Drowsiness Detection System

A Project Report
Submitted by

Hrishita Goyal B030

Rhea Gupta B031

Under the Guidance of

Dr. Mahesh Maurya

in partial fulfillment for the award of the degree of

Bachelor's of Technology
Computer Engineering

At

Mukesh Patel School of Technology Management and Engineering, Mumbai

OCTOBER, 2020

Table of contents

1. INTRODUCTION:

1.1 Motivation:

Drowsiness influences mental alertness, decreasing an individual's capability to handle a vehicle safely and expanding the possibility of a human mistakes that could lead to deaths and injuries. Furthermore, it has been indicated to slow response time, decreases awareness, and impairs judgment. The scariest part is that drowsy driving is not just falling asleep while driving. Drowsy driving can be as small as a brief state of unconsciousness when the driver is not paying full attention to the road.

Additionally, we believe that drowsiness can negatively impact people in working and classroom environments as well. Although sleep deprivation and college go hand in hand, drowsiness in the workplace especially while working with heavy machinery may result in serious injuries like those that occur while driving drowsily.

It is most important that a drowsiness warning system guarantees safety and reliance. Our solution to this problem is to build a detection system that identifies key attributes of drowsiness and triggers based on eye movements and generates an alert when someone is drowsy before it is too late. We plan to implement our project on python using various python libraries for machine learning and artificial intelligence.

1.2 Problem Statement:

Development and implementation of machine learning & artificial intelligence for drowsiness detection using pupil movements.

The system will accept live video of a person, system will detect its pupil, a frame like structure will be formed around eye. As soon as drowsiness is detected an alarm will start buzzing which can only be stopped manually.

1.3 Existing System and Need for New System:

Previous approaches to drowsiness detection primarily make assumptions about the relevant behaviour, focusing on blink rate, eye closure, and yawning .The automobile business also has tried to build several systems to predict driver drowsiness but there are only a few commercial products available today. The systems do not look at driver performance and overlook driver ability and characteristics. Naturally, most people would agree that different people drive differently. The system that being develop able to adapt to the changes of the driver's behaviour.

2. SCOPE OF PROJECT:

Evaluation of different pupil movements to check if the person is drowsy of not. We aim to build a system enables efficient way of identifying fatigue conditions and using analysis in different sections such as driving, education, workplace etc. Minimize the accidents and overhead of the drivers, employees. This project has the scope to include more detections like heart rate, yawn activities etc to get a better evaluation of his inappropriate situations.

2.1 Stakeholders:

- Drivers
- Students
- Teachers
- Employees

2.2 Feasibility of the Project:

- Technical Feasibility: The projects technically feasible as it majorly involves application of eye detection on live video. It will just require a hardware camera support that can be easily installed.
- Financial Feasibility: The cost to implement isn't much as the devices like computers, laptop already possess attached camera to detect video and it can be easily installed in the car for safe driving.
- Management Feasibility: As most of the users are tech savy much training would need to be given to them making management of this project feasible.

3. REQUIREMENTS:

3.1 Hardware-

Hardware system requirements often specify the operating system version, processor type, memory size, available disk space and additional peripherals.

- Intel Core i5 CPU
- Webcam/ External
- System type- 64 – Bit operating System

3.2 Software-

Software system requirements includes software dependencies eg libraries, driver version, framework version

- Python 3.8
- Tensorflow
- Jupyter Notebook

3.3 Libraries-

1. **Numpy-** It is a python library used for working with arrays mainly for calculation the distance
2. **Dlib-** It act as a tool kit for machine leaning applications and complex software.
3. **Threading-** The threading module provided with Python includes a simple-to-implement locking mechanism that allows you to synchronize threads.
4. **Imutils-** It contains series of convenience functions to make image processing such as translation, rotation, resizing, skeletonization and another image manipulating fuctions.
5. **Scipy-** It is used to solve scientific and mathematical problems. It is built on the numpy extension and allows the user to manipulate and visualize data with a wide range of high-level commands.
6. **Pygame**- It is open source module used to build multimedia applications.
7. **CV2-** It is library of python bindings designed to solve computer vision problems.

