Drowsiness Sight Detection Machine Learning Project

Project Report

Submitted by

Gungun Sadhwani []
Azhar Ali [023-21-0257]

Under guidance of

Dr. Sher Muhammad Daudpoto

Sukkur
IBA
University

ABSTRACT

Drowsiness remains a significant factor in car accidents, presenting a serious hazard on the road. Taking necessary precautions like getting ample sleep before driving, consuming beverages like coffee or energy drinks, and recognizing signs of fatigue by taking breaks can help reduce these risks. Recent statistics underscore the prevalence of drowsiness as a leading cause of accidents, contributing to more than 30% of such incidents and resulting in a tragic loss of lives annually. Addressing this issue requires technology adept at identifying driver fatigue. Our approach involves leveraging sophisticated image processing techniques that specifically target facial features and eye movements. By analyzing these cues, our algorithm predicts instances of potential drowsiness. When rapid eye blinking, a common sign of fatigue, is detected, the system alerts the driver with an auditory signal. This technology aims to prevent accidents and prioritize the safety of both the driver and fellow commuters on the road.

Contents

1. Introduction:

Feeling tired can be like being on the edge of sleep, where all you want is to rest. It's especially risky when you're doing something that needs your attention, like driving car. When someone is really tired while driving, it can lead feeling extremely worn out, making accidents more likely.

Existing systems that try to detect tiredness in drivers use complex calculations and high-tech gadgets. But they're not very practical for driving because they can be uncomfortable, like the Electroencephalography (EEG) and Electrocardiography (ECG) systems. Making technology that can spot tiredness while driving is a big challenge but very important for preventing accidents caused by tiredness.

Because tiredness is such a big problem on the roads, we need better ways to deal with it. One way is to use a system that watches the driver using a regular camera. We've come up with a system that can spot signs of tiredness by looking at the driver's face and eyes in videos. We use a special computer algorithm called a CNN to do this.

When it notices signs of tiredness, it alerts the driver so they know they might be feeling too sleepy to drive safely.

2. System Design

2.1 Algorithm Used:

A. Eye Aspect Ratio (EAR):

This system looks at the driver's eyes and measures how often they blink and for how long. It uses a specific formula to do this, kind of like a math problem.

B. Video Watching:

Imagine the system is watching a video of the driver's face. It's not just any video—it's a live video, like a video call.

C. Face and Eye Watching:

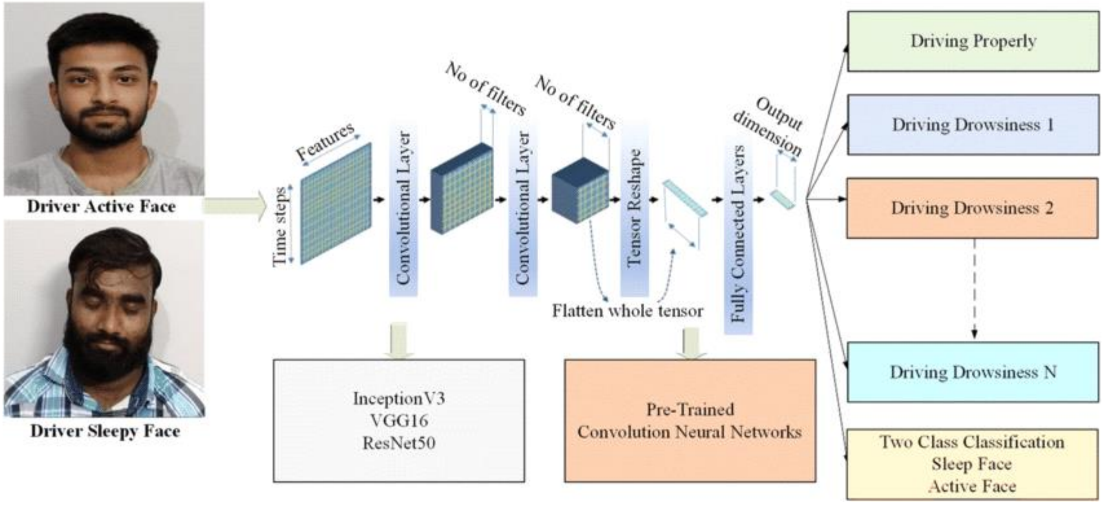
It's really good at recognizing different parts of the driver's face, especially the eyes. It keeps an eye on how the driver's eyes move and blink.

