Intelligent Classroom Monitoring System using Image Proccesing

A report submitted for the course named Project - III(CS421)

Submitted By

KONADA GOPALAKRISHNA
SEMESTER - VII
20010123

Supervised By

DR.RAJKUMARI BIDYALAKSHMI DEVI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SENAPATI,
MANIPUR
NOV,2022

Abstract

For an effective classroom environment, it becomes important to track the activities and the state of the students taking the lecture. This becomes increasingly important owing to the difficulties faced by students and teachers in the transition from offline to online teaching due to the pandemic, followed by the gradual reinstating of offline teaching. This project entails the usage of deep learning algorithms and computer vision techniques including facial detection and recognition for real-time attendance of the students in both offline and online settings. Moreover, engagement detection practices, such as drowsiness detection and gaze detection to maintain the sanctity of a classroom have also been proposed.

Declaration

In this submission, I have expressed my idea in my own words, and I have adequately cited and referenced any ideas or words that were taken from another source. I also declare that I adhere to all principles of academic honesty and integrity and that I have not misrepresented or falsified any ideas, data, facts, or sources in this submission. If any violation of the above is made, I understand that the institute may take disciplinary action. Such a violation may also engender disciplinary action from the sources which were not properly cited or permission not taken when needed.

Konada Gopalakrishna
20010123

DATE:

Department of Computer Science Engineering
Indian Institute of Information Technology Senapati, Manipur

Dr. Rajkumari Bidyalakshmi Devii Email:bidyalakshmi@iiitmanipur.ac.in
Assistant Professor Contact No: +91 7005126046

To Whom It May Concern

This is certify that the Dissertation entitled **"Intelligent Classroom Monitoring System using Image Proccesing",** submitted by **Konada Gopalakrishna** , has been carried out under my supervision and that this work has not been submitted elsewhere for a degree,diploma or a course

Signature of Supervisor

(Dr.Rajkumari Bidyalakshmi Devi)

Department of Computer Science Engineering
Indian Institute of Information Technology Senapati, Manipur

Dr. Kishorjit Nongmeikapam
Assistant Professor

Email: kishorjit@iiitmanipur.ac.in
Contact No: +91 8974008610

To Whom It May Concern

This is certify that the Dissertation entitled **"Intelligent Classroom Monitoring System using Image Proccesin ",** submitted by **Konada GopalaKrishna** ,has been successfully carried out in the department of Computer science and this work has not been submitted else where for a degree,diploma or a course.

Signature of HOD

(Dr. Kishorjit Nongmeikapam)

Signature of the Examiner 1 ....

Signature of the Examiner 2 ....

Signature of the Examiner 3 ....

Signature of the Examiner 4 ....

Acknowledgement

I would like to express my sincere gratitude to several individuals for supporting me throughout my Project. First, I wish to express my sincere gratitude to my supervisor, *Dr Rajkumari Bidyalakshmi Devi*, for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my project and writing of this thesis. Her immense knowledge, profound experience and professional expertise has enabled me to complete this project successfully. Without his support and guidance, this project would not have been possible. I could not have imagined having a better supervisor in my study.

Konada Gopalakrishna

Contents

List of Figures

Chapter 1

Introduction

A real-time attendance and engagement tracking is an innovative solution leveraging advanced technologies such as computer vision and machine learning to automate the process of tracking attendance and gauging participant engagement in various settings. This system utilizes cameras and sensors to capture real-time data, analyze facial features or other relevant cues, and intelligently determine the presence of individuals. Additionally, it can assess engagement levels by analyzing factors like facial expressions, body language, and interactions.This technology offers a streamlined and efficient alternative to traditional attendance tracking methods, eliminating the need for manual processes and reducing the likelihood of errors. Moreover, the engagement detection aspect provides valuable insights into the level of participation and interest during meetings, classes, or events. The Smart Attendance and Engagement Detection System represents a cutting-edge solution with the potential to enhance productivity and engagement across various domains.

1.1 Introduction of the Project

In the pursuit of fostering an effective classroom environment, it is imperative to monitor student activities and their status during lectures. This becomes particularly crucial during the challenging transition from offline to online teaching due to the pandemic and subsequent efforts to reintroduce offline teaching. This project leverages deep learning algorithms and computer vision techniques, including facial detection and recognition, to facilitate real-time attendance tracking for students in both offline and online learning settings. Additionally, the project introduces engagement detection practices such as drowsiness detection and gaze detection to uphold the integrity of the classroom.