4. LITERATURE REVIEW:

- We read a variety of papers which included literature from past research projects, conferences, and journals on the drowsiness detection system.
- A comprehensive search was studied and reviewed to identify key studies, reports and researches initiative addressing drowsiness toward driving issues.
- It is attended to investigate the available knowledge in the field and to distinguish the most encouraging indicators of drowsiness drivers.
- Most of these methodologies have only been developed in the laboratory or have had a limited application on-road.
- In the current development of the drowsiness detection system, the possible criteria used is the driver's current state, especially relating to the eye and eyelid movements and physiological state changes.

5. DESIGN AND ANALYSIS:

5.1 Algorithms

Facial landmark prediction algorithm:

Every Human Face has 68 distinct points on his face. The position on each of these 68 facial points is stored in a matrix. So, with the help of this algorithm, the positions of both left eye and right eye can be obtained.

Figure 5.1: Points for FLP Algorithm

Eye Aspect Ratio (EAR Algorithm)

Eye Aspect Ratio is the ratio of vertical height and horizontal length of the eye. When an eye is open, the EAR value is above 0.3 for typical Indian people. However, when eyes are closed, the EAR value falls to 0.19 or even less.

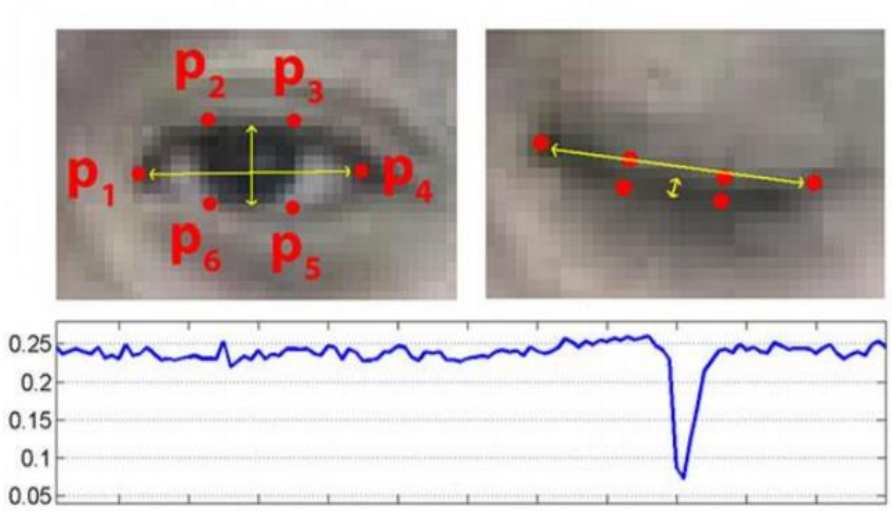


Figure 5.2: EAR value for eye close/open

5.2 Design Methodology

Step 1: Configure the camera and assign the video properties.

Step 2: Obtain the live feed using the start function.

Step 3: Convert the video into individual frames.

Step 4: Detect the face using Haar Cascade Algorithm.

Step 5: Crop frames such that only the face is retained.

Step 6: Recognize the face and continue

Step 7: Detect the eye region and using edge detection procedure, separate left and right eyes.

Step 8: Perform EAR Algorithm to detect if the eye is close or open.

Step 9: If EAR is below 0.3, conclude its drowsiness.

Step 10: If the status is found drowsy, initiate an alarm.

Step 11: Send an alert of drowsy driver and manually stop the alarm once the driver is awake.

