**CAPSTONE PROJECT REPORT**

**(Project Term July-November 2019)**

**DRIVER DROWSINESS DETECTION**

Submitted By:

**Shashank Kumar Agrahari** **Registration Number:** 11605831

Project Group Number: CSERGC0001

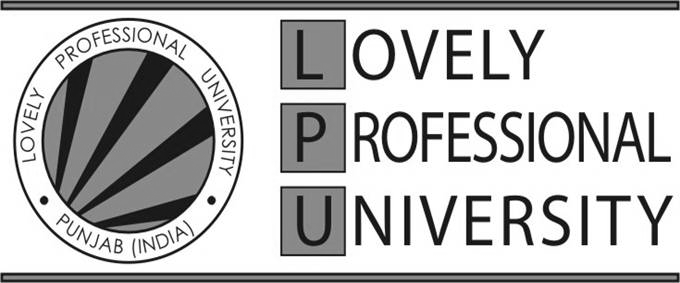
Course Code CSE439

Under the Guidance of

Mr. Chirag Sharma

(Assistant Professor)

**School of Computer Science and Engineering**





**TOPIC APPROVAL PERFORMA**

School of Computer Science and Engineering (SCSE)

**Program :** P132::B.Tech. (Computer Science & Engineering)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COURSE CODE :** | CSE439 | **REGULAR/BACKLOG :** | Regular | **GROUP NUMBER :** CSERGC0001 |
| **Supervisor Name** : | Chirag Sharma | **UID :** 16717 |  | **Designation :** Assistant Professor |

**Qualification :** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Research Experience :** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SR.NO.** | **NAME OF STUDENT** | **REGISTRATION NO** | **BATCH** | **SECTION** | **CONTACT NUMBER** |
| 1 | Shashank Kumar Agrahari | 11605831 | 2016 | K1621 | 7571969332 |

**SPECIALIZATION AREA** : Program Methodology and Design **Supervisor Signature:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROPOSED TOPIC** : Driver Drowsiness Detection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Qualitative Assessment of Proposed Topic by PAC** | | | |
| **Sr.No.** | **Parameter** | | | **Rating (out of 10)** |
| 1 | Project Novelty: Potential of the project to create new knowledge | | | 6.36 |
| 2 | Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students. | | | 7.00 |
| 3 | Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program. | | | 7.00 |
| 4 | Project Supervision: Project supervisor’s is technically competent to guide students, resolve any issues, and impart necessary skills. | | | 7.82 |
| 5 | Social Applicability: Project work intends to solve a practical problem. | | | 7.18 |
| 6 | Future Scope: Project has potential to become basis of future research work, publication or patent. | | | 7.00 |
| **PAC Committee Members** | | | | |
| PAC Member (HOD/Chairperson) Name: Gaurav Pushkarna | | UID: 11057 | Recommended (Y/N): Yes | |
| PAC Member (Allied) Name: Pradeep Kumar | | UID: 16473 | Recommended (Y/N): Yes | |
| PAC Member 3 Name: Dr. Vijay Kumar Garg | | UID: 14085 | Recommended (Y/N): Yes | |

**Final Topic Approved by PAC: Driver Drowsiness Detection**

**Overall Remarks:** Approved

**PAC CHAIRPERSON Name:** 11024::Amandeep Nagpal **Approval Date:** 29 Apr 2019

11/13/2019 11:14:23 AM

**DECLARATION**

We hereby declare that the project work entitled “DRIVER DROWSINESS DETECTION” is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr. Chirag Sharma, during July to November 2019. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: CSERGC0001

Name of Student: Shashank Kumar Agrahari

Registration Number: 11605831

(Signature of Student 1)

Date:

**CERTIFICATE**

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

**Mr. Chirag Sharma**

**Assistant Professor**

**School of Computer Science and Engineering,**

Lovely Professional University,

Phagwara, Punjab.

Date :

­­

**ACKNOWLEDGEMENT**

It would be our immense pleasure to show deep sense of gratitude and take this golden opportunity as a chance to represent our project and earn appreciation from our regarded guide and mentor **Mr. Chirag Sharma (**Dept. of Computer Science**)** for the direction, knowledge, and bolster that he has given all through this work. Our present work would never have been achievable without his guidance inputs and tutoring.

We would also like to thank every one of our friends, The Courses which helped us learn the technology and come up with such a great outcome. We feel very honored to present the Project. It is all the Hard Work of team-members who consistently worked to complete the project.

**ABSTRACT**

In the recent times there have so many cases of Accidents in the National. The National Highway Traffic Safety Administration estimates that drowsy driving is responsible for 77,000 crashes and 44, 000 injuries and 800 deaths alone in 2013. However, these numbers are underestimated as only some of the countries’ data was available for analysis. The Reports Suggest that on an Average there are 6000 deaths all over the world due to drowsy driving on the National Highway. In recent years driver fatigue is one of the major causes of vehicle accidents in the world. A direct way of measuring driver fatigue is measuring the state of the driver i.e. drowsiness. So, it is very important to detect the drowsiness of the driver to save life and property. This project is aimed towards developing a prototype of drowsiness detection system. This system is a real time system which captures image continuously and measures the state of the eye according to the specified algorithm and gives warning if required.

Though there are several methods for measuring the drowsiness but this approach is completely non-intrusive which does not affect the driver in any way, hence giving the exact condition of the driver. For detection of drowsiness the per closure value of eye is considered. So when the closure of eye exceeds a certain amount then the driver is identified to be sleepy or simply when the driver turns here and there just for ambiguous activities then the alarm is popped ahead of the driver to make him/her aware or atleast to make him sound for the condition, For implementing this system several OpenCv libraries are used including Haar-cascade. To solve this Problem first we made a Face detection using the Viola Jones Algorithm and then modified to detect the closed eyes also and added a playsound library to pop an alarm whenever needed. However, Viola jones has so many limitations regarding Ambient Light as it will only work for the Ambient light would not be a good option. so, we came up with the Height of Gradients Method to solve this Problem, using dlib Library, Which uses Boundaries of the Facial Features.

There are so many concepts hidden in Computer Vision Which we have tried to cover as much as we could have done.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **TOPIC** | **PAGE** |
| 1.  3.  4.  5. | Declaration…….……………………………………………………………  Certificate.……………………………………………………………….  Acknowledgment....……………………………………………………...  Abstract………………………………………………………………….. | (ii)  (iii)  (iv)  (v) |
| **6.** | **Chapter 1: INTRODUCTION**   * 1. Objective   2. Description   3. Scope | 9-13 |
| **7.**  **8.** | **Chapter 2: SYSTEM DESCRIPTION**   * 1. Customer/ User Profile   2. Assumptions and Dependencies   3. Functional Requirements   4. Non Functional Requirements   **Chapter 3: Design**   * 1. [Introduction](#_17dp8vu)   2. [Object detection](#_3rdcrjn)   3. [Face detection](#_26in1rg)   4. [Eye detection](#_lnxbz9) | 14-16  17-23 |
| **9.** | **Chapter 4: MATHEMATICAL CONCEPTS**   * 1. [Introduction](#_1ksv4uv)   2. [Principal Component Analysis (PCA)](#_44sinio)   3. [Eigen face approach](#_2jxsxqh)       1. Eigen value and Eigen vector      2. [Face image representation](#_z337ya) | 24-27 |
| **10.** | **Chapter 5: IMPLEMENTATION**   * 1. [Implementation](#_1ci93xb) of the project      1. Haar Features      2. [Integral Image](#_2bn6wsx)      3. [Adaboost](#_qsh70q) Classifier | 28-31 |
| **11.** | 5.1.4 Cascade Classifier  **Chapter 6** : **DLIB**  6.1 Introduction to Facial Landmarks Recognition …  6.2 Challenges Faced in Facial Landmarks Recognition… | 32-34 |
| **12.** | **Chapter 7: OpenCV** | 35 |
| **13.** | **Chapter 8: Result**  8.1 [Conclusion](#_49x2ik5)  8.2 [Future work Possible](#_147n2zr)  8.3 Gantt Chart  8.4 Code snippet  **BIBLIOGRAPHY** | 36-42  43 |

### 

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE.NO** | **TOPIC** | **PAGE** |
| 1. | Flow chart showing entire process of drowsiness detection system | 17 |
| 2. | ER Diagram………………..………………….............................. | 18 |
| 3. | Data Flow Diagram………………………………………..…...... | 19 |
| 4. | Level 1 DFD…..….…………..………………………………….. | 20 |
| 5. | Level 2 DFD………………………………………………..……. | 21 |
| 6. | Different Features of Haar……………………………..………… | 29 |
| 7. | Integral Image …...…………………………………….………… | 29 |
| 8. | Adaboost Weak Learner………………………………………..... | 30 |
| 9. | Adaboost Classifier…………………………………………….... | 31 |
| 10. | Casscade Classifier……………………………………………… | 31 |
| 11. | Dlib………………………………………………………………. | 32 |
| 12. | Eye in open state with head position straight……………...…….. | 36 |
| 13. | Eye in closed state with head position straight………………….. | 36 |
| 14. | Main Window…………………………………………………… | 38 |
| 15. | Cam and Detect Window……………………………………….. | 38 |
| 16. | Gantt Chart………………………………………………………. | 40 |
| 17. | Code Snippet…………………………………………………….. | 41 |

**Chapter 1: INTRODUCTION**

### 1.1 Objective

The Reasons Behind drowsiness can be many and some of them are low sleep or low quality sound sleep which leads to elongation of sleep and unhealthy behaviors, and the most impacting reason behind it is that if anyone drives for a very long duration for like 10 hours or more in summers or in nights, it is very much natural to feel sleepy while driving. We think sleepiness is sometimes due to the person sitting nearby, sleepy people spread their sleepy lines and creates a wanting in the driver’s intuition that he also wants to sleep. We have noticed that truck drivers also feel sleepy or drowsy while driving, the reasons are they have very tangled paths to cover and they are most of the times alone and find nothing to do alone and end up getting bored which results in sudden accidents due to drowsy driving. Several surveys on road accidents says that around 30 percent of accidents are caused by fatigue of the driver. When driver drives for more than normal period for human then excessive fatigue is caused and also results in tiredness which drives the driver to sleepy condition or loss of consciousness.

**1.2 Description**

Drowsiness is a phenomenon which reduces alert or soundness of a human. It is a biological effect which can aroused due to several reasons as discussed above, but there are few home-science tips to avoid drowsiness which may or may not help. So, we must come up with something, which can validate the problem and help us get to know about how to overcome the problem using different Technologies.

In chapter 1, in initial sections different types of methods for measuring the drowsiness of the driver are mentioned which includes Vehicle based measures, Physiological measures, Behavioral measures. Using those methods an intelligence system can be developed which would alert the driver in case drowsy condition and prevent accidents. Advantages and disadvantages corresponding to each and every system is explained. Depending on advantages and disadvantages the most suitable method is chosen and proposed. Then the approach for entire system development is explained using a flow chart which includes capturing the image in real time continuously, then dividing it into frames. Then each frames are analyzed to find face first. If a face is detected then then next task is to locate the eyes. After the positive result of detecting eye the amount of closure of eye is determined and compared with the reference values for the drowsy state eye. If drowsy condition is found out then driver is alarmed else repeatedly the loop of finding face and detecting drowsy condition is carried out.

In latter sections object detection, face detection and eye detection and eye detection is explained in detailed manner. Because face is a type of object hence a few studies on object detection is done. In face detection and eye detection different approaches for both are proposed and explained.

