Vision-Based Real-Time Drowsiness Detection with OpenCV and Dlib

A MINOR PROJECT REPORT

Submitted by

Vidhya V - RA2211003020335
Dhanarajan K - RA2211003020347
Gajasri Saravanan - RA2211003020370

Under the guidance of

Dr. J. Faritha Banu
(Associate Professor, Department of Computer Science and Engineering)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM, CHENNAI -600089

APRIL 2025

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University U/S 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report titled **“Vision-Based Real-Time Drowsiness Detection with OpenCV and Dlib”** is the bona-fide work of **Vidhya V [REG NO: RA2211003020335], Dhanarajan K [REG NO: RA2211003020347], Gajasri Saravanan [REG NO: RA2211003020370]** who carried out the minor project work under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an occasion on this or any other candidate.

SIGNATURE	SIGNATURE
Dr. J. Faritha Banu	**Dr. J. Sutha M.E. Ph.D.**
Associate Professor	**Professor and Head**
Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai.	Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai.

Submitted for the minor project viva-voce held on ___________ at SRM Institute of Science and Technology, Ramapuram, Chennai -600089.

INTERNAL EXAMINER 1 **INTERNAL EXAMINER 2**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM, CHENNAI - 89

DECLARATION

We hereby declare that the entire work contained in this minor project report titled "**Vision-Based Real-Time Drowsiness Detection with OpenCV and Dlib**" has been carried out by **Vidhya V [REG NO: RA2211003020335], Dhanarajan K [REG NO: RA2211003020347], Gajasri Saravanan [REG NO: RA221100302070]** at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089, under the guidance of **Dr. J. Faritha Banu, Associate Professor,** Department of Computer Science and Engineering.

Place: Chennai

Date:

Signature: **Vidhya V**

Dhanarajan K

Gajasri Saravanan

ACKNOWLEDGEMENT

We place on record our deep sense of gratitude and very sincere thanks to our lionized Chairman **Dr. R. SHIVAKUMAR** for providing us with the requisite infrastructure throughout the course.

We take the opportunity to extend our deep sense of gratitude and hearty and sincere thanks to our **Dean, Dr. M. SAKTHI GANESH. Ph.D.** for maneuvering us into accomplishing the project.

We take the privilege to extend our hearty and sincere gratitude to the Professor and Chairperson of the Department**, Dr. K. RAJA M.E. PhD.** for his suggestions, support and encouragement towards the completion of the project with perfection.

We take the privilege to extend our hearty and sincere gratitude to the Professor and Head of the Department**, Dr. J. SUTHA M.E. Ph.D.** for his suggestions, support and encouragement towards the completion of the project with perfection.

We express our hearty and sincere thanks to our **guide Dr. J. FARITHA BANU M.E. Ph.D. Associate Professor,** Computer Science and Engineering Department for her encouragement, consecutive criticism and constant guidance throughout this project work.

Our thanks to the teaching and non-teaching staff of the Computer Science and Engineering Department of SRM Institute of Science and Technology, Ramapuram Campus, for providing necessary resources for our project.

ABSTRACT

Driver drowsiness is a critical factor contributing to a large number of road accidents globally. Fatigue reduces a driver’s alertness and response time, which can lead to severe consequences, especially during long-distance travel or night-time driving. This project presents a real-time drowsiness detection system that aims to prevent accidents by continuously monitoring the driver's eye movements and issuing timely alerts when signs of drowsiness are detected.

The system uses computer vision and machine learning techniques to analyze facial features in real time. A webcam is used to capture live video of the driver’s face. Through the use of Python, OpenCV, and the Dlib library, facial landmarks are identified, and the eye region is extracted from each video frame. The Eye Aspect Ratio (EAR) is then calculated to determine whether the eyes are open or closed. A prolonged decrease in the EAR value indicates eye closure, which is a strong indicator of drowsiness. If the system detects that the eyes have been closed for a specific number of consecutive frames, it triggers an alarm to alert the driver.

The approach is non-intrusive, requiring no physical contact or wearable sensors, making it comfortable and practical for real-world use. It relies solely on visual cues and can operate efficiently in real time with minimal hardware requirements. The system architecture includes modules for image capture, face and eye detection, eye state analysis, and alert generation.

This project demonstrates a cost-effective and efficient method to detect driver fatigue using image processing. By issuing alerts at the right moment, it can significantly reduce the chances of accidents caused by drowsy driving. Future enhancements can focus on improving detection in low-light conditions, handling facial obstructions such as sunglasses, and integrating the system with vehicle automation features for improved road safety.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
INTRODUCTION

1.1 INTRODUCTION

In recent years, road safety has emerged as a significant public concern due to the increasing number of traffic accidents caused by driver fatigue and drowsiness. According to global traffic studies, drowsiness impairs a driver's ability to make quick decisions, slows reaction time, and often leads to loss of vehicle control. Long hours of continuous driving, monotonous road conditions, and insufficient rest are some of the key factors that contribute to drowsy driving. Recognizing these risks, researchers and developers have started exploring technology-based solutions to detect drowsiness in real time and prevent potential accidents.

This project introduces a Real Time Drowsiness Detection System, which focuses on monitoring the driver's eye behavior using computer vision techniques. The primary goal is to detect early signs of fatigue by analyzing the state of the eyes — whether they are open or closed — and alerting the driver through an alarm system if signs of drowsiness are detected. This solution is based on non-intrusive methods, using a webcam to capture live video and applying image processing and machine learning algorithms to analyze the driver's facial features.

The system uses the Eye Aspect Ratio (EAR), which is calculated from facial landmarks detected using the Dlib library, to determine whether the eyes are in a closed state. If the EAR remains below a threshold for a specific duration, it indicates a drowsy condition, and an alert is generated.

Unlike invasive systems that rely on physiological sensors or wearable devices, this model operates solely through visual monitoring, making it more practical for real-world applications. It is designed to work efficiently with minimal hardware and can be deployed in a variety of vehicles.

Overall, this project aims to enhance driving safety by detecting drowsiness in its early stages and preventing accidents through timely warnings. As a low-cost, effective, and scalable solution, it offers great potential for integration into modern intelligent transport systems.

1.2 PROBLEM STATEMENT

Drowsiness while driving is a major cause of road accidents, often resulting in serious injuries and fatalities. Traditional methods of detecting fatigue, such as self-reporting or physical sensors, are either unreliable or intrusive, making them impractical for continuous monitoring. There is a need for a real-time, non-intrusive system that can accurately detect signs of driver fatigue and issue timely alerts to prevent accidents. The challenge lies in developing a solution that can function effectively under varying lighting conditions, facial orientations, and without requiring drivers to wear any special equipment. This project aims to address this issue by using computer vision techniques to detect drowsiness based on eye movements, providing a practical and efficient safety mechanism for drivers.

1.3 AIM OF THE PROJECT

The primary aim of this project is to develop a Real Time Drowsiness Detection System that can effectively monitor and identify signs of fatigue in drivers to prevent road accidents. The system is designed to function using non-intrusive computer vision techniques that analyze eye movements to determine the alertness level of the driver. By continuously observing the open or closed state of the eyes through a webcam and calculating the Eye Aspect Ratio (EAR), the system can accurately detect prolonged eye closure, which is a strong indicator of drowsiness.

