A

Report on

"Driver Drowsiness Detection System"

Submitted To

KIT's College of Engineering, Kolhapur

In Partial Fulfillment of the Requirement for the Degree Of

Final Year of Engineering

(COMPUTER SCIENCE AND ENGINEERING)

submitted by

PRN No	Name of Student
1718000226	**Revati Patil**
1718000227	**Ruchita Patil**
1819010035	**Suhasini Sonavane**
1819010037	**Supriya Yadav**

under the guidance of

Mr. G. V. Otari

Kolhapur Institute of Technology's

College of Engineering, Kolhapur

Year 2020-21

CERTIFICATE

This is to certify that Ms. Revati Patil, Ms. Ruchita Patil, Ms. Suhasini Sonavane, Ms. Supriya Yadav have completed the Project on subject entitled "Driver Drowsiness Detection System", in the fulfillment of the requirement for the award of Final Year (Computer Science and Engineering) of KIT's College of Engineering, Kolhapur in the academic year 2020-21.

Date :

Place:

Mr. G. V. Otari Project Guide	External Examiner	Dr. M.S. Kalas Head of CSE

Dr. V. V. Karjinni
Director

ACKNOWLEDGMENT

We would like to express our deep gratitude towards Mr.G.V.Otari and Dr.M.S.Kalas Head of Department for their constant encouragement and guidance. Their support and cooperation throughout the project work has been of immerse help to us.

We express our sincere thanks to all the teaching and non-teaching staff and all those who have directly or indirectly helped in making project a success.

PRN No	**Name of Student**
1718000226	Revati Patil
1718000227	Ruchita Patil
1819010035	Suhasini Sonavane
1819010037	Supriya Yadav

Contents

1 ABSTRACT

The main idea behind this project is to develop a nonintrusive system which can detect fatigue of any human and can issue a timely warning. Drivers who do not take regular breaks when driving long distances run a high risk of becoming drowsy which they often fail to recognize early enough. According to the expert's studies show that around one quarter of all serious motorway accidents are attributable to sleepy drivers in need of a rest, meaning that drowsiness causes more road accidents than drink-driving. This system will monitor the driver eyes using a camera and by developing an algorithm we can detect symptoms of driver fatigue early enough to avoid the person from sleeping. So, this project will be helpful in detecting driver fatigue in advance and will give warning output in form of alarm and message will be displayed on the screen.

2 INTRODUCTION

Each year hundreds of people lose their lives due to traffic accidents around the world. Unfortunately Iran ranks first in the world in terms of road fatalities and each year approximately thirty thousands of fellow countrymen lose their lives in these events. The role of human factor in accidents cannot be ruled out. According to national statistics in 90 to 95 percent of car accidents in Iran, human factor plays a pivotal role. In general, the driver fatigue accounts for 25 percent of accidents and approximately 60 percent of road accidents result in death or serious injury. In a study by the National Transportation Research Institute (NTSRB) in which 107 random car accidents had been selected, fatigue accounted for 58% of the all accidents. A main cause of fatigue is sleeplessness or insomnia. Drivers' drowsiness is a major contributing factor in severe road accidents that claims thousands of lives every year.

Driver drowsiness and fatigue is a major factor which results into numerous vehicle accidents. Developing and maintaining technologies which can efficiently detect or prevent drowsiness at the wheel and alert the driver before am mishap is a major challenge in the field of accident prevention systems. Because of the dangerous that drowsiness can cause on the roads some methods need to be developed for preventing counteracting its effects. With the advent of modern technology and real time scanning systems using cameras we can prevent major mishaps on the road by alerting car driver who is feeling drowsy through a drowsiness detection system. The point of this undertaking is to build up a prototype drowsiness detection system. The spotlight will be put on planning a framework that will precisely monitor the open or shut condition of the driver's eyes continuously. 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 the observation of eye movements and blink patterns in a sequence of images of a face.

3 REVIEW OF LITERATURE

This part presents the literatrary survey of drowsiness detection approaches. According to the Survey on Driver Fatigue-Drowsiness Detection System, the detection system includes the processes of face image extraction, yawning tendency, blink of eyes detection, eye area extraction etc.

There are many experiments done with OpenCv for android also which is available for cheap smartphones as well. Other experiments conducted have resulted in utmost accuracy when camera were placed at different locations.

OpenCv is predominantly a technique for real time image processing which has free of cost implementations on latest computer vision algorithms. It has all required computer vision algorithms.

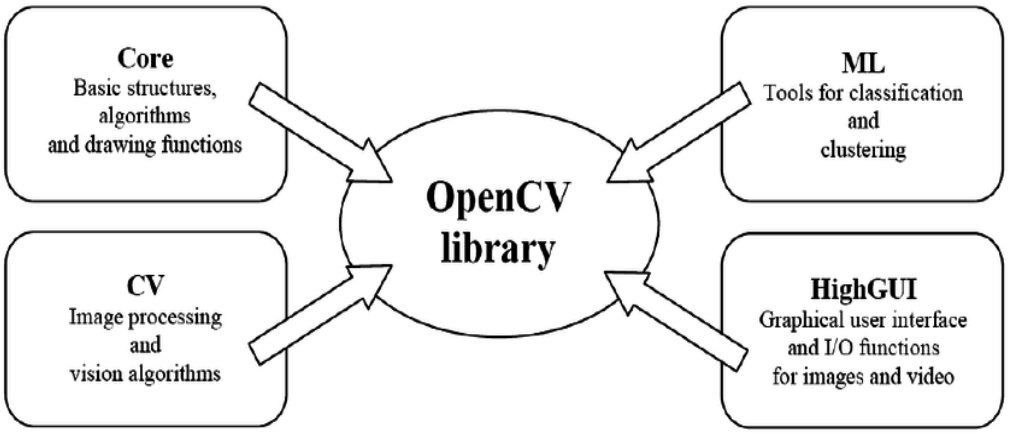


Fig1: OpenCv library

The outcomes of the study showed that the prototype is efficient, dependable and accurate as well in detecting the drowsiness of the driver. All this was done in realtime and represents non-intrusive fatigue monitoring.

4 SYSTEM ANALYSIS

4.1 Existing System

In the past few years, many researchers have been working on the development of safety systems using the different techniques. The most accurate techniques are based on physiological measures like brain waves, heart rate, pulse rate, respiration, etc. However, these techniques are intrusive since they require electrodes to be attached to the drivers, causing annoyance to them. A representative project in this line is the MIT Smart Car where several sensors (electrocardiogram, electromyogram, respiration, and skin conductance) are embedded in a car and visual information for sensor confirmation are used. In the advanced safety vehicle (ASV) project conducted by Toyota the driver must wear a wristband in order to measure his heart rate. Others techniques monitor eyes and gaze movements using a helmet or special contact lenses. These techniques, though less intrusive, are still not acceptable in practice.

4.1 Requirements

The functional and non-functional requirements are as follows

4.2.1 Functional Requirements

Functional requirement define the fundamental actions that system must perform.

- **Calculating Real Time Video**

  The system will capture frames and frames will be pre-processed.

- **Facial Landmark Detection**

  We will use Dlib library to detect facial landmarks and a threshold value is used to detect whether driver is drowsy or not. These facial landmarks are then used to compute the EAR (Eye Aspect Ratio) and MAR (Mouth Aspect Ratio).The EAR and MAR values will be compared with the threshold value. If the EAR value is less than the threshold value and MAR value is greater than the threshold value, then this would indicate a state of fatigue.

- **Output**

  The system will decide whether the driver is drowsy or not and based on that alarm will be triggered.

4.2.2 Non Functional Requirements

Non Functional requirements define the needs in terms of performance, design constraints, reliability and maintainability.

- **Reliability**

  The system is self-reliable.

- **Extensibility**

  New modules can added in future versions.

- **Usability**

  System is easy to learn and very efficient to use.

- **Cost**

  System is cost-effective.

4.2.3 Usability Requirements

- The system is designed for a user friendly environment in an effective way.

4.3 Problem Definition

Driver's inattention might be the result of a lack of alertness when driving due to driver drowsiness. Unlike driver drowsiness involves no triggering event but, instead, is characterized by a progressive withdrawal of attention from the road and traffic demands. It might have the same effects, i.e. decreased driving performance, longer reaction time, and an increased risk of crash involvement.

Designing a prototype Drowsiness Detection System which will focus on continuously and accurately monitoring the state of the driver's eyes in real time to check whether they are open or closed for more than a given period of time.

4.4 Objectives

Driver drowsiness detection is a car safety technology which spares the life of the driver by avoiding mishaps when the driver is getting languid.

• The primary goal is to initially plan a framework to distinguish driver's sluggishness by persistently checking eye and mouth conditions.
• To caution the driver on the identification of laziness by triggering the alarm and by giving alert message.
• To reduce the accidents.

5 PROPOSED SYSTEM

The proposed algorithm conducts the detection process by monitoring the video sequence of the drivers and image processing techniques. The system consists of three well-defined phases, namely the face detection, eye tracking, yawning detection.

The sequences of images from the camera are fed to the system. Initially, the system doesn't know the initial position of the face. The system grabs the first image and tries to find the face region in the image. Due to unfavorable lighting conditions or initial head orientation of the driver, the localization might fail. So the system grabs another frame and repeats the same process until the face region is detected with certainty and then eye tracking and yawing detection will done.

5.1 Purpose

Driver drowsiness detection is a car safety technology which helps to save the life of the driver by preventing accidents when the driver is getting drowsy. The main objective is to first design a system to detect driver's drowsiness by continuously monitoring eye and mouth from the face.

Every year many people lose their lives due to fatal road accidents around the world and drowsy driving is one of the primary causes of road accidents and death. Fatigue and micro sleep at the driving controls are often the root cause of serious accidents. However, initial signs of fatigue can be detected before a critical situation arises and therefore, detection of driver's fatigue and its indication is ongoing research topic. Most of the traditional methods to detect drowsiness are based on behavioral aspects while some are intrusive and may distract drivers, while some require expensive sensors. But in our project the system captures the live video and detects driver's face in every frame by employing image processing techniques. The system is capable of detecting facial landmarks, computes Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) to detect driver's drowsiness based on adaptive thresholding. Machine learning algorithms have been employed to test the efficacy of the proposed approach.

5.2 Scope

- The system captures the live video and detects driver's face in every frame by employing image processing techniques.
- The system is capable of detecting facial landmarks, computes Eye Aspect Ratio (EAR) and Mouth Aspect (MAR) to detect driver's drowsiness based on adaptive thresholding.
- This system deals with automatic driver drowsiness detection based on visual information.
- System will capture the video through camera and after processing, it will alert the driver based on the results.

5.3 Modules Involved

- **Segmentation of face:**

The face is segmented from the input image that is initially whatever the video that is monitored by the camera will be fragmented into the frames and this frames will be given as inputs for segmenting the face.

- **Eyes condition:**

The position of the driver's eye is determined by using appropriate threshold. In this work, edge detection of the eyes region is considered.

- **Yawning Detection:**

Facial lanmarking used in segmentation of various parts of the image, the mean-based clustering was utilized for yawning detection. The objective function was to obtain then minimum distance between the classes, or basically between the image pixels.

6 REQUIREMENTS

6.1 Hardware Interface

- Laptop with basic hardware
- Webcam

6.2 Software Interface

- **Python:**

  Python is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and supports multiple programming paradigms, including procedural, object-oriented, and functional programming

- **HTML and CSS:**

  HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format.

  Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. CSS is easy to learn and understood but it provides powerful control over the presentation of an HTML document.

- **Different libraries:**

OpenCV

OpenCV is an open source computer vision library accessible in python coding language to code for visionary capabilities of our smart pc.