Theoretical base for designing the entire system is explained which includes Principal Component Analysis (PCA) and Eigenface approach. We know that the structure of face is complex and multidimensional. A face needs great calculating methods and techniques for recognizing it. In this my approach will treating a face as a two-dimensional structure and accordingly it should be recognized. Principal Component Analysis (PCA) is used for face recognition for this context. This idea involves the projection of face images onto that particular face space. Then we encode the variation or difference among the desired known faces. Eigen face decides and defines the face space. We represent these faces as eigenvectors. These vectors consists of all sets of faces. Cases of similarity with different features of our face appears like nose, Eyes, lips etc.

The Eigenface approach uses the PCA for recognition of the images. The system performs by projecting pre extracted face image onto a set of face space that represents significant variations among known face images. Eigenface approach includes EigenValues and EigenVectors, Face Image Representation, Mean and Mean Centered Images, Covariance Matrix, Eigen Face Space.

### Drowsiness

Drowsiness is defined as a decreased level of awareness portrayed by sleepiness and trouble in staying alarm but the person awakes with simple excitement by stimuli. It might be caused by an absence of rest, medicine, substance misuse, or a cerebral issue. It is mostly the result of fatigue which can be both mental and physical. Physical fatigue, or muscle weariness, is the temporary physical failure of a muscle to perform ideally. Mental fatigue is a temporary failure to keep up ideal psychological execution. The onset of mental exhaustion amid any intellectual action is progressive, and relies on an individual's psychological capacity, furthermore upon different elements, for example, lack of sleep and general well-being. Mental exhaustion has additionally been appeared to diminish physical performance. It can show as sleepiness, dormancy, or coordinated consideration weakness.

In the past years according to available data driver sleepiness has gotten to be one of the real reasons for street mishaps prompting demise and extreme physical injuries and loss of economy. A driver who falls asleep is in an edge of losing control over the vehicle prompting crash with other vehicle or stationary bodies. Keeping in mind to stop or reduce the number of accidents to a great extent the condition of sleepiness of the driver should be observed continuously.

### Measures for detection of Drowsiness

The study states that the reason for a mishap can be categorized as one of the accompanying primary classes: (1) human, (2) vehicular, and (3) surrounding factor. The driver's error represented 91% of the accidents. The other two classes of causative elements were referred to as 4% for the type of vehicle used and 5% for surrounding factors.

Several measures are available for the measurement of drowsiness which includes the following

**Vehicle based measures.**

Vehicle-based measures survey path position, which monitors the vehicle's position as it identifies with path markings, to determine driver weakness, and accumulate steering wheel movement information to characterize the fatigue from low level to high level. In many research project, researchers have used this method to detect fatigue, highlighting the continuous nature of this non-intrusive and cost-effective monitoring technique.

1. Sudden deviation of vehicle from lane position.
2. Sudden movement of steering wheels.
3. Pressure on acceleration paddles.
4. Sudden accelerated Brakes.
5. For each measure threshold values are decided which when crossed indicated that driver is drowsy.

**Advantages:**

1. It is noninvasive in nature.
2. Provides almost accurate result.

**Disadvantages:**

1. Vehicle based measures mostly affected by the geometry of road which sometimes unnecessarily activates the alarming system.
2. driving style of the current driver needs to be learned and modeled for the system to be efficient.
3. The condition like micro sleeping which mostly happens in straight highways cannot be detected.

##### Behavioral measures.

Certain behavioral changes take place during drowsing like

* 1. Yawning
  2. Amount of eye closure
  3. Eye blinking
  4. Head position

**1.3 Scope**

The Giant Automobile Industries who are currently working on it and the start-up automobile companies who aim at larger than life experiences for customers and aim at making their journeys safer by providing them an alarm system. The protype model can also work in very dim light conditions as it is usually low light conditions inside the cars, buses and trucks and even with person wearing spectacles. It is really working well according to our test cases, but in future I guess some extra features can be added to the prototype to meet the demands and need of the Scenario.

Drawbacks of the system were explained and the required future work to remove those drawbacks and to build a robust intelligent driver assistance system was emphasized. Finally, the conclusive part includes the overall performance of the proposed and implemented system. After comparing the results of the different algorithms we reached to a conclusion that Viola Jones Algorithm which also works fine but there is one more algorithm called Height of Gradients and It performs better than the latter due to some reasons as Viola Jones is Based upon Haar-like Features so it is a very ambiguous way to determine the Haar-like-features and find the optimal features among them and so on, Whereas the Height of gradients is based on the Pixel Brightness and Darkness of a Human Face and make boundaries according to that, Which is really faster and works better even in Ambient Light. It is Implemented using the dlib Library.

**Chapter 2: SYSTEM DESCRIPTION**

### 2.1 Customer/User Profile:

The Customer or User will be our Automobile Companies Who aim at better Care Service to their Customers. There are two types of customers, the first one is the Giant Companies of who are integrating such types of projects with their automobiles and the second type is a personal user who extensively travels for longer durations and wants a solution for his/her drowsiness Problems. The Companies want a very perfect working model which we can achieve by looking at different approaches available for different techniques and algorithms. There are many Strategies which are related to it, for example Viola jones algorithm, height of gradients problems, deep learning and haar-hog and linear svm.

Among all these four strategies, the most precise technique depends on human physiological measures. This procedure is executed in two ways: measuring changes in physiological signs, for example, brain waves, heart rate, and eye flickering; and measuring physical changes, for example, sagging posture, inclining of the driver's head and the open/shut conditions of the eyes. In spite of the fact that this procedure is most precise, it is not reasonable, since detecting electrodes would need to be put straightforward onto the driver's body, and thus be irritating and diverting to the driver. Also, long time driving would bring about sweat on the sensors, reducing their capacity to screen precisely.

Hence this approach will be mostly focusing on amount of eye closure also called (**PERCLOS**) percentage of closure as it provides the most accurate information on drowsiness. It is also non-intrusive in nature, hence does not affect the state of the driver and also the driver feels totally comfortable with this system. Environmental factors like road condition does not affect much to this Sy

The case of micro nap is also detected according the given threshold value. The development of this system includes face identification and tracking, detection and location of the human eye, human eye tracking, eye state detection, and driver fatigue testing. The key parts of the detection framework fused the detection and location of human eyes and driver fatigue testing. The improved technique for measuring the PERCLOS estimation of the driver was to compute the proportion of the eyes being open and shut with the aggregate number of frames for a given period.

**2.2 Assumptions and Dependencies:**

1. **State of eye:**

In this stage, we find the actual state of the eye that if it is closed or open or semi closed or open. The identification of eyes status is most important requirement. It is achieved by an algorithm which will be clarified in the later parts. We channelize a warning message if we obtain that the eyes are in open state or semi open state up to a particular threshold value. If the system detects that the eyes are open then the steps are repeated again and again until it finds a closed eye

**b. Ambient Source of Light:**

In this Stage, we must ensure that the system must receive optimum amount of light so that at least the face of the user is visible in the camera and it is able to detect the face properly and also the opening and closing states of the eyes.

**c. Hardware Requirements:**

In this Stage, we assume that we have proper and optimum resources to perform the activity. Some of the Requirements are GPU(Graphical Processing Unit), Webcam with good clarity. Some Important Libraries like dlib, OpenCV and Scipy are must.

**2.3 Functional Requirements:**

1. **Image Capture:**

For, now we are Capturing Images from our Webcam in our Laptops to take images the input type of the image is Video Streaming. Hence, when we start our Program our Webcam will start taking the video of the Person sitting in front of it, and will detect the face and then eyes of the person sitting in front of the web cam then finally, will check for the person’s eyes whether it is open or closed. Is open it will do just nothing but if the eyes are closed then it will pop an alarm before the person to make him aware of his drowsiness.

1. **Dividing into Frames:**

We are dealing with real time situation where video is recorded and has to be processed. But the processing or application of algorithm can be done only on an image. Hence the captured video has to be divided into frames for analyzing.

##### Face Detection:

In this stage we detect the region containing the face of the driver. A specified algorithm is for detection of face in every frame. By face detection we mean that locating the face in a frame or in other words finding location of facial characters through a type of technology with the use of computer. The frame may be any random frame. Only facial related structures or features are detected and all others types of objects like buildings, tree, bodies are ignored.

##### d. Eye Detection:

After successful detection of face eye needs to be detected for further processing.

In our method eye is the decision parameter for finding the state of driver. Though detection of eye may be easier to locate, but it’s really quite complicated. At this point it performs the detection of eye in the required particular region with the use of detection of several features. Generally, Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver.

* 1. **Non Functional Requirements**

**a. Response Time:**

In this Stage, we focus on Response time of the System, that is after how much time lapse would it should it take to pop the alarm before the drowsy driver in the drowsiness condition.

1. **Utilization:**

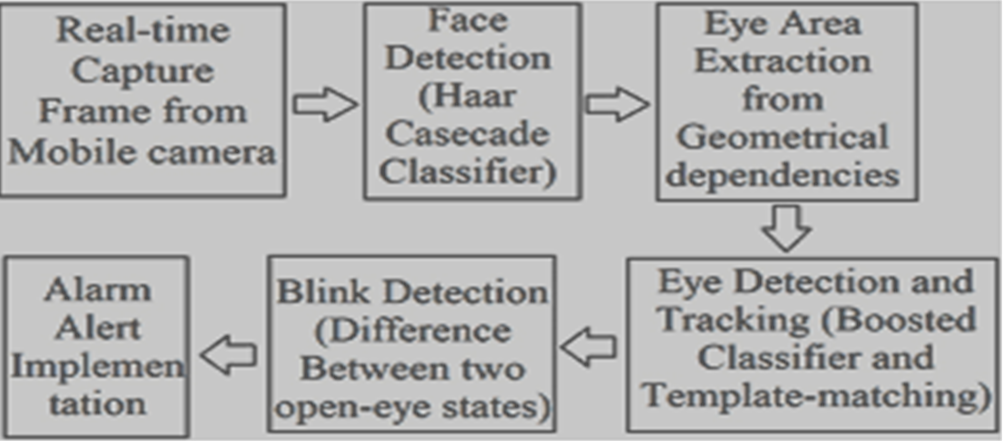
In this Stage, We focus on the Best possible Utilization of Product by Providing best Possible Specifications and requirements needed to build the System.

1. **Scalability:**

In this Stage, We focus on making the Prototype of the model Scalable that is, It must be affordable and any Customer who needs it should be able to purchase the Product with a very reasonable price.

