

Driver Drowsiness Detection Using OpenCV, Dlib and Python

Anandu Santhosh¹, Arya Manohar², Dishnu S Dinesh³, Sravan Soman⁴,
Usha Gopalakrishnan⁵, Vinod P R⁶

U.G Student, Department of Computer Engineering, College of Engineering, Chengannur, Kerala, India¹

U.G Student, Department of Computer Engineering, College of Engineering, Chengannur, Kerala, India²

U.G Student, Department of Computer Engineering, College of Engineering, Chengannur, Kerala, India³

U.G Student, Department of Computer Engineering, College of Engineering, Chengannur, Kerala, India⁴

Associate Professor, Department of Computer Engineering, Musaliar College of Engineering and Technology,
Pathanamthitta, Kerala, India⁵

Assistant Professor, Department of Computer Engineering, College of Engineering, Chengannur, Kerala, India⁶

ABSTRACT: In today's world, the number of road accidents that occur is increasing very rapidly. Some of these road accidents are minor whereas some of them may seriously injure people or even take their lives. These road accidents may involve collision of vehicles with each other or crashing the vehicles on to the buildings or others. Road accidents may even result in the death of people. It is estimated that the total deaths due to road accidents in India is around 1,50,000 per year which is approximately 400 accidents per day. Almost 3 lives are lost in every 10 minutes due to road accidents. This shows that reducing the rate of the road accidents is necessary. The proposed method uses python along with two libraries OpenCV and dlib for facial landmark detection. A mathematical value Eye Aspect Ratio (EAR) is then calculated for determining whether the eyes are closed or not.

KEYWORDS: Drowsiness detection, EAR, Facial landmark

I. INTRODUCTION

Driver drowsiness detection using OpenCV, dlib and python uses mainly the concept of a mathematical value called Eye Aspect Ratio (EAR), which is a simple and effective approach. The proposed method has several advantages over the traditional MATLAB based face detection methods:

A) Speed: MATLAB is fabricated from java which in turn is fabricated from C. Therefore, the computer initializes by interpreting the code and converting it into java and then finally executes the script, when a code is scripted and run on MATLAB. But in OpenCV, C/C++ library functions are used, which directly provides the computer with the machine language code and hence helps in faster execution. With OpenCV, there is more utilization of time and resources in image processing and less in interpreting.

B) Portability: Any device which runs on C can run OpenCV because OpenCV runs on C. It can toil well with Windows, mackintosh or Linux.

C) Cost: MATLAB is very much expensive than OpenCV. MATLAB costs around USD2150 but OpenCV is free of cost. The base MATLAB has commercial, single user License and so it is expensive. But OpenCV is a BSD license, so it is free of cost.

Paper is organized in the following manner. Section II presents the literature survey conducted for this paper. Section III describes the concepts and actual methodology for driver drowsiness detection. Section IV presents experimental results, showing the images of drowsiness detection for various people. Finally, Section V presents conclusion.

II. LITERATURE SURVEY

The literature survey was conducted on various face detection methods, existing drowsiness detection methods and image processing techniques. [1] uses the front camera of the driver's mobile phone, which is placed on the windshield of the car and constantly uses the front camera to analyze the behavior of the driver, i.e., whether the driver's eyes are closed or not. The front camera constantly captures the image of the driver and with the help of facial expression detection checks whether the eyes of the driver are closed or not. The image captured is compared with thousands of previously stored expression and to give the best result, machine learning algorithms are used. A timer is started after it is detected that the eyes are closed and after a particular time an alert is generated. This alert is used to trigger an external low powered buzzer with the help of bluetooth signal sent via mobile of the driver. At first the phone is placed in the windshield using a holder so that driver need not hold it all times and also it remains stable. The images are captured at a rate of 5 frames per second. When the face is detected, about 30 points are placed on the face like eyes, nose etc. Captured images are then compared with images previously stored in the server and it returns magnitude of different facial expressions in the drowsy state. Server returns a value in the range 0-100. Now whenever this value reaches 100 (fully closed), a timer is started. The timer is stopped if the eyes are opened in between. If the timer crosses a time period of 3sec, an alert is generated to wake up the driver. The phone rings as well as an alarm is triggered.

