A Project Report on

Driver Drowsiness Detection System

A Dissertation submitted to JNTU Hyderabad in partial fulfilment of the academic requirements for the award of the degree.

Master of Technology

in

Computer Science and Engineering

Submitted by

DEVUNI SATHISH 23H51D5803

Under the esteemed guidance of

G RAVI KUMAR

(Associate Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC & JNTUH, Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NBA.)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2023- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

Department Of Computer Science and Engineering

CERTIFICATE

This is to certify that the Mini Project report entitled " **Driver Drowsiness Detection System** " being submitted by **Devuni Sathish** ***(23H51D5803)*** in partial fulfilment for the award of **Master of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embody in this project report have not been submitted to any other University or Institute for the award of any Degree.

GUIDE
G Ravi Kumar
Associate Professor
Dept. of CSE

Head of Dept.
Dr. S. SIVA SKANDA
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure I want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **G Ravi Kumar**, Associate Professor, Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. S. Siva Skanda,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. G. Devadas**, Dean-Academic, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

Finally, I express my sincere thanks to Mr. Ch. Gopal Reddy, Secretary, CMR Group of Institutions, for his continuous care. I sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

Devuni Sathish 23H51D5803

ABSTRACT

A project on drowsiness detection system. A countless number of people drive on the highway day and night. Taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep. Due to which it becomes very dangerous to drive when feeling sleepy. The majority of accidents happen due to the drowsiness of the driver. So, to prevent these accidents we will build a system using Python, OpenCV, and Keras which will alert the driver when he feels sleepy. Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving.

The **objective** of this intermediate Python project is to build a drowsiness detection system that will detect that a person’s eyes are closed for a few seconds. This system will alert the driver when drowsiness is detected.

The requirement for this Python project is a webcam through which we will capture images. You need to have Python (3.6 version recommended) installed on your system, then using pip, you can install the necessary packages.

1. **OpenCV –** pip install opencv-python (face and eye detection).
2. **TensorFlow –** pip install tensorflow (keras uses TensorFlow as backend).
3. **Keras –** pip install keras (to build our classification model).
4. **Pygame –** pip install pygame (to play alarm sound).

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

INTRODUCTION

The Driver Drowsiness Detection System presented in this project addresses a critical issue in transportation safety – the risk of accidents caused by driver fatigue. Fatigue-related accidents have long been a concern, with drowsy driving leading to impaired judgment, slower reaction times, and increased collision risks. To combat this, automated drowsiness detection systems have emerged as vital tools for preventing accidents and saving lives. This project focuses on developing a robust and efficient drowsiness detection system using computer vision and machine learning techniques. By analyzing facial features, eye movements, and physiological signals, the system can accurately identify signs of drowsiness in drivers in real-time. Integrated with an alert mechanism, such as auditory warnings or haptic feedback, the system promptly notifies drivers when fatigue is detected, prompting them to take necessary breaks or corrective actions. This introduction sets the stage for understanding the significance and urgency of implementing drowsiness detection systems to enhance driver safety and prevent accidents on our roads.

The Driver Drowsiness Detection System presented in this project operates within the domain of computer vision and machine learning applied to transportation safety. In this domain, advanced technologies are utilized to analyze driver behavior and detect signs of drowsiness in real-time, thereby preventing accidents caused by driver fatigue. The system employs several key libraries and algorithms to achieve accurate and efficient drowsiness detection. MediaPipe, a state-of-the-art library for facial landmark detection, is used to track key facial features, particularly focusing on eye movements and eyelid behavior. OpenCV (cv2) is employed for image processing tasks, allowing for the extraction of relevant facial features and the calculation of metrics such as the Eye Aspect Ratio (EAR). The system's detection algorithm relies on the EAR metric, which quantifies eye openness and closure, enabling the differentiation between alert and drowsy states. By continuously monitoring changes in EAR values and eye behavior, the system can accurately detect signs of drowsiness and promptly alert the driver. This introduction provides an overview of the domain, libraries, and detection techniques employed in the Driver Drowsiness Detection System, highlighting their importance in enhancing driver safety on the roads.

1.1 PROBLEM DEFINITION

Driver drowsiness poses a significant safety risk on roads, leading to a high number of accidents globally. The problem arises when drivers experience fatigue, causing impaired judgment, slower reaction times, and an increased likelihood of accidents. Traditional methods of addressing drowsy driving, such as periodic breaks or caffeine consumption, are reactive and may not always be effective. Therefore, there is a need for proactive measures to detect and mitigate drowsiness in drivers in real-time.

The Driver Drowsiness Detection System aims to address this problem by developing an automated system capable of accurately detecting signs of drowsiness in drivers and providing timely alerts to prevent accidents. The system utilizes computer vision techniques to analyze facial features and eye movements, focusing on indicators such as eyelid behavior and Eye Aspect Ratio (EAR). By continuously monitoring these indicators, the system can identify when a driver is becoming drowsy and prompt them to take necessary breaks or corrective actions to ensure road safety.

Overall, the problem definition revolves around the need for an efficient and reliable method to detect and mitigate driver drowsiness, thereby reducing the risk of accidents and promoting safer driving behaviour.

1.2 OBJECTIVE

Develop an Automated Drowsiness Detection System: The primary objective of the project is to develop an automated system capable of detecting signs of drowsiness in drivers using computer vision techniques.

Real-time Detection: Implement a system that can monitor driver behavior in real-time, continuously analyzing facial features and eye movements to identify signs of drowsiness promptly.

Accurate Detection: Ensure the system can accurately differentiate between alert and drowsy states by analyzing key indicators such as eyelid behavior and Eye Aspect Ratio (EAR).

Prompt Alerts: Integrate an alert mechanism into the system to notify drivers when signs of drowsiness are detected, prompting them to take immediate corrective actions to ensure road safety.

User-friendly Interface: Develop a graphical user interface (GUI) that is intuitive and non-distracting, allowing for easy interaction with the system and providing clear feedback to the driver.

Enhance Road Safety: Ultimately, the overarching objective of the project is to enhance road safety by preventing accidents caused by driver fatigue through proactive drowsiness detection and timely alerts.

1.3 SCOPE

- **Facial Landmark Detection:** The project will focus on implementing facial landmark detection techniques to track key facial features, particularly the eyes, which are crucial for drowsiness detection.

- **Eye Movement Analysis:** The system will analyze eye movements, including blink duration and frequency, to identify signs of drowsiness in drivers.

- **Eye Aspect Ratio (EAR) Calculation:** The project will calculate the Eye Aspect Ratio (EAR) using the positions of facial landmarks around the eyes to quantify eye openness and closure.

- **Real-time Processing:** The system will operate in real-time, continuously monitoring driver behavior and providing immediate alerts when signs of drowsiness are detected.

- **Alert Mechanism:** An alert mechanism will be integrated into the system to notify drivers through auditory or visual cues when drowsiness is detected.

- **Graphical User Interface (GUI):** A user-friendly GUI will be developed to facilitate interaction with the system, allowing users to upload images for analysis and view drowsiness detection results.

