# Real Time Drivers Drowsiness Detection and alert System by Measuring EAR

Rajneesh
M. Tech (ECE)
RIEIT near Ropar, Dist. S.B.S
Nagar, Punjab-144533, India

Anudeep Goraya
M. Tech (ECE)
RIEIT near Ropar, Dist. S.B.S
Nagar, Punjab-144533, India

Gurmeet Singh
B. Tech (ECE)
NIT Kurukshetra
Haryana-136118
India

## ABSTRACT

On road driver's fatigue and drowsiness is contributing more than 30%[1] of reported road accidents. Driver drowsiness can be estimated by monitoring biomedical signals ,visual assessment of driver's bio-behavior from face images , by monitoring drivers performance or by combines all above techniques. Proposed algorithm is based on live monitoring of EAR (Eye aspect Ratio) by application of Image processing. HD live video is decomposed in continues frames and facial landmarks has been detected using pre trained Neural Network based Dlib functions. Dlib functions are trained using HAAR Cascade algorithm. Intel's Open source Image processing libraries (OPEN CV) is used as primary Image processing tool. Python Language is used as main codding language. EAR is calculated by calculating Euclidean distance between measured eye coordinates . Blink and microsleep detection mechanism is implemented by monitoring EAR against a threshold value. Blinks and drowsiness level are displayed on monitor screen with microsleep detection audio warning .

## General Terms

Image Processing, Computer vision, deep learning, neural networks.

## Keywords

EAR, Microsleep, Drowsiness, OPENCV, Dlib, Python.

## 1. INTRODUCTION

Driver in-alertness is an important resulting from sleep deprivation or sleep disorders and is an important factor in the increasing number of the accidents on today's roads. Drowsy driver warning system can form the basis of the system to possibly reduce the accidents related to driver's drowsiness. In a year there were 824 fatalities (2.2% of all fatalities) recorded in NHTSA's FARS database that were drowsy-driving-related. According to the results of the study presented at the International Symposium on Sleep Disorders, fatigue of drivers is responsible for 30% of road accidents[1]
Different techniques are used in driver-fatigue monitoring systems. These techniques are divided into four categories. The first category includes intrusive techniques, which are mostly based on monitoring biomedical signals, and therefore require physical contact with the driver. The second category includes non-intrusive techniques based on visual assessment of driver's bio-behavior from face images. The third category includes methods based on driver's performance, which monitor vehicle behavior such as moving course, steering angle, speed, braking, etc. Finally, the fourth category combines techniques from the above mentioned three categories. The computer vision based techniques from the second category are particularly effective, because the drowsiness can be detected by observing the facial features and visual bio-behavior such as head position, gaze, eye openness, eyelid movements, and mouth openness. Proposed algorithm is based on computer vision method. The main focus is on the detection of blinks by estimating the EAR(Eye aspect Ratio). This is achieved by monitoring the eyes of the driver throughout the entire video sequence. An IR camera will be used for capturing live video of driver eyes in all light conditions and frames will extracted for image processing scheme of video capturing .Algorithm can be divided broadly in following sub modules.

- Frame Acquisition
- Facial landmark detection
- Eye Localization and tracking
- Extracting eye geometrical coordinates
- Measuring EAR
- Monitoring of EAR for blinks detection
- Estimation of micro sleep periods between blinking
- Audio Visual warning on micro sleep detection.

**EAR** is define as per below formula , where p1,p2….p6 is the eye coordinates as discussed in further text

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

**Microsleep:** A micro-sleep is a temporary episode of sleep or drowsiness which may last for a fraction of a second or up to 30 seconds where an individual fails to respond to some arbitrary sensory input and becomes unconscious[2]

## 2. LITERATURE REVIEW

### 2.1 Face and Eye Detection by CNN Algorithms[3]

In this paper a novel approach to critical parts of face detection problems is given, based on analogic cellular neural network (CNN) algorithms. The proposed CNN algorithms find and help to normalize human faces effectively while cause for most accident related to the vehicles crashes. Driver fatigue their time requirement is a fraction of the previously used methods. The algorithm starts with the detection of heads on colour pictures using deviations in colour and structure of the human face and that of the background. By normalizing the distance and position of the reference points, all faces should be transformed into the same size and position. For normalization, eyes serve as points of reference. Other CNN algorithm finds the eyes on any grayscale image by searching characteristic features of the eyes and eye sockets. Tests made on a standard database show that the algorithm works very fast and it is reliable[3].