D. Detecting Signs of Tiredness:

By paying attention to how often and how long the driver blinks, the system tries to guess if the driver might be feeling tired. When it thinks the driver might be too tired to drive safely, it gives a warning.

E. Alerting the Driver:

This warning isn't just a message on a screen—it's also a sound, like a beep or a sound you'd hear on your phone. It's meant to get the driver's attention and let them know they might be feeling too sleepy to drive well.



2.2 Dataset

Captured the raw dataset source from kaggle. Data preprocessing is performed so as to make the dataset compatible to the program. Two categories of data which helps in determining the difference which is open and close. Equal number of images in the data so as to avoid uneven biasness.

2.3 Technique Used

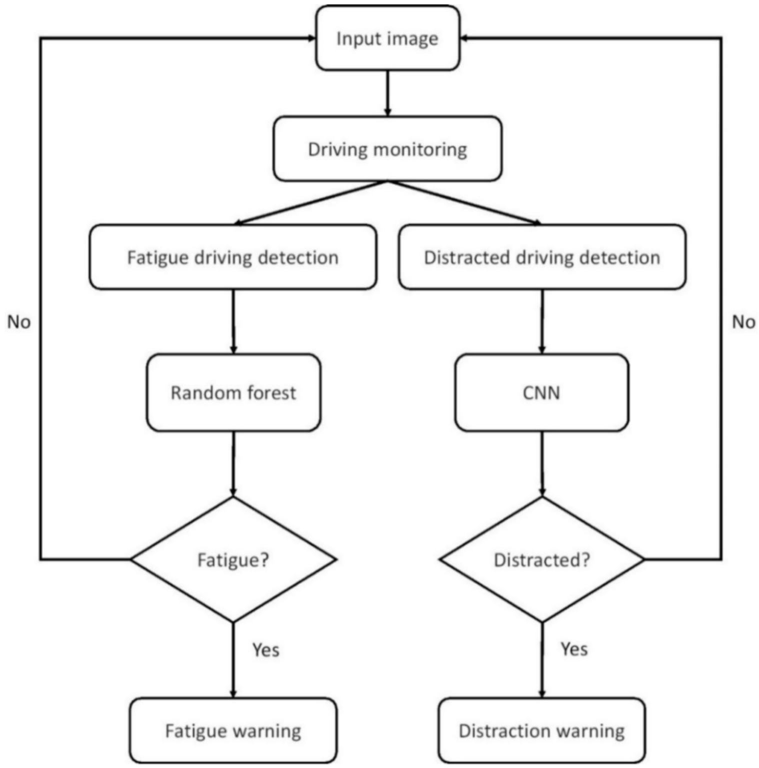
- **Eye Aspect Ratio (EAR):** EAR is a mathematical formula used to calculate the ratio of different measurements taken from the driver's eyes. Specifically, it measures the ratio between the width and height of the eye and the distance between certain points on the eye to detect patterns associated with blinking and eye closure.
- **Facial Landmark Detection:** It identifies and tracks specific points on the driver's face, particularly the corners and edges of the eyes. This helps in accurately calculating the EAR.
- **Real-time Analysis:** The system continuously monitors the changes in the driver's eye shape and structure captured in live video frames. By computing the EAR for each frame, it assesses patterns related to eye blinks and closures.
- **Threshold-based Alerting:** If the calculated EAR falls below a predefined threshold value for a specific duration (checked over multiple frames), indicating prolonged or frequent eye closures, the system triggers an alert to notify the driver of potential drowsiness.

2.4 DATA PROCESSING

- **Capturing Video:** The system uses a camera to record live video of the driver's face while they're driving.
- **Face Recognition:** It looks at each frame of the video and identifies where the driver's face is. It pays special attention to the eyes.
- **Calculating Blinking:** By tracking the movement and shape of the driver's eyes in each frame, the system figures out how often the driver blinks.