[2] uses MCT AdaBoost classifier for the face detection. First, image frame from the camera is retrieved by video query process. The input image is pre-processed by Gaussian filtering for removing noise. Using MCT AdaBoost classifier, the proposed algorithm detects the face in the preprocessed image. MCT feature is robust to the lighting conditions [5,6].

In [3], the main concept is detecting the driver's face, and set it to Region Of Interest (ROI). Next use ROI to detect eyes and mouth. This process is done by getting input from infrared 2D camera and processing by MATLAB R2015a. In flow chart there are five steps, namely, image acquisition, face detection, eye detection, mouth detection, and eyes closure and yawning detection.

- (a) Input image from infrared: At first, record driver video from driver's seat, using a camera, facing approximately 15 degrees to driver's face. Frame rate is set as 25 frame per second and input scale is 200×200 pixel. Made of infrared camera is MD-tech series 02. It uses 850nm light range.
- (b) Face detection: Haar-like feature is a face detection method. This method was invented by Paul Viola and Michael Jones [7]. This step is used for finding driver's face. Harr-like feature is the method to classify face by determining differential of shading rectangle and normal rectangle, and comparing it with threshold and polarity. Following equation determines sum of rectangle area:

$$i_s(x_1, x_2, y_1, y_2) = \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} i(x, y)$$

where $i(x,y)$ means strength of shading x and y mean sum of shading value. x_1 , x_2 , y_1 and y_2 are positions of rectangle's corner. In this method after detecting driver's face, author will set it to ROI, and another area will be set to black.

- (c) Split image to upper-half and lower-half: Face image is split to two halves. This method is for limiting the scope of detecting driver's eyes and mouth.
- (d) Detecting driver's eyes and mouth: This step use for detecting driver's eyes and mouth by using Haar like feature. With the camera settings the driver's image will be little tilted. So this step will rotate image approximately by 3 degrees to the left and right.
- (e) Detecting eyes closure and yawning: To classify data author use Support Vector Machine (SVM). Before train data using SVM, normal images are turned into vector data by extracting histogram. Histogram Of Oriented Gradients (HOG) is method to separate images to many cells and collecting histogram from every single

gradient, and calculate scale and vector of that cell. SVM can be used for classifying between closed or open eyes and between normal and yawning mouth.

In [4], first the video is recorded using a webcam. To capture front face image, the camera will be positioned in front of the driver. The frames are extracted from the video to obtain 2-D images. Using Histogram Of Oriented Gradients (HOG) and linear Support Vector Machine (SVM) for object detection[8], face is detected in the frames. After detecting the face, facial landmarks like positions of eye, nose, and mouth are marked on the images. Eye aspect ratio, mouth opening ratio and position of the head are calculated from the facial landmarks, and using these features and machine learning approach, a decision is obtained about the driver drowsiness. An alarm will be sent to the driver to alert him/her, if drowsiness is detected.

III. METHODOLOGY

This paper focuses on two libraries OpenCV and dlib and also on a mathematical concept called eye aspect ratio (EAR):

(a) OpenCV

OpenCV means Open Source Computer Vision Library. It is an open source library used for computer vision and machine learning functions. OpenCV is written in C++ and has cross platform support. Its C++, Python, Java and MATLAB interfaces support Windows, Linux, Android, iOS and Mac OS. OpenCV was developed to provide a common tool for developing computer vision applications and to increase the use of machine perception in the commercial products. OpenCV aims towards real-time vision applications.

(b) Dlib

Dlib is a general purpose open-source software library and has cross platform support. It is a tool of C++ which consists of machine learning algorithms and several tools for creating complex software which can be used to solve real world problems. Its development has began in 2002 and it has grown to include a wide variety of tools. It has features to support networking, threads, graphical user interfaces, data structures, linear algebra, machine learning, image processing, data mining, XML and text parsing, numerical optimization, Bayesian networks, and many more. Since it is open-sourced, it can be used in any app development free of cost.

(c) Eye Aspect Ratio (EAR)

Eye aspect ratio is the ratio of height to width of the eye.

