,

AwakeGuard: Real-Time Drowsiness Detection System

Capstone project submitted as part of the fulfillment of the course curriculum for the third semester of the

Master of Science

in

Artificial Intellegence and Machine Learning

by

Akanksha Kumari

(MSA23003)

under the guidance of

Dr. Saurabh Shukla

Indian Institute of Information Technology, Lucknow

2023-25

Declaration by Candidate

I hereby declare that the project titled **"AwakeGuard: Real-Time Drowsiness Detection System"** is the outcome of my original work, which I have undertaken with full dedication and commitment. This project was conducted under the expert guidance and supervision of **Dr. Saurabh Shukla** at the Indian Institute of Information Technology, Lucknow. I confirm that the research and development presented in this project are entirely my own, and no part of this work has been copied or plagiarized from any other source.

I further declare that this project has not been submitted, either in whole or in part, for the award of any other degree, diploma, or academic credit at this or any other educational institution. All sources of information, literature, and references used during the course of this project have been duly acknowledged in the report. I understand and respect the ethical guidelines regarding plagiarism and intellectual property, and I have ensured that this work adheres to the highest standards of academic integrity.

I take full responsibility for the content and accuracy of this project and the research conducted, and I am confident that it reflects my learning, efforts, and academic abilities.

Akanksha Kumari
Roll No: MSA23003
(Department of Information Technology)

Certificate by Supervisor

This is to certify that the capstone project titled "**AwakeGuard: Real-Time Drowsiness Detection System**" submitted by Akanksha Kumari (Roll No.: MSA23003) has been carried out under my guidance and supervision. This project has been conducted as part of the fulfillment of the course curriculum for the third semester of the Master of Science degree in Artificial Intelligence and Machine Learning.

The work presented in this report is, to the best of my knowledge, a result of the student's independent efforts, and it meets the academic standards required for this degree program. I hereby endorse the project and recommend it for evaluation.

Dr. Saurabh Shukla

(Department of Computer Science)

Indian Institute of Information Technology, Lucknow

Date: ___________ Signature: __________________

Acknowledgment

I would like to express my heartfelt gratitude to my supervisor, **Dr. Saurabh Shukla**, for his invaluable guidance, encouragement, and continuous support throughout the development of this project. His expertise, insightful feedback, and constructive criticism have been pivotal in shaping this work. His mentorship not only provided me with the necessary academic direction but also motivated me to push through the challenges and complexities encountered during the research process. His patience and commitment to my growth have been truly inspiring, and I am deeply thankful for his unwavering belief in my abilities.

I am also profoundly grateful to the esteemed faculty and dedicated staff at the Indian Institute of Information Technology, Lucknow. Their collective effort in fostering an inspiring academic environment and providing access to resources and facilities has been essential for the successful completion of this project. The knowledge shared by the faculty, along with their encouragement and guidance, has played a significant role in honing my research skills and enhancing my understanding of the subject matter.

I would also like to express my sincere appreciation to my peers and friends, whose companionship and assistance made this journey both enriching and memorable. Their encouragement, whether through brainstorming ideas, reviewing my work, or offering moral support, has been a constant source of motivation. Their contributions, both academically and emotionally, have been invaluable, and I am thankful for their camaraderie during this time.

Lastly, I extend my gratitude to my family for their love, understanding, and belief in me. Their constant support provided the strength and resilience needed to persevere and successfully complete this project.

This project would not have been possible without the collective support and encouragement of all the individuals mentioned above. I am deeply thankful to each one of them for their contributions.

Abstract

A system for detecting drowsiness developed to enhance road safety by tracking driver alertness in real-time. As driver fatigue remains a leading cause of road accidents worldwide, AwakeGuard leverages advanced technologies including machine learning, computer vision, and real-time video processing to detect early indicators of fatigue and alert the driver before fatigue leads to dangerous driving behavior. The system continuously analyzes the driver's facial expressions, eye movements, and head position through a high-resolution camera, using computer vision algorithms to detect subtle indicators of fatigue, such as eyelid drooping, head nodding, and gaze deviation. Integrated seamlessly with existing vehicle safety infrastructure, AwakeGuard provides non-intrusive, real-time alerts that encourage the driver to take corrective actions, like pulling over or taking a break. The system offers a user-friendly interface, with customizable, gentle warnings, ensuring a non-disruptive experience. By providing an effective and proactive solution to fatigue-related accidents, AwakeGuard contributes to reducing the risk of collisions, saving lives, and improving overall road safety. Furthermore, its potential to integrate with other Advanced Driver-Assistance Systems (ADAS) positions it as a key component in the evolution of smart vehicle safety technologies, ultimately aiming for safer roads globally.

Contents

Chapter 1

Introduction

AwakeGuard is developed to predict and help reduce the dangers of drowsy driving though monitoring drivers in real-time and classifying their alertness levels. The system leverages facial recognition and machine learning techniques to identify when a driver is becoming fatigued and proactively issues alerts to prevent accidents. The data used in this system comprises video footage, which captures facial landmarks to assess the driver's level of alertness. The state of alertness is labeled, with normal states marked as 'Alert' and fatigued states as 'drowsy'. By studying these facial characteristics, AwakeGuard aims to provide early warnings that could save lives.

Driver fatigue detection poses unique challenges, as real-time monitoring requires sophisticated image processing and machine learning models to achieve accuracy. Existing vehicle safety systems primarily focus on crash mitigation, such as automatic braking, but these do not address the root cause—driver drowsiness. By integrating AwakeGuard into vehicles, drivers can receive real-time feedback that goes beyond standard safety features. While traditional methods of fatigue detection, like questionnaires or subjective assessments, have limitations, AwakeGuard offers a data-driven approach that continuously monitors the driver's state and adapts to changing conditions. This system relies on facial cues, such as blinking frequency and eye closure duration, analyzed using algorithms for real-time performance.

