DRIVER DROWSINESS DETECTION SYSTEM

Application Development -II Report Submitted
In partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

By

R. TARUNI	**21N31A05L4**
T. JAYANTH NAIDU	**21N31A05P6**
T. AJAY	**21N31A05P4**

Under the esteemed guidance of
MS. C. S. N. DURGA
Associate Professor

Department of Computer Science and Engineering

Malla Reddy College of Engineering & Technology

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with 'A' Grade)
Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100
Website: www.mrcet.ac.in

2023-2024

Malla Reddy College of Engineering & Technology

(Autonomous Institution- UGC, Govt. of India)
(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with 'A' Grade)
Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100
website: www.mrcet.ac.in

CERTIFICATE

This is to certify that this is the Bonafide record of the Application development-2 entitled **"Driver Drowsiness Detection System"** submitted by **R.Taruni (21N31A05L4) , T.Jayanth Naidu (21N31A05P6) and T.Ajay (21N31A05P4)** of B.Tech in the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering, Department of CSE during the year 2023-2024.

Internal Guide	**Head of the Department**
Dr. S. RAHAMAT BASHA **Associate Professor**	**Dr. S. SHANTHI** **Professor**

External Examiner

DECLARATION

We hereby declare that the project titled "**Driver Drowsiness Detection System**" submitted to Malla Reddy College of Engineering and Technology (UGC Autonomous), affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a result of original research carried-out in this thesis. It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

R. Taruni - (21N31A05L4)

T. Jayanth Naidu - (21N31A05P6)

T. Ajay - (21N31A05P4)

ABSTRACT

In recent years, driver fatigue is one of the major causes of vehicle accidents in the world. A direct way of measuring driver fatigue is measuring the state of the driver i.e., drowsiness. So, detecting drowsy driver is very important to avoid all these mishaps. The ultimate aim of this study is to create a smart alert system which can be used in building intelligent vehicles that can automatically avoid drowsy driver impairment. Though there are several methods for measuring the drowsiness, this approach is completely non-intrusive which does not affect the driver in any way, hence giving the exact condition of the driver.

In this proposed project, we address a drowsy driver alert system that has been developed using a technique in which the Video Stream Processing (VSP) is analyzed by eye blink concept through an Eye Aspect Ratio (EAR) and Euclidean distance of the eye. When the driver's drowsiness is detected, the camera issues a beep sound as warning message, thereby alerting the driver with the help of facial detection by using Dlib's 68 facial landmark predictor.

Keywords - Driver drowsiness; eye detection; smart alert system; fatigue; Eye Aspect Ratio (EAR); Facial landmark

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF OUTPUTS

1. INTRODUCTION

1.1 PURPOSE AND OBJECTIVE

Humans have always invented machines and devised techniques to ease and protect their lives, for mundane activities like traveling to work, or for more interesting purposes like aircraft travel. With the advancement in technology, modes of transportation kept on advancing and our dependency on it started increasing exponentially. It has greatly affected our lives as we know it. Now, we can travel to places at a pace that even our grandparents wouldn’t have thought possible. In modern times, almost everyone in this world uses some sort of transportation every day. Some people are rich enough to have their own vehicles while others use public transportation. However, there are some rules and codes of conduct for those who drive irrespective of their social status. One of them is staying alert and active while driving. Neglecting our duties towards safer travel has enabled hundreds of thousands of tragedies to get associated with this wonderful invention every year. While on road, an automobile wields the most power and in irresponsible hands, it can be destructive and sometimes, that carelessness can harm lives even of the people on the road. One kind of carelessness is not admitting when we are too tired to drive. In order to monitor and prevent a destructive outcome from such negligence, ma1ny researchers have written research papers on driver drowsiness detection systems. But at times, some of the points and observations made by the system are not accurate enough. Hence, to provide data and another perspective on the problem at hand, in order to improve their implementations and to further optimize the solution, this project has been done.

Fig 1: Drowsy Driver

Fig 2: Webcam in front of Driver's Seat to detect drowsy driver

Our current statistics reveal that just in 2015 in India alone, 148,707 people died due to car related accidents. Of these, at least 21 percent were caused due to fatigue causing drivers to make mistakes. This can be a relatively smaller number still, as among the multiple causes that can lead to an accident, the involvement of fatigue as a cause is generally grossly underestimated. Fatigue combined with bad infrastructure in developing countries like India is a recipe for disaster. Fatigue, in general, is very difficult to measure or observe unlike alcohol and drugs, which have clear key indicators and tests that are available easily. Probably, the best solutions to this problem are awareness about fatigue-related accidents and promoting drivers to admit fatigue when needed. The former is hard and much more expensive to achieve, and the latter is not possible without the former as driving for long hours is very lucrative. When there is an increased need for a job, the wages associated with it increases leading to more and more people adopting it. Such is the case for driving transport vehicles at night. Money motivates drivers to make unwise decisions like driving all night even with fatigue. This is mainly because the drivers are not themselves aware of the huge risk associated with driving when fatigued. Some countries have imposed restrictions on the number of hours a driver can drive at a stretch, but it is still not enough to solve this problem as implementation is very tough and difficult to implement it.

1.2 SCOPE OF THE PROJECT

The scope of drowsiness detection encompasses various domains, including transportation safety (like in cars, trains, and planes), workplace safety (especially in industries where alertness is critical), and healthcare (such as monitoring patients for signs of fatigue or sleep disorders). Additionally, it can be applied in consumer devices like wearables to promote personal wellness and safety.

In this project, the author will focus on these following procedures:

- Basic concept of drowsiness detection system
- Familiarize with the signs of drowsiness
- Determine the drowsiness from these parameters - Eye blink - Area of the pupils detected at eyes - Yawning
- Data collection and measurement.
- Integration of the methods chosen.
- Coding development and testing.
- Complete testing and improvement.

