FATIGUE DETECTION AND ALERT SYSTEM FOR VEHICLES

A Project Report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

A. Revanth-1215316302

B. Sai Ram Varma-1215316310

Ch. Rajeev Varma-1215316312

M. Atal Varma-1215316330

Under the esteemed guidance of

Mrs. T. Jyotsna Rani

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GITAM

(Deemed to be University)

VISAKHAPATNAM

APRIL 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the project report entitled "**FATIGUE DETECTION AND ALERT SYSTEM FOR VEHICLES**" is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date:

Registration No.	**Name**	**Signature**
1215316302	**A.Revanth**	
1215316310	**B. Sai Ram Varma**	
1215316312	**Ch. Rajeev Varma**	
1215316330	**M. Atal Varma**	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled "**FATIGUE DETECTION AND ALERT SYSTEM FOR VEHICLES**" is a bona-fide record of work carried out by students **A.Revanth-1215316302, B.Sai Ram Varma-1215316310, Ch.Rajeev Varma-1215316312, M.Atal Varma-1215316330** submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide | **Head of the Department**

Mrs. T. Jyotsna Rani | **Dr. K. Thammi Reddy**

Assistant Professor | **Professor**

ACKNOWLEDGEMENTS

The project opportunity I had was a great chance for learning and professional development. Therefore, I consider myself a very lucky individual as I was provided with an opportunity to be a part of it.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to Prof **Mrs.T.jyotsna Rani** who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path.

I express my deepest thanks toast Asst Prof **Dr. P.Sankara Rao** and Asst Prof **E.Neelima** for taking part in useful decisions & giving necessary advice and guidance, as mentors. I choose this moment to acknowledge their contribution gratefully.

I would also like to thank Asst Prof, **Mr . P**. **Surya Chandra**, A.M.C who helped us a lot in the successful completion of mini project report.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

Sincerely

A. Revanth

B. Sai Ram Varma

Ch.Rajeev Varma

M.Atal Varma.

TABLE OF CONTENTS

1.ABSTRACT

On road driver's fatigue and drowsiness is contributing more than 30% 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 driver's performance or by combining all the 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 continuous frames and facial landmarks will be 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 are implemented by monitoring EAR against a threshold value. Drowsiness level is displayed on monitor screen with microsleep detection audio warning.

2.INTRODUCTION

This chapter outlines the need for fatigue detection system in vehicles and the overview of the real time image processing system designed in this project. Driver fatigue is a significant factor in a large number of vehicle accidents. Recent statistics estimate that annually 1,200 deaths and 76,000 injuries can be attributed to fatigue related crashes. There has been ample development in the field of safety in case of accidents in the form of air bags, etc. However, the development of technologies for detecting or preventing drowsiness at the wheel is a major challenge in the field of accident avoidance systems. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its affects. The aim of this project is to develop a prototype drowsiness detection system. This system will primarily measure and record the real time features of the driver or the driving pattern and continuously evaluate them on the basis of the levels predetermined to indicate fatigue. The driver may show signs of fatigue in numerous ways. The project surveys different methods of eye detection and checking whether the detected eyes are open or closed. On the basis of the number of times the eyes are found to be closed, it can be determined whether the person is drowsy or not. Another method of fatigue detection is the evaluation of the features of the mouth. If the driver yawns, showing signs of fatigue, it can be used to trigger the alarm system. Expressions indicating extreme anger, disappointment, shock, excitement, etc. show that the driver is not in the best state to drive and may falter at it. This should be used to remind the driver in the form of an appropriate alarm to either park the car or consciously regain the composure to drive safely.

Fatigue detection in the non-intrusive form may be done efficiently by increasing the number of parameters on which the driver is being monitored. This will lead to a complex algorithm which can be easily flexible to detect fatigue. Such a driver fatigue detection system has tremendous scope in the car industry. There are millions of cars being manufactured every year and it is a universally accepted fact that fatigue among drivers is a potent accident factor. Hence, car manufacturing companies are the biggest market for such a system.

Mercedes Benz has introduced a fatigue detection system in certain models of high-end cars. However, the proposed prototype that this project aims at creating shall be far more inexpensive as compared to that model, thus creating a bigger market for this product. Indian roads are prone to fatigue related accidents essentially among truck drivers, long distance bus drivers and BPO employees during the late hours. Introduction of such a system in long distance vehicles and in public transport would reduce the number of accidents. 4 1.1 Need for the project Several studies have investigated the relationship between driver fatigue and crash risk and have attempted to quantify the risk increase.

In a case control study of New Zealand drivers, Connor et al compared 571 crash-involved drivers with 588 non-crash involved drivers driving in the same area and at the same times. Driver variables were taken from accident registration and additional interviews. Taking into account possible confounding variables (gender, age, socio-economic status, annual kilometers, speed, road type), they found a strong relationship between acute fatigue (based on loss of sleep the night before) and crash involvement. Crash risk was eight times higher for drivers with a score? 4 on the Stanford Sleepiness Scale (95percent confidence interval 3.4-19.7); 5,5 times higher for driving between 2 and 5 am (95 percent interval 1.4-22.7); and almost 3 times higher when drivers had slept for less than 5 hours in the past 24-hour period (95percent confidence interval 1.4-5.4).