- **Testing and Evaluation:** The project will involve testing the system using a variety of scenarios and datasets to evaluate its accuracy and effectiveness in detecting drowsiness.

- **Documentation and Reporting:** Comprehensive documentation will be provided, including project specifications, implementation details, and evaluation results, to ensure reproducibility and future scalability of the system.

- **Limitations:** The project will acknowledge and document any limitations of the system, such as environmental factors that may affect detection accuracy or hardware/software constraints.

- **Future Extensions:** While the initial scope of the project will focus on basic drowsiness detection, potential future extensions may include additional features such as driver identification, fatigue trend analysis, or integration with vehicle control systems for automated interventions.

Modules of Driver Drowsiness Detection System:

- **MediaPipe**: Utilized for facial landmark detection, MediaPipe provides pre-trained models for accurate tracking of key facial features, including the eyes, nose, and mouth.

- **OpenCV (cv2)**: OpenCV is a widely-used library for computer vision tasks. In this project, it is used for image processing operations, such as loading images, resizing, and drawing shapes, as well as for calculating the Eye Aspect Ratio (EAR) from detected facial landmarks.

- **Pyttsx3**: This library is used for text-to-speech conversion, enabling the system to audibly alert the driver when signs of drowsiness are detected.

- **tkinter**: Tkinter is a standard GUI toolkit for Python. It is used to create the graphical user interface (GUI) for the drowsiness detection system, providing buttons for image upload and display areas for visual feedback.

- **PIL (Python Imaging Library)**: PIL is used for basic image processing tasks, such as loading and resizing images. It is used in conjunction with OpenCV for image manipulation within the GUI.

- **NumPy**: NumPy is a fundamental package for numerical computing in Python. It is used for handling arrays and mathematical operations, particularly in the calculation of EAR from facial landmarks.

CHAPTER 2
EXISTING SYSTEM

2.1 Existing System

The existing systems for drowsiness detection in drivers typically fall into two categories: hardware-based systems and software-based systems.

1. **Hardware-based Systems**:
   - These systems often involve the use of specialized sensors or cameras installed in vehicles to monitor driver behavior. They may include features such as:
     - Infrared cameras to track eye movements and pupil dilation.
     - Steering wheel sensors to detect changes in grip strength or steering patterns.
     - Wearable devices, such as EEG headbands or heart rate monitors, to measure physiological signals indicative of drowsiness.
   - These systems are often integrated into vehicles as part of advanced driver assistance systems (ADAS) or in-cabin monitoring systems.
2. **Software-based Systems**:
   - Software-based systems rely on computer vision and machine learning algorithms to analyze images or video streams of the driver's face captured by standard cameras or webcams. They may include features such as:
     - Facial landmark detection to track key facial features, particularly focusing on the eyes and mouth.
     - Eye tracking algorithms to monitor eye movements and blink patterns.
     - Machine learning models trained on labeled datasets to classify driver states as alert or drowsy based on features extracted from facial images or video frames.
   - These systems are often implemented as standalone applications or integrated into existing in-vehicle infotainment systems.

Existing systems may vary in terms of accuracy, reliability, and implementation complexity. Some systems may require specialized hardware or calibration, while others rely solely on software-based approaches. Additionally, the effectiveness of these systems may depend on factors such as lighting conditions, camera quality, and the presence of occlusions or distractions within the vehicle cabin. Overall, existing systems aim to enhance driver safety by providing early warnings of drowsiness and promoting timely interventions to prevent accidents caused by driver fatigue.

2.2 Problems in existing system:

While existing drowsiness detection systems offer valuable capabilities in enhancing driver safety, they are not without limitations. Some common problems and challenges associated with existing systems include:

- **Hardware Complexity and Cost**: Many existing systems rely on specialized hardware components, such as infrared cameras, steering wheel sensors, or EEG headbands, which can be expensive to install and maintain. This complexity and cost may limit the accessibility of these systems, especially for individual drivers or smaller vehicle fleets.
- **Calibration and Maintenance**: Hardware-based systems may require frequent calibration and maintenance to ensure accurate and reliable performance. Factors such as changes in lighting conditions, camera angles, or environmental factors can affect system accuracy and may necessitate periodic adjustments or recalibrations.
- **False Alarms and Inaccuracies**: Existing systems may be prone to false alarms or inaccuracies, particularly in challenging driving conditions or for drivers with unique facial features or behaviors. False alarms can lead to driver frustration and reduced trust in the system, while inaccuracies may compromise safety by failing to detect genuine instances of drowsiness.
- **Limited Compatibility**: Some existing systems may have limited compatibility with certain vehicle models or may require modifications to the vehicle's design or onboard systems for integration. This limited compatibility can restrict the adoption of these systems across different vehicle types and fleets.
- **Privacy Concerns**: Systems that rely on cameras or sensors to monitor driver behavior may raise privacy concerns among drivers regarding the collection and use of personal data. Ensuring user privacy and data security is essential for the widespread acceptance and adoption of drowsiness detection systems.
- **Environmental Factors**: Environmental factors such as glare, reflections, or occlusions within the vehicle cabin can pose challenges for accurate drowsiness detection. Existing systems may struggle to perform reliably under such conditions, leading to reduced effectiveness in real-world driving scenarios.
- **Driver Adaptation**: Drivers may adapt their behavior in response to system alerts, leading to a phenomenon known as "alarm fatigue" where drivers become desensitized to repeated warnings. Effective drowsiness detection systems should account for driver adaptation and provide alerts that are timely and meaningful.

2.3 Existing Systems:

Several specified systems exist in the domain of drowsiness detection for drivers, both in research and commercial applications. Some notable examples include:

1. Nissan Driver Attention Alert System:

Nissan's Driver Attention Alert System utilizes steering wheel sensors to monitor driver inputs and detect changes in driving behavior indicative of drowsiness or fatigue. The system provides visual and auditory alerts to prompt the driver to take a break or rest

.

2. SmartEye DMS (Driver Monitoring System):

SmartEye DMS is a commercial driver monitoring system that utilizes infrared cameras and advanced computer vision algorithms to track driver eye movements, facial expressions, and head position. It provides real-time alerts to prevent accidents caused by drowsy or distracted driving.

3. Seeing Machines Guardian System:

The Seeing Machines Guardian System is an in-cabin monitoring system designed for commercial vehicles and heavy machinery. It uses specialized cameras and machine learning algorithms to detect signs of driver fatigue or distraction, providing immediate alerts to fleet operators.

4. EyeSight Driver Assist Technology (Subaru):

Subaru's EyeSight Driver Assist Technology includes features such as lane departure warning, adaptive cruise control, and pre-collision braking. It also incorporates a driver monitoring system that tracks eye movements and alerts the driver if signs of drowsiness or distraction are detected.

5. **FORD Driver Alert System**:

The FORD Driver Alert System is a feature available in certain Ford vehicles that monitors driver behavior using steering wheel inputs and vehicle movement patterns. It provides visual and auditory warnings when signs of drowsiness or fatigue are detected.