The project focuses on building an automated solution that provides real-time feedback in the form of alerts, such as an alarm sound, whenever a drowsy state is detected. This warning aims to restore the driver’s attention and avoid potential collisions or mishaps. The goal is to design a lightweight and cost-effective model that can run efficiently on standard computing hardware without the need for complex setups or wearable sensors.

In addition, the system aims to be adaptable and reliable under different environmental conditions, including slight facial obstructions and moderate lighting changes. Ultimately, the project aspires to enhance road safety by providing a scalable solution that can be integrated into various vehicles to minimize the risks associated with fatigue-related driving.

1.4 PROJECT DOMAIN

The project titled “Real-Time Drowsiness Detection System” falls under the domain of Computer Vision and Artificial Intelligence, with a strong foundation in Image Processing and Machine Learning. It leverages advanced techniques in facial recognition and pattern analysis to identify signs of fatigue and drowsiness in real-time, specifically targeting applications in road safety and smart vehicle systems.

Computer Vision is a field of Artificial Intelligence that trains computers to interpret and understand the visual world. In this project, we use OpenCV and Dlib libraries to process video inputs from a webcam, detect the face and eye region of the driver, and analyze blinking patterns and eye closure durations to detect drowsiness. This is achieved by tracking facial landmarks and calculating the Eye Aspect Ratio (EAR) to determine if the eyes are open or closed.

Machine Learning is also an integral part of this project. A trained model helps in classifying eye states and triggering alerts when abnormal blinking or prolonged eye closure is detected. The system is non-intrusive and can function efficiently in real-world conditions without requiring any physical contact or wearables, making it user-friendly and practical.

The domain of this project addresses critical safety issues, particularly in the transportation industry, where driver fatigue is a major cause of road accidents. By combining Computer Vision with real-time data processing, this system represents a significant advancement in intelligent driver assistance technologies.

1.5 SCOPE OF PROJECT

The Real-Time Drowsiness Detection System is a vision-based solution designed to detect signs of driver fatigue and issue timely alerts to prevent road accidents. This project falls within the scope of Computer Vision, Artificial Intelligence, and Image Processing, offering both practical and technological contributions.

1. **Real-Time Monitoring**: The system is capable of capturing live video feed from a webcam installed on the vehicle dashboard. It processes each video frame in real-time to detect the driver's facial landmarks, especially focusing on the eyes.
2. **Non-Intrusive Fatigue Detection**: Unlike traditional EEG or heart rate-based systems, this model does not require any physical contact with the driver. This ensures greater comfort and usability for everyday use, especially in commercial transport systems.
3. **Eye State Analysis**: The software calculates the Eye Aspect Ratio (EAR) using facial landmark detection. A low EAR value for a continuous number of frames is classified as a sign of drowsiness. The system raises an alarm if the eyes remain closed beyond a set threshold.
4. **Driver Alert Mechanism**: On detecting drowsiness, the system triggers an audible alert to wake up the driver. This immediate feedback loop helps in reducing the chances of accidents due to micro-sleeps.
5. **Scalability and Future Expansion**: The application can be scaled further to include head posture tracking and mouth yawning detection for enhanced accuracy. Future integration with vehicle control systems could automate safety responses, like slowing down the vehicle.
6. **Applicability**: This project can be applied in commercial transportation, long-haul trucking, and personal driving assistance. It also holds potential for integration with ADAS (Advanced Driver Assistance Systems).
7. **Feedback and Quality Control**: User feedback after sessions will be used to adjust token rewards and helper reputation, creating a self-regulating ecosystem where quality is maintained without centralized intervention.

Chapter 2
LITERATURE SURVEY

Drowsiness detection in real-time has been an area of growing interest in both academic and industrial research due to its significant impact on road safety. Numerous studies have shown that fatigue is a leading cause of road accidents, especially in long-distance travel and commercial transport. To combat this issue, researchers have proposed various methods for detecting driver drowsiness using physiological signals, behavioral cues, and machine learning models. The goal is to create systems that are both accurate and non-intrusive, capable of functioning in real-time under varying environmental conditions.

2.1 PREVIOUS WORK ON VISION-BASED DROWSINESS DETECTION

2.1.1 Research Using Eye Aspect Ratio (EAR)

Researchers have explored EAR-based detection for identifying eye closure over time. This method uses facial landmarks to compute the vertical and horizontal distances of the eyes, making it effective for blink detection and drowsiness analysis. A low EAR for a sustained period suggests sleep onset or fatigue.

The strength of this approach lies in its simplicity and real-time applicability. It does not require advanced computational resources and performs well in controlled environments. However, it can struggle in low-light conditions or when the subject is wearing glasses.

2.1.2 Yawning Detection Techniques

Yawning is another key symptom of fatigue, often paired with slow blinking. Studies using lip movement and mouth aspect ratio have been proposed to detect yawning using video sequences. Researchers utilized image thresholding and motion tracking to detect wide mouth openings over time.

While yawning detection provides an extra layer of verification for fatigue, it is sensitive to speech-related movements and lighting variations. Combining it with eye analysis can increase the overall accuracy of fatigue detection systems.

2.2 MACHINE LEARNING AND DEEP LEARNING IN DROWSINESS DETECTION

2.2.1 Support Vector Machines (SVM) and Classical ML

Several projects have used SVMs to classify eye states (open/closed) based on extracted features like pixel intensity, HOG (Histogram of Oriented Gradients), or landmark ratios. These models are easy to train, require limited data, and perform relatively well in binary classification.

Despite their ease of implementation, classical ML techniques often struggle with generalization across diverse subjects and backgrounds. Their performance degrades in noisy environments unless strong preprocessing techniques are applied.

2.2.2 Convolutional Neural Networks (CNN)

CNNs have become popular due to their ability to learn spatial hierarchies from raw image data. CNN-based models can automatically extract eye patterns, making them less sensitive to individual variations.

The downside is that CNNs require large annotated datasets and considerable computing power. However, they offer better accuracy and robustness than traditional approaches, especially in varying lighting and real-world settings.

2.3 HARDWARE-ORIENTED SOLUTIONS

2.3.1 EEG and Biometric Sensors

Some literature suggests the use of EEG headbands to measure brain wave patterns that indicate drowsiness. These wearable systems directly measure mental alertness levels, offering high accuracy.

However, EEG-based systems are intrusive, expensive, and uncomfortable for drivers. Their complexity also limits widespread adoption in everyday vehicles.

2.3.2 IR Sensors and Eye Tracking Glasses

Infrared sensors have been implemented for detecting eye movements even in low-light or nighttime conditions. Commercial systems such as Tobii and Smart Eye use IR cameras to track pupil movements precisely.

Although effective, these solutions are often costly and require precise calibration. Their deployment is typically seen in high-end automobiles or research prototypes rather than budget consumer applications.

2.4 INTEGRATION WITH VEHICLE SAFETY SYSTEMS

2.4.1 ADAS (Advanced Driver Assistance Systems)

Recent studies have focused on integrating drowsiness detection into ADAS frameworks, which already include features like lane detection, automatic braking, and collision warnings. Combining these systems improves overall safety by allowing cross-sensor validation.

Integration with ADAS, however, requires compliance with vehicle safety standards, and compatibility with car control modules, which adds complexity to implementation and increases cost.

2.4.2 Alerting and Response Mechanisms

Studies emphasize the importance of driver feedback. Alert mechanisms include audio alarms, vibrating seats, or dashboard notifications. Some advanced setups are designed to gradually increase alert intensity based on the duration of drowsiness.