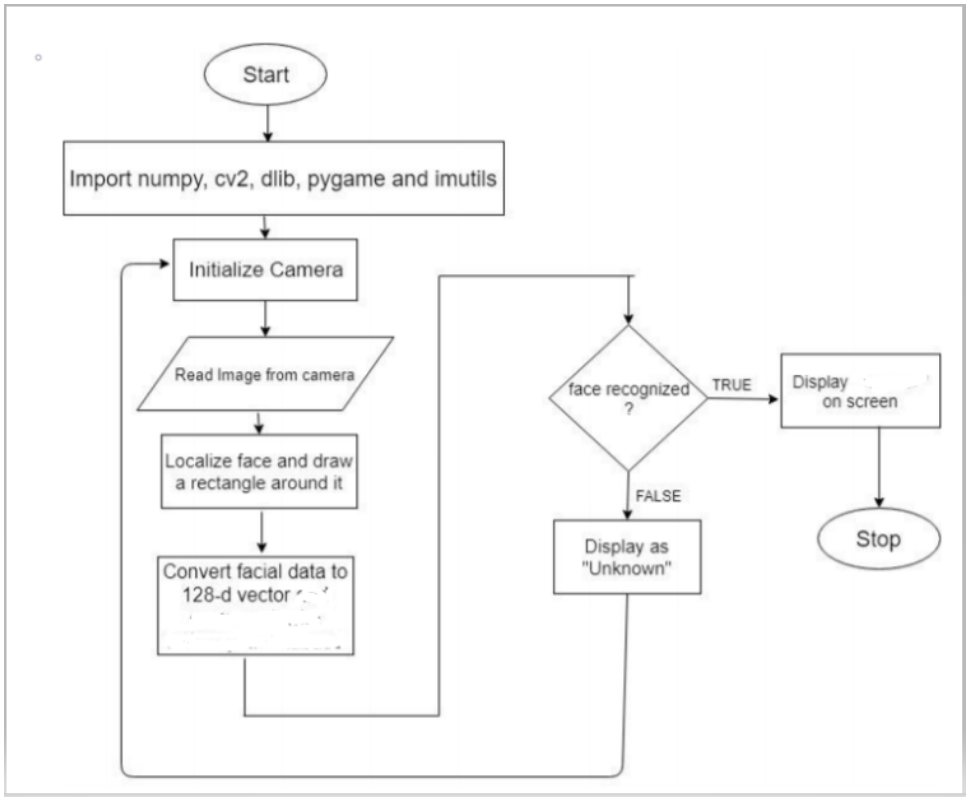


Figure 5.3: Flowchart of Facial Recognition

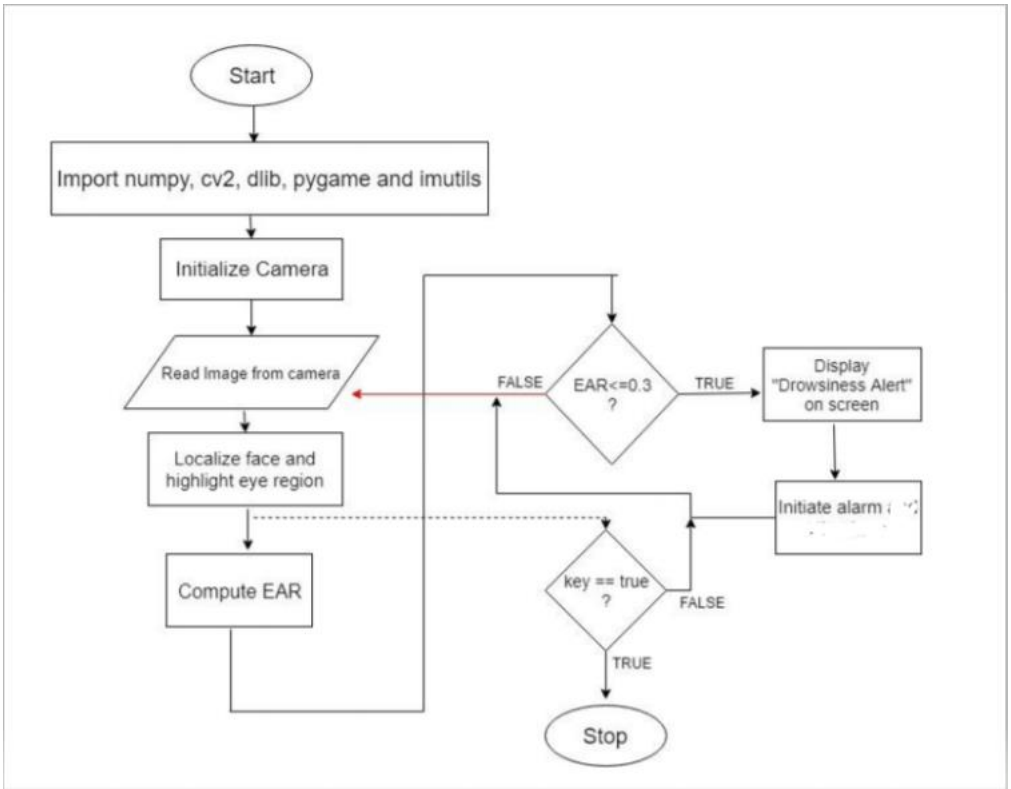