**Chapter 3: DESIGN OF THE PROTOTYPE**

### System Design



###### Figure 1: Flow chart showing entire process of drowsiness detection system

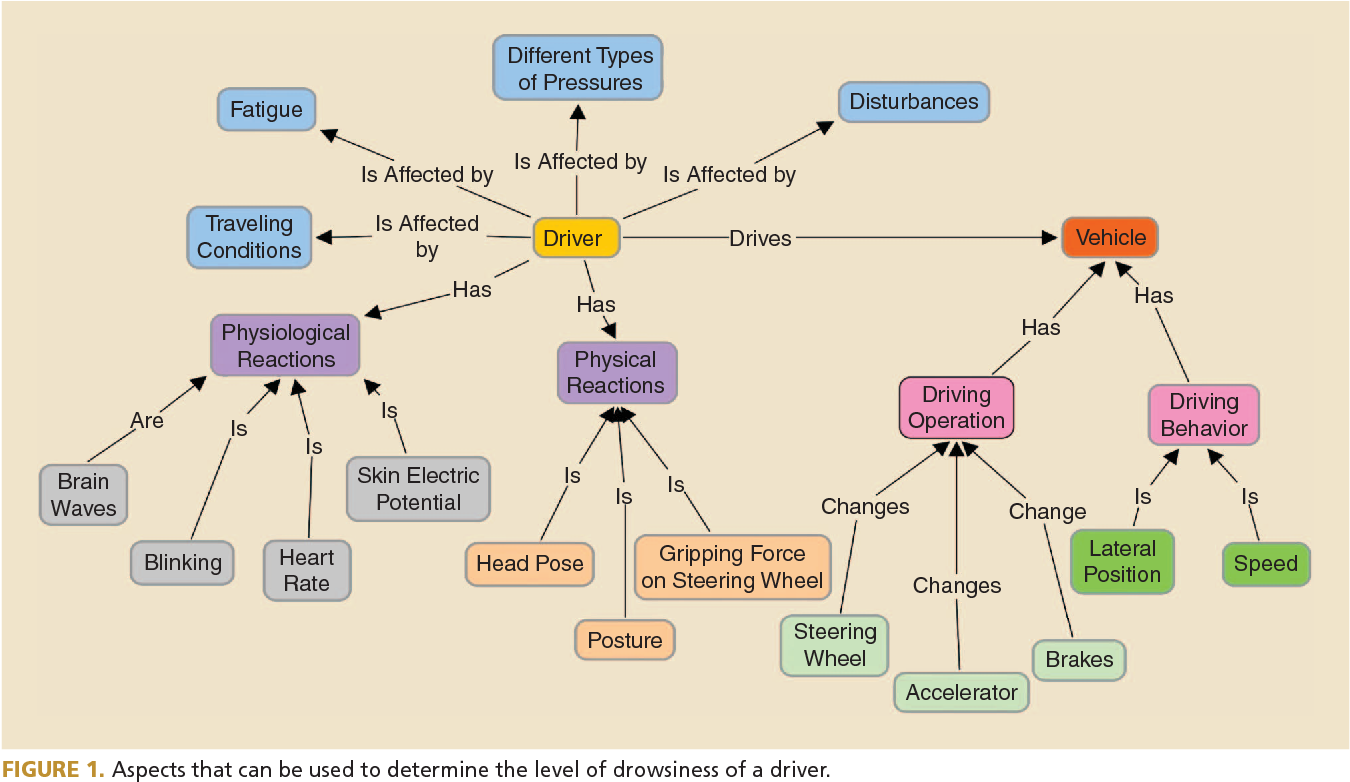
This chapter deals with System Design. The System Design here describes the details of the System and the steps involved in it. The Steps Involved are Real-time Capture Frame from Mobile Camera, Face detection, Eye Area Extraction from Geospatial Dependencies, Alarm Alert, Blink Detection, Eye Detection at last of the Step. The Procedures are defined in the above diagram.

### Use-Case Diagram

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

**ER Diagram:**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

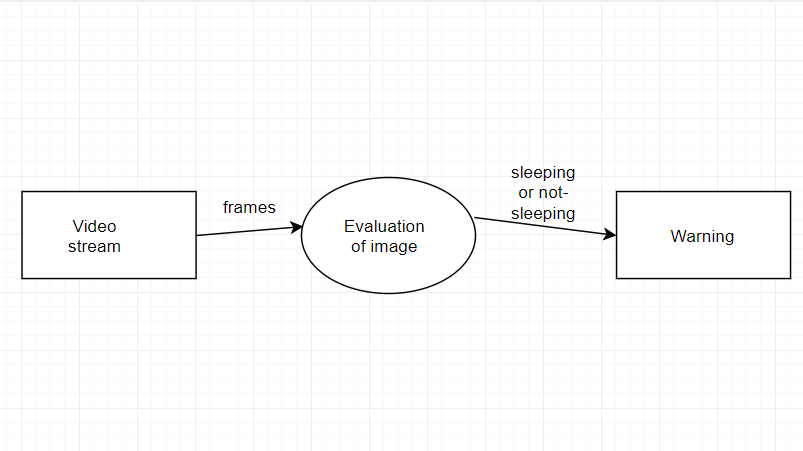


**Figure 2: ER diagram of the Driver Drowsiness Detection**

**Data Flow Diagram:**

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical

**Level 0:**



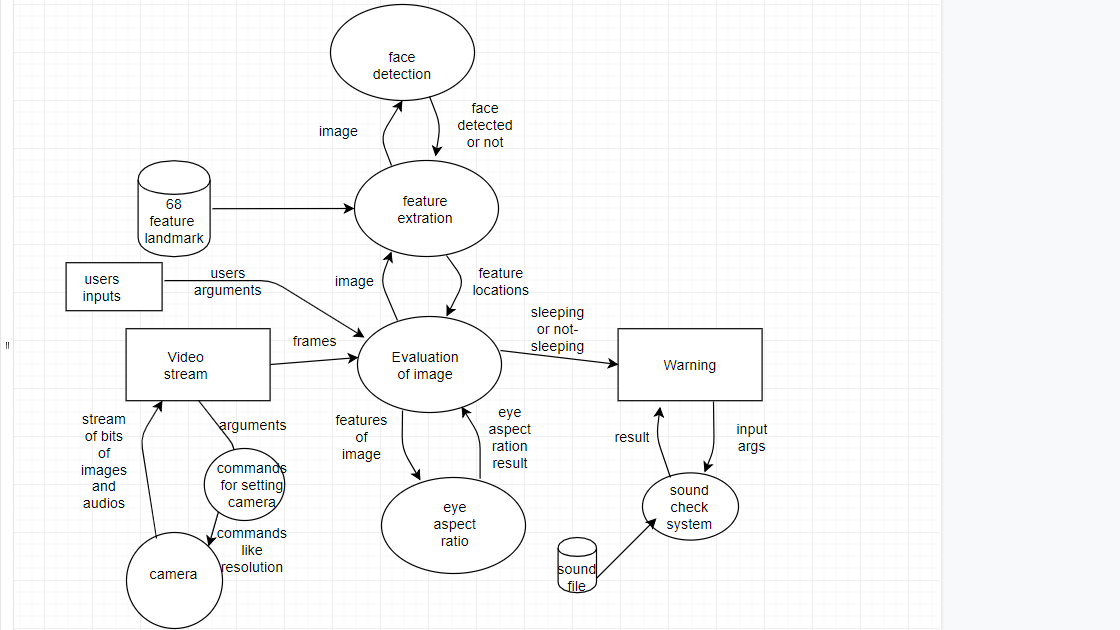
**Figure 3: Level 0 diagram of the Driver Drowsiness Detection**

**Level 1:**



**Figure 4: Level 1 diagram of the Driver Drowsiness Detection**

**Level 2:**



**Figure 5: Level 2 diagram of the Driver Drowsiness Detection**

### Face detection

We know that face is also a type of object. So we can consider detection of face as a particular case of object detection. In this type of object type of class detection, we try to know where the objects in the interest image are located and what is their size which may belongs to a particular class. The work of algorithm that is made for face detection is mostly concentrated on finding the front side of the face. But the algorithm that are developed recently focus on more general cases. For our case it may be face in the tilted position or any other portion of the faces and also it finds the possibility of multiple faces. Which means the rotation axis with respect to the present observer from the reference of face in a particular. Or even if there is vertical rotation plane then also it is able to solve the purpose. In new type of algorithm it is considered that the picture or video is a variable which means that different condition in them like hue contrast may change its variance. The amount of light may also affect. Also the position of the input may vary the output. Many calculations actualize the face-detection.

### Eye Detection

Poor contrast of eyes generally creates a lots of problems in its detection. After successful detection of face eye needs to be detected for further processing. In our method eye is the decision parameter for finding the state of driver. Though detection of eye does not look complex but the actual process is quite hectic. In this case it performs the detection of eye in the specified region with the use of feature detection. Generally Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver. Eye detection is divided into two categories: eye contour detection and eye position detection. Basically eyes are detected based on the assumption that they are darker than other part of the face. Hence Haar Features of similar type can be moved throughout the upper part of the face to match with the feature of eye leading to location of eye.

We consider as potential eye areas, the non-skin locales inside face district. Clearly, eyes ought to be inside a face area and eyes are not distinguished as skin by the skin identifier. In this way, we need to discover eye-simple sets among a decreased number of potential eye regions. In recent years several eye detection methods have been developed. Deformable template is one of the popular method in identifying the human eye. In this method, a model of eye is designed first and then eye position is obtained by recursive method. But this method strongly depends on initial position of the eye which should be near the actual position of eye. In the template matching aspect, the proposed algorithm is based on eigen features and neural networks for the extraction of eyes using rectangular fitting from gray-level face images.

This method does not need a large set of training images in its advantage and does by eigen features and sliding window. But this algorithm fails if the user uses glasses or having beard. We know that using Haar features in AdaBoost results in increasing computational efficiency and accuracy than other methods for face detection. But Haar feature has a limitation i.e. discriminant capability.

Although the Haar features vary with different patterns, sizes and positions, they can only represent the regular rectangular shapes. But for our case of eye detection eye and iris is of round shape. Hence eyes can be represented by learning discriminate features to characterize eye patterns. So an approach towards probabilistic classifier to separate eyes and non-eyes are much better option for better accuracy and for robustness.

**Chapter 4: MATHEMATICAL** **CONCEPTS**

### Introduction

Chapter 4 deals with the theoretical and mathematical explanation of the various approach made for face and eye detection. In the beginning of this chapter Principal Component Analysis (PCA) method was described. Latter through this Eigenface approach is explained which includes mathematical description of its subparts such as Eigen Values and Eigen Vectors, Face Image Representation, Mean and Mean Centered Images, Covariance Matrix and Eigen Face Space.

### Principal Component Analysis (PCA)

Principal component analysis (PCA) was invented in 1901 by Karl Pearson. If the resulted data is repeated again and again or has redundancy the PCA helps in reducing this redundancy. PCA basically removes the variables to reduce redundancy. So, after reduction of variables we will get less variables named as Principal Components. Principal components will generally represent all the variables present in the obtained variable. But it only reduction of variables does not solve the purpose. Main Problem appears when we try to achieve face recognition in a more and high dimensional space. The main objective of PCA is to decrease the no of dimension as well as retain more and more possible variation in the given data set. But we know that reduction in dimension results in information loss as information are directly linked with dimension. Hence, we can overcome the problem of data loss by choosing the best principal components as main principal components determines the low dimension. Though use of PCA has many advantages but mostly it is used for eigen face approach. In eigen face approach the reduction of size of the data base is achieved for recognizing the test images. The obtained images are stored in the data base in vector form which are also called feature vectors. And these are found out from set of Eigen face obtained by projecting it over trained image. So, basically PCA is used for Eigen face approach for the reduction of dimensionality with our causing the loss of data.

### Eigen face approach

Eigen face approach for face recognition is very efficient and helpful because of its speed of operation simplicity in using and capability of learning. In computer vision face detection is done by use of eigen face which are basically set of eigen vectors. This approach is basically an appearance-based approach which does face recognition by capturing the variation in a set of face images and this information is used for comparison and encoding of each individual face in proper manner. What we mean by eigen faces is that they are Principal components of distributed faces which are represented in the form of covariance matrix of set of faces. In this method a face image is represented in the form of one-dimensional matrix. We know we can represent a face in two-dimensional form of pixels as N x N matrix in N2 dimension space. These N x N matrices is shifted to the form of row matrix. Many works on this were already done but it has ignored the fact of face stimulus which assumes that the given predefined measurements on face recognition are important and adequate. Which means that coding and encoding of available face images probably give information of face images which point outs the important significant features. But a chance is there that the obtained features may or may not be related to the known and required facial feature like nose, eyes, lips, hair etc. So, the extraction of required information from a face image is required. After extraction is done, we encode it with high efficiency and the result is compared with a database of faces encoded in the same fashion. For this purpose, we capture the variation with a collection of face images which is a very simple approach for the extraction of the information content. The next step is to find the Principal Component of the face distribution or from the obtained covariance matrix the Eigenvectors of the set of face images can be found out. Every row of image is considered as a vector stacked one after another in a single row which helps in displaying the Eigenvector as a sort of face. A linear combination of face images is taken to represent each face images. We find that the sum total of all expected eigenfaces is decided by total number of given input images in the prepared set. An approximation can be done for faces by the use of Eigenface for those having large eigenvalues which set the most variance in in case of available set of images. To increase the computational efficiency, use of fewer Eigenface is done.