In a case-control study, Cummings et al compared crash-involved drivers with a similar group of non-crash involved drivers at the same location, direction, time and day. They found the crash risk was fourteen times higher for drivers who had reported to have almost fallen asleep behind the wheel (95 percent confidence interval 1.4-147).[2] The data collected in the 100 Car Naturalistic Driving Study shows that driving while fatigued increases a driver’s risk of involvement in a crash or near-crash by nearly four times. Studies of professional drivers (bus, lorry, truck) show that it takes around 9 or 10 hours of driving, or 11 hours of work, before crash risk starts to rise.

Hamelin found that after 11 hours of work span the crash risk doubles. The effect of task duration is practically always entangled with the effects of the time of day and sometimes also with the length of time awake and previous lack of sleep. The duration of a trip may be of lesser importance compared to these other factors - many fatigue-related accidents occur after driving for only a few hours. Short trips can also end up in fatigue-related crashes because time of day and long and irregular working hours are stronger predictors of fatigue than time spent driving. Connor et al. also note blind spots in the research on driver fatigue.

The association of non-medical (lifestyle) determinants of fatigue with crash has not been the subject of thorough research. There is still a lack of knowledge concerning the contribution of increasing total hours of work, and shift schedules to driver fatigue. Whereas research into fatigue and sleep apnea in truck drivers has led to awareness of these problems and some modification of work conditions, occupationally induced fatigue in potentially much larger numbers of commuters has received little attention.

The Federal Motor Carrier Safety Administration (FMCSA), the trucking industry, highway safety advocates, and transportation researchers have all identified driver drowsiness as a high priority commercial vehicle safety issue. Drowsiness affects mental alertness, decreasing an individual's ability to operate a vehicle safely and increasing the risk of human error that could lead to fatalities and injuries. Furthermore, it has been shown to slow reaction time, decreases awareness, and impairs judgment. Long hours behind the wheel in monotonous driving environments make truck drivers particularly prone to drowsy-driving crashes. 5 Successfully addressing the issue of driver drowsiness in the commercial motor vehicle industry is a formidable and multi-faceted challenge. Operational requirements are diverse, and factors such as work schedules, duty times, rest periods, recovery opportunities, and response to customer needs can vary widely. In addition, the interaction of the principal physiological factors that underlie the formation of sleepiness, namely the homeostatic drive for sleep and circadian rhythms, are complex.

While these challenges preclude a single, simple solution to the problem, there is reason to believe that driver drowsiness can nevertheless be effectively managed, thus resulting in a significant reduction in related risk and improved safety. Addressing the need for a reduction in crashes related to driver drowsiness in transportation will require some innovative concepts and evolving methodologies. In-vehicle technological approaches, both available and emerging, have great potential as relevant and effective tools to address fatigue. Within any comprehensive and effective fatigue management program, an on-board device that monitors driver state in real time may have real value as safety net. Sleepy drivers exhibit certain observable behavior, including eye gaze, eyelid movement, pupil movement, head movement, and facial expression. Non-invasive techniques are currently being employed to assess a driver's alertness level through the visual observation of his/her physical condition using a remote camera and state-of-the-art technologies in computer vision. Recent progress in machine vision research and advances in computer hardware technologies have made it possible to measure head pose, eye gaze, and eyelid movement accurately and in real time using video cameras.

The alertness monitoring technologies monitor - usually on-line and in real time - bio-behavioral aspects of the operator; for example, eye gaze, eye closure, pupil occlusion, head position and movement, brain wave activity, and heart rate. To be practical and useful as driver warning systems, these devices must acquire, interpret, and feed-back information to the operator in real world driving environments. As such, there exists a need and, thus, ongoing efforts are underway, to validate operator-based, on-board fatigue monitoring technologies in a real-world naturalistic driving environment. In view of the need for non-intrusive relatively inexpensive modules for fatigue detection in vehicles, this project has been formulated to monitor the driver's actions and reactions, check for fatigue and on detection of fatigue or drowsiness, stimulate the appropriate plan of action, which could be an alarm or deceleration of the vehicle, etc

3.LITERATURE REVIEW

3.1 FACE AND EYE DETECTION BY CONVOLUTIONAL NEURAL NETWORKS ALGORITHMS

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 vehicle's crashes. Driver fatigue their time requirement is a fraction of the previously used methods. The algorithm starts with the detection of heads on color pictures using deviations in color 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 FACE DETECTION USING HAAR CASCADE

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.

3.3 EYE DETECTION USING MORPHOLOGICAL AND COLOR IMAGE PROCESSING

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.

3.4 ALGORITHM FOR EYE DETECTION ON GREY INTENSITY FACE

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 centers 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.

3.5 REAL-TIME FACE DETECTION USING EDGE ORIENTATION MATCHING

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 than 93% on that database.

4.PROBLEM IDENTIFICATION AND OBJECTIVES

4.1 PROBLEM IDENTIFICATION

The attention level of driver degrades because of less sleep, long continuous driving or any other medical condition like brain disorders etc. Several surveys on road accidents says that around 30 percent of accidents are caused by fatigue of the driver. When driver drives for more than normal period for human then excessive fatigue is caused and also results in tiredness which drives the driver to sleepy condition or loss of consciousness

4.2 OBJECTIVES

4.2.1 Efficiency

Our proposed method is to design and develop a low-cost system, which is based on embedded platform for drowsiness detection that keeps the driver focused on the road.