OpenCV was expected for computational capability and having a high focus on ongoing picture location and distinguishing proof. OpenCV is coded with streamlined C and can take work with multicore processors. If we need progressively programmed improvement utilizing Intel models [Intel], you can purchase Intel's Integrated Performance Primitives (IPP) libraries [IPP]. These comprise of low-level schedules in different algorithmic regions which are streamlined. OpenCV consequently utilizes the IPP library, at runtime if that library is introduced.

The Origin of OpenCV

OpenCV left an Intel Research action proposed to drive CPU-raised applications. Towards this end, Intel moved various endeavors that included constant beam following and moreover 3D show dividers. One of the product engineers working for Intel at the time was visiting schools. He saw that several top school social affairs, like the MIT Media Lab, used to have well-made similarly as inside open PC vision frameworks—code which was supplied starting with one understudy then onto the next and which gave each resulting understudy an important establishment while building up his own vision application. Rather than rehashing the fundamental capacities from starting, another understudy may begin by adding to that which preceded.

OpenCV Structure and Content

OpenCV left an Intel Research movement planned to drive CPU raised applications. Towards this end, Intel pushed various endeavors that included continuous beam following and moreover 3D show dividers. One of the product engineers working for Intel at the time was visiting schools.

He saw that two or three top school social events, like the MIT Media Lab, used to have well-made similarly as inside open PC vision foundations.

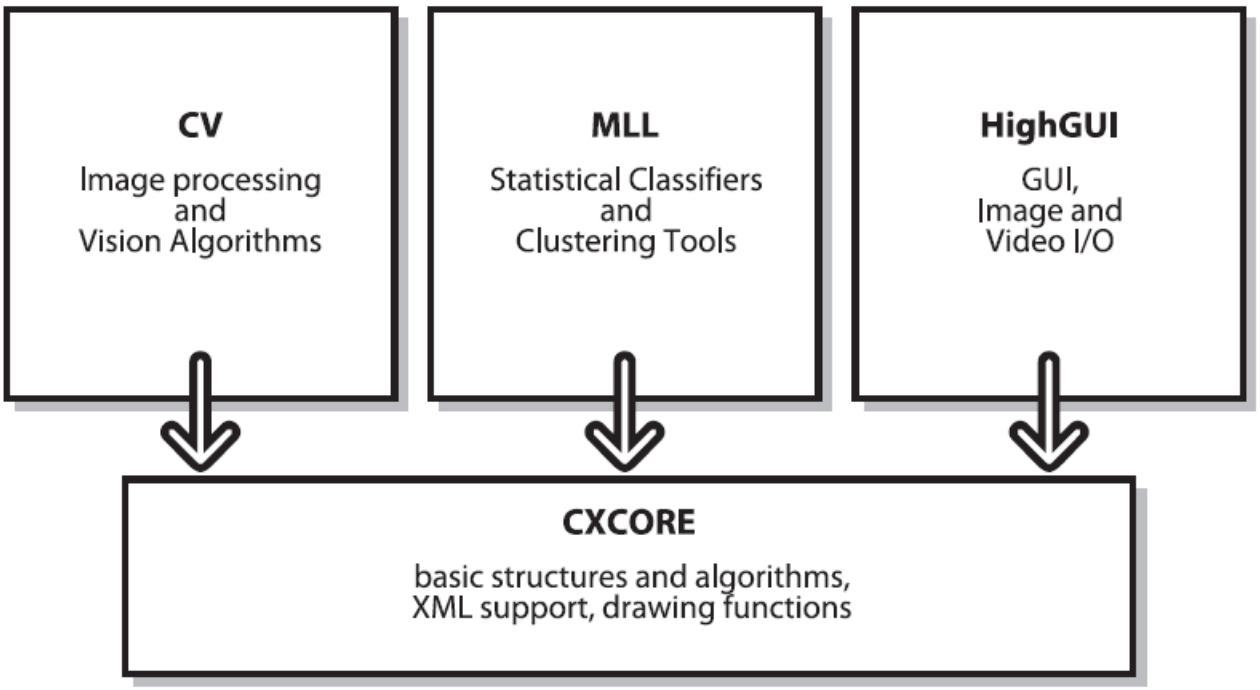


Fig2: Parts of OpenCV

Dlib Library

The dlib library serves us with a facial landmark detector as well as facial landmark predictor.

Scipy Library

SciPy is a python library that is useful in solving many mathematical equations and algorithms. It is designed on the top of Numpy library that gives more extension of finding scientific mathematical formulae like Matrix Rank, Inverse, polynomial equations, LU Decomposition, etc. Using its high level functions will significantly reduce the complexity of the code and helps in better analyzing the data.

- **Operating System:**

  Windows or Ubuntu

7 ARCHITECTURE

The general diagram of system has been shown in the figure. As it can be seen, the image received from camera is sent to central processor to be processed and then it will operate considering condition of drivers' face.

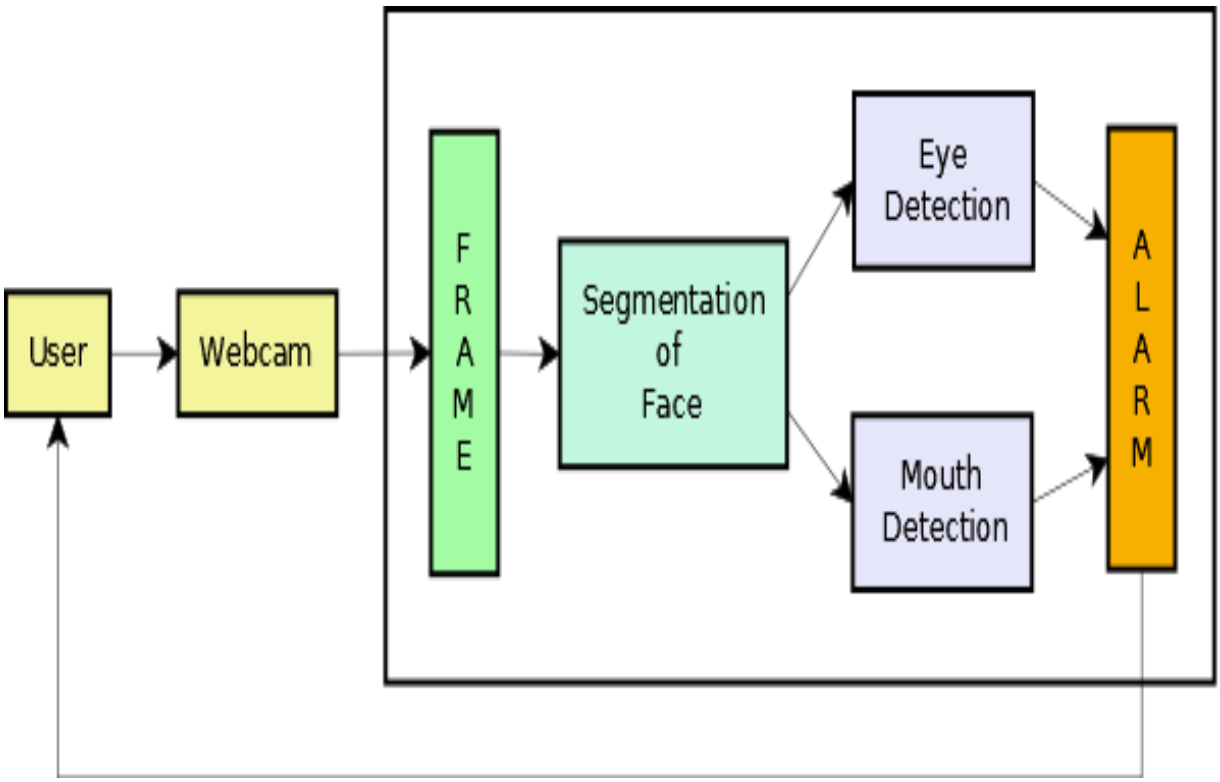


Fig3: Architecture of the System

8 DESIGN

8.1 Flowchart

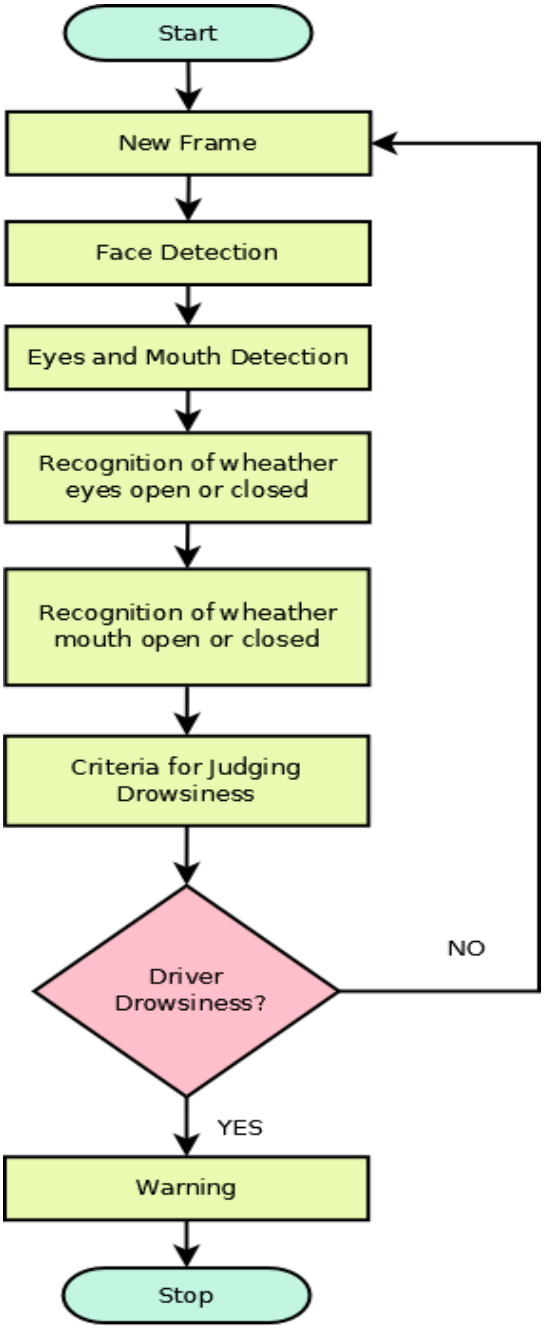


Fig4: Flowchart

8.2 Use case Diagram

Use case diagram is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

A use case defines the interactions between external actors and the system under consideration to accomplish a goal. Actors must be able to make decisions, but need not be human: "An actor might be a person, a company or organization, a computer program or computer system hardware, software, or both. Actors are always stakeholders, but not all stakeholders are actors, since they "never interact directly with the system, even though they have the right to care how the system behaves". For example, "the owners of the system, the company's board of directors, and regulatory bodies such as the Internal Revenue Service and the Department of Insurance" could all be stakeholders but are unlikely to be actors.

In our proposed system sequence diagram there are two actors i.e. the user and system were the camera monitors users face and system records the video and creates images for segmentation of face. Segmentation of face is done in order to extract only the eye and mouth region and discard the surrounding region which we are not interested in. Then the conditions for fatigue and non-fatigue are checked. If fatigue is detected then alarm is generated, if no fatigue is detected then no alarm is generated.

