IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

IJARSCT
Impact Factor: 6.252

Volume 2, Issue 5, May 2022

Developing Driver Safety System by Detecting and Monitoring Drowsiness of Driver

Prasanna P. Dhole[1], Sahil S. Kale[2], Avinash P. Mahashabde[3], Arpit P. Wath[4], Shivam P. Shete[5], Nidhi G. Gupta[6]

Students, Department of Computer Science and Engineering[1,2,3,4,5]

Professor, Department of Computer Science and Engineering[6]

Sipna College of Engineering & Technology, Amravati, Maharashtra, India

Abstract: *Drowsy driving is the state of a person driving in a sleepy condition. Drowsy driving is a very serious and dangerous issue that can lead to fatal accidents on the road. Not only does drowsy driving risk the life of the driver behind the wheel, but also the lives of people boarding the same vehicle and people along the road. Every year, thousands of accidents are caused by drowsy driving. To prevent such accidents, a system has to be developed that detects and monitors the drowsiness and sleeping condition of the driver and gives an alert to the driver. This paper discusses the solution to the same discussed problem, which is an AI (artificial intelligence) based system that detects and monitors the drowsy and sleeping conditions of the driver. This driver safety system is designed using Python, OpenCV, and the Dlib model. OpenCV is a computer vision library that maintains a directory of pre-trained Haar cascades which are used in face detection. The Dlib library's pre-trained facial landmark detector is used to estimate the location of 68 (x, y)-coordinates that monitor the aspect ratios of eyes and mouth to detect drowsiness. The proposed system is tested in real time with inputs from cameras and videos.*

Keywords: Python, Dlib, OpenCV, Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), drowsiness detection, face detection, facial landmark

I. Introduction

Drowsy driver detection is the most important process to prevent any road accidents in the current epoch. So the purpose of the project is to create a system that can identify a user's drowsiness and inform them so that an accident can be avoided. In addition, this system will identify if the user is yawning and will notify the user accordingly. This system can now be installed on any machine. This technology determines whether or not a user is drowsy in real time. We need to recognize the user's face. As a result, the Viola Jones method consumes less processing resources, is faster, and is more precise. The Viola Jones algorithm discussed in [1] recognizes the face and turns it to grayscale. To identify whether user sleeping or not, we must first determine whether or not the user's eyes are open. The suggested approach utilizes the Eye-Aspect-Ratio to detect this (EAR). When the eyes are open and closed, the average eye aspect ratio is 0.339 and 0.141, respectively as explained in [2]. However, based on the EAR formula, it can be determined that if the EAR value is suddenly reduced, the driver is most likely in the process of closing his eyes. As a result, the falling average EAR value from 0.339 to 0.141 signals the user's eyes blinking or shutting and alerts them till they open their eyes. We must now locate the eye landmarks in the face in order to calculate the EAR. Referring [3] we'll utilize Dlib's 68 facial landmark model, which is a pre-trained model that can be simply used with Python, to find these landmarks. It is used to calculate the location of 68 coordinates (x, y) that map a person's facial points. The user's yawn can also be used to detect their level of drowsiness. The Mouth Aspect Ratio is used to detect yawns (MAR) [4]. We need to know how far apart the user's upper and lower lips are. When a person is talking, the distance will be within a certain range, but when the person yawns, the distance will be significantly more than the range. To determine the distance between two lips, we must first identify the lip's landmarks, which we will do using the Dlib's face landmark model once more. Then we'll just calculate the distance between the top lip's midpoint and the lower lip's midpoint. If the distance between the two points is more than the threshold, then it will send an alert to the user. OpenCV is an image processing and computer vision library that provides a predefined repository of Haar cascades that are used for face detection in the proposed system. Haar cascades are notoriously prone to false-positives. Therefore, we have used another image processing library Dlib, which might be slower than OpenCV Haar cascade, but Dlib's face landmark detector method is more accurate. OpenCV is used for image processing in the proposed system. The below Fig 1. represents the flow of drowsiness detection.