## 2.2 Face Detection using Haar Cascades[4]

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle[4].

## 2.3 Eye Detection Using Morphological and Color Image Processing[5]

Eye detection is required in many applications like eye-gaze tracking, iris detection, video conferencing, auto-stereoscopic displays, face detection and face recognition. This paper proposes a novel technique for eye detection using color and morphological image processing. It is observed that eye regions in an image are characterized by low illumination, high density edges and high contrast as compared to other parts of the face. The method proposed is based on assumption that a frontal face image (full frontal) is available. Firstly, the skin region is detected using a color based training algorithm and six-sigma technique operated on RGB, HSV and NTSC scales. Further analysis involves morphological processing using boundary region detection and detection of light source reflection by an eye, commonly known as an eye dot. This gives a finite number of eye candidates from which noise is subsequently removed. This technique is found to be highly efficient and accurate for detecting eyes in frontal face images[5].

## 2.4 Algorithm for Eye Detection on Grey Intensity Face[6]

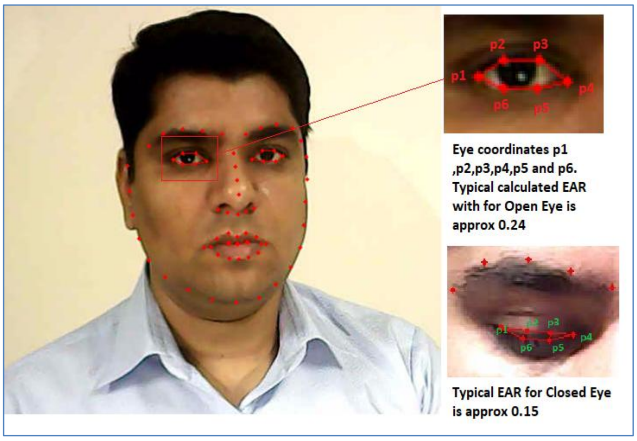
This paper presents a robust eye detection algorithm for grey intensity images. The idea of our method is to combine the respective advantages of two existing techniques, feature based method and template based method, and to overcome their shortcomings. Firstly, after the location of face region is detected, a feature based method will be used to detect two rough regions of both eyes on the face. Then an accurate detection of iris centres will be continued by applying a template based method in these two rough regions. Results of experiments to the faces without spectacles show that the proposed approach is not only robust but also quite efficient[6].

## 2.5 Real-Time Face Detection Using Edge-Orientation Matching[7]

In this paper we describe our ongoing work on real-time face detection in grey level images using edge orientation information. We will show that edge orientation is a powerful local image feature to model objects like faces for detection purposes. We will present a simple and efficient method for template matching and object modelling based solely on edge orientation information. We also show how to obtain an optimal face model in the edge orientation domain from a set of training images. Unlike many approaches that model the grey level appearance of the face our approach is computationally very fast. It takes less than 0.08 seconds on a Pentium II 500MHz for a 320x240 image to be processed using a multi-resolution search with six resolution levels. We demonstrate the capability of our detection method on an image database of 17000 images taken from more than 2900 different people. The variations in head size, lighting and background are considerable. The obtained detection rate is more than 93% on that database[7].

## 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 given in section-1. Abrupt dip in EAR value against a set threshold can be used for blink detection and microsleep detection.



Eye coordinates p1 ,p2,p3,p4,p5 and p6. Typical calculated EAR with for Open Eye is approx 0.24

Typical EAR for Closed Eye is approx 0.15

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

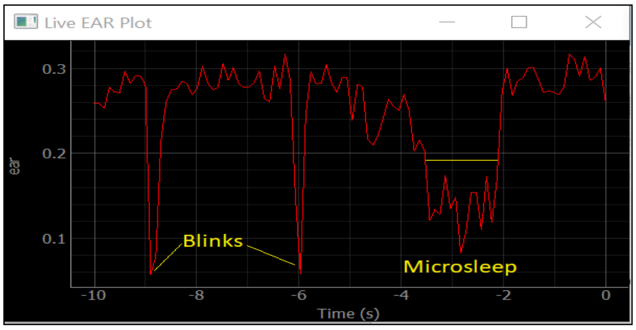


**Figure-2:** EAR Plot showing blink detection and micro sleep detection.

## 3.1 TOOLS AND IMAGE PROCESSING LIBRARIES

Following optimized tools and image processing libraries are used by author for implementation of presented algorithm.