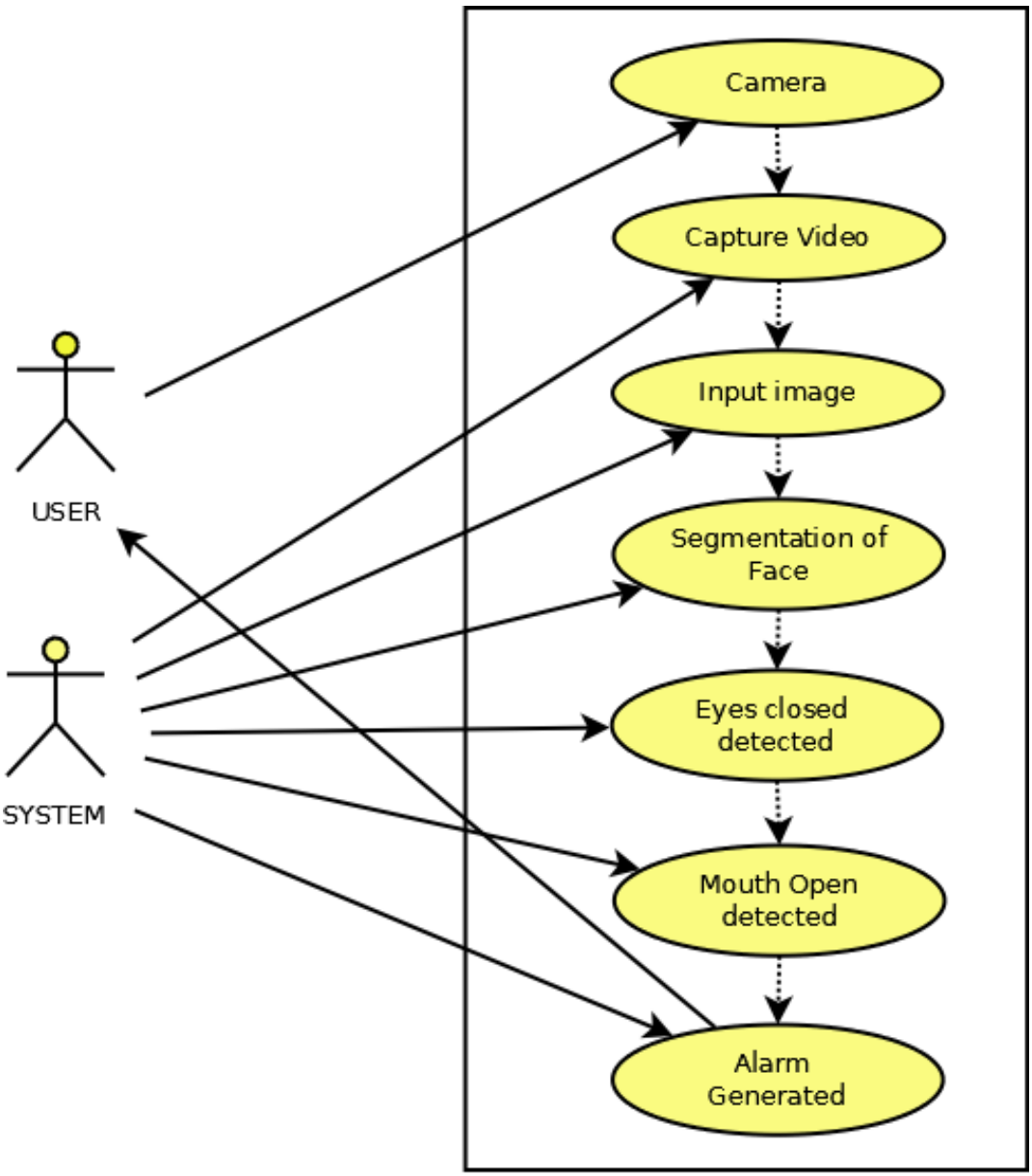


Fig5: Use Case Diagram of the System

8.3 Sequence Diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner

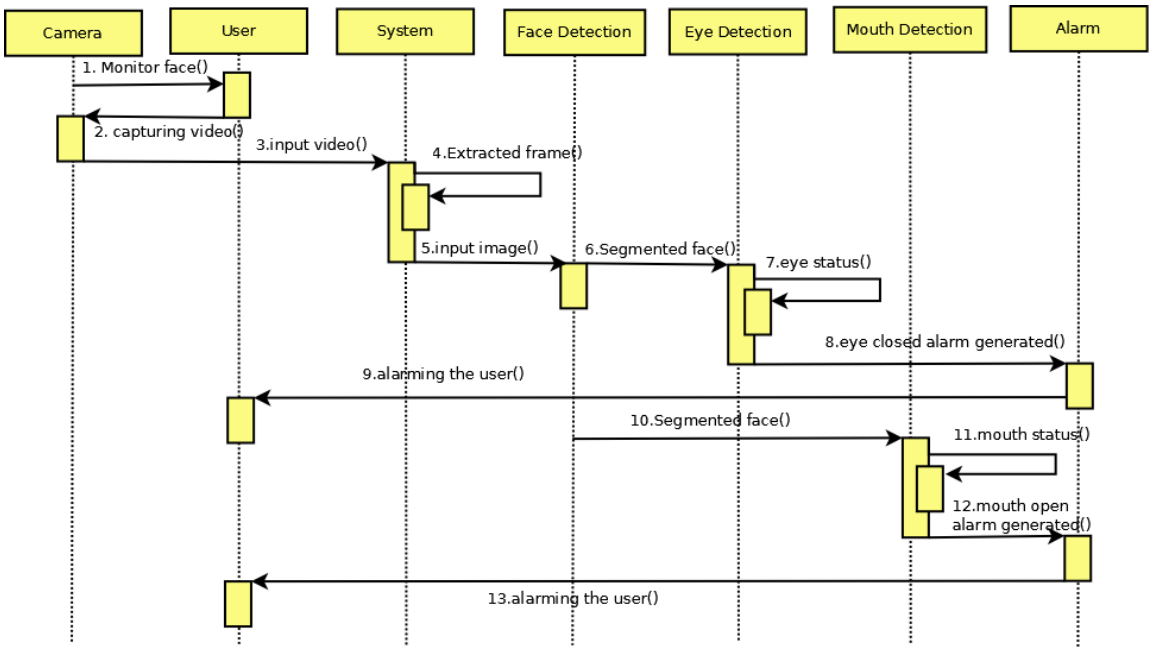


Fig6: Sequence Diagram of the System

8.4 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the flow of data through information system, modeling its process aspects. They can also be used for visualization of data processing.

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

Level 0: In this level, the person's face is filmed by a camera in the first step by receiving 30fps video sequence. Four different video sequences captures image frames.

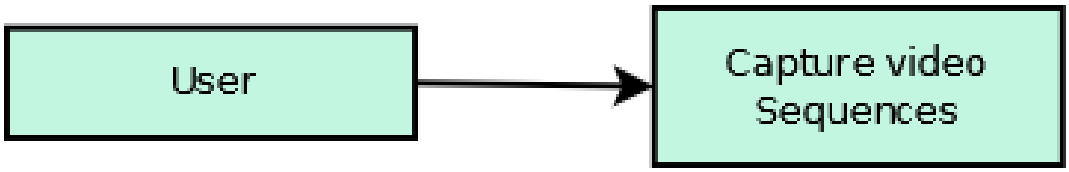


Fig7: Level 0 Data Flow Diagram

Level 1: After the video is captured , the position of the eyes and mouth are detected. The eye movements and mouth movements are tracked.

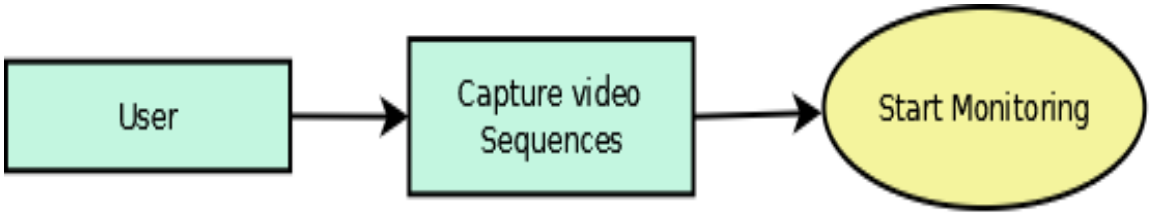


Fig8: Level 1 Data Flow Diagram

Level 2: Here, after the monitoring, if the eye condition of the driver is not proper or, he is yawning due to drowsiness or fatigueness, an alarm is generated to alert the driver. The camera is positioned so that we can monitor the driver's mouth and especially the eyes. We want to measure the movement of the mouth, eyes and eyelids with the aid of the camera. The camera is connected directly to the processor's video input, and it must meet certain technical requirements including for example adequate resolution, frame rate.

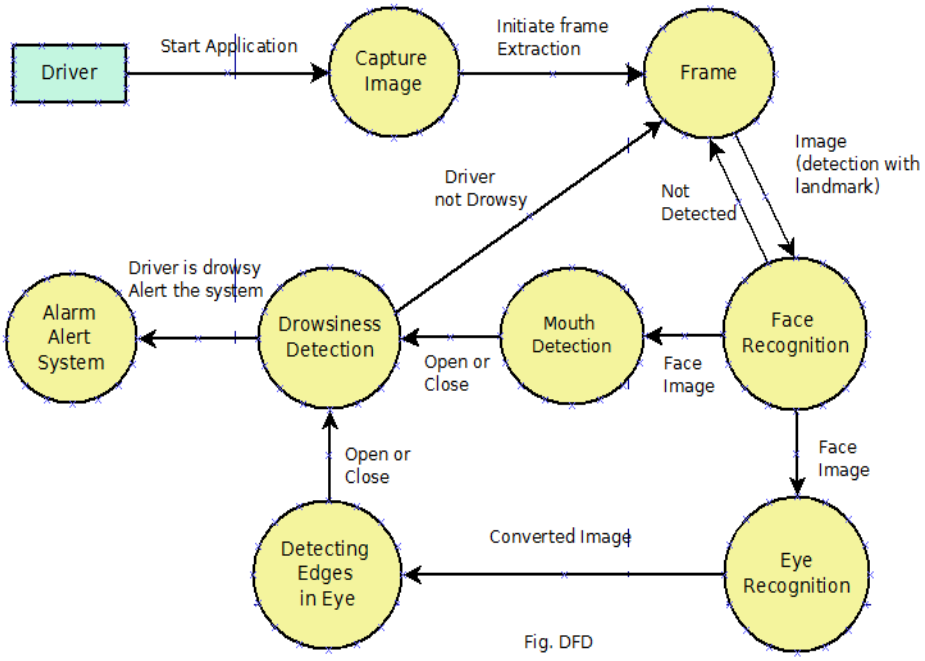


Fig9: Level 2 Data Flow Diagram

9 IMPEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. Thus it can be considered as the critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implement stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve change over and evaluation of change over methods.
Certain behavioral changes take place during drowsy driving like:

1. Yawning
2. Amount of eye closure
3. Eye blinking

9.1 Classifiers for Face Detection

9.1.1. HAAR Cascade Classifier

In haar cascade classifier primarily the haar structures are slide over one by one on an image, throughout the pixel values masked in black portion are added similarly all the pixel values overlaid in the white part are added, finally the sum values are compared and accordingly a threshold value is determined.

The classifier works on the principle of haar wavelet comparison and returns true value for object/face detection. This process is fast but not completely accurate as it may happen that a certain section of image has similar wavelets to that of the desired output.

In cascade classifiers there are n number of weak classifiers arranged in a cascade form. They are placed in such a manner that the first weak classifier is the simplest and then the complexity in each subsequent weak classifier increases linearly making the last weak classifier most complex. The combination of all these weak classifiers forms a strong

classifier. The main advantage of this classifier is its time efficiency.

