**Drowsiness detection using OpenCV**

**V.Vishesh kumar, K.Manudeep, B.Bhupal Raju, K.Dhana Sree Devi,**

**Dept of IT, Vardhaman college of engineering, Hyderabad.**

**\*Corresponding author Email: visheshkumar420@gmail.com.**

**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 bags 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.

**INTRODUCTION**

Driver lethargy is considered as the major cause of vehicle accidents.

Technological development for detecting and preventing driver’s drowsiness at the wheel is one of the major tasks for the developers. To avoid such disasters on roads, evolution of technology is required.

Development of prototype drowsiness detection system is the primary object of this project. This unique system is designed in such a way that it detects the open or closed state of driver’s eyes with accuracy at early sight and therefore avoids accidents. Detection of drowsiness is focused on blinking patterns and iris movements by sequential image capturing.

Firstly, we chose to detect eye blink patterns using python through procedure called geometric manipulation of intensity levels. The following algorithms are used.

Firstly, we capture a facial image using a webcam. Later, it is pre-processed by binarizing the image. It detects the top and sides of the face to narrow down the area around the eyes. Using the sides of the face, the center of the face is detected and is used as reference while computing the left and right eye. Moving from the top area, horizontal facial averages are calculated. Major changes in these averages are used to spot the eye area. This showed a minor change in horizontal average when eyes were closed, resulting the detection of eye blink. However, python arrayed major limitations.

Python requires extreme processing capacities. It also had certain problems with real time processing speed. It is possible to process only 4-5 frames per second. It is even slow on low RAM systems. The eye blink and the head movement takes hardly seconds and though, the MATLAB program is designed to detect an eye blink, the performance can be future improved. Hence, OpenCV has been preferred.

OpenCV is an open source optimized library focuses on real time applications. It is designed for computational efficiency that builds up organized vision applications effortlessly. In this , we use Haartraining applications for eyes and face detection, thus creating a classifier providing a set of positive and negative samples.

Following are the steps to be followed: collect a data set of eyes and face which has to be stored in more than one directory recorded by a text file. High quality data is required for a classifier to perform better.

To build a vector output file, Utility application createsample() can be used and training procedure can be repeated. Before normalizing and resizing the image to a certain width and height, it extracts the positive samples from it. Any image that does not contain any object of interest is converted into negative sample. Thus, in order to learn an object, 3 negative background images are required.

These negative images are gathered in a single file and indexed. Training of an image is done through boosting. In training, we learn about various classifiers one at a time. Each classifier in this set contains a weak classifier typically composed of a single variable decision tree known as Stumps. During the training of each classifier one by one, data points are re-weighted such that the focus is mainly data points were errors are made. This is a continuous process until the total errors of the dataset emerges from the combined weighted votes of the decision trees, drops at a certain point.

Such algorithm is effective when a bulk of training data is available. We need a face and eye classifiers for which we utilize learning objects method to create our own haarclassifier.xml files. We take around 2000 to 3000 positive and negative samples. Training them is a time consuming process. Finally, face.xml and haarcascade-eye.xml files are created. These xml files are used directly for detecting a object. It detects sequential objects (like face and eyes). haaracascadeeye.xml is designed only for open eyes and closed eyes cannot be detected.

Thus, when a blink is captured in more than 5 frames, then the driver is considered to be drowsy and is immediately alarmed.

.

**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. Unexpected plunge in EAR esteem against a set edge can be utilized for flicker location and microsleep recognition.



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

**3.1Measurement of EAR**

Each eye has been mentioned as 6 (x, y)-coordinates in landmarks retuned Dlib predictor function, beginning it at the left-corner of the eye (just like you were glimsing at the person) Later working sequentially around the rest of the space. However ,the width and the stature of these directions are interlinked. Designer would include a situation which reflect the link known as the eye angle proportion (EAR).

EAR (Eye aspect Ratio) = 

Where p1, p2, ..., p6 are 2D facial indicator spots. The numerator of the following situation records distinction amongst the vertical eye milestones whereas the denominator records the separation amongst even eye tourist spots, weighting the denominator suitably as we find only one lot of flat focuses and two ranging of vertical focuses. The angle eye proportion is unevenly compatible when the eye is open but it rapidly drops to zero when a blink takes place continuously . At times, when a person blinks the eye, perspective proportion disappears quickly taking it to zero. when eye's perspective proportion is constant, it rapidly falls at zero and there it increases another time, displaying the occurance of solitary squint.

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

For instance, EAR falls low than the required rate and holds at following rate and 1 second in it is described as flicker and COUNTER records the total count of blinks. Suppose EAR falls low than required for in more of 3 it is tagged as microsleep which is displayed on languor scale and DEVENT variable records count of lethargy instances. Authorised logic to step back from false flicker recognition is included by the designer.

**3.2 EAR PLOT**





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

**4. RESULTS**

It applies facial(face) landmark locations to take out the eye area from the face. Compute the EAR and give an alert by message and sound, when eyes has been closed for a certain no. of frames as shown in figure 3.



**Figure 3:** Test results

**5.CONCLUSION**

In this manner, we've effectively planned a model tiredness location framework with OpenCV programming.

**Future scope:**

The framework created was effectively tried, its confinements distinguished and a likely arrangement of activity created.

In this detection system, it needs to be help a vehicle automatically when the drowsiness level crosses a particular limit. Rather than threshold fatigue level it is proposed to style endless scale detection system. It observes the level constantly, when the level surpass the limit then a sign is created which leads to handling of vehicle’s hydraulic braking system.

**6. REFERENCES**

* Association for Safe International Road Travel (ASIRT), Road Crash Statistics.http://asirt.org/initiatives/informingroadusers/road-safety-facts/road-crash-statistics, 2016
* <https://en.wikipedia.org/wiki/Microsleep>
* Journal of VLSI Signal Processing 23, 497–511 (1999) c °1999 Kluwer Academic Publishers. Manufactured in The Netherlands.
* [https:// HYPERLINK "https://docs.opencv.org/trunk/d7/d8b/tutorial\_py\_face\_detection.html"docs.opencv.org/trunk/d7/d8b/tutorial\_py\_face\_de HYPERLINK "https://docs.opencv.org/trunk/d7/d8b/tutorial\_py\_face\_detection.html"tection.html](https://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html)
* Eye Detection Using Morphological and Color Image Processing Tanmay Rajpathaka, Ratnesh Kumar and Eric Schwartzb
* A Robust Algorithm for Eye Detection on Grey Intensity Face without Spectacles- JCS&T Vol. 5 No. 3
* 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.

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