A MINOR PROJECT REPORT

ON

EngageWise

Submitted in partial fulfilment of the requirement

for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(Artificial Intelligence & Machine Learning)

BY

D Haritha Bhaskara Vignya (20P61A6615)

Shreejit Chander Cheela (20P61A6647)

Sidhi Anish Kumar (20P61A6648)

Under the esteemed guidance of

Dr K. Shirisha Reddy

Professor

Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

DEPARTMENT

OF

COMPUTER SCIENCE & ENGINEERING

[Artificial Intelligence & Machine Learning]

CERTIFICATE

This is to certify that the minor project titled "**EngageWise"** submitted by **D.H.B Vignya(20P61A6615), Shreejit Cheela(20P61A6647), Anish Kumar(20P61A6648)** in BTech IV-I semester Computer Science & Engineering [AI & ML] is a record of the bonafide work carried out by them.

The results embodied in this report have not been submitted to any other University for the award of any degree**.**

INTERNAL GUIDE

Dr. K. Shirisha Reddy

HEAD OF THE DEPARTMENT

Dr. K. Shirisha Reddy

PROJECT GUIDE

Mrs P Navya

EXTERNAL EXAMINER

DECLARATION

We, **D.H.B Vignya, Shreejit Cheela, Anish Kumar,** bearing hall ticket numbers **20P61A6615, 20P61A6647, 20P61A6648** hereby declare that the minor project report entitled "**EngageWise**" under the guidance of **Dr. K. Shirisha Reddy**, Department of Computer Science Engineering(AI & ML), **Vignana Bharathi Institute of Technology, Hyderabad**, have submitted to Jawaharlal Nehru Technological University Hyderabad, Kukatpally, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering(AI & ML).

This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

D.H.B Vignya (20P61A6615)

Shreejit Cheela (20P61A6647)

Anish Kumar (20P61A6648)

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Chairman, **Dr. N. Goutham Rao** and secretary, **Dr. G. Manohar Reddy** who took keen interest to provide us the infrastructural facilities for carrying out the project work. Self-confidence, hard work, commitment and planning are essential to carry out any task. Possessing these qualities is sheer waste, if an opportunity does not exist. So, we whole-heartedly thank **Dr. P. V. S. Srinivas**, Principal, and **Dr. K. Shirisha Reddy**, Head of the Department, Computer Science and Engineering (AI & ML) for their encouragement and support and guidance in carrying out the project.

We would like to express our indebtedness to the project coordinator, Mrs.P**.Navya**, Assistant Professor, Department of CSE (AI & ML) for her valuable guidance during the course of project work.

We thank our Project Guide, **Dr. K. Shirisha Reddy**, for providing us with an excellent project and guiding us in completing our mini project successfully.

We would like to express our sincere thanks to all the staff of Computer Science and Engineering (AI & ML), VBIT, for their kind cooperation and timely help during the course of our project. Finally, we would like to thank our parents and friends who have always stood by us whenever we were in need of them.

ABSTRACT

As final-year students approaching our time to work in a continuous, streamlined environment that requires a lot of focus to get things done, we often find ourselves procrastinating and losing focus or attention sitting in front of a computer or a laptop. Such real-life incidents have inspired us to develop an assistant that uses real-time video to generate a detailed report on how attentive a person sitting in front of the computer is while working or studying. We decided to name this AI assistant EngageWise to reflect its mission of fostering focused and informed engagement during work and study sessions. EngageWise is designed to assist individuals in maintaining their attention and productivity, ultimately helping them achieve their goals efficiently in a world full of distractions.

Keywords: Machine Learning, Deep Learning, Computer Vision, Python, Flask, Web Development

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

[Artificial Intelligence & Machine Learning]

VISION

To achieve global standards of quality in technical education with the help of advanced resources and automated tools to bridge the gap between industry and academia.

MISSION

- Build the students technically competent on global arena through effective teaching learning process and world-class infrastructure.
- Inculcate professional ethics, societal concerns, technical skills and life-long learning to succeed in multidisciplinary fields.
- Establish competency centre in the field of Artificial Intelligence and Machine Learning with the collaboration of industry and innovative research.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS):

The graduates of Computer Science and Engineering with Specialization in Artificial Intelligence & Machine Learning will be able to:

PEO1: Domain knowledge: Impart strong foundation in basic sciences, Mathematics, engineering and emerging areas by Advanced tools and Technologies

PEO2: Professional Employment: Develop Professional skills that prepare them for immediate employment in industry, government, entrepreneurship and Research.

PEO3: Higher Degrees: Pursue higher studies and acquire masters and research.

PEO4: Engineering Citizenship: Communicate and work effectively, engage in team work, achieve professional advancement, exhibit leadership skills, and ethical attitude with a sense of social responsibility.

PEO5: Life Long Learning: Lead in their field and respond to the challenges of an ever-changing environment with the most current knowledge and technology

PROGRAM OUTCOMES (POS):

The graduates of Computer Science and Engineering with Specialization in Artificial Intelligence & Machine Learning will be able to:

1. **Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend

and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long Learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOS):

The graduates of Computer Science and Engineering with Specialization in Artificial Intelligence & Machine Learning will be able to:

PSO 1: Understand and Apply Multi-Disciplinary and core concepts with emerging technologies for sustaining with the Dynamic Industry Challenges.

PSO 2: Design Automated Applications in Machine Learning, Deep Learning, Natural Language Processing and Relevant Emerging areas for visualizing, interpreting the datasets.

PSO 3: Develop Computational Knowledge, project and Interpersonal skills using innovative tools for finding an elucidated solution of the real-world problems and societal needs.

COURSE OBJECTIVE

- Identify and compare technical and practical issues related to the area of course specialization.
- Outline annotated bibliography of research demonstrating scholarly skills
- Prepare a well-organized report employing elements of technical writing and critical thinking.
- Demonstrate the ability to describe, interpret and analyze technical issues and develop competence in presenting.

COURSE OUTCOMES

CO1 - Demonstrate a sound technical knowledge of their selected seminar topic.

