DROWSY DRIVER DETECTION

TEAM MEMBERS

KUSHAL N – ENG17CS0115

MANISH KUMAR.S – ENG17CS0121

MOHAMMED ZAHID PASHA – ENG17CS0129

MANOJKUMAR MANGALORE -ENG17CS0124

Overview

- **Abstract**
- **Problem statement**
- **Literature Survey**
- **Requirement Specification**
- **System Design**
- **System Architecture Diagram**

ABSTRACT

- The number of major road accidents that occur per day is on a rise and most of them are attributed to being the driver’s fault. According to the survey done in 2015, drivers are held responsible for approximately 78% of the accidents. To minimize the occurrence of these incidents a monitoring system that alerts the driver when he succumbs to sleep is proposed.

PROBLEM STATEMENT

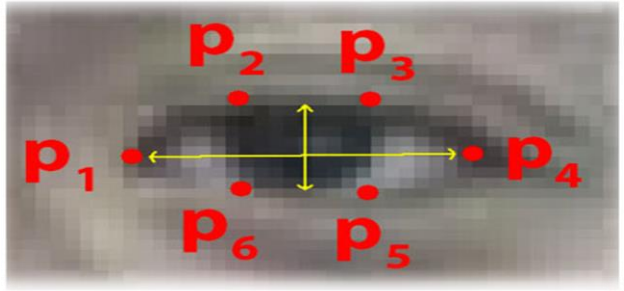
- Drowsy driving is a serious problem that leads to thousands of accidents each year. Motor vehicle collisions lead to significant death and disability as well as significant financial cost to both security and individual due to the driver impairments. Drowsiness is one of the factors for collisions. In India, no monitoring device is used to measure the drowsiness of driver.

Literature Survey

1. Real-Time Eye Blink Detection using Facial Landmarks

AUTHORS: Tereza Soukupova and Jan Cech

- Algorithms used:
- **[1].Description of features**
- For every video frame, the eye landmarks are detected. The eye aspect ratio (EAR) between height and width of the eye is computed.
- $$EAR = \frac{|p2-p6|+|p3-p5|}{2|p1-p4|}$$
- where p1,...,p6 are the 2D landmark locations, depicted in Fig.



- [2].Classification

  - It generally does not hold that low value of the EAR means that a person is blinking. A low value of the EAR may occur when a subject closes his/her eyes intentionally for a longer time or performs a facial expression,yawning,etc.,or the EAR captures a short random fluctuation of the landmarks.

Drawbacks :-

- They are Robust to low image quality.
- Fixed blink duration for all subjects was assumed, although everyone’s blink lasts differently.

2. Drowsy Driver Detection System Using Eye Blink Patterns

AUTHORS: Taner Danisman, Ian Marius Bilasco, Chabane Djeraba, Nacim Ihaddadene.

- Algorithms Used:
- [A].**Eye Blink Detection**
- In order to detect different eye patterns (Open and closed), we first perform a contrast stretching.To perform contrast stretching we use (1)(2)and (3)

$$I(x,y) = \text{pow}\left(\frac{I(x,y) - \text{low}_{in}}{\text{err}_{in}}, \gamma\right) * \text{err}_{out} + \text{low}_{out} \qquad (1)$$

$$\text{err}_{in} = \text{high}_{in} - \text{low}_{in} \qquad (2)$$

$$\text{err}_{out} = \text{high}_{out} - \text{low}_{out}. \qquad (3)$$

- Then, we horizontally divide the region of interest into two halves (Upper and Lower) with respect to the line passing through the center of the eye pupil.

- Because of the circular shape of the eye, an open eye pattern has a horizontal symmetry property, whereas a closed eye does not have this property. Therefore, we proposed to use symmetry property as a discrimination function for open and closed eyes.We use (4)(5)and(6) to find the horizontal symmetry of eye.
-

$$I_{dif} = VF(Up(I')) - \text{Low}(I'). \quad (4)$$

$$I_{sum} = \sum_{i=0,j=0}^{width,height} I_{dif}(i,j) \quad (5)$$

$$I_{state} = \begin{cases} Open & I_{sum} < T \\ Closed & I_{sum} \geq T \end{cases} \quad (6)$$

- **B. Drowsiness Detection**
- We define three states for the driver drowsiness as seen in TABLE. The typical eye blink duration is less than 400ms on average and 75ms for minimum. For this reason, we used TDrowsy=400ms and Tsleeping=800ms.

