DRIVER DROWSINESS DETECTION SYSTEM

Abstract - *Drivers who do not take regular breaks when driving long distances run a high risk of becoming drowsy, a state which they often fail to recognize early enough according to the experts. According to research it is proven that more than one quarter of the road accidents happen due to the drowsiness of the driver which means that more accidents are caused due to tired drivers rather than the drivers who are drunk. In state of extended speed range, a system can notify drivers about their inattentiveness and state of fatigue with the help of an alarm. There are many such alarm systems which are focusing on alerting the driver because accidents caused due to human error are increasing day by day and causing many deaths and injuries worldwide. The main reasons that cause fatal crashes and highway accidents these days are the drowsiness and sleeping of the driver. Driver drowsiness detection system is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving. This system will alert the driver when the drowsiness is detected and hence prevent the accidents from taking place.*

Key Words**:** *Drowsiness Detection, Eyes Detection, Yawn detection, Image processing, Face detection*

1. INTRODUCTION

In this driver drowsiness detection project, we will be making a drowsiness detecting system. An infinite number of people travel daily which includes the bus drivers, taxi drivers and private cars as well. It has been observed that majority of the accidents happening on the roads are due to the fatigue state and drowsiness of the driver. So, we have built a system using Python, OpenCV, pygame and matplotlib to detect the drivers face and notify him in case drowsiness is detected. The objective of this project is to build a drowsiness detection system that will detect a person’s eyes and will alert the driver when drowsiness is detected.

2.Survey existing system

Drowsiness detection systems can be divided into three main categories. Drowsiness detection is based on these three parameters. A detailed review on these measures will provide insight on the present systems, issues associated with them and the enhancements that need to be done to make a robust system.

1) Vehicle based measured system: A number of metrics, including deviations from lane position, movement of the steering wheel, pressure on the acceleration pedal, etc., are constantly monitored and any change in these threshold values may indicate that the driver is drowsy.
2) Behavioral based measured system: The behavior of the driver, including yawning, eye closure, eye blinking, head pose, etc. is captured through a camera and the driver is notified if any of these drowsiness symptoms are detected.
3) Physiological based measured system: The correlation between physiological signals ECG (Electrocardiogram) and EOG (Electrooculogram) which include pulse rate, heart beat and brain information are also used to detect the drowsiness of the driver.

The various techniques of drowsiness detection that are mostly used for the detection purpose are:

1. Electrocardiogram (ECG) and electroencephalogram (EEG):

   The heart rate varies significantly between the different stages of drowsiness and therefore ECG signal is used to determine the heart rate and is also used to detect drowsiness. The Electroencephalogram is a physiological signal which is most commonly used these days to measure drowsiness.
2. LBP (local binary pattern):

   Local binary patterns have aroused increasing interest in image processing and computer vision. Local binary pattern determines local structures of images efficiently by comparing each pixel with its neighbouring pixels. This local binary pattern technique is mostly used for detecting emotions on the face like anger, happiness, sadness and excitement. LBP is used in drowsiness detection for detecting the face of the driver.

3. Steering Wheel Movement (SWM):

The movement of the steering wheel is measured using steering angle sensor and it is commonly used as a vehicle-based safety measure for detecting the level of driver drowsiness. The driver's steering wheel behaviour is measured using an angle sensor mounted on the steering column. The number of micro-corrections on the steering wheel reduces when the driver is drowsy wherein the corrections are higher when the driver is in a normal condition.

4. Optical Detection:

   The most efficient and common implementation of an optical sensor system uses infrared or near- infrared LEDs to light the driver's eyes and are then monitored by a camera system to determine the drowsiness. Various computer algorithms analyse the blink rate and duration and then determine the drowsiness. The camera system also monitors facial features, head position and sudden head nods. Such signs of drowsiness which also include yawning are detected by the optical detection.