##### Eigenvalues and eigenvectors:

In linear algebra, a linear equation in matrix form is represented by Ax= D.

The eigenvectors of a linear operator are non-zero vectors which, when operated by the operator. The result of this is a scalar multiple of them. For the eigenvector X the obtained scaler called eigenvalue (λ). A vector which is paralleled by linear transformation is called an Eigenvector. It is one of the properties of matrix. When we calculate a matrix on it then the magnitude of the vector is changed. The direction of vector remains as it is. So, we define as Ax = λx, where A is represented as a vector function. Then transforming the RHS part and writing it as (A − λI)x = 0, where I is called the identity matrix. The above form is a homogeneous equation and is fundamental part of linear algebra. Existence of non-trivial solution is decided by considering that if and only if Det (A − λI) = 0, where Det represents determinant. When it is evaluated, we deal with the polynomial of degree n. This is known as the characteristic polynomial of A. If we represent the dimension of A by NxN then the solutions results in n roots of the characteristic polynomial. So it gives n Eigenvalues of A which satisfy the Axi = λixi , where i = 1,2,3,. n. If the obtained eigenvalues are all distinct then we get n associated linearly independent eigenvectors with unique directions.

##### Face Image Representation

In this approach we represent set of let’s say m images of each having size NxN. This is done by vectors of size N2. We represent each face Γ1, Γ2, Γ3… Γn. All those ₼obtained feature vectors are stored in the matrix with size NxN. One example is shown below which describes the entire process.

As we can see that the size of covariance matrix will be N2 X N2 which is huge actually and we need to find the Eigenvectors for the covariance matrix. But the large size make it time consuming and tedious. To encounter this problem we go for calculating ATA .

Now let’s consider the eigenvectors Vi of ATA such that ATA Xi = λiXi.

The eigenvectors Vi of ATA are X1… Xn2. Now for simplifying we multiply the above equation with A both sides and we get

From above we clearly see that Eigenvectors responding to AAT is now firmly computed by reduction in dimension where AXi is the Eigenvector and λi is the Eigenvalue.

##### Eigen face space

Let’s say we have a covariance matrix AAT. So the eigenvectors corresponding to that matrix which is denoted by Ui where Ui represents facial images. Those eigenfaces basically look like ghostly. Only those eigenvectors are accepted which corresponds to Eigenface in the face space and discarded faces are faces are having eigenvalues zero. This method helps to reduce the Eigen face to a great extent. Rank of Eigen faces are decided according to their usefulness to characterize the variation among the images. Here we project the face image into the face space by Ωk = UT(Γk− Ψ); k=1,. ,M, where (ΓkΨ) is the mean centered image. Hence we obtain the projection of each image as Ω1 for projection of image1 and Ω2 for projection of image2 and hence forth.

**Chapter 5:** **IMPLEMENTATION**

The implementation of the drowsiness detection system with our laptop’s WebCam, As it is just a prototype Model we did not integrate any Raspberry-pi or Camera to make it Product but, it can be easily converted into a Product and installed into systems of trucks and Cars. Giant Automobile Companies like BMW and Mercedes have already jumped into the space of Drowsiness detection as they conclude customers as their most precious resources and care about their customer’s Safety. 30% of all The Accidents on Highways are mostly because of Drowsiness of the Driver followed by many Uncertain reasons which cannot be avoided but we can avoid the Accidents by popping the alarms whenever the camera detects a perclos(condition when the eyes of the driver gets closed or seems to be dull), by this the driver may get alarmed about his/her situation can save himself/herself from such kind of hazardous situations.

### Implementation of the Project

The face detection method used in OpenCv is developed in 2001 by Paul Viola and Michael Jones, very well referred to as the Viola-Jones method. Though this method can be used for several objects but most specifically here it is used for face and eye detection in real time.

**Viola-Jones algorithm has four stages:**

**(i)** Haar like Features

(ii) Integral Image

(iii) AdaBoost Classifier

(iv) Cascade Classifier

#### Haar features

Haar-like featuresare digital image features used in object detection. Or we can say that these are rectangle shaped dark and light areas having similar kind of features like our face. So, basically we move those features throughout our face to find the output of each feature.

1. The eyes region is darker than the upper-cheeks.
2. nose bridge region is brighter than the eyes.
3. The eyebrows are alike in all the faces
4. The lower lips are darker than upper lips when in grayscale.

So this features of face are used for developing haar like features. Each feature is

related to a special location in the face.

Output of Rectangle features:

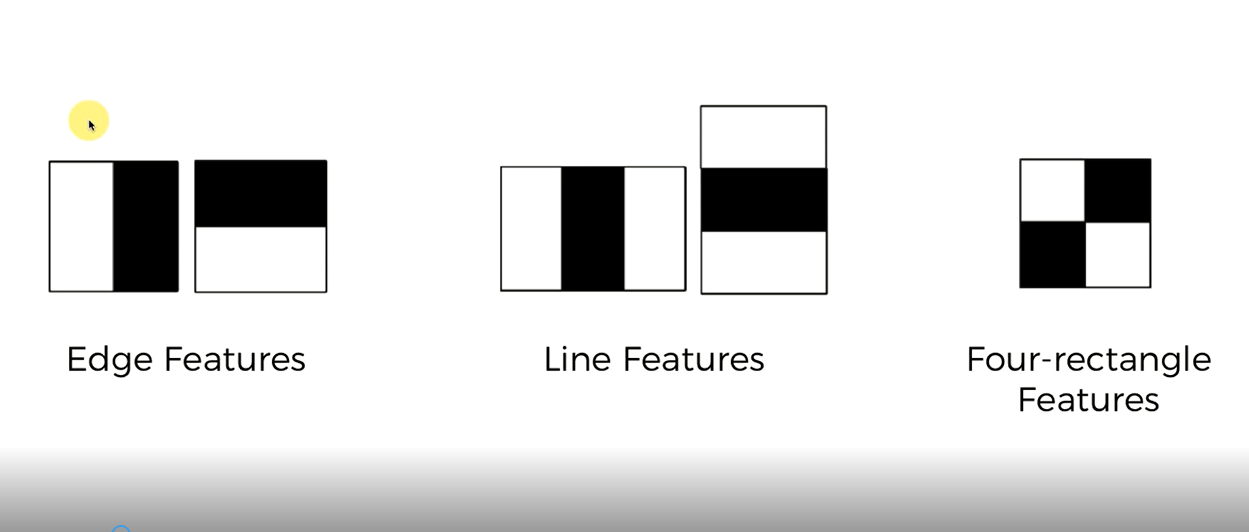
We will move the related kind of rectangle throughout the face to get different

values.

Value = ∑ (pixels in black area) - ∑ (pixels in white area).

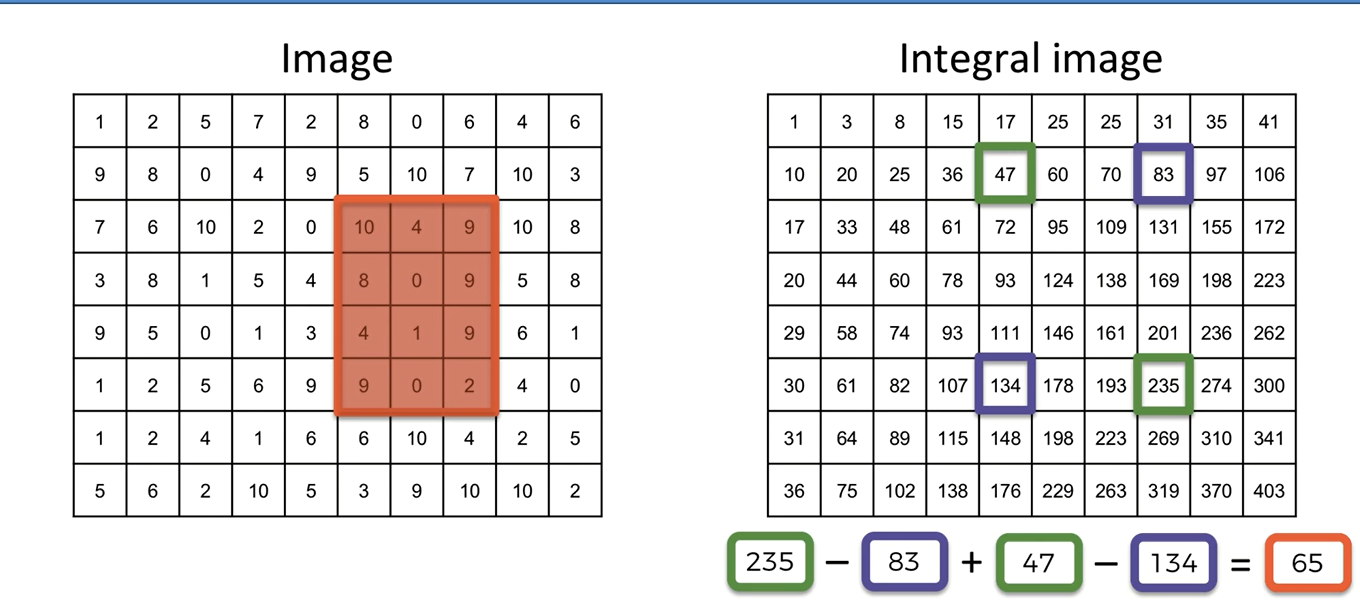
Three types: two-, three-, four-rectangles, Viola and Jones used two-rectangle

features.



###### Figure 6: Different features used for Haar cascade

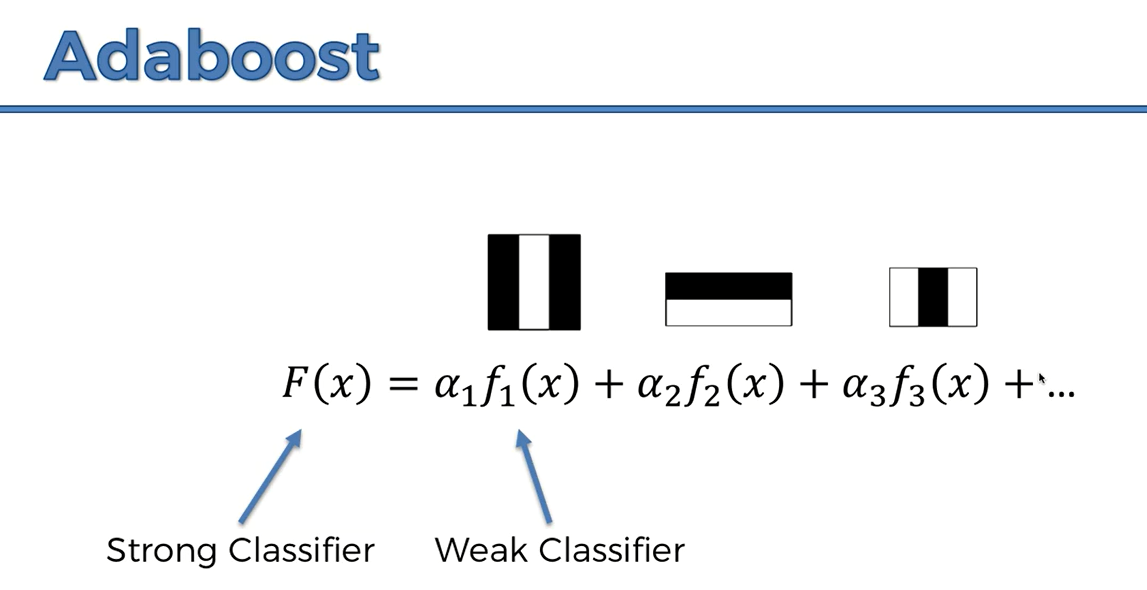
* + 1. **Integral Image**



###### Figure 7: figure shows integral image formation

Here the pixel value of each box is modified by sum of all those box left and above it so that we can use a formula mentioned below to get the output of Haar features with much less calculation reducing the time of calculation.