These specified systems often combine hardware components, such as cameras or sensors, with software algorithms for drowsiness detection and alerting. They are typically integrated into vehicle platforms and may offer advanced features such as adaptive interventions or integration with vehicle control systems.

Comparison between Existing Systems and Developed System:

Hardware Requirement:

Existing Systems: Many existing systems require specialized hardware components, such as infrared cameras, steering wheel sensors, or wearable devices, to monitor driver behavior.

Developed System: The developed system relies primarily on standard hardware components, such as webcams or onboard cameras commonly found in vehicles, making it more accessible and cost-effective to implement.

Accuracy and Reliability:

Existing Systems: Hardware-based systems may offer high accuracy and reliability, especially those utilizing specialized sensors or EEG-based devices. However, they may also be prone to false alarms or inaccuracies due to calibration issues or environmental factors.

Developed System: The developed system aims to achieve comparable levels of accuracy and reliability through computer vision and machine learning algorithms. While it may not match the precision of some hardware-based systems, it offers a practical solution that can be easily deployed in a wide range of vehicles without additional hardware requirements.

Integration Complexity:

Existing Systems: Integrating hardware-based systems into vehicles may require modifications to the vehicle's design or onboard systems, leading to increased complexity and cost.

Developed System: The developed system is software-based and can be implemented as a standalone application or integrated into existing in-vehicle infotainment systems with minimal integration complexity.

Real-time Monitoring:

Existing Systems: Both hardware-based and software-based systems can offer real-time monitoring of driver behavior, providing timely alerts when signs of drowsiness are detected.

Developed System: The developed system operates in real-time, continuously monitoring driver facial features and eye movements to detect drowsiness promptly and trigger alerts when necessary.

Cost and Accessibility:

Existing Systems: Hardware-based systems may be expensive to install and maintain, limiting their accessibility to certain vehicle models or fleet operators.

Developed System: The developed system is cost-effective and accessible, requiring only standard hardware components and open-source libraries for implementation. It can be easily deployed in a wide range of vehicles, including consumer vehicles, commercial fleets, and public transportation.

2.4 GAPS IN EXISTING SOLUTIONS:

Despite the advancements in drowsiness detection systems, there are several gaps and areas for improvement in existing solutions:

Limited Accessibility: Many existing systems rely on specialized hardware components or proprietary software, making them costly and inaccessible to the general public. There is a need for more affordable and readily available solutions that can be easily integrated into a wide range of vehicles.

Reliability in Real-world Conditions: Existing systems may struggle to maintain accuracy and reliability in real-world driving conditions, such as varying lighting conditions, occlusions, or distractions within the vehicle cabin. Improvements are needed to enhance the robustness of these systems under diverse environmental conditions.

Adaptability to Driver Preferences: Drowsiness detection systems may not account for individual differences in driver behavior or preferences. Customizable settings and adaptive algorithms could improve user satisfaction and system effectiveness by accommodating varying driving styles and preferences.

Integration with Vehicle Control Systems: While some existing systems provide alerts to drivers, there is potential for further integration with vehicle control systems to implement automated interventions, such as adjusting seat positions, activating seat vibrators, or triggering lane departure warning systems, to prevent accidents caused by drowsy driving.

Privacy and Data Security: Concerns about privacy and data security may hinder the widespread adoption of drowsiness detection systems. Addressing these concerns through transparent data handling practices and robust security measures is essential to building trust among users and stakeholders.

Real-time Feedback and Intervention: Existing systems may provide alerts after drowsiness has already been detected, leaving little time for drivers to react. Implementing real-time feedback and intervention mechanisms could enable proactive measures to prevent drowsiness before it becomes a safety hazard.

User Experience and Acceptance: Improving the user experience and acceptance of drowsiness detection systems is crucial for their effectiveness. User-friendly interfaces, clear communication of alerts, and minimizing false alarms can enhance user satisfaction and encourage adoption.

CHAPTER 3
PROPOSED METHOD

3.1 OVERVIEW OF PROPOSED METHOD

The proposed method for drowsiness detection in drivers utilizes computer vision techniques and machine learning algorithms to accurately identify signs of drowsiness in real-time. The method consists of several key components:

Facial Landmark Detection: The first step involves detecting facial landmarks using the MediaPipe library. This allows the system to locate key points on the driver's face, particularly focusing on the eyes.

Eye Aspect Ratio (EAR) Calculation: Once facial landmarks are detected, the system calculates the Eye Aspect Ratio (EAR) using the positions of specific landmarks around the eyes. The EAR metric quantifies eye openness and closure, providing a measure of drowsiness.

Real-time Monitoring: The system continuously monitors changes in the EAR value over time, analyzing eye movements and blink patterns to detect signs of drowsiness as they occur in real-time.

Drowsiness Classification: Using machine learning techniques, the system classifies the driver's state as either alert or drowsy based on the calculated EAR value and other features extracted from facial landmarks. A trained classifier, such as a Support Vector Machine (SVM) or Convolutional Neural Network (CNN), is used for this purpose.

Alert Generation: If the system detects signs of drowsiness, it triggers an alert mechanism to notify the driver. This can include auditory alerts using the pyttsx3 library, visual cues within the graphical user interface (GUI), or potentially haptic feedback through vibration sensors.

User Interaction: The system provides feedback to the user through a user-friendly GUI, allowing them to visualize their drowsiness status and take appropriate actions, such as taking a break or adjusting their driving behavior.

Evaluation and Testing: The proposed method is evaluated using real-world driving scenarios and benchmark datasets to assess its accuracy, reliability, and effectiveness in detecting drowsiness. Performance metrics such as precision, recall, and F1-score are used to evaluate the system's performance.

3.2 Requirement Analysis

Here we have used HARDWARE & SOFTWARE products for the completion of project and there are User Requirements & Project Requirements.

3.3 Technologies Used :

The technologies used for the drowsiness detection system involves several key libraries and frameworks that support computer vision, machine learning, image processing, and user interaction. The primary backend technologies used are:

Python:

The entire project is implemented in Python, a versatile and widely-used programming language that offers extensive libraries and frameworks for computer vision, machine learning, and GUI development.

MediaPipe:

Purpose: Used for real-time facial landmark detection.

Functionality: Provides pre-trained models to accurately detect and track facial features such as eyes, nose, and mouth.

OpenCV (cv2):

Purpose: Used for image processing and manipulation.

Functionality: Facilitates operations such as reading images from video streams, resizing images, and drawing shapes on images. It also aids in calculating the Eye Aspect Ratio (EAR) by processing the coordinates of facial landmarks.

Pyttsx3:

Purpose: Used for text-to-speech conversion.

Functionality: Generates auditory alerts to notify the driver when drowsiness is detected.

tkinter:

Purpose: Used for developing the graphical user interface (GUI).

Functionality: Provides a user-friendly interface for uploading images, displaying real-time feedback, and interacting with the drowsiness detection system.

PIL (Python Imaging Library):

Purpose: Used for basic image processing tasks.

Functionality: Handles operations like loading and resizing images, often used in conjunction with OpenCV for GUI image display.

NumPy:

Purpose: Used for numerical operations.

