MAJOR PROJECT-1

SYNOPSIS REPORT

For

Drowsiness Detection System

Submitted By

Chetanshi Pandey	**Kashish Kasana**	**Addya Pandey**	**Preeti Kumari**
R2142211120	R2142210406	R2142210885	500091864

UPES

UNIVERSITY OF TOMORROW

Cloud Software Operations Cluster
School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

Under the guidance of-
Dr. Khushboo Jain
Assistant Professor- SG
Department of Data Science

Table of Contents

Topic **Page Number**

Table of Content
1. Introduction 1-22
1.1Purpose of the Project 1
1.2 Target Beneficiary 1
1.3 Project Scope 1-2
1.4 References

2 Project Description 2-5
2.1 Data/ Data structure 2-3
2.2 SWOT Analysis 3-4
2.3 Project Features 4
2.4 Design and Implementation Constraints 5
2.5 Design diagrams 5

3 System Requirements 6
3.1 User Interface 6
3.2 ML Models Used 6

4 Non-functional Requirements 7
4.1 Performance requirements 7
4.2 Security requirements 7
4.3Software Quality Attributes 7

5 Code Implemented 8-9
5.1 Backend 8
5.2 Frontend 9

6 Millstone completed 10

1. Introduction

Many road accidents which lead to death are because of drowsiness while driving. Drivers who drive long hours like truck drivers, bus drivers are likely to experience this problem. It is highly risky to drive with lack of sleep and driving for long hours will be more tiresome[1]. Due to the drowsiness of the driver causes very dangerous consequences, it is estimated that 70,000 to 80,000 injures & crashes happen worldwide in a year. Even deaths have reached 1000-2000 every year. There are many unofficial deaths which are not confirmed by drivers that it was due to their drowsiness. This takes lives of many innocent people. It is a nightmare for a lot of people who travel across world. It is very important to identify the driver drowsiness and alert the driver to prevent crash.

The goal of this research is the detection of the indication of this fatigue of the driver. The acquisition system, processing system and warning system are the three blocks that are present in the detection system. The video of the driver's front face is captured by the acquisition system, and it is transferred to the next stage[2] i.e., processing block. The detection is processed online and if drowsiness of driver is detected, then the warning system gives a warning or alarm. The methods to detect the drowsiness of the drive may be done using intrusive or nonintrusive method i.e., with and without the use of sensors connected to the driver. The cost of the system depends on the sensors used in the system. Addition of more parameters can increase the accuracy of the system to some extent. The motivation for the development of cost effective, and real-time driver drowsiness system with acceptable accuracy are the motivational factors of this work. Hence, the proposed system detects the fatigue of the driver from the facial images, [3]and image processing technology and machine learning method are used to achieve a cost effective and portable system.

1.1 Purpose of the Project

The Drowsiness Detection System enhances road safety by monitoring drivers' facial features through a webcam and detecting signs of fatigue in real time. It uses computer vision techniques, including eye aspect ratio (EAR) and yawning detection, to identify drowsiness. Upon detecting signs of fatigue, the system triggers an alarm, alerting the driver to prevent accidents caused by drowsiness. This project is particularly beneficial for long-distance and commercial drivers, helping reduce road accidents due to fatigue.

1.2 Target Beneficiary

The Drowsiness Detection System primarily benefits long-distance drivers, transportation service operators, and fleet managers. It helps prevent accidents by alerting drowsy drivers, enhancing safety for both individuals and companies in the transportation sector.

1.3 Project Scope

The Drowsiness Detection System project will be completed in 16-weeks' time, covering research, development, testing, and deployment. It operates on a 16-weeks' timeline with designated roles for team members, focusing on facial recognition to detect drowsiness.

i. Requirements and feasibility done by Chetanshi, Addya

ii. Design and modelling done by Chetanshi, Preeti
iii. Coding and implementation done by Chetanshi Pandey
iv. Documentation done by Chetanshi, Kashish, Preeti, Addya
v. Testing and maintenance done by Chetanshi, Kashish

The project scope involves managing hardware compatibility and performance tuning to ensure reliability across different environments.

1.4 References APA format

[1] Satish, K., Lalitesh, A., Bhargavi, K., Prem, M. S., & Anjali, T. (2020, July). Driver drowsiness detection. In *2020 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0380-0384). IEEE.