Many designs and prototypes have been implemented in automobiles to avoid such accidents by keeping the whole focus and concentration on accurately monitoring the open and closed state of the driver's eye in real time.

4.2.2 Safety

Nowadays the driver safety in the car is one of the most wanted system to avoid accidents. Our objective of the project is to ensure the safety system. For enhancing the safety, we are detecting the eye blinks of the driver and estimating the driver status and control the car accordingly.

4.2.3 Non-intrusive Approach

Using a non-intrusive drowsiness detection system is the most promising direction to build a real-life applicable solution. A solution that could be enhanced by taking advantage of different methods.

5.SYSTEM METHODOLOGY

5.1 PROJECT WORK FLOW

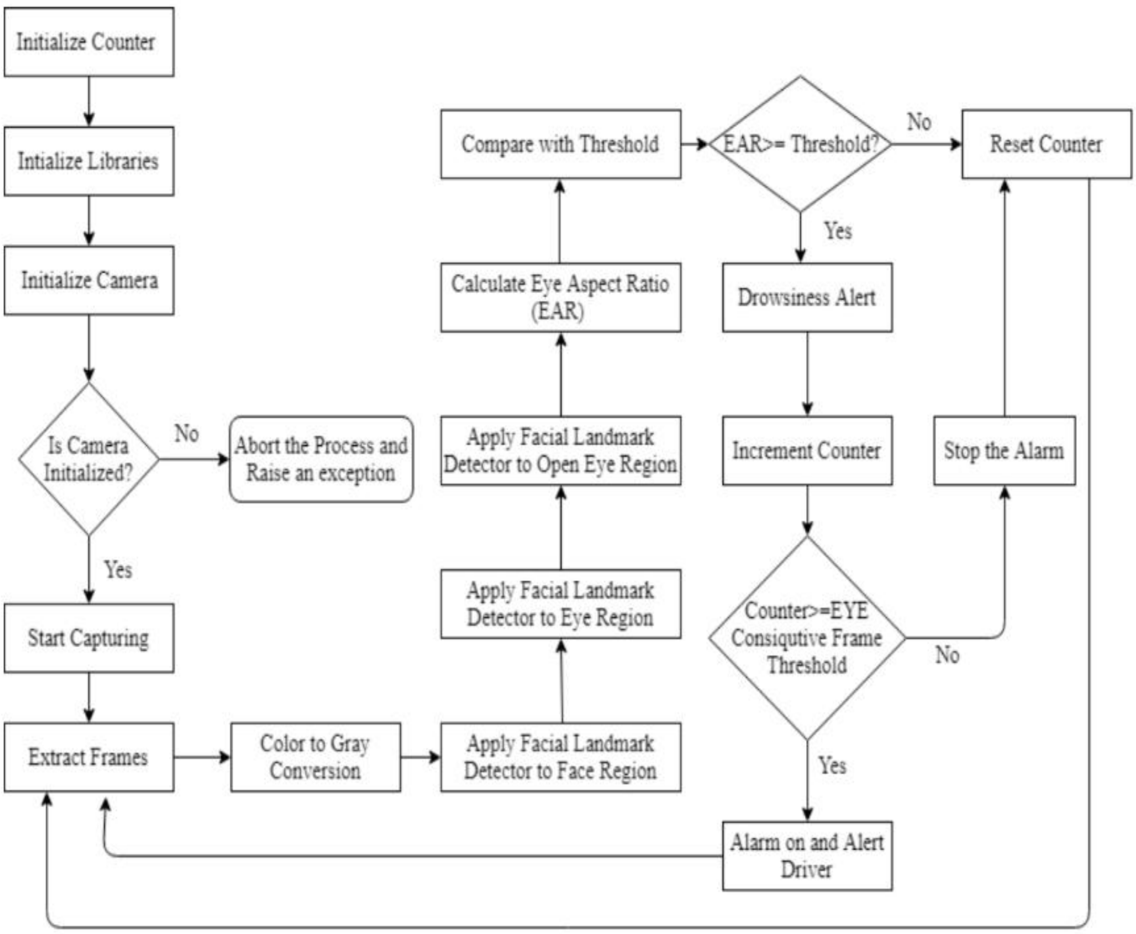