The deployment of AwakeGuard contributes to public safety by using cutting-edge machine learning techniques to detect signs of drowsiness and prevent accidents. AwakeGuard can be enhanced with additional data to improve detection capabilities and adapt to a wide range of user demographics.

1.1 Background

Driver fatigue is recognized as one of the leading factors contributing to road accidents globally. According to studies conducted by traffic safety organizations, drowsy driving poses risks similar to those associated with impaired driving under the influence of alcohol or drugs. Fatigue significantly affects reaction time, cognitive processing, and decision-making abilities, making drivers more prone to errors and accidents. The increasing demand for long-distance travel and daily commutes in the modern era amplifies this problem. This growing concern necessitates the development of advanced and reliable technological solutions to detect and prevent drowsy driving before accidents occur. The implementation of AwakeGuard aims to address these challenges by employing real-time video analysis and machine learning to monitor driver alertness and provide timely warnings.

1.2 Relevance of the Project

Drowsy driving poses a substantial risk to public safety, often underrecognized in its impact compared to other driving impairments. Research and statistics from various transportation safety bodies indicate that drowsiness contributes to a significant number of vehicle accidents, injuries, and fatalities each year. Unlike impairments due to alcohol or drug consumption, which are often associated with specific demographics, drowsiness can affect any driver, regardless of age or experience level, making it a widespread and universal concern. Recognizing and mitigating this risk through advanced detection techniques can greatly enhance road safety and reduce the number of incidents linked to driver fatigue.

1.3 Problem Statement

To develop a system that performs real-time detection of driver fatigue indicators to prevent accidents.

1.4 Objective

The AwakeGuard project aims to achieve the following objectives:

- Lay the groundwork for future research into additional driver impairments and expand the system's capabilities for comprehensive driver monitoring.
- Design the system to integrate smoothly with existing vehicle infrastructure and other safety features, enhancing overall vehicle safety.
- Utilize advanced machine learning algorithms and computer vision techniques to accurately identify drowsy states based on facial features and behaviors.
- Design the system to integrate smoothly with existing vehicle infrastructure and other safety features, enhancing overall vehicle safety.

1.5 Scope of project

The AwakeGuard Program aims to develop, implement, and validate a comprehensive system that can monitor driver alertness in real-time and provide timely alerts to prevent accidents caused by driver fatigue. The project scope includes the research, development, testing, and deployment phases, ensuring the system is robust, accurate, and user-friendly.

1.6 Tools and Technologies

Image Processing libraries like OpenCV, Dlib, and Mediapipe for real-time image and video processing, including facial landmark detection and eye tracking, A Camera is needed for capturing the Driver and other sensors , Integrated Development Environments (IDE) like vscode is required to writing and testing code.

1.7 Methodology

The AwakeGuard system's methodology involves a step-by-step process to capture and analyze video data, focusing on facial and eye landmarks to determine drowsiness levels. The system processes input in real-time, ensuring that alerts are generated without delay when fatigue is detected.
Steps involved:

- **Camera Initialization:** The system initializes a high-resolution camera capable of capturing the driver's facial features in various lighting conditions.
- **Frame Capturing:** Continuous frames of the driver's face are captured for analysis.
- **Facial and Eye Detection:** The system detects the driver's face and identifies key facial landmarks, especially around the eyes, using pre-trained machine learning models.
- **EAR Calculation:** The Eye Aspect Ratio (EAR) is computed to assess the openness of the eyes. This metric helps determine if the driver's eyes are closed or partially closed.
- **Threshold Assessment:** The EAR is compared to a predefined threshold. If the ratio stays below this threshold for a set number of consecutive frames, the system flags it as a sign of drowsiness.
- **Alert Activation:** When drowsiness is detected, AwakeGuard activates visual and auditory alerts to draw the driver's attention and prevent accidents.

Chapter 2

System Requirement

The Drowsy Driver Detection System requires a high-resolution infrared or standard camera to capture the driver's facial expressions and eye movements, integrated with a real-time processing unit such as a microcontroller (e.g., Raspberry Pi or Arduino). The system should be capable of running image processing libraries like OpenCV and machine learning frameworks like TensorFlow or PyTorch. Sufficient storage for collected data and models, possibly managed by cloud services like AWS or Google Cloud, is essential. The alert mechanism necessitates audio playback systems or vibration motors for immediate feedback. Finally, the system must be non-intrusive, work effectively under varying lighting conditions, and integrate seamlessly with the vehicle's existing infrastructure via protocols such as CAN Bus.

2.1 Software Requirement

- **Libraries and Frameworks:** OpenCV, Dlib, MediaPipe, PyGame, etc.
- **IDE:** Visual Studio Code
- **Python:** The script is written in Python, so Python installation is necessary. Python 3.x is recommended.

2.2 Hardware Requirement

- **Camera:**
  - A webcam or any compatible camera connected to your system. The script uses `cv2.VideoCapture(0)` to access the default camera (0 indicates the first available camera).
- **System Performance:**
  - The script's performance can vary based on the resolution and frame rate of the camera feed and the computational load of the image processing algorithms.

