A Report on

DRIVER DOWSINESS DETECTION SYSTEM

Submitted for partial fulfillment of award of

BACHELOR OF TECHNOLOGY

degree

In

Computer Science/Computer Science -AIML

By

AMAN YADAV (20003215300)
AYUSH SINGH (2000321530042)
ISHAAN CHAUDHARY (20003215300)

ROHIT VAISTH

SUPERVISOR

ABES ENGINEERING COLLEGE, GHAZIABAD

Dr. A P J ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

MARCH 2022

Certificate

Certified that **ISHAAN CHOUDHARY** has carried out the project work under the mentorship of **MR ROHIT VAISHTH** presented in this report entitled **“DRIVER DOWSINESS DETECTION SYSTEM”** for the award of **Bachelor of Technology** from ABES Engineering College, Ghaziabad, under my supervision. The report embodies result of original work and studies carried out by Student himself/herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.

MR ROHIT VAISHTITH
Designation:

Address:

Date:

Acknowledgement

We take this opportunity to thank our teachers and friends who helped us throughout the project.

First and foremost, I would like to thank my guide for the project **MR ROHIT VAISHTITH (Computer Science Department**) for her/his valuable advice and time during development of project.

We would also like to thank **Dr. Pankaj Kumar Sharma (HOD, Computer Science Department**) for his constant support during the development of the project.

Name 1 : AMAN YADAV

Roll No. 2000321530017

Signature

Name 2: AYUSH SINGH

Roll No. 2000321530042

Signature

Name 3: ISHAAN CHOUDHARY

Roll No. 20003215300

Signature

Declaration

We hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name: AMAN YADAV, AYUSH SINGH, ISHAAN CHOUDHARY

Roll No.: 2000321530017,2000321530042, 2000321530055

Date: 10/03/2022

Abstract

When it comes to safety while driving many factors come in our mind like over speeding, rash driving, alcohol, violation of traffic signs. But we often ignore a cause of mentioned factors which is Drowsiness.

Many a times the cause is drowsiness of driver that is feeling of being sleepy. We often sideline the fact that our body needs rest which sometimes can be lethal. Sleep deprived drivers are a part of nearly 40% accidents. This is an alarming number. This is not only increasing yearly but also have been ignored by majority. So, we decide to change this.

Our system will be enhanced to detect the drowsiness of driver. The system designed is a non-intrusive real-time monitoring system. The priority is on improving the safety of the driver without being obtrusive. In this project the eye blink of the driver is detected. If the driver eyes remain closed for more than a certain period of time, the driver is said to be drowsy and an alarm is sounded. The programming for this is done in OpenCV using the DLIB library for the detection of facial features.

KEYWORD: Drowsiness, Driver monitoring, Python, OpenCV, D-lib

CONTENTS

CHAPTER NO. **TITLE** **PAGE NO.**

ABSTRACT **iii**

1 INTRODUCTION **Page No**

1.1Problem Statement

1.2Objective Of Project

1.3Project Overview

1.3.1 About project

2 Literature Review

3 Modules used

3.1OpenCV

3.2 D-lib

4 Flowchart

5 Coding

6 Output

7 Conclusion

8 Reference

9 Appendices

INTRODUCTION

PROBLEM STATEMENT:

The enhancement in the technology for detecting the Drowsiness is a great challenge in accident-avoidance systems. We generally focused on the factors like over speeding, rash driving, alcohol, violation of traffic signs. But we often ignore
the drowsiness of driver.

OBJECTIVE :

The aim of our project is to develop a driver drowsiness system. Our focus will be placed on designing a system that will accurately monitor the open or closed state of the driver's eyes in real time.

By monitoring the eyes it is believe that the early stages of drowsiness can be detected to prevent any kind of accident.

PROJECT OVERVIEW

- TAKING INPUT FROM CAMERA

  We will first take input from camera (Dash Cam) which we will use as an input data. We will use the dash cam at steering wheel. It will be a compact camera.