[2] Saranya, N., Priyanka, V., Harini, T., Akhila, B., Hemalatha, M. A., & Sre, R. K. (2023, January). Driver State Monitoring System Using AI. In *2023 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-6). IEEE.

[3] Kaufman, H., Woods, J., Dravida, S., & Tekalp, A. (1983). Estimation and identification of two-dimensional images. *IEEE transactions on automatic control*, *28*(7), 745-756.

[4] Soukupova, T., & Cech, J. (2016, February). Eye blink detection using facial landmarks. In *21st computer vision winter workshop, Rimske Toplice, Slovenia* (Vol. 2).

[5] Geetavani, B., Rao, N. K., Shalini, N., Reddy, V. A., Amrutha, A., & Sultana, A. (2021, November). Drivers Fatigue Detection and Alarming Using Ear and MOR Values. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 1023-1029). IEEE.

[6] Essel, E., Lacy, F., Elmedany, W., Albalooshi, F., & Ismail, Y. (2022, October). Driver Drowsiness Detection Using Fixed and Dynamic Thresholding. In *2022 International Conference on Data Analytics for Business and Industry (ICDABI)* (pp. 552-557). IEEE.

[7] Xu, M., Chen, D., & Zhou, G. (2020). Real-time face recognition based on Dlib. In *Innovative Computing: IC 2020* (pp. 1451-1459). Springer Singapore.

[8] Soo, S. (2014). Object detection using Haar-cascade Classifier. *Institute of Computer Science, University of Tartu*, *2*(3), 1-12.

[9] Saini, V., & Saini, R. (2014). Driver drowsiness detection system and techniques: a review. *International Journal of Computer Science and Information Technologies*, *5*(3), 4245-4249.

[10] Deng, W., & Wu, R. (2019). Real-time driver-drowsiness detection system using facial features. *Ieee Access*, *7*, 118727-118738.

2. PROJECT DESCRIPTION

2.1 Data/ Data structure

The data structure used in the projects are:

i. Arrays/Lists (NumPy Arrays)

Eye Landmarks & Lip Landmarks: Arrays from the facial landmarks predictor are used to represent specific coordinates of the face (e.g., eyes, lips).

NumPy is used to manipulate the coordinates for calculating the **Eye Aspect Ratio (EAR)** and the **Mouth Open Ratio (MOR).**

ii. **Tuples**
Tuples are used to store **(x, y)** coordinates of facial landmarks, such as the eyes and lips, retrieved from dlib's 68-point facial landmark predictor.

iii. **Video Stream Data (Frames)**
The frames captured from the webcam act as the primary data source, processed in real-time using OpenCV to detect facial features.
The frame-by-frame processing allows the detection of eyes and lips for further analysis.

iv. **Arguments (ArgumentParser)**
Argparse is used to manage input arguments, allowing for flexibility in selecting the webcam index and providing the path to the alert sound file.

2.2 SWOT Analysis

Strengths	**Weaknesses**
1. **Real-time Monitoring:** Your project processes real-time data from a webcam to detect drowsiness, which can alert users instantly when drowsiness indicators are detected. 2. **Cost-effective:** The project uses common libraries like OpenCV and dlib, which are open-source and free, making it accessible and inexpensive. 3. **Automation:** The system is fully automated once started, making it hands-off for the user. 4. **Machine Learning Techniques:** I'm using established detection algorithms like Haar cascades and dlib's 68-point facial landmark detection to track eye movements and yawning.	1. **Accuracy Limitations:** The thresholds for EAR and lip distance are hardcoded, meaning they might not work equally well for all users unless personalized. 2. **Environmental Limitations:** Conditions like poor lighting or obstructed faces can affect performance, and the webcam's quality might be a limiting factor. 3. **Limited Scope:** It only detects visual symptoms of drowsiness (e.g., eyes, mouth) and doesn't factor in other indicators like heart rate or overall behaviour. 4. **Hardware Dependency:** The system relies on a working webcam, which might not always be available or suitable in all environments (e.g., different vehicles, lighting conditions).
Opportunities	**Threats**
1. **Market Demand**: There's growing demand for driver assistance systems, particularly for improving road safety in commercial and	1. **Privacy Issues:** Since the system monitors the user's face, it could raise privacy concerns regarding data storage and processing.

personal driving, making my project relevant. 2. **Sensor Integration**: my system could be expanded by integrating with additional sensors (e.g., steering wheel sensors or wearables), providing a more comprehensive fatigue detection solution. 3. **Mobile and Wearable Adaptation:** You could adapt this technology for mobile phones or wearable devices, making it portable and widely applicable.	2. **Technological Advancements:** Emerging technologies in driver monitoring could make my solution less relevant over time. 3. **Regulatory Compliance:** Depending on the region, driver monitoring technologies might be subject to regulation, and your system could face challenges if it doesn't meet future legal standards.