These mechanisms are simple yet effective. However, improper calibration can lead to false positives or alarm fatigue, which may annoy drivers instead of helping them.

2.5 REAL-WORLD APPLICATION AND LIMITATION

2.5.1 Commercial Applications in Transport

Drowsiness detection systems have been deployed in commercial trucks, taxis, and fleet vehicles. In public transport, such systems help ensure passenger safety during long journeys.

While beneficial, large-scale implementation requires continuous monitoring, regular system calibration, and training for drivers, which adds operational overhead.

2.4.2 Challenges in Natural Driving Conditions

Real-world environments present challenges like lighting variations, driver accessories (glasses, hats), and face occlusion. Many academic models perform well in lab conditions but struggle in dynamic, real-world driving environments.

Researchers are now focusing on building more robust models using diverse datasets, including variations in age, gender, and ethnicity, to ensure universal reliability. These improvements are crucial for developing commercially viable and scalable systems.

Chapter 3
PROJECT DESCRIPTION

The Real-Time Drowsiness Detection System is a computer vision-based application developed using Python, OpenCV, and Dlib. Its primary purpose is to monitor a driver's eye movements in real time to detect signs of drowsiness or fatigue. The system captures video input through a webcam, processes each frame to locate the driver's face, and then focuses on the eye regions. By calculating the Eye Aspect Ratio (EAR) using facial landmark points, it determines whether the eyes are open or closed.

If the eyes remain closed for a specified number of consecutive frames, it is identified as a drowsy state, and an alarm is triggered to alert the driver. This non-intrusive and cost-effective solution addresses a major safety concern—driver fatigue, which is a leading cause of road accidents. The system runs efficiently in real-time and is suitable for integration into both personal and commercial vehicles.

Beyond real-time detection, the project can be further extended with features like head pose estimation, face orientation tracking, and integration with mobile apps to enhance usability. The simplicity of hardware requirements, combined with powerful image processing algorithms, makes this system a reliable tool for reducing drowsiness-related incidents on the road.

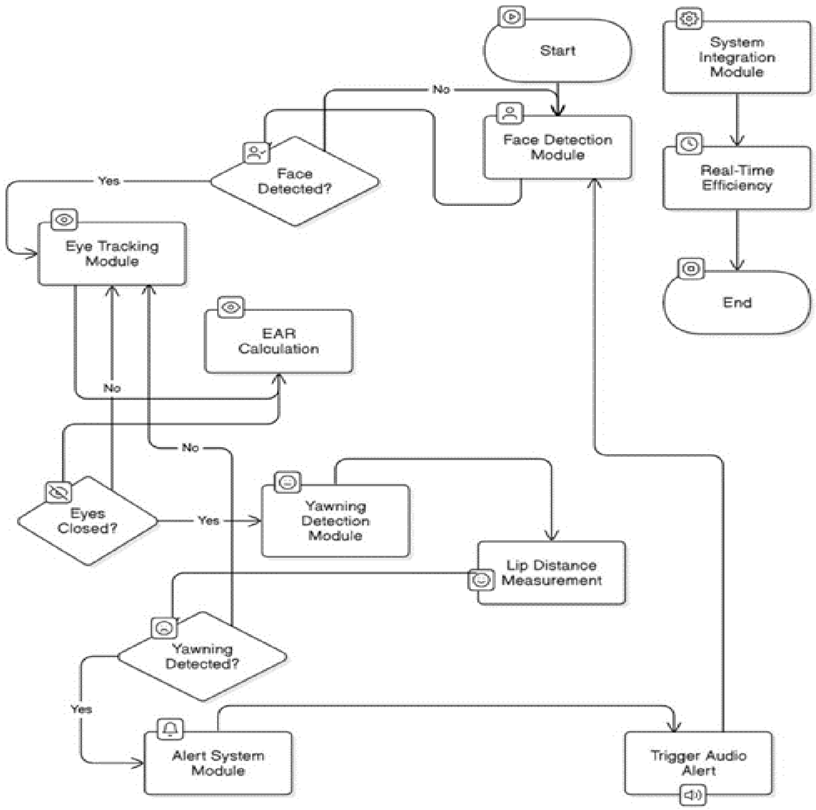

Fig. 3.1 Flow Diagram for Drowsiness Detection System

3.1 PURPOSE AND MOTIVATION

The primary purpose of this project is to address the rising number of road accidents caused by driver fatigue. Drowsy driving is a critical issue, especially in long-haul transportation and night travel. This system aims to provide a smart, real-time safety mechanism that monitors driver alertness using facial cues—without interfering with the driver's comfort or focus.

The motivation stems from the need for non-invasive, affordable solutions that can be deployed in commercial fleets and personal vehicles. By detecting fatigue early, this system can help prevent life-threatening incidents and promote safer driving environments.

3.2 CORE TECHNOLOGY STACK

This project is developed using Python, along with key libraries like OpenCV for image processing and Dlib for facial landmark detection. The system utilizes a webcam to continuously capture the driver’s face and analyze eye movements in real-time.

The algorithm calculates the Eye Aspect Ratio (EAR) to determine whether the eyes are open or closed. These calculations are lightweight, making the system suitable for real-time deployment even on lower-end hardware. Threading and efficient frame handling are used to maintain smooth performance without lag.

3.3 SYSTEM WORKFLOW

The system begins by capturing live video frames from a webcam. It first detects the face using Haar Cascade classifiers and then identifies key facial landmarks, particularly around the eyes.

The eye landmarks are used to calculate the EAR, which helps determine eye closure over time. If the system detects that eyes are closed for several consecutive frames (e.g., more than 5 seconds), it classifies the driver as drowsy and triggers an alarm. This loop runs continuously as long as the system is active.

3.4 FACIAL LANDMARK DETECTION

Facial landmark detection plays a central role in the system’s accuracy. The Dlib library provides a pre-trained 68-point model that detects critical facial features such as eyes, nose, lips, and jawline.

For this project, the landmarks around the eyes (points 36–48) are used to compute the EAR. This approach is robust under varying head poses and minor lighting changes. The landmarks help in isolating eye regions even when the face is slightly tilted, increasing detection reliability.

3.5 EYE ASPECT RATIO(EAR) AND BLINK ANALYSIS

The Eye Aspect Ratio is a key mathematical feature calculated from the distances between eye landmarks. When eyes are open, the vertical distances are relatively large compared to the horizontal ones. As the eyes close, the vertical distances decrease, lowering the EAR.
A threshold (typically 0.3) is set, below which the system considers the eyes to be closed. When the EAR remains below this threshold across several frames, it indicates drowsiness.

3.6 ALERT MECHANISM

Upon detecting prolonged eye closure, the system instantly triggers an alert. This alert can be an audio signal such as a buzzer or beep, which helps snap the driver out of their drowsy state.
The alert system can be extended to include visual notifications or even connected to external devices like vehicle braking systems. In future versions, integration with mobile notifications or fleet management dashboards is also possible.

3.7 HARDWARE AND SOFTWARE REQUIREMENTS

The system is designed to run on basic hardware, making it cost-effective. It requires a standard webcam, a system with at least 512MB RAM, and a modern operating system (Windows 7 or above). Software requirements include Python 3.x, OpenCV, Dlib, NumPy, and SciPy. The lightweight design ensures the system remains responsive even on entry-level machines, making it suitable for integration into embedded systems or vehicle infotainment platforms.

