**MACHINE LEARNING PROJECT BASED LAB**

**ON**

**IMPLEMENTATION OF DROWSINESS DETECTION WHILE DRIVING**

**Submitted in partial fulfillment of the**

**Requirements for the award of the Degree of**

**Bachelor of Technology**

**In**

**Computer Science & Engineering**

**By**

**Batch No:** 24

|  |  |  |
| --- | --- | --- |
| **Serial No** | **Student-id** | **Sign. Of Student** |
| **1.** | **170030013** | **A.Harish** |
| **2.** | **170030282** | **D.Sai Deepika** |
| **3.** | **170031512** |  |



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(DST-FIST Sponsored Department)**

**KONERU LAKSHMAIAH EDUCATION FOUNDATION**

Green Fields, Vaddeswaram, Guntur District-522 502

**2019-2020**

**KONERU LAKSHMAIAH EDUCATION FOUNDATION**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(DST-FIST Sponsored Department)**



**CERTIFICATE**

This is to certify that this project based skilling report entitled **“Implementation of Drowsiness Detection While Driving ”** is a bona fide work done by **D.Sai Deepika (170030282), A.Harish (170030013), (170031512)**  in partial fulfillment of the requirement for the award of degree in **BACHELOR OF TECHNOLOGY** in **Computer Science and Engineering** during the academic year 2019-2020.

**Faculty in Charge Head of the Department**

**KONERU LAKSHMAIAH EDUCATION FOUNDATION**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(DST-FIST Sponsored Department)**



**DECLARATION**

We hereby declare that this project based report entitled “Implementation of Drowsiness Detection While Driving” has been prepared by us in partial fulfillment of the requirement for the award of degree “**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING**” during the academic year 2019-2020.

We also declare that this project based report is of our own effort and it has not been submitted to any other university for the award of any degree.

**Date: 2th November 2019**

**Place: Vaddeswaram**

|  |  |  |
| --- | --- | --- |
| **Serial No** | **Student-id** | **Sign. Of Student** |
| **1.** | **170030013** | **A.Harish** |
| **2.** | **170030282** | **D.Sai Deepika** |
| **3.** | **170031512** |  |

**ACKNOWLEDGEMENT**

Our sincere thanks to Mrs. Prof K. **Swarna** for rendering outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **Mr.V. Harikiran,** Head of the Department for Computer Science and Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project work.

We would like to place on record the deep sense of gratitude to the honorable Vice Chancellor, K L University for providing the necessary facilities to carry the concluded project work.

Last but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially our classmates and our friends for their support in the completion of our project work.

**By:**

|  |  |  |
| --- | --- | --- |
| **Serial No** | **Student-id** | **Sign. Of Student** |
| **1.** | **170030013** | **A.Harish** |
| **2.** | **170030282** | **D.Sai Deepika** |
| **3.** | **170031512** |  |

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Title** | **Page no** | |
| 1 | Introduction | | 6 |
| 2 | Literature Survey | | 8 |
| 3 | Methodology | | 10 |
| 4 | Source Code | | 12 |
| 5 | Result and Discussion | | 14 |
| 6 | Conclusion | | 16 |
| 7 | Future Scope | | 16 |
| 8 | References | | 16 |

**INTRODUCTION**

This project is the implementation for “**Drowsiness Detection While Driving**”. In recent years, there has been growing interest in intelligent vehicles. A notable initiative on intelligent vehicles was created by the U.S. Department of Transportation with the mission of prevention of highway crashes. The ongoing intelligent vehicle research will revolutionize the way vehicles and drivers interact in the future. The US National Highway Traffic Safety Administration estimates that in the US alone approximately 100,000 crashes each year are caused primarily by driver drowsiness or fatigue. Thus incorporating automatic driver fatigue detection mechanism into vehicles may help prevent many accidents. One can use a number of different techniques for analyzing driver exhaustion.