**Open CV:** OpenCV (Open-source Computer Vision) is the Swiss Army knife of computer vision. It has a wide range of modules that can help us with a lot of computer vision problems. But perhaps the most useful part of OpenCV is its architecture and memory management. It provides you with a framework in which you can work with images and video in any way you want, using OpenCV's algorithms or your own, without worrying about allocating and reallocating memory for your images. Open CV libraries and functions are highly optimized and can be used for real time image and video processing. OPENCV's highly optimized image processing function are used by author for real time image processing of live video feed from camera.

**DLib:** Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge. Open Source Dib library is used by author for implementation of CNN(Neural Networks). Highly optimized Pre-learned facial shape predictor and detectors functions are used by author for detection of facial landmarks. Facial landmarks were further used for extracting eye coordinates.

**Python:** Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability. Python language is used by author due to his cross platform compatibility as main coding language for algorithm. Open CV and Dlib libraries are integrated in python interpreter for using readymade optimized functions.

**PyQT5:** PyQt5 is a Python binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in. PyQt is free software developed by the British firm Riverbank Computing. PyQt supports Microsoft Windows as well as various flavors of UNIX, including Linux and Mac OS. highly optimized open source libraries of Python is used by author for GUI design and graphical data plotting.PyQt5 PyQtgraph function is used by author for live graphical monitoring of EAR values.

**PyCharm:** Open source community version of PyCharm platform is used as main coding editor by author.

## 3.2 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):

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

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. As shown in Figure-2, 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 consider 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.