- LIVE IMAGE CAPTURING

  Captured image will treated as data collected, which is being used here. Then the image will be taken in program as an Input which will encounter the OpenCV

- FACE DETECTION

  By using image captured by camera we detect the face from it.
  We use OpenCV and D-lib for face recognition.
  In OpenCV we Haar-cascade file for frontal face and eye.
  The Haar-cascade file are in .xml format. Means they are pre-trained module for detecting face and eyes.

- Analysis of Eyes Movements

  This process involves the facial processing of person that Involves

  - Eye detection

    This is done by using the facial landmarks we have Using D-lib.

  - Recognition of eye state

In this step we will recognize the current state of eyes using the landmarks of eyes which we will take to next step .

- Eye Aspect Ratio Calculation

The average eye aspect ratio is 0.339 and 0.141 when the eyes are opened and closed respectively.

- Checking the condition

The parameter which are set by us will check if they are in Given conditions which are the aspect ratio of eye
The condition is if the aspect ratio off eye is found near to 0.141 continuous for a given time frame

- ALERTING THROUGH BUZZER

If the desired condition found to be violated then the buzzer will be starts ringing.

LITERATURE REVIEW

Review 1:

In June, 2014, Enosis's et .al. described 'Driver Drowsiness Detection through \HMM based Dynamic Modeling'. They proposed a new method of analyzing the facial expression of the driver through Hidden Markov Model (HMM) based dynamic modeling to detect drowsiness. They have implemented the algorithm using a simulated driving setup. Experimental results verified the effectiveness of the proposed method.

Review 2:

In 2008, Hong So et. AL described 'A Partial Least Squares Regression-Based Fusion Model for Predicting the Trend in Drowsiness'. They proposed a new technique of modeling driver drowsiness with multiple eyelid movement features based on an information fusion technique—partial least squares regression (PLSR), with which to cope with the problem of strong collinear relations among eyelid movement features and, thus, predicting the tendency of the drowsiness. The predictive precision and robustness of the model thus established are validated, which show that it provides a novel way of fusing multi- features together for enhancing our capability of detecting and predicting the state of drowsiness.

Review 3 :

In June, 2012, A. Cheng et. al. described 'Driver Drowsiness Recognition Based on Computer Vision Technology’. They presented a nonintrusive drowsiness recognition method using eye-tracking and image processing. A robust eye detection algorithm is introduced to address the problems caused by changes in illumination and driver posture. Six measures are calculated with percentage of eyelid closure, maximum closure duration, blink frequency, average opening level of the eyes, opening velocity of the eyes, and closing velocity of the eyes. These measures are combined using Fisher’s linear discriminated functions using a stepwise method to reduce the correlations and extract an independent index. Results with six participants in driving simulator experiments demonstrate the feasibility of this video-based drowsiness recognition method that provided 86% accuracy.

LIBRARY

OpenCV:

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

D-lib :

D-lib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. D-lib allows you to use it in any application, free of charge.

It gives us 68 facial landmarks pre built.
Facial landmarks are used to localize and represent salient regions of the face, such as:

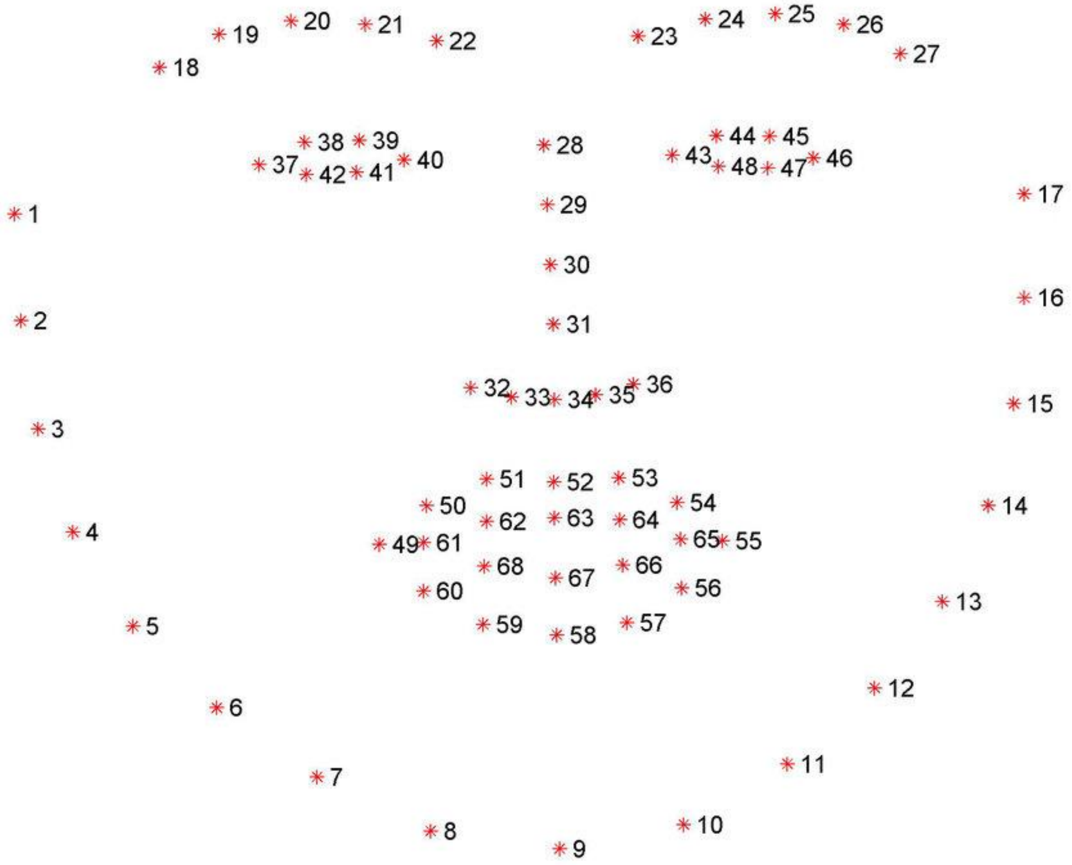
- Eyes

- Eyebrows

- Nose

- Mouth

- Jawline

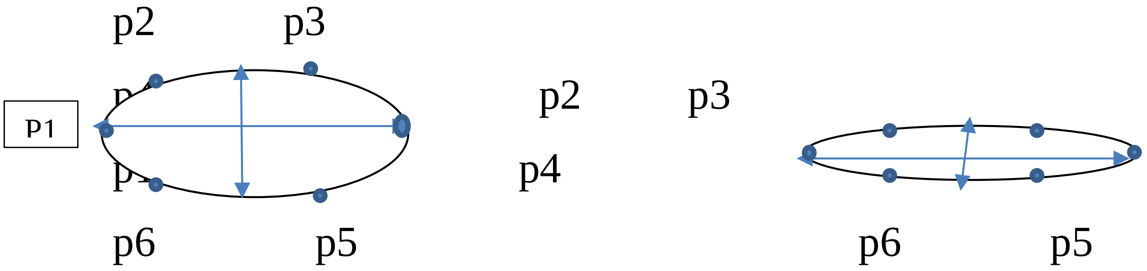


The indexes of the 68 coordinates can be visualized on the image above.