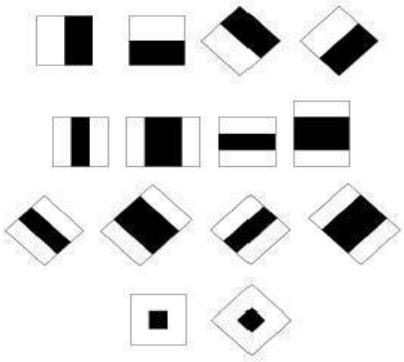


Fig10: HAAR Features

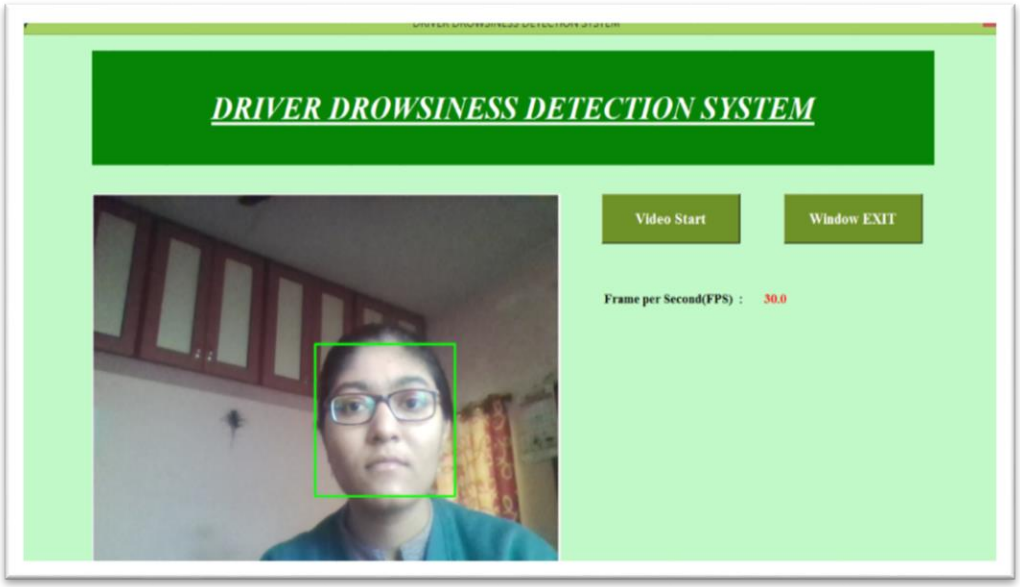


Fig11: Face Detection

9.1.2. Classifier Used

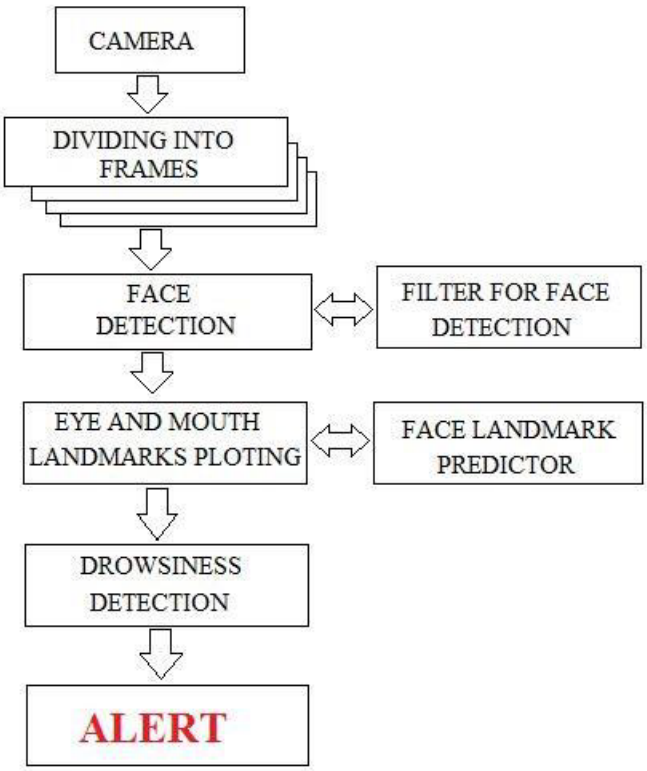
Histogram of Gradients(HOG) features are capable of capturing the pedestrian or object outline/shape better than Haar features. On the other hand, simple Haar-like features can detect regions brighter or darker than their immediate surrounding region better than HOG features. In short HOG features can describe shape better than Haar features and Haar features can describe shading better than HOG features.

That is also why Haar features are good at detecting frontal faces and not so good for detecting profile faces. This is because the frontal face has features such as the nose bridge which is brighter than the surrounding face region. But the profile face most prominent feature is its outline or shape, hence HOG would perform better for profile faces.

HOG and Haar-like features are complementary features; hence combining them might even result in better performance. HOG features are good at describing object shape hence good for pedestrian detection. Whereas Haar features are good at describing object shading hence good for frontal face detection.

HAAR cascade classifier is affected by the varying light intensity. Also if an object has HAAR wavelets similar to that of a face it recognizes that object as a face. On the other hand these limitations are overcome by HOG classifier as it works on the principle of segmentation. We are using HAAR classifier in this system.

Fig12: Perfect detection of 68 Facial Landmarks



9.1.3. Algorithm

1. At first, a camera is set up that monitors a stream for faces (OpenCV library is used for rapid and accurate image processing).

- Each pixel in the given image is classified as a skin pixel or a non-skin pixel. The different skin regions in the skin-detected image are identified by using connectivity analysis to whether each region identified is a face or not.

2. If a face is detected, the landmarks of facial features like eyes and mouth are mapped on the face using dlib library.

- Facial Landmark- It is a inbuilt classifier used to determine the position of 68(x, y) coordinates that map to facial structures on the face.

- The indexes of the 68 coordinates can be seen on the image below:

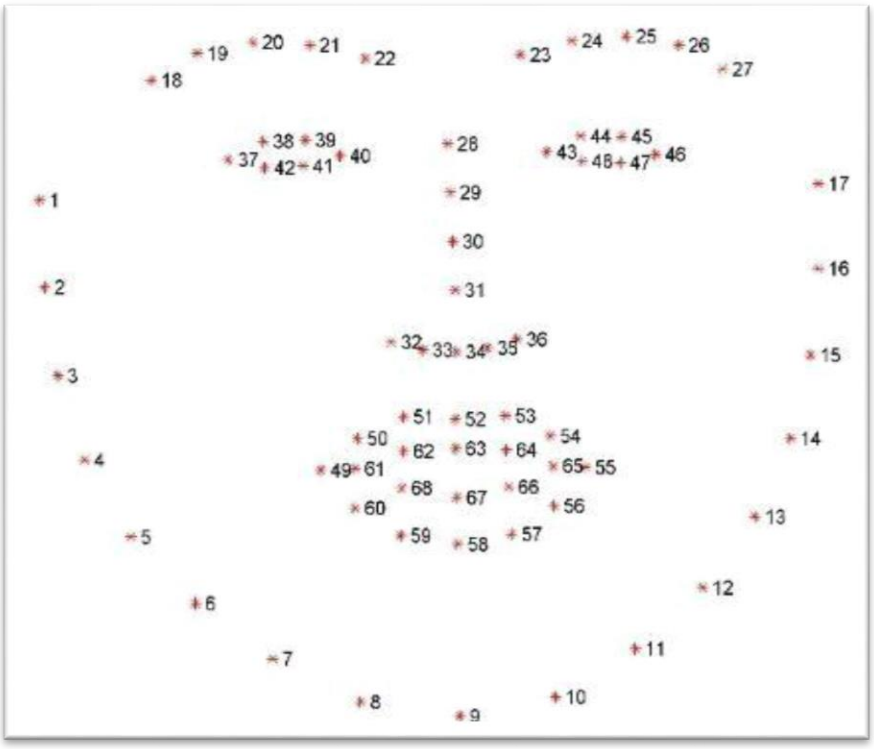


Fig13: 68 co-ordinates of Facial Landmarks

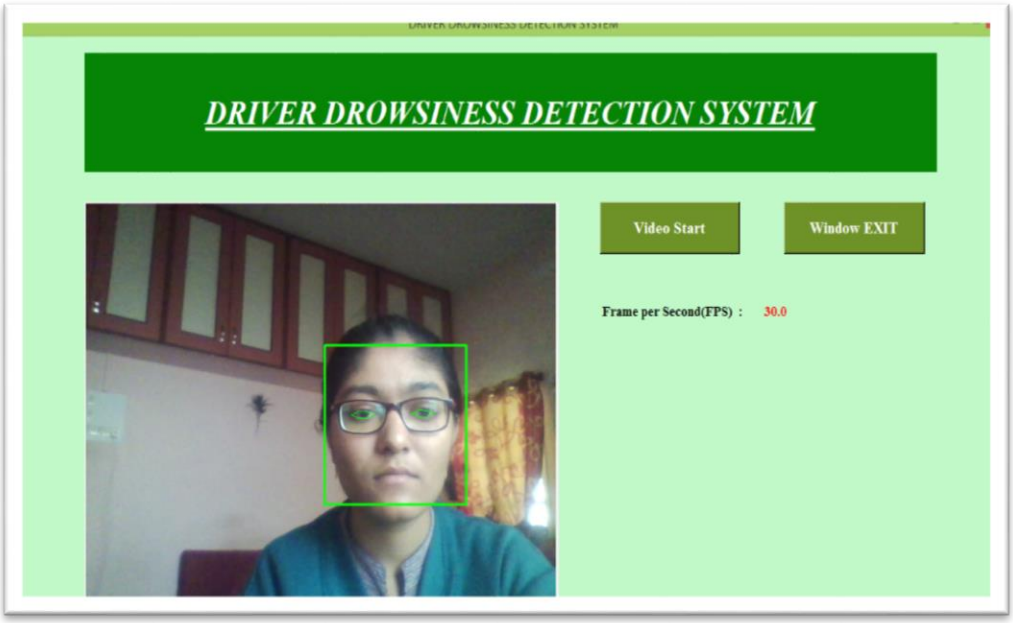


Fig14: Eyes Detection

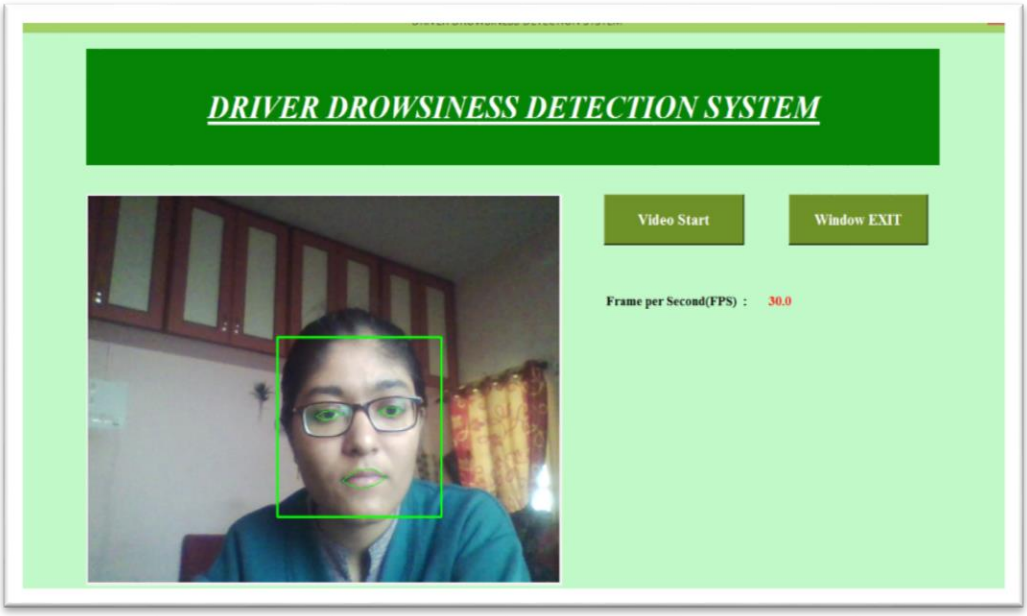


Fig15: Mouth Detection

3. After locating the eye and mouth landmarks, the eye aspect ratio and mouth aspect ratio is calculated to decide whether the driver is drowsy or not.(The eye aspect ratio and mouth aspect ratio is calculated by computing the Euclidean distance between the landmarks using SciPy library.)