### 3.3 Algorithm flow diagram

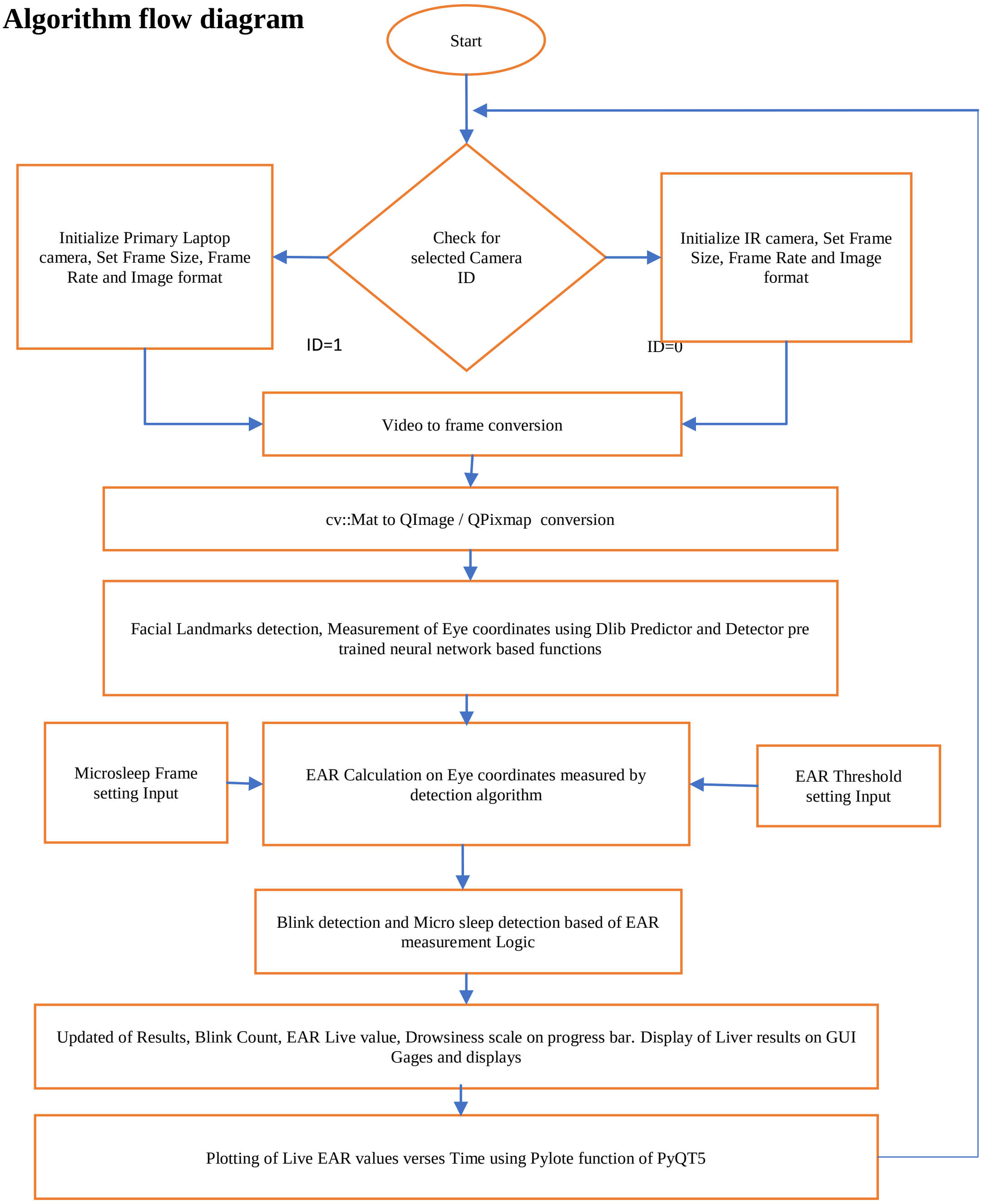

Start
Check for selected Camera ID
Initialize Primary Laptop camera, Set Frame Size, Frame Rate and Image format
Initialize IR camera, Set Frame Size, Frame Rate and Image format
ID=1
ID=0
Video to frame conversion
cv::Mat to QImage / QPixmap conversion
Facial Landmarks detection, Measurement of Eye coordinates using Dlib Predictor and Detector pre trained neural network based functions
Microsleep Frame setting Input
EAR Calculation on Eye coordinates measured by detection algorithm
EAR Threshold setting Input
Blink detection and Micro sleep detection based of EAR measurement Logic
Updated of Results, Blink Count, EAR Live value, Drowsiness scale on progress bar. Display of Liver results on GUI Gages and displays
Plotting of Live EAR values verses Time using Pylote function of PyQT5

**Figure-3 :** Flow diagram of Proposed Algorithm

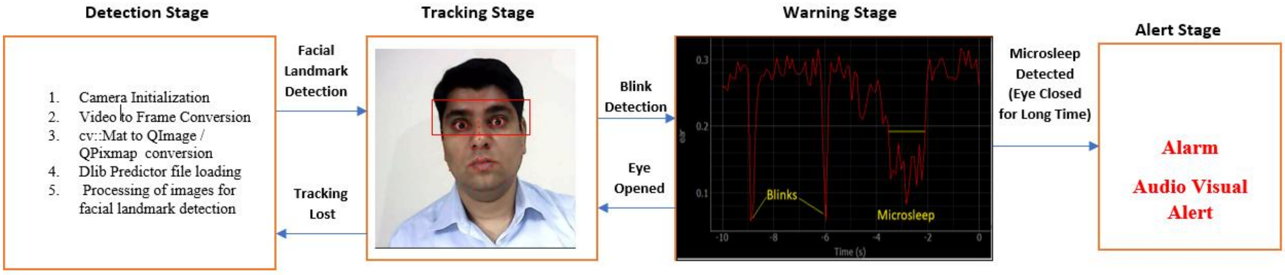

Detection Stage
1. Camera Initialization
2. Video to Frame Conversion
3. cv::Mat to QImage / QPixmap conversion
4. Dlib Predictor file loading
5. Processing of images for facial landmark detection
Facial Landmark Detection
Tracking Lost
Tracking Stage
Blink Detection
Eye Opened
Warning Stage
0.3
0.2
0.1
ear
Blinks
Microsleep
-10
-8
-6
-4
-2
0
Time (s)
Microsleep Detected (Eye Closed for Long Time)
Alert Stage
Alarm
Audio Visual Alert

**Figure-4 :** Algorithm Steps

## 4. RESULTS

The author plot the EAR measurements with respect to time . Each dip in EAR curve corresponds to eye blink and width of EAR dip represents duration of eye closed and formed basis of Microsleep detection. Algorithm performs well in repeated experiments on different type of faces. EAR threshold adjustment provided in GUI front end is useful for fine tuning of algorithm on different type of faces. No false detection and missed detection of blinks and microsleep was observed.

