**Mini Project II**

**<Safe Vehicle Driving With Eye Aspect Ratio>**

Version 1.0



**Institute of Engineering & Technology**

ManishYadav

(161500308)

Sahil Sharma

(161500473)

Shashi Kumar Verma

(161500507)

Supervised By

Mohd. Amir Khan

Technical Trainer

Department of Computer Engineering & Applications

**Chapter 1**

**Introduction**

## In this Project we mainly provide the facility to the people who are driving car at night what we gonna do in this project is that we make a device in which we use camera through which we detect the increasing ratio of the person eye who are driving the car in the ratio is continuously increases per minute then we start the siren in the car so that the person slow down the car and safe from any tragedy.Accidents are the most common cause of death for the peoples who drive vehicles. Accidents happen, even to the people who are careful, but many accidents may be avoidable if simple precautions are taken. One of the major reasons for the accident is drink and drive. We can avoid these accidents by providing the eye blink and alcohol sensors to the drivers. The eye blink switch responds to the voluntary blink of the eye and requires minimal calibration. It discriminates between voluntary and involuntary blinks.

### Aim of the project

The aim of this proposed system is to develop user-friendly, efficient, accurate and low-cost application using image and video processing algorithms to detect driver drowsiness to reduce road accidents and increase driver safety.

### Objectives of the project

1. Research on the identified areas relevant to the project and come up with the literature review. The literature review of the project consists of existing driver safety systems and its functionalities along with the comparison table of the existing systems.
2. Design the system architecture according to the gathered information from the research.
3. Implement the finalized design of the proposed system.
4. Test and evaluate the implemented system.
5. Submit the deliverable on time.

**Chapter 2**

**Software Requirement Analysis**

## 2.1 Define Problem

Automobile has become a major part in our lives. They are used for transportation of people, items and any many other from one place to another. Whatever the automobile type (car, van, bike etc.) is being used for transportation, the safety should be a number one priority. While transporting people or any other item we should think about the safety of the passengers of the vehicle, public and private and public properties. According to Ministry of Transport, New Zealand (2018) number of road deaths has been increased from 284 to 393 within the period of 2011 to 2018. They also state that the number of reported injuries in transportation has amplified from 11,000 to 13,000 within a period of 5 years until 2017. Above statistics confirms that there should be a lot of improvements in the automobile and transportation industry for safety and the wellbeing of humans as well as the safety of private and public property.

There are various factors which contributes to road injuries and deaths. Some of them are lost control of the vehicle, speeding, alcohol, driver drowsiness, weather conditions, vehicle conditions and so on (Ministry of Transport New Zealand, 2018). Most of the time drivers do not care about the state of themselves prior to driving a vehicle. Even though the driver is tired, sleepy or has consumed alcohol he/she tries to drive a vehicle without any anxiety. As a result, it causes road accidents, injuries, loss of lives and damage to property. There aren’t many systems which could predict the conditions of the driver prior to a trivial incident which could assist the driver immensely. Therefore, this project mainly focuses on the driver drowsiness and how it can be identified using computer vision and facial features in order to provide necessary alerts to the driver when required as a warning, so that the driver can decide whether to continue driving or not and take a proactive approach to such incidents rather than a reactive approach.

## 2.2 Solution

Since the road safety is the main priority, most of the high-end vehicle manufacturers implement safety systems in their vehicles. Unfortunately, these vehicles are expensive and most of the people can’t afford to buy them. Therefore, our research questions focus on improving the automobile safety practices using computer vision and low-cost hardware and improving the automobile safety practices by monitoring facial gestures of human.

The proposed solution will be implemented using image processing, computer vision and facial recognition techniques to increase the efficiency and the accuracy of the system. A camera will be the main hardware device to capture images of the eyes and this will reduce the cost of other expensive hardware devices such as embedded sensors and chips.

**2.3 Project motivation and purpose**

The goal of this project is to develop a system that can accurately detect sleepy

driving and make alarms accordingly, which aims to prevent the drivers from drowsy

driving and create a safer driving environment. The project was accomplished by a

Webcam that constantly takes image of driver, a beagle board that implement image processing algorithm of sleepy detection, and a feedback circuit that could generate alarm and a power supply system.

**2.4 Functions and Features**

This system has many features that make it unique and functional. These features include:

1. Eye extraction, use open and close to determine sleepiness

2. Daytime and night detection

3. Real time image processing and detection

4. Little inference and potential hazard to driver’s normal driving

5. Portable size with car cigarette charger socket power supply

**2.5 Limitation**

Even though the completion percentage of this project is 100%, still the prototype is not production ready. The simulation needs more and more testing and the current solution needs to be improved massively due to the fact that this is a safety application. The team has faced below limitations throughout the time span of this particular project which signifies that the solution is not production ready.

1.The simulation will not perform accurately and effectively during inacceptable illumination (light) conditions.

2.The simulation will not work completely at night times when there is no light at all.

3.Face and Eye detection algorithms do not labor during inacceptable illumination conditions including night times.

4.Eye detection is not possible when the user wears sun glasses while using the simulation. This is out of current scope of the proposed solution.