CO2 - Undertake problem identification, formulation, and solution.

CO3 - Design engineering solutions to complex problems utilizing a system approach.

CO4 - Organize a detailed literature survey and build a report with respect to technical publications.

CO5 - Make use of recent technology for building technical reports.

PROJECT OBJECTIVES

1. **Comprehensive Focus Analysis:** In-depth scrutiny of an individual's concentration and attention levels, providing a detailed understanding of their cognitive engagement.
2. **Real-time Reporting:** Continuous and immediate delivery of performance feedback, enabling users to make timely adjustments and improvements.
3. **Accurate Distraction Detection:** Precise identification of diversions or interruptions in the user's workflow or task, facilitating focused productivity.
4. **User Flexibility:** A system designed to accommodate individual preferences and adapt to varying user needs and requirements.
5. **Informative Visualization:** Presenting data in visually appealing and insightful formats, enhancing comprehension and decision-making.
6. **User-Friendly Interface:** An intuitive and easily navigable design that ensures a seamless and enjoyable user experience.
7. **Long-term Progress Tracking:** Consistent monitoring and recording of an individual's performance trends over an extended period, enabling goal setting and growth assessment.
8. **Data Privacy and Security:** A robust commitment to safeguarding user data, ensuring confidentiality, and protecting against unauthorized access or breaches.

PROJECT OUTCOMES

1. **Real-time Distraction Detection:** The system accurately identifies and alerts users in real-time when they exhibit signs of distraction, such as frequent blinking or yawning during tasks.
2. **Enhanced User Productivity:** Users experience improved productivity and focus through proactive notifications on the screen and awareness of their attentiveness levels with the values.
3. **Detailed Focus Reports**: EngageWise performs comprehensive calculations summarizing users' attention patterns, including periods of intense focus, distractions, and drowsiness episodes.
4. **Privacy-Centric Design:** EngageWise ensures stringent adherence to data privacy and ethical guidelines, safeguarding user information and upholding anonymity while collecting attention-related data, ensuring user trust and compliance with privacy regulations.
5. **Wide Applicability:** EngageWise caters to a diverse user base, including students, professionals, teachers, and gamers, by offering a versatile solution for monitoring and enhancing concentration during various activities.

PROJECT MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
Engage Wise	✓	✓	✓	✓	✓	✓			✓	✓	✓		✓	✓	

CO-PO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
CO1	✓	✓	✓	✓	✓								✓	✓	
CO2		✓	✓	✓					✓				✓		✓
CO3			✓		✓	✓			✓				✓		✓
CO4		✓		✓	✓					✓			✓	✓	
CO5					✓					✓				✓	

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

TERM	ABBREVIATION
AI	Artificial Intelligence
OpenCV	Open-Source Computer Vision
EEG	Electroencephalography
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
PERCLOS	Percentage of eyelid closure over the pupil over time
EAR	Eye Aspect Ratio
MAR	Mouth Aspect Ratio

CHAPTER - 1

1. INTRODUCTION

You don't need more time; you need more focus! is a powerful quote by the author and entrepreneur Jesse Itzler. As students, we sometimes find it hard to focus on our work, be it studying for exams or working on a project on a computer or laptop. We often get distracted by other elements on our screens, toggling between pages and app icons. This behavior wastes time and hampers our attentiveness.

We came up with the idea of building a focus app that analyses how attentive the person is based on their facial movements and gestures and generates a report after the end of the session.

1.1) Existing Model:

The existing models for an attention or focus detection systems are spread across various domains like attention heatmaps, pupil tracking, Eye Aspect Ratio (EAR), drowsiness detection systems and EEG based attention detection systems.

1. Attention Heatmaps: These models involve creating heatmaps of where a person's gaze or attention is concentrated on a screen. While not OpenCV-specific, this technique can be used to provide visualizations of attention patterns of the person.

2. Pupil Tracking: Pupil tracking is a common technique for assessing focus and attention. OpenCV can be used to detect and track the position of the pupil in the eye, which can provide insights into a person's focus based on the direction of their gaze.

3. Eye Aspect Ratio (EAR): The EAR is a measure of eye openness and can be used to detect blinks and assess alertness. In the proposed model, we use this ratio to detect the blinks and keep a check on how many times a person has blinked during the session.

4. Deep Learning Models: Deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been used to analyse facial expressions, eye movements, and other factors to predict attention and focus.

1.2) Proposed Model:

Incorporating elements from the existing models mentioned earlier, our comprehensive approach aims to generate a detailed report at the end of each user-selected session duration. By amalgamating OpenCV and Machine Learning in the initial prototype, we intend to

accurately identify instances when the user experiences distractions or lacks focus. The ultimate goal is to develop an interactive application, potentially in the form of a mobile or web app, that embodies these functionalities seamlessly.

1.3) Aim & Objectives:

1. **Real-Time Attention Monitoring:** Develop a real-time attention monitoring system that tracks and assesses a person's level of attention and focus while working or studying on a computer.
2. **User-Friendly Assistant:** Create a user-friendly AI assistant that seamlessly integrates into a person's workflow without causing distractions or interruptions.
3. **Early Warning System**: Develop an early warning system that alerts users when their attention starts to wane, helping them take proactive measures to regain focus and productivity.
4. **Detailed Reports**: Generate detailed reports and insights on attention patterns, including the duration of focused work, distractions, and periods of reduced attention.
5. **Customizable Settings**: Allow users to customize the system's sensitivity and alert thresholds to match individual preferences and needs.
6. **Privacy and Ethics**: Ensure strict adherence to privacy and ethical guidelines, respecting user data and anonymity while collecting attention-related information.

1.4) Scope:

The existing models are only directed towards the attention of students during classroom sessions or eLearning. We aim to develop a system that can be used by every individual working on a computer.

EngageWise can be used by students studying for their semester examinations, working professionals completing their tasks, teachers grading the students' assignments, gamers who want to track their level of immersion and attention during gaming sessions and so on.