▪ Eyes Open and Close using EAR formula

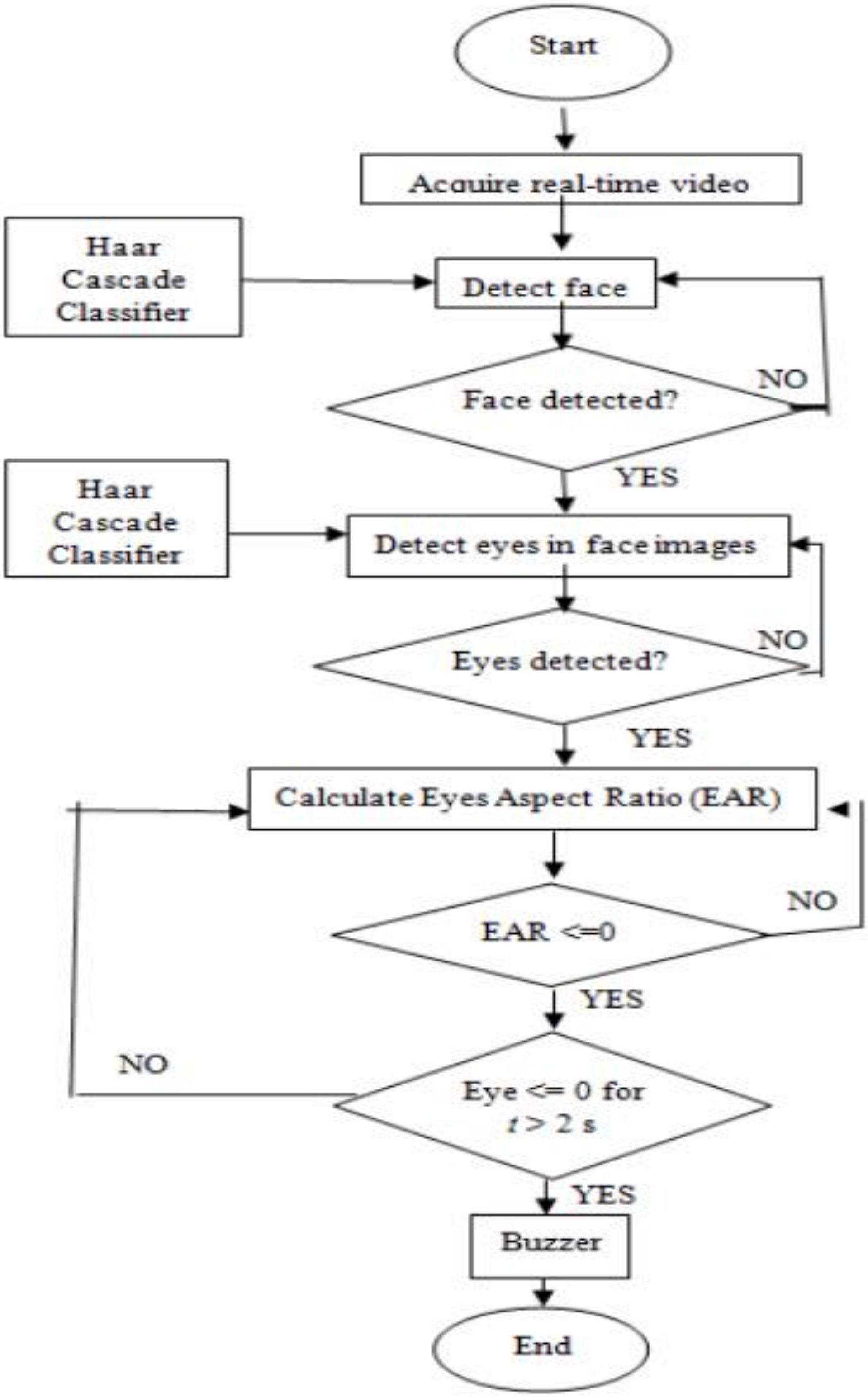
EAR stands for Eye Aspect Ratio

<u>EAR = (p2- p6) +(p3- p5) / 2(p1- p4)</u>



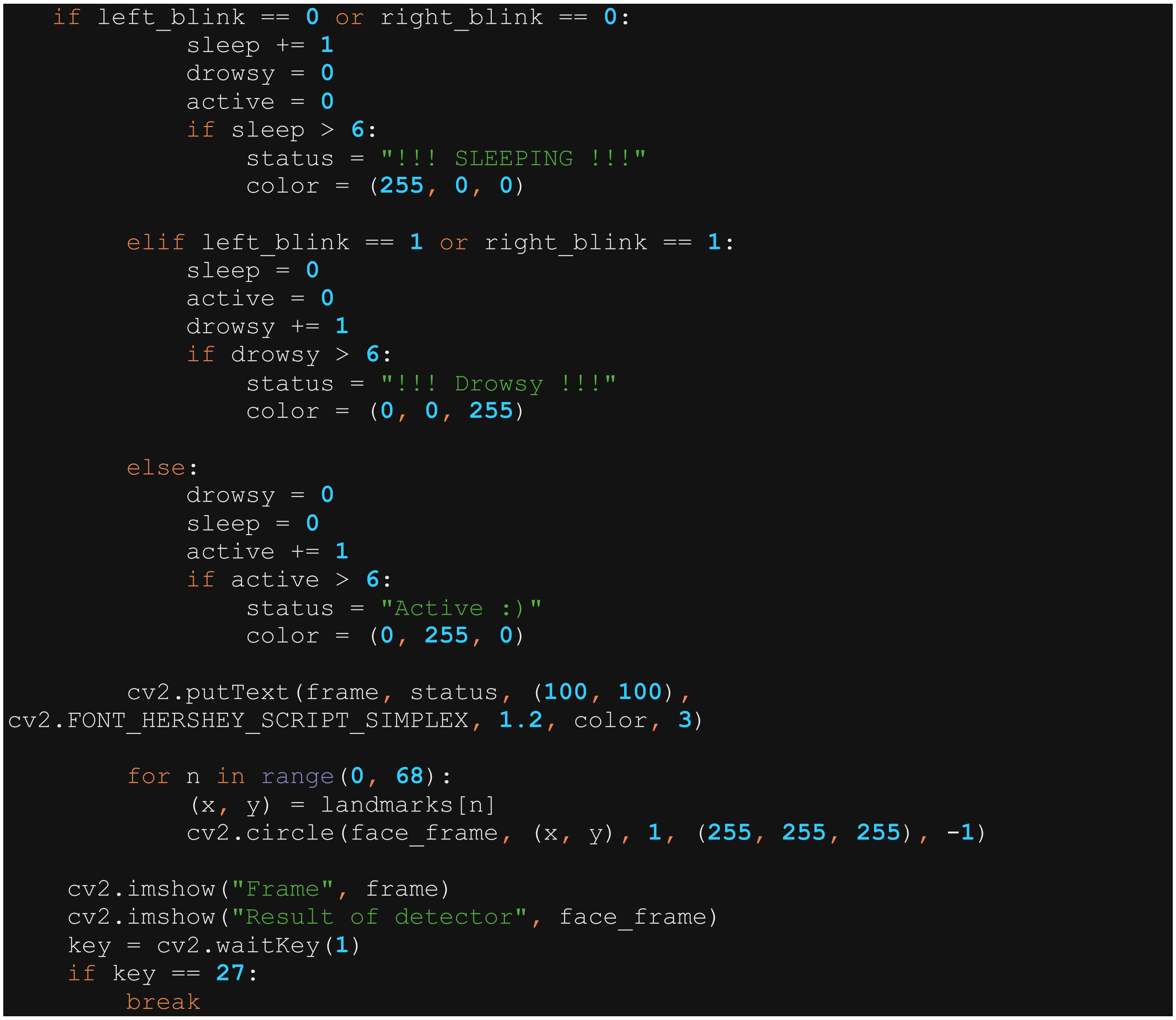
Open Eye Closed Eye

FLOWCHART


Start
Acquire real-time video
Haar Cascade Classifier
Detect face
Face detected?
NO
YES
Haar Cascade Classifier
Detect eyes in face images
Eyes detected?
NO
YES
Calculate Eyes Aspect Ratio (EAR)
EAR <=0
NO
YES
Eye <= 0 for
t > 2 s
NO
YES
Buzzer
End