One set of techniques places **sensors** on standard vehicle components, e.g., steering wheel, gas pedal, and analyzes the signals sent by these sensors to detect drowsiness. It is important for such techniques to be adapted to the driver, since Abut and his colleagues note that there are noticeable differences among drivers in the way they use the gas pedal. A second set of techniques focuses on **measurement of physiological signals** such as heart rate, pulse rate, and Electroencephalography (EEG). It has been reported by researchers that as the alertness level decreases EEG power of the alpha and theta bands increases. Hence providing indicators of drowsiness. However this method has drawbacks in terms of practicality since it requires a person to wear an EEG cap while driving.

A third set of solutions focuses on **computer vision** systems that can detect and recognize the facial motion and appearance changes occurring during drowsiness. The advantage of computer vision techniques is that they are non-invasive, and thus are more amenable to use by the general public. There are some significant previous studies about drowsiness detection using computer vision techniques. Most of the published research on computer vision approaches to detection of fatigue has focused on the analysis of blinks and head movements. In many driving situations, drivers are not even aware of their sleepiness or drowsiness prior to actually falling asleep. It has been proposed to monitor the facial characteristics of the vehicle driver, to anticipate when the driver is becoming drowsy, and to alert the driver before the driver falls asleep. However the effect of drowsiness on other facial expressions has not been studied thoroughly.

Video imaging systems have been proposed for use in vehicles to monitor the driver and/or passengers in the vehicle. Prior known driver drowsiness detection techniques have proposed processing the video images from the cameras to determine a precise measurement of the percent of closure of both eyes of the driver. The percent of eye closure is then used to determine if the driver has become drowsy.

For example, such approaches may monitor the eyelid position of each eye and determine a driver drowsiness condition based when the eyes of the driver are greater than or equal to eighty percent (80%) closure. While the aforementioned proposed technique is able to use the percent of closure of the eye of the driver as an indicator of driver drowsiness, such a technique is generally costly.

Accordingly, it is therefore desirable to provide for an alternative low-cost driver drowsiness detection system for detecting a driver drowsy condition, particularly for use in a vehicle.

The present invention provides for a low-cost system for detecting a drowsy condition by monitoring a person's eye. The system includes a video imaging camera oriented to generate images of a person, including an eye. The system also includes a processor for processing the images generated by the video imaging camera.

The processor monitors the acquired image and determines whether the eye is in one of an open position and a closed position. The processor further determines a time proportion of eye closure as the proportion of a time interval that the eye is in the closed position, and determines a drowsiness condition when the time proportion exceeds a threshold value.

According to one aspect of the present invention, the camera is located in a vehicle for monitoring the eye of the driver of the vehicle, and the system determines a driver drowsy condition. According to a further aspect of the present invention, the processor may further output a signal indicative of the determined driver drowsiness condition so as to initiate a countermeasure, such as provide a visual or audible alarm in the vehicle so as to mitigate the driver drowsy condition.

In this presents a design of a unique solution for detecting driver drowsiness state in real time, based on eye conditions. The system tends to use a web camera to capture a series of images. These captured images may further be stored as individual frames in our system. The frames so formed, are provided as input to face detection software. In terms, our required feature (eye) is extracted from the image. Individually working on each eye, the system establishes a condition and suggests a specific number of frames with the same eye condition that may be registered. The result of these images may be taken as input to obtain the level of drowsiness that a driver may encounter at any certain stage while driving a vehicle.

Our drowsiness detector hinged on two important computer vision techniques:

1. Facial landmark detection 2. Eye aspect ratio

[Facial landmark prediction](https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/) is the process of localizing key facial structures on a face, including the eyes, eyebrows, nose, mouth, and jawline.

Once we have our eye regions, we can apply the eye aspect ratio to determine if the eyes are closed. If the eyes have been closed for a sufficiently long enough period of time, we can assume the user is at risk of falling asleep and sound an alarm to grab their attention.

