Real Time Eye Gaze Detection Using Machine Learning Techniques

Achudan TS
Department of Electrical and Electronics Engineering
Mepco Schlenk Engineering College, Sivakasi, Virudhunagar- 626005, India.
Phone number: +91 9994301185
achuts4@gmail.com
Gobinath N
Department of Electrical and Electronics Engineering
Mepco Schlenk Engineering College, Sivakasi, Virudhunagar- 626005, India.
Phone Number: +916383227981
gobi1996007@gmail.com

DOI: 10.29322/IJSRP.10.01.2020.p9721
http://dx.doi.org/10.29322/IJSRP.10.01.2020.p9721

Abstract:
Drowsiness and exhaustion of vehicle drivers reduce the driver's abilities to control the vehicle, his natural reflex, and perception. Such diminished vigilance level of drivers is observed at night driving or prolonged driving, causing an accident and pose a severe threat to commuters. Therefore, it is an absolute requisite in the automobile industry to assimilate the driver assistance system that can detect drowsiness and fatigue of the drivers. This paper proposes a prototype for monitoring the driver's vigilance using a computer vision system in real-time. Eye-tracking is one of the critical parts in future driver assistance systems since human eyes contain much-needed information about the driver's condition, such as attention level, gaze, and fatigue level. One common problem faced by many eye-tracking methods proposed so far is their sensitivity to change in lighting conditions. This tends to limit their scope for automotive applications significantly. This paper illustrates an eye-gaze detection and tracking method that works under realistic and variable lighting conditions in real-time. Keywords: OpenCV, eye gaze, image processing, Machine Learning, drowsiness detection.

Keywords: OpenCV, eye gaze, image processing, Machine Learning, drowsiness detection

INTRODUCTION

The increasing number of traffic accidents due to a driver's diminished vigilance level is a severe problem for society. Driver's abilities to control the vehicle, his natural reflex, and perception decline drastically due to fatigue and drowsiness, reducing the driver's vigilance level. These ultimately results in a threat, not only to other lives but also to their own. According to the U.S. National Highway Traffic Safety Administration (NHTSA), in 2017, drowsiness and falling asleep while driving is responsible for at least 91,000 automobile crashes annually. The annual average of nonfatal injuries is 40,000, and 1,306 fatalities are from these crashes. More than a hundred billion dollars in damage occurs annually due to drowsiness related crashes. These figures only present the casualties happening from midnight to early morning, and underestimates the involvement of sleepiness because the above data does not include crashes during daytime hours. Vehicles having systems that are intelligent can detect drowsiness and alert the driver.

I. THE BASE IDEA

The central concept of Driver Drowsiness Detection is to capture a driver's face from a camera and accurately be able to calculate their drowsiness level, processing it in real-time. To achieve the requirements as mentioned earlier, Open CV library can be used, for its convenience and compatibility. Various tools used for our prototype are discussed below.

A. OPENCV

Open-source computer vision is a popular computer vision library that has the class for face recognition using the latest computer vision algorithms. Face recognition build on the geometric features of a face is probably the most intuitive approach to face recognition.

B. SCIPY

SCIPY is a library that uses NumPy for various mathematical functions. SciPy uses NumPy arrays as the underlying data structure and comes with modules for various commonly used tasks in scientific programming, including linear algebra, ordinary differential equation solving, integration calculus, and signal processing. SciPy is used in this scenario to calculate the Euclidian distance between the vector points plotted on the eyes.

C. DLib

The face detection or any object detection can be done with any of the OpenCV algorithms. But detecting facial landmarks is a complicated process. The shape predictor's primary job is to localize the critical points of interest along with the shape. In the context of facial landmarks, our goal is to use shape prediction methods to detect facial structures on the face that are crucial.

Facial landmarks are detected in a two-step process:
Step 1: Localizing the face in an image.
Step 2: Detecting the key facial structures of the face.

To localize the face in the image, OpenCV algorithms are used. The crucial task of the process lies in its second step, to detect the key facial structures in the face region. There are a variety of facial landmark detectors, but all methods mostly try to localize and label the following facial regions such as Right and left eyebrow, Right eye, Left eye, Mouth, Nose, and Jaw. This method of Dlib starts by using a training set of labeled facial landmarks on an image. These images are labeled manually, specifying (x, y)-coordinates of regions surrounding each facial structure specifically. Distance between pairs of input pixel's probability is also calculated. Given this training data, using the pixel intensities, an ensemble of regression trees are trained to estimate the facial landmark positions directly. The indexes of the 68 coordinates can be visualized on the image below: Where p1, p2, p3, p4, p5, and p6 are 2D facial landmark locations. The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, appropriately weighting the denominator since there is one more set of vertical points than horizontal points.