CODE

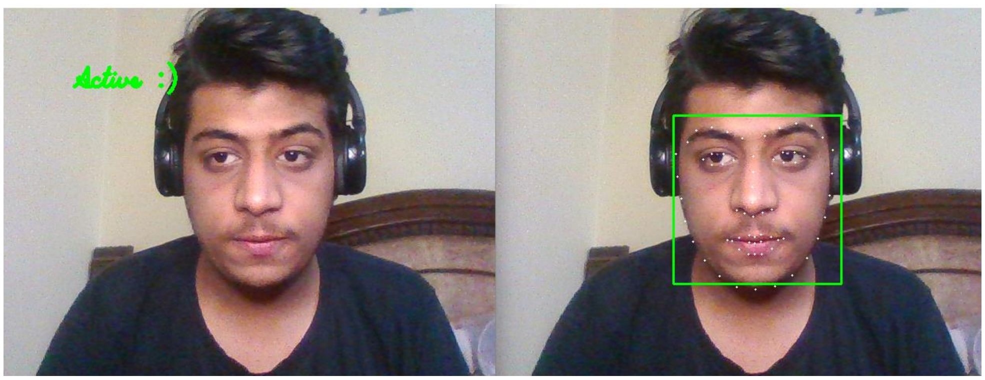
```
import cv2
import numpy as np
import dlib
from imutils import face_utils

cap = cv2.VideoCapture(0)

detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

sleep = 0
drowsy = 0
active = 0
status = ""
color = (0, 0, 0)


def compute(ptA, ptB):
    dist = np.linalg.norm(ptA - ptB)
    return dist


def blinked(a, b, c, d, e, f):
    up = compute(b, d) + compute(c, e)
    down = compute(a, f)
    ratio = up / (2.0 * down)

    # Checking if it is blinked
    if ratio > 0.25:
        return 2
    elif (ratio > 0.21) and (ratio <= 0.25):
        return 1
    else:
        return 0


while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)
    for face in faces:
        x1 = face.left()
        y1 = face.top()
        x2 = face.right()
        y2 = face.bottom()

        face_frame = frame.copy()
        cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

        landmarks = predictor(gray, face)
        landmarks = face_utils.shape_to_np(landmarks)

        left_blink = blinked(landmarks[36], landmarks[37],
                             landmarks[38], landmarks[41], landmarks[40],
landmarks[39])
        right_blink = blinked(landmarks[42], landmarks[43],
                              landmarks[44], landmarks[47], landmarks[46],
landmarks[45])
```



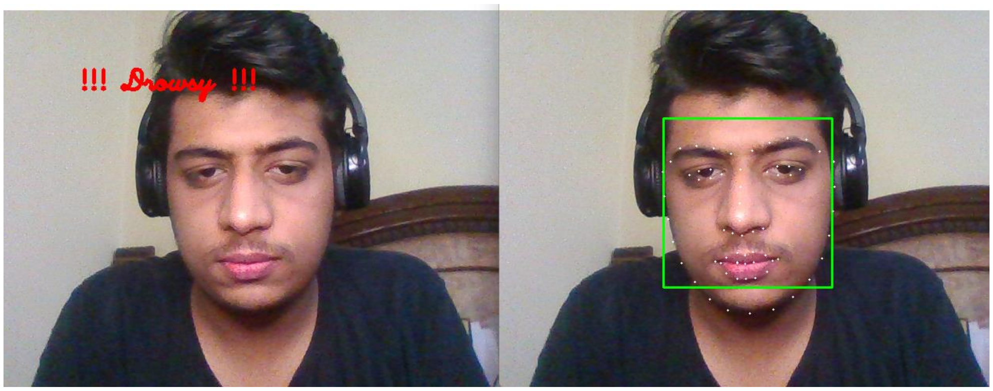
```
   if left_blink == 0 or right_blink == 0:
            sleep += 1
            drowsy = 0
            active = 0
            if sleep > 6:
                status = "!!! SLEEPING !!!"
                color = (255, 0, 0)

        elif left_blink == 1 or right_blink == 1:
            sleep = 0
            active = 0
            drowsy += 1
            if drowsy > 6:
                status = "!!! Drowsy !!!"
                color = (0, 0, 255)

        else:
            drowsy = 0
            sleep = 0
            active += 1
            if active > 6:
                status = "Active :)"
                color = (0, 255, 0)

        cv2.putText(frame, status, (100, 100),
cv2.FONT_HERSHEY_SCRIPT_SIMPLEX, 1.2, color, 3)

        for n in range(0, 68):
            (x, y) = landmarks[n]
            cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)

    cv2.imshow("Frame", frame)
    cv2.imshow("Result of detector", face_frame)
    key = cv2.waitKey(1)
    if key == 27:
        break
```