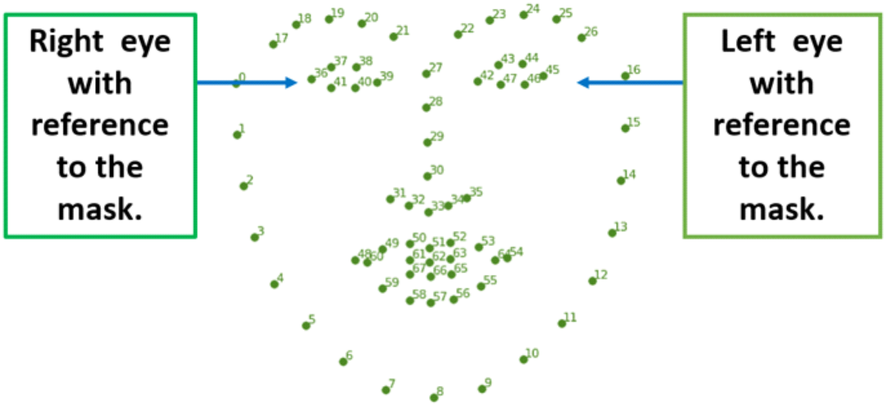
- **Determining Drowsiness:** Based on how frequently and for how long the driver's eyes are closed (like when they blink for longer periods), the system makes a guess about whether the driver might be getting too tired to drive safely.
- **Alerting the Driver:** When it thinks the driver might be too tired, the system gives a warning. It might be something visual on the screen, like a message, and also something auditory, like a sound or beep, to get the driver's attention.

2.5 ARCHITECTURE



2.6 TECHNOLOGY USED

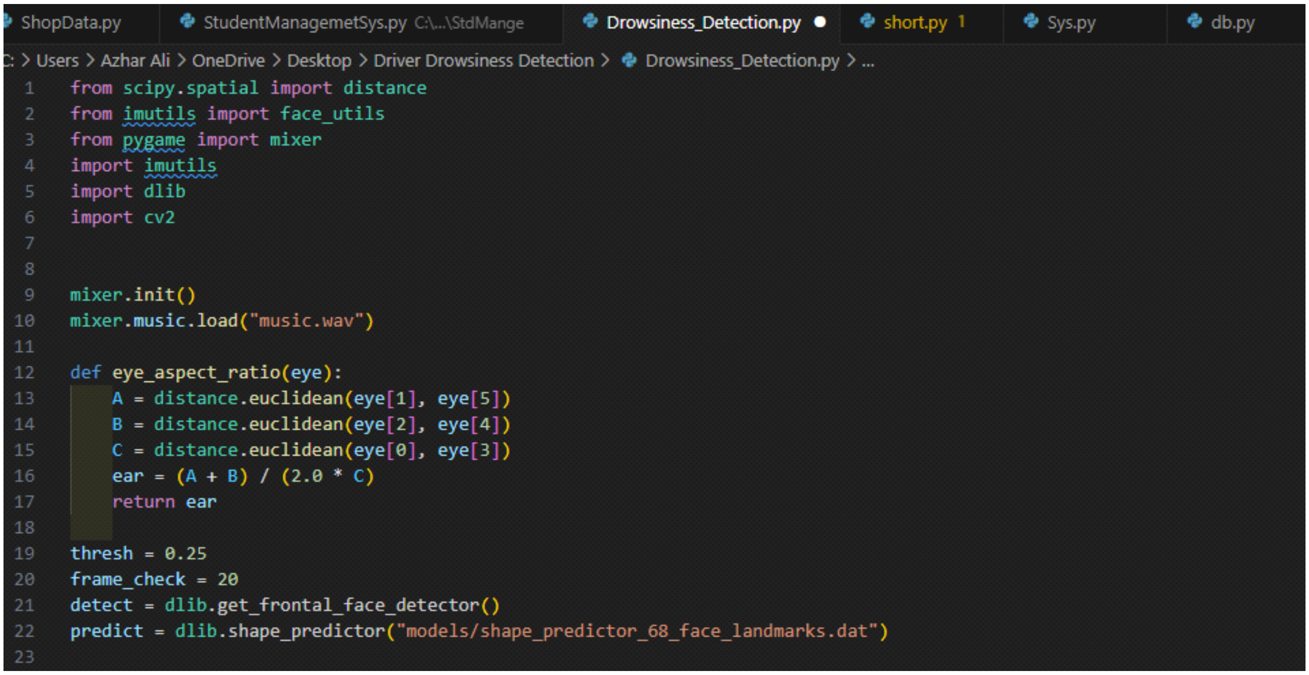
- **OpenCV:** This is a computer vision library used for image and video processing. It helps in capturing live video frames and performing operations on them, like converting them to grayscale or detecting facial landmarks.
- **dlib:** It's a toolkit used for machine learning and computer vision tasks. In this project, it's used for facial detection and landmark prediction, allowing the system to identify specific points on the driver's face, especially around the eyes.
- **Python:** The programming language used to build and execute the drowsiness detection system. Python provides a flexible and powerful environment for implementing various algorithms and integrating different libraries.
- **pygame:** This library is used to play sound alerts when the system detects signs of drowsiness in the driver. It helps in creating an auditory warning to grab the driver's attention.
- **imutils:** A library used for convenience functions in OpenCV. It simplifies various tasks like resizing images or videos, making the code cleaner and more manageable.