Drowsiness Level	Description
Awake	Blink durations < T_{Drowsy}
Drowsy	Blink durations > T_{Drowsy} and Blink durations < $T_{Sleeping}$
Sleeping	Blink durations > $T_{Sleeping}$

- Drowsiness detection is directly connected to the eye blink detection component. The timing of eye closed events is compared with the threshold values.

Drawbacks:

1. Presences of glasses effect the performance of this method.
2. No common database exists for comparing our results for drowsiness; therefore it only give the results for eye-blink detection.
3. presences of glasses affect the core components of the system including face detection, eye detection and symmetry calculation.

3. Drowsiness Detection Based on Eye Closure.

AUTHORS:B. Mohana, C. M. Sheela Rani

- Algorithms Used:
- [1]. **Pre-processing**
  - The video is obtained from a camera focused on the driver's face. The processing rate of the acquired video is 25 frames per second. These frames are then flipped and converted to grayscale.

- **[2].Facial mapping using Dlib**
  - The algorithm is implemented using Dlib python library that contains a landmark's facial detector with pre trained models. It estimates and maps a person's face in the form of facial points with 68 cartesian co-ordinates. The 68-point iBUG 300 dataset was used to train the dlib facial landmark predictor.

Landmarks as depicted by dlib facial predictor.

- [3].**Eye closure detection.**
  - Eye aspect ratio (EAR) is a parameter that determines eye state used to figure out if it is open or closed. It can be calculated using facial landmarks plotted by the 68 facial landmark point plot provided by python's dlib library. It uses the points in the region of interest to compute.
- $$EAR = \frac{|p2-p6|+|p3-p5|}{2|p1-p4|}$$
- where p1,...,p6 are the 2D landmark locations, depicted.

- **Drawbacks:**

- Occlusions such as spectacles are no hindrance in eye closure detection.
- Accuracy of detection decreases under bad illumination conditions.

Requirement Specification

Hardware Requirements

- Processor : Any Processor above 500 MHz
- Ram : 4GB
- Hard Disk : 512GB
- Input Device : Webcam,Standard Keyboard.
- Output Device : Buzzer,High display monitor.

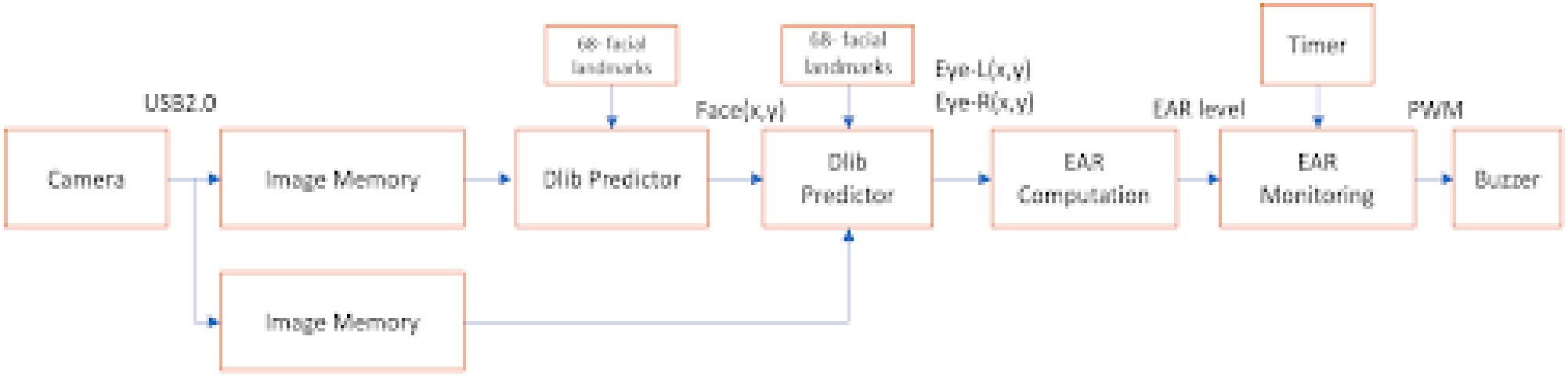
Software Requirements

- Operating System : Windows/UBUNTU
- Technology used : Python(version 3.7) ,opencv,dlib

System Design

System Architecture Diagram

- System architecture is the conceptual model that defines the structure, behavior, and the view of a system.