**LITERATURE SURVEY**

Drivers fatigue is a significant factor in a large number of vehicle accidents. Recent statistics estimate that annually 1,200 deaths and 76,000 injuries can be attributed to fatigue related crashes. The development of technologies for detecting or preventing drowsiness at the wheel is a major challenge in the field of accident avoidance systems. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its effects. The focus will be placed on monitoring the open or closed state of the driver’s eyes in real-time and the yawing state. By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident.

Detection of fatigue involves a sequence of images of a face, and the observation of eye movements and blink patterns. Once the position of the eyes is located, the system is designed to determine and detect fatigue.

− Background Elimination

− Face Detection

Face detections concerned with finding whether or not there are any faces in gray scale images and if present then returns the image location and content of each face. Face detection module was developed for single images but its performance can be further improved if a video stream is available.

Real time visions modules have been facilitated due to advances in computing technologies. These modules interact with humans. The face detection is challenging as it needs account for all possible appearance variations caused by changes in illuminations, facial features, occlusions, etc. It also has to detect faces that appear at different scale pose with in-plane rotations. The Face Detection can be of two types:

1) We want to find one particular person for a large database. In this type of search the system searches through the database and result is the most closely matched template. This operation may take time so it need not be done in real time.

2) We need to survey a particular area. Here we need rapid classification and identification i.e. the data needs to be identified in real time data. Here the continuous video stream is converted into frames for recognition. Algorithms used for Face Detection are Geometrical features, Eigen faces, Template Matching, Graph Matching, etc.

* GEOMETRICAL FEATURES

Geometric features are features of objects constructed by a set of geometric elements like points, lines, curves or surfaces.

These features can be corner features, edge features, Blobs, Ridges, salient points image texture and so on, which can be detected by feature detection methods. Geometric features are:

A. Fiducial Points

For geometric approach a set of fiducial points or anthropometrical points are used. 37 such points are found out. Some points need to be extracted manually and some are detected automatically. After point detection these coordinates may be corrected manually to improve their location. These coordinates may be stored with the corresponding image in the database.

B. Feature Choosing

Geometric features maybe presented by segments, perimeters and area of some figures formed by the detected points. 15 such segments are formed from the coordinates.

C. The Feature Set Optimization

Once feature set are obtained it is optimized by the presenting technique. The point is to find the feature space with the maximum distance between the clusters and minimum ones between the patterns of one cluster.

D. Face Recognition based on the Features

The features are stored together with a person identification photograph in database. When the tested image normalized on the rotation, scale and intensity levels fiducial points are detected and the values of the features are calculated. All the images stored in database are the patterns in the feature face.