Chapter 3

Literature Survey

S.No.	Authors/ Source	Title/Topic	Summary/Key Points	Key Technologies/ Methods
1.	GeeksforGeeks	Driver Distraction and Drowsiness Detection System (D-Cube)	Discusses a system designed for detecting driver distraction and drowsiness using image processing and computer vision techniques to ensure road safety.	OpenCV, Machine Learning, Image Processing
2.	OpenCV Documentation	Open Source Computer Vision Library	Comprehensive documentation on OpenCV, outlining various image processing tools and algorithms crucial for implementing real-time drowsiness detection systems.	Image Processing, Feature Detection, Real-Time Analysis

S.No.	Authors/ Source	Title/Topic	Summary/Key Points	Key Technologies/ Methods
3.	Musale, Pansambal	Real-Time Driver Drowsiness Detection System Using Computer Vision Techniques	Details the use of computer vision algorithms for real-time monitoring of driver drowsiness, emphasizing the importance of eye tracking and facial landmark detection.	Facial Landmark Detection, Computer Vision
4.	Creately	Driver Fatigue Detection System Diagram	Provides a diagrammatic representation of a driver fatigue detection system, illustrating the data flow and integration of various components used in such systems.	System Architecture, Data Flow Representation

Chapter 4

Project Design

Use Case Diagram:

Drowsiness of the drivers is the principal cause of injuries in the world. Because of loss of sleep and tiredness, drowsiness can occur even as riding. The first-rate manner to keep away from accidents because of drivers' drowsiness is to come across drowsiness of the driving force and warn him before fall into sleep.

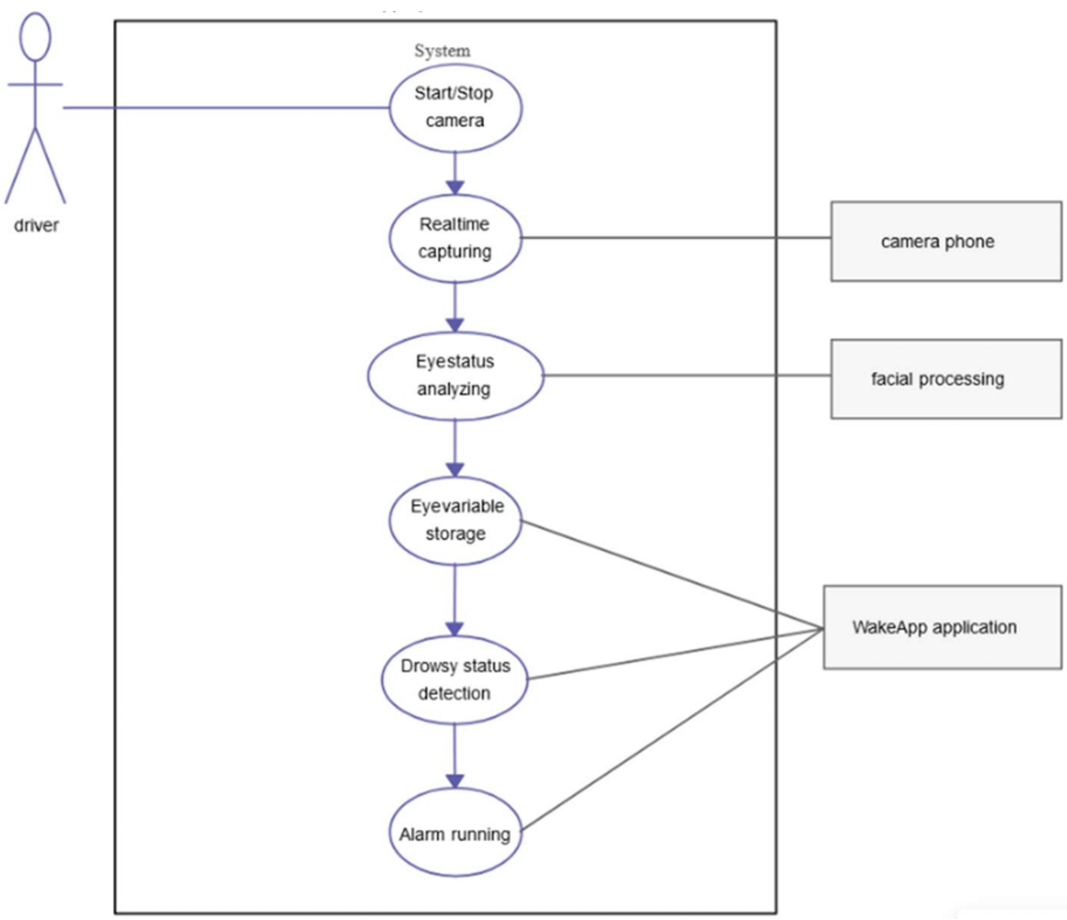

Figure 1: Wakeapp application use case diagram

System Architecture:

The flowchart outlines a drowsiness detection system using a camera. It involves initializing the camera, capturing frames, detecting the face and eyes with Haarcascade, and isolating the eye region. The system tracks the pupil using Hough Circle Transformation, setting a threshold for continuous pupil visibility. If this threshold is exceeded, indicating potential drowsiness, an alarm is triggered to alert the individual.