2.3 Project Features

The features of this project are:

i. **Webcam Integration:** The system integrates with the webcam to capture video frames and analyse them for signs of drowsiness. It allows selecting different webcams via command-line arguments.

ii. **Real-time Drowsiness Monitoring:** The system continuously monitors the user's face in real-time via a webcam, analysing facial landmarks to detect signs of drowsiness[9], such as drooping eyes and yawning.

iii. **Use of Haar Cascade Classifier for Face Detection:** For face detection, the system uses a Haar Cascade Classifier from OpenCV, which provides a balance between speed and accuracy in detecting faces before applying further landmark detection.

iv. **Facial Landmark Detection:** The system uses dlib's 68 facial landmarks to map key points on the face, such as the eyes and mouth, enabling precise tracking of eye closure and yawning.

v. **Eye Aspect Ratio (EAR) Calculation[4]:** It calculates the Eye Aspect Ratio (EAR) based on specific facial landmarks around the eyes. If the EAR falls below a certain threshold for consecutive frames, it signals eye closure, indicating drowsiness.

vi. **Mouth Open Ratio (MOR[5]:** The system measures lip distance using facial landmarks to detect yawning, which is another indicator of drowsiness. When the mouth open ratio (MOR) exceeds a predefined threshold, the system registers a yawn.

vii. **Alert Mechanism:** If drowsiness is detected (either via eye closure or yawning), the system triggers an alarm sound to alert the user. This uses the play sound module to play an audio file specified in the code.

viii. **Visual Feedback:** The system provides visual feedback by drawing contours around the eyes and mouth on the video feed to indicate the areas being monitored.

2.4 Design and Implementation Constraints

i. **Hardware Constraints:** Webcam Quality.

ii. **Software Constraints:** Libraries and Dependencies, Limited to Predefined Facial Features, Threshold Sensitivity.

iii. **Environmental Constraints:** Lighting Conditions, Camera Positioning.

iv. **Real-time Processing Constraints:** Frame Processing Speed**,** Thread Management**.**

v. **Implementation Constraints:** Alarm Path Dependency, File Size and Execution Time.

2.5 Design diagram

The Figure1 outlines the flow of operations in a **Drowsiness Detection System** using computer vision and machine learning techniques The process begins by capturing live video from a camera, where the resolution of the video is a key input. OpenCV, a widely used computer vision library, is used to extract frames from the continuous video stream. Each frame is then passed through a face detection algorithm using Haar Cascades, which is effective in detecting human faces. This is a classical object detection technique based on machine learning. Once a face is detected, the system uses dlib's machine learning algorithms to mark 68 key facial landmarks on the face, including the eyes, mouth, and head position. Then EAR, MOR, NLR Calculation is done. These values (EAR, MOR, and NLR) are compared to predefined thresholds to determine whether the driver is showing signs of fatigue. Thresholds are used to decide if a particular value (e.g., eye closure or yawning) crosses a limit indicative of drowsiness. In parallel with checking thresholds, the system monitors for yawning to further reinforce the signs of drowsiness. If the **Eye Aspect Ratio (EAR)** falls below the threshold and yawning or head bending is detected, the system concludes the driver is drowsy and, If drowsiness is detected, the system triggers an alert, typically an audio warning, to wake the driver and prevent a possible accident.

