Drowsiness detection using OpenCV

ABSTRACT

The new way of security system which can be discussed during this project is predicated on machine learning and AI. Passenger security is that the main concern of the vehicles designers where most of the accidents are caused thanks to drowsiness and fatigue driving so as to supply better security for saving lives of passengers bag are designed but this method is beneficial after accident is accord. But main problem remains we see many accidents happening and lots of them are losing their lives. during this project we are using openCV library for image processing and giving input as user live video and training data to detect if person in video is closing eyes or showing any symptoms of drowsiness and fatigue then application will verify with trained data and detect drowsiness and lift alarm which can alert driver.

Keywords

EAR, Drowsiness, OPENCV, Dlib, Python, Anaconda.

1. INTRODUCTION

Driver fatigue could even be an enormous believe an outsized number of auto accidents. the event of technologies for detecting or preventing drowsiness at the wheel could even be a significant challenge within the world of accident avoidance systems. methods got to be developed for counteracting its affects. the most target are becoming to be placed on designing a system which may accurately monitor the open or closed state of the driver's eyes in real-time. By monitoring the eyes, it's believed that the symptoms of driver fatigue are often detected early enough to avoid a car accident. Detection of fatigue involves the observation of eye movements and blink patterns during a sequence of images of a face. Initially, we decided to travel about detecting blink patterns using python the algorithm used was as follows. First, we input the facial image employing a webcam. the absolute best and sides of the face were detected to narrow down the earth where the eyes exist. Using the edges of the face, the center of the face was found which can be used as a reference when computing the left and right eyes. Moving down from the absolute best of the face, horizontal averages of the face area were calculated. Large changes within the averages were wont to define the attention area. There was little change within the horizontal average when the eyes were closed which was wont to detect a blink. However, python had some serious limitations. The processing capacities required by python were very high. python was capable of processing only 4-5 frames per second. On a system with a coffee RAM this was even lower. As we all know an eye fixed blink could even be a matter of milliseconds. Also, a drivers head movements are often pretty fast. Though the MATLAB program designed by us detected an eye fixed blink, the performance was found severely wanting. this is often where OpenCV came in. it's designed for computational efficiency and with a robust concentrate on real time applications. It helps to form sophisticated vision applications quickly and simply. we've used the Haartraining applications in OpenCV to detect the face and eyes. This creates a classifier given a gaggle of positive and negative samples. The steps were as follows: - Gather a knowledge set of face and eye. These should be stored in one or more directories indexed by a document. many high-quality data is required for the classifier to figure well. The utility application createsamples() is employed to form a vector file. Using this file, we'll repeat the training procedure. The Viola Jones cascade decides whether or not the thing during an image is analogous to the training set. Any image that doesn't contain the thing of interest are often became negative sample. So on determine 3 any object it's required to wish a sample of negative background image. of these negative images are put in one file then it's indexed. Training of the image is completed using boosting Each classifier within the group could even be a weak classifier. These weak classifiers are typically composed of 1 variable decision tree called stumps. Between training each classifier one by one, the data points are reweighted so as that more attention is paid to the info points where errors were made. This process continues until the entire error over the dataset arising from the combined weighted vote of the choice trees falls below a selected threshold. This algorithm is effective when an outsized number of coaching data are available. So, we used the training objects method to make our own haarclassifier .xml files. Training them could even be a time intensive process. Finally, face.xml and haarcascade-eye.xml files are created. These xml files are directly used for object detection Haarcascade-eye.xml is meant just for open eyes. When a blink lasts for quite 5 frames, the drive is judged to be drowsy and an alarm is sounded.

2. LITERATURE REVIEW

2.1 EXISTING SYSTEM:

There is various methods like detecting objects which are almost vehicle and front and rear cameras for detecting vehicles approaching almost vehicle and bag system which may save lives after accident is accorded.