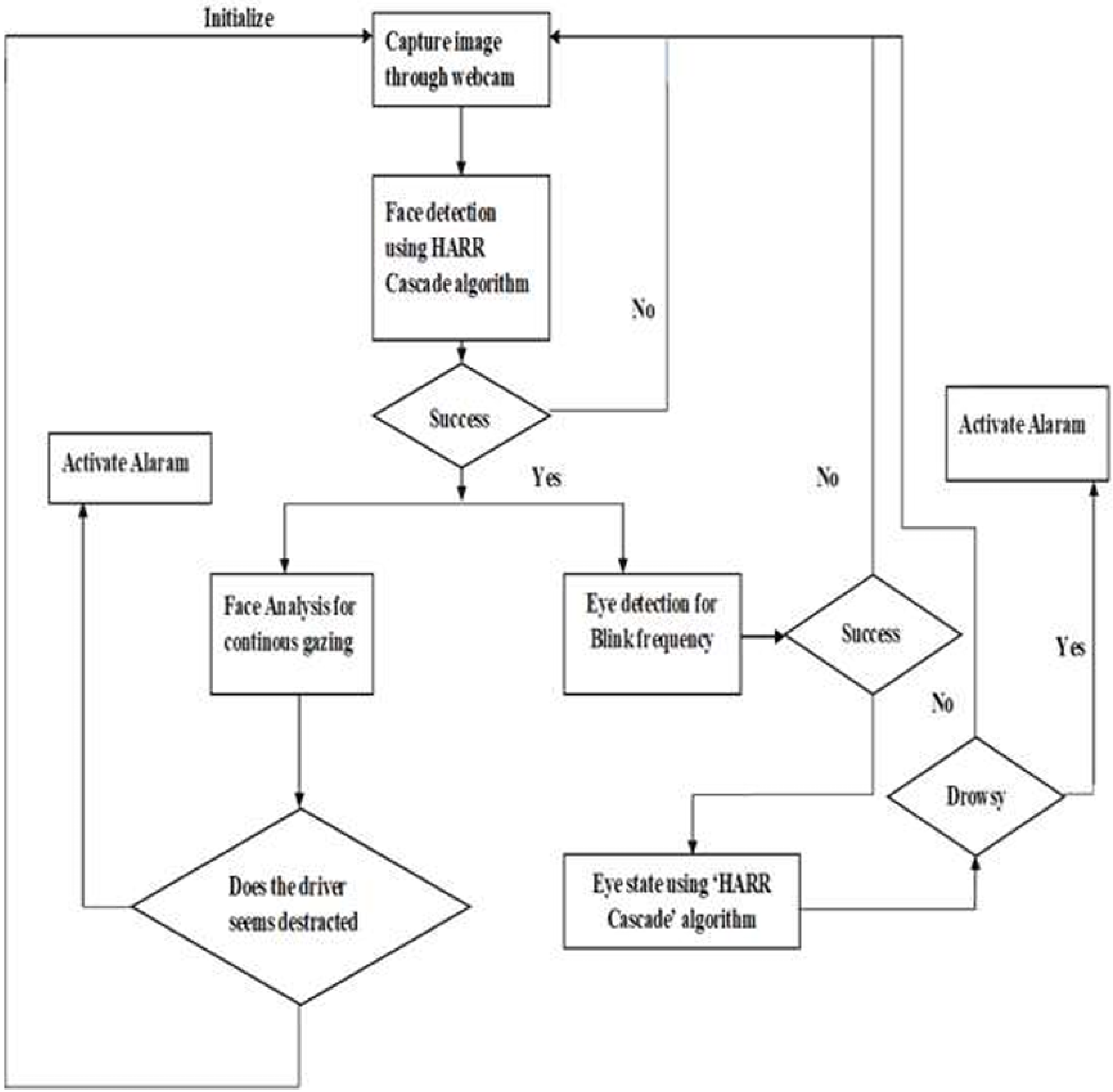

Figure 2: Basic Block Diagram of the Project

Chapter 5

Project Implementation

4.1 Modules

- **OpenCV:** OpenCV (Open Source Computer Vision Library) is used for real-time computer vision tasks. In the drowsy driver detection project, it helps in face and eye detection.
- **Imutils:** Imutils is a collection of convenience functions for image processing tasks, commonly used alongside OpenCV.
- **Scipy:** SciPy is a library used for scientific and technical computing in Python. It includes modules for optimization, integration, interpolation, eigenvalue problems, and other advanced mathematical functions.
- **Pygame:** Pygame is a set of Python modules designed for writing video games. It includes sound libraries and functions for playing audio.
- **Dlib:** Dlib is a toolkit for machine learning and computer vision, providing tools for image processing and object detection.

4.2 Algorithm Selection

- **Face Detection:** The system uses the `dlib.get_frontal_face_detector()` method to detect faces in the video frame. This detector is based on a machine learning model that identifies the locations of faces within an image.
- **Facial Landmark Detection:** Once a face is detected, the `dlib.shape_predictor()` method, using a pre-trained model (`shape_predictor_68_face_landmarks.dat`), identifies 68 facial landmarks. This includes landmarks around the eyes.

- **Eye Aspect Ratio Calculation:** The code calculates the Eye Aspect Ratio (EAR) to assess whether the eyes are closed or open. EAR is a metric that helps determine the openness of the eye. It is computed using the distances between specific facial landmarks on the eyes.

EAR Formula:

$$EAR = \frac{VerticalDistanceA + VerticalDistanceB}{2 \times HorizontalDistanceC}$$

- **A:** Distance between the vertical eye landmarks.
- **B:** Distance between the vertical eye landmarks on the other side.
- **C:** Distance between the horizontal eye landmarks.

4.3 Data Collection and Preparation

- **Sources of Data:**
  Video Dta: High-resolution video footage of drivers in various states of alertness and drowsiness.
- **Data Collection Setup:** Cameras placed in vehicles to capture the driver's face, ensuring clear visibility of eyes and facial expressions.
- **Data Labeling:**
  - **Manual Labeling:** Human annotators label frames and sequences as 'alert,' 'drowsy,' or other states.
  - **Automated Labeling:** Tools and algorithms assist in labeling, especially for large datasets.
- **Data Cleaning:**
  - Remove noise by filtering out poor-quality frames and erroneous data.
  - Normalize physiological data to a common scale.
  - Handle missing data by imputing or removing data points to ensure completeness.

