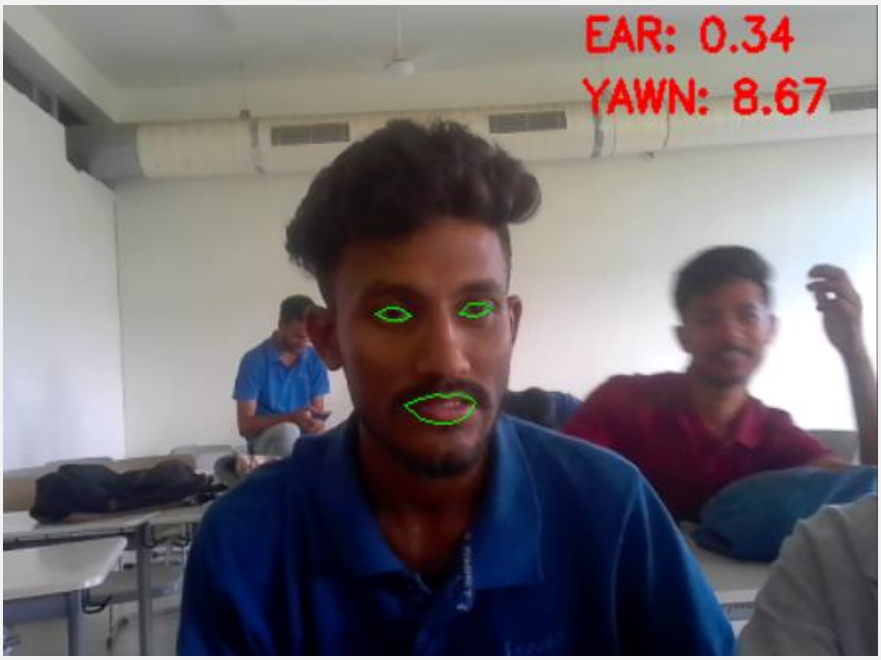
```
if key == ord("q"):
break

cv2.destroyAllWindows()
vs.stop()
```

System Testing

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	NSGY	Straight Face, Good Light, WithGlasses	Non Drowsy	Non Drowsy
T02	YTGN	Tilted Face, Good Light, No Glasses	Drowsy	Drowsy
T03	YTGY	Tilted Face, Good Light, With Glasses	Drowsy	Drowsy

6.1 Test Cases and Test Results

Note: Testing is performed manually



Project Planning

7.1 SYSTEM MODEL

The framework is created utilizing the incremental model. The center model of the framework is first created and afterwards augmented in thisway in the wake of testing at each turn. The underlying undertaking skeleton was refined into expanding levels of ability.

At the following incremental level, it might incorporate new execution backing and improvement.