Figure 5.4: Flowchart of Drowsiness Detection

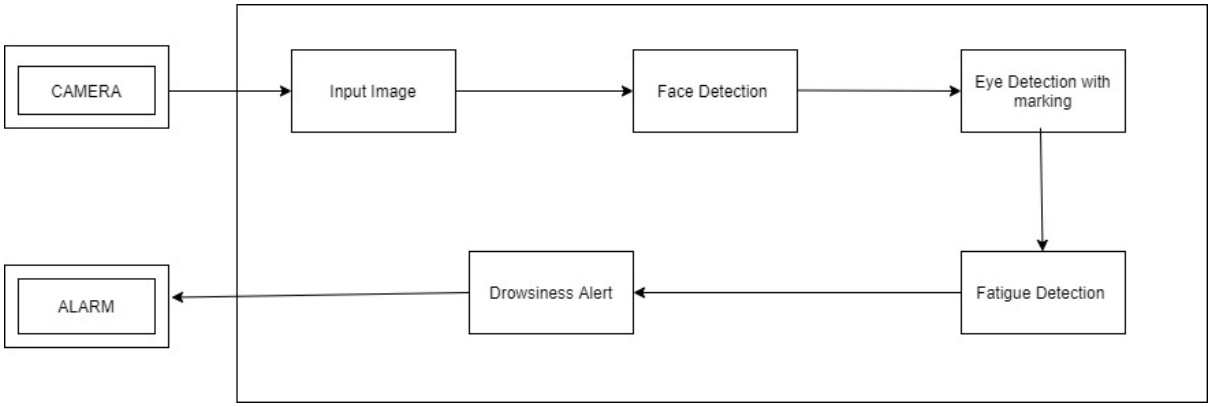
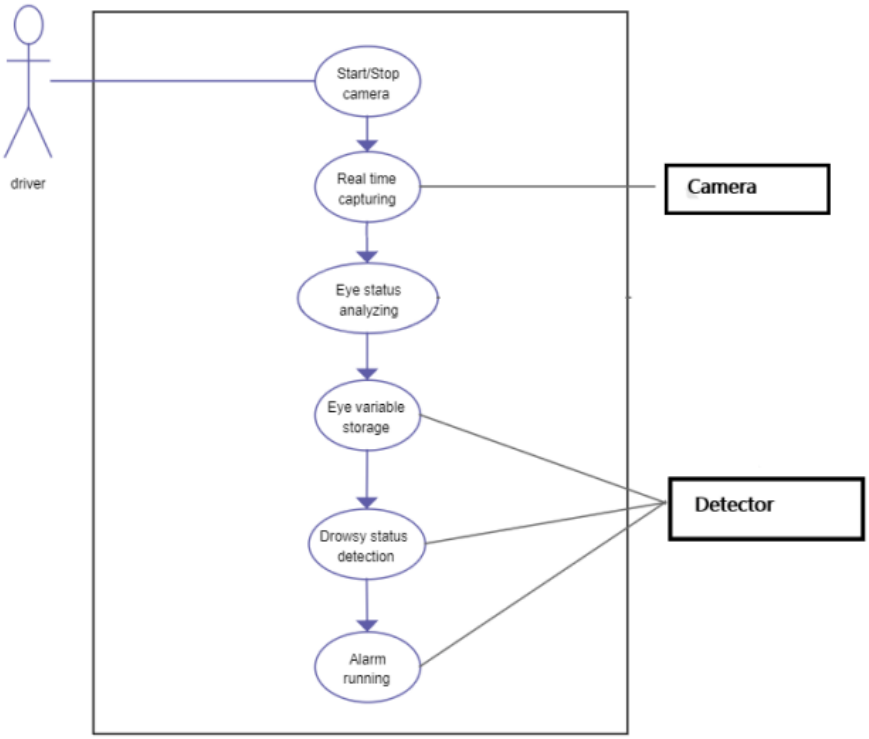