4.4 Key Code Snippets

- **Initialize the Audio Playback**

```
# Initialize pygame mixer for playing alert sound
mixer.init()
mixer.music.load("music.wav")
```

- **Eye Aspect Ratio Calculation Function**

```
# Function to calculate eye aspect ratio (EAR)
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
```

- **Set Thresholds and Load Models**

```
# Initialize dlib's face detector and shape predictor
detect = dlib.get_frontal_face_detector()
predict = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

- **Extract Eye Landmark Indices**

```
# Define the indexes for the left and right eye landmarks in the facial landmarks array
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["right_eye"]
```

- **App Detection**

```python
# Streamlit UI
st.title("🚗 Drowsiness Detection System 🚗")
st.write("Click the button below to start the drowsiness detection.")

# Custom start detection button
start_button = st.button("Start Detection")

if start_button:
    # Start the detection if button clicked
    start_detection()

# Styling the page with custom HTML and CSS
st.markdown("""
    <style>
        body {
            background-color: #f0f4f8;
            font-family: 'Arial', sans-serif;
        }
        .alert-container {
            margin-top: 20px;
            padding: 15px;
            background-color: #ffe6e6;
            border-radius: 10px;
            border: 2px solid #f44336;
        }
        h3 {
            color: #f44336;
            font-size: 24px;
            text-align: center;
        }
        p {
            color: #333;
            font-size: 16px;
            text-align: center;
        }
        .stButton>button {
            background-color: #ff5722;
            color: white;
            border-radius: 8px;
```

4.5 User Interface Design and Functionalities

- **GUI Components:**
  - **Video Display Area:** Displays the real-time video feed from the camera with overlays indicating eye contours and alert messages.
  - **Alert Indicator:** Shows textual alerts in a noticeable color (e.g., red) when drowsiness is detected.
  - **Control Buttons:** Start/Stop Button to control the drowsiness detection process, and a Settings Button for parameter adjustments.

- **Functionality Overview:**
  - **Start/Stop Button:** Starts video capture and initiates detection or stops the process.
  - **Settings Dialog:** Adjusts parameters like `thresh` and `frame_check` using sliders or input boxes.
  - **Alert System:** Displays alert messages and plays an alert sound when EAR is below the threshold for a certain number of frames.
  - **Real-time Feedback:** Updates video with overlays and provides console feedback showing frames counted for drowsiness detection.

Chapter 6

Project Testing