* + 1. **AdaBoost Classifier**

Adaboost stands for “Adaptive” boost. Here we construct a strong classifier as linear combination of weak classifier as there are so many features which are absolutely invalid in finding the facial features. It can be for.





###### Figure 8: figure shows Adaboost Classifier (Weak Learners)



**Features of weak classifiers:**

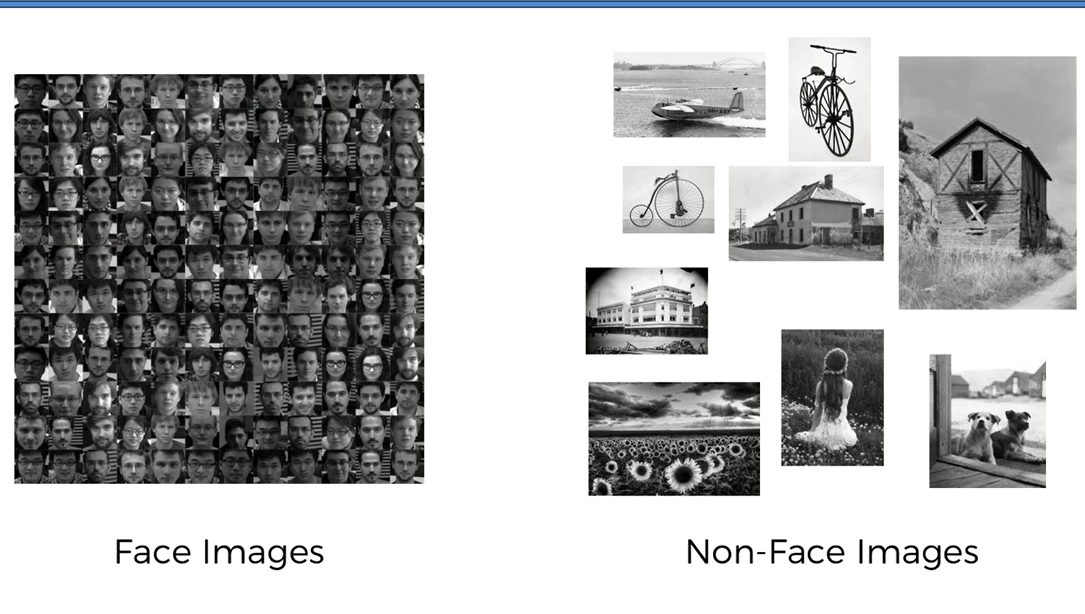
Each single rectangle is regarded as a simple weak classifier. Each weak classifier is assigned a weight function as per its importance of position. Finally, the strong classifier is formed by their linear combination. When we use haar like features we end up having millions of features then we have to choose some four to five weak features or learners that when combined produce optimum results.



###### Figure 9: figure shows Adaboost

#### Cascade Classifiers

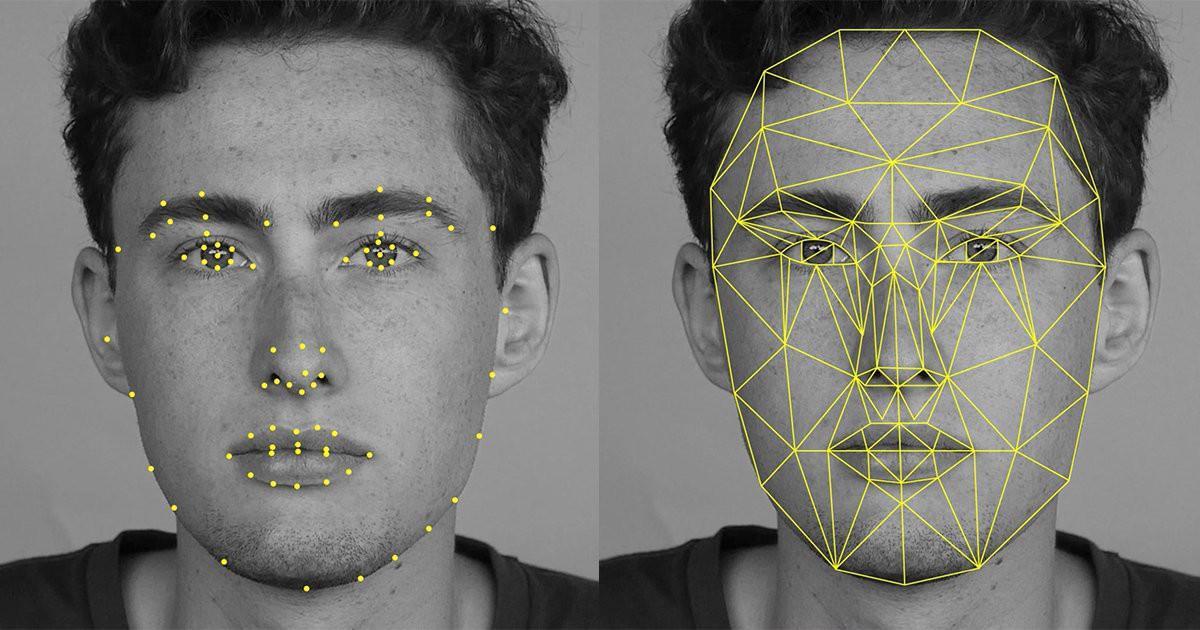
After going through Adaboost stage now let’s say we have 600 no of strong classifiers. So to detect if a frame contains a face or not: Instead of applying all the 600 features on a window, group the features into different stages of classifiers and apply one-by-one. If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.



###### Figure 10: figure shows Cascade Classifier

**Chapter 6: Introduction to Dlib Library**

Dlib is a general purpose [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) software [library](https://en.wikipedia.org/wiki/Library_(computing)) written in the programming language [C++](https://en.wikipedia.org/wiki/C%2B%2B). Its design is heavily influenced by ideas from [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [component-based software engineering](https://en.wikipedia.org/wiki/Component-based_software_engineering). Thus it is, first and foremost, a set of independent software components. It is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) released under a [Boost Software License](https://en.wikipedia.org/wiki/Boost_(C%2B%2B_libraries)#License).



###### Figure 11: figure shows dlib

**6.1 Introduction to Facial Landmarks**

A landmark is a recognizable natural or man-made feature used for navigation feature that stands out from its near environment and is often visible from long distances. Facial landmarks is defined as the detection and localization of certain key points points on the face which have an impact on subsequent task focused on the face, like animation, face recognition, gaze detection, face tracking, expression recognition, gesture understanding etc. Facial landmark are a prominent feature that can play a discriminative role or can serve as anchor points on a face graph. Facial landmarks are the nose tip, eyes corners, chin, mouth corners, nostril corners, eyebrow arcs, ear lobes etc. For ease of analysis most landmark detection algorithm prefers an entire facial semantic region, such as the whole region of a mouth, the region of the nose, eyes, eyebrows, cheek or chin. The facial landmarks are classified in two groups, primary and secondary , or fiducial and ancillary. This distinction is based on reliability of image features detection techniques. For example, the corners of the mouth, of the eyes, the nose tips and eyebrows can detected relatively easily by using low level image features, e.g. SIFT, HOG. The directly detected landmarks are referred as fiducial. The fiducial group of landmarks and they play a more prominent role in facial identity and face tracking. The search for secondary landmarks is guided by primary landmarks. The secondary landmarks are chin, cheek contours, eyebrow and lips midpoints, non extremity points, nostrils. It takes more prominent role in facial expression.

**6.2 Challenges of Landmark Localization:**

Despite the conceptual simplicity of facial landmarks detection, In computer vision there are some challenges. The emerging applications like surveillance system, gesture recognition requires that landmark localization algorithms should run in real time parallel with the computational power of an embedded system, such as intelligent cameras. Such type of application requires a more robust algorithms against a confounding factors such as illumination effects, expression and out of plane pose. There are four main challenges in localizing facial landmarks are as follows: Variability: Landmark appearances differ due to extrinsic factors such as partial occlusion, pose, illumination, camera resolution and expression, also due to intrinsic factors such as face variability between individuals. Facial landmarks can sometimes be only partially observed due to hand movements or self-occlusion due to extensive head rotations or occlusions of hair. Also facial landmark detections are difficult because of illumination artifacts and facial expressions. A facial landmark localization algorithm that delivers the target points in a time in an efficient manner and works well across all intrinsic variations of faces has not yet been feasible. Accuracy and number of landmarks require: Based on the intended application the number of landmarks and its accuracy varies. For example, In face recognition or in face detection tasks, primary landmarks like two mouth corner, four eyes corner and nose tips may be adequate. On the other hand, higher level tasks face animation or facial expression understanding require greater number of landmarks e.g. from 20-30 to 60 80 with higher accuracy. Fiducial landmarks are need to be determine with more accuracy because they often guide the search of secondary landmarks. Lack of globally accepted and error free dataset: Most of the dataset provides annotations with different markups and accuracy of their fiducial point is questionable. The accuracy of landmark localization algorithm is largely depend on the data set used for training. Each algorithm uses different dataset to train and evaluate performance so it is difficult to compare algorithms. Acquisition conditions: Acquisition conditions, such as resolution, background clutter, illumination can affect the landmark localization performance. The landmark localizers trained in one database have usually inferior performance when tested on another database.

**Chapter 7:** **Introduction to OpenCV Library**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV 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, OpenCV makes it easy for businesses to utilize and modify the code.