The indexes of the 68 coordinates can be visualized on the image below:

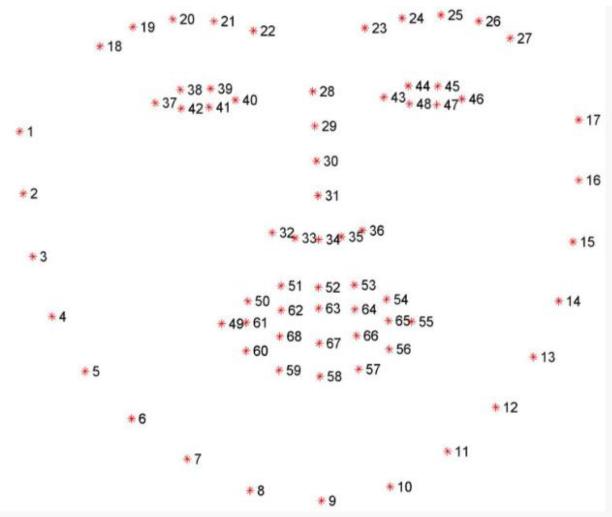

Fig., 1.0 68 point coordinates of human face.

Where p1, p2, p3, p4, p5 and p6 are 2D facial landmark locations.

II. THE DROWSINESS DETECTOR ALGORITHM

The eye aspect ratio is the crucial feature of the drowsiness detection algorithm. The eye aspect ratio is the ratio of the mean of Euclidean distance between eyelids to the Euclidean distance between the left and right edges of the eye. Since the distance between eyelids is directly proportional to the EAR, the decrease in distance between eyelids decreases the EAR value. This can be used to find Eye openness and closeness. The points are displayed in the image below.

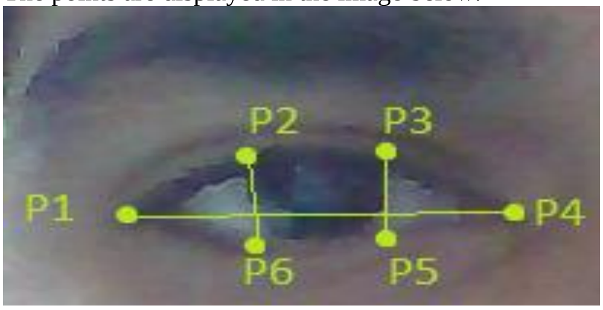

Fig. 2.0 Plots marked against a human eye

The formula to calculate Eye Aspect Ratio value is given below.

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

The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, weighting the denominator appropriately since there is only one set of horizontal points but two sets of vertical points.
The EAR of the above figure 2.0 is about 0.4 which means that the eyes are open.

A. EYES OPENESS DETECTION

The eye openness can be predicted using EAR values. When the eyes open, the Euclidean distance between the upper eyelids and lower eyelids is maximum. Since the distance is maximum, the eye aspect ratio is also more. From this value, we can find that the eyes are open. These annotations form a part of the 68-point Dataset, which the Dlib facial landmark predictor was trained. From these 68 points, 37 to 48 are representing the left and right eyes of humans. These points are used for further analysis of drowsiness detection.

ISSN 2250-3153

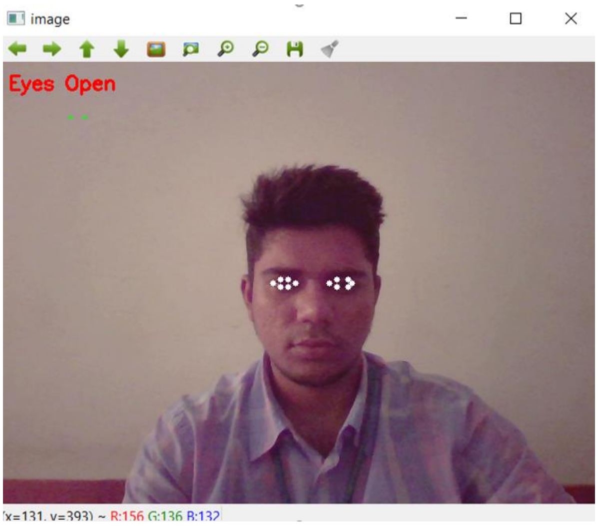

Fig., 2.1 The detection of eye in open state

B. EYES CLOSENESS DETECTION

The actual distance between eyelids must be zero to prove that eyes are closed. But drowsiness explains the state of sleep where eyes go from an open state to a closed state. It means that we have to find in between where the EAR value is about less than 0.3. The below image shown is with eyes closed.