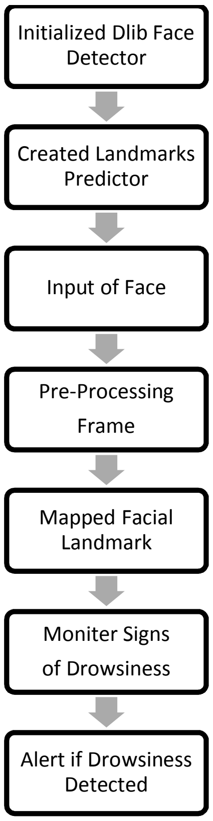
Initialized Dlib Face Detector
Created Landmarks Predictor
Input of Face
Pre-Processing Frame
Mapped Facial Landmark
Moniter Signs of Drowsiness
Alert if Drowsiness Detected

Figure 1: Block diagram of real-time detection of drowsiness.

II. Face Detection & Processing Methodologies

A face detector must be able to determine whether or not an image of any size contains a human face and, if so, where it is located. The algorithm must minimize the actual prior probability of a given image having a face. In order to attain acceptable performance, both the false negative and false positive rates must be reduced. The following methods discussed can be used for image detection and processing.

VIOLA-JONES ALGORITHM

The Viola-Jones algorithm, created by Paul Viola and Michael Jones in 2001, is an object-recognition framework that enables for real-time detection of visual properties. The Viola-Jones algorithm discovers the location on the coloured image after detecting the face on the grayscale image. Viola-Jones draws a box and searches within the box for a face. Viola-Jones, despite being an outdated framework, is highly powerful, and its application has proven to be particularly significant in real-time face identification [5], employing the approaches outlined below [6].

- HAAR-based features
- Integral Image Formation
- AdaBoost Technology
- A classifier cascade

In a HAAR-based feature representation, features are chosen based on pixel intensities. The values of the pixels are not taken into account. The image and several HAAR templates are scalar products of HAAR-based features. For feature calculation, integral image creation is used. It simply considers the image's four corners. The required characteristics are chosen using adaptive boosting (AdaBoost). The algorithm's computing time is cut in half thanks to the usage of adaptive boosting. To build a strong classifier chain, a cascade of classifiers is used. A command is provided by the OpenCV library.

HAAR CASCADE METHOD

OpenCV includes a large number of pre-trained classifiers. There are classifiers for smiles, eyes, face, nose, and so on. Some of these functions can be used to train classifiers. The classifiers can be trained for the process of detection of face This is known as HAAR training. Here, a cascade function is trained from a number of images, both positive and negative as referred in [6]. Each feature is a single value obtained by subtracting sum of pixels under various regions of the images. The pixels used for extraction is different for each feature. All the extracted features will not be useful for the required process. Haar Cascading method is used to identify whether a face is in the images or not. These are XML files that may be found in the opencv/data/haarcascades/folder directory. We utilised the classifier's "detectMultiScale" module for "Face detection." The function returns a rectangle around the identified face with coordinates (x, y, w, h). "scaleFactor" and "minNeighbors" are two key parameters that must be set according to the data. For the face detection in an image, we first use the CascadeClassifier method of the cv2 module to import the Haar cascade file of the frontal face by using the command [7]:

face_cascade = cv2.CascadeClassifier('haar_cascade_files/Haarcascade_frontalface_default.xml').

DLIB'S SHAPE PREDICTOR

To identify facial landmarks, we used the Dlib's shape predictor and analyzed the input images with a pre-trained file called 'shape predictor 68 face landmarks.dat'. They employed manually labeled facial landmarks to train the detector, defining particular (x, y) - coordinates of regions surrounding each facial component. A combination of regression trees is utilized to train the file for pixel intensity-based estimation of face landmark positions. The facial landmark detector is used to estimate the location of 68 (x, y) - coordinates on the face that correspond to facial structure. The annotations in Fig. 2 are based on 68 point iBUG 300-W dataset which the Dlib facial landmark predictor was trained on [8].

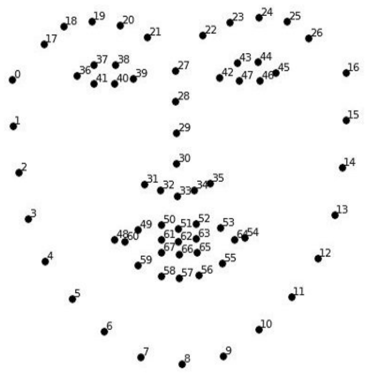

Fig. 2: Annotation based on 68 point iBUG 300-W dataset.

