Drowsy Driver Detection and Alertness System

Deepanshu Mehta
Department of Information Technology
Panjab University (UIET), Chandigarh, India

Abstract— In this paper, drowsy driver detection and alertness system is described using the concepts and algorithms of computer vision. In this paper, the main focus is to detect the face and eyes of the driver and then alerting the driver when found drowsy. In this paper, the drowsy state of the driver is described only based on the opening or closing state of the eyes of the driver. This system is developed using a non-intrusive machine-vision-based concepts. This paper also includes the algorithm which has been used for implementing this system.

Keywords: Drowsiness Detection, Computer Vision, Eye Detection

I. INTRODUCTION

Driver fatigue is a significant factor in a large number of vehicle accidents. Fatalities have occurred as a result of car accidents related to driver inattention, such as distraction, fatigue, and lack of sleep. Studies and experiments have substantiated the fact that driving performance deteriorates with increased drowsiness. The US National Highway Traffic Safety Administration has estimated approximately 100,000 crashes each year caused mainly due to driver fatigue or lack of sleep.

Autonomous systems designed to analyze driver exhaustion and detect driver drowsiness can be an integral part of the future intelligent vehicle to prevent accidents caused by sleep. A variety of techniques have been employed for vehicle driver fatigue and exhaustion detection [1]. Driver operation and can be implemented by capturing the emotions of the driver by using computer vision and deep learning algorithms. These are non-intrusive ways of driver drowsiness detection. In this application, we will detect how long a given person's eyes have been closed. If their eyes have been closed for a certain amount of time, we'll assume that they are starting to doze off and play an alarm to wake them up and grab their attention [2].

II. DROWSINESS CHARACTERISTICS

Driver fatigue is a significant factor in a large number of vehicle accidents. Fatalities have occurred as a result of car accidents related to driver inattention, such as distraction, fatigue, and lack of sleep. Studies and experiments have substantiated the fact that driving performance deteriorates with increased drowsiness. The US National Highway Traffic Safety Administration has estimated approximately Drowsiness is a physiological state with a tendency to fall asleep[2]. Technically, drowsiness is different from the fatigue that is the lack of willingness to continue performing the same activity. Fatigue occurs by performing tasks that are always performed in the same way using the same muscle groups, their repetition rate is high and are usually performed by adopting forced postures such as monitoring a screen. A person may be fatigued without being drowsy, but conditions that produce fatigue such as driving cars over great distances unmask the presence of physiological drowsiness but do not cause fatigue[3][4].

III. MATERIALS AND METHODS USED

This section describes different aspects of the system considered in its implementation; they include the functional requirements as well as the tools used and devices selected for system testing in different study cases[6].

Softwares used are Anaconda, python, kernel, jupyter lab and libraries used are numpy, pandas, matplotlib, Scipy, sklearn, TensorFlow, OpenCV, dlib, Keras and Thanos. System is implemented using the following flowchart as in Fig.1:

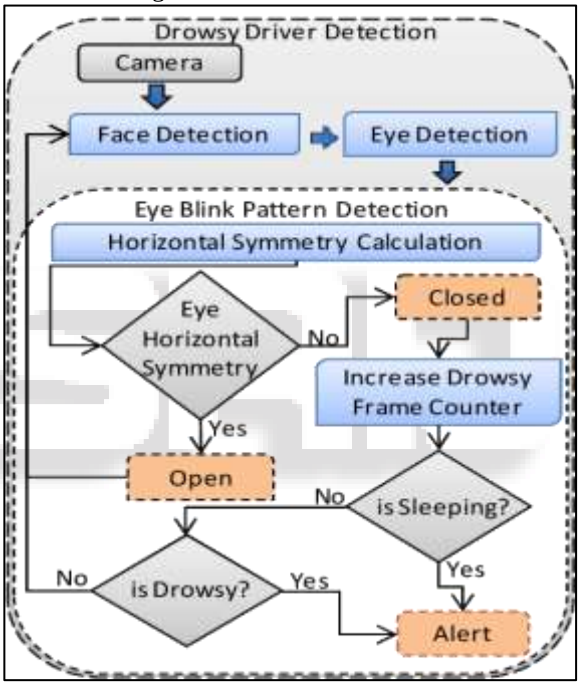

Fig. 1: Drowsy Driver Detection Flow Graph

IV. ALGORITHM USED

Eye blinks can be detected by referencing significant facial landmarks. Many software libraries can plot significant facial features within a given region of interest. Python's dlib library uses Kazemi and Sullivan's One Millisecond Face Alignment with an Ensemble of Regression Trees to implement this feature[3][7].

The program uses a facial training set to understand where certain points exist on facial structures. The program then plots the same points on region of interests in other images, if it exists. The program uses priors to estimate the probable distance between key-points. The library outputs a 68-point plot on a given input image as shown in Fig.2.