3.8 SCALABILITY AND FUTURE IMPROVEMENTS

While the current system is limited to eye-tracking, future enhancements could include head pose estimation, yawning detection, and mouth movement analysis for more accurate fatigue detection. Additionally, such as infrared-based eye tracking for low-light conditions or machine learning-based classification models can make the system more adaptable. Integration with IoT, mobile apps, or vehicle telematics could turn this into a complete driver monitoring ecosystem.

Chapter 4
PROPOSED SYSTEM

The proposed system is designed to detect driver drowsiness in real-time using computer vision techniques. It uses a webcam to continuously capture facial video, from which eye landmarks are extracted using Dlib's facial recognition model. The system calculates the Eye Aspect Ratio (EAR) to determine eye closure. If the EAR falls below a threshold for a sustained period, it triggers an audio alert. The architecture is lightweight, efficient, and non-intrusive. Challenges include poor lighting and obstruction due to eyewear, but its advantages—low cost, ease of integration, and real-time performance—make it highly scalable for transportation safety applications.

4.1 GENERAL ARCHITECTURE

The system consists of a webcam for capturing live video, a Python-based processing module using OpenCV and Dlib for facial landmark detection, and an alert unit that notifies the driver upon drowsiness detection. The processing follows a pipeline: Face Detection → Eye Landmark Extraction → EAR Calculation → Alert Trigger.

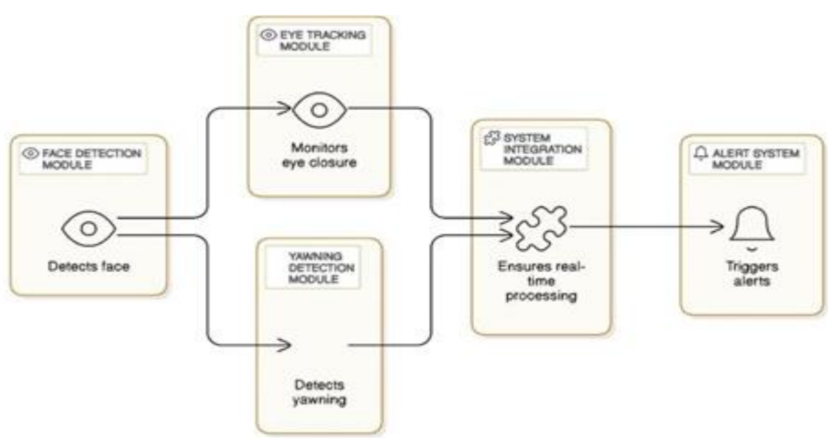

Fig. 4.1 Proposed Driver Drowsiness Detection System.

4.1.1 Real-Time Video Processing

Frames are captured continuously and processed instantly. The system performs EAR (Eye Aspect Ratio) calculations in real-time, ensuring prompt alerts without noticeable delays.

4.1.2 Facial Landmark-Based Analysis

Using Dlib’s 68-point facial model, the system accurately detects eyes, mouth, and nose. This helps isolate eye regions regardless of head tilt, making the system robust under different facial orientations.

4.1.3 Alert Mechanism

A buzzer or audio signal is activated when eye closure exceeds a threshold, warning the driver to regain attention.

4.2 CHALLENGES

Challenges include performance under low lighting, drivers wearing spectacles or masks, varying head angles, and motion blur at higher vehicle speeds.

4.3ADVANTAGES

- The system is non-intrusive, requiring only a webcam, which ensures driver comfort without the need for wearable sensors.
- Real-time detection allows for immediate alert generation, helping prevent accidents caused by micro-sleeps or prolonged eye closure.
- It is cost-effective and affordable, using open-source libraries like OpenCV and Dlib, and low-cost hardware such as a standard webcam.

4.4 SYSTEM SCALABILITY

This system can be extended with additional features like yawning detection, head pose tracking, and cloud connectivity for fleet monitoring.

4.5 PRACTICAL APPLICATIONS

Useful in long-distance transport, commercial logistics, public buses, and personal vehicles—anywhere driver fatigue is a safety concern.

- It can be installed in personal vehicles to monitor drivers during long-distance or night-time travel, reducing the risk of fatigue-related accidents.
- In commercial trucking and logistics fleets, the system helps ensure driver safety by providing continuous fatigue monitoring during extended shifts.
- Public transportation systems like buses and intercity coaches can use this technology to improve passenger safety by detecting when a driver is becoming inattentive.

Chapter 5

IMPLEMENTATION AND TESTING

The system is implemented using Python, OpenCV, and Dlib for facial detection and EAR calculation. It is tested with webcam input under various lighting and facial conditions to verify drowsiness detection accuracy and responsiveness. The system was implemented using Python as the programming language, with OpenCV for real-time video processing and Dlib for facial landmark detection. The Eye Aspect Ratio (EAR) was calculated for every frame to monitor eye closure. Testing was conducted using multiple individuals under different lighting conditions, head angles, and with/without spectacles

5.1 INPUT

The primary input is a live video stream captured through a webcam focused on the driver's face. This visual data is processed frame-by-frame to monitor eye movement and detect signs of fatigue.

5.1.1 Live Video Feed

The system relies on a continuous video stream captured using a webcam focused on the driver's face. This live feed serves as the primary input, allowing real-time monitoring of facial expressions and eye movement.

5.1.2 Facial Landmark Detection

From the captured video frames, the system identifies facial landmarks using a pre-trained model (Dlib's 68-point predictor). These landmarks help in isolating the eye region, which is critical for fatigue analysis.

5.1.3 Eye Region Extraction

The eye coordinates are extracted from facial landmarks in each frame. These regions are then used to calculate the Eye Aspect Ratio (EAR), which determines whether the eyes are open or closed.

5.1.4 Frame-by-Frame Analysis

Each video frame is processed individually to ensure precise and real-time tracking. This frame-wise approach allows the system to detect even short blinks or prolonged eye closures.

5.1.5 Environmental Factors

Lighting conditions, camera angle, and driver movement affect input quality. The system is designed to handle moderate variations, but extreme shadows or obstructions (e.g., sunglasses) can impact detection accuracy.

5.2 SOURCE CODE

The source code is written in Python and uses OpenCV for real-time video capture and Dlib for facial landmark detection. It loads a pre-trained 68-point facial landmark model to detect eyes and calculates the Eye Aspect Ratio (EAR) to determine eye openness. If the EAR remains below a threshold for several frames, an alarm sound is triggered using the playsound or winsound module. The code runs in a loop, processing each video frame continuously for real-time drowsiness detection and alert generation.