Functionality: Assists in handling arrays and performing mathematical operations, particularly for calculating the EAR from the coordinates of detected facial landmarks.

3.4 Workflow

Facial Landmark Detection:

MediaPipe detects facial landmarks from the input image or video stream, identifying key points around the eyes and other facial features.

Eye Aspect Ratio (EAR) Calculation:

OpenCV processes the detected landmarks to compute the EAR, which helps determine the eye's state (open or closed).

Real-time Monitoring:

Continuous monitoring of the EAR value to detect drowsiness in real-time.

Drowsiness Classification:

A simple rule-based approach or a machine learning classifier determines whether the driver is alert or drowsy based on EAR and other extracted features.

Alert Generation:

Pyttsx3 generates an auditory alert, and the GUI (tkinter) displays visual feedback when drowsiness is detected.

User Interaction:

The tkinter-based GUI allows users to upload images for analysis, view real-time detection results, and receive alerts.

This backend technology stack provides a robust foundation for implementing a real-time driver drowsiness detection system, leveraging Python's rich ecosystem of libraries for computer vision, image processing, and user interaction.

3.2.2 Front End Technology

The frontend of the drowsiness detection system primarily focuses on providing an intuitive and user-friendly interface for interacting with the backend functionalities. The primary frontend technology used is:

Tkinter:

Purpose: Tkinter is a standard GUI (Graphical User Interface) library in Python. It is used to create the graphical interface for the application.

Functionality:

- **Widgets**: Tkinter provides various widgets such as buttons, labels, and file dialog boxes that facilitate user interaction with the application.
- **Layout Management**: It manages the layout and organization of the interface components.
- **Event Handling**: Handles events like button clicks and image uploads, triggering corresponding backend processes.

Components and Workflow

Image Upload Interface:

File Dialog: Utilizes `tkinter.filedialog.askopenfilename` to allow users to select an image file from their local system.

Display Area: Displays the uploaded image within the application window using `tkinter.Label` and `PIL.ImageTk`.

Real-time Monitoring and Feedback:

Display Window: Shows real-time feedback from the drowsiness detection system, including visual indicators of the detected facial landmarks and EAR values.

Alerts: Displays visual alerts within the GUI when drowsiness is detected, in addition to triggering auditory alerts.

User Interaction Elements:

Buttons: Provides buttons for actions like uploading an image and starting the real-time monitoring process.

Labels: Displays instructions, statuses, and alerts to the user.

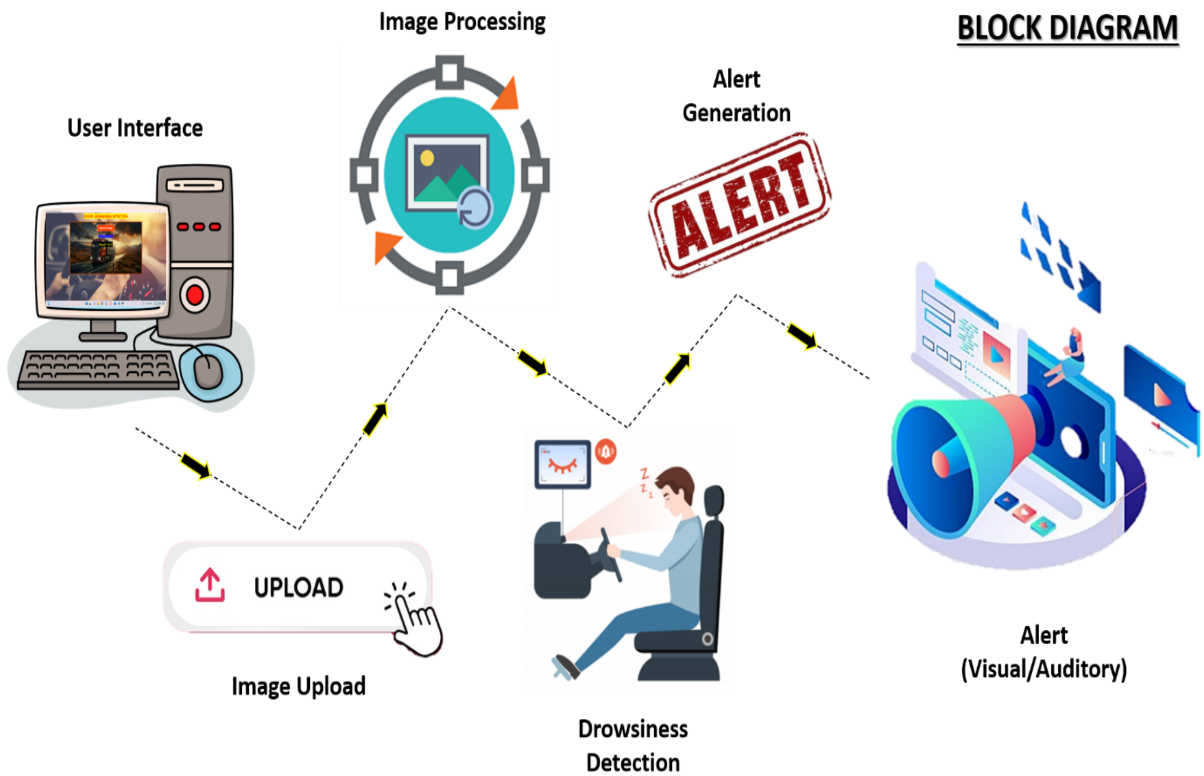
Hardware Specifications

Operating System: Windows 10.
Processor: i3 and above.
Ram: 4 GB.
Hard Disk Space: 5 GB.

Benefits:

- **Safety**: By detecting drowsiness early, the system can help prevent accidents caused by driver fatigue.
- **User-Friendly**: The intuitive UI makes it easy for users to interact with the system and understand the alerts.
- **Real-Time Alerts**: The system provides immediate feedback, allowing users to take action before it's too late.
- **Scalability**: The modular architecture allows for future enhancements, such as incorporating additional sensors or improving detection algorithms.

3.5 SYSTEM ARCHITECHTURE



In this block diagram:

- **User Interface**: Interacts with the user, allowing them to upload images for processing.
- **Image Processing**: Processes the uploaded image (e.g., resizing, preprocessing).
- **Drowsiness Detection**: Analyzes the processed image to detect signs of drowsiness.
- **Alert Mechanism**: Generates alerts (visual, auditory) if drowsiness is detected.
- **Integration**: Handles communication between different components and coordinates the flow of data and control.

Each component performs a specific function within the system, and data flows sequentially from one component to the next. This block diagram provides a high-level overview of the system architecture, illustrating the main components and their interactions.

Architecture Overview:

- The user uploads an image through the UI.
- The image processing module prepares the image for analysis.
- The drowsiness detection module analyzes the image to determine if the user is drowsy.
- If drowsiness is detected, the alert mechanism generates visual and auditory alerts.
- The integration module ensures smooth communication and data transfer between components.