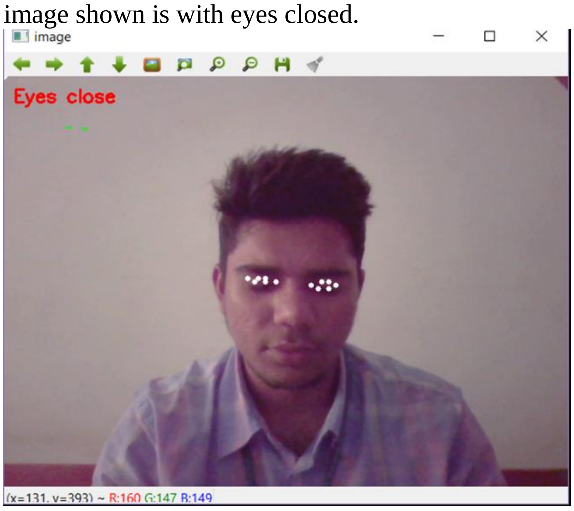

Fig., 2.2 The detection of eye in closed state

III. THE DROWSINESS CHECKING

The python program is written in such a way that the algorithm keeps on tracking for human eyes. When the eyes are detected, the eye aspect ratio value is calculated run time. The standard blink rate of a human is 300 milliseconds. So the tracking system should not consider normal blinking as drowsiness as the eye aspect ratio during blink goes below 0.2. So, whenever the eye aspect ratio value decreases below 0.2, the timer starts. If the eye aspect ratio value is maintained below 0.2 for more than 1.5 seconds, the alert system is triggered. This awakens the vehicle driver from drowsiness.

IV. HARDWARE

The Raspberry Pi 3 microcontroller is used to implement this whole idea and use it for practical application. Due to its high-performance CPU and higher frame rate, Raspberry Pi 3 and Raspberry Pi Camera is used. Raspberry Pi 3 Model B supports Python and OpenCV library. Also, the paper is done by using the Anaconda Python IDE. We apply OpenCV Version 3.4.0 for various features of computer vision. The Haar Cascade Classifier, warpAffine, and template matching are supported in the OpenCV library. Raspberry Pi is connected with a buzzer so that whenever the drowsiness is detected, the buzzer is triggered to alert the car driver.

V. Optimization for Precise Response

The existing detection of drowsiness is based on public datasets available, containing faces of various ethnicity. While scrutinizing, it is understood that a particular car is driven by almost the same set of people every day. In such cases, using public datasets is not only less precise but also time-consuming. Instead of using a public dataset, a local dataset can be prepared in local or cloud storage using an optimized algorithm to maintain the quality of the data in the local dataset.

A. Preparation of dataset

A separate dataset is prepared for every person driving the vehicle. The photo of distinct drivers having the high resolution is stored as a reference for the data which is going to be collected in runtime. The collected data is our dataset having a normal distribution from the higher spectrum of quality.

B. Image comparison technique

The technique used in finding the similarity between images is the Structural Similarity Measure. The results of this algorithm are better compared to the Mean square error technique. The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission. It is often regarded as a full reference metric that needs two images from the same image capture— a reference image and a processed image. The processed image is typically in a compressed state. It may, for example, be obtained by saving a reference image as a JPEG (at any quality level) then reading it back in. SSIM is best known in the video industry as well as in still photography.

C. Creation of Hash map

The Hash map is created with a key ranging from 0.61 to 1.0. The key represents the structural similarity image index (ssim), which is approximated to two decimal places. This results in a dataset having 40 images with varying quality. Here the already exiting image is not replaced by the new image if the key of both the images are similar. The dataset holds the already existing image.

D. Collection of data

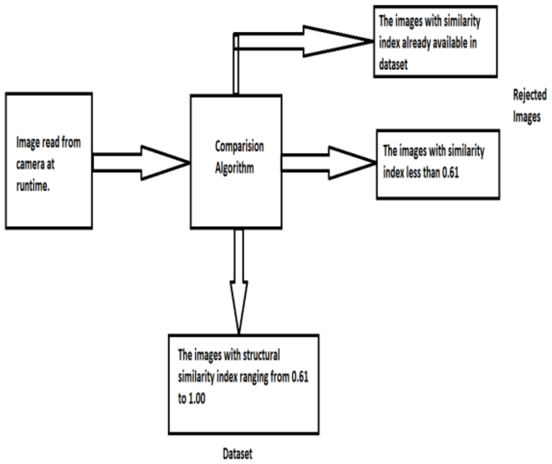

Fig., 3.0 Flow chart of collection of data

The figure 3.0 shows the flow of how the image is being collected. Images of the driver are captured from the camera, and the captured images is compared with the reference image that is already in the dataset. If the quality of the image meets our expectations; the compared image is stored in our dataset, which is usually kept in the range of .61 to .99. The value of the Hash map saves the image with respect to their keys. The data is ignored if the quality of the image is below .61.