5. Eye Blinking Based Technique:

   In this technique the eye blinking rate and eye closure duration is measured to detect driver's drowsiness. Because when driver is sleepy at that time his/her eye blinking rate and the gaze between the eyelids are different from the normal situations so that they easily detect drowsiness. This type of system places camera remotely to acquire video and then computer vision methods are applied sequentially to localize face, eyes and eyelids and positions to measure ratio of closure. Using these eyes closure and blinking ratio we can detect drowsiness of driver. Such systems are mounted in a discreet corner of the car and they monitor for any signs of the head tilting, eyes drooping, or the mouth yawning simultaneously.

6. Yawning Based Technique:

   This technique detects driver's drowsiness based on the yawning measurement. This involves several steps which include the real time detection and tracking of driver's face. This technique also detects the mouth region and determines when the driver yawns or opens his mouth in order to gain more supply of oxygen which a strong sign for drowsiness of driver. Hence, this technique is successful to determine the drowsiness.

2.1 Limitation in existing system

Limitations in the vehicle based measured systems:

Steering Wheel Movement (SWM) is measured using steering angle sensor and it is a commonly used vehicle- based safety measure which detects the level of driver's drowsiness. The driver's steering behavior is measured using an angle sensor mounted on the steering column. Nissan, Renault and various car companies have adopted SWMs but it works in very limited situations because they can function reliably only at particular environment and are very dependent on the geometric characteristics of the road and are lesser dependent on the extent of the kinetic characteristics of the vehicle.

Limitations in behavioral based measured system:

The main limitation of using a behavioral based approach is lighting condition. Normal cameras do not perform and function well at night. In order to overcome this limitation, some researchers have used active illumination by utilizing an infrared Light Emitting Diode (LED) Although these works fairly well at night, these LEDs are considered less robust during the day. In addition to this, another limitation of behavioral measure is that in various drowsiness determining commercial products it is observed that driver state cannot be related to the driving performance and vehicle status based on behavioral measures only.

Limitations in Physiological based measured system:

The reliability, accuracy and efficiency of driver drowsiness detection by using physiological signals is very high compared to other methods. However, the intrusive nature of measuring physiological signals is an area that has to be focused on. Researchers have used wireless devices to measure physiological signals in a less intrusive manner by placing the electrodes on the body and obtaining signals using wireless technologies like Zigbee, Bluetooth in order to overcome this but it seems very expensive for the general public.

3. Problem statement and objective

Current drowsiness detection systems monitor the driver's condition and requires complex, computation and expensive equipment which is not comfortable to wear during driving and is not suitable for driving conditions for e.g.: Electroencephalography (EEG) and Electrocardiography (ECG), i. e. detecting the brain frequency and measuring the rhythm of heart. A drowsiness detection system which uses a camera placed in front of the driver is more suitable to be use. A drowsiness detection algorithm that is reliable and accurate will determine the physical signs that will indicate drowsiness. In improper Lighting intensity when the driver tilts his face left or right, problems occur to detect the eyes and the mouth region. Therefore, this project aims to analyze all the previous research and methods, and hence propose a method to detect drowsiness by using video or webcam. It analyses the video and images that have been recorded and comes up with a system that can analyze each frame of the video separately.

Calculation of various parameters to detect the drowsiness:

1. Using the dlib's 68-point facial landmark detector shown in the figure below we will detect the region of interests that is the eyes, upper lip and the lower lip.

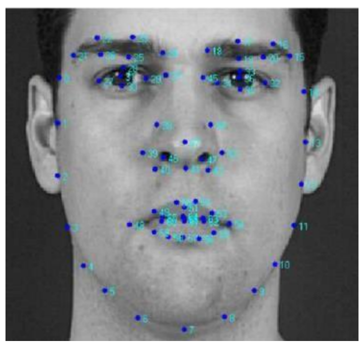

Fig -1: 68-point facial landmarks

2. After detecting the co-ordinates of the eyes, we need to find the EAR value for the eyes. Eye aspect ratio or the EAR value can be computed according to the position of eyes using the formula given below:

   $EAR = |P2 - P6| + |P3 - P5| / 2 |P1 - P4|$

   where Pi, i = 1, 2, . . ., 6 is the coordinate of eye landmarks

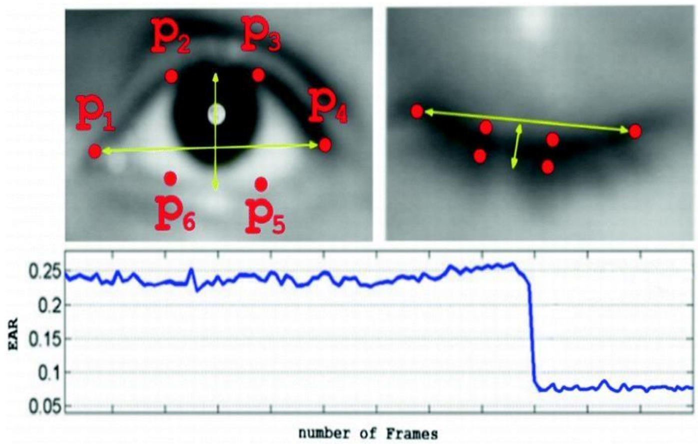

Fig -2: Calculation of Eye aspect ratio (EAR)

3.1 Objective:

The project focuses on these various objectives, which are:

- To suggest ways to detect drowsiness while driving.
- To observe eyes from the video images of participants in the experiment of driving simulation conducted by MIROS that can be used as an indicator of drowsiness.
- To investigate the physical changes that take place during drowsiness.
- To develop a system that use eyes closure as a way to detect drowsiness.

4 Scope:

In this project, we will focus on these following procedures:

- Basic concept of drowsiness detection system
- Observe and understand the various signs of drowsiness
- Determine the drowsiness from these parameters:
- Eye blink rate, Eye closure and Eye aspect ratio value (EAR)
- Data collection and measurement of the parameters.
- Integration of the methods chosen to detect drowsiness.
- Coding development and testing of the system.
- Complete testing and improvement of the software.

5 Design details:

The use case diagram below explains the design of the working of the driver drowsiness detection system.

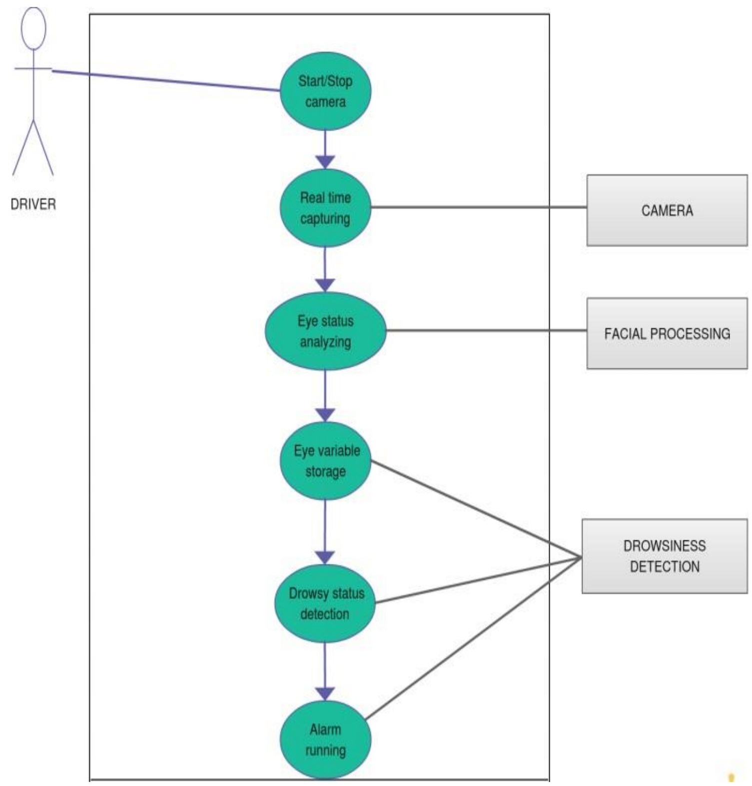

Fig -5 : Use case diagram of detection system

6 Conclusion:

During the final year project investigating the signs of drowsiness and collecting the data from the video of the experiments has been the main job scope. This data will be used as parameters to develop the system in detecting drowsiness. The project objectives have been achieved which is to research and review the several existing techniques used for the driver drowsiness detection system.