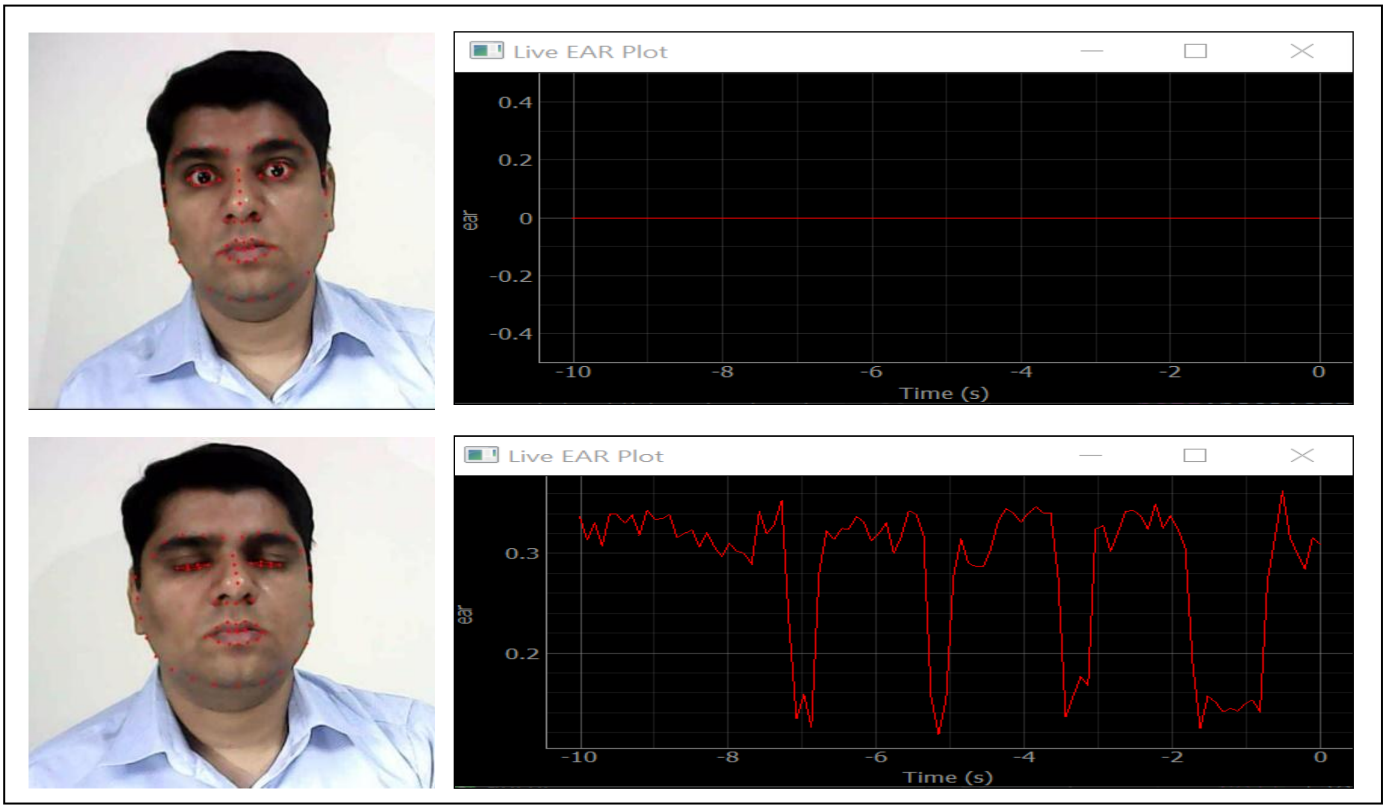


**Figure 5:** Simulation Results, EAR Plots for open Eye and blinking eye.

## 4.1 Detection of facial landmarks at different facial angles

Author evaluate the performance of algorithm on different facial angles . Performance of algorithm found excellent in facial landmark detection at all possible facial angles provided eyes are visible to camera.