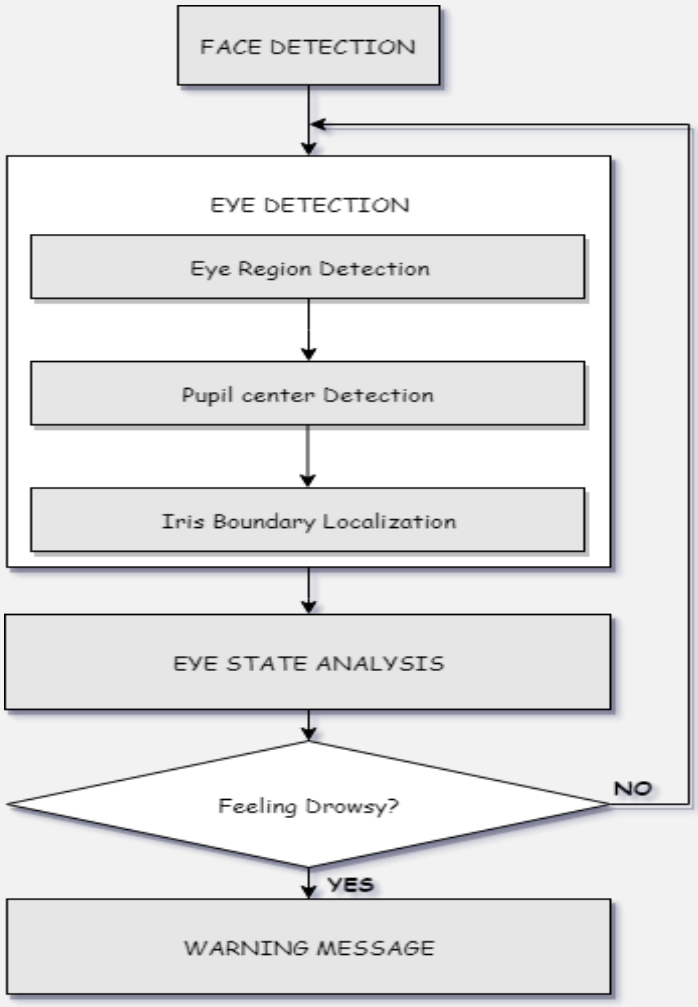
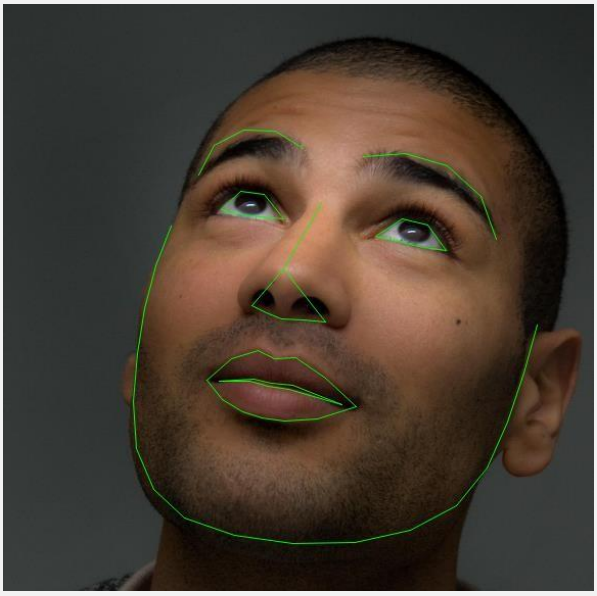


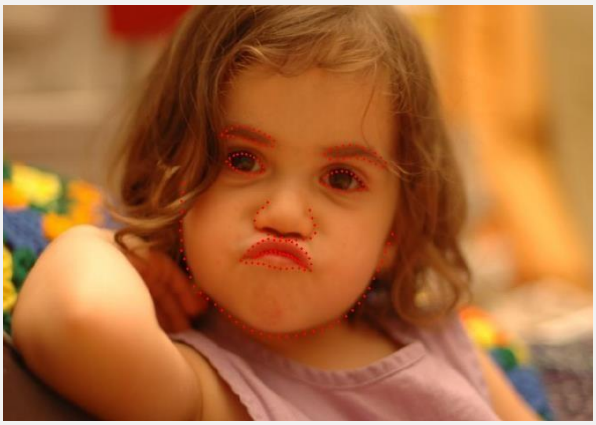
Figure 4: Block diagram

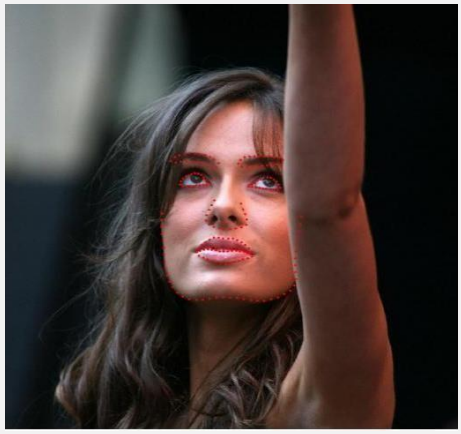
Implementation

- In our program we used Dlib, a pre-trained program trained on theHELEN dataset to detect human faces using the pre-defined 68 landmarks.