For the progress of project, we have started developing the algorithm to detect the drowsiness and also implemented few techniques in this project which have been found in the existing systems that we researched about. Further adjustment of theses algorithms and the techniques is made so that the detection system offers best functionality, accuracy and efficiency as compared to the existing systems.

Code explanation:

Explanation for each line of the code:

1. `import cv2`: Imports the OpenCV library for computer vision tasks.
2. `import os`: Imports the os module for interacting with the operating system.
3. `import numpy as np`: Imports the NumPy library for numerical computations.
4. `from keras.models import load_model`: Imports the `load_model` function from Keras for loading a pre-trained model.
5. `from pygame import mixer`: Imports the `mixer` module from Pygame for audio playback.
6. `mixer.init()`: Initializes the Pygame mixer.
7.`sound=mixer.Sound("C:/Users/subha/Downloads/Drowsinessdetection/Drowsiness detection/alarm.wav")`: Loads the audio file for the alarm sound.
8.`face= cv2.CascadeClassifier("C:/Users/subha/Downloads/Drowsiness detection/Drowsiness detection/haar cascade files/haarcascade_frontalface_alt.xml")`: Loads the Haar cascade classifier for detecting faces.
9.`leye=cv2.CascadeClassifier('C:/Users/subha/Downloads/Drowsiness detection/Drowsiness detection/haar cascade files/haarcascade_lefteye_2splits.xml')`: Loads the Haar cascade classifier for detecting left eyes.
10.`reye=cv2.CascadeClassifier('C:/Users/subha/Downloads/Drowsiness detection/Drowsiness detection/haar cascade files/haarcascade_righteye_2splits.xml')`: Loads the Haar cascade classifier for detecting right eyes.
11. `lbl = ['Close', 'Open']`: Initializes a list containing the labels for closed and open eyes.
12.`model=load_model("C:/Users/subha/Downloads/Drowsiness detection/Drowsiness detection/models/cnnCat2.h5")`: Loads the pre-trained Keras model for drowsiness detection.
13. `path = os.getcwd()`: Retrieves the current working directory path.
14. `cap = cv2.VideoCapture(0)`: Initializes the video capture object to capture video from the default camera.
15. `font = cv2.FONT_HERSHEY_COMPLEX_SMALL`: Specifies the font style for drawing text on the frame.
16. `count = 0`: Initializes a counter variable.
17. `score = 0`: Initializes a score variable.
18. `thicc = 2`: Initializes the thickness of the bounding box.

19. `rpred = np.array([99])`: Initializes a NumPy array to store the prediction for the right eye.
20. `lpred = np.array([99])`: Initializes a NumPy array to store the prediction for the left eye.

The remaining lines of code constitute a while loop that performs the drowsiness detection task frame by frame. Each iteration of the loop does the following:

- Reads a frame from the video capture object.
- Converts the frame to grayscale.
- Detects faces, left eyes, and right eyes in the grayscale frame using the Haar cascade classifiers.
- Processes the detected right and left eyes, including resizing and normalization.
- Makes predictions on the processed right and left eyes using the loaded model.
- Determines the eye status (open or closed) based on the predictions.
- Updates the score based on the eye status.
- Displays the eye status and the score on the frame.
- If the score exceeds a threshold, an alarm sound is played and a bounding box is drawn around the frame.
- Shows the frame on the screen.
- Checks for the 'q' key press to exit the loop.

Finally, the video capture object is released, and all windows are closed.