4. Further if the eye aspect ratio and mouth aspect ratio varies abruptly from the pre-defined threshold value for a specific amount of time then the buzzer alerts the driver in real time.

9.1.4. Description of Features

If the distance between eye lids is measured for determining eye closure then it may not be the best parameter as this measure varies from person to person. Hence aspect ratio is the flawless parameter to exactly determine eye closure.

Aspect ratio: Aspect ratio is an image projection attribute that describes the proportional relationship between the width and height of an image, in this case eye. The aspect ratio is generally constant when the eye is open

and starts tending to zero while closing of eye. Since eye blinking is performed by both eyes synchronously the aspect ratio of both eyes is averaged.

$$EAR = \frac{|CD| + |EF|}{2 * |AB|}$$

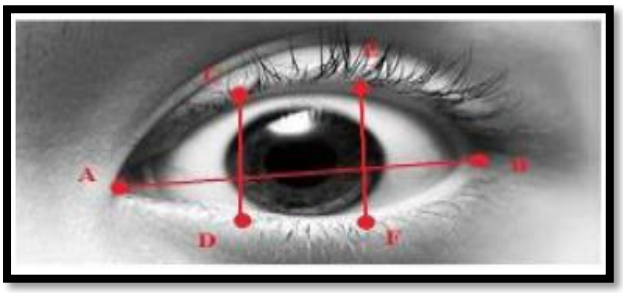


Fig16: Coordinates for Eyes

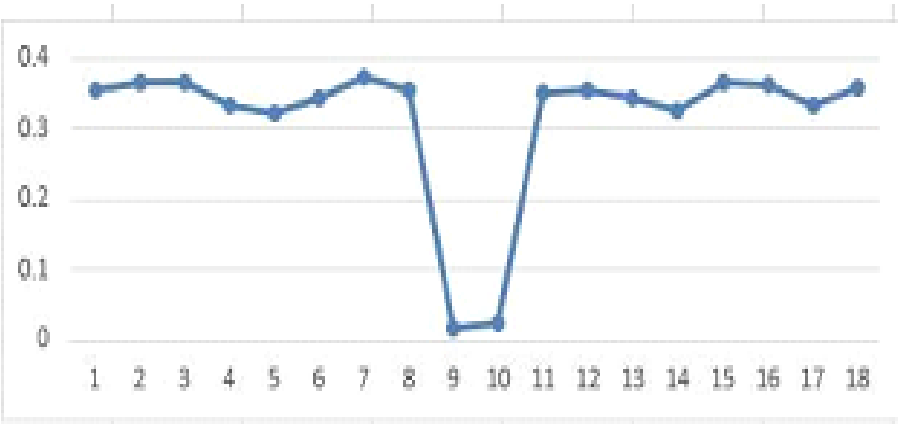


Fig17: Variation in EAR with Eyes opening and closing

From the graph it is can be seen that the threshold value is 0.3.upto the 8th frame the eye aspect ratio is above the threshold value indicating that the eye is open but as soon as the eye closes the eye aspect ratio drops drastically i.e. from the 8th frame to 12th frame the eye is shut again from the 12th frame as the eye is opened the eye aspect ratio increases above 0.3.

Similarly to determine the yawning parameter the aspect ratio of the mouth is calculated. It is calculated by the following formula,

$$MAR = \frac{|CD| + |EF| + |GH|}{3 * |AB|}$$

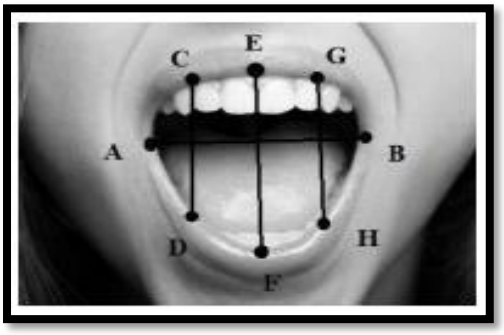


Fig18: Coordinates for Mouth

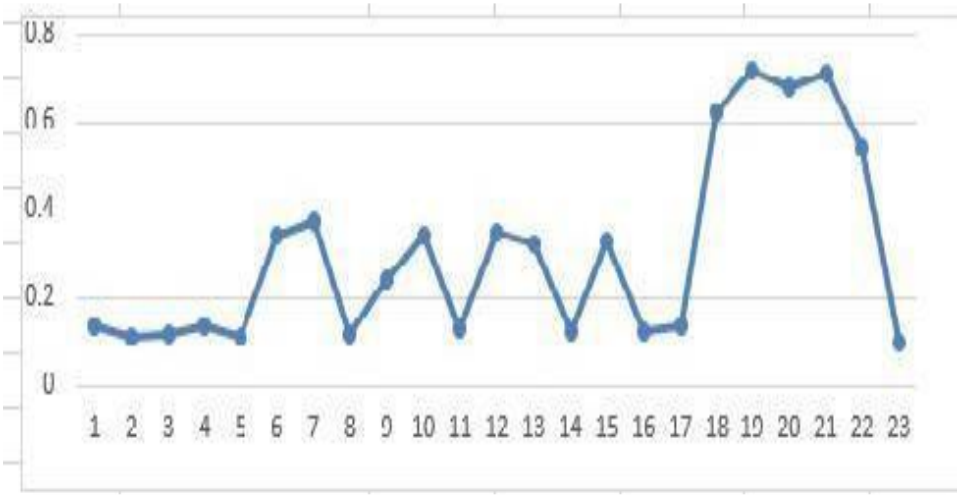
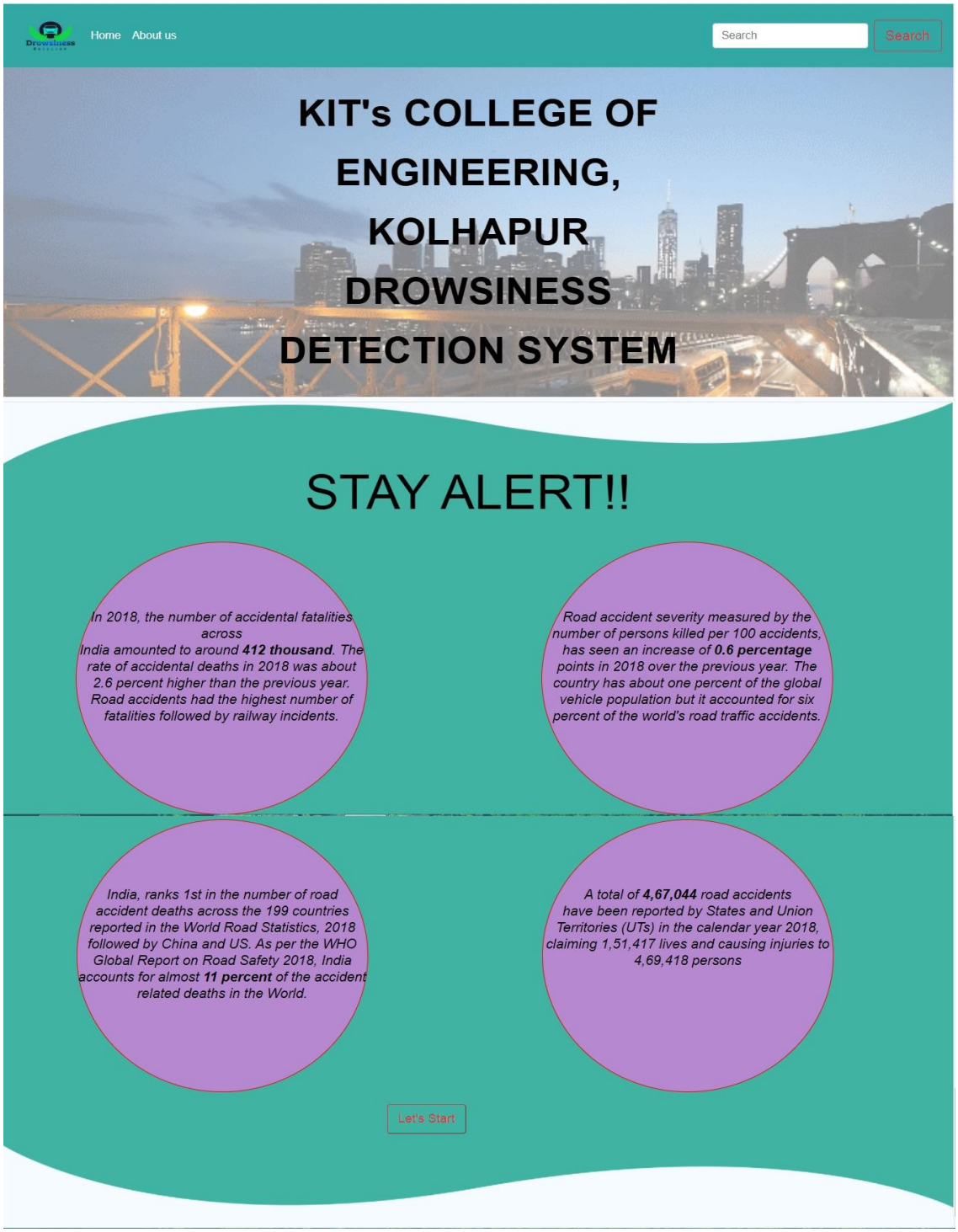


Fig19: Variation of MAR with Mouth opening and closing

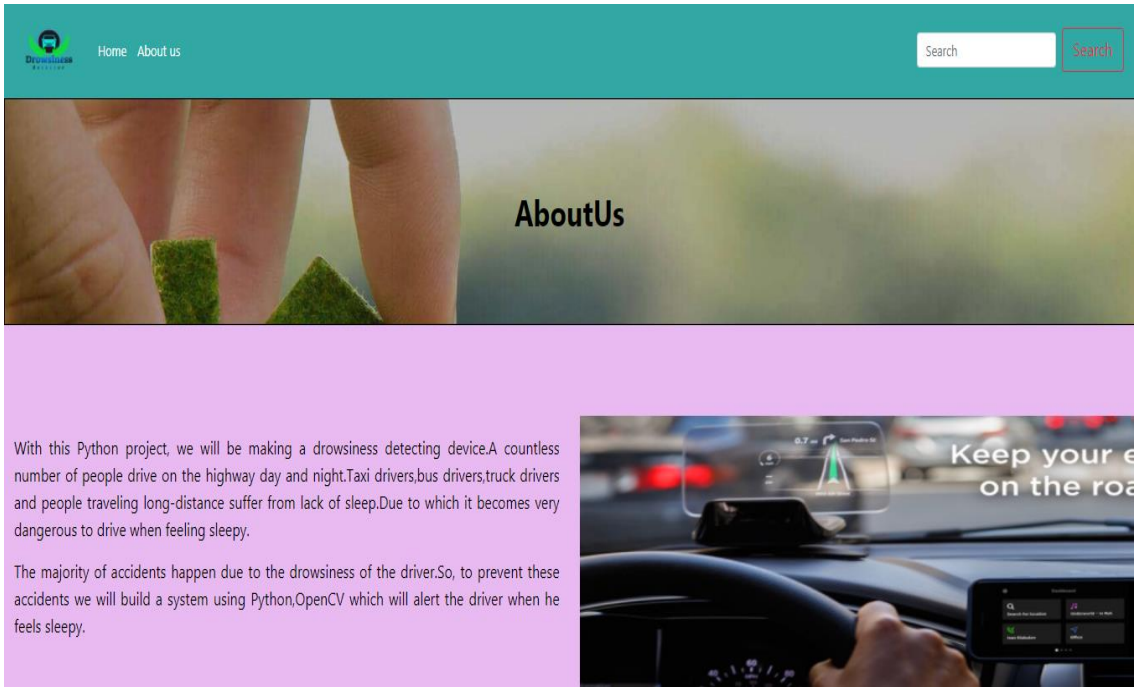
From the graph it is clearly visible that when the mouth is close the mouth aspect ratio almost zero which is case of first 5 frames. When the mouth is slightly open the mouth aspect ratio increases slightly. But in the frames from 17th to 23rd where the mouth aspect ratio is significantly high it is clear that the mouth is wide open most probably for yawning.