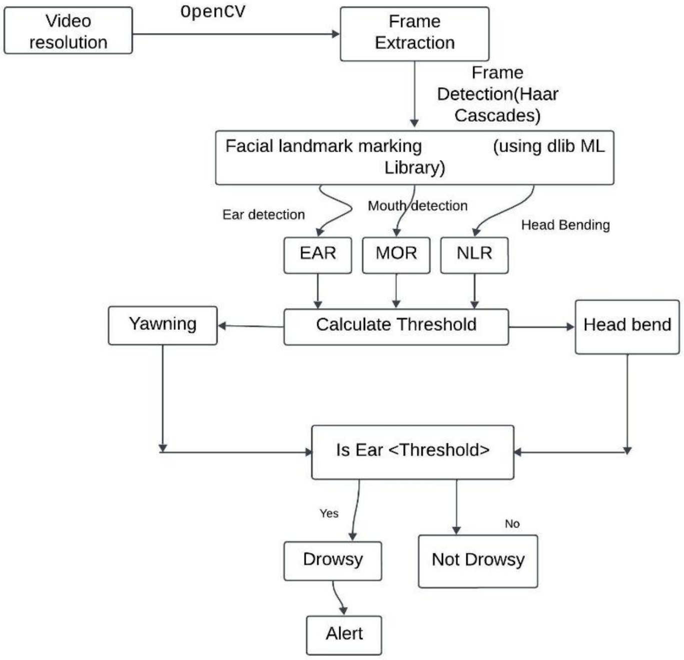


Fig 1: Design diagram

3. SYSTEM REQUIREMENTS

3.1 User Interface

i. **Live Video Feed:** The primary UI component is a live video feed that captures the driver's face in real-time using a connected webcam. This stream shows the user's face and any detected facial landmarks.

ii. **Drowsiness Alerts:** Visual cues on the interface, such as a flashing message (e.g., "Drowsiness Alert" or "Yawn Detected"), indicate when the system detects signs of drowsiness.

iii. **Metrics Display:** The Eye Aspect Ratio (EAR) and Yawn Detection values are shown on the interface, allowing the user to see real-time data related to eye closure and yawning frequency.

iv. **Audio Alert Trigger:** When drowsiness is detected, an audio alert is activated (e.g., an alarm sound), providing auditory feedback through the system's speakers.

3.2 ML Model Used

The ML algorithms used in this project are:

i. **OpenCV:-** It's a video capture protocol through which the system access the webcam and continuously process frames of the live video feed.

ii. **Haar cascade [8] ML Algorithm:-** An Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Paul Viola and Michael Jones in 2001, and it's popular for its real-time performance in tasks like face detection.

iii. **Dlib machine learning library:-** dlib's pre-trained model [7] (shape predictor) uses a trained dataset to detect and track facial landmarks for eyes and mouth in each frame. The 68-point facial landmark model used for detecting eyes and lips is based on the work of Kazemi and Sullivan (2014) in their paper "One Millisecond Face Alignment with an Ensemble of Regression Trees". This model is integrated into the dlib library for robust real-time face landmark detection.

iv. **Eye Aspect Ratio (EAR):-** The eye aspect ratio was introduced by Tereza Soukup ova and Jan Cech in their paper titled [4]"Real-Time Eye Blink Detection using Facial Landmarks" (2016). In this work, they proposed using the EAR as a reliable measure for blink detection, which can be extended to detect drowsiness. The EAR formula used in my code comes directly from their method, where they set the EAR threshold at 0.3 based on experimental data.

v. **Mouth Open Ratio (MOR):-** MOR[6] is widely used in research for yawning detection. One such paper is "Yawning Detection for Monitoring Driver's Drowsiness Based on Two Visual Features" (2015), which analyses lip distances and yawning behaviours.

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Performance requirements

i. **Real-Time Detection:** The system must process video frames at a rate that allows real-time drowsiness [10] detection. Ideally, this would be around 20–30 frames per second (FPS) to ensure smooth and continuous monitoring of the driver.

ii. **Low Latency:** The delay between the detection of drowsiness signs and the triggering of alerts must be minimal to ensure prompt intervention.

iii. **Resource Efficiency:** The system should operate efficiently on moderate hardware, such as a standard laptop or desktop with minimal CPU and memory usage. The usage of multi-threading is essential to ensure smooth video processing and alarm activation without lags.

iv. **Scalability:** The system should be scalable to handle different camera resolutions and varying processing power.

4.2 Security requirements

i. **Data Privacy:** Since the system deals with sensitive video data, it must ensure that video streams are processed locally, and no data is transmitted to external servers or third-party applications without user consent.

ii. **Restricted Access**: The application should ensure that unauthorized users cannot alter the system settings, such as changing the threshold values for detection, alarm settings, or sensitivity configurations.

iii. **Malware Protection:** The system should be protected from malware that could tamper with the detection logic or alarm functions, ensuring the integrity of the software.