2. SYSTEM ANALYSIS

2.1 HARDWARE REQUIREMENTS

Laptop with basic hardware:

- Intel Core – I3 Processor
- 4GB RAM (MIN)
- 500GB HDD (MIN)

Web Cam:

- 5mp (MIN)

SOFTWARE REQUIREMENTS

Python:

- Python 3

Libraries:

- Open CV
- Numpy
- Scipy
- Imutils
- Dlib
- Pygame etc

Operating System

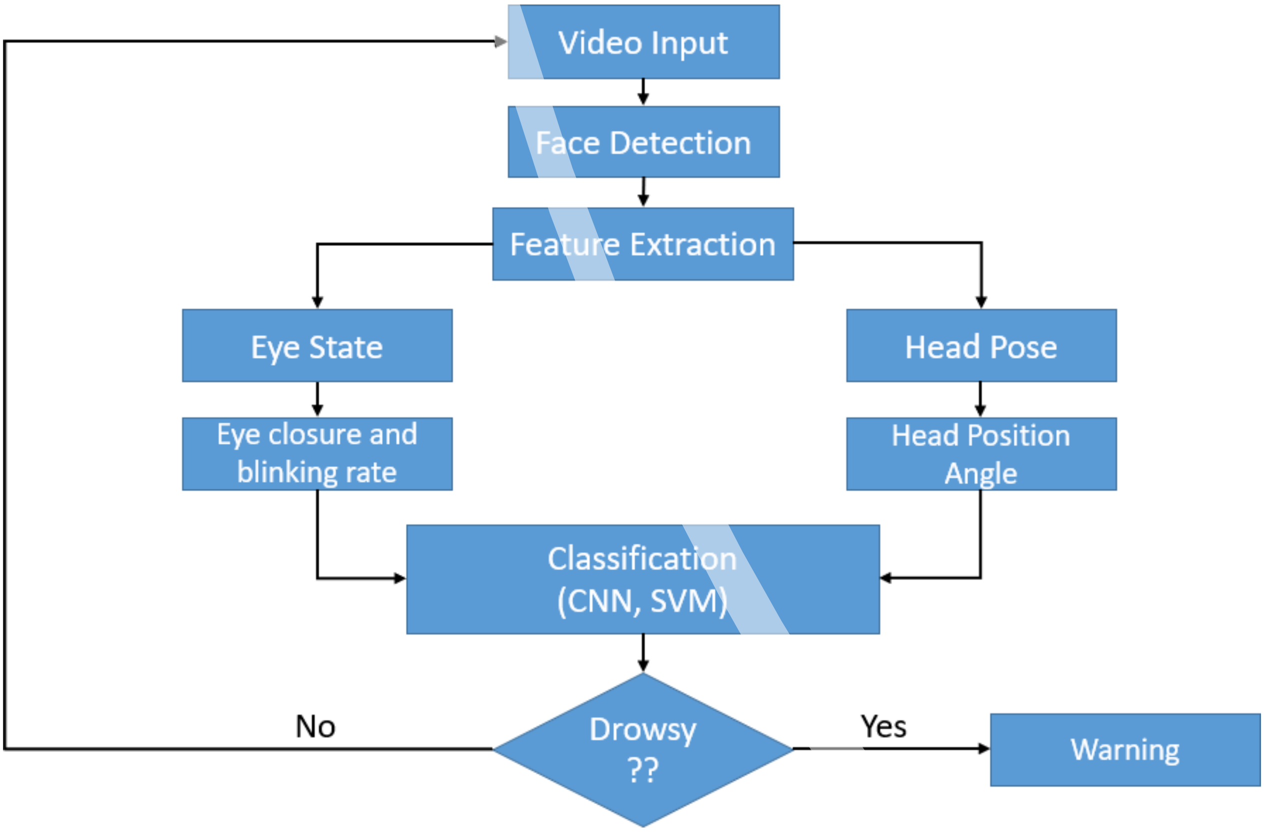
- Windows or MAC or Ubuntu

MODULES USED IN THE PROJECT

- Module – 1: Drowsiness_Dectection
- Module – 2: shape_predictor_68_face_landmarks
- Module – 3: alarm1
- Module – 4: alarm2