CHAPTER - 2

2. LITERATURE SURVEY

With EngageWise, we aim to develop a personal assistant that alerts the user when there are signs of drowsiness and decreased attention levels. One can seldom find an exact implementation of EngageWise; the probability is nearly negligible. However, it is a compilation of many computer vision tasks put in one place.

We started this journey considering the driver drowsiness or alert systems as our predecessors, as the principle behind EngageWise and these existing systems overlap a little.

Paper I - Camera-based Estimation of Student's Attention in Class

Author – Mirko Raca

Our idea to determine if a person is attentive or feeling drowsy based on the non-verbal behaviour (facial, gestures) is inspired from this paper. The author of the paper [1] had a few objectives which aligned with our interests. The author carries out their research in a classroom, where a teacher and students are the actors, to determine the attention of the group of students during a lecture considering many non-verbal gestures like head pose estimation, gaze detection (to determine where the students are looking) and facial expressions. The base for determining the above-mentioned criteria is facial landmark detection, which is used in this paper and incorporated in our research.

Paper II - Students' Attention Assessment in eLearning based on Machine Learning

Authors - Qingshan Deng and Zhili Wu

The authors of the paper [2] conducted research to measure students' attentiveness in an eLearning environment based on facial detection and eye states. While this research is similar to EngageWise, we addressed some of its limitations. The authors of the paper considered only the state of the eye [open, closed] to assess students' visual attention. We extended this principle to detect if the person has their eyes closed or open, therefore incrementing the blink count.

Paper III - AttenQ- Attention Span Detection Tool for Online Learning

Authors – Pooja Koshti, Arya Paryani , Juhi Talreja, Vidya Zope

The authors of [3] aimed to solve the attention deficit problems of students in online learning. AttenQ was designed as a tool to help teachers assess the students' attention level so that the teacher can decide on what innovations they should come up with to keep the students attentive. The proposed solution uses computer vision and feature detection methods to monitor students' activities and provide teachers with information about their attention span. The system takes into account factors such as face detection, eye position estimation, head pose estimation, and drowsiness detection to determine whether a student is attentive or not.

Paper IV - How can a Robot Calculate the Level of Visual Focus of Human's Attention

Authors – Partha Chakraborthy, Mohammad Abu Yousuf, Zahidur Rahman Zahid, Nuruzzaman Faruqui

The paper [4] discusses how a robot can calculate the level of visual focus of human attention. The authors of the paper proposed a system that can determine the level of visual focus of attention by analysing eye movements and facial expressions. The system defines four levels of attention: high, medium, low, and no attention. In this paper, the scale of high-level attention is defined as ten, and the minimum scale of low-level attention is defined as zero. The level of attention is calculated based on the count of eyeball moves and time.

Paper V- Detection of Eye Blinking and Yawning for Monitoring Driver's Drowsiness in Real Time

Authors – Narender Kumar, Dr.N.C.Barwar

The authors of [5] have proposed a real time system using real time image processing, face/eye detection techniques, eye blink rates and yawning, which is designed a non-intrusive real time monitoring system for detecting driver's drowsiness. This study forms a stepping-stone to calculate the blink and yawn detection in our research.

Paper VI - Drowsiness detection using Eye-Blink frequency and Yawn count for Driver Alert

Authors - Maganti Manasa, Vikas B, K. Subhadra

Our proposed model incorporates blink frequency and yawn count to determine if the person is drowsy or not. This idea is inspired from the paper [6] which uses EAR and PERCLOS to detect the blink and yawn counts to analyse if the driver is feeling drowsy. The methodology and libraries used in this research are incorporated in EngageWise with a few changes.

Paper VII- Driver Fatigue Detection Based on Eye Tracking

Authors - Ling Gan, Bing Cui and Weixing Wang

In the paper [7], the authors developed a system for monitoring driver fatigue and attention using various sensors and algorithms. The system aims to detect signs of fatigue, such as changes in mouth shape, eye closing time, and eye position, and alert drivers to prevent accidents caused by drowsy driving. We leveraged changes in mouth shape, eye position and closing time in EngageWise to determine if the person is drowsy or active.

Paper VIII - Driver Drowsiness Detection System and Techniques: A Review

Authors-Vandna Saini, Rekha Saini

Although EngageWise is nowhere related to driver drowsiness detection systems that detect and alert a driver if he is drowsy or sleepy in a moving truck, EngageWise follows a same system to detect if a person in front of the system is not active during the session.

Both the systems have a common functionality. The camera system used here monitors facial features and head position for signs of drowsiness, such as yawning and sudden head nods. Additionally, eye blinking rate and eye closure duration are measured to detect drowsiness.

Paper IX- A Survey on State-of-the-Art Drowsiness Detection Techniques

Authors – Muhammad Ramzani, Hikmat Ullah Khan, Shahid Mahmood Awani, Amina Ismail Mahwish Ilyas and Ahsan Mahmood

The authors of [9] have conducted in-depth research in the field of driver drowsiness detection systems that have been developed till date.

This is a systematic review paper that discusses the best possible techniques, measures, tools, and classification methods for detecting driver drowsiness. The paper is divided into seven main sections, which include a detailed analysis and evaluation of selected papers, a review of drowsiness detection techniques, a comparative study of drowsiness detection

techniques, and a discussion of classification methods used for drowsiness detection. The paper concludes with a summary of the findings and suggestions for future research. The study aims to provide a comprehensive understanding of the state-of-the-art research in the field of drowsiness detection systems.