4.3 Software Quality Attributes

i. **Reliability**: The system must consistently and accurately detect signs of drowsiness or yawning without false positives or negatives. It should remain stable during extended periods of monitoring (e.g., several hours of continuous usage).

ii. **Usability**: The interface must be user-friendly and straightforward, requiring minimal interaction from the user. Setup and configuration should be easy for non-technical users.

iii. **Maintainability**: The codebase should be modular and well-documented to allow easy maintenance, updates, and the addition of new features or improvements.

iv. **Portability**: The software should run on multiple operating systems, such as Windows, Linux, and macOS, with minimal modification.

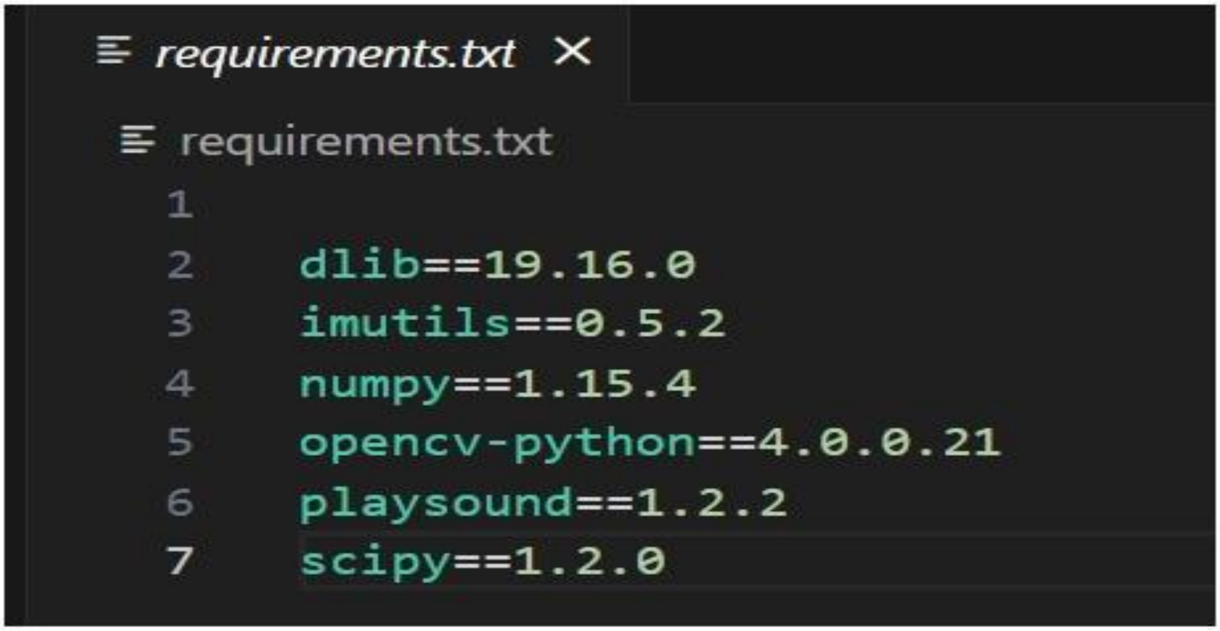
v. **Extensibility**: The architecture should allow for the integration of additional features, such as head tilt detection or integration with other safety systems, without significant redesign.

vi. **Resilience**: The system should handle errors gracefully, such as when the webcam feed is interrupted or the alert sound file is missing, and provide meaningful feedback to the use