5.2.1 Importing our required Python packages:

detect_drowsiness.py

```
# import the necessary packages
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 playsound
import argparse
import imutils
import time
import dlib
import cv2
```

Sound Alarm

```
def sound_alarm(path):
# play an alarm sound
playsound.playsound(path)
```

eye_aspect_ratio function

```
def eye_aspect_ratio(eye): 21
# compute the euclidean distances between the two sets of
# vertical eye landmarks (x, y)-coordinates
A = dist.euclidean(eye[1], eye[5])
B = dist.euclidean(eye[2], eye[4])
# compute the euclidean distance between the horizontal
# eye landmark (x, y)-coordinates
C = dist.euclidean(eye[0], eye[3])
# compute the eye aspect ratio
ear = (A + B) / (2.0 * C)
# return the eye aspect ratio
return ear
```

Parsing command Line Argument

```
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--shape-predictor", required=True,
help="path to facial landmark predictor")
ap.add_argument("-a", "--alarm", type=str, default="",
help="path alarm .WAV file")
ap.add_argument("-w", "--webcam", type=int, default=0,
help="index of webcam on system")
args = vars(ap.parse_args())
```

Defining EYE_AR_THRESH

EYE_AR_THRESH = 0.3

EYE_AR_CONSEC_FRAMES = 48

COUNTER = 0

ALARM_ON = False 22

Facial landmark predictor

initialize dlib's face detector (HOG-based) and then create

the facial landmark predictor

print("[INFO] loading facial landmark predictor...")

detector = dlib.get_frontal_face_detector()

predictor = dlib.shape_predictor(args["shape_predictor"])

Extracting the eye regions

grab the indexes of the facial landmarks for the left and

right eye, respectively

(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]

(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

Instantiate VideoStrem

start the video stream thread

print("[INFO] starting video stream thread...")

vs = VideoStream(src=args["webcam"]).start()

time.sleep(1.0)

loop over frames from the video stream

while True:

frame = vs.read()

frame = imutils.resize(frame, width=450)

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

detect faces in the grayscale frame

rects = detector(gray, 0) 23

Facial landmark detection to localize each of the important regions of the face:

```
# loop over the face detections
for rect in rects:
# determine the facial landmarks for the face region, then
# convert the facial landmark (x, y)-coordinates to a NumPy
# array
shape = predictor(gray, rect)
shape = face_utils.shape_to_np(shape)
# extract the left and right eye coordinates, then use the
# coordinates to compute the eye aspect ratio for both
eyes
leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]
leftEAR = eye_aspect_ratio(leftEye)
rightEAR = eye_aspect_ratio(rightEye)
# average the eye aspect ratio together for both eyes
ear = (leftEAR + rightEAR) / 2.0
```

Visualize each of the eye regions

```
# compute the convex hull for the left and right eye, then
leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0),1)
cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0),1) 24
```

Check to see if the person in our video stream is starting to show symptoms of drowsiness:

```
# check to see if the eye aspect ratio is below the blink
# threshold, and if so, increment the blink frame counter
if ear < EYE_AR_THRESH:
```

```
COUNTER += 1
# if the eyes were closed for a sufficient number of
# then sound the alarm
if COUNTER >= EYE_AR_CONSEC_FRAMES:
# if the alarm is not on, turn it on
if not ALARM_ON:
ALARM_ON = True
# check to see if an alarm file was supplied,
# and if so, start a thread to have the alarm
# sound played in the background
if args["alarm"] != "":
t = Thread(target=sound_alarm,
args=(args["alarm"],))
t.deamon = True
t.start()
# draw an alarm on the frame
cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
# otherwise, the eye aspect ratio is not below the blink
# threshold, so reset the counter and alarm
else:
COUNTER = 0
ALARM_ON = False 25
```