Fig:5.1 Project work flow

5.2 State Diagram

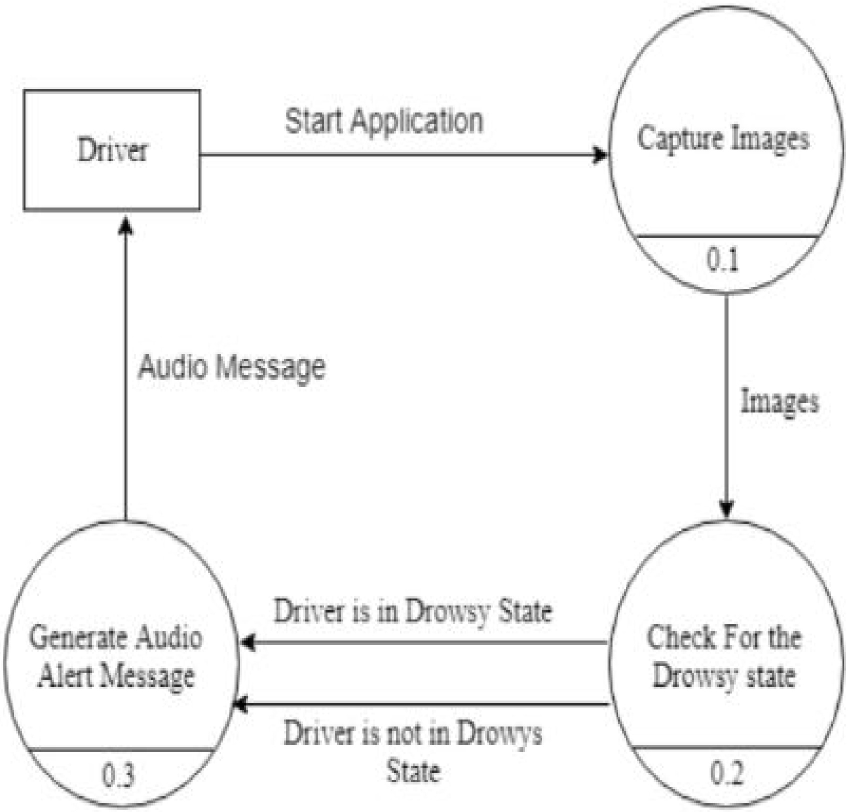

Fig:5.2 State diagram

5.3 Sequence diagram

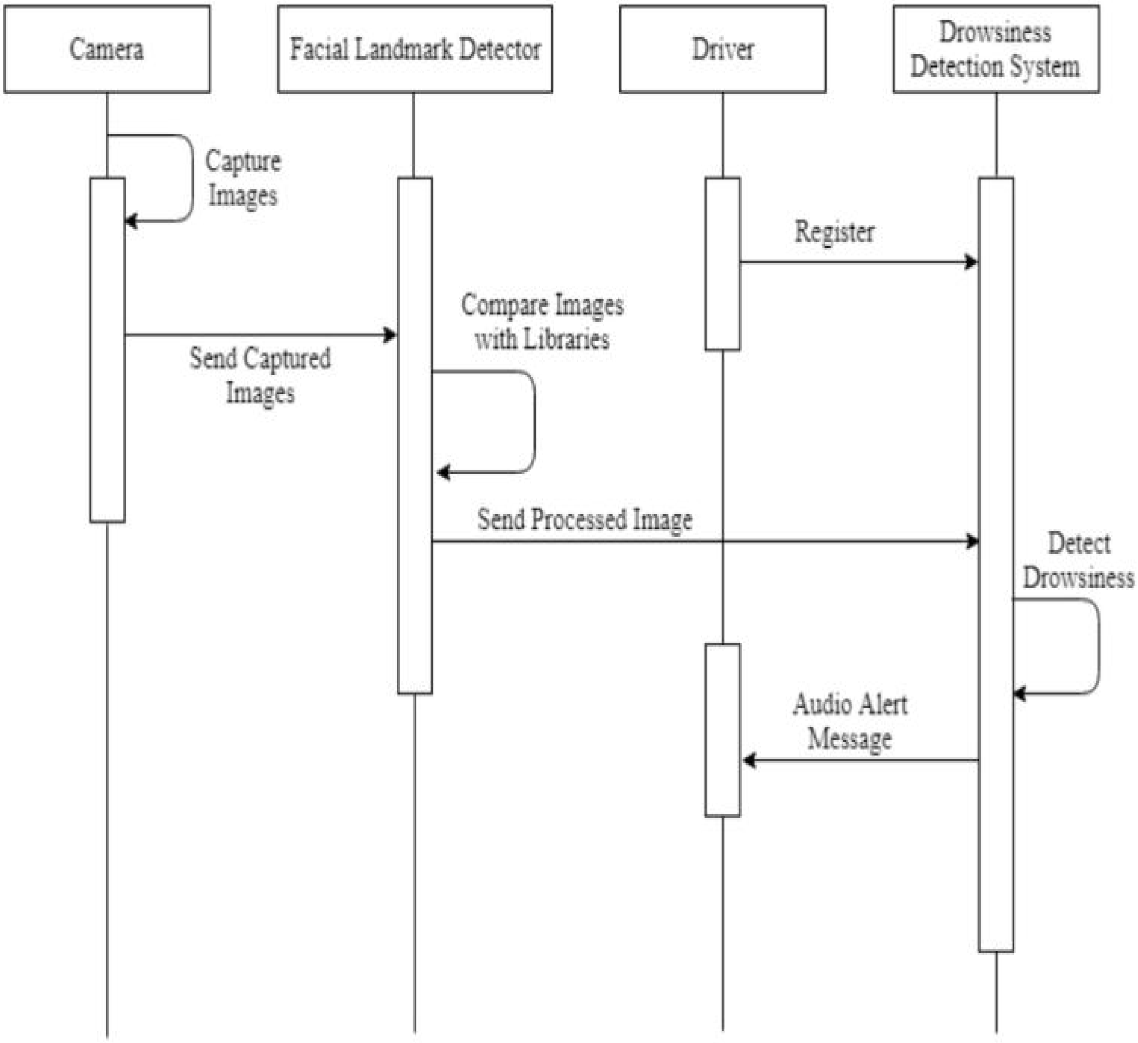

Fig:5.3 Sequence diagram

6.OVERVIEW OF TECHNOLOGIES

6.1 HARDWARE REQUIRMENTS

1. Processor intel-i5 onwards
2. Solid State Disk-1gb
3. System Memory-4gb
4. Camera-30fps

6.2 SOFTWARE REQUIREMENTS

1. Python 3.7
2. Open-CV
3. Python library dlib
4. Shape_predictor_68_face_landmarks.dat
5. HAAR Cascading files

7.IMPLEMENTATION

7.1 CODE

```
# import the necessary packages
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import playsound
import argparse
import imutils
import time
import dlib
import cv2
 def sound_alarm(path):
```

```
# play an alarm sound
	playsound.playsound(path)
def eye_aspect_ratio(eye):

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