Facial Landmarks	Points
Jaw	1-17
Right Eyebrow	18-22
Left Eyebrow	23-27
Nose	28-36
Right Eye	37-42
Left Eye	43-48
Mouth	49-68

Table 1. Annotation of different landmarks points.

Table 1 represents the mapping of facial landmarks with points.

The detected facial landmarks are represented in Figure 3.

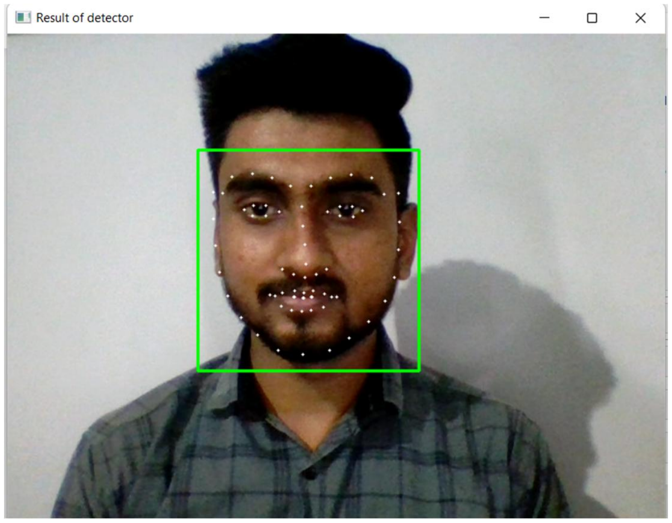

Figure 3: 68 Face Landmarks points detected.

III. Implementation

The implementation of the proposed system can be better understood with the help of the flowchart shown in Figure 4. The proposed system calculates the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) from the points detected with the help of Dlib's 68_landmark_detector file.

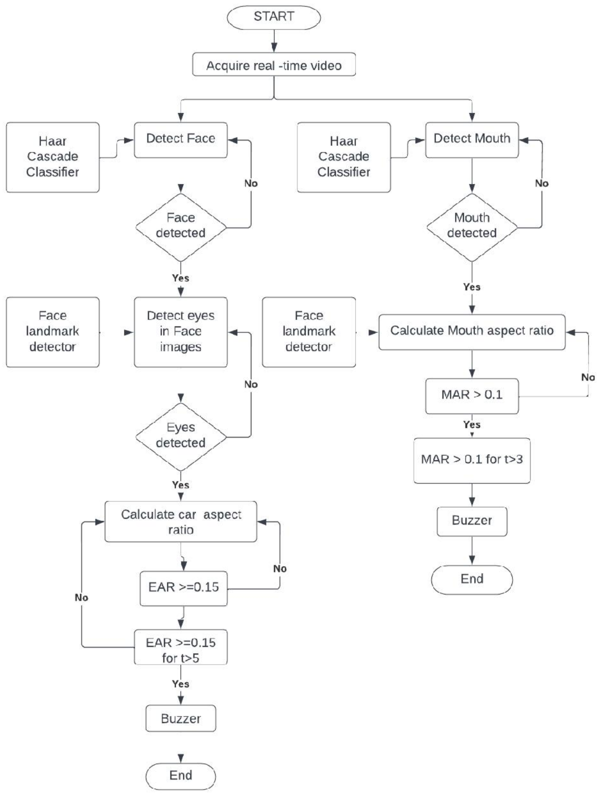

Fig. 4: Flowchart of implementation of proposed system.

CALCULATION OF EAR:

The Eye Aspect Ratio (EAR) estimates the state of the eye opening. Using the above facial points, we can deduce that the eye coordinates are represented by points 37 to 48. It represents the coordinates of the right eye from 37–42, and the coordinates of the left eye from 43–48.

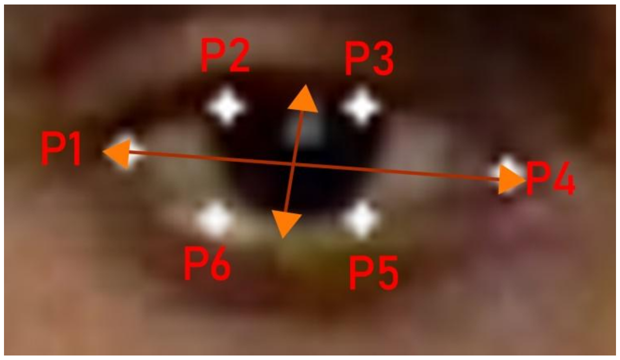

Fig. 5: Representation of right eye points used for calculations.