3.6 SOURCE CODE

drowsiness_detection.py

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'

import cv2
import mediapipe as mp
import numpy as np
from scipy.spatial import distance as dist

mp_face_mesh = mp.solutions.face_mesh

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

def detect_drowsiness(image_path):
    EYE_AR_THRESH = 0.3

    frame = cv2.imread(image_path)
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    with mp_face_mesh.FaceMesh(static_image_mode=True, max_num_faces=1,
min_detection_confidence=0.5) as face_mesh:
        results = face_mesh.process(frame_rgb)

        if not results.multi_face_landmarks:
            return "No face detected"

        for face_landmarks in results.multi_face_landmarks:
            # Define landmark indices for the left and right eyes
            left_eye_indices = [33, 160, 158, 133, 153, 144]
            right_eye_indices = [362, 385, 387, 263, 373, 380]

            left_eye = np.array([(face_landmarks.landmark[i].x, face_landmarks.landmark[i].y) for i
in left_eye_indices])
            right_eye = np.array([(face_landmarks.landmark[i].x, face_landmarks.landmark[i].y) for i
in right_eye_indices])
```

```
            left_eye *= [frame.shape[1], frame.shape[0]]
            right_eye *= [frame.shape[1], frame.shape[0]]

            left_ear = eye_aspect_ratio(left_eye)
            right_ear = eye_aspect_ratio(right_eye)
            ear = (left_ear + right_ear) / 2.0

            print(f"Left EAR: {left_ear}, Right EAR: {right_ear}, Average EAR: {ear}")

            if ear < EYE_AR_THRESH:
                return "!!! You are in Drowsy Mode !!!"
            else:
                return "You are in Alert Mode"

    return "No face detected"
```

gui.py

```
import tkinter as tk
from tkinter import filedialog
from tkinter import messagebox
from PIL import Image, ImageTk, ImageDraw
import os
import pyttsx3
import threading
from drowsiness_detection import detect_drowsiness

def upload_image():
    file_path = filedialog.askopenfilename()
    if file_path:
        load_image(file_path)
        result = detect_drowsiness(file_path)
        result_label.config(text=f"{result}", font=("Helvetica", 20, "bold"))
        if result == "!!! You are in Drowsy Mode !!!":
            alarm()

def load_image(file_path):
    img = Image.open(file_path)
    img = img.resize((250, 250), Image.Resampling.LANCZOS)
    img = ImageTk.PhotoImage(img)
    panel.config(image=img)
    panel.image = img

def alarm():
```

```
    threading.Thread(target=speak, args=("Danger! Danger! Danger! Drowsiness detected its danger! Please take a break.",)).start()
    messagebox.showwarning("Drowsiness Detected", "Drowsiness detected! Please take a break.")

def speak(text):
    engine = pyttsx3.init()
    engine.say(text)
    engine.runAndWait()

def set_frame_background():
    frame.update_idletasks()  # Ensure the frame dimensions are updated
    frame_width = frame.winfo_width()
    frame_height = frame.winfo_height()

    frame_background_image = Image.open("frame_background.jpg")
    frame_background_image = frame_background_image.resize((frame_width, frame_height), Image.Resampling.LANCZOS)
    frame_background_image = ImageTk.PhotoImage(frame_background_image)

    frame_background_label.config(image=frame_background_image)
    frame_background_label.image = frame_background_image

def create_rounded_button_image(width, height, radius, color):
    image = Image.new("RGBA", (width, height), (255, 255, 255, 0))
    draw = ImageDraw.Draw(image)
    draw.rounded_rectangle((0, 0, width, height), radius, fill=color)
    return ImageTk.PhotoImage(image)

# Create the main window
root = tk.Tk()
root.title("Drowsiness Detection System")
root.geometry(f"{root.winfo_screenwidth()}x{root.winfo_screenheight()}")

# Load and set background image
background_image = Image.open("background.jpg")
background_image = background_image.resize((root.winfo_screenwidth(), root.winfo_screenheight()), Image.Resampling.LANCZOS)
background_image = ImageTk.PhotoImage(background_image)

background_label = tk.Label(root, image=background_image)
background_label.place(relwidth=1, relheight=1)

# Create frame for content with transparent-like background b33c00
frame = tk.Frame(root, bg="#b33c00", bd=5)
frame.place(relx=0.5, rely=0.1, relwidth=0.60, relheight=0.6, anchor='n')
```

```
# Create label for the frame background image
frame_background_label = tk.Label(frame)
frame_background_label.place(relwidth=1, relheight=1)

#frame.tk_setPalette(background="#ff6600")

# Create rounded button image
rounded_button_image = create_rounded_button_image(200, 50, 20, "#ff4000")

# Create upload buttons and labels
panel = tk.Label(text="DRIVER DROWSINESS
DETECTION",bg="YELLOW",fg="RED",font=("Comic Sans MS",20,"bold") )
panel.pack(pady=5)

upload_btn = tk.Button(frame, text="Upload Image", command=upload_image, font=("serif",
18,"bold"), fg="white", compound="center")
upload_btn.config(image=rounded_button_image, width=200, height=50, borderwidth=0,
highlightthickness=0)
upload_btn.pack(pady=20)

panel = tk.Label(frame, text="Image Uploading!!! please wait",bg="blue",fg="white")
panel.pack(pady=10)

result_label = tk.Label(frame, text="Result: N/A", font=("Comic Sans MS", 20), bg='#000000',
fg="yellow")
result_label.pack(pady=20)