When a face is given as input, python's dlib library outputs 68 points on the face as shown in the figure given below:

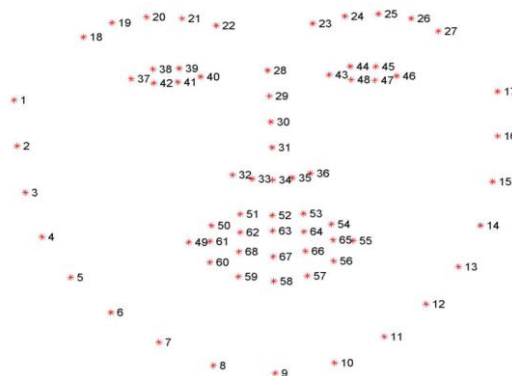


FIGURE 1: FACIAL LANDMARKS

For eye region we need to consider the points from 37-46. Eye aspect ratio can be defined by the following equation:

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

EAR is approximately constant while the eye is open and will immediately fall to zero when the eye is closed.

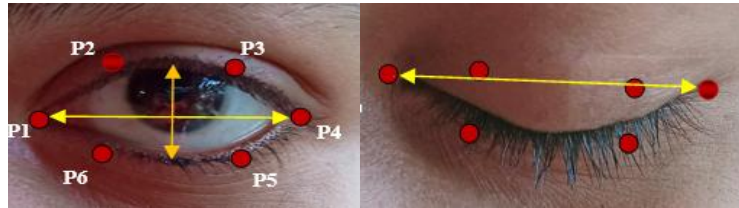


FIGURE 2: (A) AN EYE THAT IS FULLY OPEN WITH THE EYE FACIAL LANDMARKS PLOTTED (B) AN EYE THAT IS CLOSED

The eye aspect ratio will be monitored to see if the value falls but does not increase again, thus verifying that the person has closed their eyes. The following figure shows the block diagram for drowsiness detection:



FIGURE 3: BLOCK DIAGRAM

- (a) Image acquisition: Image is captured by the camera as frames.
- (b) Facial landmark detection: Using python's dlib library facial land mark detection is done. After that the eye region is extracted.
- (c) EAR calculation: EAR is calculated to determine if the eye is closed or not.
- (d) Alarm: If the eye is closed for sufficiently long period of time an alarm is played. First of all, two constants are defined: threshold value and frame number limit. Threshold is fixed as 0.3 and the frame number limit is fixed to be 48(based on research works and experimental results). A variable value Count is initialised to 0. First, we'll setup a camera and check for faces. If a face is found, facial landmark detection will be applied and the eye regions will be extracted. Now we have the extracted eye regions and this can be used to compute the eye aspect ratio to determine if the eyes are closed. Whenever the EAR is less than threshold, Count will be incremented by 1. Otherwise Count value will remain 0. If the Count value exceeds the frame number limit given, then an alarm will be sounded to wake up the driver. Else next frame will be captured by the camera. In short, whenever EAR remains lower than the given threshold for a sufficiently long period of time, the proposed algorithm assumes the driver to be drowsy and an alarm will be played. The flowchart for the proposed method is shown below. TH denote the threshold value which is initialised at the start of the algorithm and frnum denote the limit on the number of frames for which count remains less than TH.