5.Driver must look forward at all times in order to identify the eye open and closure states. Eye states cannot be identified for the side views of the face.

6.Processing power of the computational resource (laptop) affects the performance and the accuracy of the proposed simulation.

7.Quality of the web camera used in the simulation affects the performance and the accuracy of the simulation.

8.Sometimes the camera doesn’t capture frames and results in total frames count to be zero (0). This will also result in zero eye detection and will calculate the actual eye percentage to be (0/0) \*100 = infinity. In computer world infinity is Not a Number (NaN). Therefore, current system alarm activation is disabled for such cases.

**Chapter 3**

**Overview**

**2.1 What is Machine Learning**

Machine Learning is defined as an application of artificial intelligence where available information is used through algorithms to process or assist the processing of statistical data. While Machine Learning involves concepts of automation, it requires human guidance. Machine Learning involves a high level of generalization in order to get a system that performs well on yet unseen data instances**.**

Data science, machine learning and artificial intelligence are some of the top trending topics in the tech world today. Data mining and Bayesian analysis are trending and this is adding the demand for machine learning. This tutorial is your entry into the world of machine learning.

Machine learning is a discipline that deals with programming the systems so as to make them automatically learn and improve with experience. Here, learning implies recognizing and understanding the input data and taking informed decisions based on the supplied data. It is very difficult to consider all the decisions based on all possible inputs. To solve this problem, algorithms are developed that build knowledge from a specific data and past experience by applying the principles of statistical science, probability, logic, mathematical optimization, reinforcement learning, and control theory.

**2 2What is Open CV**

Open CV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. Open CV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, Open CV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million The library is used extensively in companies, research groups and by governmental bodies.

**2.3 What is dlib**

It‘s a landmark’s facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person’s face like image below.

Starting by the image capture that we are going to work on, we will use OpenCV to capture the image’s webcam in an “infinite” loop and thus give the impression of watching a video.

**2.4 What is Imutils**

A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3.

**2.5 What is Numpy**

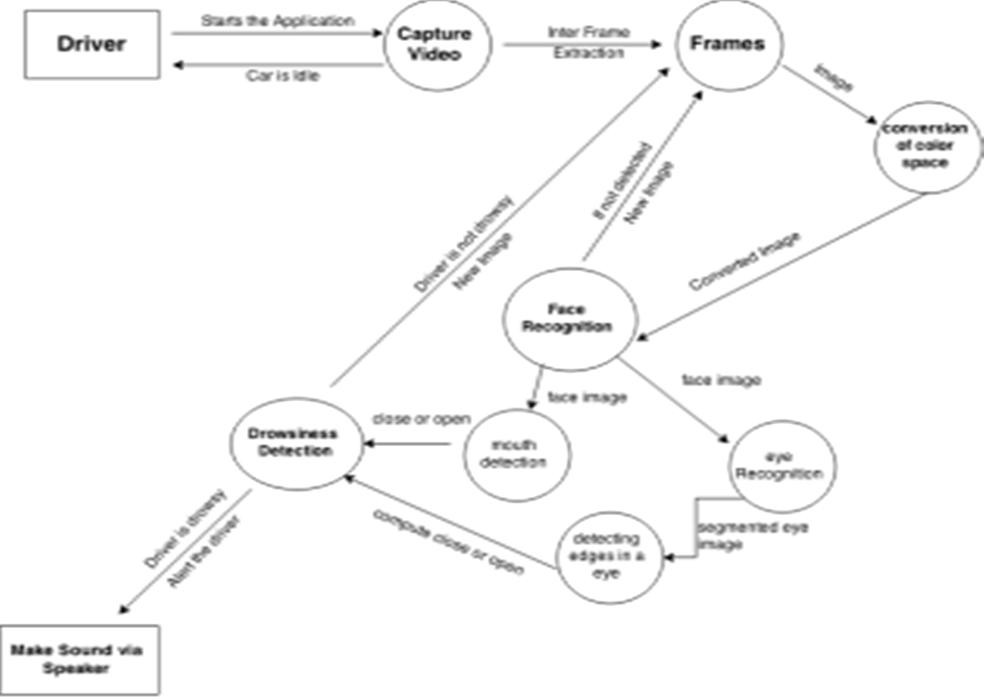
NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided.