Paper X - Drowsiness Detection System using Eye Aspect Ratio Technique

Authors - Saravanaraj Sathasivam, Azmi Sidek, Abd Kadir Mahamad, Sharifah Saon, Hussein Ali Ameen, Mohamad Md Som

Our idea to use Eye Aspect Ratio to determine if the person's eyes are open or closed is inspired from this research [10]. The driver's images are processed to detect the drowsiness of the driver by performing live monitoring of Eye Aspect Ratio (EAR) using pre-trained Neural Network based dlib functions. The EAR technique calculates the ratio of the eyes using Python interpreter Anaconda and PyCharm in order to recognize drowsiness. The system can identify the occurrence of microsleep easily and alert the driver through an alarm for any positive drowsy condition. The developed system occupied with the Pi camera, Raspberry Pi 4, and GPS module are used to detect and analyse continuously the state of eye closure in real-time. The system can recognize whether the driver is drowsy or not, with the initial, wearing spectacles, dim light, and microsleep condition experimental conducted successfully give 90% of accuracy.

Summary:

To recapitulate, EngageWise is developed as a personal assistant for detecting drowsiness and decreased attention levels. Drawing inspiration from various research papers, the system combines elements of facial behaviour analysis and non-verbal cues to assess attentiveness. Notable influences include research on student attention assessment in classrooms, attentiveness in eLearning, and driver drowsiness detection techniques. EngageWise's methodology incorporates features like blink frequency and yawn count, similar to the techniques employed in these papers. While its primary focus is on personal attention monitoring, EngageWise shares common functionalities with driver drowsiness detection systems, such as assessing facial features and head position for signs of drowsiness.

CHAPTER – 3

3. DESIGN

3.1) Hardware Requirements:

The success of real-time computer vision applications, such as facial landmark detection and distance estimation, often hinges on the capabilities of the underlying hardware. For the Python-based facial analysis code snippet under consideration, the hardware requirements are notably influenced by the need for efficient image processing.

- Multicore Processor
- RAM – 4 GB RAM recommended
- Web Cam for capturing real time video (720p resolution)
- An integrated Graphical Processing Unit for better video feed and output (GPU)

3.2) Software Requirements:

- **Anaconda Navigator**: Anaconda Navigator was instrumental in managing and organizing the diverse Python libraries and dependencies required for our project, ensuring smooth compatibility and efficient package management.
- **VS Code:** Visual Studio Code (VS Code) served as our primary integrated development environment (IDE), offering a user-friendly interface and a wide range of extensions for Python development, greatly enhancing our coding and debugging efficiency.
- **Python:** Python, as the core programming language, formed the backbone of our project, providing versatile libraries for computer vision, machine learning, and real-time data processing.
- **Web Browser**: Web browsers were essential for testing and deploying our project. They enabled us to develop the web-based interface for our application and conduct real-time testing and monitoring.
- **HTML, CSS, and JS:** These web technologies played a pivotal role in creating an interactive and visually appealing user interface for our project, enhancing the user experience and accessibility.
- **Flask**: Flask, a micro web framework for Python, facilitated the development of our web application, enabling seamless integration of our machine learning models and real-time monitoring capabilities.

3.3) Model Architecture:

Our model captures the real-time video through the user's webcam, calculates how far the user is from the camera, keeps a track of the blink frequency and yawn count, therefore display an alert that the user is drowsy if they blink multiple times.

The flowchart below clearly explains all the events starting from the user initiating the session to ending the session.