* EYE DETECTION

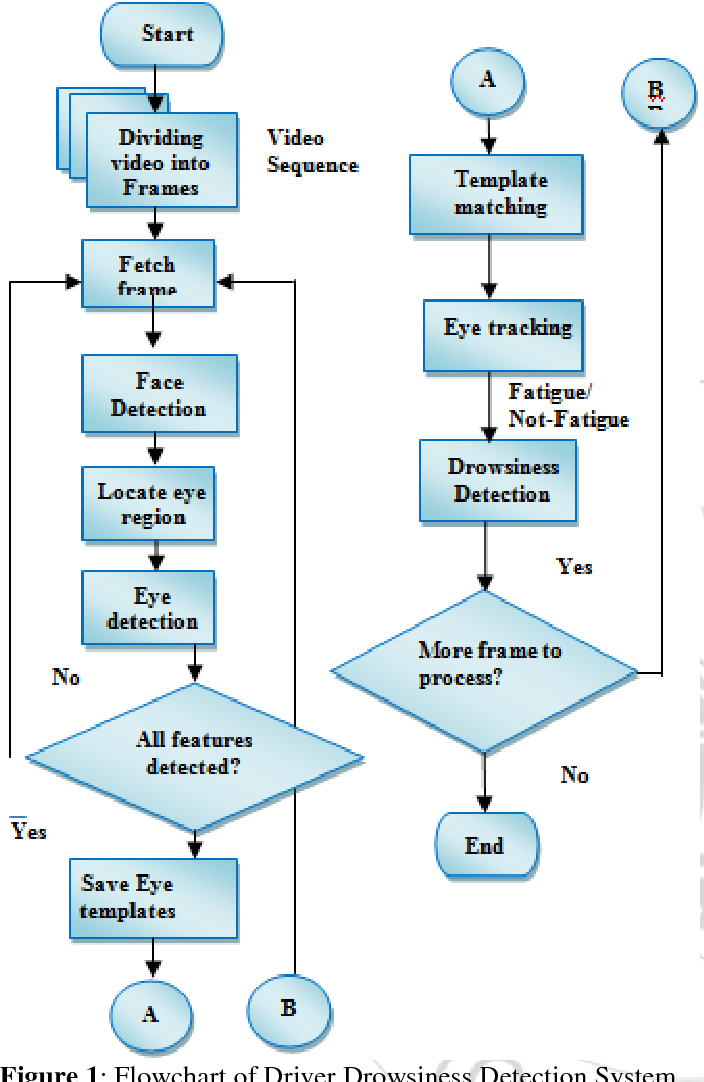
Depending upon the source light point of view there are two approaches namely based on ambient or infrared light. All of them search for characteristics of the eye. There are some algorithms that search for features like blackest pixels in the image, pixels that correspond to pupil or iris and are known as feature based algorithms. Other algorithms are trying to best fit a model to the pupil/iris contour and are known as model based algorithms. In featured based algorithms the features required are isolated from the entire image. This has an advantage of low computing resources. On the other hand, model-based approaches do not explicitly detect features but rather find the best fitting model that is consistent with the image.

Starburst algorithm is a robust eye-tracking algorithm that extracts the location of the pupil center and the corneal reflection so as to relate the vector difference between these measures to coordinates in the scene image.

The ETAR algorithm has a feature based approaches. The algorithm continues with determination of an optimal binary segmentation threshold.