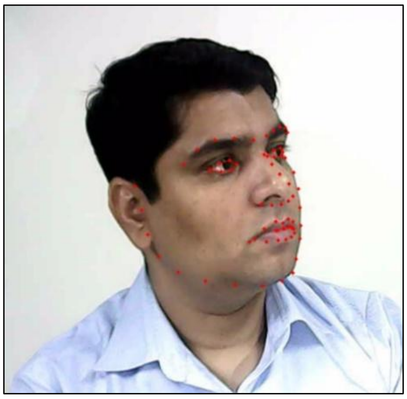


**Figure-6:** Left View

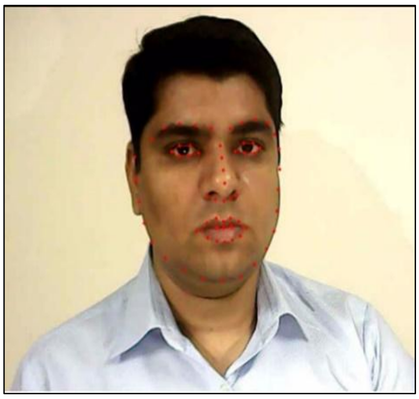


**Figure-7:** Straight View

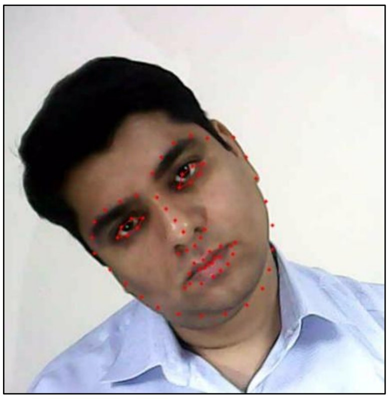


**Figure-8:** Right Tilted View

**Figure-9:** Upward view

**Figure-10:** Right View

**Figure-11 :** Downward view

## 4.2 Result of Eye Detection in dark condition using IR camera

Main application of propose algorithm is in detection of drivers drowsiness state during driving . Light condition in car will effect performance of algorithm. Illumination may not be proper during night hours and may effect performance of facial landmark detection. Author evaluated the performance of algorithm in dark condition using IR camera and found satisfactory.

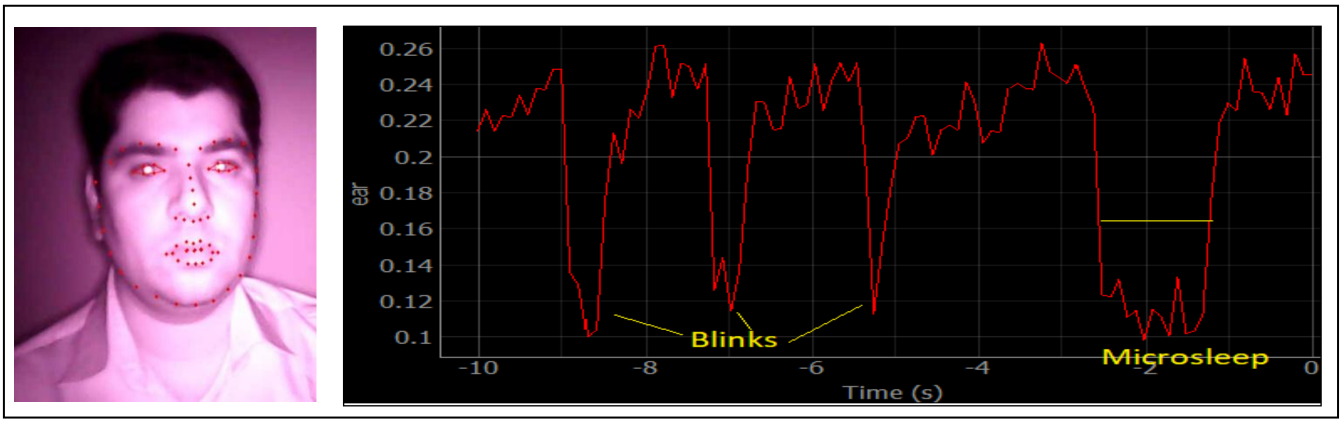


**Figure-12:** Results and eye detection in Dark light using IR Camera ( Measured EAR : 0.262, EAR Threshold Set to  : 0.17)