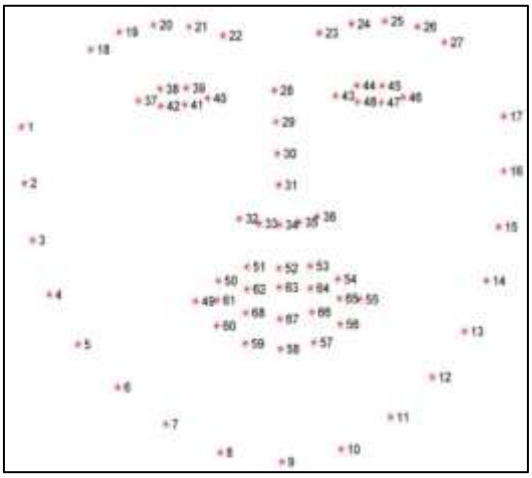

Fig. 2: Pre-defined facial landmarks

For eye blinks, we need to pay attention to points 37-46, the points that describe the eyes. In Real-Time Eye Blinking Using Facial Landmarks [2], Soukupová and Čech derive an equation that represents the Eye Aspect Ratio. The Eye Aspect Ratio is an estimate of the eye-opening state.

Based on Figure, the eye aspect ratio can be defined by the below equation.

"The Eye Aspect Ratio is a constant value when the eye is open, but rapidly falls to 0 when the eye is closed." The person's eyeblinks are obvious. A program can determine if a person's eyes are closed if the Eye Aspect Ratio falls below a certain threshold.[5][7]

Eye aspect ratio is calculated using the following formula:

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

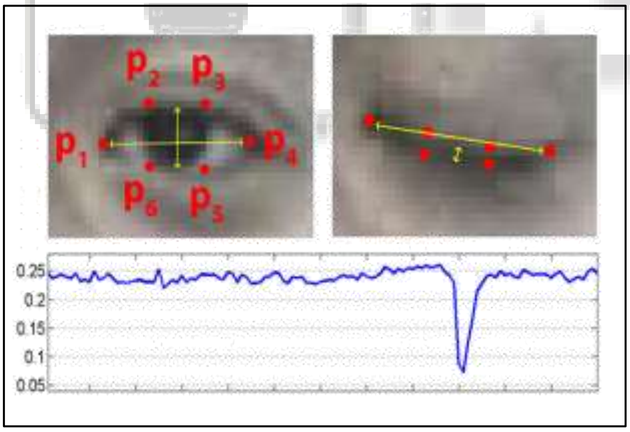

Fig. 3: Graph showing threshold value equal to 0.05 approx.

From plot shown in Fig.3, we can infer that if EAR value is too low i.e. 0.05 or below then eyes of the person would be considered as closed.

V. ALGORITHM USED

A. Training the convolutional neural network for eyes Module:

The goal of this module is to define CNN architecture and then train it. The conventfactory.py file in CNN sub package of PyImage search will store a class that enables us to access our CNN architectures. The train_network.py will enable us to test our network on images[5].

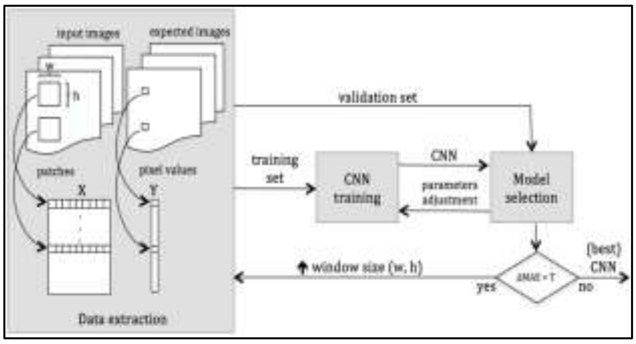

Fig. 4: Training of CNN Flow Graph

The above diagram shows the basic logic behind training CNN. Once the CNN is trained then the program would be able to determine the eye region of the human face and it will display a green lining around the eyes. After this, our next step would be to check whether the eyes are closed or not according to the algorithm mentioned above.

B. Eyes Detection Module:

This module will detect the eyes of the driver and through this, we can determine the drowsiness state of the driver by checking if his/her eyes are opened or closed. If the eyes are closed for 20 consecutive frames then the driver is in drowsy state and an alarm will be activated [2].

C. Drowsy Detection Alert Module:

This is the module where our actual application runs. This module will make sure that whenever the driver enters into drowsiness state then an alarm will be activated if the eyes of driver are closed for 3 seconds or for 20 consecutive frames [7].

VI. RESULTS

This section presents the results on the detection of visual indicators of drowsiness. Collecting the data set to properly evaluate the system is a challenge, this is because dangerous drowsiness events are not guaranteed to occur during daily driving for application testing. These readings are based on our own samples.