# Update frame background after main loop starts
root.after(100, set_frame_background)

root.mainloop()
```

CHAPTER 4
RESULTS

4.1 RESULTS

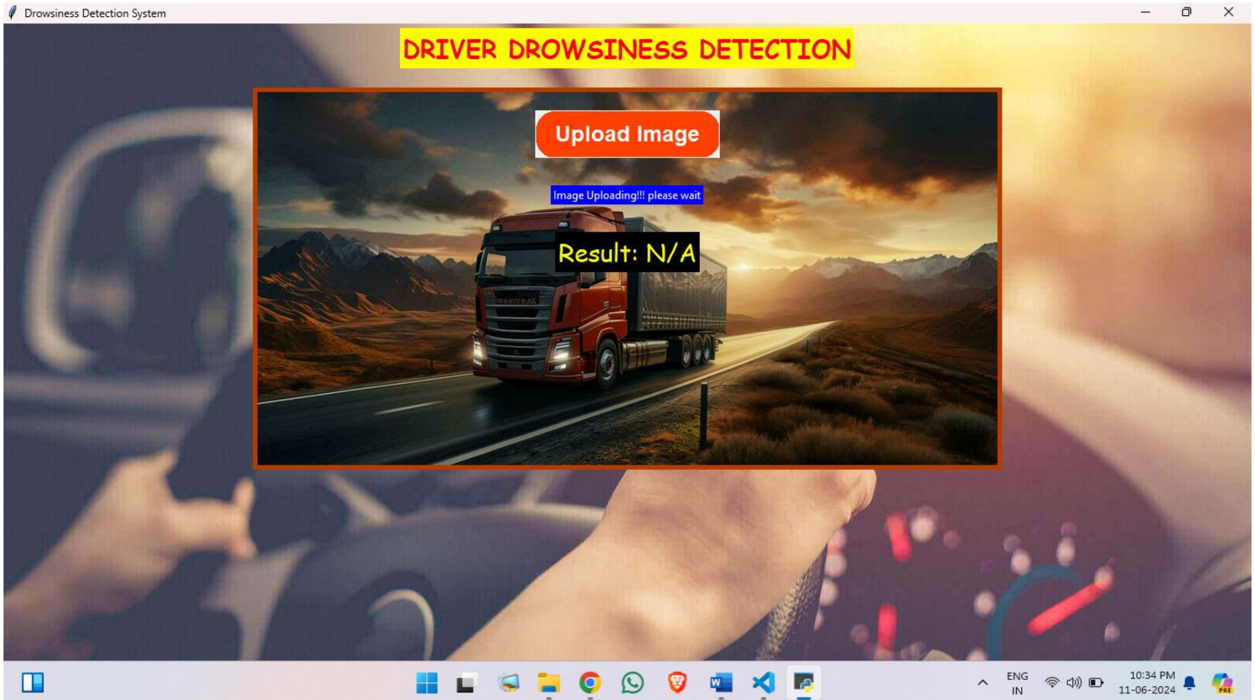


Fig-1: *DASHBOARD*

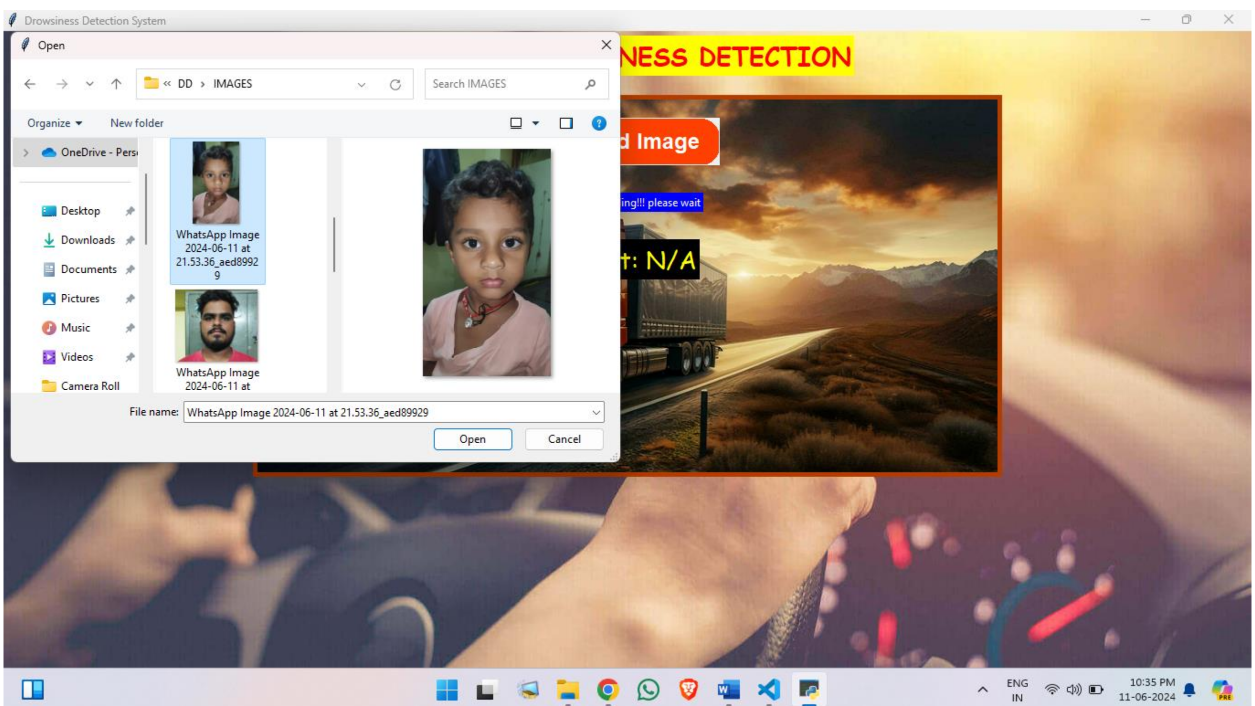


Fig-2: *UPLOADING IMAGE*

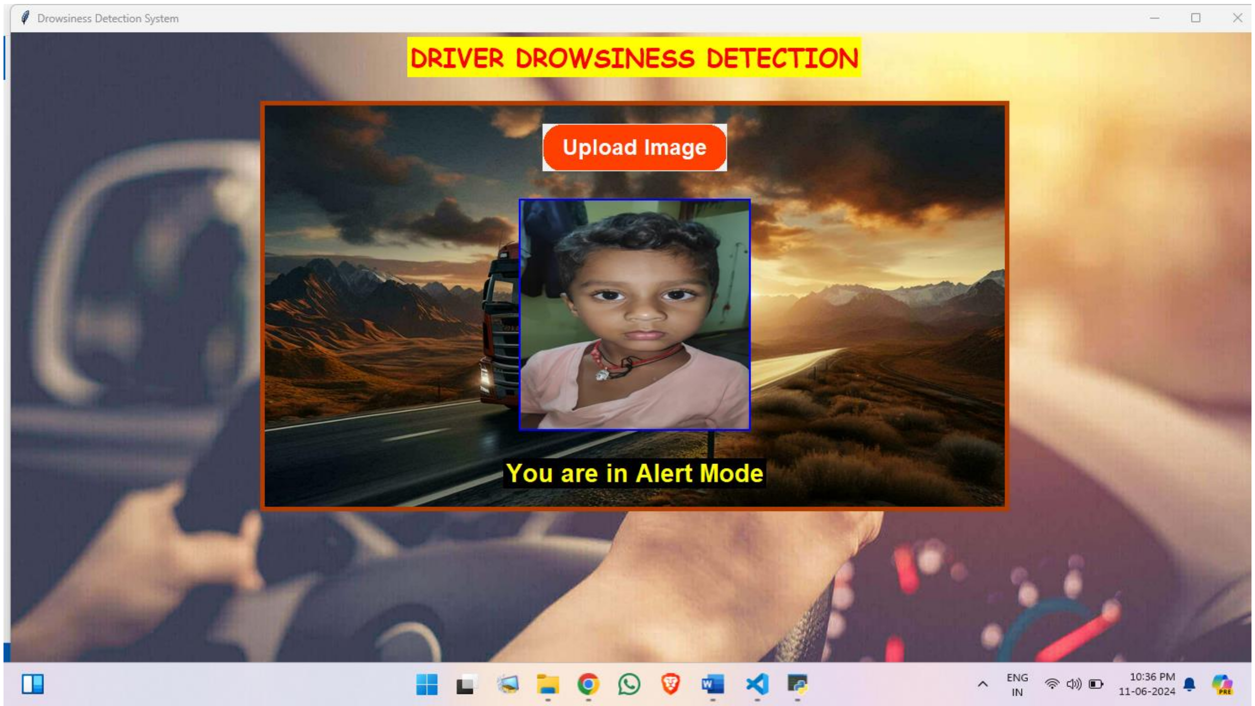


Fig-3: *Image uploaded and detected*

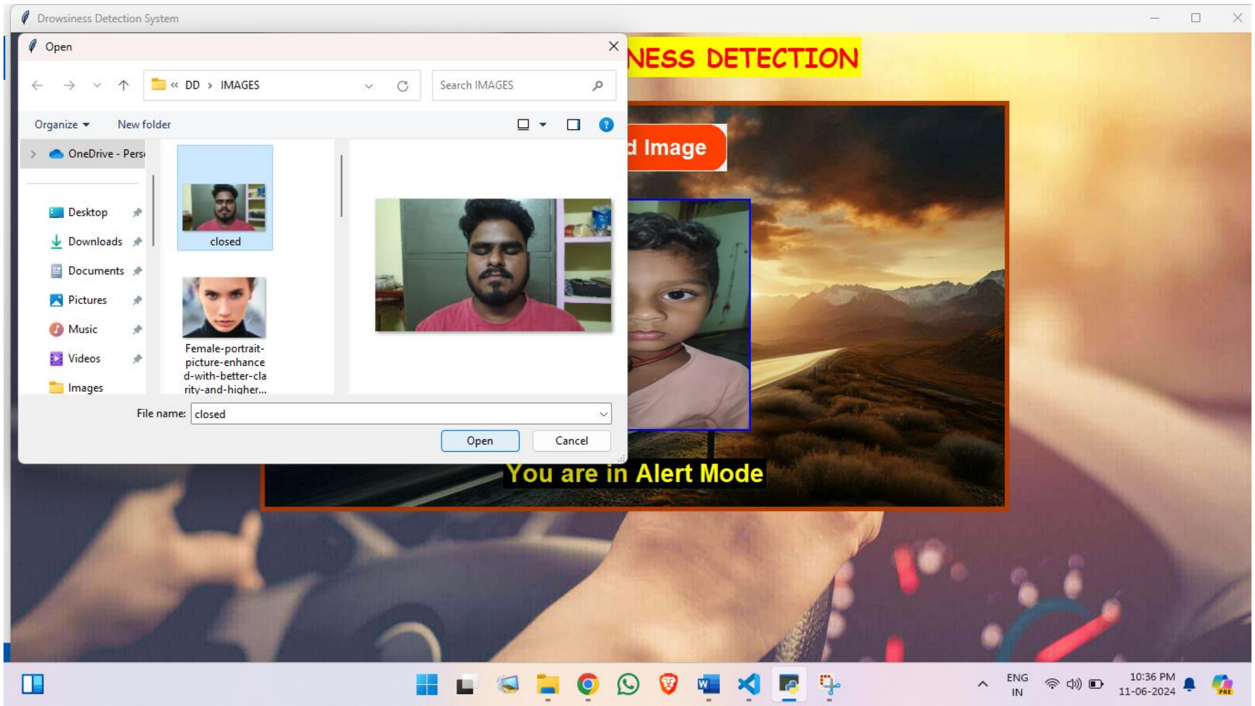


Fig-4: *Next Image uploading*

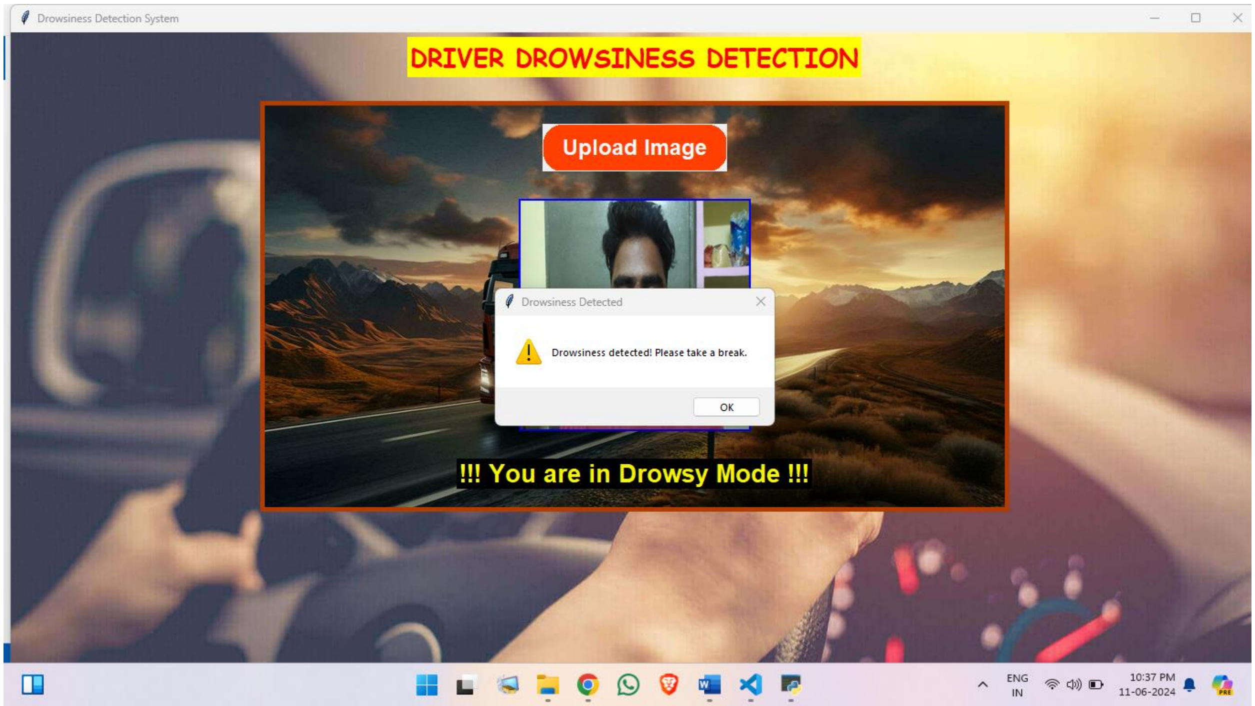


Fig-5: Image *Detection and giving alert with audio and statement*

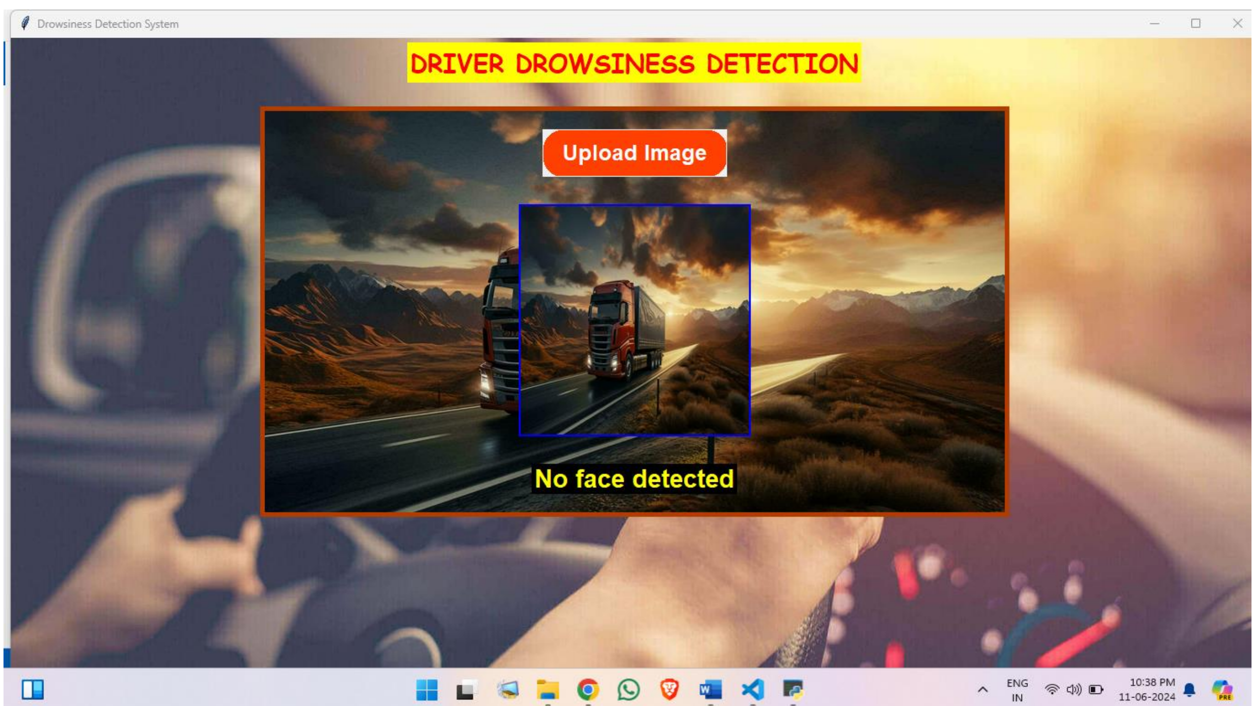


Fig-6 : *uploading no face image and detected*

4.2 Data Collection:

- **Camera Setup:** Install cameras in vehicles to record drivers under controlled and naturalistic conditions.
- **Simulated Driving Environments:** Use driving simulators to induce drowsiness and record data.
- **Annotations:** Manually label data with states of drowsiness, typically using expert reviews or self-reports from drivers.

Performance Metrics for Evaluation

Common Metrics:

- **Accuracy:** Measures the overall correctness of the system in classifying drowsy and non-drowsy states.
- **Precision:** The ratio of true positive drowsiness detections to the total predicted positives.
- **Recall (Sensitivity):** The ratio of true positive drowsiness detections to all actual positives.
- **F1-Score:** The harmonic mean of precision and recall, providing a single metric that balances both.
- **Receiver Operating Characteristic (ROC) Curve:** Plots the true positive rate against the false positive rate, illustrating the trade-off between sensitivity and specificity.
- **Area Under the ROC Curve (AUC):** Summarizes the ROC curve into a single value representing the overall ability of the model to distinguish between classes.

Specific Metrics for Drowsiness Detection:

- **Eye Aspect Ratio (EAR):** Measures the ratio of distances between facial landmarks around the eyes to detect eye closure.
- **Mouth Aspect Ratio (MAR):** Measures the openness of the mouth to detect yawning.
- **Perclos (Percentage of Eye Closure):** Measures the percentage of time the eyes are closed over a given period, a key indicator of drowsiness.