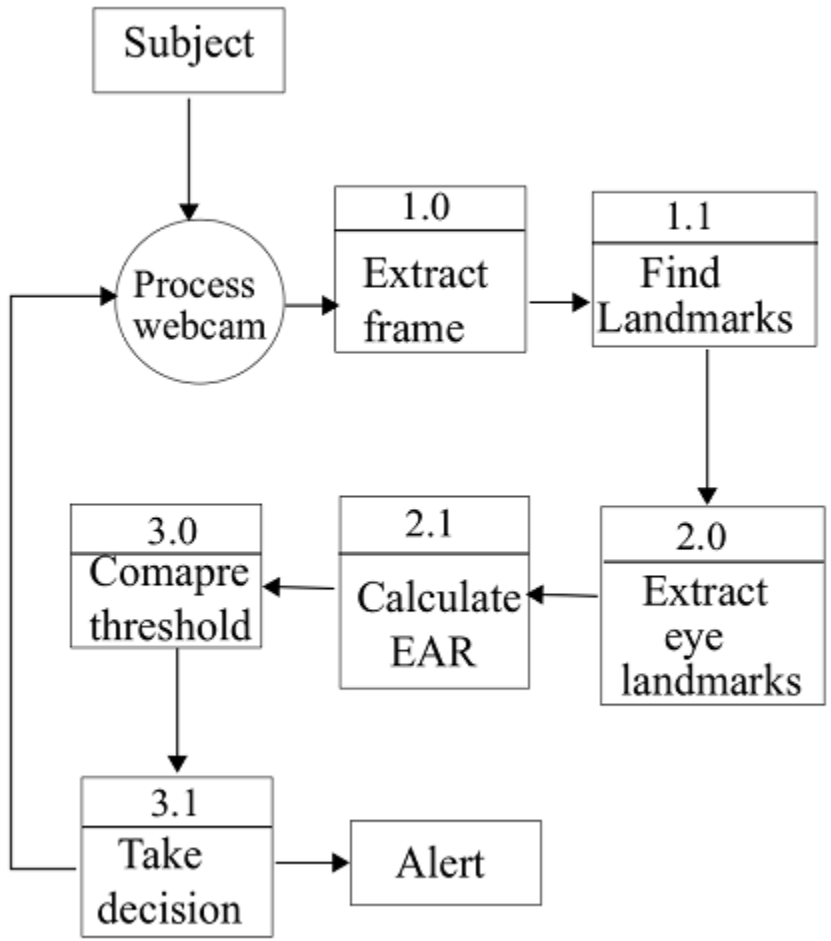
- The figure shows us the architecture of the developed model and we see that the computing eye aspect ratio (EAR) gives us the best results among the used algorithms.

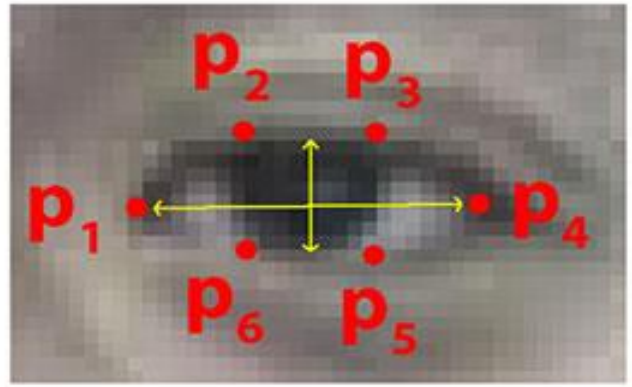
- The model basically uses dlib predictor using 68 facial landmarks to detect face and eyes of person.