2.2 DISADVANTAGES OF EXISTING SYSTEM:

Most of the prevailing systems use external factors and inform user about problem and save user after accident is accord but from research most of the accidents are thanks to faults in user like drowsiness, sleeping while driving.

2.3 PROPOSED SYSTEM:

To affect this problem and supply an efficient system a drowsiness detection system are often developed which may be placed inside any vehicle which can take live video of river as input and compare with training data and if driver is showing any symptoms of drowsiness system will automatically detect and lift alarm which can alert driver and other passengers.

2.4 ADVANTAGES OF PROPOSED SYSTEM:

This method will detect problem before any problem accord and inform driver and other passengers by raising alarm in this OpenCV based machine learning techniques are used for automatic detection of drowsiness.

3. ALGORITHM

Blink detection can be estimated by measuring EAR (Eye aspect Ratio) using OPENCV functions and DLIB’s pre trained Neural network-based prediction and detector function. EAR can be measured from eye coordinates returned from OPENCV using EAR formula. Abrupt dip in EAR value against a set threshold can be used for blink detection and microsleep detection.

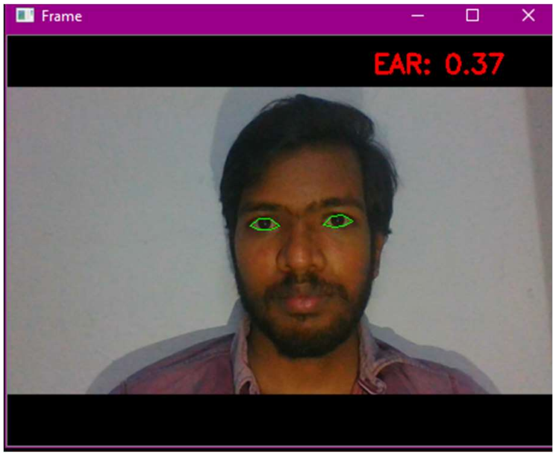

Figure-1: Results of facial Landmark detection and identification of eye coordinates.

3.1 MEASUREMENT OF EAR

Each eye is represented by 6 *(x, y)*-coordinates in landmarks retuned Dlib predictor function, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the remainder of the region. There is a relation between the width and the height of these coordinates. Author then derive an equation that reflects this relation called the eye aspect ratio (EAR):

$$\text{EAR (Eye aspect Ratio)} = \frac{|p2-p6|+|p3-p5|}{2|p1-p4|}$$

Where *p1, ..., 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, weighting the denominator appropriately since there is only *one* set of horizontal points but *two* sets of vertical points. the eye aspect ratio is approximately constant while the eye is open, but will rapidly fall to zero when a blink is taking place. When the person blinks the eye aspect ratio decreases dramatically, approaching zero. Eye aspect ratio is constant, then rapidly drops close to zero, then increases again, indicating a single blink has taken place. **Python Function for calculating EAR**

```
Def
eye_aspect_ratio(self,eye):
A = dist.euclidean(eye[1],
eye[5])
B = dist.euclidean(eye[2],
eye[4])
C = dist.euclidean(eye[0],
eye[3])

ear = (A + B) / (2.0 * C)
return ear
```

Algorithm for detection of Blinks and Microsleep

```
if ear < Threshold: # EAR
Threshold
        COUNTER += 1
if ear < Threshold:
DBAR+=10 if ear>
Threshold:      DBAR=0
if COUNTER > 2: # Blink
Detection
if ear > Threshold:
TOTAL +=1
COUNTER=0
if DBAR>TDBAR: # Microsleep Detection
        DEVENT+=1
```

If in any case EAR drops lower than set threshold and remains for at least 1 seconds it is detected as blink and COUNTER stores value of blink no. If further, EAR remains lower than threshold for more than 3 it is considered as microsleep and will be displayed on drowsiness scale and DEVENT variable stores no of drowsiness events. Proper logic to avoid false blink detection is implement by author.