5 CODE IMPLEMENTATION

5.1 Code (Backend)

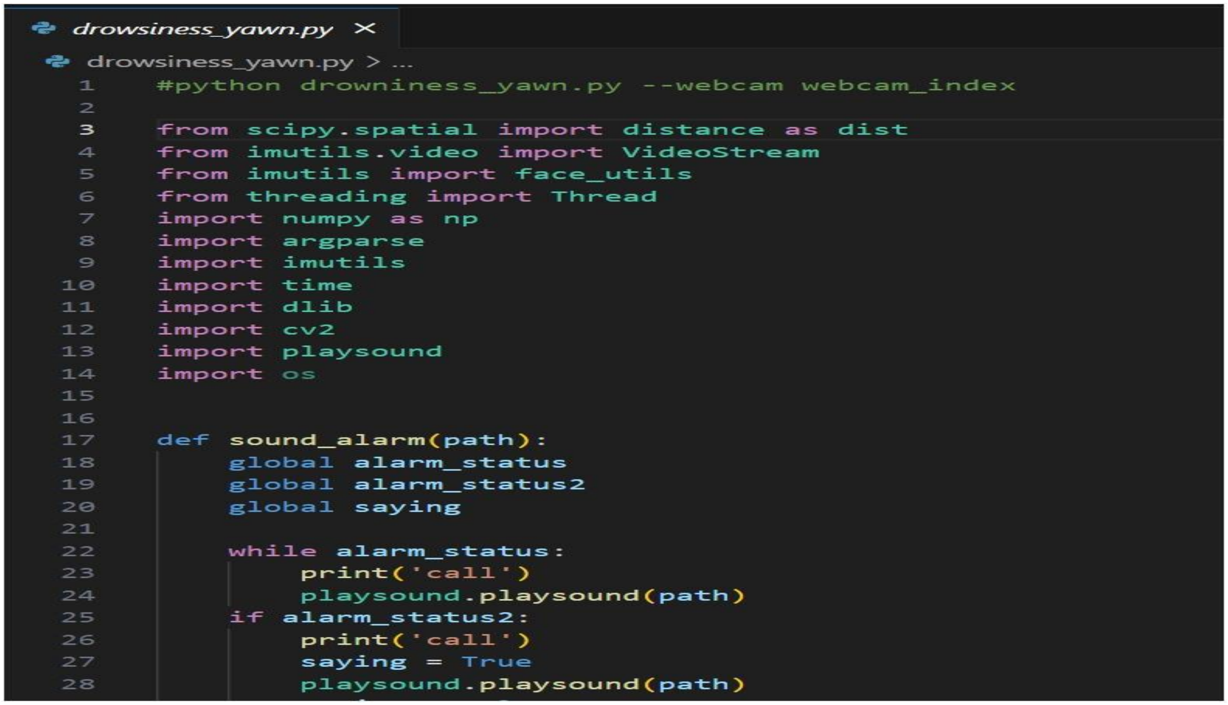
Figure 2 represents the snapshot of the requirement.txt file in which we cover all requirement needed for my project like dlib ML Algorithm, numpy, opencv-python, playsound, and some more.



```

dlib==19.16.0
imutils==0.5.2
numpy==1.15.4
opencv-python==4.0.0.21
playsound==1.2.2
scipy==1.2.0
```

Fig 2: Snapshot of Requirements.txt

Figure 3 represents the snapshot of the drowsiness_yawn.py is shown in which all the functions for alarm, calculation and comparisons are build.

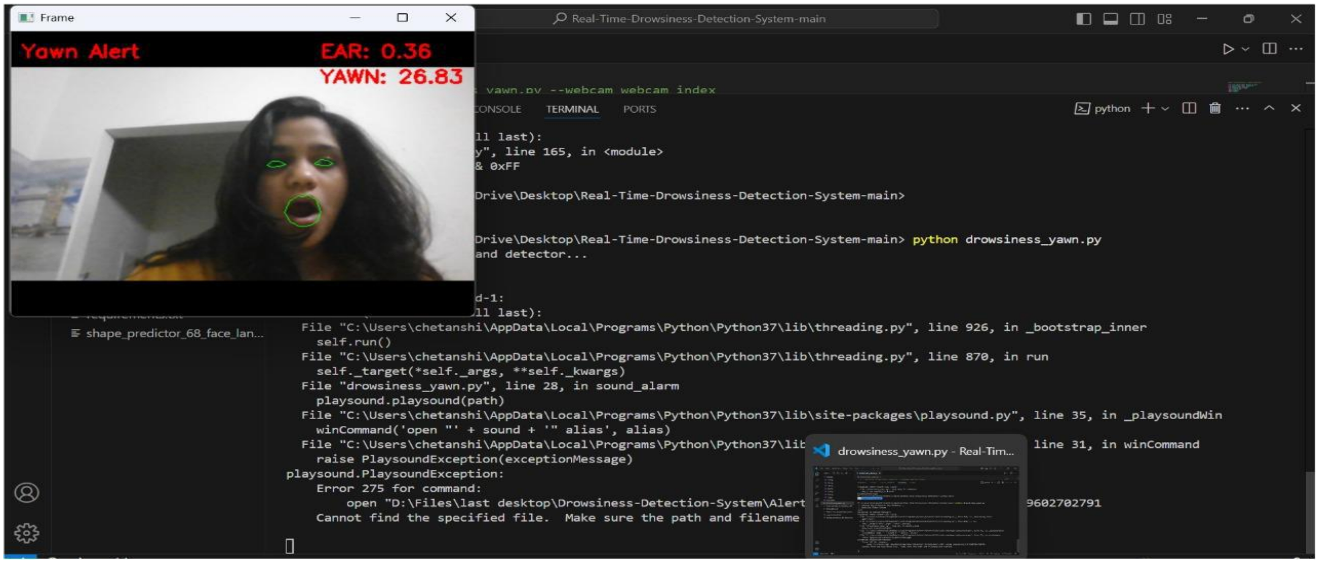


```python
#python drowniness_yawn.py --webcam webcam_index

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 argparse
import imutils
import time
import dlib
import cv2
import playsound
import os


def sound_alarm(path):
    global alarm_status
    global alarm_status2
    global saying

    while alarm_status:
        print('call')
        playsound.playsound(path)
    if alarm_status2:
        print('call')
        saying = True
        playsound.playsound(path)
```