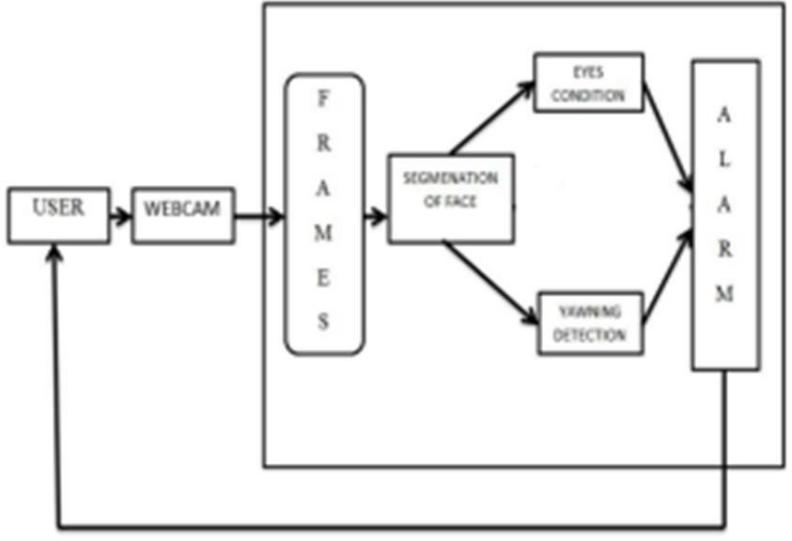
**2.6 what is pyrebase**

Firebase is a backend as a service, with which you can build web, android and iOS apps without setting up a custom backend. It comes with various libraries like authentication, storage, database, notifications etc. While **Pyrebase** is a unofficial Python wrapper for firebase SDK

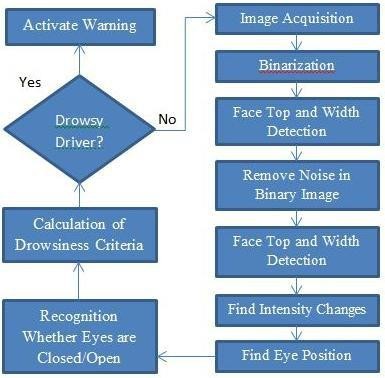
**Chapter 4**

**Software Design**

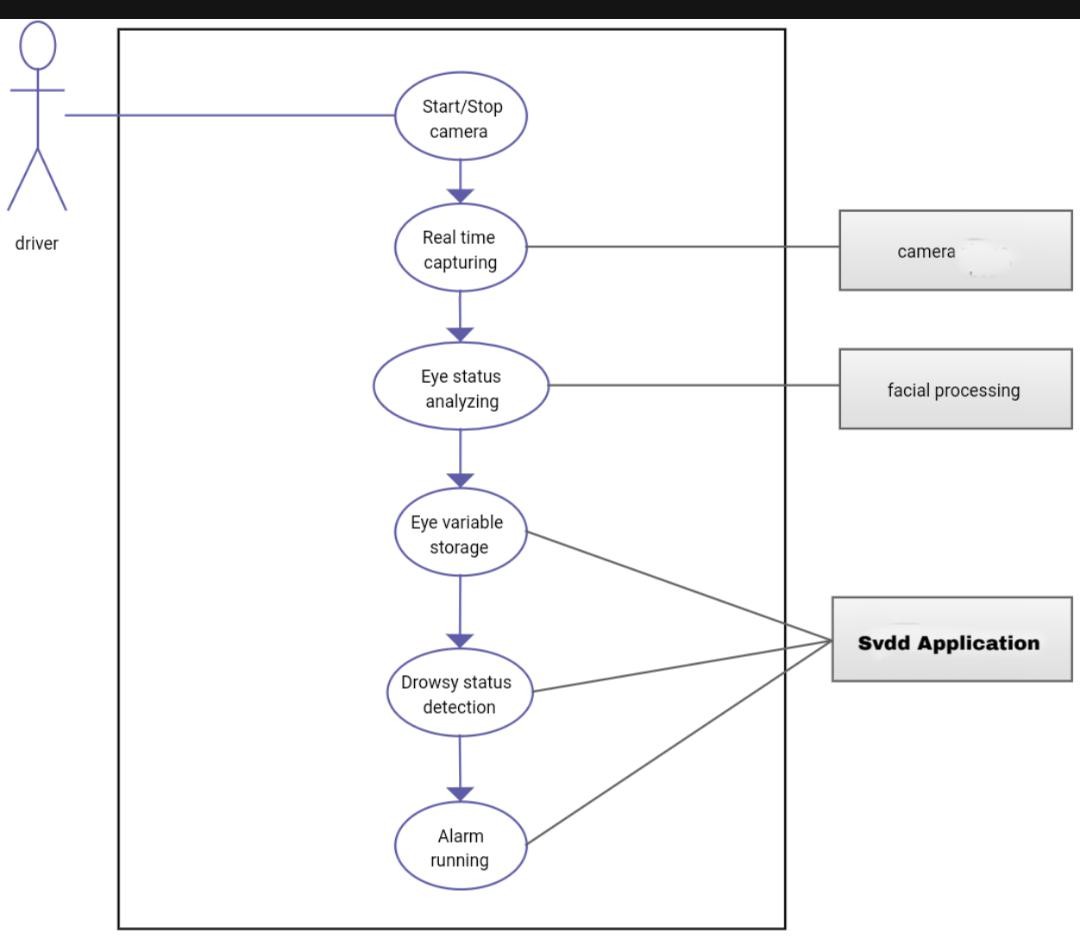
**4.1Data Flow Diagram**

**4.2 Architecture Diagram**

**4.3 Flow Chart**



**4.4 Use case Diagram**



### 4.4.1 Pre-Processing Stage

Pre-processing stage is the primary stage of the proposed driver safety and drowsiness detection system which consists of two different processes. They are;

* Face and Eye Detection stage
* Image processing

The camera will capture images of the driver and send it through the pre-processing stage to detect the face and eyes of the driver. In this stage image processing algorithms are used to convert the live video stream to a sequence of digital images. Once converted, the system will pass those images to position detection stage.