As shown in Fig. 5, closing one's eyes causes the value of EAR to decrease. EAR is defined as the ratio of height and width of the eye and was computed using equation (1).

$$EAR = \frac{|P2-P6|+|P3-P5|}{2|P1-P4|} \qquad \text{Equation (1)}$$

When the eyes are closed, the value of |p2 – p6| + |p3 – p5| decreases, resulting in a decrease in the EAR value. To begin, we will create a variable called EAR THRESHOLD and assign it a value [9]. After capturing the facial landmark points, EAR THRESHOLD value computed and compared with the threshold value 0.15 [10]. If the value is less than the threshold then the counter value is incremented, else the counter value is set back to zero. If the counter value reaches to three, an alarm is triggered. We will calculate the eye aspect ratio for each captured frame and increment the counter if it is greater than the EAR THRESHOLD. If the counter reaches 40, it means that the value of EAR was greater than EAR THRESHOLD for 40 frames, and we can now say that the person is drowsy and the alert would be given to user with a buzzer. The average eye aspect ratio is 0.339 and 0.141 when the eyes are open and closed, respectively. However, using the EAR calculation, it can be established that if the EAR value abruptly decreases, the driver is most likely closing his eyes. As a result, the user eyes blink or close, and if the average EAR value falls from 0.339 to 0.141, alerting them till they open their eyes [2].

CALCULATION OF MAR:

The Mouth Aspect Ratio (MAR) is used to determine whether or not a person is yawning. We can deduce from the facial points represented in Fig. 3, that points 49 to 68 represent the mouth coordinates. However, we will only use eight points, which are 61–68. The mouth aspect ratio (MAR) calculation formula is shown in equation (2) and representation of the mouth points in Fig. 6.

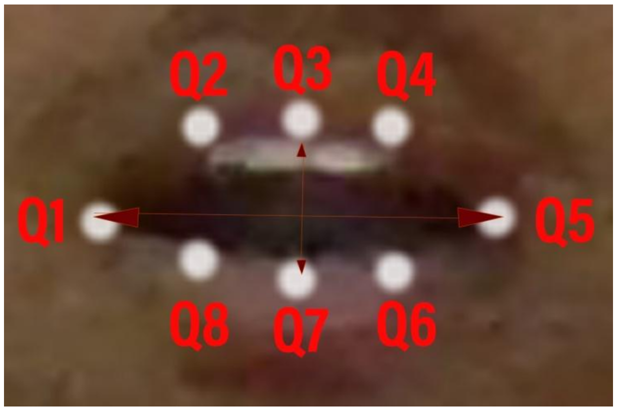

Fig. 6: Representation of mouth points used for calculations

$$MAR = \frac{|Q2-Q8|+|Q4-Q6|}{2|Q1-Q5|} \qquad \text{Equation (2)}$$

We yawn automatically when we are sleepy. We can tell if a person is drowsy or not by observing whether or not person is yawning [9]. When the driver opens his mouth while speaking, we do not consider it yawning. It is also comparable to EAR. If the calculated MAR value is greater than the threshold value of 0.1 [10], the counter variable is increased. If the value of the counter variable also reaches a certain level, the person is said to be drowsy.

IV. Result & Conclusion

The proposed system is capable of determining the drowsiness of the driver with implementation of above discussed methodologies. The results of the proposed system could be seen in the following figures, which represent the identification of the state of the driver as active, drowsy or sleeping. Based on the state of driver the proposed system would take some action and give an alert based on state. If driver is drowsy or sleeping it would give an alert based on the blinking and closing of an eye and yawning. The below Fig. 7, shows the active state of the driver.

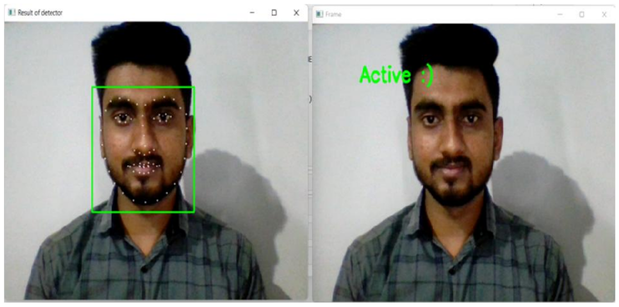

Fig. 7: Active state of driver

The below Fig. 8, represents the drowsy state of driver.