Pic 1: In this pic, our application is detecting the eyes of the driver

Pic 2: In this pic, our application is alerting the driver

	P 1	P 2	P 3	P 4	P 5	P 6
Sample 1	7.880	9.775	8.258	6.976	7.438	9.743
Sample 2	8.755	7.864	8.465	9.643	7.534	8.537
Sample 3	8.622	7.849	8.489	9.166	7.738	8.922
Sample 4	7.827	9.381	8.299	7.881	8.292	7.282

Sample 5	4.288	6.822	5.272	6.282	5.811	6.888
Sample 6	2.976	4.977	3.211	3.767	5.785	2.007
Sample 7	4.788	6.722	5.282	6.282	7.821	6.828
Sample 8	8.282	7.228	7.928	8.219	7.119	8.181

Table 1: Values taken from 8 Samples for Calculating EAR

From the Table 1 readings, we will calculate eye aspect ratio using the formula 1 which will then be stored in a variable. A threshold value will be already defined which may be equal to 0.05 and if the value goes below this value then according to that drowsy state will be determined. If the eyes are closed for 20 consecutive frames or for 3 seconds then an alarm would be activated.

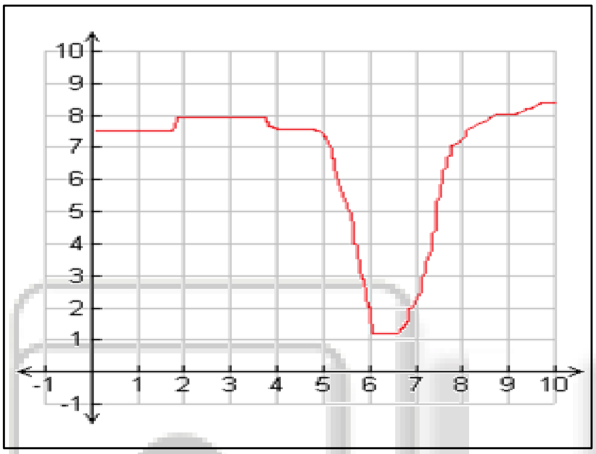

Fig. 6: graph representing EAR values for above samples

The graph shows EAR vs no. of samples graph.

$$EAR = |p2 – p6| + |p3 – p5| / 2|p1 - p4|$$

We calculate the value of EAR by putting values of p1, p2, p3, p4, p5 and p6 from the data point table got above from the code itself. After getting the value of EAR from each sample, we plot it on the graph[6].

As we can see by checking the first five points, the eyes must be opened completely. Eyes opened were opened completely because of which the EAR values were high in the graph. In the sixth and seventh sample, we can see the EAR value quickly drops which implies the eyes were closed. To visualize, read the following cases:

1) If EAR value is high, it means the numerator in the above equation is high, which implies distance between p2 and p6 is high and also distance between p3 and p5 is high. As p2 and p3 is on upper eyelid and p5 and p6 is on the lower eyelid, it concludes that distance between upper and lower eyelid is high and means the eyes are opened completely.
2) If EAR value is low, it means the numerator value is low means the distance between p3 and p5 is low. As p2 and p3 is on upper eyelid and p5 and p6 is on the lower eyelid, it concludes that distance between upper and lower eyelid is very low and means the eyes is closed completely which leads to beeping of the alarm.

VII. CONCLUSIONS

The study has shown promising results in applying the vehicular driver surveillance based on computer vision techniques. The implemented system allows an efficient detection of the indicators that appear in drowsiness, as long as the measurements are carried out under the established conditions. The above results show that the system is working even if the driver is wearing spectacles. The increase in the processing characteristics made it possible to develop an application of computer vision, capable of detecting the face and visual indicators present in a person who suffers from drowsiness such as the state of the eyes.

REFERENCES

[1] A. Patil, A. Patil, S. Shenoy, S. Prof, and S. Pansambal, “Literature Survey : Drowsiness Detection using Computer Vision Technology,” no. January, pp. 191–193, 2016.
[2] I. R. Nair, “A Survey on Driver Fatigue-Drowsiness Detection System,” Int. J. Eng. Comput. Sci., vol. 5, no. 11, pp. 19237–19240, 2016.
[3] A. D. Lakshmi, B. Nivetha, A. J. Kumar, and R. Malar, “The Literature Survey Based On Car Safety System That Spots Driver Errors,” Int. J. Eng. Res. Dev., vol. 11, no. 11, pp. 42–44, 2015.
[4] P. Choudhary, R. Sharma, G. Singh, S. Das, and S. G. Dhengre, “A Survey Paper On Drowsiness Detection & Alarm System for Drivers,” Int. Res. J. Eng. Technol., pp. 1433–1437, 2016.
[5] M. A. Assari and M. Rahmati, “Driver drowsiness detection using face expression recognition,” 2011 IEEE Int. Conf. Signal Image Process. Appl. ICSIPA 2011, pp. 337–341, 2011.
[6] A. Ullah, S. Ahmed, L. Siddiqui, and N. Faisal, “Real Time Driver’s Drowsiness Detection System Based on Eye,” Int. J. Sci. Eng. Res., vol. 6, no. 3, pp. 125–131, 2015.
[7] P. Fields, “Computer Vision-Based Method for Automatic Detection of Crop Rows in Advances in Intelligent Systems and Computing,” no. January, 2018.