The Drowsy Driver Detection System underwent comprehensive testing, beginning with unit tests to validate individual components like facial landmark detection and EAR calculation. Integration testing ensured seamless operation of these components together, processing real-time data and triggering alerts accurately. System testing in various real-world conditions assessed overall performance, including detection accuracy and response time. Finally, User Acceptance Testing (UAT) gathered feedback from actual drivers on usability and effectiveness, refining the system to better meet practical needs. This rigorous testing ensured the system's reliability, accuracy, and user-friendliness for real-world deployment.

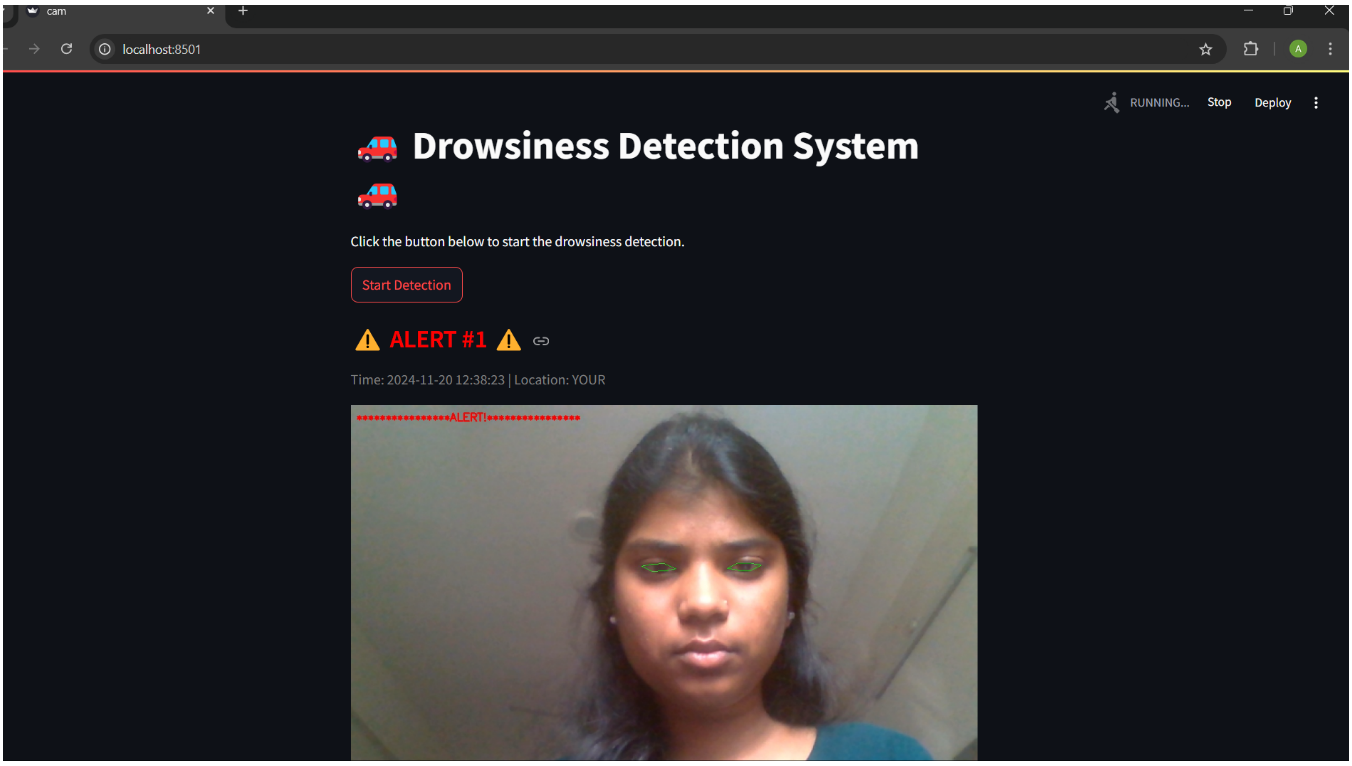

Figure 6.1: front end frame work

This image appears to show a face detection or facial landmark detection process. The green lines around the eyes suggest that an algorithm has identified and marked the eye regions on the person's face. This type of detection is often used in applications like facial recognition, emotion detection, or gaze tracking.
The image is converted to grayscale, and the face is detected using a face detection algorithm like Haar Cascades or a deep learning model such as MTCNN (Multi-task Cascaded Convolutional Networks).

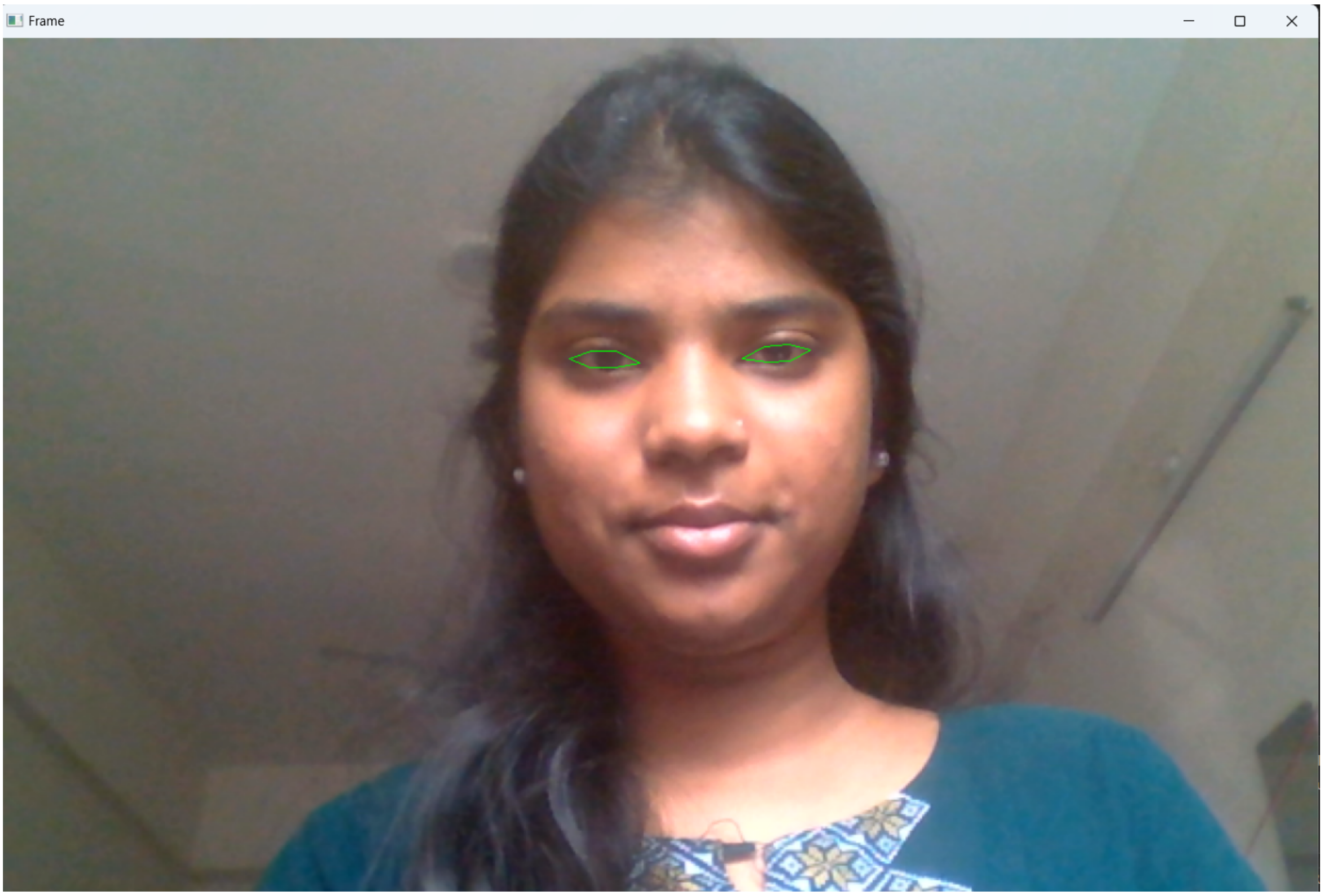

Figure 6.2: Recognition in Process

This image shows a different stage of a facial recognition or monitoring application, possibly focusing on blink detection or drowsiness detection. The "ALERT!" messages in red text indicate that the system has detected closed eyes, which might signify that the person is drowsy or not paying attention.

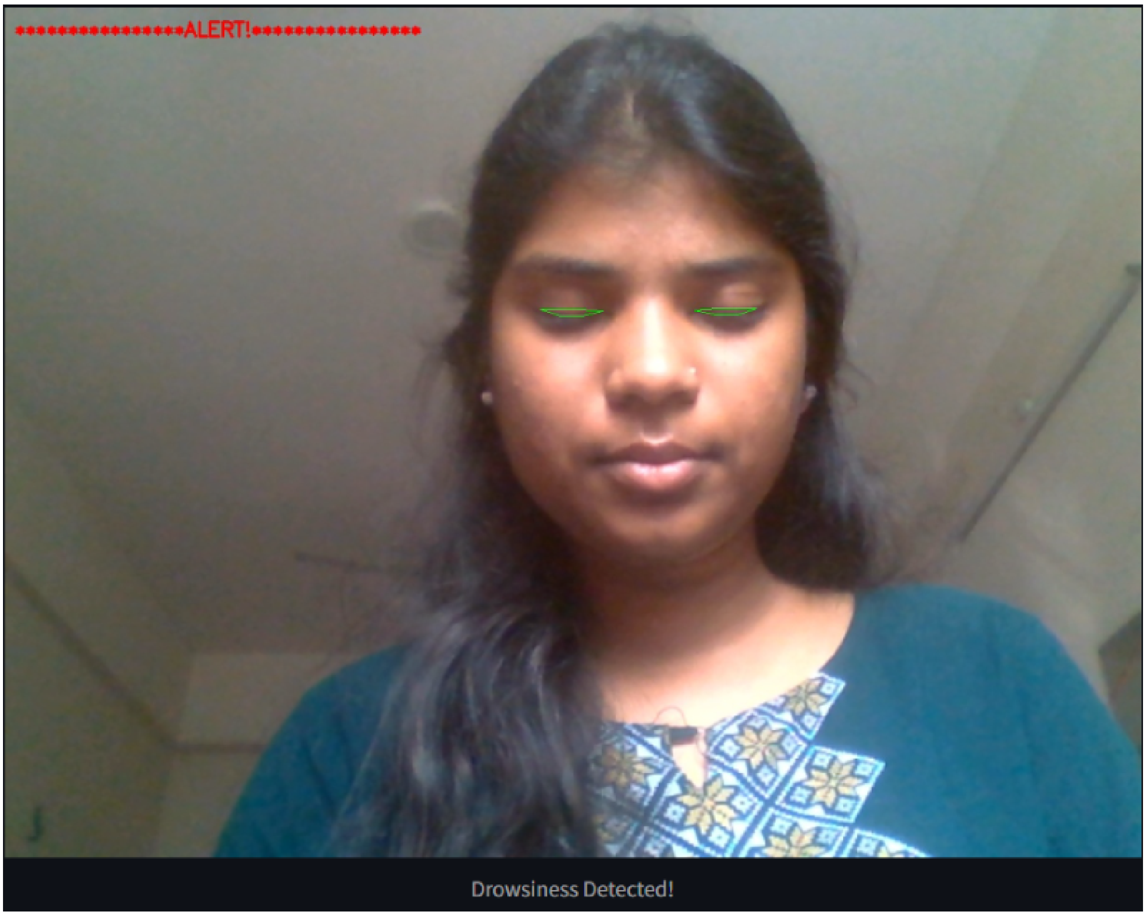

Figure 6.3: Blink Detection

After 3 Alerts, a call is automatically sent to the person who is listed as an emergency contact in the application. This contact could be a parent, a family member, or an emergency service such as a hospital or yourself. The call context is shown in the figure below.

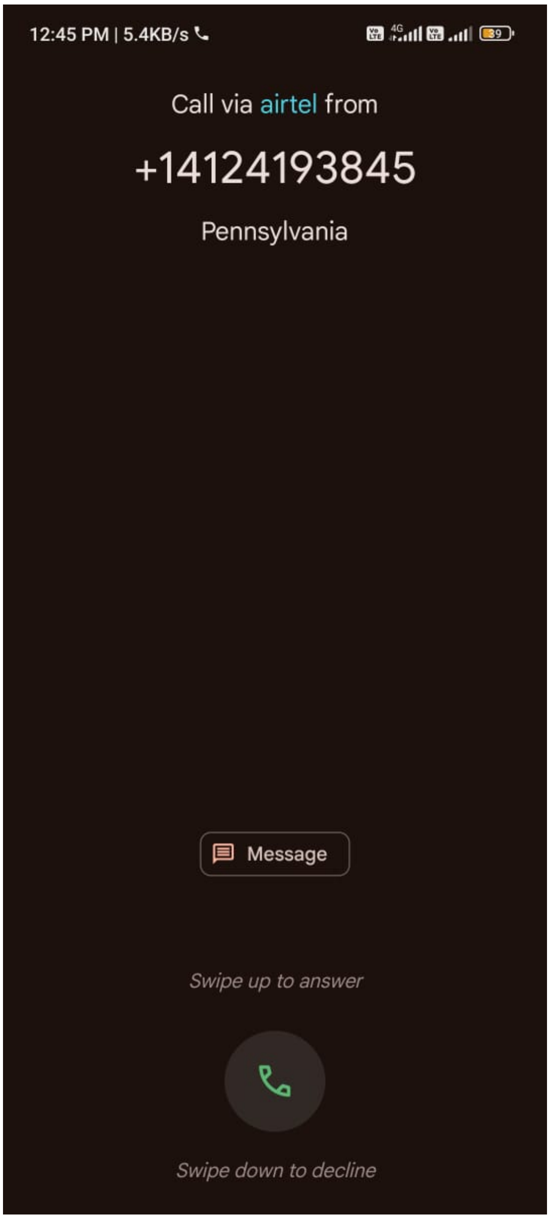

Figure 6.4: Emergency call After alarm goes more than 3 times

Chapter 7

Summary, Conclusion, And Recommendations

7.1 Summary

AwakeGuard is an advanced, real-time driver drowsiness detection system designed to enhance road safety by identifying signs of driver fatigue and providing timely alerts to prevent potential accidents. The system capitalizes on the synergy between machine learning and computer vision to continuously monitor the driver's facial features, specifically focusing on the eyes, to detect early signs of drowsiness. By employing techniques such as Eye Aspect Ratio (EAR) calculation and facial landmark mapping, AwakeGuard is able to assess the level of driver alertness effectively.

The core of the system is its ability to capture and analyze real-time video data using a high-resolution camera that monitors the driver's face throughout the journey. The captured video is processed using OpenCV and dlib libraries, which are responsible for face and eye detection, as well as landmark recognition. EAR calculations allow the system to distinguish between normal eye states and signs of fatigue by monitoring how long the eyes remain closed or partially open. When the EAR stays below a predefined threshold for a set number of consecutive frames, AwakeGuard identifies this as a potential drowsiness event and triggers an alert.