9.2 Screenshots

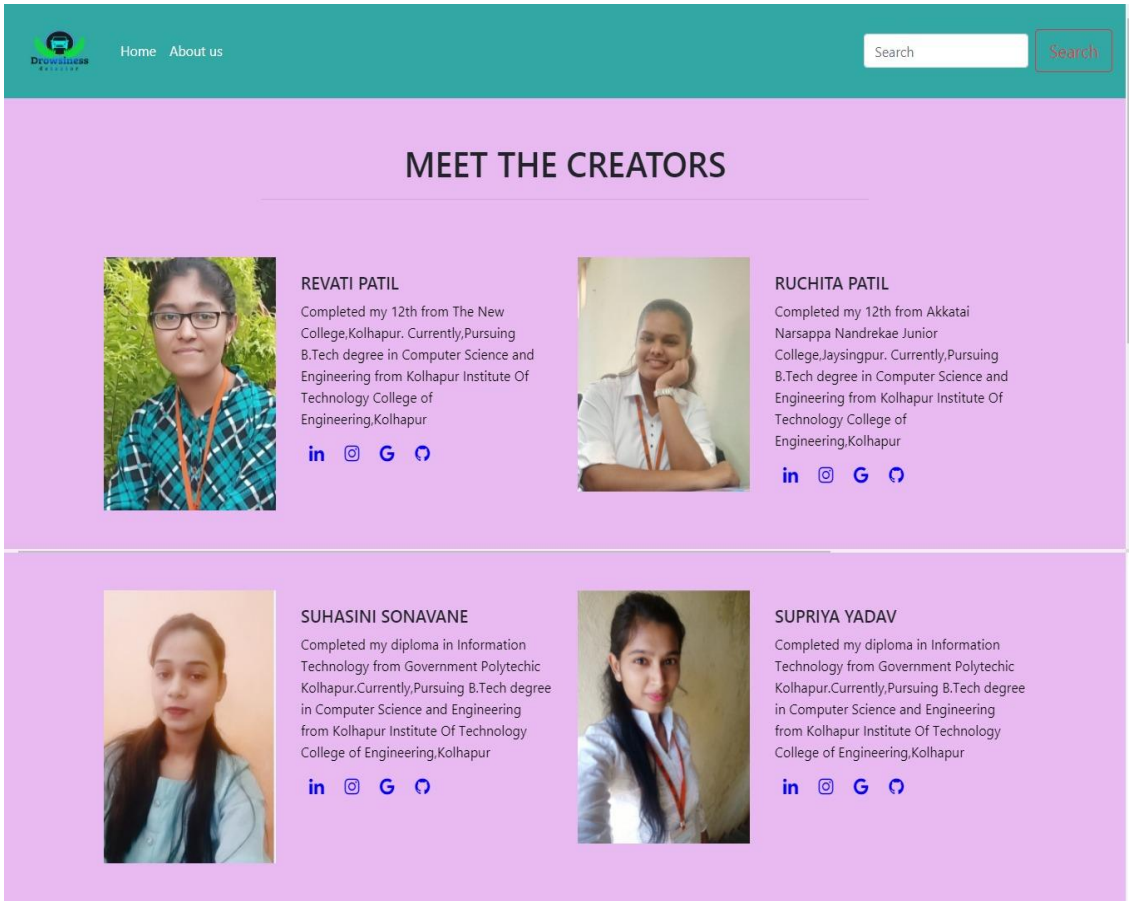
Home Page:

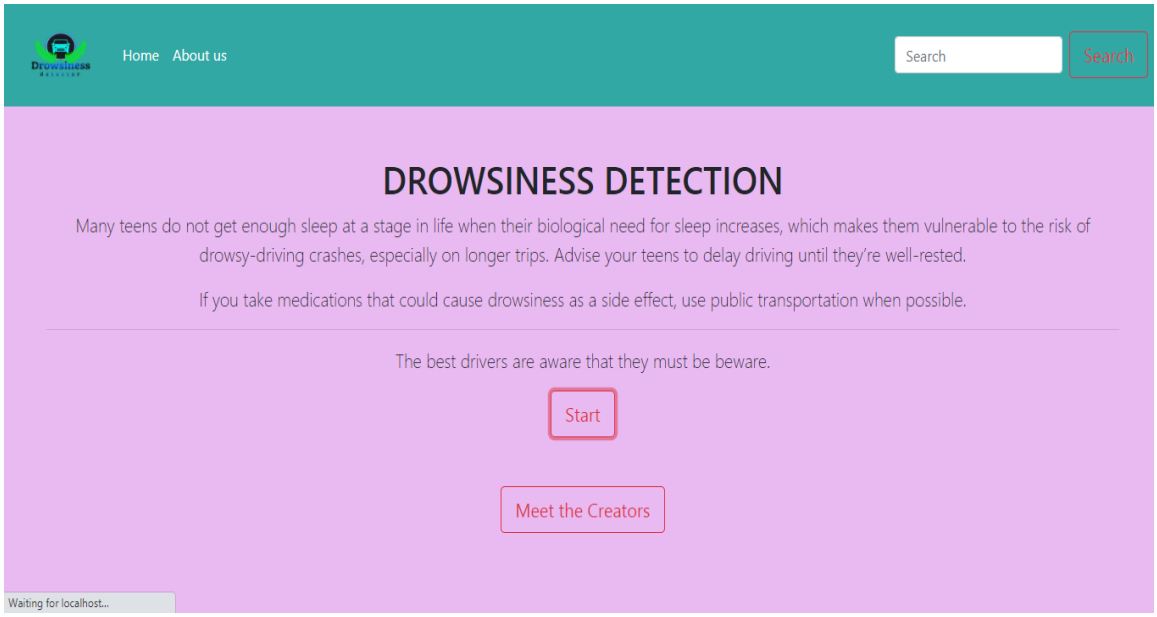


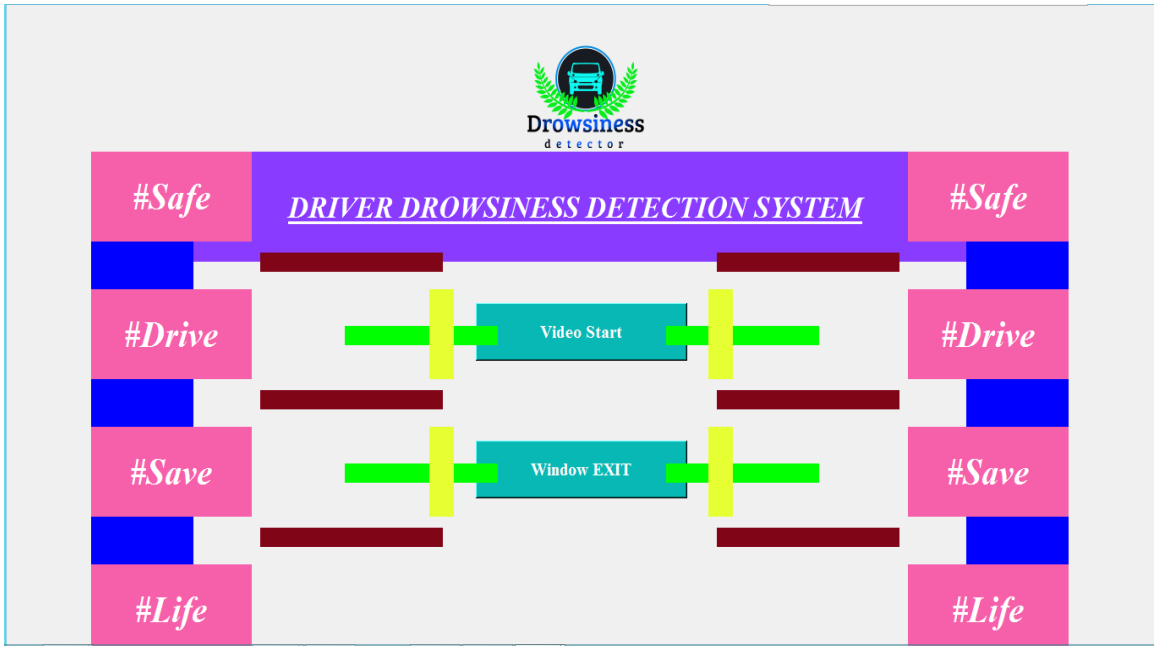
About us Page:



Meet the Creators Page:

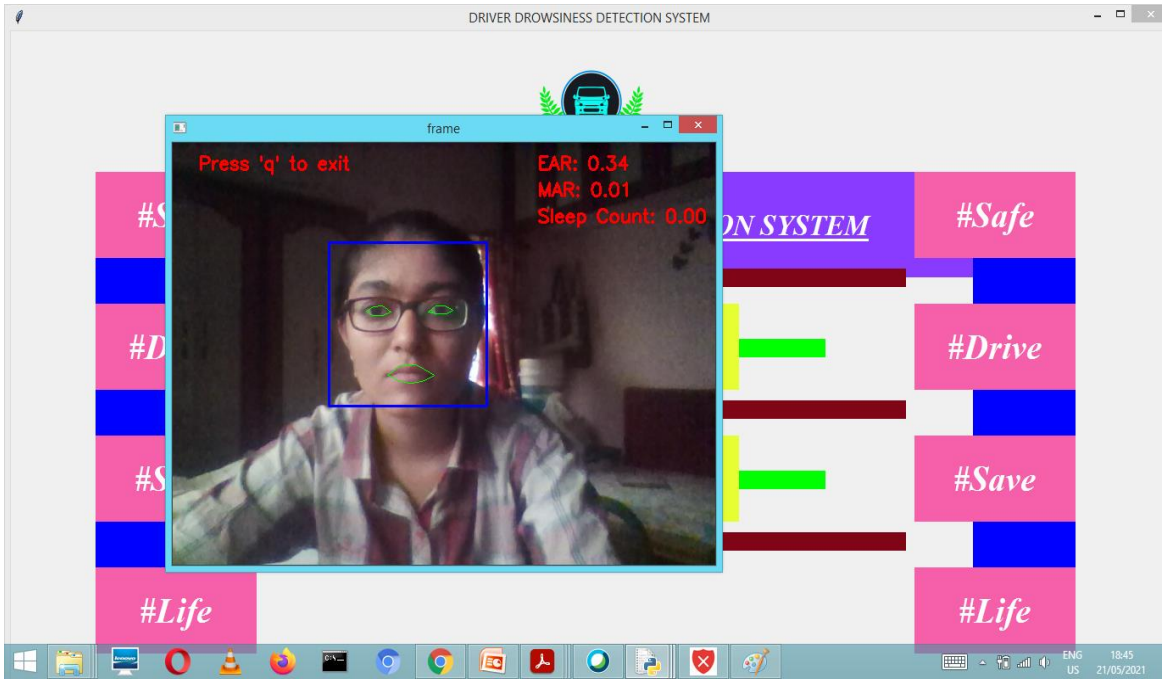



Drowsiness
detector
Home
About us
Search
Search
DROWSINESS DETECTION
Many teens do not get enough sleep at a stage in life when their biological need for sleep increases, which makes them vulnerable to the risk of drowsy-driving crashes, especially on longer trips. Advise your teens to delay driving until they're well-rested.
If you take medications that could cause drowsiness as a side effect, use public transportation when possible.
The best drivers are aware that they must be beware.
Start
Meet the Creators
Waiting for localhost...