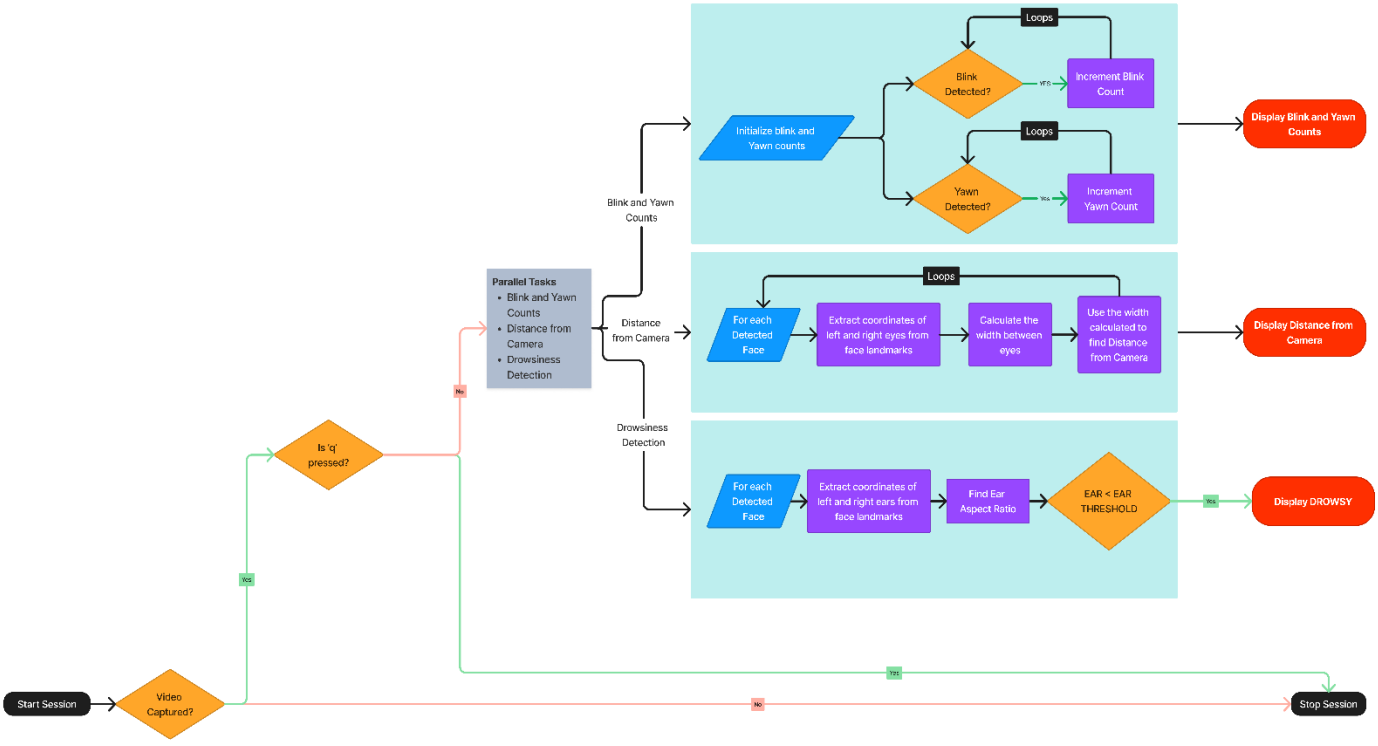

Figure 3.3

3.4) Algorithms

- **Haar Cascade Classifier**: Although it's not explicitly mentioned in the code, the cv2.VideoCapture and dlib.get_frontal_face_detector functions often use Haar Cascade Classifiers for face detection. Haar Cascades are a machine learning object detection method used to identify objects in images or video.
- **Facial Landmark Detection (dlib)**: The code utilizes dlib's facial landmark detection model. While not named in the code, dlib uses a combination of shape predictors and trained models to locate specific facial landmarks (such as eyes, nose, mouth, etc.) on detected faces.
- **Eye Aspect Ratio (EAR)**: The EAR is a blink detection technique. It calculates the ratio of distances between specific points on the eye, typically using the landmarks of the left and right eyes. If this ratio falls below a threshold, a blink is detected.
- **Mouth Aspect Ratio (MAR)**: The MAR is used for yawn detection. It computes the ratio of the vertical distance between the upper and lower lip to the horizontal distance between the corners of the mouth. A yawn is detected if this ratio exceeds a threshold.
- **Gaussian Blur**: cv2.GaussianBlur is applied to the video frames. Gaussian blurring is a technique for reducing noise and detail in an image, which can improve the accuracy of facial feature detection.

3.5) Libraries

- **openCV:** OpenCV is a popular computer vision library used for capturing video frames, image manipulation, and various computer vision operations
- **Scipy:** The Scipy library built on Numpy, is used to perform complex scientific and mathematical peroblems,In our model, we used scipy to calculate distance between the points
- **cvzone:** Cvzone is a library used for facial detection. It has utilities like FaceDetector and FaceMesDetector
- **imutils:**The imutils library provides a number of convenience functions to carry out image processing and computer vision tasks. In our model, we used imutils to resize the video frames and obtain the landmarks of the mouth
- **dlib:** The dlib library is used for a number of reasons.We used dlib to utilize the facial shape predictor to detect the landmarks on the face
- **collections:** The collections library is imported to utilize the OrderedDict to store the facial landmark indices in the form of a dictionary

CHAPTER - 4

4. IMPLEMENTATION

In this section, we present the implementation details of the Blink and Yawn Detection System using Computer Vision and Facial Landmarks. The system leverages various computer vision techniques and libraries to perform real-time detection of blink and yawn events.

4.1) Facial Landmark Detection

The system relies on facial landmarks to track the movements of eyes and the mouth. The facial landmarks are detected using the dlib library, which provides a pre-trained shape predictor model ("shape_predictor_68_face_landmarks.dat"). These landmarks are used to calculate various metrics related to eye and mouth movement.

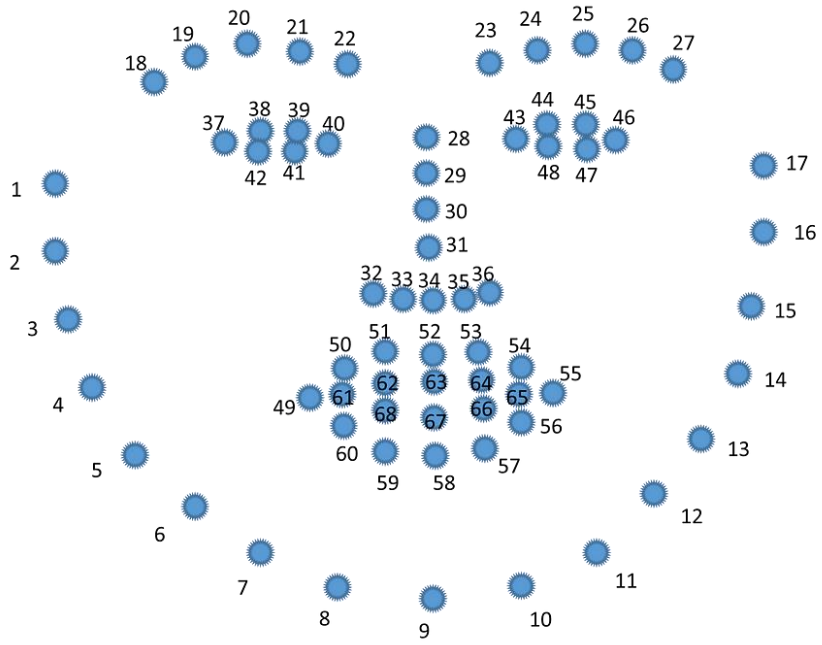

Figure 4.2

The key landmarks used in the system include:

- Mouth landmarks (Indexes 48 to 68)
- Right eyebrow landmarks (Indexes 17 to 22)
- Left eyebrow landmarks (Indexes 22 to 27)
- Right eye landmarks (Indexes 36 to 42)
- Left eye landmarks (Indexes 42 to 48)

- Nose landmarks (Indexes 27 to 35)
- Jaw landmarks (Indexes 0 to 17)

4.2) Blink Detection

Blink detection is performed by measuring the Eye Aspect Ratio (EAR). The EAR is calculated using the Euclidean distances between key points of the left and right eyes. If the EAR falls below a predefined threshold (EAR_THRESHOLD), it is considered as a blink event. The system keeps track of the blink count.