Displaying the output frame:

```
# draw the computed eye aspect ratio on the frame to help
# with debugging and setting the correct eye aspect ratio
# thresholds and frame counters
cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
# show the frame
```

```
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
# if the `q` key was pressed, break from the loop
if key == ord("q"):
break
# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

5.3 OUTPUT

The output of the Real-Time Drowsiness Detection System is primarily a real-time visual and audio alert triggered when the driver shows signs of fatigue. Once the eyes remain closed beyond the defined threshold (EAR < 0.3 for 48 consecutive frames), the system generates an on-screen warning along with an audible alarm to alert the driver. Additionally, the Eye Aspect Ratio (EAR) is continuously displayed on the screen, allowing real-time monitoring. Eye contours are drawn on the video feed for visual clarity, confirming that facial landmarks have been successfully detected. The system provides fast, responsive output with minimal lag, ensuring immediate feedback to prevent accidents.

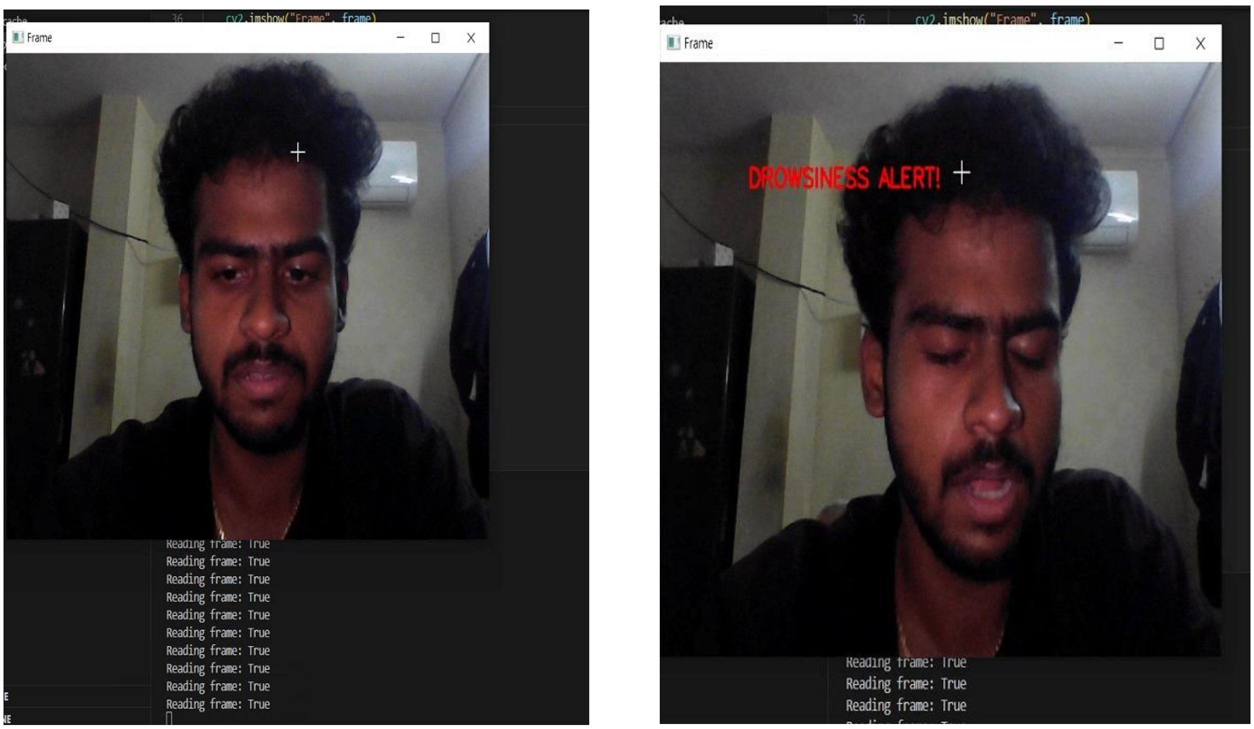

Fig. 5.1 Output

5.3.1 Real-Time Alert Mechanism

The most critical output feature of the Real-Time Drowsiness Detection System is its real-time alert mechanism. When the system detects prolonged eye closure indicating drowsiness, it triggers an instant alert to notify the driver. This immediate response is crucial in preventing road accidents, especially during long drives or late-night travel. The system combines visual and auditory cues to ensure the driver is aware of their fatigue state. The alert mechanism is lightweight and works seamlessly without interrupting the driver's concentration or requiring physical intervention.

- **Visual Alert:** A red-colored text ("DROWSINESS ALERT!") appears on the video feed.
- **Auditory Signal:** An alarm or buzzer sound is played using the playsound module.
- **Threshold Logic:** Triggered when EAR remains below 0.3 for 48 frames continuously.
- **Driver Engagement:** Alerts encourage the driver to take a break or rest if fatigue persists.
- **Customizable Output:** Alarm sound and threshold values can be modified for user preferences.

5.3.2 On-Screen Display and Monitoring

In addition to audible warnings, the system offers a live on-screen display that visually communicates the driver's drowsiness status. This component provides real-time feedback to both the driver and the developer or observer. It includes important diagnostics, such as the EAR value and a graphical outline of detected eye regions, which improves interpretability and helps in debugging and performance evaluation. This visual representation is especially useful during system testing and can also serve as a monitoring tool for fleet managers or supervisors.

- **EAR Display:** The live Eye Aspect Ratio is shown on the top-right corner of the frame.
- **Eye Region Highlighting:** The left and right eyes are outlined using green convex hulls.
- **Frame Labeling:** Warnings are labeled directly on the video feed using cv2.putText.
- **Status Clarity:** Real-time frame analysis helps confirm if detection is working accurately.
- **Monitoring Tool:** Useful for observing driver behavior trends over time.
- **Debugging Aid:** Helps in identifying misdetections or camera alignment issues.
- **User-Friendly Interface:** Designed to be simple and readable while driving.

5.4 SETTING THE ENVIRONMENT

Before developing and running the Real-Time Drowsiness Detection System, a proper environment must be established to support the libraries and hardware dependencies involved. The system requires specific software tools and compatible hardware components to function smoothly in real-time. Setting up the environment includes installing essential Python libraries, configuring IDEs, and ensuring the system has a functioning. A balanced software-hardware integration ensures accurate facial landmark detection and fast video frame processing. For this project, the setup is intentionally kept lightweight to allow deployment even on entry-level systems. Below is a breakdown of the software and hardware environments required.

5.4.1 Software Environment

The software setup for this project involves using Python as the core programming language due to its simplicity and the availability of powerful libraries for computer vision and machine learning. The software environment should be configured to ensure smooth development, testing, and real-time execution of the application. Compatibility with various libraries and the use of a reliable code editor or IDE are essential for optimal performance.

Key Components of the Software Environment:

- **Python Version:** Python 3.6 or above is recommended for compatibility with OpenCV and Dlib.
- **Libraries to Install:**
  - opencv-python for image and video processing
  - dlib for facial landmark detection
  - imutils for frame manipulation
  - numpy and scipy for numerical operations
  - playsound for alarm audio playback

- **IDE/Editor:** Use PyCharm, VS Code, or Jupyter Notebook for an efficient coding experience.
- **Command Line Tools:** Ensure pip is available for easy installation of dependencies.
- **Facial Landmark Model:** Download and link the shape predictor file (shape_predictor_68_face_landmarks.dat) from Dlib's official repository.
- **Operating System:** Compatible with Windows 7/8/10 or Linux-based systems.

5.4.2 Hardware Environment

The hardware environment refers to the physical and technical components required to deploy and test the drowsiness detection system. Since this is a real-time video processing application, basic but reliable hardware is needed to ensure smooth execution and minimal lag. The project is designed to be lightweight, allowing it to run on systems with modest configurations, which makes it practical for broader use.

Essential Components of the Hardware Environment:

- **Processor:** Intel Pentium IV or higher (Core i3/i5 preferred for faster computation)
- **RAM:** Minimum 512MB; 2GB or more recommended for better performance

- **Storage:** At least 20GB of free space to install Python, libraries, and store sample outputs
- **Webcam:** A USB webcam or built-in laptop camera with at least 640x480 resolution for face tracking
- **Display:** SVGA color monitor or laptop screen for displaying real-time frame analysis
- **Keyboard/Mouse:** Standard input devices to run the application and respond to alerts
- **Internet Access (Optional):** Required initially to install Python packages and download the shape predictor model
- **Power Backup (Recommended):** To ensure uninterrupted operation during testing or field deployment.

Chapter 6

RESULTS AND DISCUSSION

The Real-Time Drowsiness Detection System was implemented and tested under various real-world conditions to evaluate its reliability and performance. The system successfully detected drowsiness by analyzing eye closure over time, using the Eye Aspect Ratio (EAR) metric. The facial landmarks were accurately detected using Dlib, and eye status was monitored constantly during live video streaming.

The alert system functioned effectively, triggering a buzzer sound when the driver's eyes remained closed beyond the specified threshold. The interface was responsive, with minimal lag, even on systems with basic hardware specifications. The system's non-intrusive nature made it practical for prolonged usage without causing discomfort to the driver.

Tests were conducted under different lighting conditions and head orientations. While the system performed well in normal lighting, slight performance drops were observed under low-light conditions and with users wearing glasses. Overall, the project met its objectives and demonstrated a strong potential for real-time application in vehicles to improve road safety.

6.1 WORKING OF THE PROPOSED SYSTEM

The proposed system works by integrating computer vision and real-time monitoring techniques to detect driver drowsiness effectively. A webcam captures continuous video footage of the driver’s face. Each frame is resized and converted into grayscale to simplify processing. The system then uses a Haar Cascade Classifier to detect the face region and Dlib’s 68-point facial landmark model to localize the eyes.

From the identified eye region, the Eye Aspect Ratio (EAR) is calculated using the vertical and horizontal distances between eye landmarks. The EAR remains fairly constant when the eye is open but drops significantly when the eye is closed.