3. SYSTEM DESIGN & UML DIAGRAM

3.1 SYSTEM ARCHITECTURE



DATAFLOW DIAGRAM

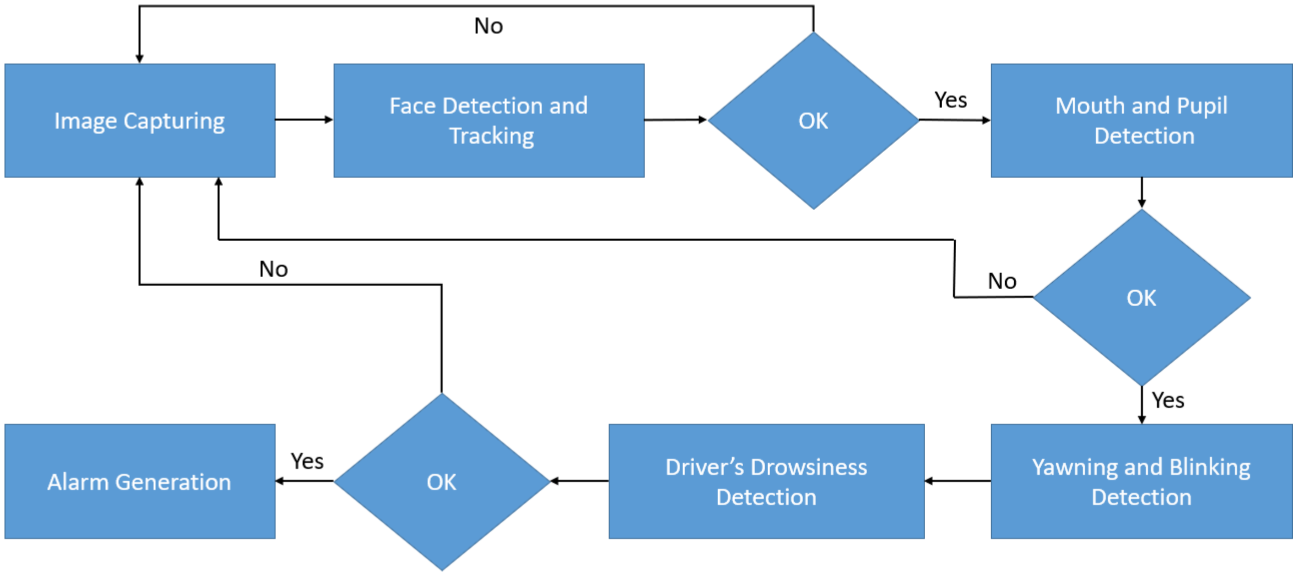


Fig 4: Dataflow Diagram

3.2 UML DIAGRAMS
CLASS DIAGRAM

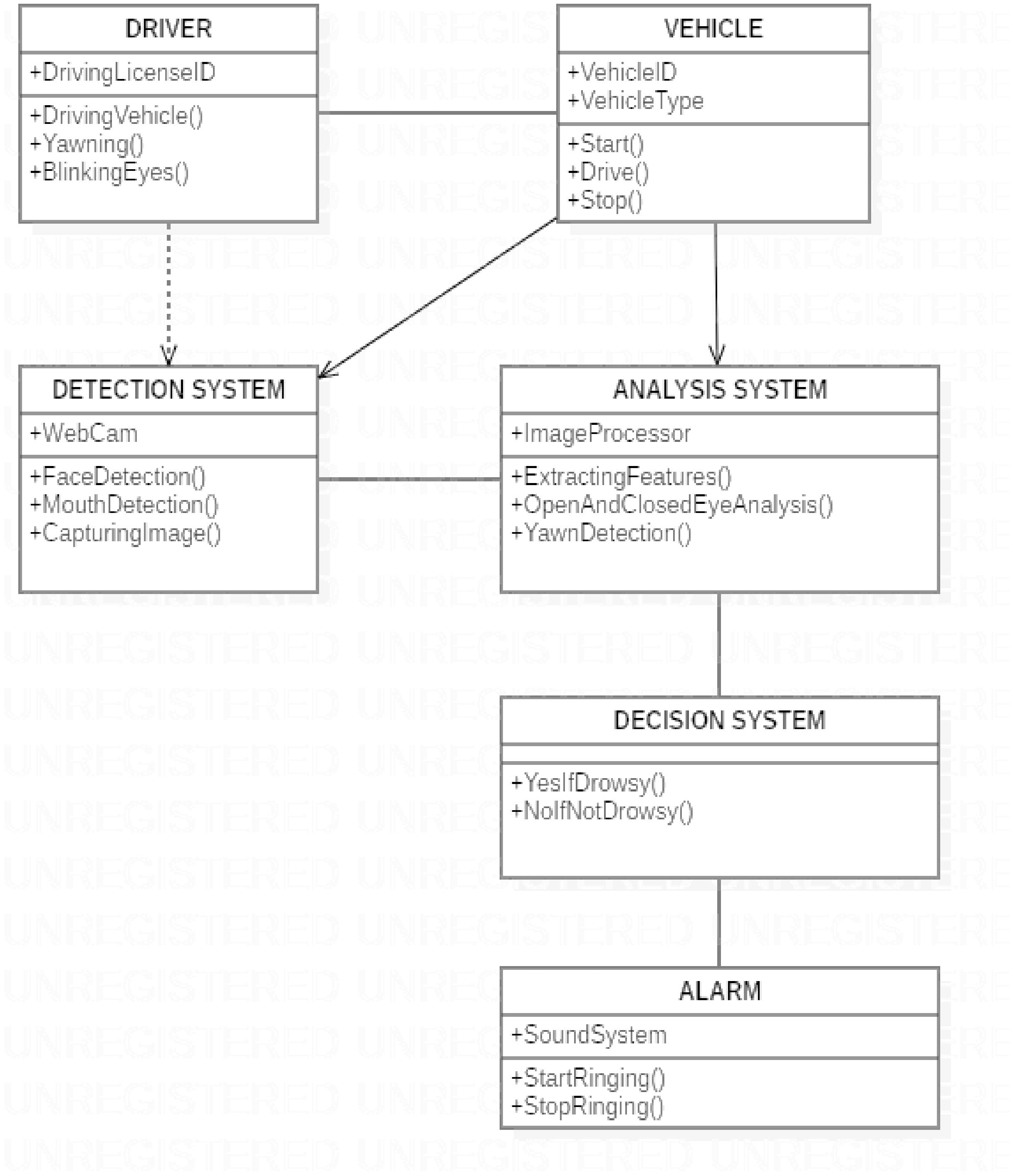


Fig 5: Class Diagram

USE CASE DIAGRAM

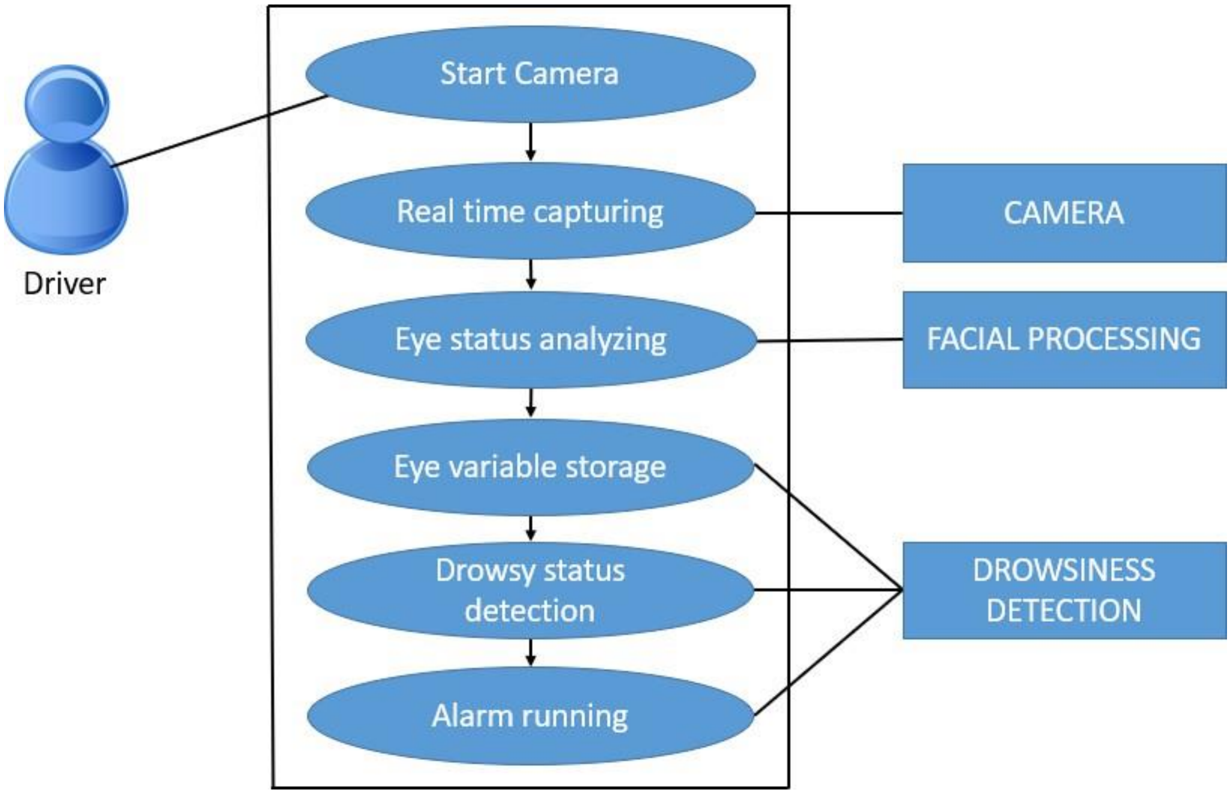


Fig 6: Use case Diagram

SEQUENTIAL DIAGRAM

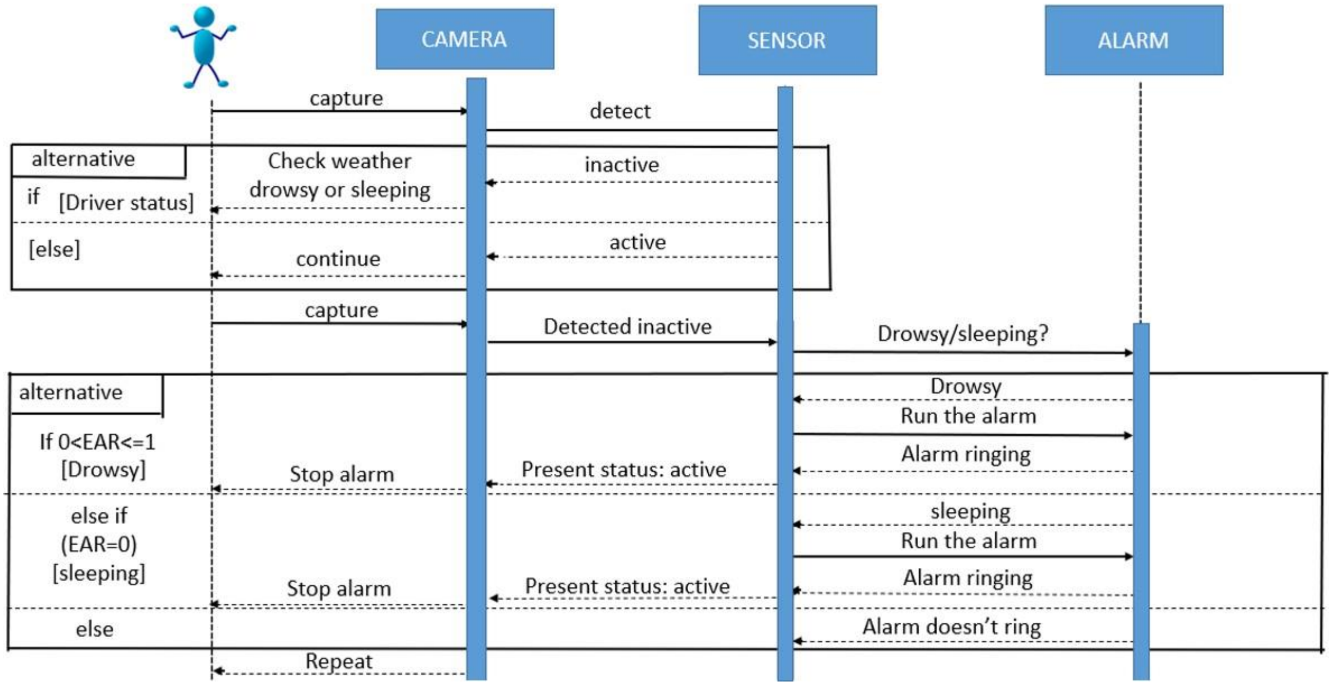


Fig 7: Sequential Diagram

ACTIVITY DIAGRAM

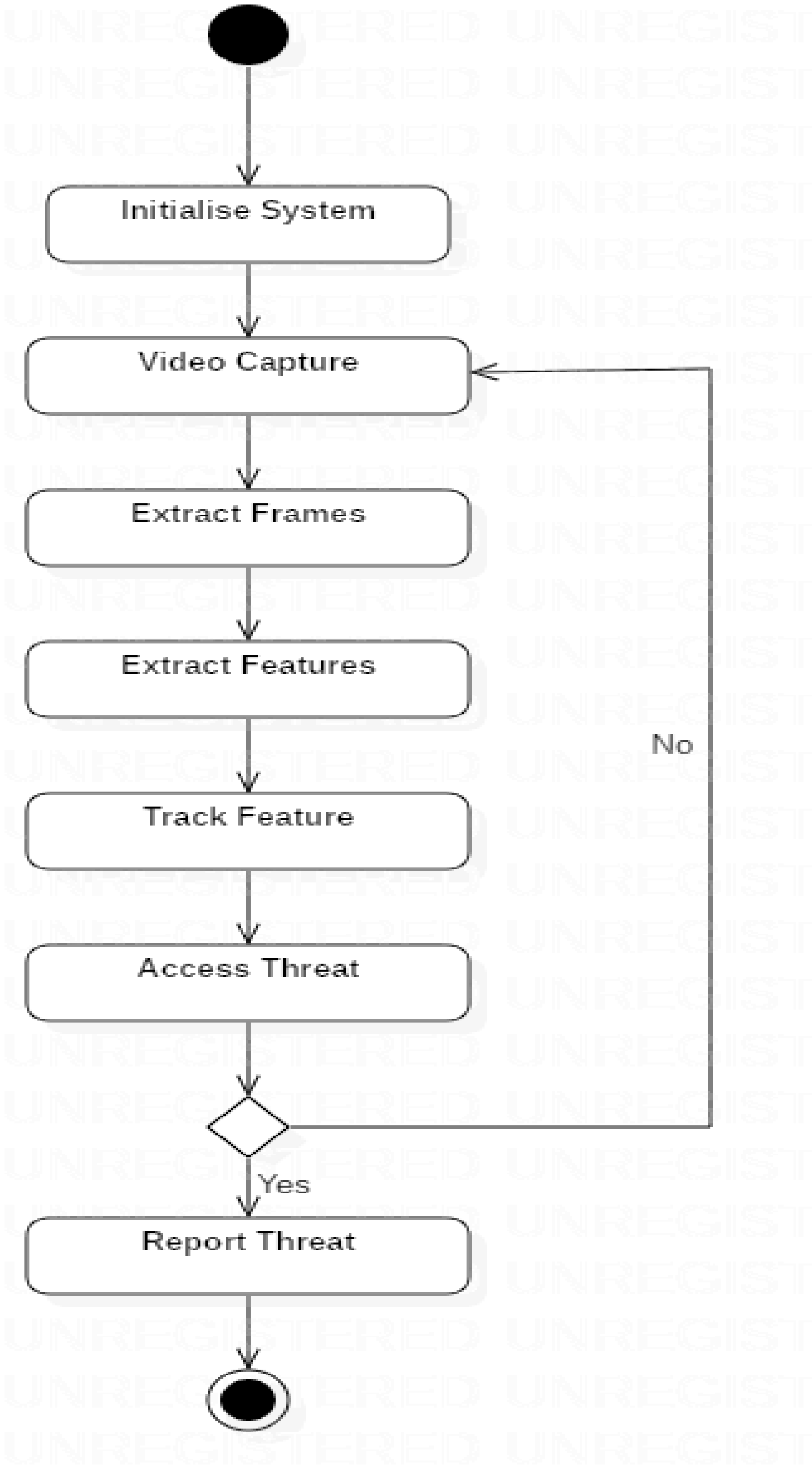


Fig 8: Activity Diagram

4. METHODOLOGY

In this project, we use a camera to obtain the real time video of the driver and convert that to video frames, these video frames are analyzed to detect faces and check if the driver is in a drowsy condition basing on the facial features. If the driver is drowsy he will be alerted by the system. This method is an efficient way to detect the drowsiness compared to existing methods as there are no external conditions are considered. The hardware requirement is also minimal.