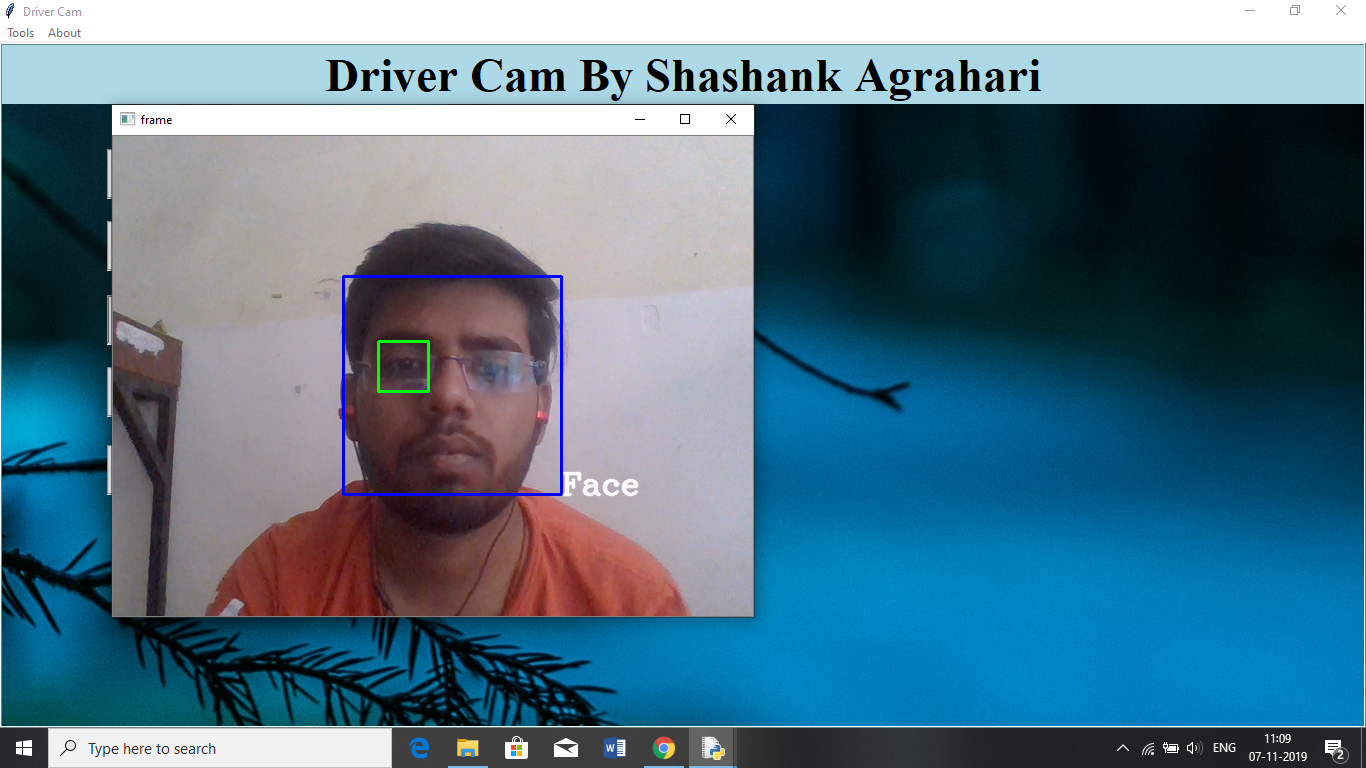
The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, 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 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

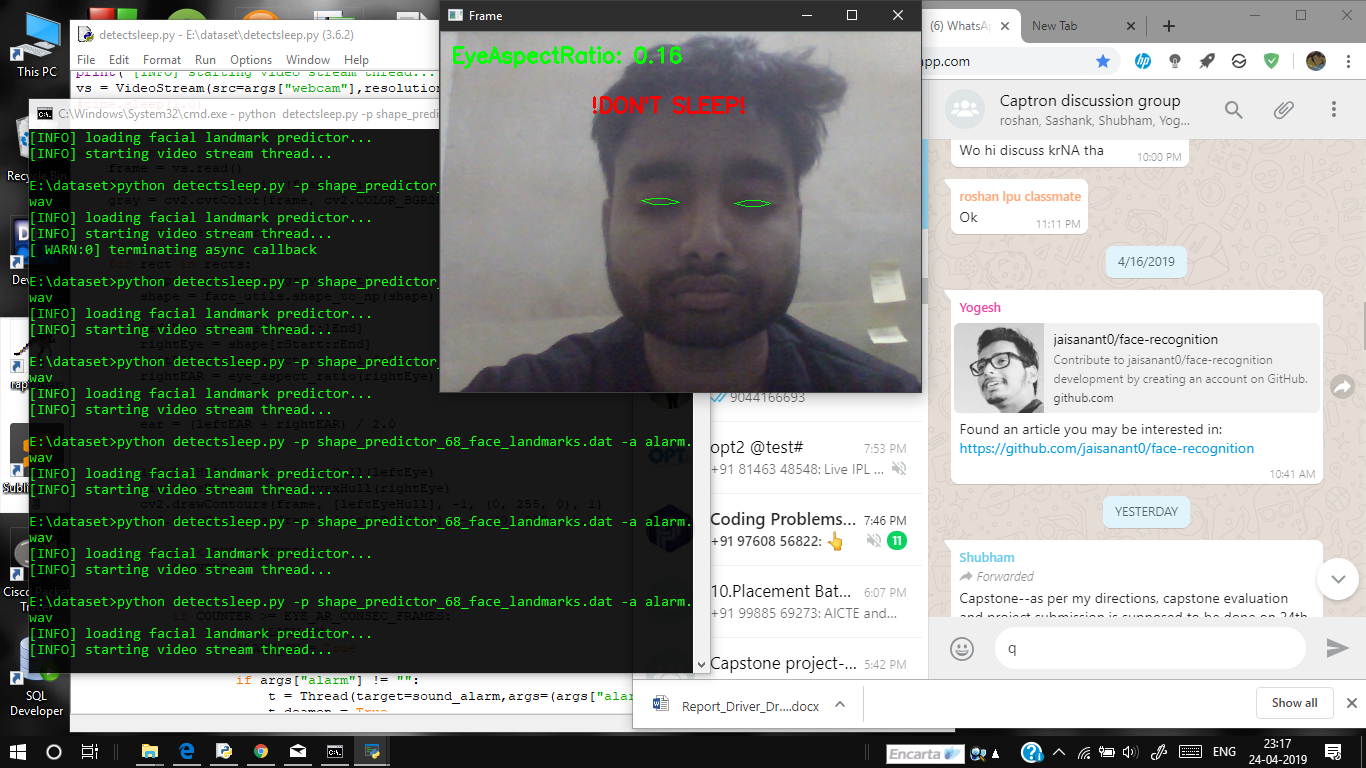
It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUD Aand OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a template interface that works seamlessly with STL containers.

**Chapter 8: RESULTS**

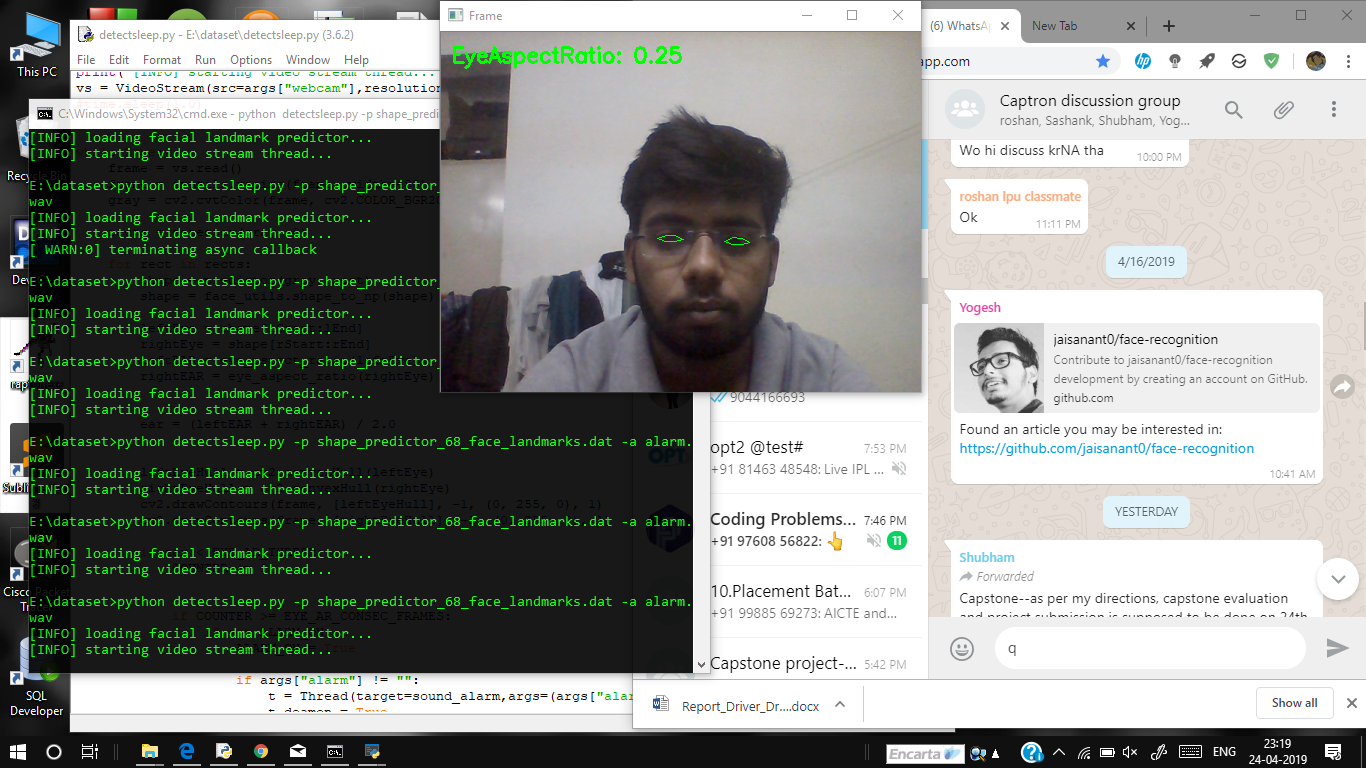
We all have taken some tests to check whether our prototype is working good or not, one by one all the members of our group came together with a different pose to add adversariality to the results. The best thing is that, It even works with deem light conditions and person with spects. After we all took tests, the prototype model was accurately and precisely detected the closed state of the eyes and popped up the alarm on time.