1.2 Existing System

In traditional classroom settings, attendance is typically taken manually, consuming valuable lecture time. The shift to online teaching has introduced various online attendance systems, but these often lack real-time monitoring and engagement detection capabilities.

1.3 Problems in Existing System

The existing systems, whether manual or online, are prone to inefficiencies and do not address the challenges posed by the evolving nature of teaching environments. Manual attendance is time-consuming, while online systems may lack features crucial for maintaining classroom engagement.

1.4 Proposed System

The proposed system utilizes the Local Binary Patterns Histogram, a key aspect is the utilization of the Local Binary Patterns Histogram algorithm for facial recognition. This algorithm is chosen for its efficiency in detecting faces in various lighting conditions and orientations, making it suitable for diverse classroom settings. The system's implementation involves updating a centralized database based on the detected presence or absence of students. This database serves as a valuable resource for tracking attendance trends over time, facilitating data-driven decision-making for educators.In addition to real-time attendance tracking, the system addresses the issue of student engagement during lectures. Drowsiness detection is achieved through facial landmark detection, which captures the coordinates of the eyes. The Eye Aspect Ratio (EAR) metric is then computed to determine whether a student's eyes are open or closed, providing insights into their level of attentiveness.Gaze detection is another crucial aspect incorporated into the system. It involves tracking the direction of the student's gaze, enabling the system to assess whether the student is actively focused on the lecture or potentially distracted.Furthermore, mobile-phone detection enhances the system's ability to monitor engagement. By analyzing the presence of mobile phones in the vicinity of the student, educators can gain insights into potential distractions and address them in real-time.Testing of the system has been conducted rigorously using both publicly available datasets and real-world data collected through webcam live streams. This ensures the system's reliability and effectiveness in various scenarios.Looking ahead, the project timeline outlines the integration of real-time attendance and drowsiness detection in the current semester as part of the minor project. Subsequently, the project aims to expand its capabilities by incorporating gaze detection and mobile phone detection. Additionally, a web interface will be developed in the next semester to enhance user interaction and accessibility.

1.5 Advantages of Proposed System

Efficient Attendance Tracking: The system employs facial recognition algorithms to provide real-time attendance tracking, eliminating the need for manual efforts. Adaptability: The system is designed to function seamlessly in both offline and online teaching environments, ensuring flexibility. Engagement Detection: Incorporation of drowsiness detection, gaze detection, and mobile-phone detection enhances classroom engagement monitoring.

Chapter 2

Requirement Analysis

2.1 Functional Requirements

2.1.1 Real-time Attendance Tracking

In the pursuit of classroom efficiency, the system is mandated to harness the prowess of deep learning algorithms, orchestrating a ballet of facial detection and recognition seamlessly intertwining with the intricate threads of both offline and online educational realms. An imperative lies in the system's ability to harmonize in real-time, orchestrating the symphony of updating student attendance status in a database through the prism of facial recognition precision.

2.1.2 Engagement Detection

A poetic dance unfolds as the system transcends mere attendance tracking, delving into the realms of student engagement. Through the balletic finesse of drowsiness detection, facial landmarks pirouette gracefully, unveiling the coordinates of the eyes. The ethereal metric, known as the Eye Aspect Ratio, waltzes onto the stage, unraveling the delicate performance of open-eyed attentiveness. Gaze detection takes center stage, choreographing the scrutinizing scrutiny of student focus. Mobile-phone detection pirouettes discreetly, adding an enchanting layer to the ballet.

2.1.3 Interface

A digital canvas unfolds within the confines of a terminal, woven with threads of HTML/CSS-like elements, inviting users into an immersive and interactive experience. In this text-based interface, real-time attendance and engagement metrics elegantly pirouette, offering users a seamless choreography of data visualization and interaction. The terminal becomes the stage where the project's functionality is performed, providing a unique and efficient space for managing attendance and engagement metrics in real-time."

2.2 Non-Functional Requirements

2.2.1 Performance

The system is not a mere spectator but a maestro, demanding real-time crescendos in response to attendance and engagement inquiries. It is beholden to an unwavering standard of accuracy, a virtuoso hitting the right notes in facial recognition and engagement detection, echoing a symphony of precision

2.2.2 Usability

The user interface, a balletic stage for both students and educators, must be a tapestry of simplicity and intuition. It beckons users with a siren's call, requiring minimal training as it pirouettes effortlessly through the intricacies of operation.