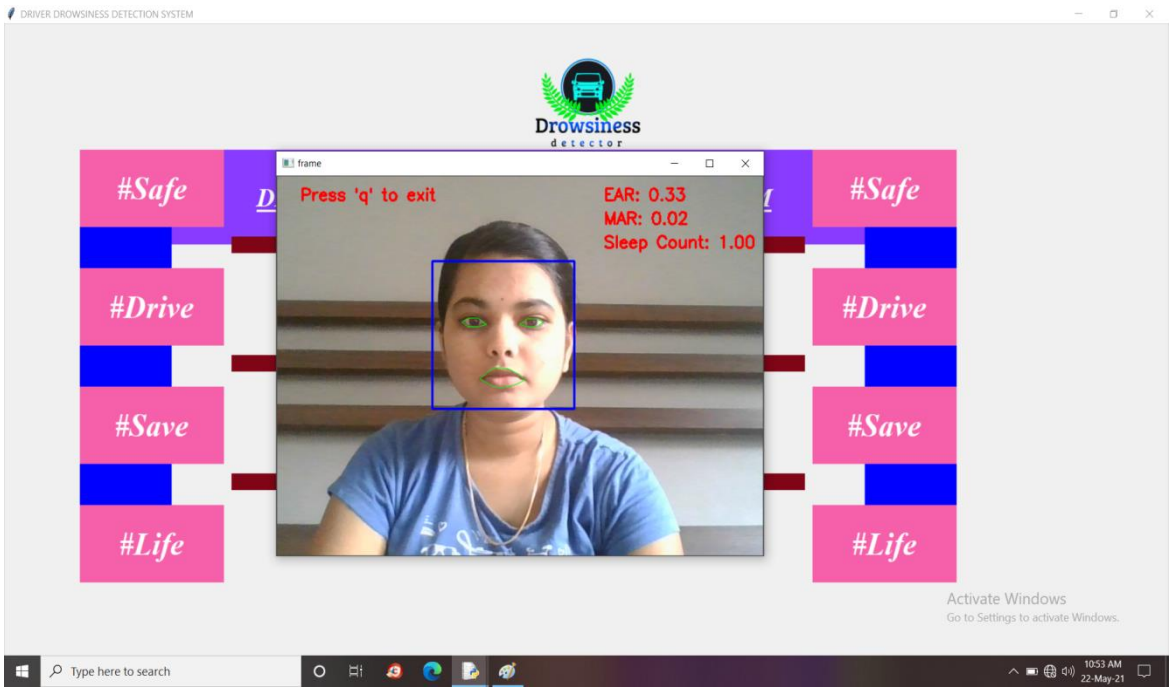

Drowsiness
detector
#Safe
DRIVER DROWSINESS DETECTION SYSTEM
#Safe
#Drive
Video Start
#Drive
#Save
Window EXIT
#Save
#Life
#Life

Case 1: Eyes open and Mouth closed- Non-fatigue

Sample 1:



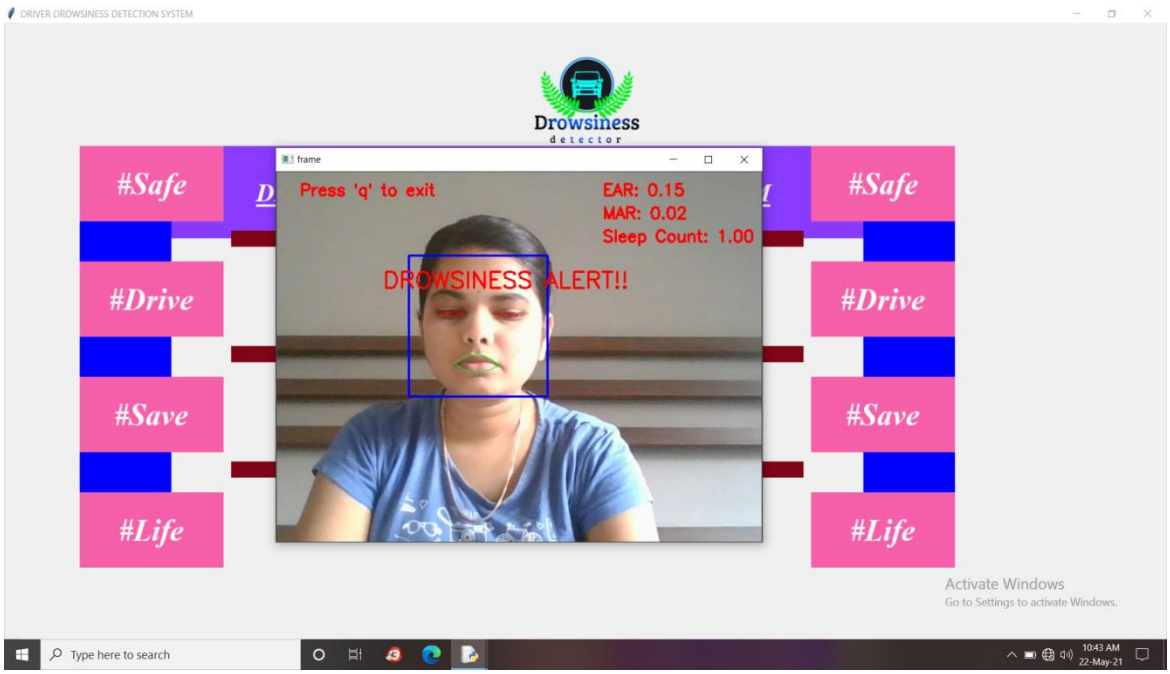
Sample 2:



Case 2: Eyes closed and Mouth closed- Fatigue

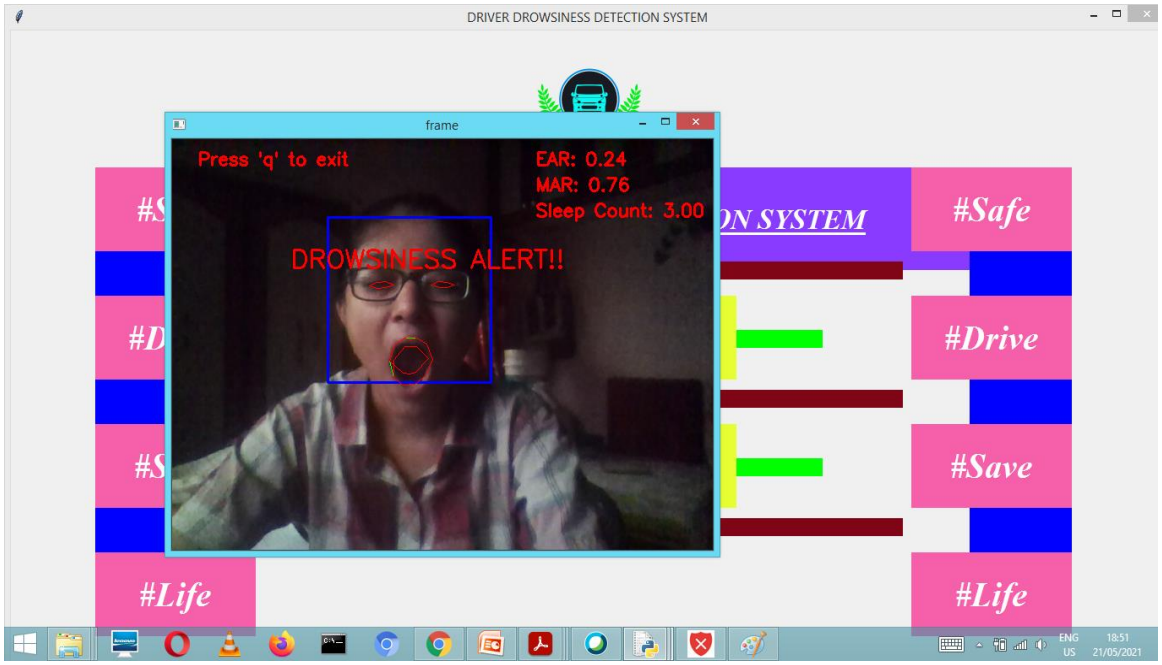
Sample 1:

Sample 2:

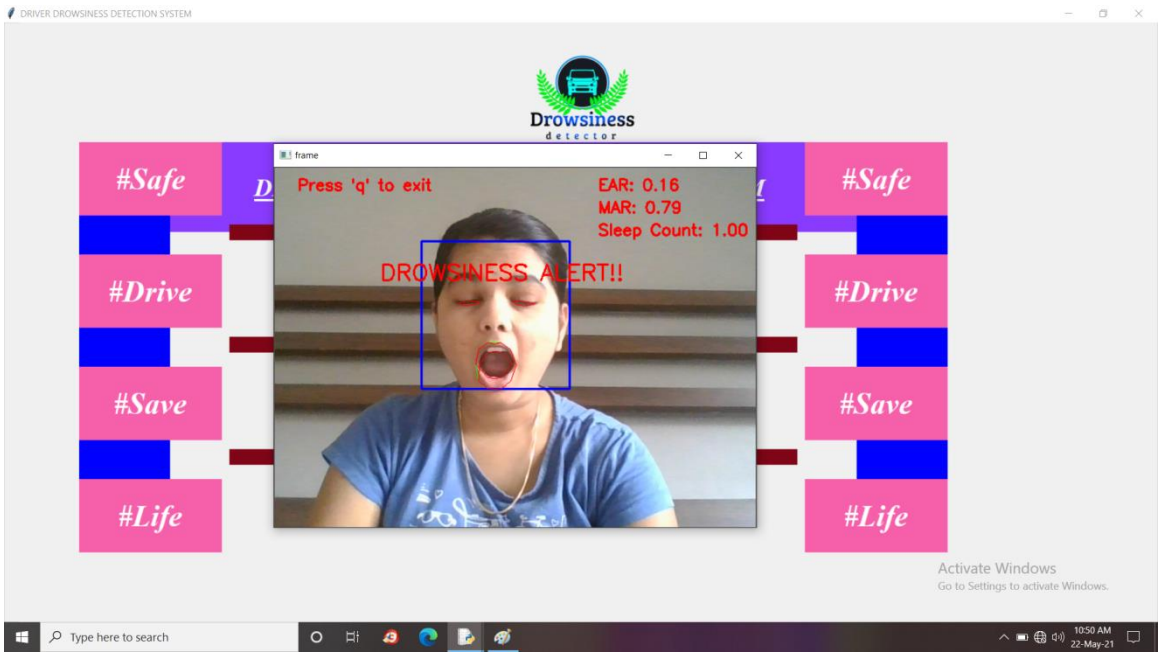


Case 3: Eyes closed and Mouth opened- Fatigue

Sample 1:

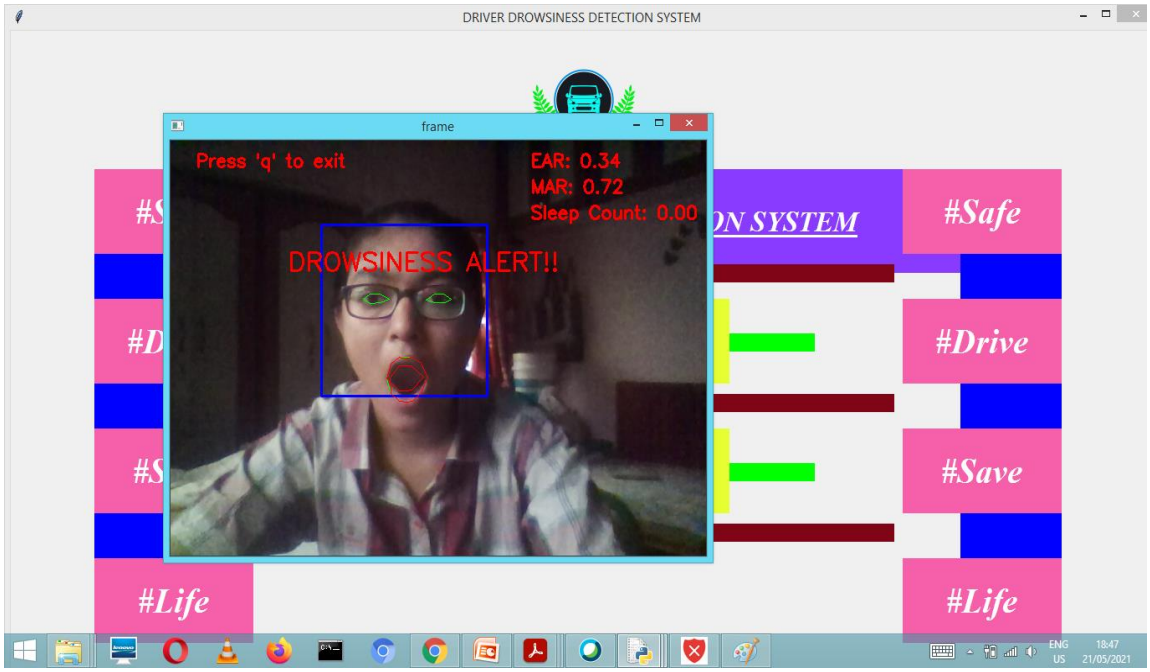


Sample 2:

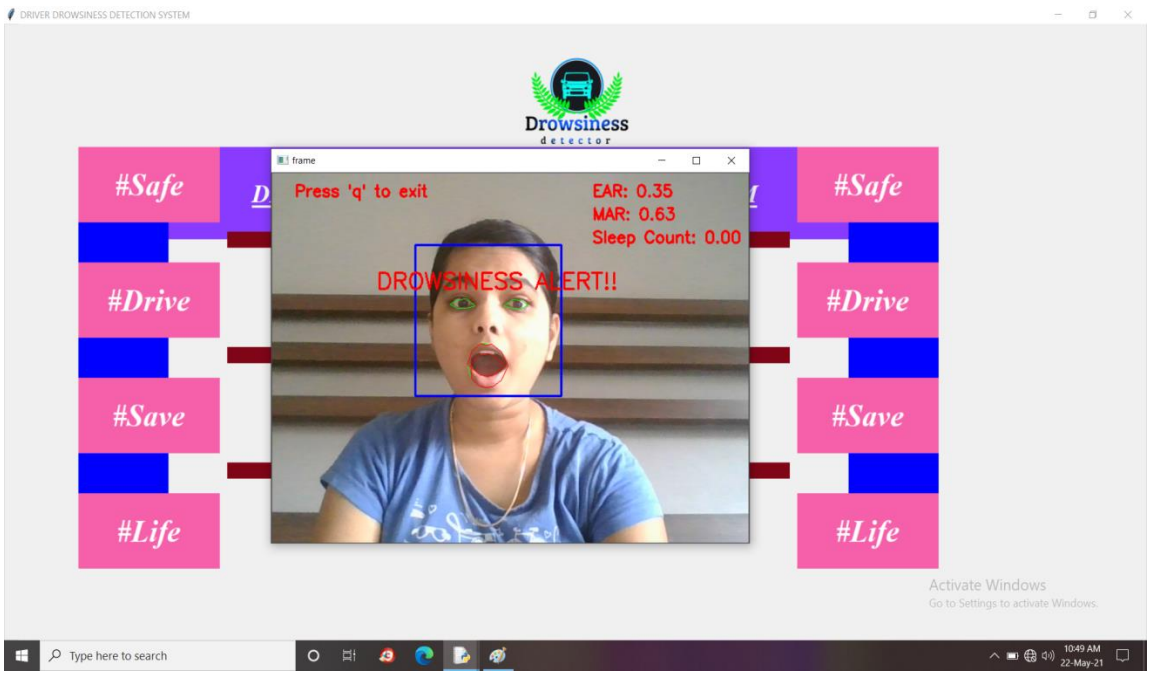


Case 4: Eyes opened and Mouth opened- Fatigue

Sample 1:



Sample 2:



9.3 Assumption and Dependencies

For the better result there has to be some assumptions and dependencies which gives the output with less error. These are given as follows.

- Input video must have good quality
- Frame in the video should have good effect
- Face of the suspect in the video must be very clear
- Front view of the face is more preferable :

  Person's face in the video should be present in the front view because of which the process of detection becomes very efficient and fast. If the face orientation is slightly tilted or not perfectly centered, then detection process will takes more time to detect the face.

10 TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover faults or defects in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the software system meets its requirements and user exceptions and does not fail in an unacceptable manner.

A good test case is the one that has a high probability of finding an as yet undiscovered error. A successful test is the one that uncovers a undiscovered error. Testing may be carried out during the implementation phase to verify that the software behaves as intended by its designers and after implementation is complete.

10.1 Software Testing

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

10.2 Software Testing Types

Software testing life cycle is the process that explains the flow of the tests that are to be carried on each step of software testing of the product. The V-model i.e. Verification and validation model is a perfect model which is used in the improvement of the software project. This model contains software development life cycle on one side and software testing life cycle on the other hand side. A checklist for software tester sets a baseline that guides him to carry on the day-to-day activities.

10.2.1 Black Box Testing

It explains the process of giving the input to the system and checking the output, without considering how the system generates the output. It is also called as Behavior Testing.

Functional Testing

In this type of testing, the software is tested for the functional requirements. This checks whether the application is behaving according to the specification.

Performance Testing

This type of testing checks whether the system is performing properly, according to the user's requirements. Performance testing depends upon the Load and Stress Testing that is internally or externally applied to the system.

Integration Testing

Integration Testing is the phase in software testing in which individual software modules are combined and tested as a group. This mostly focuses in the design and construction of the software architecture. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items.

Integration testing is further classified into Bottom-up Integration and Top-Down Integration testing.

Bottom-up Integration Testing

It is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested. All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing.

Top-Down Integration testing

It is an approach to integrated testing where the top down integrated modules are tested and the branch of the module is tested step by step until the end of the related module.

System Testing

System testing is the testing conducted on a complete, integrated system, to evaluate the system's compliance with the specified requirements. This type of software testing validates that the system meets its functional and non-functional requirements and also intended to test beyond the bounds defined in the software/hardware requirement specifications.

10.2.2 White Box Testing

It is the process of giving the input to the system and checking, how the system processes the input, to generate the output. It is mandatory for a tester to have the knowledge of the source code.

Unit Testing

This type of testing is done at the developer's site to check whether a particular piece/unit of code is working fine. Unit testing deals with testing the unit as a whole.

10.3 A list of sanity Test Cases

Prior to any testing it's mandatory to write test cases, so that it helps in Giving an approach, description, pre-conditions to achieve the expected result. Test cases are reusable, helps in regression testing while releasing various versions.

Test Cases for the Applications

Test case ID	Test Case	Input	Expected output	Action	Result
TID1	Detecting face	Input Frame	Face detected	Segmentation of face	Pass
TID2	Detecting eyes	Segmentation of face	Eyes detected	Edges of the eyes	Pass
TID3	Detection of mouth	Segmentation of face	Detected mouth	Clustered mouth with large hole	Pass
TID4	Driver's fatigue condition	Eyes open and Mouth closed	Non Fatigue	No Alarm	Pass
TID5	Driver's fatigue condition	Eyes closed and Mouth closed	Fatigue	Alarm generated	Pass
TID6	Driver's fatigue condition	Eyes closed and Mouth opened	Fatigue	Alarm Generated	Pass
TID7	Driver's fatigue condition	Eyes opened and Mouth opened	Fatigue	Alarm Generated	Pass

11 CONCLUSION AND FUTURE WORK

11.1 Conclusion

The system proposed in this paper is acceptable level of performance. The high fatalities of road accidents, which is primarily due to human errors committed out of fatigue, justifies the use of this system to alarm drivers at the time of driving. High-speed data processing and great accuracy distinguish this system from the similar ones. The development and improvement of this system can save the lives of millions of people annually.

Implementation of drowsiness detection with Python and OpenCV was done which includes the following steps: Successful runtime capturing of video with camera. Captured video was divided into frames and each frame was analyzed and successful detection of face followed by detection of eyes and mouth takes place. 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. i.e. if for continuous 20 frames the eyes are closed and the EAR value falls below 0.3 then the state of driver is detected as drowsy. And for mouth if the MAR value is greater than 0.6 then also the driver’s state is detected as drowsy. When state of driver is detected drowsy the alarm is triggered to wake up the driver. These factors, which have been proposed for assessing the detection accuracy of the video sequence, indicate the acceptable performance of the proposed system in detecting the signs of fatigue in driver's face at the time of driving.

11.2 Future Enhancement

In future works, a driver's distraction identification system will be developed. With its complex and ever-changing nature, including the effect of the light and the condition of shooting environment, it makes the skin segmentation of human faces in color images severely affect face detection, and also makes it an important research topic. Compared with the conventional method of segmentation, we put these methods, such as adjudging the images with the light interference, enhancing the images and improved threshold segmentation. Determination of the light interference not only improves the accuracy of image segmentation in the follow-up processing, but also expands the scope of application with skin segmentation in color images.

Ultrasonic sensors are connected on the left and behind the car to detect the distance between the car and the road side and to detect any other car behind. Based on the situation the system tries to reduce the speed and to stop the car. When the driver is feeling drowsy the future technology would be enhanced so that the sensors are applied in cars and the car gives the indication to the neighboring vehicle and just moves towards the lane and park the vehicle.

12 REFERENCES

1. Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhatt, “Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio”, Peer review under responsibility of International Conference on Sustainable Computing in Science, Technology and Management

2. Ashlesha Singh, Chandrakant Chandewar and Pranav Pattarkine, “Driver Drowsiness Alert System with Effective Feature Extraction”, INTERNATIONAL JOURNAL FOR RESEARCH IN EMERGING SCIENCE AND TECHNOLOGY, VOLUME-5, ISSUE-4, APR-2018

3. Isha Gupta, Novesh Garg, Apoorva Aggarwal, Nitin Nepalia and Bindu Verma, “Real-Time Driver’s Drowsiness Monitoring Based on Dynamically Varying Threshold”, Proceedings of 2018 Eleventh International Conference on Contemporary Computing (IC3), 2-4 August, 2018, Noida, India

4. Maganti Manasa, Vikas B, K. Subhadra, “Drowsiness detection using Eye-Blink frequency and Yawn count for Driver Alert”, International Journal of Innovative Technology and Exploring Engineering (IJITEE),ISSN: 2278-3075, Volume-9 Issue-2S3, December 2019

5. K .H. Wanjale, Amit Bhoomkar, Ajay Kulkarni, Somnath Gosavi, “Use Of Haar Cascade Classifier For Face Tracking System In Real Time Video”, International Journal of Engineering Research & Technology (IJERT),Vol. 2 Issue 4, April – 2013, ISSN: 2278-0181