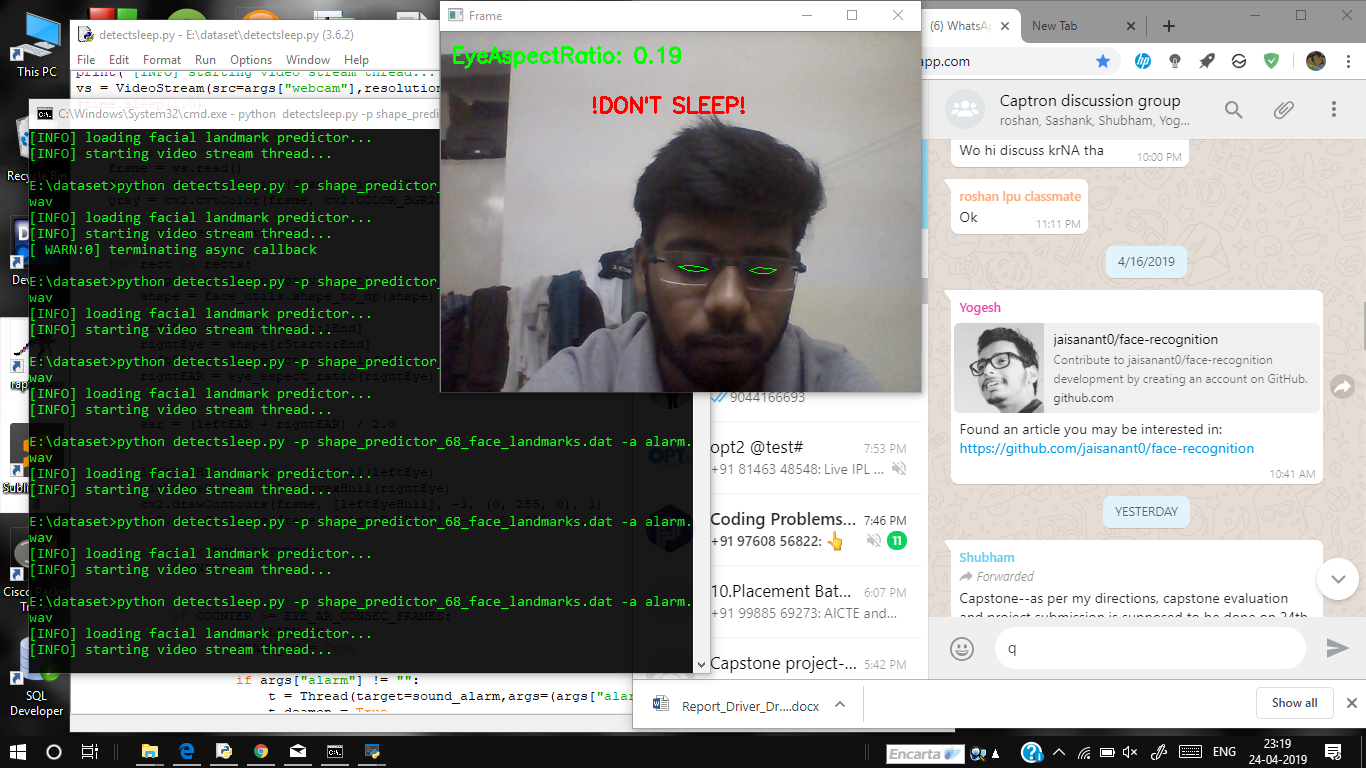


###### Figure 12: Eye in open state with head position straight.



**Figure 13: Eye in closed state with head position straight**

 **Figure 14: Eye in open state with head position straight and with glass**

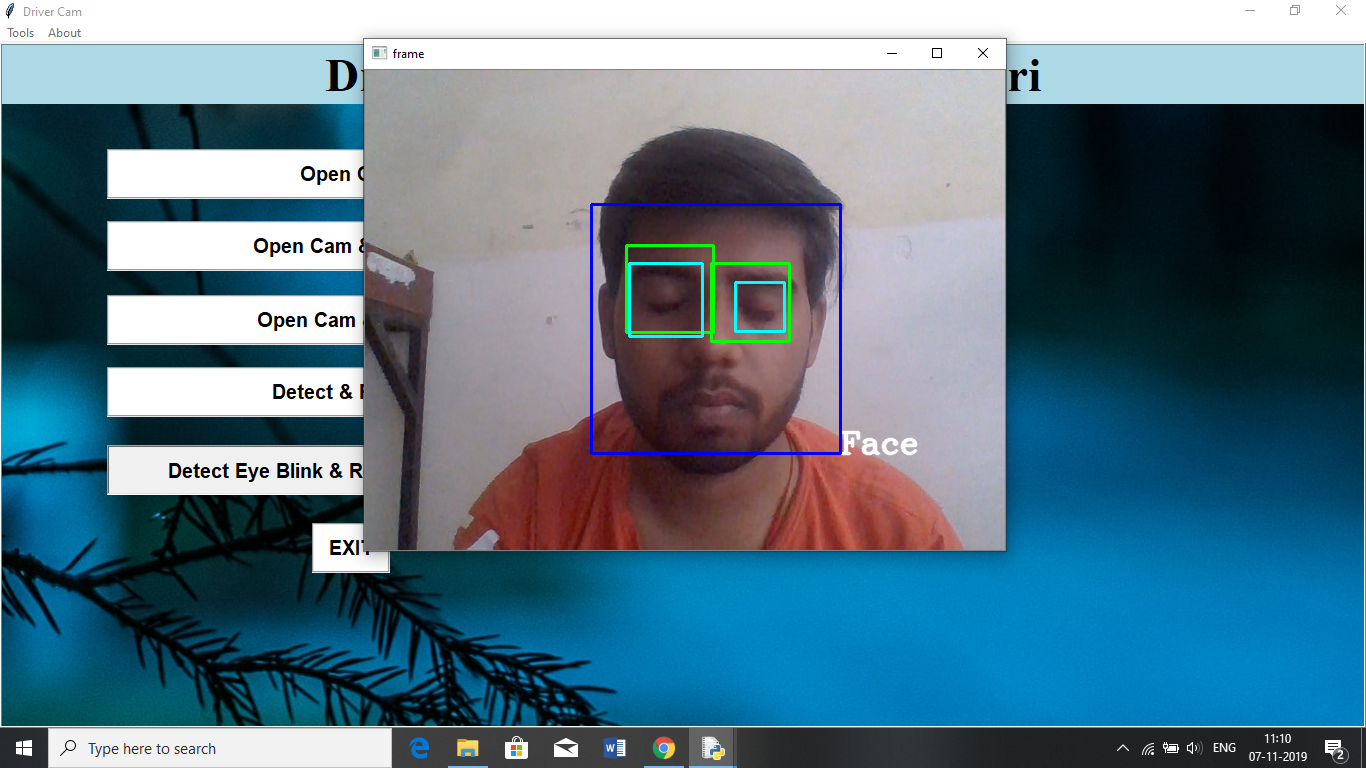


**Figure 15: Eye in closed state with head position= straight and with glass.**

The result is obtained by taking several position of head like straight, tilted (Right), tilted (Left) etc. We can observe that when the eyes are open circles appears around eye indicating open state of eye. When eyes are closed circle disappears indicating closed state of eye. In both the cases face detection occurs which is shown by a pink colored circle. In the side window the output parallels with one and zero according to the fact that eye is opened or closed respectively.



**Figure 16: Main Window.**

******

**Figure 17: Cam And Detect Window.**

## 

### 8.1 Conclusion

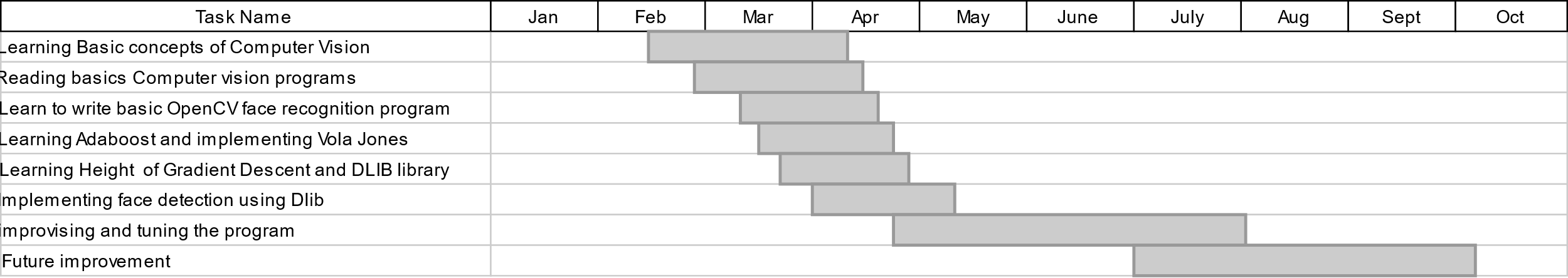
Implementation of drowsiness detection was done which includes the following steps: Successful runtime capturing of video with camera.

Captured video was divided into frames and each frames were analyzed. Successful detection of face followed by detection of eye. If closure of eye for successive frames were detected then it is classified as drowsy condition else it is regarded as normal blink and the loop of capturing image and analyzing the state of driver is carried out again and again. In this implementation during the drowsy state the eye is not surrounded by circle or it is not detected and corresponding message is shown. If the driver is not drowsy then eye is identified by a circle and it prints 1 for every successful detection of open eye.

### 8.2 Future work

Our model is designed for detection of drowsy state of eye and give and alert signal or warning may be in the form of audio or any other means. But the response of driver after being warned may not be sufficient enough to stop causing the accident meaning that if the driver is slow in responding towards the warning signal then accident may occur. Hence to avoid this we can design and fit a motor driven system and synchronize it with the warning signal so that the vehicle will slow down after getting the warning signal automatically. Also we can avoid the use of RaspberryPi which is not so fast enough for video processing by choosing our own mobile phone as the hardware. This can be done by developing a proper mobile application which will perform the same work as RaspberryPi and response will be faster and effective.