4.3 Comparative Analysis of Systems

Existing Solutions and Their Performance:

1. **Commercial Systems (e.g., Mercedes-Benz Attention Assist):**
    - **Metrics:** High accuracy in detecting drowsiness based on steering behavior and driving patterns.
    - **Data:** Uses vehicle dynamics and driver interaction with the steering wheel.
    - **Performance:** Provides real-time alerts with integrated vehicle systems.
2. **Research-Based Systems (e.g., Deep Learning Models):**
    - **Metrics:** Accuracy often exceeds 90% with high precision and recall.
    - **Data:** Utilizes visual data from cameras, often combined with physiological data.

- **Performance:** Can achieve real-time processing with optimized models on embedded systems.

3. **DIY/Open-Source Systems (e.g., OpenCV-Based Detection):**
   - **Metrics:** Moderate accuracy, precision, and recall, depending on lighting and camera quality.
   - **Data:** Relies primarily on visual data, with basic facial landmark detection.
   - **Performance:** Real-time capabilities on standard hardware, but may require tuning for specific conditions.

CHAPTER 5
CONCLUSION

CONCLUSION

The drowsiness detection system project aims to enhance safety by alerting users when signs of drowsiness are detected. This comprehensive system integrates multiple components, including a user interface, image processing, drowsiness detection algorithms, and alert mechanisms, to provide a robust solution.

Key Components:

User Interface (UI): Provides a platform for users to interact with the system, upload images, and receive feedback. The UI is designed to be user-friendly, with intuitive controls and clear display elements.

Image Processing: Preprocesses the uploaded images to ensure they are in the correct format and quality for analysis. This step may involve resizing, grayscale conversion, and noise reduction.

Drowsiness Detection: Utilizes machine learning algorithms to analyze the processed images and detect signs of drowsiness. This component is critical for accurately identifying when the user is at risk.