Right eye with reference to the mask.

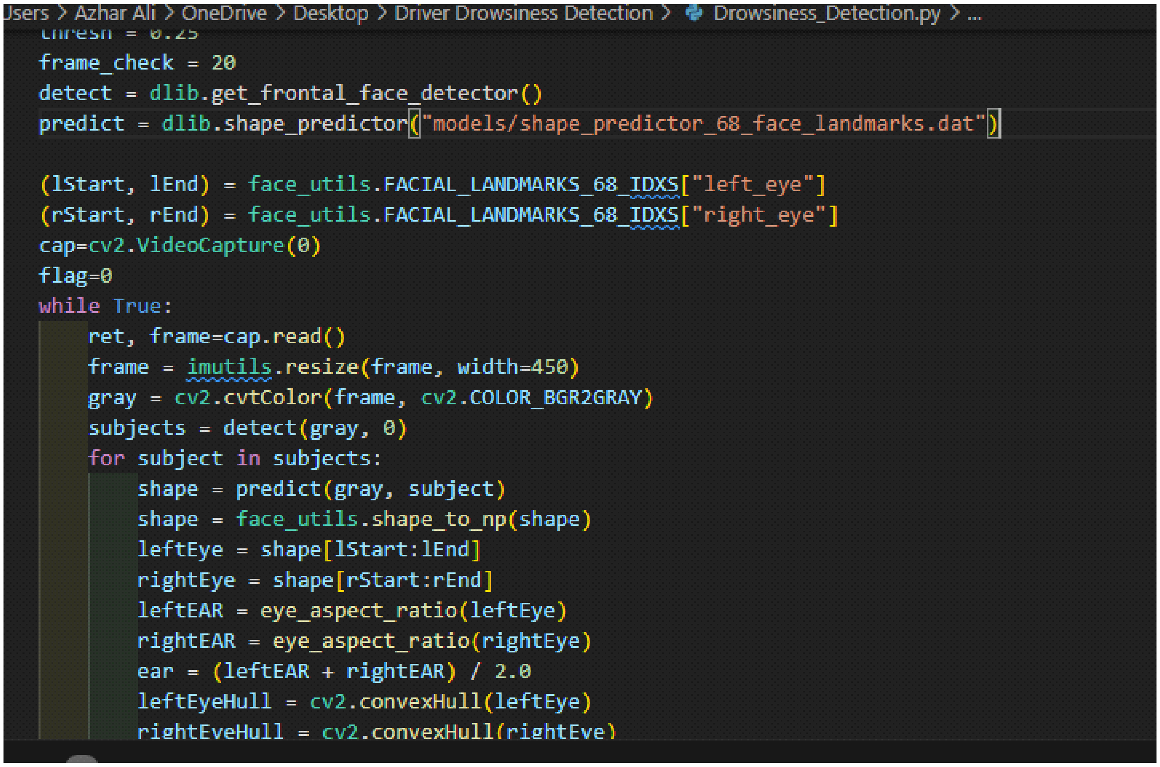
Left eye with reference to the mask.