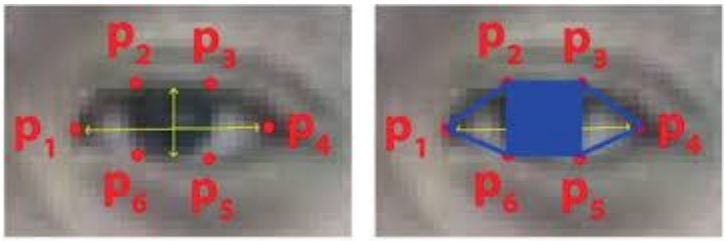

Figure 4.3(a)

The formula for calculating the EAR is given below

$$EAR_L = \frac{||P_2^L - P_6^L|| + ||P_3^L - P_5^L||}{2||P_1^L - P4^L||}$$

$$EAR_R = \frac{||P_2^R - P_6^R|| + ||P_3^R - P_5^R||}{2||P_1^R - P4^R||}$$

Figure 4.3(b)

4.3) Yawn Detection

Yawn detection is based on the Mouth Aspect Ratio (MAR). The MAR is calculated by measuring the ratio of the vertical distance between the upper and lower lip to the horizontal distance between the corners of the mouth. If the MAR exceeds a predefined threshold (MAR_THRESHOLD), it is detected as a yawn event. The system maintains a count of yawn events.

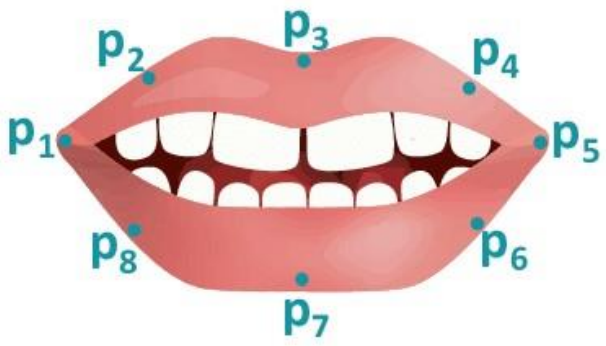

Figure 4.4

4.4) Distance Estimation

The system also incorporates distance estimation functionality using face detection. It utilizes the dlib face detector to identify faces in the video stream. The width of the detected face is used to estimate the distance from the camera, considering the known width of an average human face.

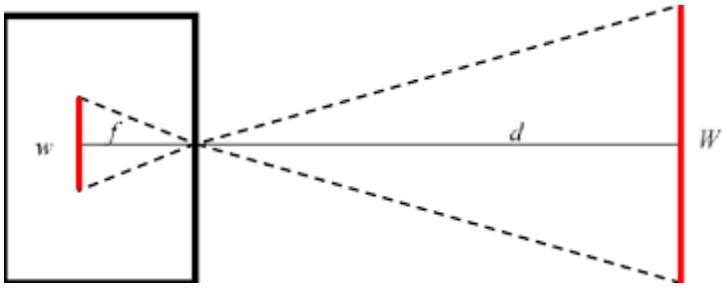

Figure 4.5

4.5) Alarm System

The system includes an alarm feature designed to activate when the individual exhibits signs of drowsiness for an extended duration, typically surpassing 10 seconds. This function aims to assist the user in promptly reawakening from potential moments of fatigue, enabling them to refocus and maintain productivity in their tasks.

4.6) User Interaction

The system provides a user interface for capturing images by pressing the 'c' key. Captured images are saved with incremental filenames (e.g., "image1.png"). The 'q' key is used to exit the program.

4.7) Real-time Visualization

The processed video frames are displayed in real-time using OpenCV's 'imshow' function. Various textual information, such as blink count, yawn count, and drowsiness status, is overlaid on the video feed to provide real-time feedback to the user.

4.8) Execution and Termination

The system continuously captures frames from the camera feed, processes them for blink and yawn detection, and displays the results. The program can be terminated by pressing the 'q' key. Upon termination, the camera is released, and all OpenCV windows are closed.

4.8) Code

cam.py [Modelling]:

```
import cv2
from scipy.spatial import distance
from cvzone.FaceMeshModule import FaceMeshDetector
from cvzone.FaceDetectionModule import FaceDetector
from imutils import face_utils
import numpy as np
import imutils
import time
import dlib
from collections import OrderedDict
number = 0
def calculate_EAR(eye):
    A = distance.euclidean(eye[1], eye[5])
    B = distance.euclidean(eye[2], eye[4])
    C = distance.euclidean(eye[0], eye[3])
    ear_aspect_ratio = (A + B) / (2.0 * C)
    return ear_aspect_ratio
def cal_MAR(mouth):
    dist_x = distance.euclidean(mouth[0], mouth[6])
    dist_y = distance.euclidean(mouth[3], mouth[9])
    mar = dist_y / dist_x
    return mar
```

```
FACIAL_LANDMARKS_IDXS = OrderedDict([
    ("mouth", (48, 68)),
    ("right_eyebrow", (17, 22)),
    ("left_eyebrow", (22, 27)),
    ("right_eye", (36, 42)),
    ("left_eye", (42, 48)),
    ("nose", (27, 35)),
    ("jaw", (0, 17))
])

cam = cv2.VideoCapture(0)
cam.set(3, 3000)
cam.set(4, 1700)
width = cam.get(3)
height = cam.get(4)
print(f"Current Camera Resolution: {width}x{height}")

hog_face_detector = dlib.get_frontal_face_detector()
dlib_facelandmark =
dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
(mStart, mEnd) = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]
detector = FaceMeshDetector()
detector2 = FaceDetector()
blink_flag = False  # Flag to track a single blink
yawn_flag = False   # Flag to track a single yawn
# Blink and Yawn detection thresholds
EAR_THRESHOLD = 0.21  # Adjusted threshold for blink detection
MAR_THRESHOLD = 0.6   # Adjusted threshold for yawn detection
blink_count = 0
yawn_count = 0

while True:
    try:
        ret, frame = cam.read()
```

```
        if not ret:
            print("Can't receive frame (stream end?). Exiting....")
            break

        frame = cv2.flip(frame, 1)  # Flip horizontally to correct the
video

        frame = cv2.GaussianBlur(frame, (7, 7), 0)

        # frame_yuv = cv2.cvtColor(frame, cv2.COLOR_BGR2YUV)
        # frame_yuv[:,:,0] = cv2.equalizeHist(frame_yuv[:,:,0]) * 0.5
        # frame = cv2.cvtColor(frame_yuv, cv2.COLOR_YUV2BGR)

        frame, faces = detector.findFaceMesh(frame, draw=False)
        frame, bboxs = detector2.findFaces(frame, draw=False)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        face = None

        for bbox in bboxs:
            x, y, w, h = bbox["bbox"]
            l, r = (x, y), (x + w, y + h)
            w = distance.euclidean(l, r)
            W, f = 6.3, 825
            d = (W * f) / w * 2.54
            try:
                cv2.putText(frame, "Distance: {}cm".format(int(d)),
(10, 90), cv2.FONT_HERSHEY_TRIPLEX, 0.7, (0, 0, 0), 2)
            except Exception as error:
                print(error)

        faces = hog_face_detector(gray)
        EAR = 0

        for face in faces:
```

```
            face_landmarks = dlib_facelandmark(gray, face)
            face_landmarks = face_utils.shape_to_np(face_landmarks)
            leftEye = face_landmarks[36:42]
            rightEye = face_landmarks[42:48]
            mouth = face_landmarks[mStart:mEnd]

            left_ear = calculate_EAR(leftEye)
            right_ear = calculate_EAR(rightEye)
            mar = cal_MAR(mouth)

            EAR = (left_ear + right_ear) / 2

            if EAR < EAR_THRESHOLD:
                if not blink_flag:
                    blink_flag = True
                    blink_count += 1
                    cv2.putText(frame, "Blink Detected!", (1100, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)
            else:
                blink_flag = False

            if mar > MAR_THRESHOLD:
                if not yawn_flag:
                    yawn_flag = True
                    yawn_count += 1
                    cv2.putText(frame, "Yawn Detected!", (1100, 60),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)
            else:
                yawn_flag = False

        cv2.putText(frame, "Blink count: {}".format(blink_count), (10,
30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)
        cv2.putText(frame, "Yawn count: {}".format(yawn_count), (10,
60), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)
```

```
        if EAR < EAR_THRESHOLD:
            cv2.putText(frame, "DROWSY", (500, 75),
cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 4)

        key = cv2.waitKey(1)
        if key == ord('c'):
            number += 1
            print(f"Image {number} has been captured")
            cv2.imwrite(f'image{number}.png', frame.copy())
        if key == ord('q'):
            cv2.destroyAllWindows()
            break

        cv2.imshow("Image", frame)

    except Exception as e:
        print(f"An error occurred: {e}")
        break
cam.release()
cv2.destroyAllWindows()
```