## 4.3 Result of Eye detection in dark light condition using IR camera under spectacles.

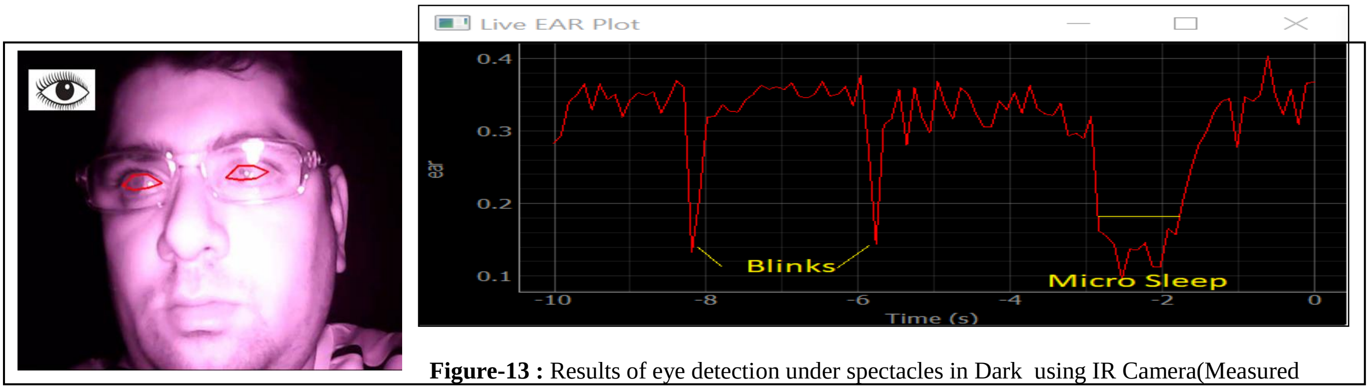


**Figure-13 :** Results of eye detection under spectacles in Dark  using IR Camera(Measured

EAR :0.262 ,Threshold 0.17)

## 4.4 Result of landmark detection on female face.

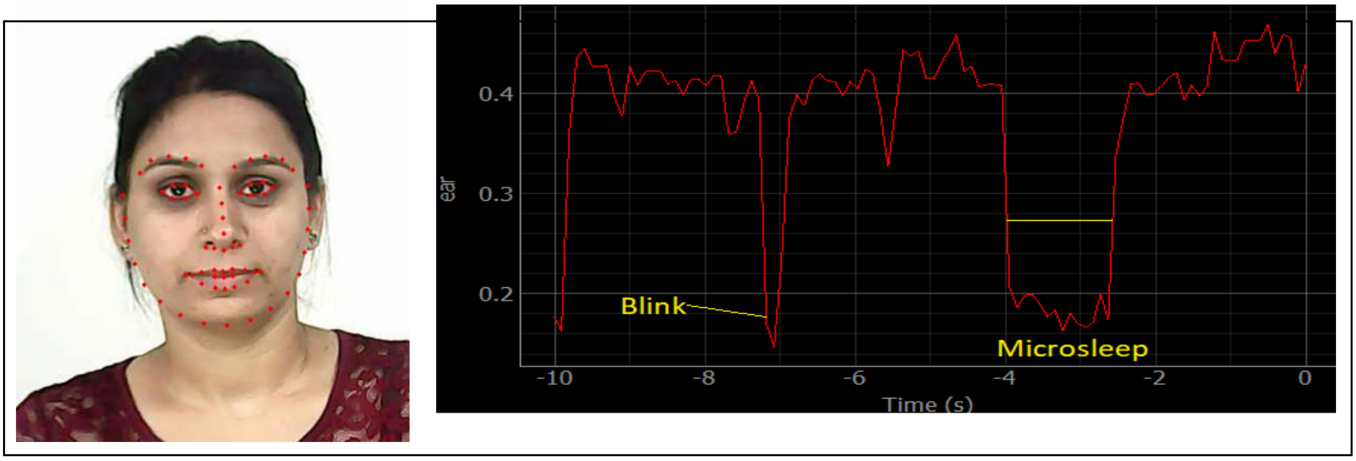


**Figure-14:** Results on female-1 face (EAR Plot on time scale, narrow pulse shows blink and 2 Sec Microsleep detected) Measured EAR : 0.429(Average of females-1)Threshold: 0.21

## 4.5 Accuracy of Blink Detection

Proposed algorithm was experimented on three male and three female faces for numbers of eye blinks as per below table.