```
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--shape-predictor", required=True,
    help= 'C:\\Users\\saira\\project')
ap.add_argument("-a", "--alarm", type=str, default="",
    help='C:\\Users\\saira\\project')
ap.add_argument("-w", "--webcam", type=int, default=0,
    help='0')
args = vars(ap.parse_args())

# define two constants, one for the eye aspect ratio to indicate
# blink and then a second constant for the number of consecutive
# frames the eye must be below the threshold for to set off the
# alarm
EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 48
```

```
# initialize the frame counter as well as a boolean used to
# indicate if the alarm is going off
COUNTER = 0
ALARM_ON = False

# initialize dlib's face detector (HOG-based) and then create
# the facial landmark predictor
print("[INFO] loading facial landmark predictor...")
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])

# grab the indexes of the facial landmarks for the left and
# right eye, respectively
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
```

```
# start the video stream thread
print("[INFO] starting video stream thread...")
vs = VideoStream(src=args["webcam"]).start()
time.sleep(1.0)
# loop over frames from the video stream
while True:
    # grab the frame from the threaded video file stream, resize
    # it, and convert it to grayscale
    # channels)
    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # detect faces in the grayscale frame
    rects = detector(gray, 0)
    # loop over the face detections
    for rect in rects:
```

```
# determine the facial landmarks for the face region, then
	# convert the facial landmark (x, y)-coordinates to a NumPy
	# array
	shape = predictor(gray, rect)
	shape = face_utils.shape_to_np(shape)
	# 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
	# compute the convex hull for the left and right eye, then
	# visualize each of the eyes
	leftEyeHull = cv2.convexHull(leftEye)
	rightEyeHull = cv2.convexHull(rightEye)
```

```
cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
 cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
 # check to see if the eye aspect ratio is below the blink
 # threshold, and if so, increment the blink frame counter
 if ear < EYE_AR_THRESH:
     COUNTER += 1
     # if the eyes were closed for a sufficient number of
     # 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
             # check to see if an alarm file was supplied,
             # and if so, start a thread to have the alarm
             # sound played in the background
```

```
if args["alarm"] != "":
            t = Thread(target=sound_alarm,
                args=(args["alarm"],))
            t.deamon = True
            t.start()
        # draw an alarm on the frame
        cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
    # otherwise, the eye aspect ratio is not below the blink
    # threshold, so reset the counter and alarm
    else:
        COUNTER = 0
        ALARM_ON = False
    # draw the computed eye aspect ratio on the frame to help
    # with debugging and setting the correct eye aspect ratio
    # thresholds and frame counters
    cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
```

```
# show the frame
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
        break

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

7.1.1 OpenCV

OpenCV stands for Open Source Computer Vision. It's an Open Source BSD licensed library that includes hundreds of advanced Computer Vision algorithms that are optimized to use hardware acceleration. OpenCV is commonly used for machine learning, 4 image processing, image manipulation, and much more. OpenCV has a modular structure. There are shared and static libraries and a CV Namespace. In short, OpenCV is used in our application to easily load bitmap files that contain landscaping pictures and perform a blend operation between two pictures so that one picture can be seen in the background of another picture. This image manipulation is easily performed in a few lines of code using OpenCV versus other methods. OpenCV.org is a must if you want to explore and dive deeper into image processing and machine learning in general.

7.1.2 Eye detection

In the system we have used facial landmark prediction for eye detection Facial landmarks are used to localize and represent salient regions of the face. Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more. In the context of facial landmarks, our goal is detecting important facial structures on the face using shape prediction methods..

This method starts by using

1. A training set of labeled facial landmarks on an image. These images are manually labeled, specifying specific (x, y)-coordinates of regions surrounding each facial structure.

2. Priors, of more specifically, the probability on distance between pairs of input pixels. The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face