Masking and landmarks of all the critical points in a face

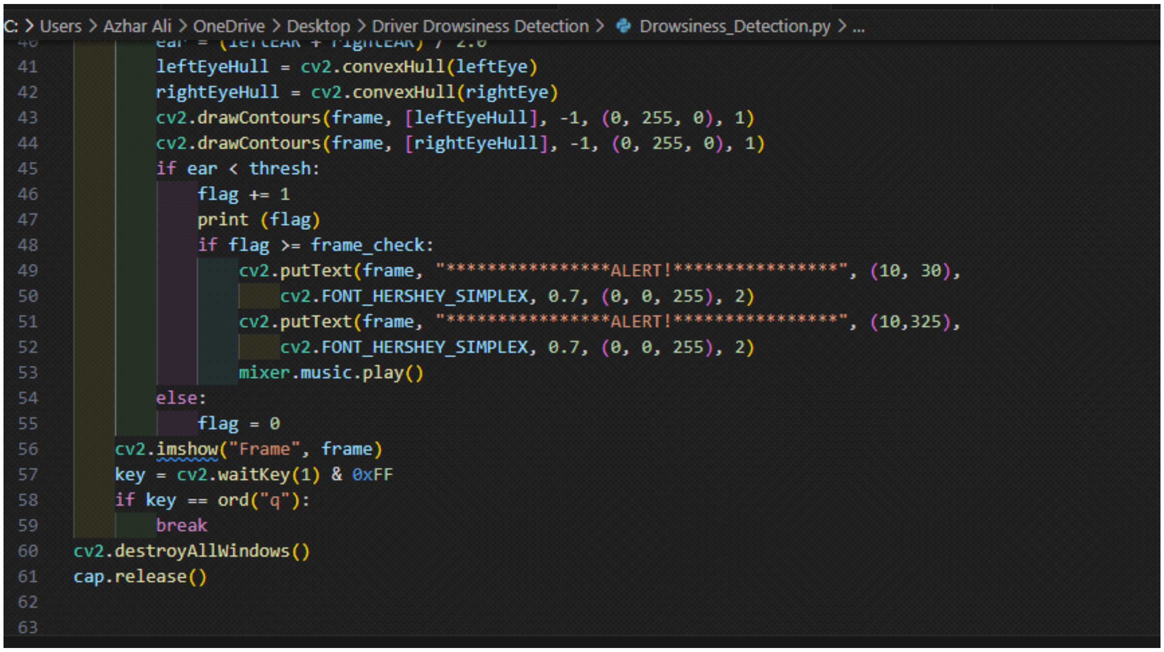
4. CODING TESTING


ShopData.py
StudentManagemetSys.py C:\...\StdMange
Drowsiness_Detection.py
short.py 1
Sys.py
db.py
C: > Users > Azhar Ali > OneDrive > Desktop > Driver Drowsiness Detection > Drowsiness_Detection.py > ...

```python
from scipy.spatial import distance
from imutils import face_utils
from pygame import mixer
import imutils
import dlib
import cv2


mixer.init()
mixer.music.load("music.wav")

def eye_aspect_ratio(eye):
    A = distance.euclidean(eye[1], eye[5])
    B = distance.euclidean(eye[2], eye[4])
    C = distance.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return ear

thresh = 0.25
frame_check = 20
detect = dlib.get_frontal_face_detector()
predict = dlib.shape_predictor("models/shape_predictor_68_face_landmarks.dat")

```


Users > Azhar Ali > OneDrive > Desktop > Driver Drowsiness Detection > Drowsiness_Detection.py > ...

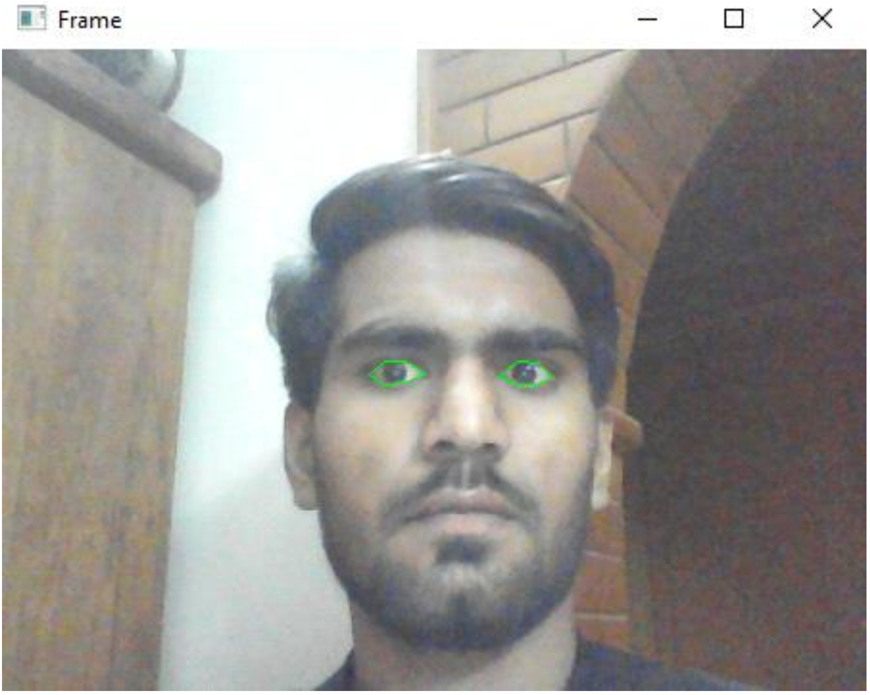
```python
thresh = 0.25
frame_check = 20
detect = dlib.get_frontal_face_detector()
predict = dlib.shape_predictor("models/shape_predictor_68_face_landmarks.dat")

(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["right_eye"]
cap=cv2.VideoCapture(0)
flag=0
while True:
    ret, frame=cap.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    subjects = detect(gray, 0)
    for subject in subjects:
        shape = predict(gray, subject)
        shape = face_utils.shape_to_np(shape)
        leftEye = shape[lStart:lEnd]
        rightEye = shape[rStart:rEnd]
        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)
        ear = (leftEAR + rightEAR) / 2.0
        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
```