Fig 3: Snapshot of Python code

5.2 Output

1 Mouth Open Ratio (MOR)- Figure 4 represents the snapshot of the output of Yawn Alert generated because of yawning.

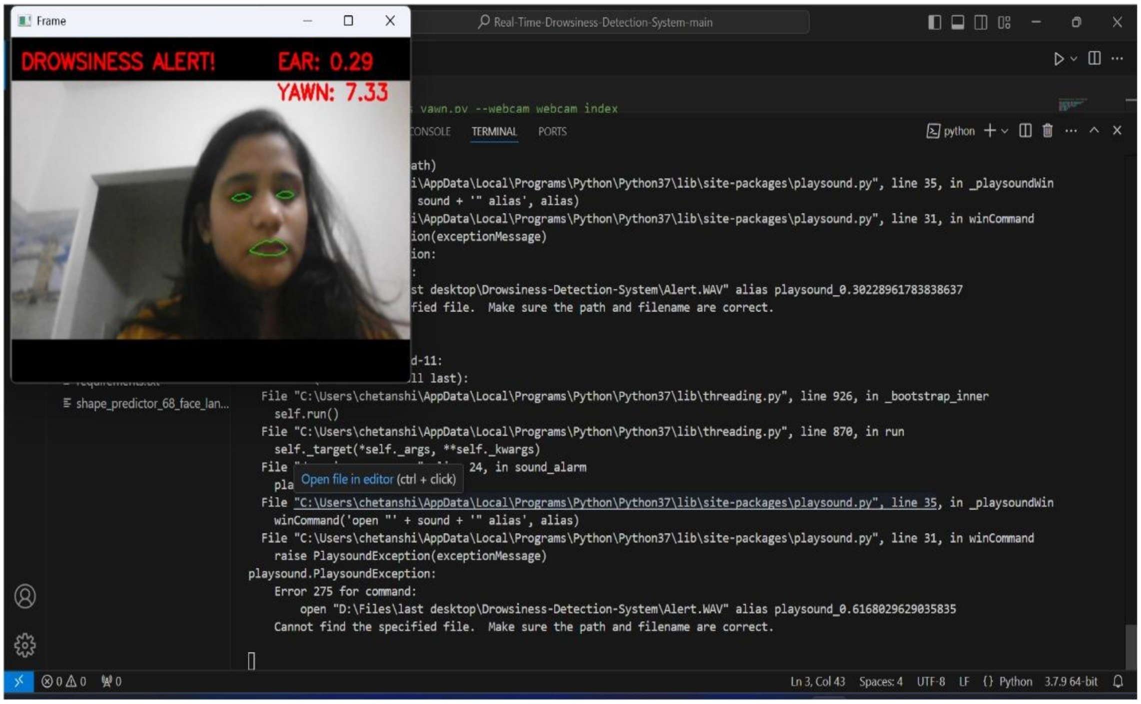

Frame
Real-Time-Drowsiness-Detection-System-main
CONSOLE
TERMINAL
PORTS
python
shape_predictor_68_face_lan...
drowsiness_yawn.py - Real-Tim...

```
Yawn Alert                EAR: 0.36
                        YAWN: 26.83
```

```
 yawn.py --webcam webcam index
ll last):
y", line 165, in <module>
& 0xFF

Drive\Desktop\Real-Time-Drowsiness-Detection-System-main>

Drive\Desktop\Real-Time-Drowsiness-Detection-System-main> python drowsiness_yawn.py
and detector...

d-1:
ll last):
  File "C:\Users\chetanshi\AppData\Local\Programs\Python\Python37\lib\threading.py", line 926, in _bootstrap_inner
    self.run()
  File "C:\Users\chetanshi\AppData\Local\Programs\Python\Python37\lib\threading.py", line 870, in run
    self._target(*self._args, **self._kwargs)
  File "drowsiness_yawn.py", line 28, in sound_alarm
    playsound.playsound(path)
  File "C:\Users\chetanshi\AppData\Local\Programs\Python\Python37\lib\site-packages\playsound.py", line 35, in _playsoundWin
    winCommand('open "' + sound + '" alias', alias)
  File "C:\Users\chetanshi\AppData\Local\Programs\Python\Python37\lib                       line 31, in winCommand
    raise PlaysoundException(exceptionMessage)
playsound.PlaysoundException:
    Error 275 for command:
        open "D:\Files\last desktop\Drowsiness-Detection-System\Alert                       9602702791
    Cannot find the specified file.  Make sure the path and filename
```

Fig 4: Snapshot of MOR output

3 **(EAR) Eye Aspect Ratio-** Figure 5 represents the snapshot of the output of Drowsiness Alert generated because of eye closing.


Frame
Real-Time-Drowsiness-Detection-System-main
CONSOLE
TERMINAL
PORTS
python
shape_predictor_68_face_lan...
Open file in editor (ctrl + click)