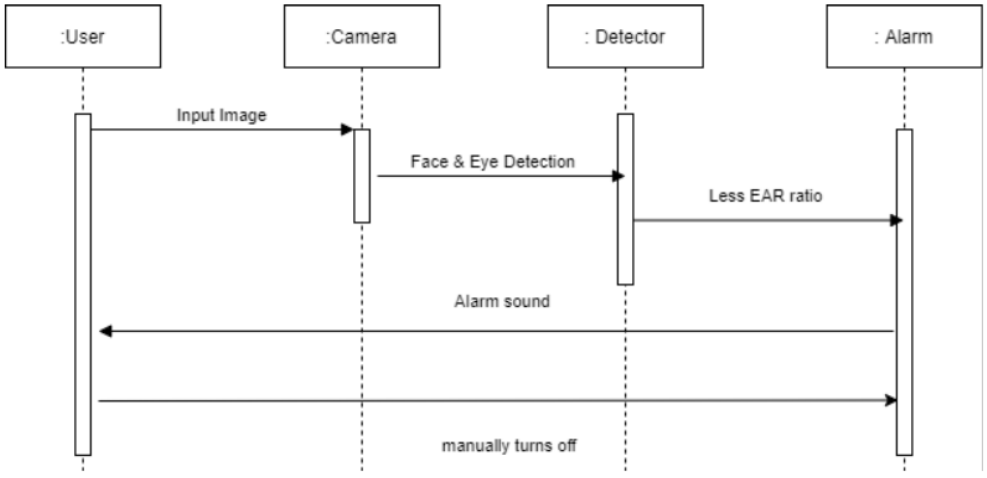
Fig 5.5: Implementation of proposed system

6.UML DIAGRAMS:

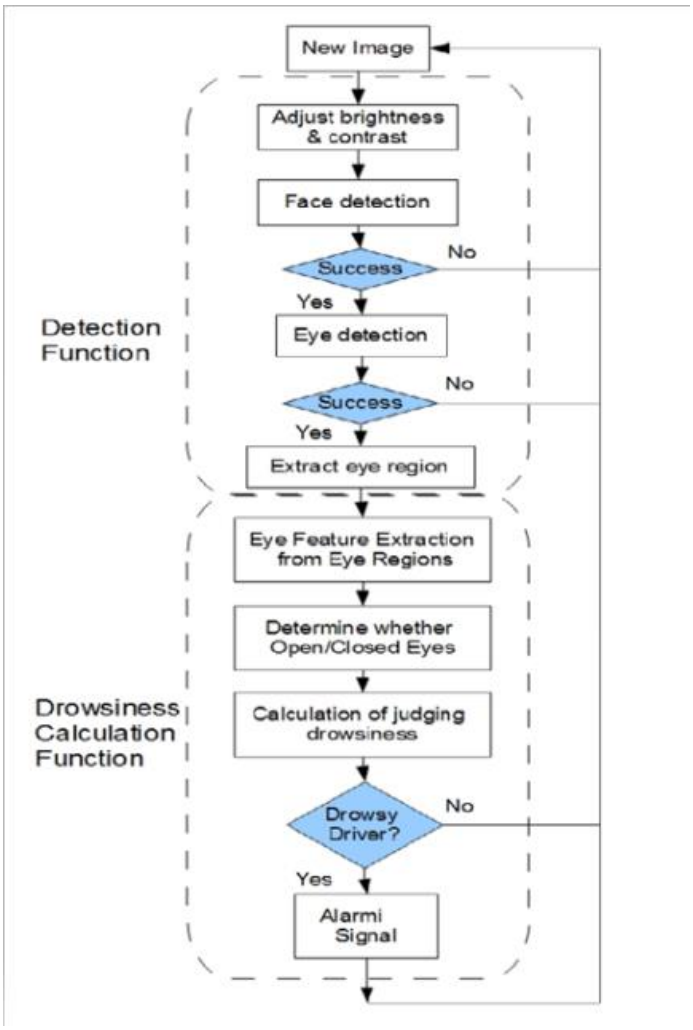
Use case diagram:



Sequence Diagram:



Activity Diagram:



7. CODE:

STEP 1

Drowsiness Detector- B030,B031

IMPORTING LIBRARIES

1. Numpy- It is a python library used for working with arrays mainly for calculation the distance
2. Dlib- It act as a tool kit for machine leaning applications and complex software.
3. Threading- The threading module provided with Python includes a simple-to-implement locking mechanism that allows you to synchronize threads.
4. Imutils- It contains series of convenience functions to make image processing such as translation, rotation, resizing, skeletonization and another image manipulating fuctions.
5. Scipy- It is used to solve scientific and mathematical problems. It is built on the numpy extension and allows the user to manipulate and visualize data with a wide range of high-level commands.
6. Pygame- It is open source module used to build multimedia applications.
7. CV2- It is library of python bindings designed to solve computer vision problems.

In [1]:
```
import numpy as np
import dlib    #To detect and localize facial landmarks
import cv2
from threading import Thread
import threading
import imutils
from imutils import face_utils
from scipy.spatial import distance as dist
import pygame
```

STEP 2

SOUND ALARM

Here we define the sound/alarm to be played once drowsiness is detected.

In [2]:
```
def sound_alarm():
    pygame.mixer.init()
    pygame.mixer.music.load("C:\\Users\\hp\\Desktop\\dataset\\Alarm.mp3")
    pygame.mixer.music.play()
```

STEP 3

DEFINING EAR(EYE ASPECT RATIO) ALGORITHM

Eye Aspect Ratio is the ratio of vertical height and horizontal length of the eye. When an eye is open, the EAR value is above 0.3 for typical Indian people. However, when eyes are closed, the EAR value falls to 0.19 or even less.

In [3]:
```
def eye_aspect_ratio(eye):

    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])   #vertical distance

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

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

    return ear
```

STEP 4

DEFINING COMMAND LINE ARGUEMENTS USING ARGPARSE LIBRARY

Command line arguments are flags given to a program/script at runtime. They contain additional information for our program so that it can execute. Here we add arguements using The executable, the file and an arguement in order to print the usage.

In [4]:
```
'''ap = argparse.ArgumentParser()

ap.add_argument("-p", "--shape-predictor", required=True, help="path to facial landmark predictor")

ap.add_argument("-a", "--alarm", type=str, default="", help="path alarm .WAV file")

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

args = vars(ap.parse_args(args=[]))'''
```

STEP 5

1.0.5 DEFINITIONS

Here we define the threshold, frames and set counter to 0 and alarm off.

In [5]:
```
EYE_AR_THRESH = 0.25
EYE_AR_CONSEC_FRAMES = 20

COUNTER = 0
ALARM_ON = False
```

STEP 6

1.0.6 Preparing for Prediction and detection

In [6]:
```
predictor_path = 'C:\\Users\\hp\\Desktop\\dataset\\shape_predictor_68_face_landmarks.dat'
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(predictor_path)
```

Facial landmark prediction algorithm: Every Human Face has 68 distinct points on his face. The position on each of these 68 facial points is stored in a matrix. So, with the help of this algorithm, the positions of both left eye and right eye can be obtained.

In [7]:
```
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
```

STEP 7

1.0.7 Applying EAR Algorithm

```
Eye Aspect Ratio is the ratio of vertical height and horizontal length of the eye. When an eye is open, the EAR value is above
0.3 for typical Indian people. However, when eyes are closed, the EAR value falls to 0.19 or even less.

Step 1: Configure the camera and assign the video properties.
Step 2: Obtain the live feed using the start function.
Step 3: Convert the video into individual frames.
Step 4: Detect the face using Haar Cascade Algorithm.
Step 5: Crop frames such that only the face is retained.
Step 6: Recognize the face and continue
Step 7: Detect the eye region and using edge detection procedure, separate left and right eyes.
Step 8: Perform EAR Algorithm to detect if the eye is close or open.
Step 9: If EAR is below 0.3, conclude its drowsiness.
Step 10: If the status is found drowsy, initiate an alarm.
Step 11: Send an alert of drowsy driver and manually stop the alarm once the driver is awake.
```