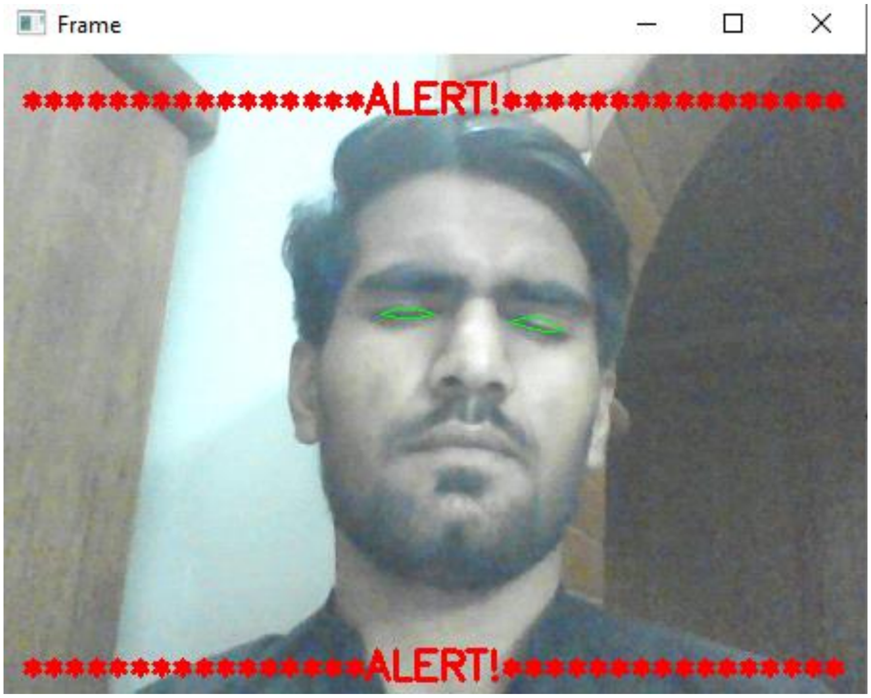
C: > Users > Azhar Ali > OneDrive > Desktop > Driver Drowsiness Detection > Drowsiness_Detection.py > ...

```
            ear = (leftEAR + rightEAR) / 2.0
            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)
            if ear < thresh:
                flag += 1
                print (flag)
                if flag >= frame_check:
                    cv2.putText(frame, "****************ALERT!****************", (10, 30),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
                    cv2.putText(frame, "****************ALERT!****************", (10,325),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
                    mixer.music.play()
            else:
                flag = 0
        cv2.imshow("Frame", frame)
        key = cv2.waitKey(1) & 0xFF
        if key == ord("q"):
            break
cv2.destroyAllWindows()
cap.release()
```

OUTPUT



J


Frame
****************ALERT!****************
****************ALERT!****************

5. Conclusion:

This project created a smart system that watches a driver's face while they're driving. It pays special attention to their eyes to see if they're blinking too much, which could mean they're getting too tired to drive safely.

When it notices that the driver might be too tired, it shows a message and plays a sound to alert them. This is to help prevent accidents caused by driving when someone is too sleepy.

While this system is a good start, there's still work to do to make it even better and more reliable. By improving how it works and making it smarter, it could become a really useful tool to keep drivers safe on the road.

6. FUTURE ENHANCEMETN

The future works may focus on the utilization of outer factors such as vehicle states, sleeping hours, weather conditions, mechanical data, etc, for fatigue measurement. Driver drowsiness pose a major threat to highway safety, and the problem is particularly severe for commercial motor vehicle operators. Twenty-four hour operations, high annual mileage, exposure to challenging environmental conditions, and demanding work schedules all contribute to this serious safety .

7. Reference

Dr. Sher Muhammad Daudpoto

Dr. Saif Hassan

www.researchgate.net