Design Methodology

A web cam is used to capture the real time live video of the driver and then the video is disintegrated to video frames. The frames are then analyzed in Open CV through which faces are detected. Now the detected faces are again analyzed by the DLIB to detect the facial landmarks by using the 68 point approach. Eyes and Mouth are extracted as the main feature sets. The metrics of the facial landmarks are obtained and are used in EAR approach based formula to calculate the EAR of that face and compared with the following frames. If the EAR value is less than the threshold value for a certain period of time which is set, an alert sound is activated alerting the driver about the drowsiness.

DLIB

Identification of faces in photographs or videos is easy, but we need additional information about the person's face, such as the person's posture, if the mouth is shut or open, whether the eyes are shut or open, if the person is gazing up and so on. There are 68 points (landmarks) on the face that may be obtained using DLIB software.

From the 68 points, the coordinates of left eye, right eye and mouth are located and are further provided to the conditional logic for monitoring of EAR and MAR values.

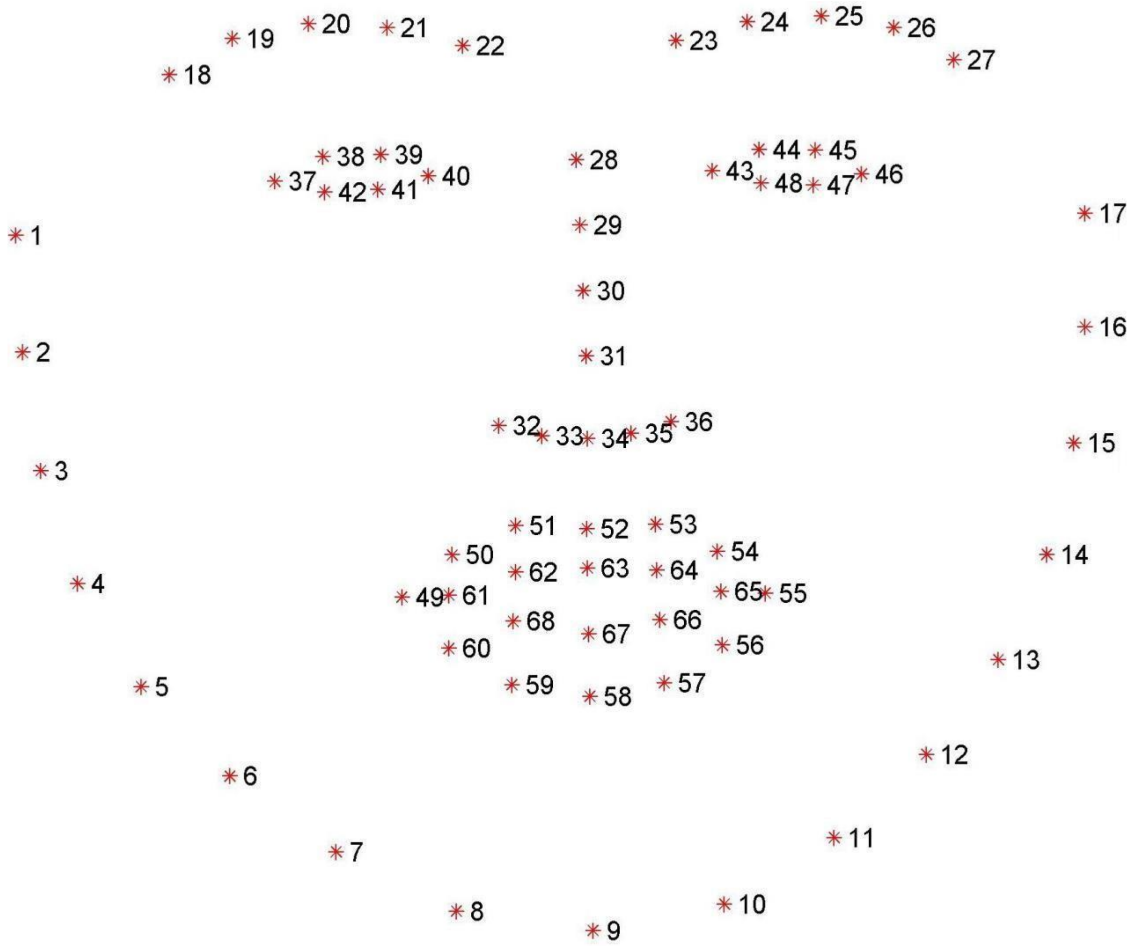


Fig 9: 68 Point View

EAR Algorithm

Part	Landmark Points
Left Eye	[37-42]
Right Eye	[43-48]

So, we calculate the EAR based on these landmark point values. The distance between locations was calculated using facial landmarks collected by the app. In order to calculate the EAR value, these distances were employed. Equation 1 was used to determine EAR, which is the height-to-width ratio of the eye As the numerator indicates the eye's height, and the denominator represents its breadth, the picture shows all the ocular landmarks in their entirety.

Referring equation 1, Calculates the distance between upper and lower eyelids using the numerator. The horizontal distance between the eyes. A higher numerator number indicates an increased EAR, whereas a lower numerator value indicates a lowered EAR. These numbers are used to detect driver fatigue in this situation.

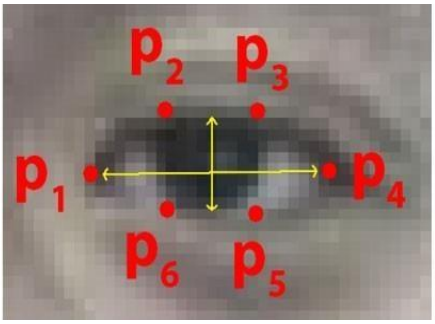


Fig 10: Eye Point View.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Fig 11: Equation 1

5. IMPLEMENTATION

5.1 SAMPLE CODE

jupyter Driver_Drowsiness_Detection_System Last Checkpoint: 11/07/2022 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [1]:

```python
# Importing OpenCV Library for basic image processing functions
import cv2
# Numpy for array related functions
import numpy as np
# Dlib for deep learning based Modules and face landmark detection
import dlib
# face_utils for basic operations of conversion
from imutils import face_utils
# mixer is to play the sound
from pygame import mixer

mixer.init()

#Initializing the camera and taking the instance
cap = cv2.VideoCapture(0)

#Initializing the face detector and landmark detector
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

sound1 = mixer.Sound('alarm.wav')
sound2 = mixer.Sound('alarm2.wav')

#status marking for current state
sleep = 0
drowsy = 0
active = 0
```

jupyter Driver_Drowsiness_Detection_System Last Checkpoint: 11/07/2022 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
#status marking for current state
sleep = 0
drowsy = 0
active = 0
status=""
color=(0,0,0)

def compute(ptA,ptB):
    dist = np.linalg.norm(ptA - ptB)
    return dist

def blinked(a,b,c,d,e,f):
    up = compute(b,d) + compute(c,e)
    down = compute(a,f)
    ratio = up/(2.0*down)