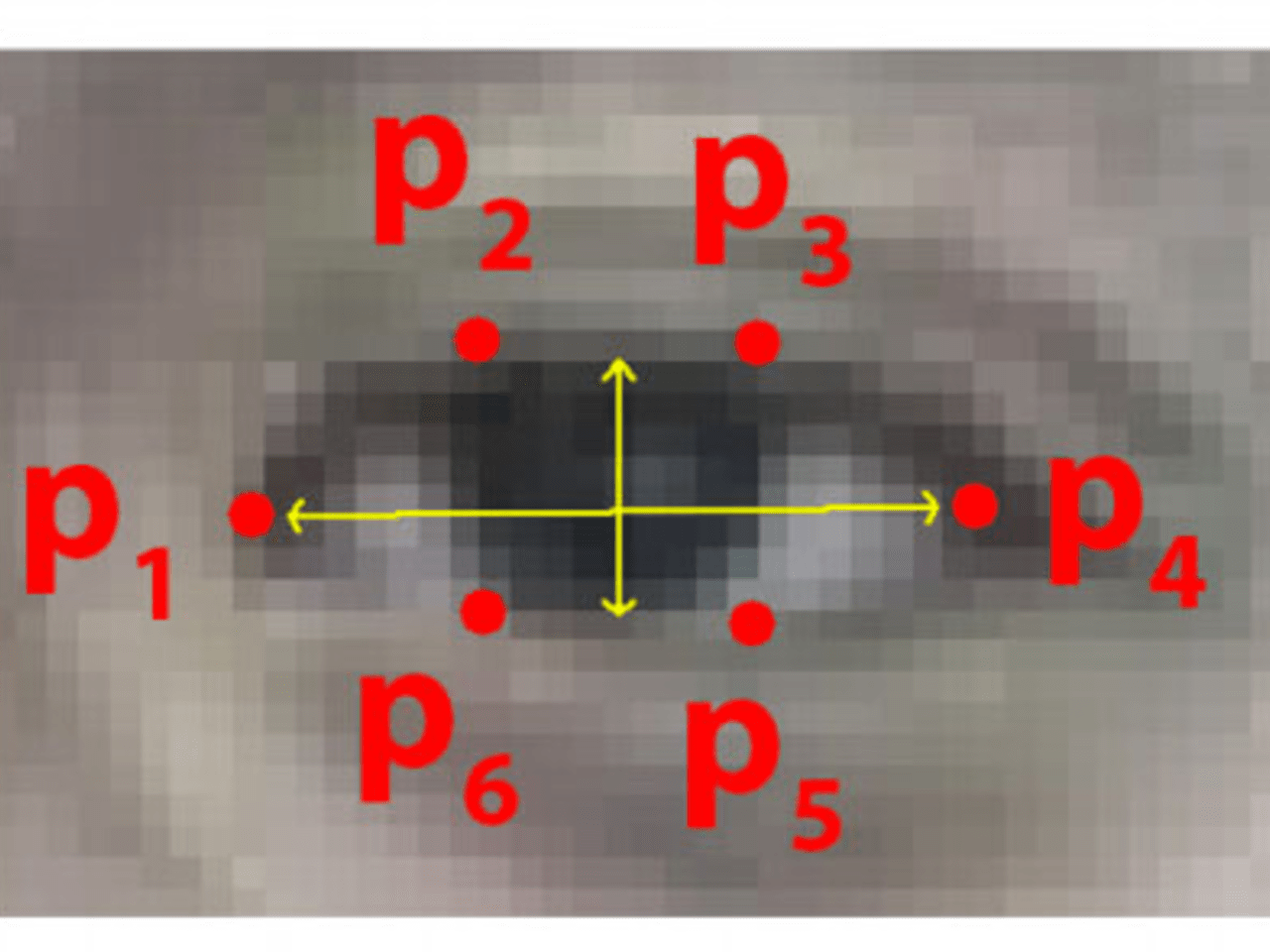
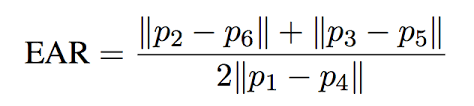
**METHODOLOGY:**

The general flow of our drowsiness detection algorithm is fairly straightforward. First, we’ll setup a camera that monitors a stream for faces. If a face is found, we apply facial landmark detection and extract the eye regions. Now that we have the eye regions, we can compute the eye aspect ratio, to determine if the eyes are closed. If the eye aspect ratio indicates that the eyes have been closed for a sufficiently long enough amount of time, we’ll sound an alarm to wake up the driver. To start our implementation, open up a new file, name it drowsiness\_detection.ipynb.



We’ll need the [SciPy](https://www.scipy.org/) package so we can compute the Euclidean distance between facial landmarks points in the eye aspect ratio calculation. We’ll also need the [imutils package](https://github.com/jrosebr1/imutils), my series of computer vision and image processing functions to make working with OpenCV easier. To detect and localize facial landmarks we’ll need the [dlib library](http://dlib.net/).

We also need to define the eye\_aspect\_ratio function which is used to compute the ratio of distances between the vertical eye landmarks and the distances between the horizontal eye landmarks. The return value of the eye aspect ratio will be approximately constant when the eye is open. The values will then rapid decrease towards zero during a blink.

If the eye is closed, the eye aspect ratio will again remain approximately constant, but will be much smaller than the ratio when the eye is open. As we can see, the eye aspect ratio is constant (indicating the eye is open), then rapidly drops to zero, then increases again, indicating a blink has taken place. We’ll be monitoring the eye aspect ratio to see if the value falls but does not increase again, thus implying that the person has closed their eyes.

If the eye aspect ratio falls below this threshold (THRESH), we’ll start counting the number of frames the person has closed their eyes for. If the number of frames the person has closed their eyes in FRAME\_CHECK, we’ll sound an alarm. You can make the drowsiness detector more sensitive by decreasing the FRAME\_CHECK similarly; you can make the drowsiness detector less sensitive by increasing it. The FLAG checks the total number of consecutive frames where the eye aspect ratio is below THRESH.