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

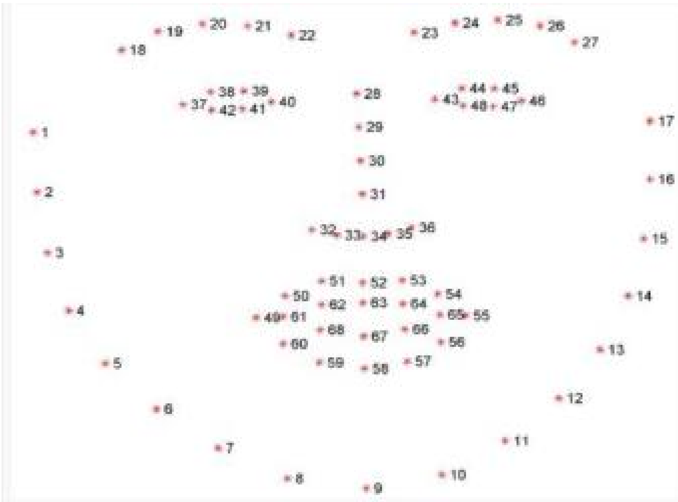

Fig:7.1 Visualizing the 68 facial landmark coordinates

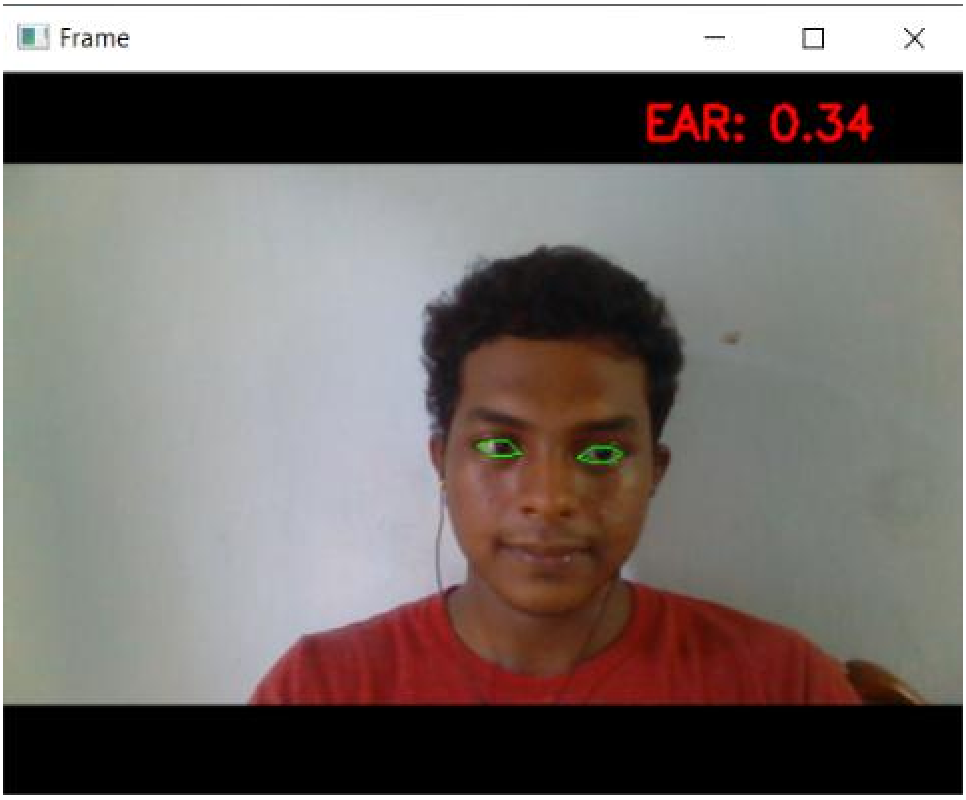

Fig:7.2 Detection of both the eyes

We can detect and access both the eye region by the following facial landmark index show below

• The right eye using [36, 42].

• The left eye with [42, 48].

These annotations are part of the 68-point iBUG 300-W dataset which the dlib facial landmark predictor was trained on. It’s important to note that other flavors of facial landmark detectors exist, including the 194-point model that can be trained on the HELEN dataset.

7.1.3 Eye Aspect Ratio Calculation

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 below. The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye. It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among individuals, and it is fully invariant to a uniform scaling of the image and in-plane rotation of the face. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes is averaged.

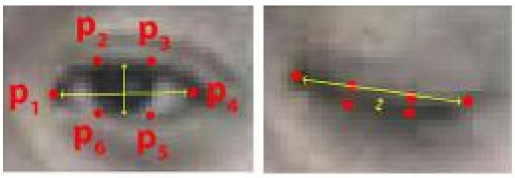

Fig:7.3 Eye landmarks

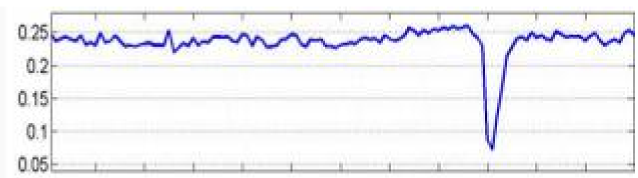

Fig7.4 EAR for single blink

7.2 TESTING

7.2.1 Eye State Determination

Finally, the decision for the eye state is made based on EAR calculated in the previous step. If the distance is zero or is close to zero, the eye state is classified as “closed” otherwise the eye state is identified as “open”

7.2.2 Drowsiness Detection

The last step of the algorithm is to determine the person’s condition based on a pre-set condition for drowsiness. The average blink duration of a person is 100-400 milliseconds (i.e. 0.1-0.4 of a second). Hence if a person is drowsy his eye closure must be beyond this interval. We set a time frame of 5 seconds. If the eyes remain closed for five or more seconds, drowsiness is detected and alert pop regarding this is triggered