If the EAR value remains below this threshold for a continuous number of frames (e.g., 48 frames), the system concludes that the driver is drowsy. It then activates an alert system, which triggers an audible alarm to awaken the driver. This process repeats continuously, frame by frame, to provide round-the-clock monitoring.

The system avoids physical contact or wearable sensors, making it a non-intrusive solution. Its simplicity, efficiency, and real-time response make it ideal for deployment in vehicles, especially in long-haul or night-time travel scenarios where drowsiness is a critical risk.

6.2 EFFICIENCY OF CURRENT SYSTEM

The current implementation of the drowsiness detection system demonstrates a high level of efficiency in terms of speed, responsiveness, and real-time accuracy. The use of lightweight and optimized algorithms allows it to run smoothly even on devices with modest hardware capabilities. It can process and analyze frames in real time, with negligible delay, making it effective for real-world driving conditions.

One of the major strengths of the system is its non-intrusive design. Unlike traditional physiological-based methods (such as EEG or heart-rate monitoring), this solution relies solely on visual input from a webcam. This increases driver comfort and makes installation easy. The Eye Aspect Ratio (EAR) method ensures that minor blinks are ignored, and only prolonged eye closure triggers the drowsiness alert, reducing false positives.

However, some limitations still exist. The system's performance may degrade in low-light environments or when drivers wear sunglasses or spectacles, as eye visibility becomes limited. Head tilts and side glances may also interfere with accurate detection.

Despite these constraints, the current system provides an estimated accuracy of 80–85% in normal conditions, which is substantial for a safety-critical application. With future improvements like infrared cameras and more advanced machine learning models, the system's robustness and reliability can be enhanced significantly.

6.3 USER EXPERIENCE AND PRACTICALITY

The user experience of the Real-Time Drowsiness Detection System is designed to be simple, non-intrusive, and user-friendly. Since the system does not require any physical contact—such as wearing sensors or headgear—it is particularly well-suited for long-duration use, such as in taxis, trucks, or buses. The camera-based monitoring ensures that the driver can operate the vehicle without distraction while still being under passive observation.

The visual output on-screen (with EAR values and eye contours displayed) provides real-time feedback, which is helpful for debugging and awareness. The alert mechanism—a buzzer sound—is loud enough to wake a drowsy driver but not so intrusive that it becomes a nuisance. The system can run in the background, allowing it to be integrated seamlessly into vehicle infotainment systems or dashboard displays.

Moreover, the practicality of the system is evident in its lightweight design. It doesn't require internet access or advanced computing power, making it deployable in a wide range of vehicles, including older models. Installation involves only a standard webcam and a device to run the Python application. These features ensure that the system can be adopted widely, even in cost-sensitive environments like public transportation or fleet services.

6.4 LIMITATIONS AND CHALLENGES

While the Real-Time Drowsiness Detection System performs efficiently in controlled environments, it does face some limitations in real-world scenarios. One of the primary challenges is varying lighting conditions. Bright sunlight, shadows, or nighttime driving can reduce the accuracy of facial landmark detection, leading to false negatives or inconsistent performance.

Another limitation involves occlusion and obstruction. Drivers wearing sunglasses, colored lenses, or masks can prevent the camera from accurately detecting eye regions, making it difficult to compute the Eye Aspect Ratio (EAR). Similarly, if the driver turns their head away from the camera or if their face is partially out of the frame, detection may fail temporarily.

Moreover, detection delay and sensitivity thresholds present a trade-off. If the threshold is too high, false positives increase; if too low, the system may fail to issue timely warnings. These thresholds must be carefully calibrated for different individuals and conditions.

Despite these challenges, the current system lays a strong foundation for further development. The integration of infrared vision, head pose estimation, or multi-modal sensors (e.g., combining video with heart rate or motion sensors) could dramatically improve robustness. Addressing these limitations will be key in evolving the prototype into a fully production-ready driver monitoring system.

6.5 COMPARISON WITH OTHER ALGORITHMS

The Real-Time Drowsiness Detection System implemented in this project primarily uses facial landmark-based eye monitoring with the Eye Aspect Ratio (EAR) as the core detection metric. This method offers a balance of simplicity, speed, and acceptable accuracy, making it well-suited for real-time applications. However, it is valuable to compare it with other commonly used algorithms in the domain of drowsiness detection.

One of the alternative approaches is EEG-based monitoring, which relies on brainwave activity to detect fatigue. While EEG offers high accuracy, it requires the driver to wear electrodes, making it intrusive and impractical for everyday use. Similarly, heart rate variability (HRV) analysis using wearable devices can provide insights into fatigue but also faces challenges in accuracy due to external factors like stress or physical activity.

Another widely studied method is machine learning classification using CNNs (Convolutional Neural Networks) on datasets of facial expressions or closed/open eyes. While CNNs are powerful and can adapt to varying conditions, they often require large, labeled datasets and powerful GPUs for training and inference, which increases complexity and cost.

In contrast, the EAR-based approach used in the project is non-intrusive, and low-cost hardware without the need for extensive training. Although it may not reach the sophistication of deep learning models in diverse environments, it offers a strong baseline solution.

TABLE 6.1. PERFORMANCE ANALYSIS

Case	Eye Detection	Eye Closure Detection	Mouth Detection	Yawn Detection	Results (Drowsiness Detection)
1	No	No	No	No	No Driver
2	Yes	No	Yes	No	No Alarm
3	Yes	Yes	No	No	Alarm
4	Yes	No	Yes	Yes	Alarm
5	Yes	Yes	Yes	Yes	Alarm

TABLE 6.2. SYSTEM COMPONENT RESPONSE ANALYSIS

Test scenario	Video Feed Quality	Face Detected	Eye Detection	EAR Value	Alarm Triggered	System Response Time
A	Clear	Yes	Yes	0.22	Yes	1.2 sec
B	Blurred	No	No	–	No	N/A
C	Medium Lighting	Yes	Yes	0.35	No	0.9 sec
D	Low Light	Yes	No	–	Yes	1.5 sec
E	Clear	Yes	Yes	0.29	Yes	1. sec

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

The Real-Time Drowsiness Detection System successfully demonstrates how computer vision and machine learning can be applied to enhance road safety. Using Python, OpenCV, and Dlib, the system processes a live video feed from a webcam to detect facial landmarks, particularly around the eyes. By calculating the Eye Aspect Ratio (EAR) for each frame, it accurately identifies signs of fatigue such as prolonged eye closure or reduced blinking. The input video is analyzed frame-by-frame in real-time, ensuring immediate response through an alert mechanism. The source code is lightweight, efficient, and operates smoothly on basic hardware. Implementation and testing across different lighting conditions and user profiles showed reliable performance, though environmental factors like glare or glasses can affect accuracy. Overall, the project is cost-effective, non-intrusive, and scalable. It holds practical applications in personal vehicles, commercial fleets, and public transport, proving to be a valuable step toward reducing fatigue-related accidents on the road

7.1.1 Implementation with Python, OpenCV, Dlib, NumPy & SciPy

The core system is built in Python, leveraging OpenCV for real-time video capture and image processing, and Dlib's pre-trained 68-point facial landmark detector to pinpoint eye regions. NumPy and SciPy handle the underlying numerical operations required to compute distances between landmark points efficiently, ensuring the Eye Aspect Ratio (EAR) can be calculated on every frame without introducing latency.