**8.3 Gantt Chart**



**Figure 18: Gantt Chart.**

**8.4 Code snippet**

**Important libraries**

import numpy

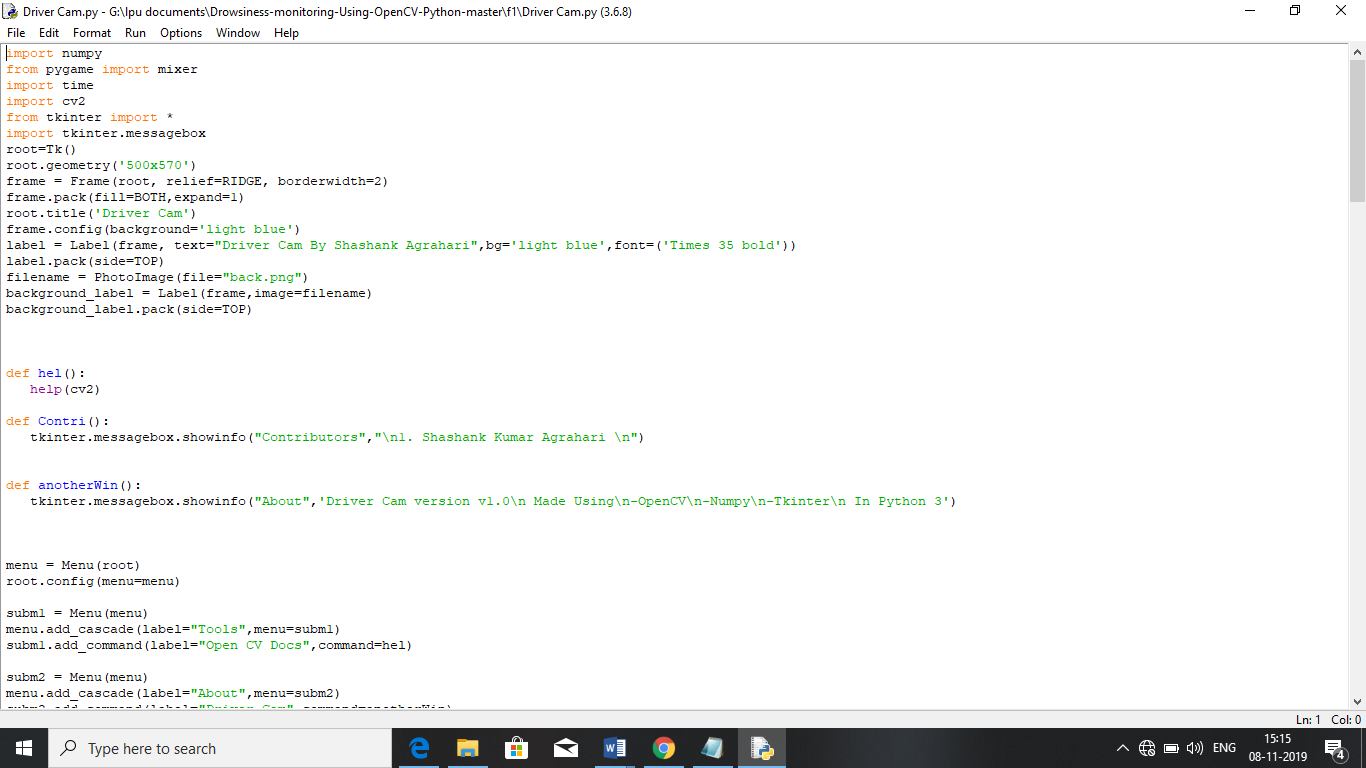
from pygame import mixer

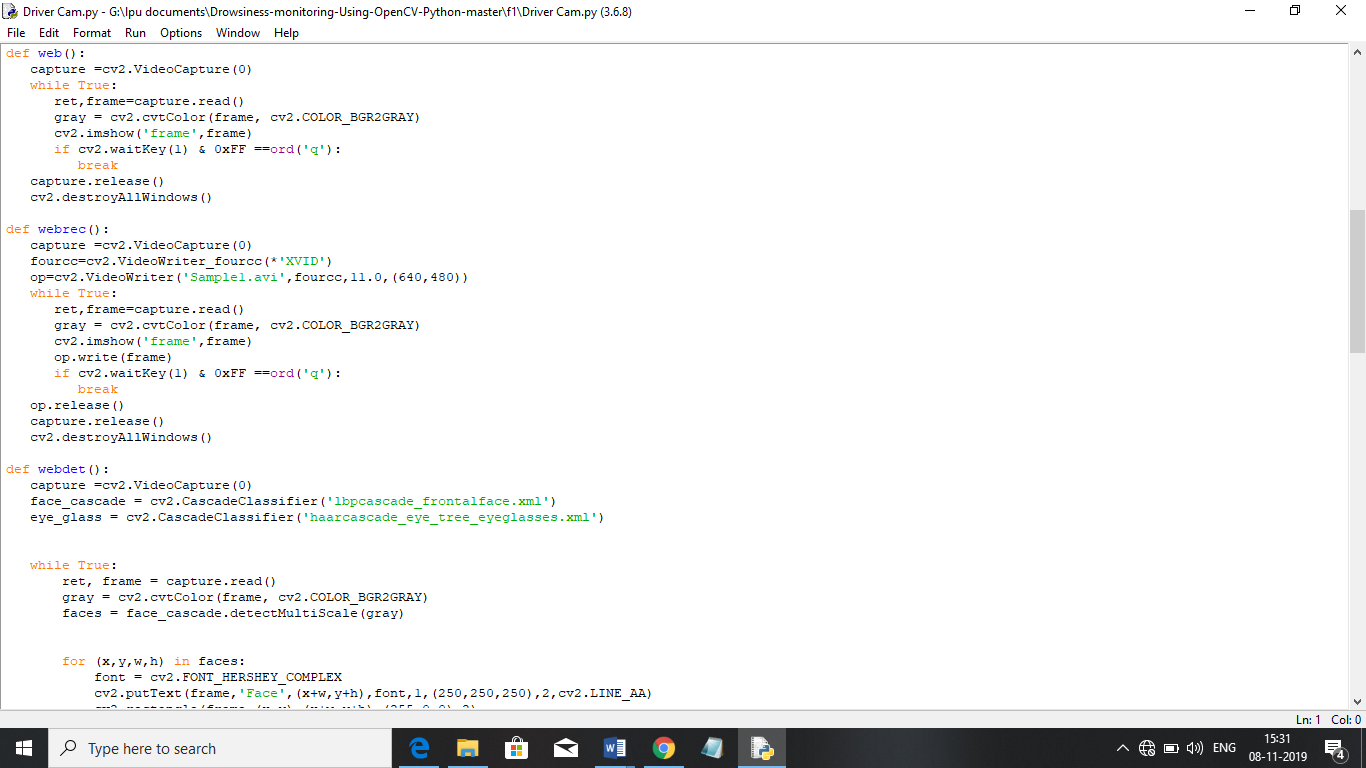
import time

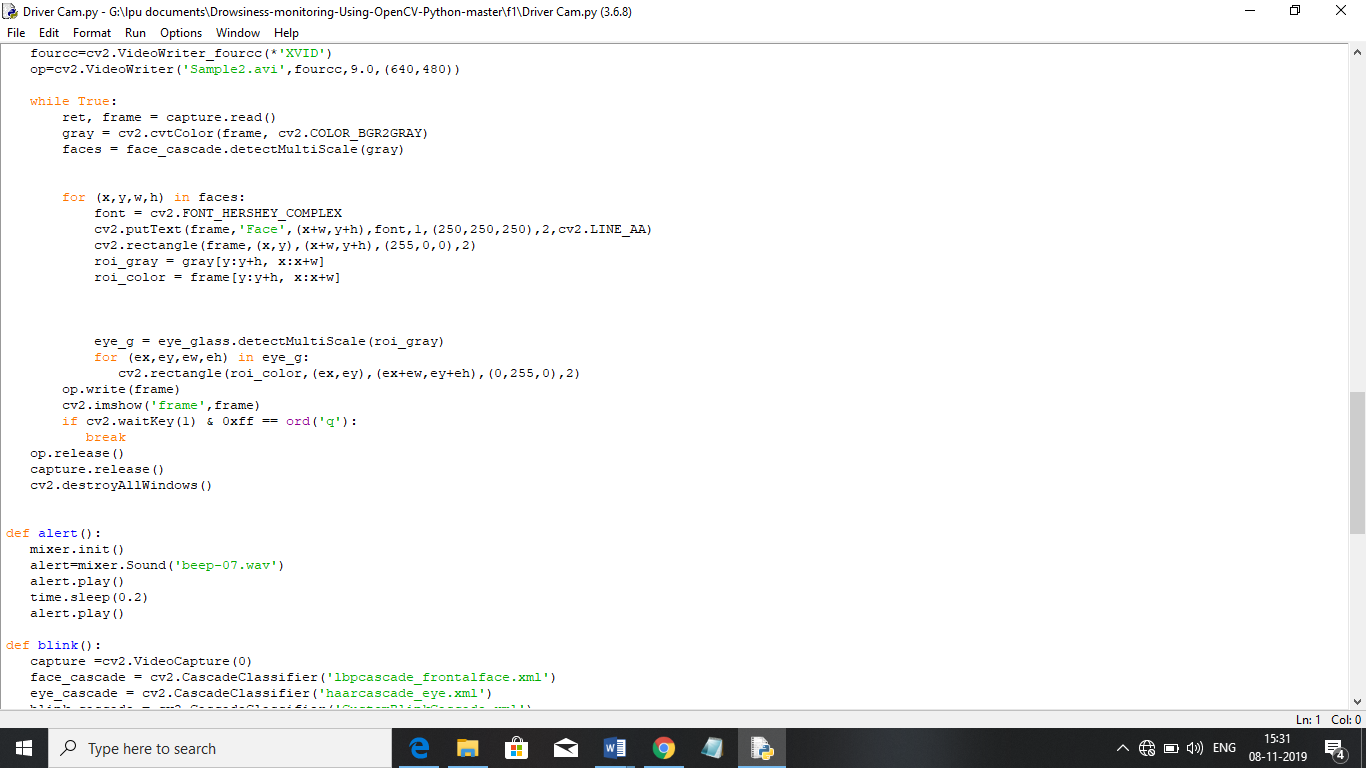
import cv2

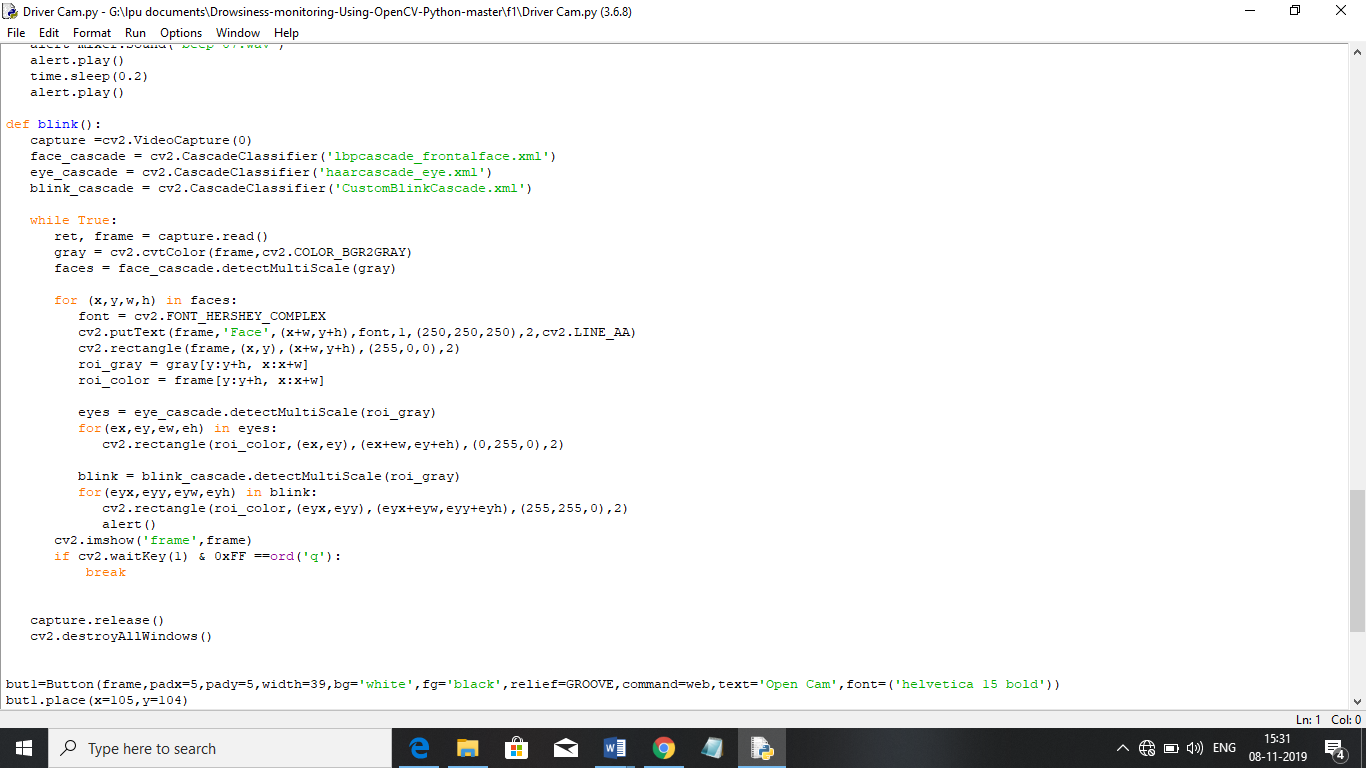
from tkinter import \*

import tkinter.messagebox









## BIBLIOGRAPHY

[1]. W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld, “Face Recognition: A Literature Survey,” ACM Computing Surveys, vol. 35, pp. 399-459, 2003.

[2] M. H. Yang, D. J. Kriegman, and N. Ahuja, “Detecting faces in images: A survey,”IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.1, , Jan. 2002, pp. 34–58.

[3]. Nan-Ning Zheng,Shuming Tang,Hong Cheng and Qing Li,Guanpi Lai and Fei-Yue Wang,”Toward Intelligent Driver-Assistance and Safety Warning Systems”,Intelligent Transportation System,IEEE 2004.

[4]. Christian Scharfenberger, Samarjit Chakraborty, John Zelek and David Clausi”, Anti-Trap Protection for an Intelligent Smart Car Door System”, 15th International IEEE Conference on Intelligent Transportation System, Anchorage, Alaska, USA, September 16-19, 2012.

[5]. K. C. Yowand, R. Cipolla, “Feature-based human face detection, “Image Vision Comput., vol.15, no.9, 1997, pp.713–735.

1. An Analysis of Viola Jones algorithm for face detection by Yi-Quin Wang, University of Malaysia Phang, 2014, pp: 15-20.
2. Implementation of Voila Jones Algorithm by Ole Helvig Jensen, university of Denmark, 2008, pp: 20-36.