In [ ]:

```
vs = cv2.VideoCapture(0)

while True:
    ret, frame = vs.read()
    if ret == False:
        print('Failed to capture frame from camera. Check camera index in cv2.VideoCapture(0) \n')
        break
   # cv2.imshow(frame)

    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rects = detector(gray, 0)

    for rect in rects:
        shape = predictor(gray, rect)    # determine the facial landmarks for the face region
        shape = face_utils.shape_to_np(shape) #converting to numpy array

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

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

        ear = (leftEAR + rightEAR) / 2.0

        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)


        if ear < EYE_AR_THRESH:
            COUNTER += 1

            if COUNTER >= EYE_AR_CONSEC_FRAMES:
                if not ALARM_ON:
                    ALARM_ON = True
                    d=threading.Thread(target=sound_alarm)
                    d.setDaemon(True)
                    d.start()

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

        else:
            COUNTER = 0
            ALARM_ON = False

        cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30), 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.release()
```

8. FUTURE SCOPE:

- The future works may focus on the utilization of outer factors such as vehicle states, sleeping hours, weather conditions, mechanical data, etc, for fatigue measurement.
- Currently there is not adjustment in zoom or direction of the camera during operation. Future work may be to automatically zoom in on the eyes once they are localized.
- Similarly for workplace and education, other factors may be considered for more insightful and in depth analysis.

9. IMPACT:

This project has the potential to have significant societal impact within our country. The main aim is to contribute to the study of driver behaviour while driving, through the development and evaluation of a drowsiness driver model system. It can help greatly in efficiency of the employees in their work and in decreasing the fatigues and laziness in their work process. It can benefit people in general, particularly at such times where everyone is working from home for their offices or attending those people who afford to lose their concentration while driving. It will ensure the student's productivity, as this system will be able to keep a track of their drowsiness and alert them with alarm.

Driving:

Most road accidents are believed to be fatigue related. An efficient drowsiness detector would help reduce these unfortunate events significantly. The alarm would alert the driver in case of a normal or a self-driving car.

Education/workplace:

Since most of the learning process and work has been shifted online due to the current scenario. The strain on the eyes is commensurate to the time spent looking at the screen. Drowsiness detection can help alert the student/teacher/employee and make their concentration levels apt to grasp the necessary information.

10. REFRENCES:

i. Sahayadhas, A., Sundaraj, K., & Murugappan, M. (2012). Detecting driver drowsiness based on sensors: a review. Sensors, 12(12), 16937-16953.

ii. Forsman, P. M., Vila, B. J., Short, R. A., Mott, C. G., & Van Dongen, H. P. (2013). Efficient driver drowsiness detection at moderate levels of drowsiness. *Accident Analysis & Prevention*, *50*, 341-350.

iii. Reddy, B., Kim, Y. H., Yun, S., Seo, C., & Jang, J. (2017). Real-time driver drowsiness detection for embedded system using model compression of deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 121-128).

iv. Khunpisuth, O., Chotchinasri, T., Koschakosai, V., & Hnoohom, N. (2016, November). Driver drowsiness detection using eye-closeness detection. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 661-668). IEEE.

v. Mehta, S., Dadhich, S., Gumber, S., & Jadhav Bhatt, A. (2019, February). Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio. In *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India*.

vi. Maior, C. B. S., das Chagas Moura, M. J., Santana, J. M. M., & Lins, I. D. (2020). Real-time Classification for Autonomous Drowsiness Detection Using Eye Aspect Ratio. *Expert Systems with Applications*, 113505.

vii. Mbouna, R. O., Kong, S. G., & Chun, M. G. (2013). Visual analysis of eye state and head pose for driver alertness monitoring. IEEE transactions on intelligent transportation systems, 14(3), 1462-1469.