VI. FUTURE SCOPE

The detection of drowsiness discussed so far helps in alerting the driver. But numerous unfortunate events have to be taken into account. Such as a sudden jerk by the driver once the alert is given, which may lead to crashes. The autonomous driving modes in self-driving cars are still under research and will be successful in the near future. In the case of self-driving cars coming into practice, the usage of drowsiness detection would be more efficient and useful. Further enhancement can be introduced by activating the self-driving mode first and then alerting the driver. This alert system is not only applicable for vehicles but also in workplaces where drowsiness leads to severe accidents, which may even be fatal.

CONCLUSION:

In the Driver Assistance System, the scope of using OpenCV for image processing are immense. It shows reliable performance in real-time conditions and also highly useful in processing images. The problems associated with human drowsiness and other human errors can probably overcome with OpenCV. Driving can be done better and smarter with the help of the driver assistance system, which uses OpenCV. This system is suitable for future improvements since it is compatible with many operating systems.

The result of the paper "REAL TIME EYE GAZE DETECTION USING MACHINE LEARNING TECHNIQUES" are in line with the expected output. From this implementation of paper, Drivers can quickly be alerted if there is a chance for an accident due to their drowsiness. This paper is also further adaptive for an upgrade.

References:

i. D'Orazio T, Leo M, Guaragnella C, Distante A (2007), "A visual approach for driver inattention detection",Pattern Recogn 40(8):2341–2355.

ii. Qiang J, Xiaojie Y (2002), "Real-Time Eye, Gaze, and face pose tracking for monitoring driver vigilance". Real-Time Imag 8:357–377

iii. Bakic, V., and Stockman, G., "Menu Selection by Facial Aspect, Proceedings Vision Interface", Quebec, Canada, May, 1999.

iv. Kim, K., and Ramakrishna, R.S., " Vision- Based Eye-Gaze Tracking for Human Computer Interface, Systems, Man and Cybernetics", IEEE SMC'99 Conference Proc., No.2, pp.324-329, 1999.

v. Mariani, R. "Face Learning Using a Sequence of Images, International Journal of Pattern Recognition and Artificial Intelligence", Vol.14, No.5, pp.631-648, 2000.

vi. Morimoto, C., Koons, D., Amir, A., and Flickner, M., "Pupil Detection and Tracking Using Multiple Light Sources, Image and Vision Computing, Special issue on Advances in Facial Image Analysis and Recognition Technology", Vol.18, No.4, pp.331-335, 2000.

vii. Pappu, R., and Beardsley, P.A., "A Qualitative Approach to Classifying Gaze Direction", Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, April, 1998.

viii. Singh,S., and Papanikolopoulos, P., " Advances in Vision-Based Detection of Driver Fatigue", Proceedings of the ITS America Ninth Annual Meeting, 1999.

ix. Smith, P., Shah, M., and Lobo, M.V., "Monitoring Head/Eye Motion for Driver Alertness with One Camera", Proceedings of the International Conference on Pattern Recognition, 2000.

x. Talmi, K., and Liu, J., "Eye and Gaze Tracking for Visually Controlled Interactive Stereoscopic Displays, Signal Processing: Image Communication", No.14, pp.799-810, 1999.

Authors' Profile:

Achudan TS

Mr. Achudan TS pursued his BE., from Mepco Schlenk engineering college, Virudhunagar in 2018. He is currently working as a software developer engineer in Test for Infosys Limited. The author has been worked in a key role in manufacturing of an Electric vehicle for Electric Solar Vehicle Championship(ESVC). He had intern at Nokia networks and solutions. His core domain is not only electrical and electronics, but also has made a wide range of applications in computer domain. He has two years of experience in software industry. He is an expert in process automation and implements devops with CICD and AWS cloud.

Gobinath N

Mr. Gobinath N pursued his BE., from Mepco Schlenk engineering college, Virudhunagar in 2018. He is currently working as a software developer engineer in SAP Hana technology for Infosys Limited. The author has worked in numerous automation and image processing projects. He had intern at Nokia networks and solutions. His core domain is not only electrical and electronics, but also has made a wide range of applications in computer domain. He has two years of experience in software industry. He is an expert in SAP ABAP development and core data services in SAP Hana platform.