7.1.2 Real-Time EAR Computation & Audio Alerts (playsound/winsound)

For each captured frame, the code computes the EAR in milliseconds, comparing it against a predefined threshold to distinguish between blinks and genuine eye closures. When prolonged eye closure is detected, the system immediately invokes playsound (or winsound on Windows).

7.1.4 Impact, Scalability & Future Enhancements

This lightweight, non-intrusive solution delivers cost-effective fatigue monitoring suitable for personal cars, commercial fleets, and public transport. Its modular architecture makes it easy to extend with features like yawning detection, head-pose tracking, or IoT integration for fleet management. Future work could incorporate infrared sensors for low-light environments and machine-learning classifiers for even greater accuracy.

7.2 FUTURE ENHANCEMENTS

One pathway to improve low-light and variable lighting robustness is to integrate infrared (IR) or near-infrared (NIR) illumination and camera modules. By augmenting the existing webcam feed with an IR light source and IR-sensitive sensor, the system can reliably detect and track facial landmarks in complete darkness or under harsh backlighting. OpenCV supports IR image acquisition, and Dlib's landmark predictor can be retrained or fine-tuned on IR datasets to maintain EAR calculation accuracy. This enhancement mitigates false negatives caused by glare or shadows, ensuring consistent drowsiness monitoring during nighttime driving or in poorly lit cabins.

Beyond eye closure, incorporating additional physiological and behavioral fatigue markers—such as yawning frequency, mouth openness, and head pose estimation—can significantly boost detection reliability. Using Dlib's full 68-point model or deploying OpenCV's Haar cascades for mouth regions, the system could calculate a Yawn Aspect Ratio (YAR) and track yawning events over time. Simultaneously, a head-pose estimation algorithm (e.g., solving the PnP problem on nose and eye landmarks) would detect nodding or tilting. Fusing EAR, YAR, and pose metrics in a rule-based or lightweight classifier reduces false alarms and captures more nuanced signs of driver fatigue.

Transitioning from threshold-based logic to data-driven machine-learning models offers another enhancement avenue. By collecting labeled driving video segments under varied conditions, one can train a supervised classifier—such as an SVM, Random Forest, or lightweight CNN—to distinguish between normal blinks and drowsy states. Frameworks like TensorFlow Lite or PyTorch Mobile would allow on-device inference at high frame rates. This shift enables the system to learn subtle spatiotemporal patterns in eye and facial movements.

REFERENCES

[1] Ranjitha, M. M., Ushasree, R., & Chitra, B. (2024). Driver Drowsiness Detection System. INTI JOURNAL, 2024(11), 1-8.

[2] Titare, S., Chinchghare, S., & Hande, K. N. (2021). Driver drowsiness detection and alert system. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 7(3), 583-588.

[3] Thulasimani, L., & SP, P. (2021). Real time driver drowsiness detection using opencv and facial landmarks. Int. J. of Aquatic Science, 12(2), 4297-4314.

[4] Hasan, M. M., Watling, C. N., & Larue, G. S. (2024). Validation and interpretation of a multimodal drowsiness detection system using explainable machine learning. Computer Methods and Programs in Biomedicine, 243, 107925.

[5] M. Siwach, S. Mann and D. Gupta, "A Practical Implementation of Driver Drowsiness Detection Using Facial Landmarks," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-4, doi: 10.1109/ICRITO56286.2022.9964990.

[6] Shiney, S. A., Seetharaman, R., Mageshwari, N., Saikiran, S., Sulaksha, B. K., Pravin, S. S., & Dheepak, P. L. M. (2024, September). A Methodology for Driver Drowsiness Detection based on OpenCV and Dlib Libraries. In 2024 5th International Conference on Smart Electronics and Communication (ICOSEC) (pp. 2047-2051).

[7] Mokashi, B., Tigadi, A. S., Chilad, B., Rudrappa, G., & Jadhav, S. (2024, May). Deep Learning Approach for Drowsiness Detection Using Facial Features. In 2024 5th International Conference for Emerging Technology (INCET) (pp. 1-7). IEEE.

[8] Essel, E., Lacy, F., Albalooshi, F., Elmedany, W., & Ismail, Y. (2024). Drowsiness Detection in Drivers Using Facial Feature Analysis. Applied Sciences, 15(1), 20.

[9] Liu, K., & Guo, P. (2024, September). Design and Implementation of Fatigue Driving Detection System Based on OpenCV. In Proceedings of the 2024 7th International Conference on Computer Information Science and Artificial Intelligence (pp. 166-172).

[10] Kumar, A. S., Gopirajan, P., Ramasamy, J., & Jackson, B.(2024, April). Machine Learning and OpenCV: A Powerful Combination for Driver Drowsiness Detection. In 2024 International Conference on Communication, Computing and Internet of Things (IC3IoT) (pp. 1-5). IEEE.

[11] Pandi, C., Anish, T. P., Selvanayaki, S., Jagadeesan, P., Vedaraj, M., & Sanjay, R. (2024, April). Detecting Driver Fatigue with Python and OpenCV. In 2024 10th International Conference on Communication and Signal Processing (ICCSP) (pp. 355-358). IEEE.

[12] Yang, L., Yang, H., Wei, H., Hu, Z., & Lv, C. (2024).Video-based driver drowsiness detection with optimised utilization of key facial features. IEEE Transactions on Intelligent Transportation Systems.

[13] Rupani, V. S. K., Thushar, V. V. S., & Tejith, K. (2024). Real-Time Drowsiness Detection Using Eye Aspect Ratio and Facial Landmark Detection. arXiv preprint arXiv:2408.05836.

[14] S. Mohanty, S. V. Hegde, S. Prasad and J. Manikandan, "Design of Real-time Drowsiness Detection System using Dlib," 2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Bangalore, India, 2019, pp. 1-4, doi: 10.1109/WIECON-ECE48653.2019.9019910.

[15] P. Singh, S. P. S. Chauhan and E. Rajesh, "Real-Time Driver Drowsiness Detection Using Dlib And openCV," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 956-960, doi: 10.1109/ICAC3N56670.2022.10074245.

[16] S. Rathod, T. Mali, Y. Jogani, N. Faldu, V. Odedra and P. K. Barik, "RealD3: A Real-time Driver Drowsiness Detection Scheme Using Machine Learning," 2023 IEEE Wireless Antenna and Microwave Symposium (WAMS), Ahmedabad, India, 2023, pp. 1-5, doi: 10.1109/WAMS57261.2023.10242860.

[17] V. Singhal, N. Soni, K. Khatri, B. K. Chokkar and K. Kumar, "Drowsiness Detection and Alert System using DLib," 2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT), Faridabad, India, 2023, pp. 242-246, doi: 10.1109/ICAICCIT60255.2023.10465683.

[18] J. Chandiwala and S. Agarwal, "Driver's real-time Drowsiness Detection using Adaptable Eye Aspect Ratio and Smart Alarm System," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 1350-1355, doi: 10.1109/ICACCS51430.2021.9441756.

.

CERTIFICATES

VELAMMAL ENGINEERING COLLEGE
AN AUTONOMOUS INSTITUTION

INSTITUTION'S INNOVATION COUNCIL
(Ministry of HRD Initiative)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NCCDCE'25

acm

CERTIFICATE OF PARTICIPATION

This is to certify that Dr./Mr./Ms. VIDHYA V

of SRM INSTITUTE OF SCIENCE AND TECHNOLOGY has presented a

paper titled VISION- BASED REAL-TIME DROWSINESS DETECTION WITH OPENCV AND Dlib

in the

10th National Conference on Cutting Edge Developments in Computing and Engineering held on 13th March, 2025.

FACULTY CO-ORDINATOR	HOD - CSE	PRINCIPAL
	Dr. B. MURUGESHWARI	Dr. S. SATISH KUMAR

VELAMMAL ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTION

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NCCDCE'25

acm

CERTIFICATE OF PARTICIPATION

This is to certify that Dr./Mr./Ms. **DHANARAJAN K**

of **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY** has presented a

paper titled **VISION- BASED REAL-TIME DROWSINESS DETECTION WITH OPENCV AND Dlib**

in the

10th National Conference on Cutting Edge Developments in Computing and Engineering held on 13th March, 2025.

FACULTY CO-ORDINATOR	**HOD - CSE**	**PRINCIPAL**
	Dr. B. MURUGESHWARI	Dr. S. SATISH KUMAR

VELAMMAL ENGINEERING COLLEGE
AN AUTONOMOUS INSTITUTION

INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NCCDCE'25

acm

CERTIFICATE OF PARTICIPATION

This is to certify that Dr./Mr./Ms. GAJASRI SARAVANAN of SRM INSTITUTE OF SCIENCE AND TECHNOLOGY has presented a paper titled VISION- BASED REAL-TIME DROWSINESS DETECTION WITH OPENCV AND Dlib in the **10th National Conference on Cutting Edge Developments in Computing and Engineering** held on 13th March, 2025.

FACULTY CO-ORDINATOR	HOD - CSE	PRINCIPAL
	Dr. B. MURUGESHWARI	Dr. S. SATISH KUMAR