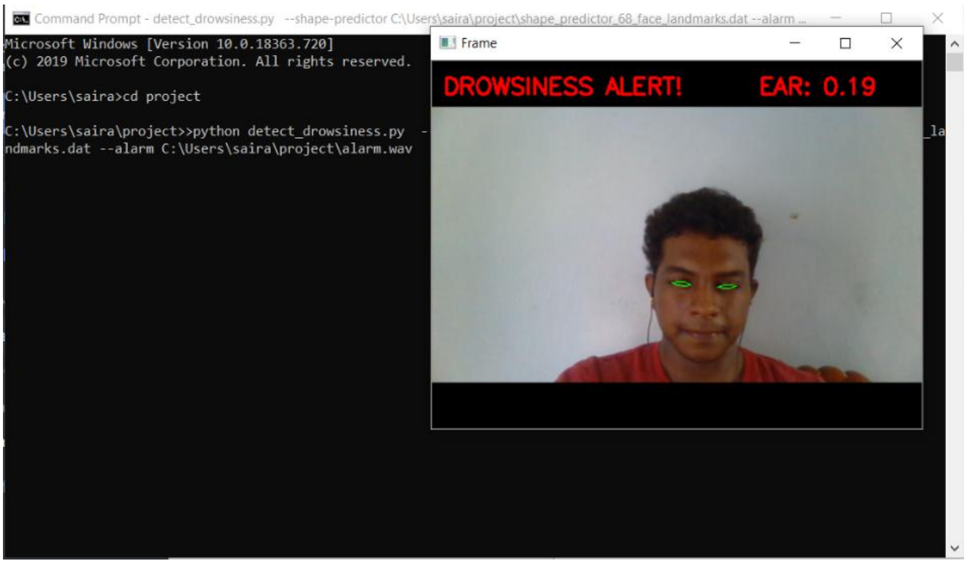

Fig:7.5 drowsiness state

8.RESULTS AND DISCUSSIONS

Implementation of drowsiness detection with Python and OpenCV was done which includes the following steps: Successful runtime capturing of video with camera. Captured video was divided into frames and each frame were analyzed. Successful detection of face followed by detection of eye. If closure of eye for successive frames were detected, then it is classified as drowsy condition else it is regarded as normal blink and the loop of capturing image and analyzing the state of driver is carried out again and again. In this implementation during the drowsy state the eye is not surrounded by circle or it is not detected, and corresponding message is shown.