### **4.4.2 Position Detection Stage**

In this stage, driver safety and drowsiness detection system will use Haar Cascade Classifier to detect the exact positions of the face and eye. Haar Cascade Classifier is a trained data set which is used for object and feature recognition. This stage consists of two different processes to detect face and eye. They are;

* Haar cascade face detection
* Haar cascade eye detection

After these processes those images will be transferred to the next stage of the system.

### 4.1.3 Detection Estimation Stage

In this stage, the proposed system will determine the state of the eyes. Basically, the system will keep track on eye lids and detect whether both the eyes are in open or closed state. If the eye lids are closed the iris (dark region of the eye) will be invisible and this will be recognized as an eye closure state.

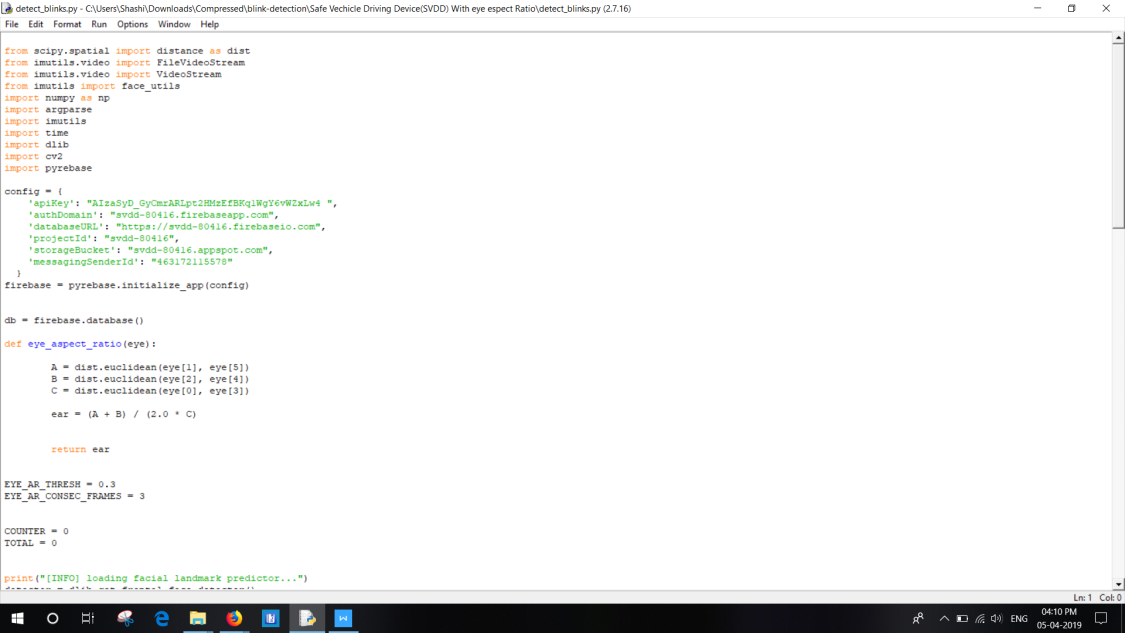
Considering the scenario system will alert the driver by providing a sound if the eye closure state percentage is higher compared to detection percentage supplied by the driver.