main.py [Flask Integration]:

```
from flask import Flask, render_template, Response
from cam import EngageWise

app = Flask(__name__)
ew = EngageWise()

@app.route("/")
def index():
    return render_template("index.html", blink=ew.blink_count,
yawn=ew.yawn_count, state=ew.state, d=ew.d)

def gen(engage_wise):
```

```
    while True:
        frame = engage_wise.get_frame()
        yield (b'--frame\r\n'b'Content-Type: image/jpeg\r\n\r\n'+ frame
+ b'\r\n\r\n')

@app.route("/video_feed")
def video_feed():
    return Response(gen(ew), mimetype='multipart/x-mixed-replace;
boundary=frame')

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000, debug=True)
```

index.html [Frontend]:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>EngageWise</title>
    <style>
        body {
            background-color: black;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
        }

        h1 {
            color: white;
```

```
        }

        #video-container {
            width: 640px;
            height: 480px;
            background-color: #000;
            border: 2px solid white;
            display: flex;
            justify-content: center;
            align-items: center;
            position: relative;
        }

        #video {
            width: 100%;
            height: 100%;
            object-fit: cover;
        }

        #buttons-container {
            margin-top: 20px;
            display: flex;
            gap: 20px;
        }

        button {
            background-color: transparent;
            color: white;
            border: 2px solid transparent;
            padding: 10px 20px;
            cursor: pointer;
            font-size: 16px;
            transition: border-color 0.3s ease-in-out, color 0.3s ease-
in-out;
        }
```

```
        .neon-button {
            border-color: #fff;
            color: #fff;
        }

        #startButton:hover {
            background-color: #39FF14;
            border-color: #00f;
            text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #00f, 0
0 40px #00f, 0 0 50px #00f, 0 0 60px #00f, 0 0 70px #00f;
        }

        #stopButton:hover {
            background-color: #FF0000;
            border-color: #00f;
            text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #00f, 0
0 40px #00f, 0 0 50px #00f, 0 0 60px #00f, 0 0 70px #00f;
        }


        p {
            color: white;
            font-size: 18px;
            margin: 10px 0;
        }
        span {
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1>EngageWise</h1>
    <div style="display: flex; margin-right: 330px;">
```

```
        <form method="GET" enctype="multipart/form-data" style="margin:
150px 30px;">
            <p>Blink Count: <span id="blinkCount">{{ blink
}}</span></p>
            <p>Yawn Count: <span id="yawnCount">{{ yawn }}</span></p>
            <p>Drowsiness Status: <span id="drowsinessStatus">{{ state
}}</span></p>
            <p>Distance From Camera: <span id="distance">{{ d }}</span>
cm</p>
        </form>
        <!-- <div style="margin: 150px 30px;">
            <p>Blink Count: <span id="blinkCount">0</span></p>
            <p>Yawn Count: <span id="yawnCount">0</span></p>
            <p>Drowsiness Status: <span
id="drowsinessStatus">Awake</span></p>
            <p>Distance From Camera: <span id="distance">0</span>
cm</p>
        </div> -->
        <div id="video-container">
            <img id="video" src="{{ url_for('video_feed') }}">
        </div>
    </div>
    <div id="buttons-container">
        <button id="startButton" class="neon-button">Start
Session</button>
        <button id="stopButton" class="neon-button">Stop
Session</button>
    </div>
    <script>
        const blinkCount = document.getElementById('blinkCount');
        const yawnCount = document.getElementById('yawnCount');
        const drowsinessStatus =
document.getElementById('drowsinessStatus');
        const distance = document.getElementById('distance');
```