Alert Mechanism: Generates visual and auditory alerts if drowsiness is detected. This dual alert system ensures that the user is promptly and effectively notified to take necessary actions, such as stopping for a rest.

Integration: Manages the communication and data flow between all components, ensuring the system operates seamlessly. This includes handling image data from the UI, processing it, analyzing it for drowsiness, and triggering alerts when necessary.

Future Enhancements:

- **Real-Time Video Analysis**: Expanding the system to analyze real-time video streams for continuous monitoring.
- **Multi-Sensor Integration**: Incorporating data from other sensors (e.g., steering patterns, heart rate monitors) to improve detection accuracy.
- **Machine Learning Improvements**: Training the detection algorithms on larger datasets to enhance accuracy and reduce false positives.

In conclusion, this drowsiness detection system is a vital tool for enhancing safety by alerting users to potential drowsiness. The project successfully integrates several advanced technologies and provides a solid foundation for future improvements and expansions.

REFERENCES

- https://docs.python.org/3/c-api/index.html
- www.tutorialspoint.com
- https://code.visualstudio.com/docs
- https://marketplace.visualstudio.com/items?itemName=ms-python.vscode-pylance
- https://numpy.org/doc/stable/reference/index.html
- https://www.tensorflow.org/
- NTHU Driver Drowsiness Detection Dataset - National Tsing Hua University.
- UTA Real-Life Drowsiness Dataset (UTA-RLDD) - University of Texas at Arlington.
- "Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks" by Bhargava Reddy et al. - IEEE CVPR Workshops, 2017. Read the paper
- "Driver Drowsiness Detection Using Machine Learning and Image Processing" by Mittal et al. - ICRITO, 2021. Read the paper
- Bhargava Reddy, Ye-Hoon Kim, Sojung Yun, Chanwon Seo, Junik Jang. "Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.
- "Real-Time Driver Drowsiness Detection and Classification on Embedded Systems Using Machine Learning Algorithms." International Information and Engineering Technology Association (IIETA).