**Group22**

**BE Project**

**Group 22: Drowsiness Detection for Drivers.**

Problem Statement: Improper and inattentive driving is one of the major causes of road accidents. Driver’s drowsiness or lack of concentration is taken into account as a dominant reason for such mishaps. If drivers might be warned before they become too drowsy to drive safely, some drowsiness related crashes might be prevented. Hence the aim of our project is to implement a system that helps a Driver to easily access a system that detects and alerts the driver when he is drowsy to prevent facing an accident.

Scope of the Project: To prevent the driver from causing an accident, we propose to provide a system that detects whether a person is drowsy while driving, and if so, alerts him through a voice message in real time. The system also captures the image of the driver in the drowsy state and saves it to use it as proof if necessary. Initially the system was implemented to be accessed solely through the web camera present on the driver’s device. To make the system easy to access, we implemented a web portal for the driver to register into and also be able to access the system.

Software requirements:

The requirement for this Python project is a webcam through which we will capture images. You need to have Python (3.6 version recommended) installed on your system, then using pip, you can install the necessary packages.

1. OpenCV: OpenCV-Python is a library of Python bindings designed to solve computer vision problems

2. Dlib: Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. To detect and localize facial landmarks we’ll need the dlib library.

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

4. SciPy: SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering. Used so we can compute the Euclidean distance between facial landmarks points in the eye aspect ratio calculation

5. Playsound: Playsound is a “pure Python, cross platform, single function module with no dependencies for playing sounds.” With this module, you can play a sound file with a single line of code: from playsound import playsound.

6. Matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. This library will be required to plot the graph of EAR vs MAR against time.

7. IP Webcam

Technologies used to create the web portal:

1. HTML

2. CSS3

3. jQuery

4. PHP

Working of the system:

System begins with the driver registering himself on the web portal created with Html, Css, Jquery and Flask. The driver can then sign in with his credentials and click on ‘Start’.

When the start button is activated, the driver is presented with two options,

a. To monitor using Webcam

b. To monitor using phone cam

When the first option is selected, we instantiate our video stream using the webcam provided and then the facial landmarks produced by dlib is used to detect and localize the eye and mouth region of the driver specifically.

The next step is to acquire the facial landmarks. The basic idea of this technique is to locate 68 specific points on the face such as corners of the mouth, along the eyebrows, on the eyes, and so forth. It is a pre-trained detector available in the dlib library that is able to find these 68 coordinates on any face.

The system continuously monitors the driver’s states such as sleepiness, yawning and if the driver falls asleep or yawns more than 4 seconds, the systems alert the driver to switch back to normal state. After the driver’s journey is completed, he can exit from the system.

Upon exit, the images of the driver are captured if the system had caught him in a drowsy state and are saved in the ‘dataset’ folder.

Along with this, a MAR and EAR graph Vs. Time is also plotted to get a clear idea of the working of the system.

When the second option is selected, we have used an Android Application named IP Webcam. Once we click on ‘Start Server’ on the application, the system monitors the driver with the same procedure as mentioned above through this application.

References:

[1]Facial landmarks with dlib, OpenCV and Python: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

[2]Eye blink detection with OpenCV, Python, and dlib: https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/

[3]Drowsiness Detection with OpenCV: https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/

[4]Drowsiness Detection System: https://github.com/fear-the-lord/Drowsiness-Detection