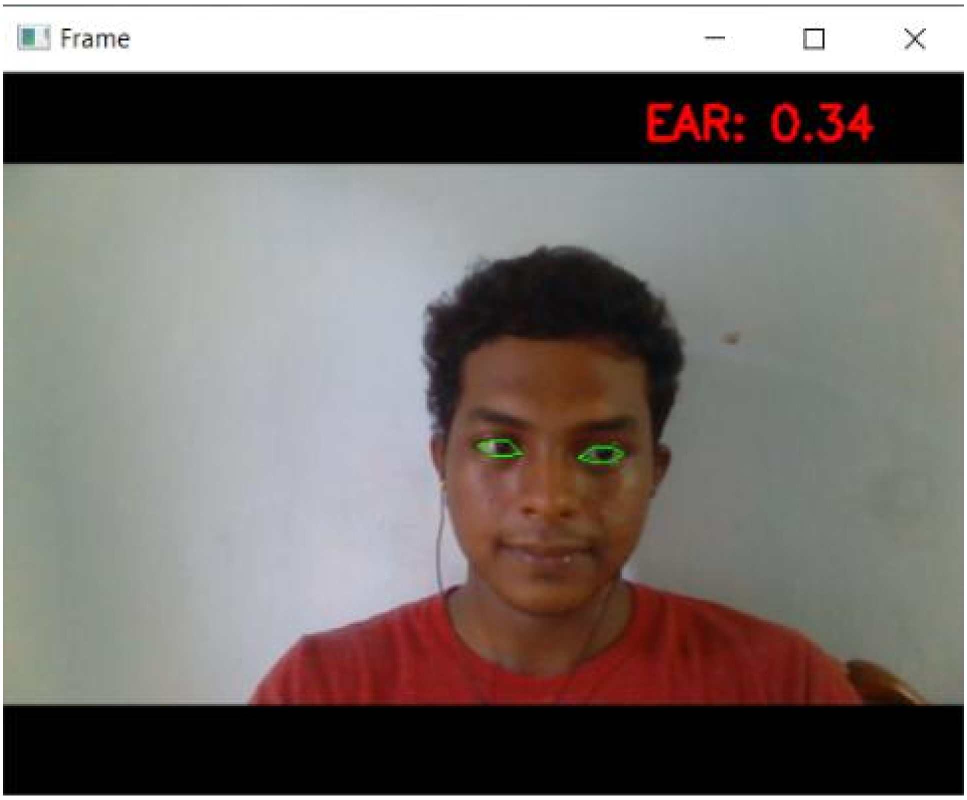

FIG:8.1 Subject is awake

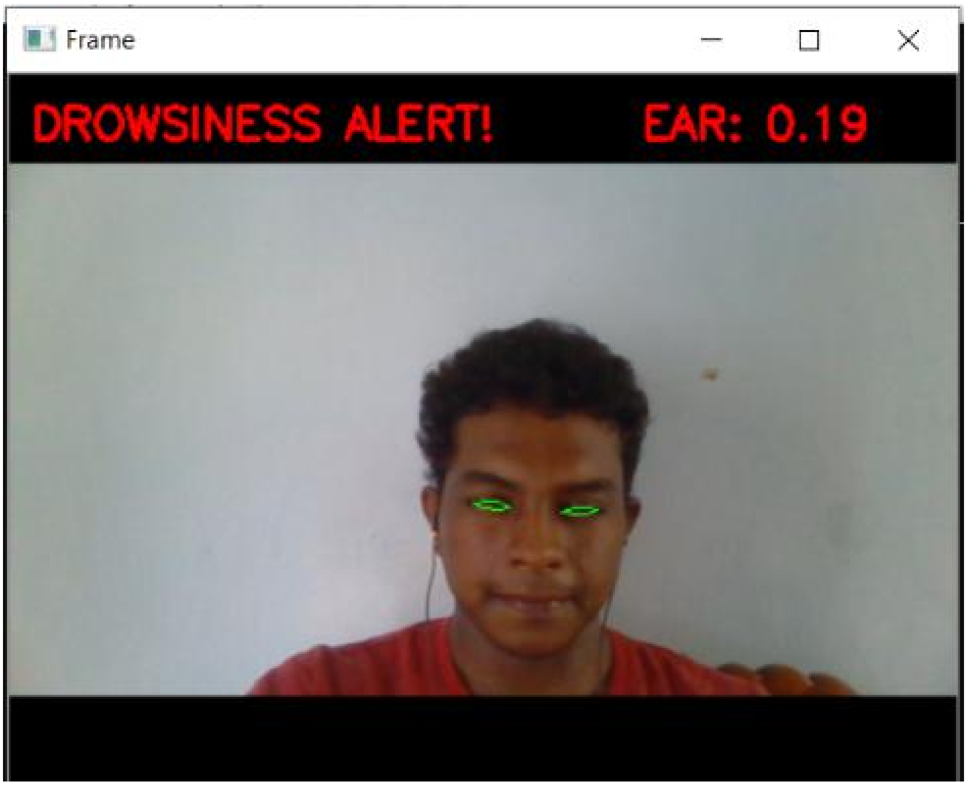

Fig:8.2 Subject is drowsy

8.1 DISCUSSIONS

A non-intrusive system to localize the eyes and monitor fatigue was developed. Information about the head and eyes position are obtained through various self-developed image processing algorithms. During the monitoring, the system is able to decide whether the eyes are opened or closed. When the eyes have been closed for two seconds, a warning signal is issued. In addition, during monitoring, the system is able to automatically detect any eye localizing error that might have occurred. In case of this type of error, the system is able to recover and properly localize the eyes.

Our model is designed for detection of drowsy state of eye and give and alert signal or warning in the form of audio alarm. But the response of driver after being warned may not be enough to stop causing the accident meaning that if the driver is slow in responding towards the warning signal then accident may occur. Hence to avoid this we can design and fit a motor driven system and synchronize it with the warning signal so that the vehicle will slow down after getting the warning signal automatically.

9.CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION

A real-time eye blink detection algorithm was presented. We quantitatively demonstrated that Haar feature-based cascade classifiers and regression-based facial landmark detectors are precise enough to reliably estimate the positive images of face and a level of eye openness. While they are robust to low image quality (low image resolution in a large extent) and in-thewild.

Limitations

Use of spectacles

In case the user uses spectacle then it is difficult to detect the state of the eye. As it hugely depends on light hence reflection of spectacles may give the output for a closed eye as opened eye. Hence for this purpose the closeness of eye to the camera is required to avoid light.

Multiple face problem

If multiple face arises in the window then the camera may detect more number of faces undesired output may appear. Because of different condition of different faces. So, we need to make sure that only the driver face come within the range of the camera. Also, the speed of detection reduces because of operation on multiple faces

9.2 FUTURE SCOPE

Our model is designed for detection of drowsy state of eye and give and alert signal or warning in the form of audio alarm. But the response of driver after being warned may not be enough to stop causing the accident meaning that if the driver is slow in responding towards the warning signal then accident may occur. Hence to avoid this we can design and fit a motor driven system and synchronize it with the warning signal so that the vehicle will slow down after getting the warning signal automatically. 17 We can also provide the user with an Android application which will provide with the information of his/her drowsiness level during any journey. The user will know Normal state, Drowsy State, the number of times blinked the eyes according to the number of frames captures.

10.REFERENCES

IEEE standard

Journal Paper,

[1] Facial Features Monitoring for Real Time Drowsiness Detection by Manu B.N, 2016 12th International Conference on Innovations in Information Technology (IIT) [Pg. 78-81]

https://ieeexplore.ieee.org/document/7880030

[2] Real Time Drowsiness Detection using Eye Blink Monitoring by Amna Rahman Department of Software Engineering Fatima Jinnah Women University 2015 National Software Engineering Conference (NSEC 2015)

https://ieeexplore.ieee.org/document/7396336

[3] Implementation of the Driver Drowsiness Detection System by K. Srijayathi International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 9, September 2013

Names of Websites referred

https://realpython.com/face-recognition-with-python/

https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/

https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/

https://www.codeproject.com/Articles/26897/TrackEye-Real-Time-Tracking-OfHumanEyesUsing-a

https://docs.opencv.org/3.4/d7/d8b/tutorial_py_face_detection.html

https://www.learnopencv.com/training-better-haar-lbp-cascade-eye-detector-opencv/