Abstract—When the driver does not get proper sleep, rest or fell sleepy, they sleep while driving and it could be fatal to driver and even the passengers. This issue should have a solution in form of a system in which they can identify drowsiness on the face of a driver and then could ring an alarm so that driver can take necessary actions after that.

The detection is done mainly in three steps, in beginning the system should identify the face and then facial features and then followed by eye tracking. In this we use correlation coefficient template. The extracted eye image and template is then matched so that the system can know if the driver is sleeping or not. The blinking is then recognized and if it fall within a certain range, the alarm will go off.

Keywords—Face detection, Eye tracking, Python, OpenCV.

I. INTRODUCTION

The Driver Drowsiness system is developed using the intrusive machine vision based concepts. In this, a web cam is focused on the face of the driver for the detection his face. After detection of face it focuses on the eyes and the state of eyes like the closed eyes or opened eyes. The movement of eyes is checked for fatigue detection. And if fatigue is detected, a warning signal is send to the driver so that he can adjust himself.

In this project, I have used python for drowsiness detection. The system deals with a specific part of the body which is the face. The input video is captured using web cam placed in front of the face the driver. If face is not detected after several frames, the system will draw the conclusion of drivers sleeping. OpenCv is used for detection of face and eye with the help of 68 landmarks of face. By using Euclidean eye aspect ratio, it can detect if the eyes are closed or opened.

For detection of drowsiness, the process involves face detection, detection of eye position and then the blinking pattern of eye. The analysis of face is done by using a “Shape predictor containing 68 landmarks”. For the detection of fatigue we use a cam which is probably a webcam in this case which is set on the face of the driver for detection of face and his/her face landmarks to determine the position of eye. For this it must look at all the features of the face and eyes by using self developed image processing. Once the system find the position of eyes, the system will check if the eyes are closed or opened and the rate of close and opening which is calls blinking. If the eyes are closed for a particular time interval then the alarm will go, which will alert the driver.

II. WORKING OF THE SYSTEM

The Driver Drowsiness system is developed using the intrusive machine vision based concepts. In this, a web cam is focused on the face of the driver for the detection his face. After detection of face it focuses on the eyes and the state of eyes like the closed eyes or opened eyes. The movement of eyes is checked for fatigue detection. And if fatigue is detected, a warning signal is send to the driver so that he can adjust himself.

In this project, I have used python for drowsiness detection. The system deals with a specific part of the body which is the face. The input video is captured using web cam placed in front of the face the driver. If face is not detected after several frames, the system will draw the conclusion of drivers sleeping. OpenCv is used for detection of face and eye with the help of 68 landmarks of face. By using Euclidean eye aspect ratio, it can detect if the eyes are closed or opened.

The system will detect the face and eyes of the driver using the command. Then it will detect if eye is closed or opened. If the time interval of the eyes closed is more than the given time interval, the alarm will go on, warning the driver. if the eyes are opened then the system will goes on tracking the eyes of the driver and it will go on.
In Fig. 1, the Flow Chart is made to show the working of the project. The working is, when the user runs the program, the webcam that is placed in front of the driver starts detecting the face and then eyes. Then the program will extract the data from the webcam and then detect if the eyes are closed or opened. If the eyes are closed then the system will play sound so that the driver is awakened. If the eyes are opened then the system will go on repeating the program. The process will go on.

![Flow Chart of the above process](image)

**III. REQUIREMENTS**

The Requirements for the above program to be achieved are

1) **Anaconda Software**
   Anaconda is a free and open source distribution of Python and R programming languages for scientific computing, that aims to simplify package management and deployment.

2) **Python**
   Python 3.6 version is used which supports OpenCV and Dlib packages for face recognition. Python is comparatively easy languages and it is interpreted and is also a general-purpose language.

3) **OpenCV**
   It is an open source library of programming functions which are mainly used for machine learning and computer vision.

4) **Dlib**
   Dlib is a modern toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. Here it is used mainly for face detection and eye detection that is done by pointing the landmarks.

5) **Webcam**
   To detect face on which the programming is done.

For eye detection we used landmarks number 40 to 46 as they are the landmarks of the eyes.
6) **Play Sound**

It is used to play sound after detecting that the driver is asleep so that he can wake up.

IV. CONCLUSION

In this way we have implemented the drowsiness detection system using python successfully. The alarm will automatically go on when the driver’s eyes are closed more than the given time interval. He system must be mounted on the vehicle in the real world for the proper execution of driver drowsiness system. This system will help in lowering the accidents which are mostly done due to driver falling asleep while driving.

The system will detect drowsiness by observing eye blinking patterns that is achieved by using Euclidean distance ratio i.e. eye blinking ratio. It is most efficient technique and the whole program is much faster than by using MATLAB.

V. REFERENCE