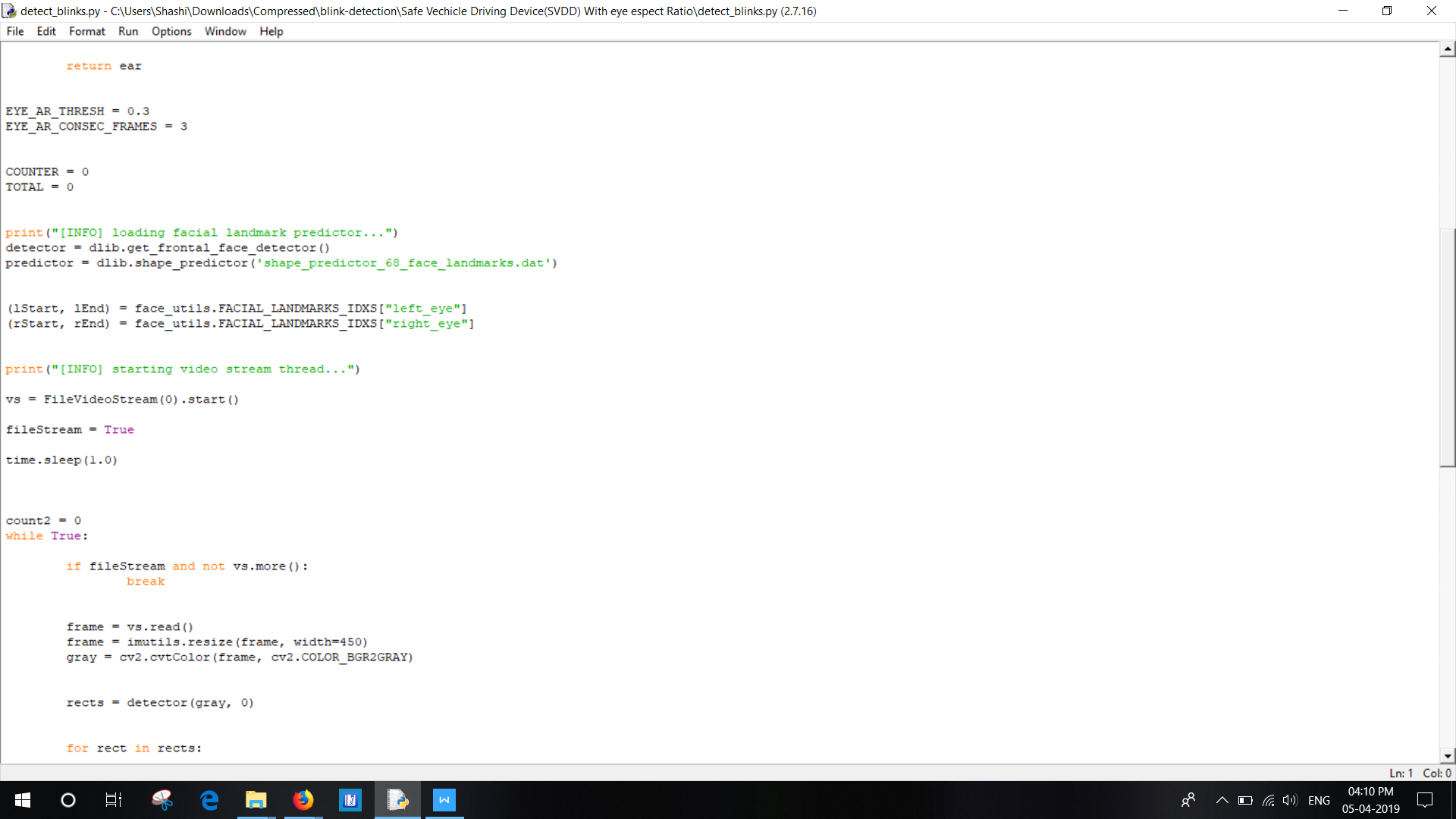
**Chapter 5**

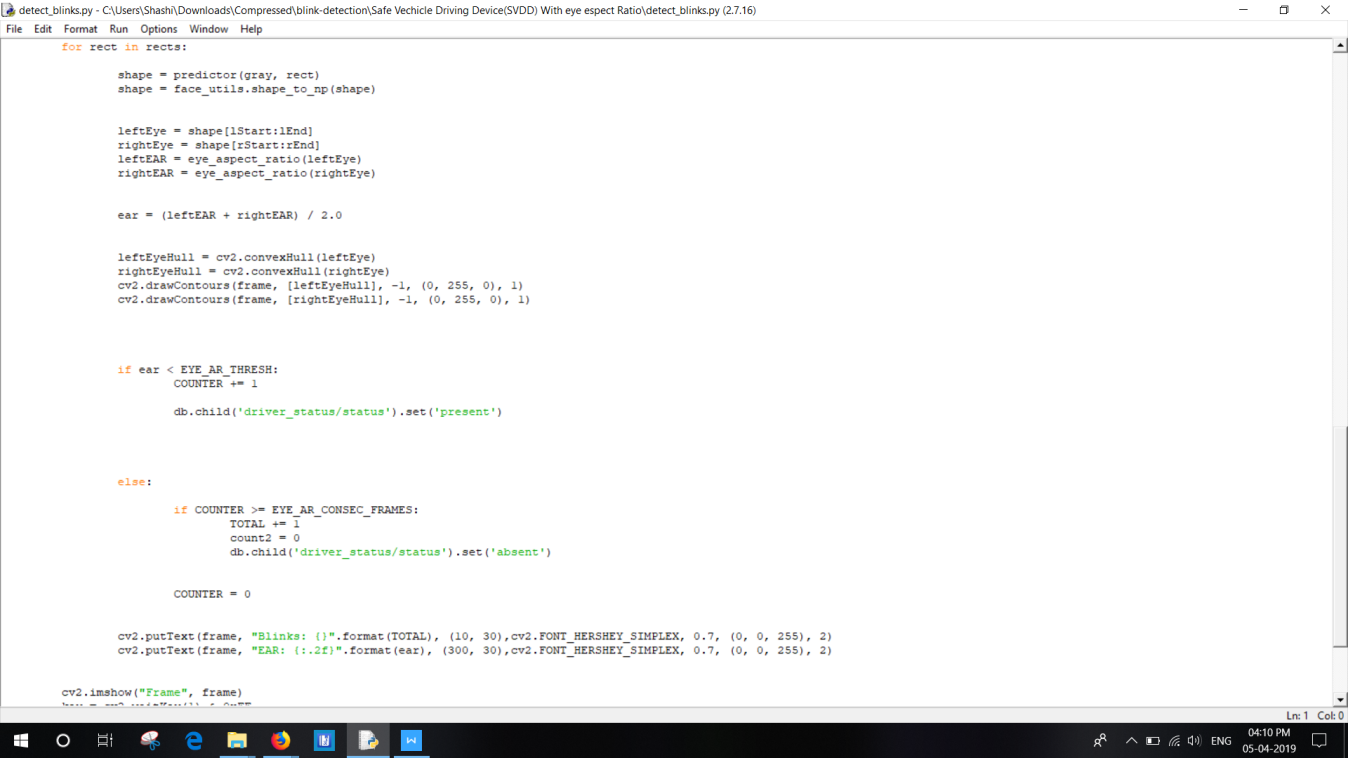
**Implementation and User Interface**

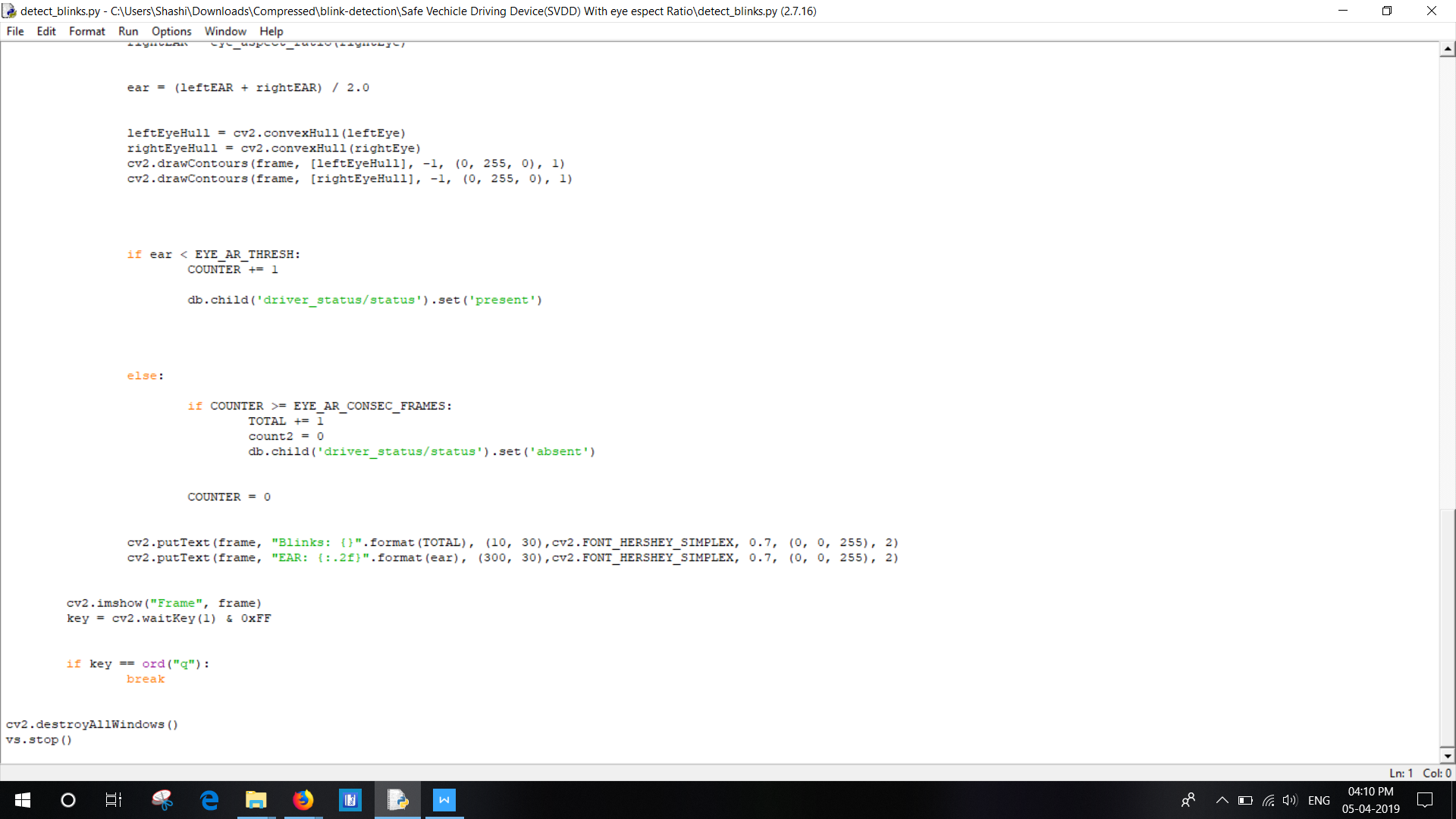
Implementation was one of the main phases since a prototype was a main deliverable of this project. Hence the team had to work hard on the implementation phase to get a stable prototype for successful completion of the project. Team only had approximately two weeks to complete coding and due to lack of expertise in the computer vision concepts the team had to do more spikes in order to understand the concepts. Team understood the difficulty of implementing face and eye detection from scratch and started researching on the existing libraries so that the development time would be reduced and could come up with a better prototype.