Blink detection results were recorded and accuracy was calculated. Accuracy of proposed algorithm found better and also false detection reduced further from base paper.

| | **Accuracy Measurements from Base Paper [8]**<br>**Accuracy Measurements from Base Paper [8]**<br>**Driver Drowsiness Detection using Eye-Closeness Detection (2016 12th International Conference on Signal-Image Technology & Internet-Based Systems) 978-1-5090-5698-9/16 $31.00 © 2016 IEEE DOI 10.1109/SITIS.2016.110** | | | | | | **Measurements of Present Algorithm** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Gender** | **Total Experimented Blinks** | **True Detection** | **False Detection** | **Missed Blinks** | **% Accuracy of Blink Detection** | **% False Detection** | **% Missed Detection** | **True Detection** | **False Detection** | **Missed Blinks** | **% Accuracy of Blink Detection** | **% False Detection** | **% Missed Detection** |
| **Male-1** | **95** | 95 | 0 | 0 | **100.00** | 0.00 | 0.00 | 95 | 0 | 0 | **100.00** | 0.00 | 0.00 |
| **Male-2** | **103** | 102 | 0 | 1 | **99.03** | 0.00 | 0.97 | 103 | 0 | 0 | **100.00** | 0.00 | 0.00 |
| **Male-3** | **100** | 99 | 1 | 1 | **99.00** | 1.00 | 1.00 | 100 | 0 | 0 | **100.00** | 0.00 | 0.00 |
| **Female-1** | **99** | 98 | 0 | 1 | **98.99** | 0.00 | 1.01 | 99 | 0 | 0 | **100.00** | 0.00 | 0.00 |
| **Female-2** | **100** | 100 | 3 | 0 | **100.00** | 3.00 | 0.00 | 100 | 0 | 0 | **100.00** | 0.00 | 0.00 |
| **Female-3** | **100** | 100 | 1 | 0 | **100.00** | 1.00 | 0.00 | 99 | 0 | 1 | **99.00** | 0.00 | 1.00 |
| **Total** | **597** | **594** | **5** | **3** | **99.50** | **0.84** | **0.50** | **596** | **0** | **1** | **99.83** | **0.00** | **0.17** |

**Figure-15 :** Test Results

## 5. CONCLUSION

In present paper, author demonstrated how to build a blink and drowsiness detector using OpenCV, Python, and Dlib opensource Libraries by measuring EAR. The first step in building a blink detector is to perform facial landmark detection to localize the eyes in a given frame from a video stream. The eye aspect ratio for each eye can be calculated using Euclidian distance functions of OPEN CV , which is a singular value, relating the distances between the vertical eye landmark points to the distances between the horizontal landmark points. Once the eye aspect ratio calculated, algorithm can threshold it to determine if a person is blinking the eye aspect ratio will remain approximately constant when the eyes are open and then will rapidly approach zero during a blink, then increase again as the eye opens. The duration of blink further provide estimation of microsleep. The proposed algorithm has been tested on personal car driver for testing purposes. For authentic results, the camera position was focused on the driver's face. Further, the algorithm has been tested in day time driving and Night time using IR camera. The results are discussed in Result section and found satisfactory. The proposed algorithm focused solely on using the eye aspect ratio as a quantitative metric to determine if a person has blinked in a video stream. However, due to noise in a video stream, subpar facial landmark detections, or fast changes in viewing angle, a simple threshold on the eye aspect ratio could produce a false-positive detection, reporting that a blink had taken place when in reality the person had not blinked. To make our blink detector more robust to these challenges further following improvements can be implemented Computing the eye aspect ratio for the $N^{\text{th}}$ frame, along with the eye aspect ratios for $N - 6$ and $N + 6$ frames, then concatenating these eye aspect ratios to form a 13 dimensional feature vector. Training a Support Vector Machine (SVM) on these feature vectors. The combination of the temporal-based feature vector and SVM classifier helps reduce false-positive blink detections and improves the overall accuracy of the blink detector.

## 6. ACKNOWLEDGEMENT

Project is supported by Rayat Institute of Engineering and Information Technology . Author deem it as a proud privilege to express his sincerest regards and gratitude to thesis guide Er. Anudeep Goraya (Associate Professor) at Rayat Institute of Engineering and Information Technology for the invaluable support she gave him on many occasions and for many interesting discussions on many topics and theories related to this paper.

## 7. REFERENCES

[1] Association for Safe International Road Travel (ASIRT), Road Crash Statistics.http://asirt.org/initiatives/informing-roadusers/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_detection.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)