Ln 3, Col 43
Spaces: 4
UTF-8
LF
{} Python
3.7.9 64-bit

```
DROWSINESS ALERT!         EAR: 0.29
                         YAWN: 7.33
```

```
 yawn.py --webcam webcam index
ath)
i\AppData\Local\Programs\Python\Python37\lib\site-packages\playsound.py", line 35, in _playsoundWin
 sound + '" alias', alias)
i\AppData\Local\Programs\Python\Python37\lib\site-packages\playsound.py", line 31, in winCommand
ion(exceptionMessage)
ion:
:
st desktop\Drowsiness-Detection-System\Alert.WAV" alias playsound_0.30228961783838637
fied file.  Make sure the path and filename are correct.


d-11:
ll last):
  File "C:\Users\chetanshi\AppData\Local\Programs\Python\Python37\lib\threading.py", line 926, in _bootstrap_inner
    self.run()
  File "C:\Users\chetanshi\AppData\Local\Programs\Python\Python37\lib\threading.py", line 870, in run
    self._target(*self._args, **self._kwargs)
  File "                 24, in sound_alarm
    pla
  File "C:\Users\chetanshi\AppData\Local\Programs\Python\Python37\lib\site-packages\playsound.py", line 35, in _playsoundWin
    winCommand('open "' + sound + '" alias', alias)
  File "C:\Users\chetanshi\AppData\Local\Programs\Python\Python37\lib\site-packages\playsound.py", line 31, in winCommand
    raise PlaysoundException(exceptionMessage)
playsound.PlaysoundException:
    Error 275 for command:
        open "D:\Files\last desktop\Drowsiness-Detection-System\Alert.WAV" alias playsound_0.6168029629035835
    Cannot find the specified file.  Make sure the path and filename are correct.
```

Fig 5: Snapshot of EAR output

6 Millstone Completed

Serial No.	**Objectives**	**Completed till.**
1.	Yawn detection through Mouth Open Ratio (MOR).	Mid Term
2.	Drowsiness Alert through (EAR) Eye Aspect Ratio.	Mid Term
3.	Alert sound generation.	Mid Term
4.	Drowsiness Alert through Neck bend.	End Term
5.	Dynamically calculate threshold from the first appearance of a person's image.	End Term