    #Checking if it is blinked
    if(ratio>0.25):
        return 2
    elif(ratio>0.21 and ratio<=0.25):
        return 1
    else:
        return 0


while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

jupyter Driver_Drowsiness_Detection_System Last Checkpoint: 11/07/2022 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code


```
while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```


```
    faces = detector(gray)
    #detected face in faces array
    for face in faces:
        x1 = face.left()
        y1 = face.top()
        x2 = face.right()
        y2 = face.bottom()

        face_frame = frame.copy()
        cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

        landmarks = predictor(gray, face)
        landmarks = face_utils.shape_to_np(landmarks)

        #The numbers are actually the landmarks which will show eye
        left_blink = blinked(landmarks[36],landmarks[37],
            landmarks[38], landmarks[41], landmarks[40], landmarks[39])
        right_blink = blinked(landmarks[42],landmarks[43],
            landmarks[44], landmarks[47], landmarks[46], landmarks[45])

        #Now judge what to do for the eye blinks
        if(left_blink==0 or right_blink==0):
            sleep+=1
            drowsy=0
            active=0
```

jupyter Driver_Drowsiness_Detection_System Last Checkpoint: 11/07/2022 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
        #Now judge what to do for the eye blinks
        if(left_blink==0 or right_blink==0):
            sleep+=1
            drowsy=0
            active=0
            if(sleep>6):
                status="SLEEPING !!!"
                color = (255,0,0)
                sound1.play()

        elif(left_blink==1 or right_blink==1):
            sleep=0
            active=0
            drowsy+=1
            if(drowsy>6):
                status="Drowsy !"
                color = (0,0,255)
                sound2.play()

        else:
            drowsy=0
            sleep=0
            active+=1
            if(active>6):
                status="Active :)"
                color = (0,255,0)

        cv2.putText(frame, status, (100,100), cv2.FONT_HERSHEY_SIMPLEX, 1.2, color,3)
```

jupyter Driver_Drowsiness_Detection_System Last Checkpoint: 11/07/2022 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
            active+=1
            if(active>6):
                status="Active :)"
                color = (0,255,0)

        cv2.putText(frame, status, (100,100), cv2.FONT_HERSHEY_SIMPLEX, 1.2, color,3)

        for n in range(0, 68):
            (x,y) = landmarks[n]
            cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)

    cv2.imshow("Frame", frame)
    cv2.imshow("Result of detector", face_frame)
    key = cv2.waitKey(1)
    if key == 27:
          break
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.13)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

In [ ]:

5.2 OUTPUT SCREENS

The tests were conducted in various conditions including:

- Different lighting conditions.
- Driver’s posture and position of the drivers face.
- Drivers with spectacles.
- Driver’s head is tilted.

Test case 1: When there is ambient light

Fig 12: Driver with Ambient Lighting

Result: As shown in Figure 12, when there is ambient amount of light, the driver's face and eyes are successfully detected.

Test case 2: Position of the driver's face

- **Center Positioned**

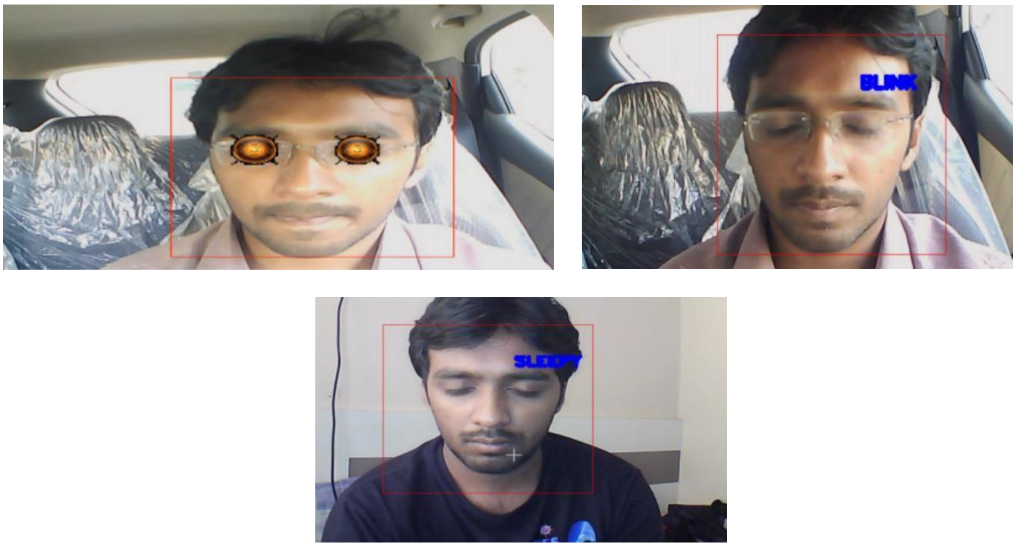


Fig 13: Driver's face in center of the frame

Result: As shown in Figure 13, When the driver's face is positioned at the Centre, the face, eyes, eye blinks, and drowsiness was successfully detected.