```
        const startButton = document.getElementById('startButton');
        const stopButton = document.getElementById('stopButton');

        const video = document.getElementById('video');

        function updateValues(data) {
            const values = data.split(',');
            blinkCount.innerText = values[0];
            yawnCount.innerText = values[1];
            drowsinessStatus.innerText = values[2];
            distance.innerText = values[3];
        }
        const source = new EventSource("/video_feed");
        source.onmessage = (event) => {
            updateValues(event.data);
        };

        startButton.addEventListener('click', () => {
            video.style.display = 'block';
            source.close();
            source.onmessage = (event) => {
                const frameData = event.data.split(',');
                updateValues(frameData[1]);
            };
        });
        stopButton.addEventListener('click', () => {
            video.style.display = 'none';
            source.close();
            source.onmessage = (event) => {
                updateValues(event.data);
            };
        });
    </script>
</body>
</html>
```

CHAPTER - 5

5. RESULTS & DISCUSSIONS

1. Active/Open Screen

2. Blink Detection:

3. Yawn Detection

4. Drowsy Detection

CHAPTER - 6

6. CONCLUSION & FUTURE ENHANCEMENT

Conclusion:

EngageWise is a sophisticated focus detection application designed to assess the attention span of individuals, including students preparing for exams, working professionals, and teachers. Leveraging advanced computer vision techniques and facial feature analysis, the application provides real-time insights into the user's level of attentiveness.

The system employs a combination of facial landmark detection, blink, and yawn analysis to gauge the user's focus during study or work sessions. By monitoring the Eye Aspect Ratio (EAR) for blink detection and the Mouth Aspect Ratio (MAR) for yawn detection, EngageWise can effectively identify signs of drowsiness or reduced attention.

Future Enhancements:

EngageWise is not entirely accurate as this is the first phase. We definitely look forward to enhance the model by fine tuning the parameters, adding a few functionalities to make it more engaging, incorporate user feedback mechanism and prioritize security and privacy of the users.

REFERENCES

[1] Raca, Mirko. (2015). Camera-based Estimation of Student's Attention in Class.

[2] Qingshan Deng and Zhili Wu 2018 *IOP Conf. Ser.: Earth Environ. Sci.* 199032042

[3] Pooja Koshti(2022) AttenQ- Attention Span Detection Tool for Online Learning

[4] Chakraborty, Partha & Yousuf, Mohammad & Zahid, Zahidur & Faruqui, Nuruzzaman. (2020). How Can a Robot Calculate the Level of Visual Focus of Human's Attention. 10.1007/978-981-15-3607-6_27.

[5] Kumar, N., & Barwar, D.N. (2014). Detection of Eye Blinking and Yawning for Monitoring Driver' s Drowsiness in Real Time.

[6] Maganti Manasa, Vikas B, K. Subhadra (2019). Drowsiness detection using Eye-Blink frequency and Yawn count for Driver Alert.

[7] Ling Gan, Bing Cui and Weixing Wang, "Driver Fatigue Detection Based on Eye Tracking," 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 2006, pp. 5341-5344, doi: 10.1109/WCICA.2006.1714090.

[8] Vandna and Rekha Saini. "Driver Drowsiness Detection System and Techniques: A Review." (2014).

[9] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas and A. Mahmood, "A Survey on State-of-the-Art Drowsiness Detection Techniques," in IEEE Access, vol. 7, pp. 61904-61919, 2019, doi: 10.1109/ACCESS.2019.2914373.

[10] S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som and H. A. Ameen, "Drowsiness Detection System using Eye Aspect Ratio Technique," 2020 IEEE Student Conference on Research and Development (SCOReD), Batu Pahat, Malaysia, 2020, pp. 448-452, doi: 10.1109/SCOReD50371.2020.9251035.

[11] B. Alshaqaqi, A. S. Baquhaizel, M. E. Amine Ouis, M. Boumehed, A. Ouamri and M. Keche, "Driver drowsiness detection system," 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA), Algiers, Algeria, 2013, pp. 151-155, doi: 10.1109/WoSSPA.2013.6602353.

[12] L. Thulasimani; Poojeevan P; Prithashasni S P. "Real Time Driver Drowsiness Detection Using Opencv and Facial Landmarks". Int. J. of Aquatic Science, 12, 2, 2021, 4297-4314.

[13] R. Vishnu et al 2021 J. Phys.: Conf. Ser. 1916 012172 "Driver drowsiness detection system with opencv and keras"

DOI 10.1088/1742-6596/1916/1/012172

[14] Y. Suryawanshi and S. Agrawal, "Driver Drowsiness Detection System based on LBP and Haar Algorithm," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 778-783, doi: 10.1109/I-SMAC49090.2020.9243347.

[15] P. P. Patel, C. L. Pavesha, S. S. Sabat and S. S. More, "Deep Learning based Driver Drowsiness Detection," 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2022, pp. 380-386, doi: 10.1109/ICAAIC53929.2022.9793253.

[16] Mohanty, Archit, and Saurabh Bilgaiyan. "Drowsiness detection system using KNN and OpenCV." Machine Learning and Information Processing: Proceedings of ICMLIP 2020. Springer Singapore, 2021.

[17] Neelanjan. (n.d.). GitHub - neelanjan00/Driver-Drowsiness-Detection: https://github.com/neelanjan00/Driver-Drowsiness-Detection

[18] Misbah. (n.d.). GitHub - misbah4064/drowsinessDetector: OpenCV Drowsiness Detector using Python and Dlib. GitHub. https://github.com/misbah4064/drowsinessDetector

[19] Steflayanto. (n.d.). GitHub - steflayanto/nwHacks-2020-Back-End. GitHub. https://github.com/steflayanto/nwHacks-2020-Back-End

[20] Zhermin. (n.d.). GitHub - zhermin/ripplecreate-attention-model: Attention Indexing Computer Vision POC for ripplecreate. GitHub. https://github.com/zhermin/ripplecreate-attention-model