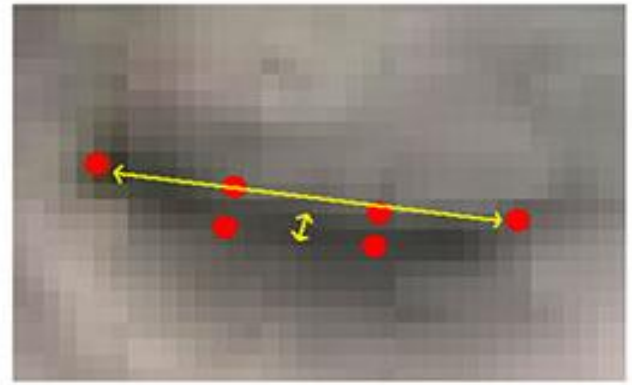
- After detecting the eyes of driver it calculates the EAR using the formula

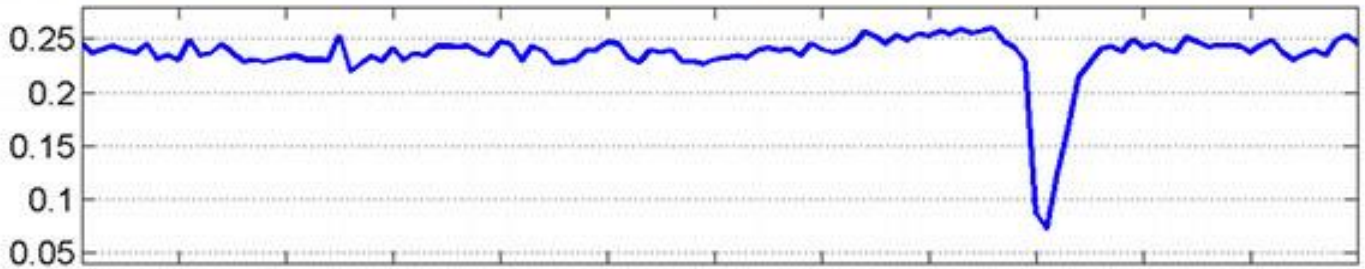
- $$EAR = \frac{|p2-p6|+|p3-p5|}{2|p1-p4|}$$

- It compares the EAR value with threshold value .If the EAR values is less than threshold value for consecutively 30 frames then message is passed to the driver using buzzer.


Subject
Process webcam
1.0
Extract frame
1.1
Find Landmarks
2.0
Extract eye landmarks
2.1
Calculate EAR
3.0
Comapre threshold
3.1
Take decision
Alert