- **Right Positioned**

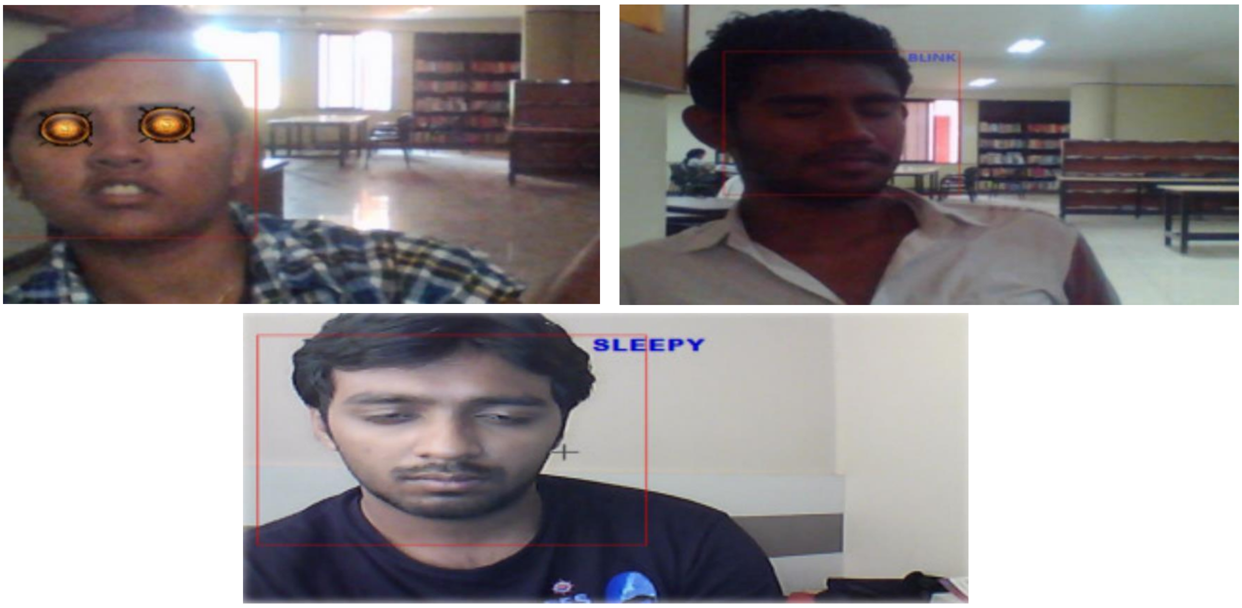


Fig 14: Driver's face to right of frame

Result: As shown in Figure 14, When the driver's face is positioned at the Right, the face, eyes, eye blinks, and drowsiness was successfully detected.

- **Left Positioned**

Fig 15: Driver's face to left of frame

Result: As shown in screen snapshot in Figure 15, when the driver's face is positioned at the Left, the face, eyes, eye blinks, and drowsiness was successfully detected.

Test case 3: When the driver is wearing spectacles

Fig 16: Driver with spectacles

Result: As shown in screen snapshot in Figure 16, When the driver is wearing spectacles, the face, eyes, eye blinks, and drowsiness was successfully detected.

Test case 4: When the driver's head is tilted

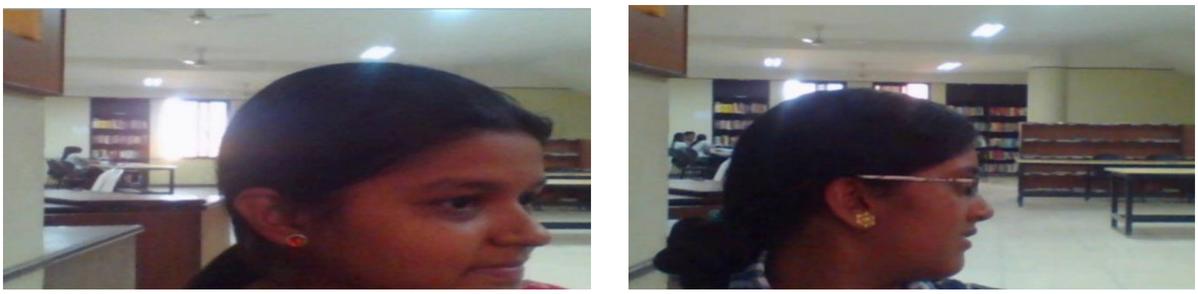


Fig 17: Various posture

Result: As shown in screen snapshot in Figure 17, when the driver's face is tilted for more than 30 degrees from vertical plane, it was observed that the detection of face and eyes failed.