OUTPUT

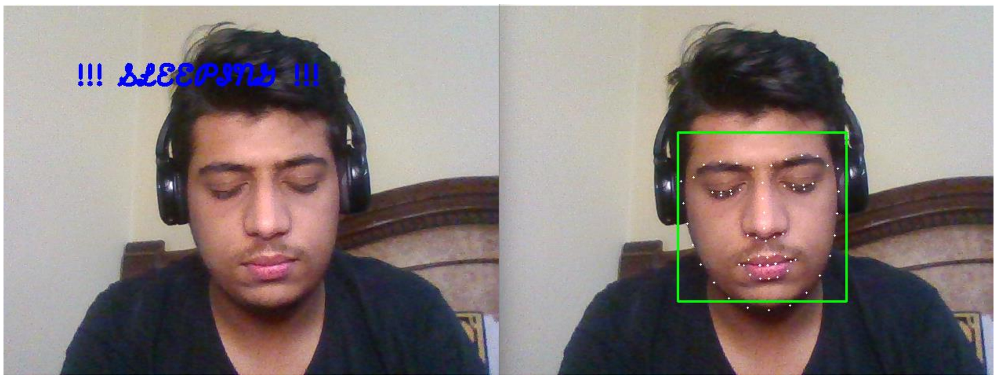
ACTIVE STATUS:



DROWSY STATUS:



SLEEPING STATUS:



CONCLUSION

Several strategies for detecting sleepiness have been presented in previous studies. Following a literature review, many strategies for detecting driver drowsiness have been discovered, each of which uses a different form of data as input to their algorithm. Following a review of many approaches, it was discovered that utilizing a camera is the ideal option, as it is simple to use and acceptable in all situations. We went to evaluate this computer vision technology and suggested a magnificent way for detecting driver drowsiness based on eyelids closing and opening detection utilizing artificial neural networks as a classification algorithm.

First of all, the video frames are acquired from the camera which could be fixed in such a way that it should not obstruct the road-view of the driver. Secondly, we will use the OpenCV for the detection of face and eyes. By using this we will able to locate the face. At last, we will use D-lib for the accurate precision of facial landmarks which includes Eyes, Nose, Lips, Eyebrows and Ears.

REFERENCES

1. https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python
2. https://www.scribd.com/book/15491045/Learning-OpenCV-Computer-Vision-with-the-OpenCV-Library
3. Learning OpenCV by Programming Knowledge [YT]
4. https://en.wikipedia.org/wiki/Dlib#:~:text=DLib%3A%20Library%20for%20Machine%20Learning
5. https://www.kdnuggets.com/2014/06/dlib-library-machine-learning.html
6. https://towardsdatascience.com/real-time-eye-tracking-using-opencv-and-dlib-b504ca724ac6
7. https://www.hindawi.com/journals/acisc/2018/1439312/
8. https://arxiv.org/ftp/arxiv/papers/2010/2010.08278.pdf