p2
p3
p1
p4
p6
p5



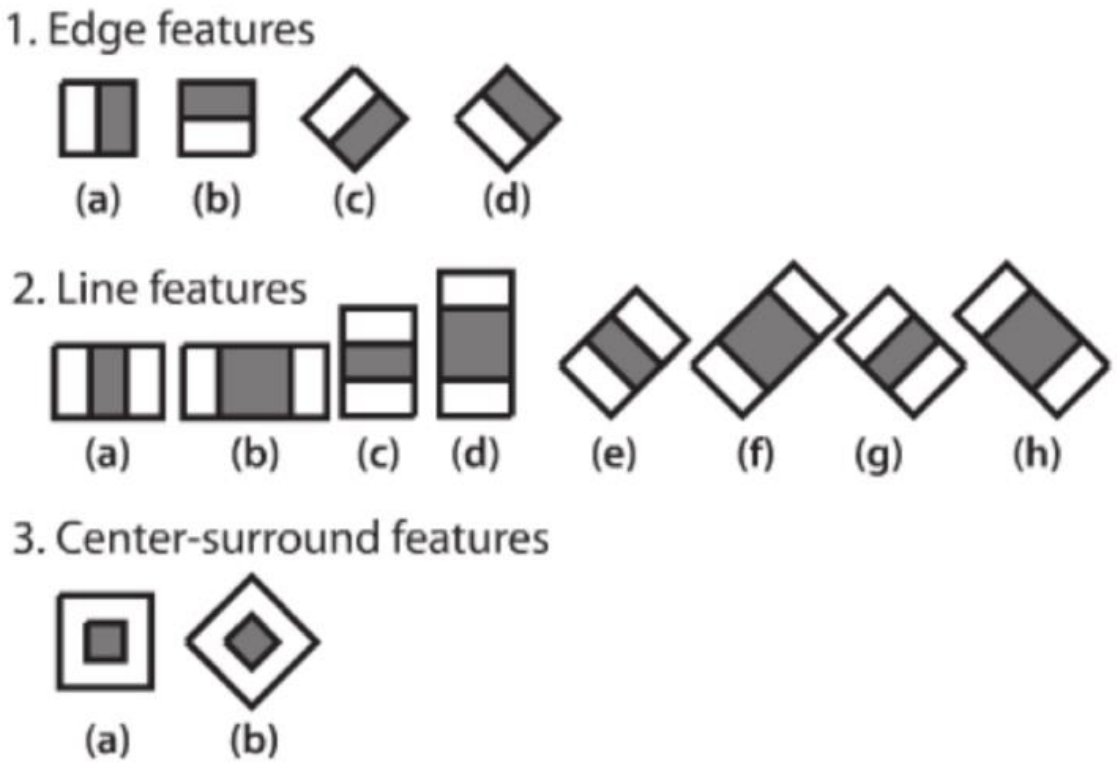

0.25
0.2
0.15
0.1
0.05

FACE DETECTION

- Face detection uses ***classifiers***, which are algorithms that detects faces in an image. Classifiers have been trained to detect faces using thousands to millions of images in order to get more accuracy. OpenCV uses two types of classifiers, LBP (Local Binary Pattern) and Haar Cascades. I will be using the latter classifier.

- Understanding Haar cascades:
- It is based on the Haar Wavelet technique to analyze pixels in the image into squares by function.
- Haar Cascades use the **Adaboost** learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers.

It has following features:



Face Detection determines the locations and sizes of human faces in arbitrary (digital) images.

In **Face Recognition**, the use of Face Detection comes first to determine and isolate a face before it can be recognized.

PSEUDO CODE:

detector = cv2.CascadeClassifier(args["cascade"])

predictor = dlib.shape_predictor(args["shape_predictor"])

- we use a faster detection algorithm (Haar cascades).Since it is faster then dlib face detector.

print("[INFO] starting video stream thread...")

vs = VideoStream(src=0).start()

For detecting faces in frame:

```
# loop over frames from the video stream
while True:
frame = vs.read()
frame = imutils.resize(frame, width=450)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
# detect faces in the grayscale frame
faces= detector.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30),
flags=cv2.CASCADE_SCALE_IMAGE)
```

```
# loop over the face detections
for (x, y, w, h) in rects:
rect = dlib.rectangle(int(x), int(y), int(x + w),
int(y + h))
# convert the facial landmark (x, y)-coordinates to a NumPy
# array
shape = predictor(gray, rect)
shape = face_utils.shape_to_np(shape)
```

Extracting left and right eye co-ordinates:

```
# extract the left and right eye coordinates, then use the
# coordinates to compute the eye aspect ratio for both eyes
leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]
leftEAR = eye_aspect_ratio(leftEye)
rightEAR = eye_aspect_ratio(rightEye)
# average the eye aspect ratio together for both eyes
ear = (leftEAR + rightEAR) / 2.0
```

Finding eye aspect ratio:

```
def eye_aspect_ratio(eye):
# compute the euclidean distances between the two sets of
# vertical eye landmarks (x, y)-coordinates
A = euclidean_dist(eye[1], eye[5])
B = euclidean_dist(eye[2], eye[4])
# compute the euclidean distance between the horizontal
# eye landmark (x, y)-coordinates
C = euclidean_dist(eye[0], eye[3])
# compute the eye aspect ratio
ear = (A + B) / (2.0 * C)
# return the eye aspect ratio
return ear
```

To check if the eye is closed and to alert the driver

```
if ear < EYE_AR_THRESH:
COUNTER += 1

# if the eyes were closed for a sufficient number of
# frames, then sound the alarm
if COUNTER >= EYE_AR_CONSEC_FRAMES:

# if the alarm is not on, turn it on
if not ALARM_ON:
ALARM_ON = True
else:
COUNTER = 0
ALARM_ON = False
```

References

[1] B. Mohana, C. M. Sheela Rani ,Drowsiness Detection Based on Eye Clousre and Yawning Detection,International Journal of Recent Technology and Engineering (IJRTE), November 2019

[2] Tereza Soukupov´a and Jan ˇCech Center for Machine Perception, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University in Prague, Real-Time Eye Blink Detection using Facial Landmarks, February 2016.

[3] Taner Danisman, Ian Marius Bilasco, Chabane Djeraba, Nacim Ihaddadene. ,Drowsy Driver Detection System Using Eye Blink Patterns, 28 Sep 2018

[4].Drowsiness detection with opencv ,Adrian Rodebrock on May 8,2017

Thank you