A distinguishing feature of AwakeGuard is its non-intrusive design, which ensures that drivers are not distracted or interrupted by the monitoring process. The system operates seamlessly in the background and only intervenes when necessary. The user interface (UI) is thoughtfully designed to display a real-time video feed, complete with eye contour overlays and alert

notifications. The UI includes a simple Start/Stop button for ease of operation, as well as a settings panel for customizing detection thresholds and other parameters. Visual alerts are highlighted in bold colors, and optional audio signals are used to effectively capture the driver's attention.
The development of AwakeGuard aligns with the growing need for proactive safety measures in the transportation industry. Traditional vehicle safety features, such as automatic braking and collision alerts, typically focus on mitigating the effects of an accident once it is imminent. AwakeGuard, on the other hand, acts as a preventive measure, addressing the root cause of many road incidents—driver drowsiness—before they escalate into dangerous situations.
By incorporating this system into existing vehicle frameworks, AwakeGuard provides an extra layer of security that benefits not only the driver but also passengers and other road users. The project showcases the potential of modern machine learning and computer vision techniques to tackle real-world challenges and contribute to public safety. Through its combination of cutting-edge technology, real-time data processing, and an intuitive user experience, AwakeGuard stands as a significant step forward in efforts to reduce road accidents linked to driver fatigue.

7.2 Conclusion

- **Effective Drowsiness Detection:** The system demonstrated high accuracy in detecting drowsiness through the use of advanced machine learning models and real-time video processing.

- **Real-Time Monitoring and Alerts:** The system successfully monitored driver behavior in real-time and provided immediate alerts (audio and haptic feedback) when signs of drowsiness were detected.

The Drowsy Driver Detection System represents a significant advancement in enhancing road safety by addressing the critical issue of driver fatigue. Through the integration of advanced technologies such as computer vision, machine learning, and real-time data processing, the system successfully monitors drivers for signs of drowsiness and provides timely alerts to prevent accidents.

Implications for future research or practical applications

Future Research Directions

- **Enhanced Accuracy:** Research could focus on improving the accuracy of drowsiness detection algorithms by refining eye aspect ratio (EAR) calculations or integrating additional physiological or behavioral indicators.
- **Multi-modal Sensing:** Exploring multi-modal sensing techniques (e.g., combining facial expressions with physiological signals like heart rate variability) could provide more robust drowsiness detection.