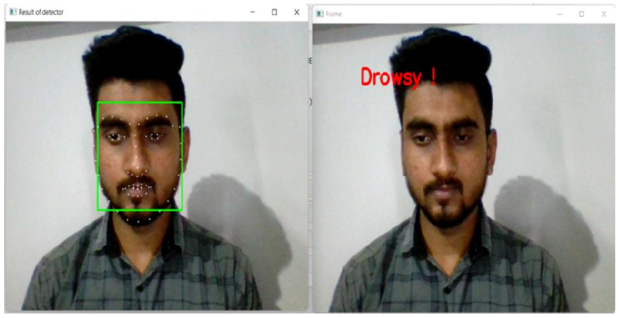

Fig. 8: Drowsy state of driver

The below Fig. 9, represents the sleeping state of driver.

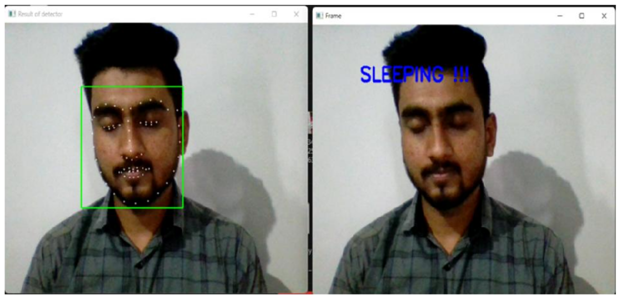

Fig. 9: Sleeping state of driver

Thus, in real time application of drowsiness detection of the proposed system detects the drowsiness state of the driver and as a future scope could be implemented with the help of hardware.

REFERENCES

[1]. Rohan Gupta, “Breaking Down Facial Recognition: The Viola-Jones Algorithm” [Online]. Available: https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999

[2]. Nora Kamarudin, Nur Anida Jumadi, Ng Li Mun, Ng Chun Keat, Audrey Huong Kah Ching, Wan Mahani Hafizah Wan Mahmud, Marlia Morsin, Farhanahani Mahmud, “Implementation of Haar Cascade Classifier and Eye Aspect Ratio for Driver Drowsiness Detection Using Raspberry Pi”, Universal Journal of Electrical and Electronic Engineering 6(5B): pp. 67-75, 2019.

[3]. Italo José, “Facial mapping (landmarks) with Dlib + python” [Online]. Available: https://twardsdatascience.com/facial-mapping-landmarks-with-dlib-python-160abcf7d672

[4]. Shruti Mohanty, Shruti V Hegde, Supriya Prasad, J. Manikandan, “Design of Real-time Drowsiness Detection System using Dlib”, 5th IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), 2019.

[5]. Great Learning Team, “Face Detection using Viola Jones Algorithm” [Online]. Available: https://www.mygreatlearning.com/blog/viola-jones-algorithm

[6]. V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N, Dr. Nagamani N P, “Driver Drowsiness Detection”, International Journal of Engineering Research & Technology (IJERT), Volume 8, Issue 15, 2020, pp. 33-35.

[7]. Sidra Mehtab, Jaydip Sen, “Face Detection Using OpenCV and Haar Cascades Classifiers”, Conference: MS (Data Science and Analytics) Minor Project Presentation At: NSHM Knowledge Campus, Kolkata, INDIA Affiliation: NSHM Knowledge Campus, March 2020.

[8]. Takrim Ul Islam Laskar , Parismita Sarma, “Facial Landmark Detection for Expression Analysis”, International Journal of Computer Sciences and Engineering, Vol.-7, Issue-5, May 2019, pp. 1617-1622.

[9]. T. V. N. S. R. Sri Mounika, P. H. Phanindra, N. V. V. N. Sai Charan, Y. Kranthi Kumar Reddy, S. Govindu, “Driver Drowsiness Detection Using Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), and Driver Distraction Using Head Pose Estimation”, ICT Systems and Sustainability Proceedings of ICT4SD 2021, Volume 1, pp. 619-627.

[10]. Vaibhav Garg, Parteek Goel, “Yawning Detection System”, International Research Journal of Engineering and Technology, Volume: 07 Issue: 06 | June 2020, pp. 538- 541.