**5.1 Coding**

**5.**

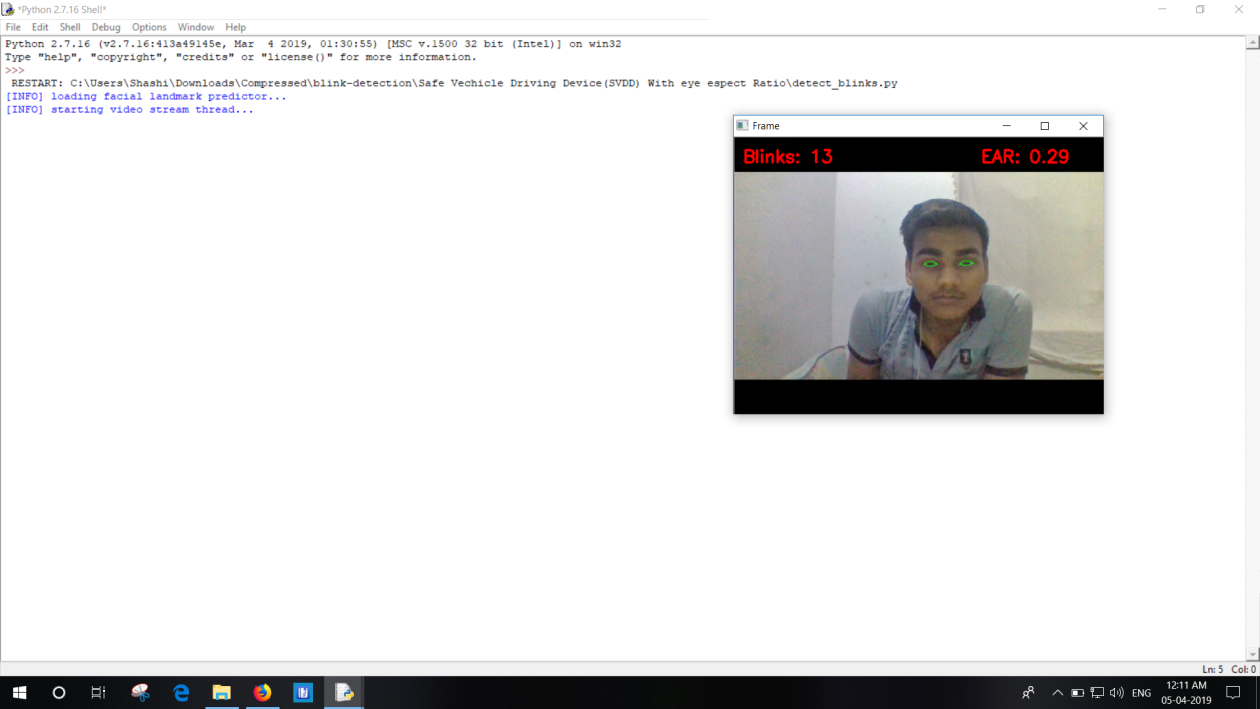
****

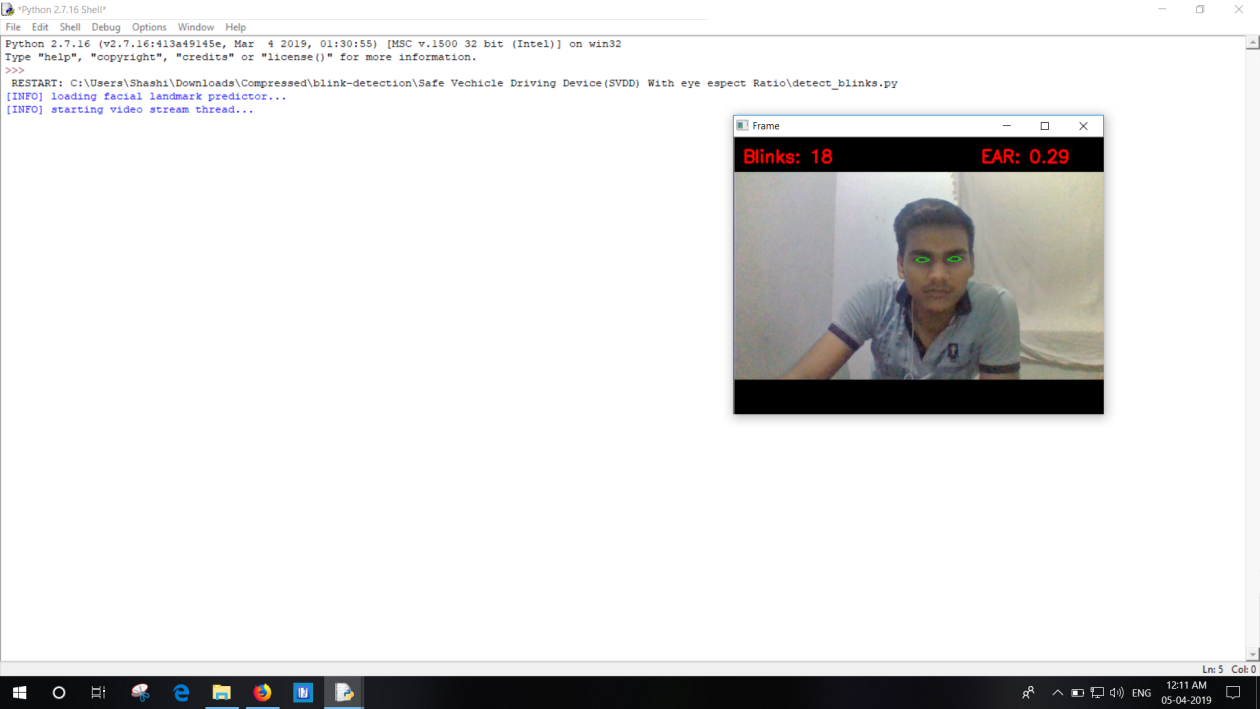
****

****

**5.2 Output**

**Count Eye Blinking**

****

****

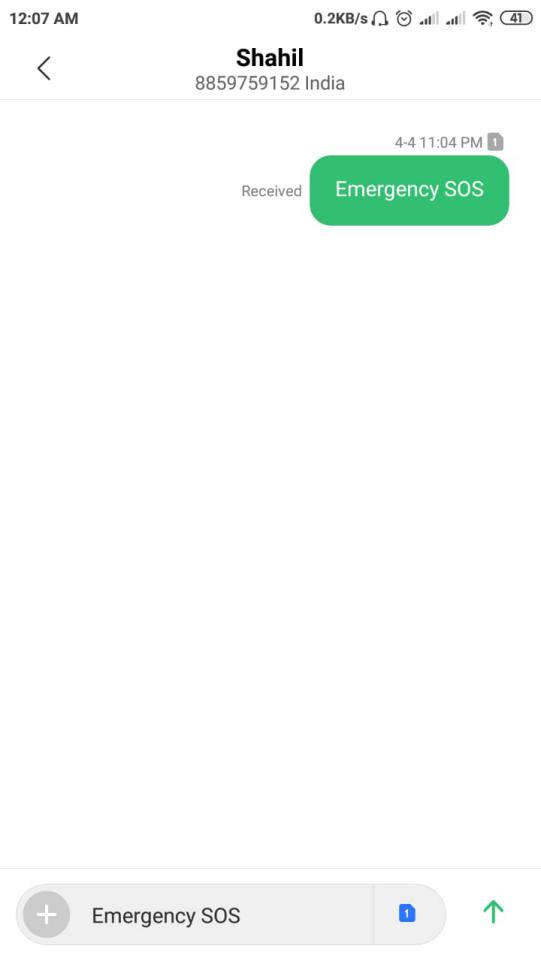
**5.3 Warning Message**

**Condtion : If drive not Sleeping no message and green screen show on app.**

****

**Condition 2. if driver sleeping then app will show red screen in app and send message**

****

****

**Chapter 6**

**References/Bibliography**

1. [https://www.seminarsonly.com](https://www.seminarsonly.com/)

(2)<https://www.mdpi.com/2078-2489/9/4/93/htm>

(3) https://[www.pyimagesearch.com/](http://www.pyimagesearch.com/)

(4) tutorial point.com

5.Wikipedia.com

**Appendices**

**from scipy.spatial import distance as dist**

**from imutils.video import FileVideoStream**

**from imutils.video import VideoStream**

**from imutils import face\_utils**

**import numpy as np**

**import argparse**

**import imutils**

**import time**

**import dlib**

**import cv2**

**import pyrebase**

**config = {**

**'apiKey': "AIzaSyD\_GyCmrARLpt2HMzEfBKq1WgY6vWZxLw4 ",**

**'authDomain': "svdd-80416.firebaseapp.com",**

**'databaseURL': "https://svdd-80416.firebaseio.com",**

**'projectId': "svdd-80416",**

**'storageBucket': "svdd-80416.appspot.com",**

**'messagingSenderId': "463172115578"**

**}**

**firebase = pyrebase.initialize\_app(config)**

**db = firebase.database()**

**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**

**EYE\_AR\_THRESH = 0.3**

**EYE\_AR\_CONSEC\_FRAMES = 3**

**COUNTER = 0**

**TOTAL = 0**

**print("[INFO] loading facial landmark predictor...")**

**detector = dlib.get\_frontal\_face\_detector()**

**predictor = dlib.shape\_predictor('shape\_predictor\_68\_face\_landmarks.dat')**

**(lStart, lEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["left\_eye"]**

**(rStart, rEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["right\_eye"]**

**print("[INFO] starting video stream thread...")**

**vs = FileVideoStream(0).start()**

**fileStream = True**

**time.sleep(1.0)**

**count2 = 0**

**while True:**

**if fileStream and not vs.more():**

**break**

**frame = vs.read()**

**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)**

**shape = face\_utils.shape\_to\_np(shape)**

**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**

**db.child('driver\_status/status').set('present')**

**else:**

**if COUNTER >= EYE\_AR\_CONSEC\_FRAMES:**

**TOTAL += 1**

**count2 = 0**

**db.child('driver\_status/status').set('absent')**

**COUNTER = 0**

**cv2.putText(frame, "Blinks: {}".format(TOTAL), (10, 30),cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)**

**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.stop()**