Landmarked Image of a person by Dlib



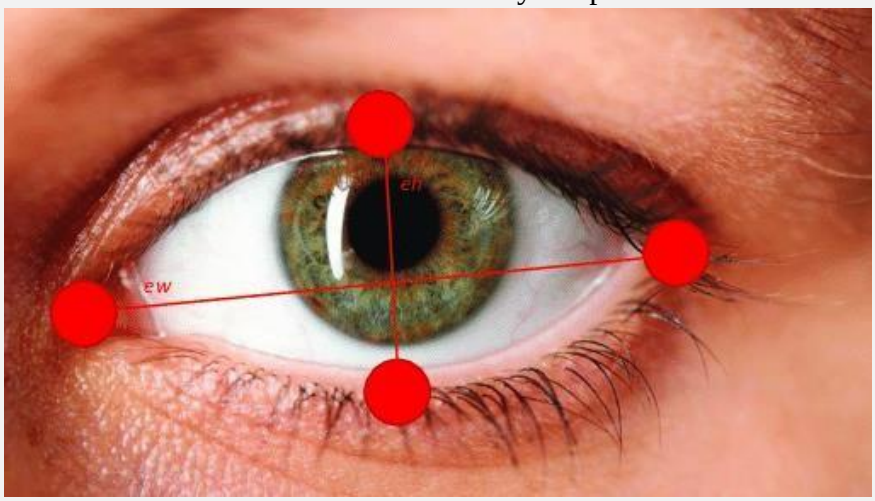


HELEN Dataset Samples

- After passing our video feed to the dlib frame by frame, we are able todetect left eye and right eye features of the face.

- Now, we drew contours around it using OpenCV.
- Using Scipy's Euclidean function, we calculated sum of both eyes' aspect ratio which is the sum of 2 distinct vertical distances betweenthe eyelids divided by its horizontal distance.

Eyes with horizontal and vertical distance marked for Eye Aspect

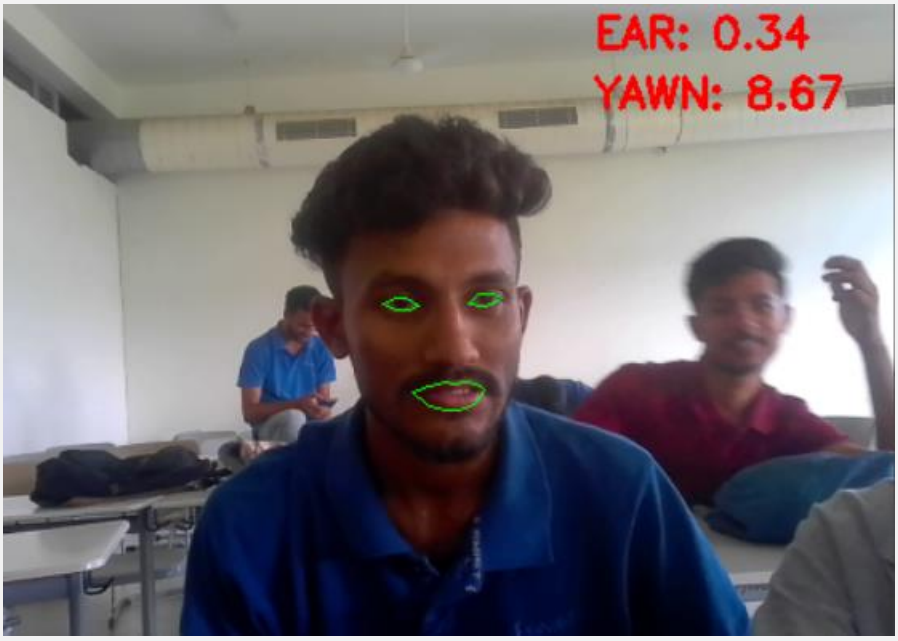


Ratiocalculation.

- Now we check if the aspect ratio value is less than 0.25 (0.25 waschosen as a base case after some tests). If it is less an alarm is sounded and user is warned.