Then on we start looping over frames in our video stream. cap.read () reads the next frame, which we then pre-process by resizing it to have a width of 450 pixels and converting it to grayscale. detect (gray, 0) applies dlib’s face detector to find and locate the face(s) in the image.

The next step is to apply facial landmark detection to localize each of the important regions of the face. We loop over each of the detected faces, we assume there is only *one* face, the driver — but we left this for loop here just in case we want to apply the technique to videos with more than oneface.

For each of the detected faces, we apply dlib’s facial landmark detector and convert the result to a NumPy array. Using NumPy array slicing we can extract the *(x, y)*-coordinates of the left and right eye, respectively. Given the *(x, y)*-coordinates for both eyes, we then compute their eye aspect ratios. We can then visualize each of the eye regions on our frame by using the cv2.DRAWCONTOURS function is often helpful when we are trying to debug our script and want to ensure that the eyes are being correctly detected and localized.

Finally, we are then ready to check to see if the person in our video stream is starting to show symptoms of drowsiness. Then we make a check to see if the eye aspect ratio is below the “blink/closed” eye threshold(THRESH). If it is, we increment FLAG, the total number of consecutive frames where the person has had their eyes closed. If FLAG exceeds FRAME\_CHECK, then we assume the person is starting to doze off and draws the text DROWSINESS ALERT!  On our frame and ring the beep sounds.

Finally, we handle the case where the eye aspect ratio is *larger* than EYE\_AR\_THRESH, indicating the eyes are open. If the eyes are open, we reset COUNTER and ensure the alarm is off. The final code block in our drowsiness detector handles displaying the output frame to our screen.

**SOURCE CODE**

from scipy.spatial import distance

from imutils import face\_utils

import imutils

import dlib

import cv2

import winsound

def eye\_aspect\_ratio(eye):

A = distance.euclidean(eye[1], eye[5])

B = distance.euclidean(eye[2], eye[4])

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

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

return ear

thresh = 0.25

frame\_check = 20

detect = dlib.get\_frontal\_face\_detector()

predict = 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"]

cap=cv2.VideoCapture(0)

flag=0

while True:

ret, frame=cap.read()

frame = imutils.resize(frame, width=450)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

subjects = detect(gray, 0)

for subject in subjects:

shape = predict(gray, subject)

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 < thresh:

flag += 1

if flag >= frame\_check:

cv2.putText(frame, "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ALERT!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", (10, 30),cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)

cv2.putText(frame, "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ALERT!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", (10,325),cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)

winsound.Beep(2000, 800)

else:

flag = 0

cv2.imshow("Frame", frame)