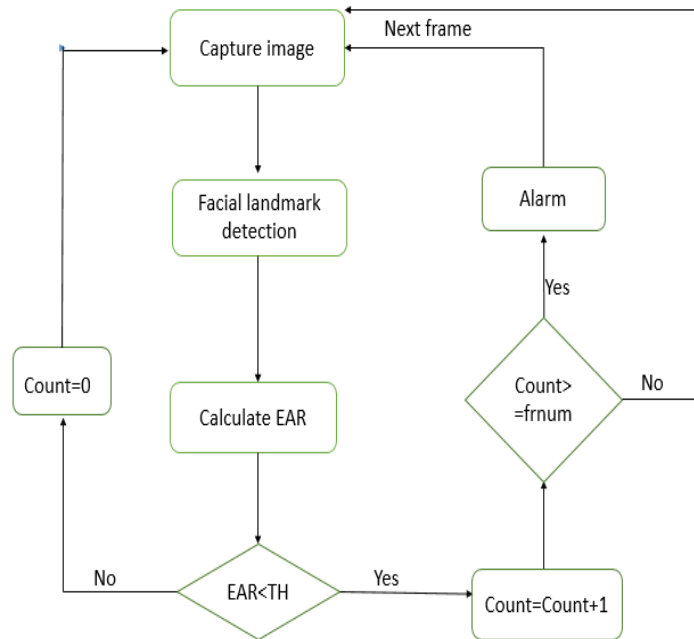


FIGURE 4: FLOW CHART

IV. EXPERIMENTAL RESULTS

The experiment was done on a laptop with a webcam used for live recording the driver. Figure below shows the results of the experiments conducted on two different people. The programming was done entirely in python language. Two figures are shown for each person. The first figure, for each person, indicates eye detection (the green coloured bounding boxes indicates eye detection). EAR value is calculated lively and shown on the frame as the video is captured. Whenever EAR falls below the threshold value and remains as such for the specified number of frames (frnum), an alarm will be played indicating drowsiness (a red coloured text will also be displayed on the frame), which is shown by the second figure (for each person). The method can be implemented as a system using raspberry pi and can be implemented in vehicles.

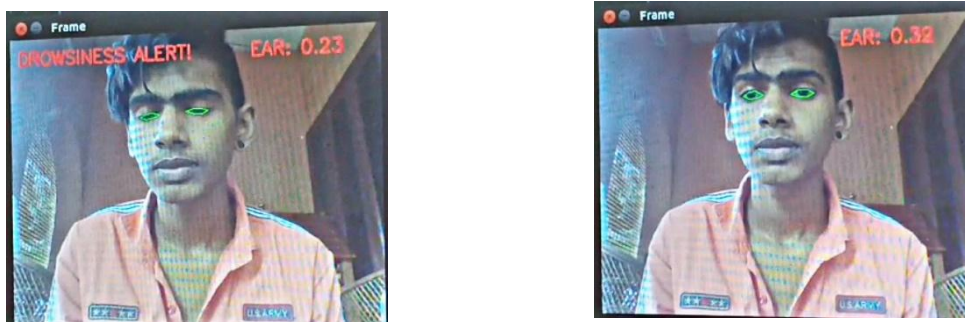


FIGURE 5: TEST 1 (A) EYE DETECTION (B) DROWSINESS DETECTION



FIGURE 6: TEST 2(A) EYE DETECTION (B) DROWSINESS DETECTION

V. CONCLUSION

This paper has proposed a methodology for implementing a fast and effective driver drowsiness detection system. Two libraries OpenCV and dlib and a mathematical concept called EAR was used for this purpose. The method was successfully tested on several people and was proved to be effective.

REFERENCES

- [1] Hemant Kumar Dua , Sanchit Goel , Vishal Sharma, " Drowsiness Detection and Alert System", IEEE International Conference on Advances in Computing, Communication Control and Networking, 2018, DOI: 10.1109/ICACCCN.2018.8748448
- [2] Jang Woon Baek, Byung-Gil Han, Kwang-Ju Kim, Yun-Su Chung, Soo-In Lee, " Real-time Drowsiness Detection Algorithm for Driver State Monitoring Systems", IEEE Electronics and Telecommunications Research Institute, 2018, DOI: 10.1109/ICUFN.2018.8436988
- [3] Wisaroot Tipprasert, Theekapun Charoenpong, Chamaporn Chianrabutra, Chamaiporn Sukjamsri, " A Method of Driver's Eyes Closure and Yawning Detection for Drowsiness Analysis by Infrared Camera", IEEE First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP),2019, DOI: 10.1109/ICA-SYMP.2019.8646001
- [4] Kumar A., & Patra R, "Driver drowsiness monitoring system using visual behaviour and machine learning", IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE),2018, DOI: 10.1109/ISCAIE.2018.8405495
- [5] C. Kublbeck and A. Ernst, "Face detection and tracking in video sequence using the modified census transformation," Image and Vision Computing, Vol. 24, pp. 564-572, 2006.
- [6] K. Ban, J. Kim, and H. Yoon, "Gender Classification of Low- Resolution Facial Image Based on Pixel Classifier Boosting," ETRI Journal vol.38, no. 2, pp. 347-355. Apr. 2016.
- [7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I
- [8] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", IEEE conf. on CVPR, 2005