3.2 EAR PLOT

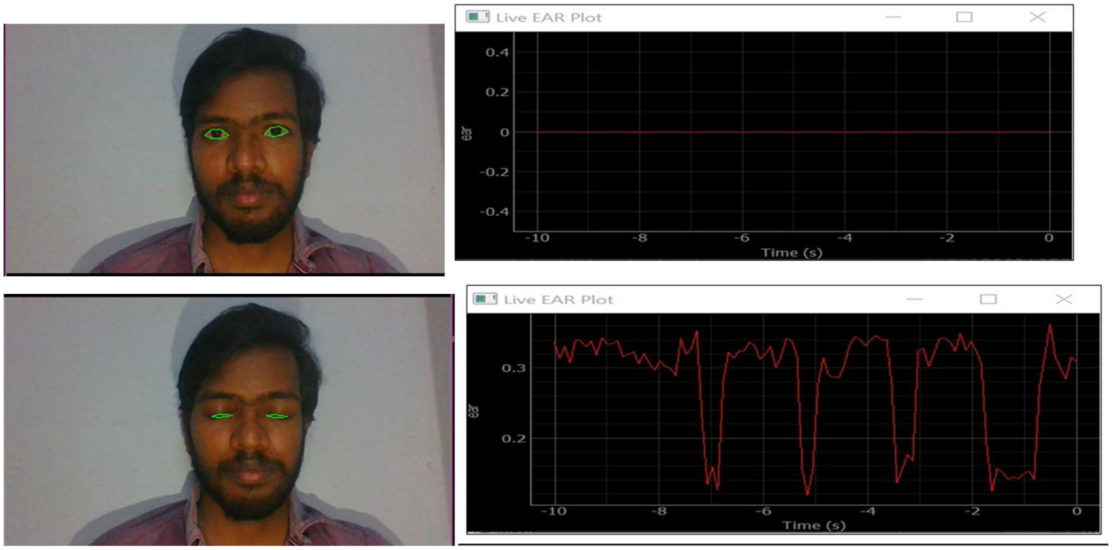

Figure 2: Simulation Results, EAR value for open Eye and close eye.

4. RESULTS

It applies facial landmark localization to extract eye regions from the face. Compute the EAR and sound an alarm if eyes have been closed for a sufficiently long enough time as shown in figure 3

Figure 3: Test results

5. CONCLUSION

Thus, we have successfully designed a prototype drowsiness detection system using OpenCV software and Haar Classifiers. The system so developed was successfully tested, its limitations identified and a future plan of action developed.

Future works: -

In the real time driver fatigue detection system, it is required to slow down a vehicle automatically when fatigue level crosses a certain limit. Instead of threshold drowsiness level it is suggested to design a continuous scale driver fatigue detection system. It monitors the level of drowsiness continuously and when this level exceeds a certain value a signal is generated which controls the hydraulic braking system of the vehicle.

6. REFERENCES

[1] Association for Safe International Road Travel (ASIRT), Road Crash Statistics.http://asirt.org/initiatives/informingroadusers/road-safety-facts/road-crash-statistics, 2016

[2] https://en.wikipedia.org/wiki/Microsleep

[3] Journal of VLSI Signal Processing 23, 497–511 (1999) c °1999 Kluwer

Academic Publishers. Manufactured in The Netherlands.

[4] https://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_de tection.html

[5] Eye Detection Using Morphological and Color Image Processing Tanmay Rajpathaka, Ratnesh Kumar and Eric Schwartzb

[6] A Robust Algorithm for Eye Detection on Grey Intensity Face without Spectacles- JCS&T Vol. 5 No. 3

[7] Froba Kebbuck: Audio- and Video-Based Biometric Person Authentication, 3rd International Conference, AVBPA 2001, Halmstad, Sweden, June 2001.
Proceedings, Springer. ISBN 3-540-42216-1.

[8] Driver Drowsiness Detection using Eye-Closeness Detection (2016 12th International Conference on Signal-Image Technology & Internet-Based Systems)