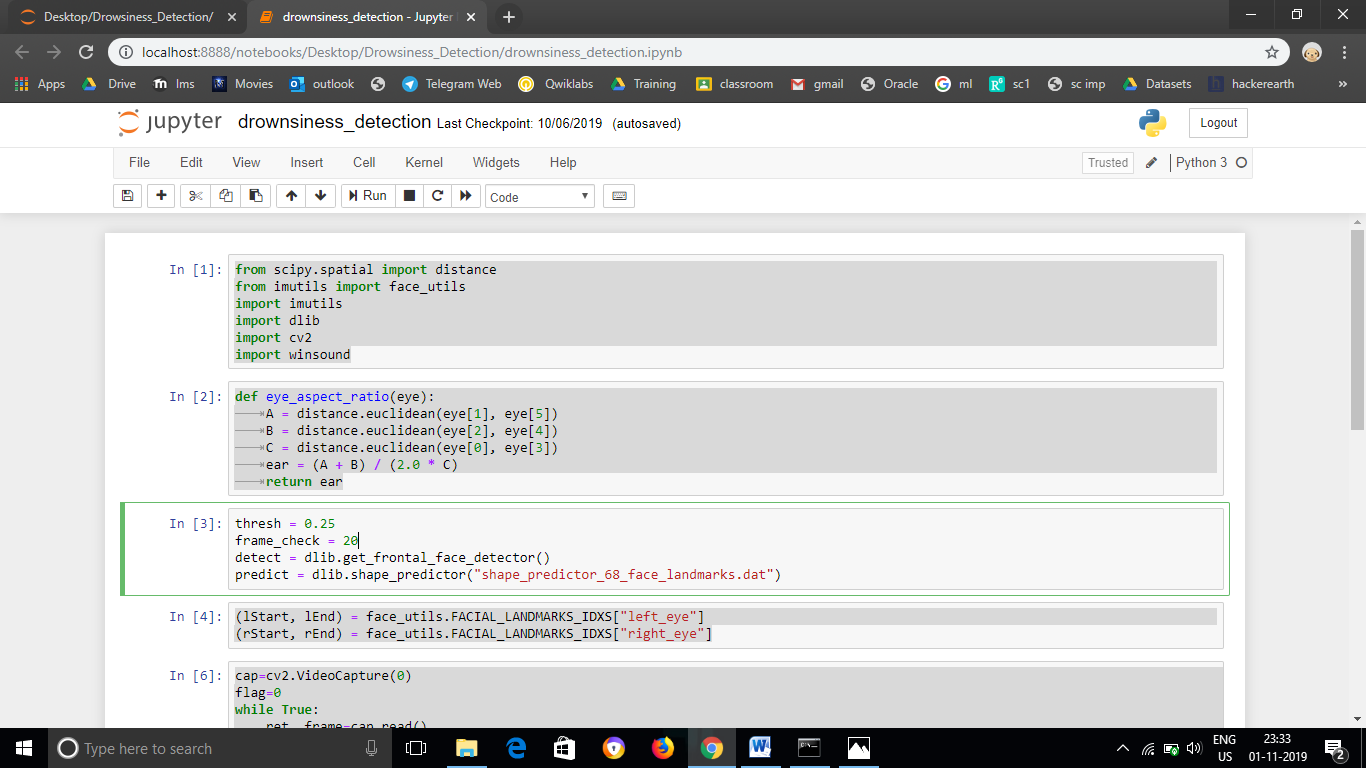
key = cv2.waitKey(1) & 0xFF

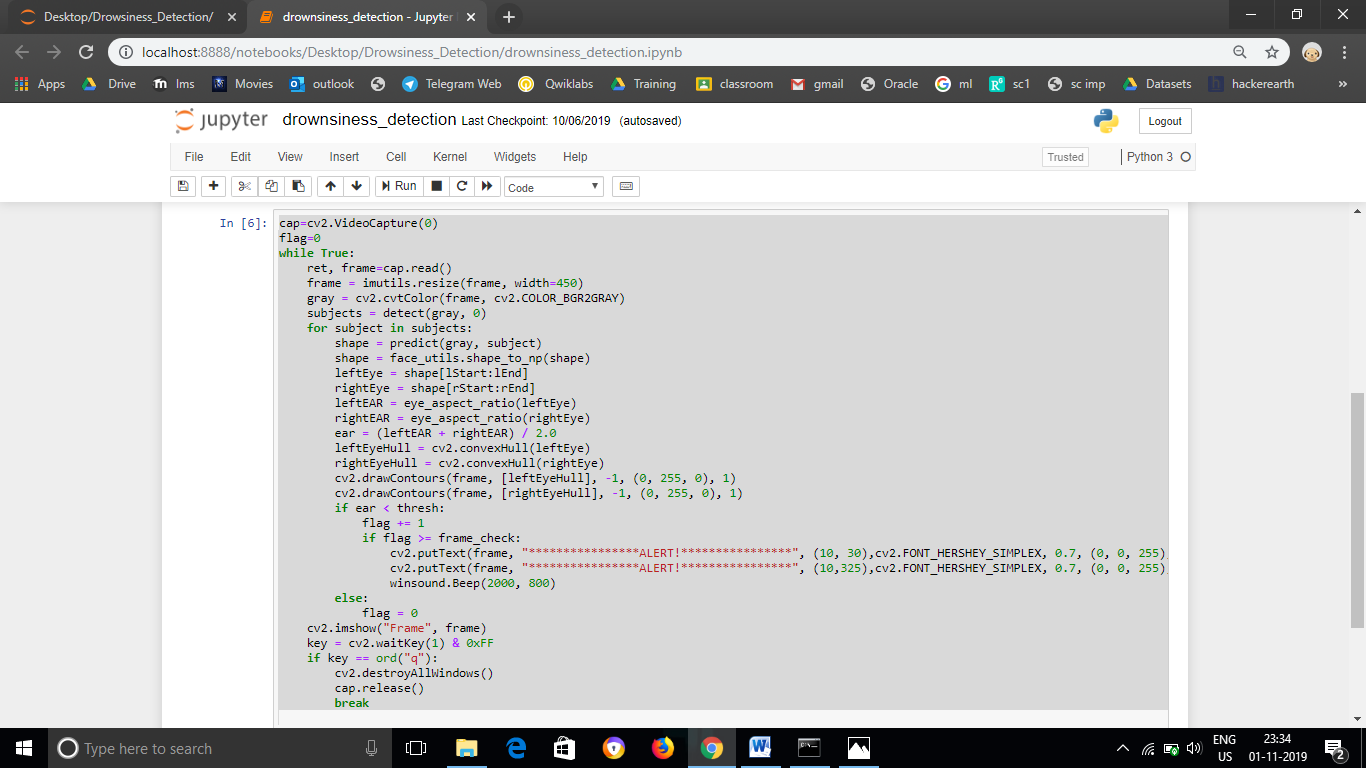
if key == ord("q"):

cv2.destroyAllWindows()

cap.release()

break





**RESULTS AND DISCUSSION**

Limitations of the proposed system are as follows: If the driver is using sunglasses then the computation doesn’t work.If there is the striking light directly on the web-camera then the system doesn’t work.

**CONCLUSION**

The drowsiness detection and correction system developed is capable of detecting drowsiness in a rapid manner. The system which can differentiate normal eye blink and drowsiness which can prevent the driver from entering the state of sleepiness while driving. The system works well even in case of drivers wearing spectacles and under low light conditions also. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for about two seconds, the alarm beeps to alert the driver and the speed of the vehicle is reduced. By doing this many accidents will reduced and provides safe life to the driver and vehicle safety. A system for driver safety and car security is presented only in the luxurious costly cars. Using drowsiness detection system, driver safety can be implemented in normal cars also.

Drowsy driving is a serious threat to drivers and traffic participants. The general flow of our drowsiness detection algorithm is fairly straightforward. A camera is setup to monitor stream of faces. After which, we apply facial landmark detection and extract the eye regions. We can compute the eye aspect ratio to determine if the eyes are closed. Video segments whose average eye state point exceeds the threshold value are detected as drowsy and the driver is alerted.

**FUTURE WORK**

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. Driver drowsiness pose a major threat to highway safety, and the problem is particularly severe for commercial motor vehicle operators. Twenty-four hour operations, high annual mileage, exposure to challenging environmental conditions, and demanding work schedules all contribute to this serious safety issue. Monitoring the driver’s state of drowsiness and vigilance and providing feedback on their condition so that they can take appropriate action is one crucial step in a series of preventive measures necessary to address this problem. 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. The vehicle manufacturers can make this system inbuilt by using the dashboard screen and speakers. The system can be effectively used in locomotives and flights for detecting driver drowsiness. System can be improved to detect and track eyes even if the driver is wearing shades.

**REFERENCES**

1. <http://www.kscst.iisc.ernet.in/spp/41_series/40S_awarded_&_selected_projs_further_devpt/40S_BE_0523.pdf>
2. <https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/>
3. <https://patents.google.com/patent/US7202792B2/en>