OUTPUT

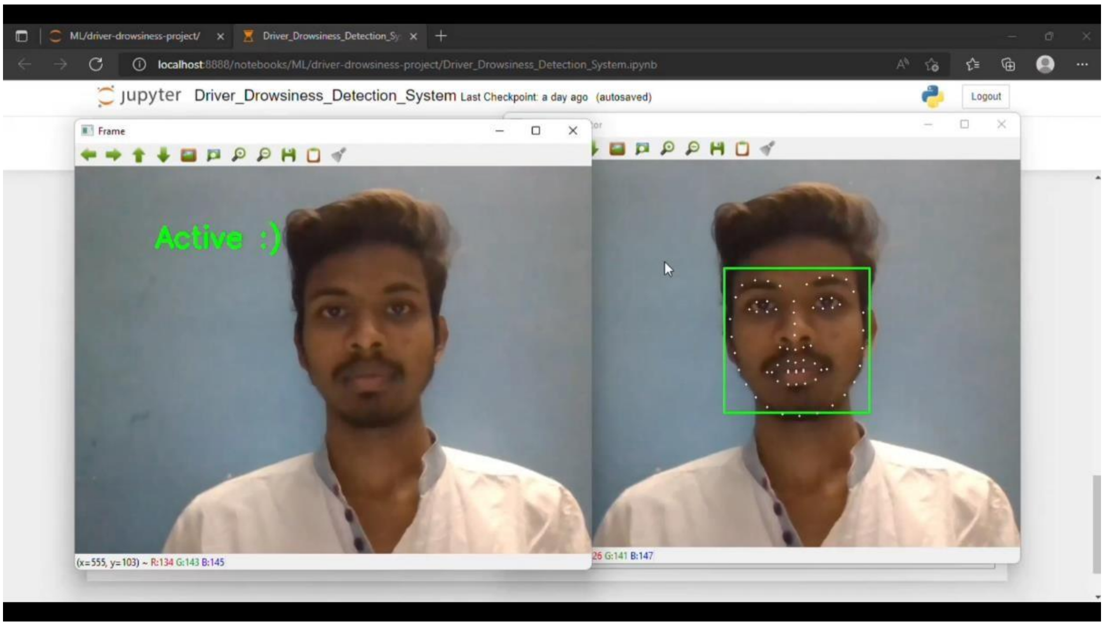


Fig 18: Person in Active state

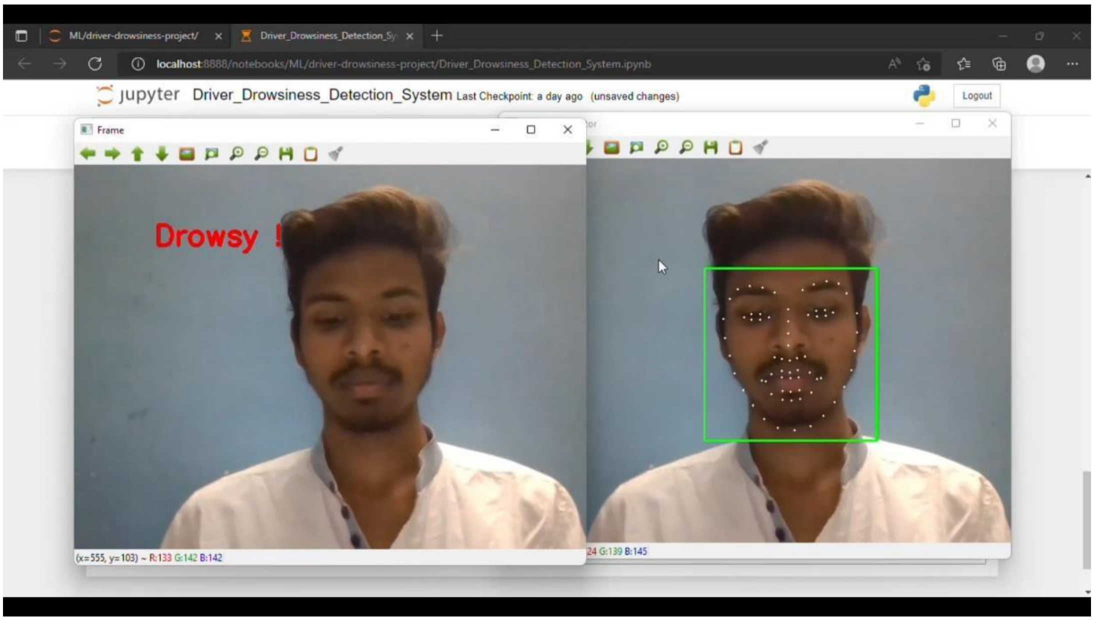


Fig 19: Person in Drowsy State

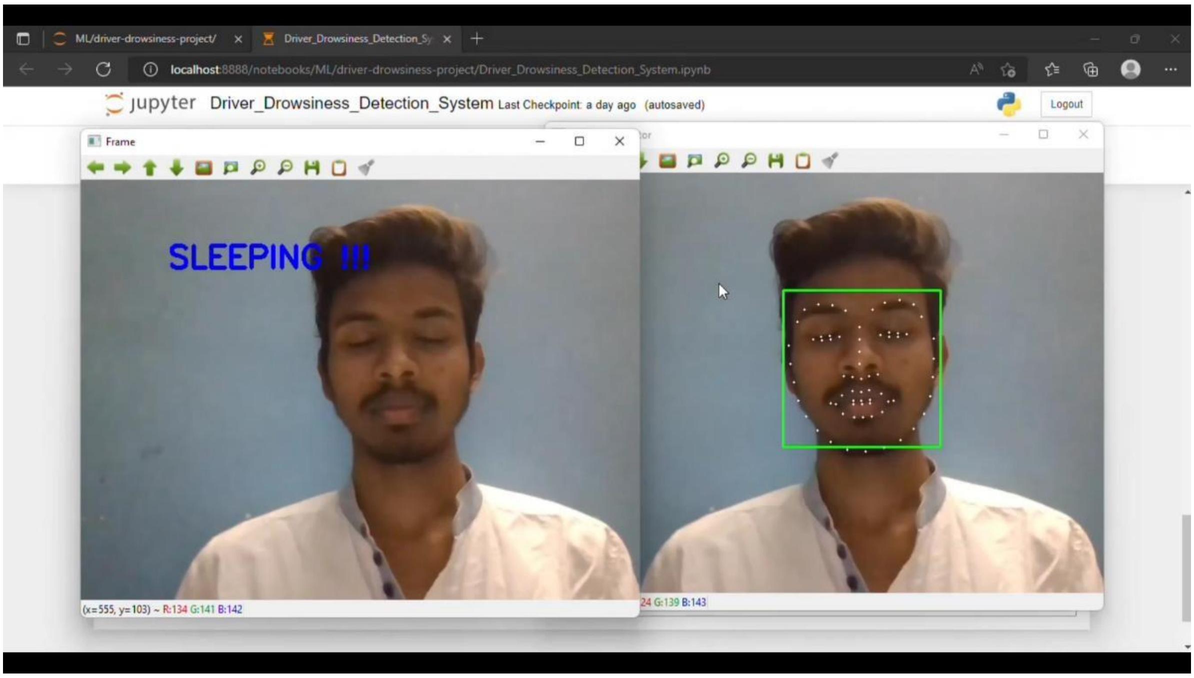


Fig 20: Person in Sleeping State

6. CONCLUSION & FUTURE SCOPE

CONCLUSION

- Driver drowsiness detection system completely meets the objectives and requirements of the system
- Driver drowsiness detection is designed mainly to keep the driver awake while driving to avoid the accident due to sleepiness. The alert signal is generated from embedded device to awake driver from sleepy state
- The system works well even in case of drivers wearing spectacles and also under low light conditions.
- Fatigue is measured by detecting Eye and face using Haar Cascade Classifier, especially facial landmarks is detected using shape-predictor and Eye Aspect Ratio (EAR) by calculating the Euclidean distance between the eyes.
- A system for driver safety and car security is presented only in the luxurious costly cars. Using drowsiness detection system, driver safety can be implemented in normal cars also.
- By developing this system, many accidents will be reduced and provides safe life to the driver and vehicle safety.

FUTURE SCOPE

- By developing this, we can implement a smart alert system in future which can be used in building intelligent vehicles that can automatically alert the drowsy driver.
- The future works may focus on the utilization of outer factors such as
  - Sleeping Hours
  - Weather Conditions
  - Vehicle States
  - Mechanical data

  for fatigue measurement.
- Currently there is no adjustment like zoom or direction of the camera during operation. Future work may be to automatically zoom in on the eyes once they are localized.
- The model can be improved incrementally by using other parameters like blink rate, yawning, state of the car, etc. If all these parameters are used it can improve the accuracy by a lot.
- We plan to further work on the project by adding a sensor to track the heart rate in order to prevent accidents caused due to sudden heart attacks to drivers.

7. BIBILOGRAPHY

1. COMPUTATIONALLY EFFICIENT FACE DETECTION; B. SCHLKOPF-A. BLAKE, S. ROMDHANI, AND P. TORR.
2. USE OF THE HOUGH TRANSFORMATION TO DETECT LINES ANDCURVES IN PICTURE; R. DUDA AND P. E. HART.
3. JAIN, “FACE DETECTION IN COLOR IMAGES; R. L. HSU, M. ABDEL-MOTTALEB, AND A. K. JAIN.
4. OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION;
   P.R. TABRIZI AND R. A. ZOROOFI.
5. http://ncrb.gov.in/StatPublications/ADSI/ADSI2015/chapter1A%20tr affic%20accidents.pdf
6. https://pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/
7. http://dlib.net/face_landmark_detection_ex.cpp.html
8. https://techvidvan.com/tutorials/driver-drowsiness-detection-system/?amp=1
9. https://www.analyticsvidhya.com/blog/2022/04/face-detection-using- the-dlib-face-detector-model/#:~:text=Dlib%20HoG%20Face%20Detection&text=HOG%20is %20basically%20a%20feature,Right%20side%20turned%20face