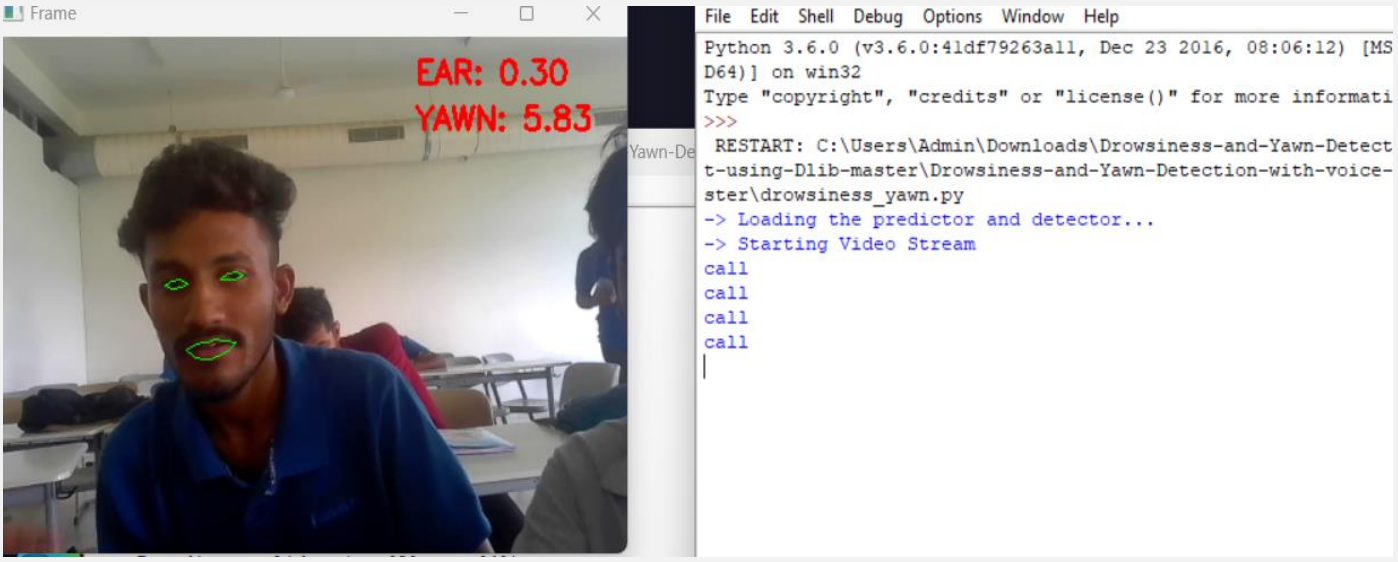
Screenshots of Project

9.1 Non-drowsy Person:



Non-drowsy state

9.2 Drowsy Person:



Conclusion and Future Scope

10.1 Conclusion

It completely meets the objectives and requirements of the system. The framework has achieved an unfaltering state where all the bugs have been disposed of. The framework cognizant clients who are familiar withthe framework and comprehend it's focal points and the fact that it takescare of the issue of stressing out for individuals having fatigue-related issues to inform them about the drowsiness level while driving.

10.2 Future Scope

The model can be improved incrementally by using other parameters like blink rate, yawning, state of the car, etc. If all these parameters are used itcan improve the accuracy by a lot.

We plan to further work on the project by adding a sensor to track theheart rate in order to prevent accidents caused due to sudden heart attacks to drivers.

Same model and techniques can be used for various other uses like Netflix and other streaming services can detect when the user is asleepand stop the video accordingly. It can also be used in application that prevents user from sleeping.

References

[1] COMPUTATIONALLY EFFICIENT FACE DETECTION; B. SCHLKOPF-A. BLAKE, S. ROMDHANI, AND P. TORR.

[2] USE OF THE HOUGH TRANSFORMATION TO DETECT LINES ANDCURVES IN PICTURE; R. DUDA AND P. E. HART.

[3] JAIN, "FACE DETECTION IN COLOR IMAGES; R. L. HSU, M. ABDEL-MOTTALEB, AND A. K. JAIN.

[4] OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION; P.R. TABRIZI AND R. A. ZOROOFI.

[5] http://ncrb.gov.in/StatPublications/ADSI/ADSI2015/chapter1A%20traffic%20accidents.pdf

[6] http://www.jotr.in/text.asp?2013/6/1/1/118718

[7] http://dlib.net/face_landmark_detection_ex.cpp.html