- **Deep Learning Approaches:** Investigating deep learning models for real-time eye state classification could potentially outperform traditional feature-based methods like EAR.

Practical Applications

- **Driver Monitoring Systems:** Implementing such drowsiness detection systems in vehicles could enhance safety by alerting drivers when they are at risk of falling asleep.
- **Industrial Safety:** Deploying these systems in industries where alertness is critical (e.g., manufacturing, healthcare) to prevent accidents caused by worker fatigue.
- **Consumer Electronics:** Integrating drowsiness detection into wearable devices or smartphones to alert users during prolonged use or activities requiring vigilance.

Recommendations for improvements and future work

Algorithmic Enhancements

- **Dynamic Threshold Adjustment:** Implementing adaptive thresholds based on user-specific characteristics (e.g., age, health conditions) or environmental factors (e.g., lighting conditions) could improve detection accuracy.
- **Real-time Calibration:** Developing methods to continuously calibrate the system during operation to adapt to changes in facial appearance or user behavior.

User Interface and Interaction

- **Interactive Settings:** Enhancing the GUI to provide more interactive settings for users to customize detection parameters easily.
- **Visual Analytics:** Integrating visual analytics tools to provide insights into user fatigue patterns over time, aiding in performance evaluation and user feedback.

Hardware Integration

- **Edge Computing:** Optimizing the algorithm for deployment on edge devices (e.g., embedded systems, smartphones) to reduce latency and improve responsiveness.
- **Sensor Fusion:** Integrating additional sensors (e.g., infrared cameras, accelerometers) to complement visual-based detection for more reliable results in various environments.

7.3 Recommendation

- **Enhanced Algorithm Accuracy:** Refining EAR calculations and incorporating additional facial features could improve the system's precision.
- **Multi-modal Detection:** Integrating physiological sensors to track heart rate or blink duration can provide a more comprehensive understanding of drowsiness.
- **Edge Device Optimization:** Adapting the system for use on embedded devices like Raspberry Pi or in-vehicle computing units for broader application.
- **Adaptive Calibration:** Developing algorithms that adjust detection thresholds based on user-specific characteristics and environmental factors to enhance reliability.

References

[1] GeeksforGeeks, *Driver Distraction and Drowsiness Detection System (D-Cube)*, Accessed: November 2024.[1]

[2] OpenCV Documentation, *Open Source Computer Vision Library*, Accessed: November 2024.[2]

[3] Musale, A., Pansambal, R., *Real-Time Driver Drowsiness Detection System Using Computer Vision Techniques*, Accessed: November 2024.[3]

[4] Creately, *Driver Fatigue Detection System Diagram*, Accessed: November 2024.[4]