2.3 Computational Resource Requirements

2.3.1 Hardware Requirements

The stage is adorned with essential props: CCTV cameras stand sentinel, capturing the images of students for attendance and engagement detection, while the humble webcam joins the ensemble, a crucial actor in the online rendition of this educational performance.

2.3.2 Software Requirements

2.4 Life Cycle Model

The project embarks on a journey, a rhythmic cadence echoing the iterative and incremental beats of development. It unfurls its narrative in chapters, beginning with the overture of real-time attendance and drowsiness detection in the minor project phase. Subsequent acts unfold in majesty, introducing gaze and mobile phone detection, culminating in a grand finale — the creation of a web interface in the major project phase, a digital crescendo elevating user interaction and accessibility to the zenith.

Chapter 3

Design

3.1 Architecture of the System

The system architecture represents the orchestrated integration of software and technical components, constituting a holistic solution for efficient and accurate attendance tracking. On the software front, the architecture is meticulously designed to handle key operations, including image capture, preprocessing, face detection utilizing OpenCV's haarcascade model, and face recognition through the Local Binary Patterns Histogram (LBPH) algorithm. This software layer ensures a systematic flow of tasks, optimizing the recognition process. On the technical side, the architecture outlines the hardware components, such as cameras (webcams or CCTV), and their seamless integration with the software modules. The synergy between software and hardware components creates a cohesive system that captures facial images, processes them with precision, and logs attendance data. This comprehensive approach addresses the intricacies of attendance tracking, providing an effective solution for diverse educational environments, be it offline or online settings.

3.1.1 Software Architecture

The software architecture of the system is intricately crafted to deliver a streamlined and effective image processing and recognition workflow. Comprising distinct modules, the architecture begins with image capture, where facial images are acquired for subsequent analysis. A crucial preprocessing step follows, addressing variability in brightness by converting the images to grayscale, thereby enhancing the reliability of downstream processes. The incorporation of OpenCV's haarcascade model is pivotal for precise face detection, ensuring the extraction of facial features for further analysis. The crux of the system lies in LBPH face recognition, a robust algorithm adept at accurately identifying registered users. This layered approach, orchestrated with precision, guarantees a systematic and efficient flow of operations. To conclude the process, the system seamlessly exports the attendance data to a

CSV file, providing a structured record for administrative use. The software architecture, with its modular design and strategic integration of key components, stands as the backbone of the system, ensuring a reliable and accurate mechanism for attendance tracking in diverse educational settings..

3.1.2 Technical Architecture

The technical architecture of the system is a pivotal component that harmonizes the symbiotic relationship between hardware and software elements, culminating in a robust and efficient attendance tracking solution. At its core, the integration of hardware involves strategically deploying cameras, be they webcams or CCTV, to capture facial images in real-time. These images serve as the foundation for subsequent processing within the software framework. The technical architecture meticulously outlines the systematic steps encompassing image capture, preprocessing, and the intricate dance between OpenCV and the LBPH algorithm for face detection and recognition. The strategic alignment of hardware and software components ensures a seamless and reliable operation, creating a responsive and accurate system for attendance tracking. The technical architecture, with its meticulous orchestration of steps and integrated approach, stands as a testament to the system's dependability and efficiency in diverse educational environments. .

3.2 UML Diagrams

UML diagrams provide a visual representation of the system's structure and interactions, enhancing the understanding of its design.

3.2.1 Project Workflow

Webcam: The webcam captures a video stream of the student's face. Preprocess: The preprocess module prepares the video stream for facial recognition by resizing and cropping the image. Facial image: The facial image is the output of the preprocess module. Facial recognition: The facial recognition module compares the facial image to a database of student images to identify the student. Attendance database: The attendance database stores information about students, such as their name, roll number, and attendance history. Stored images and student details: The stored images and student details are used by the facial recognition module to identify students. Detect eyes and calculate EAR: This module detects the student's eyes and calculates the eye aspect ratio (EAR). The EAR is a measure of how open the eyes are. If the EAR is too low, the system may infer that the student is drowsy. Attendance: The attendance module tracks the student's attendance based on the output of the facial recognition module. Drowsiness warning: The drowsiness warning module issues a warning to the student if the system detects

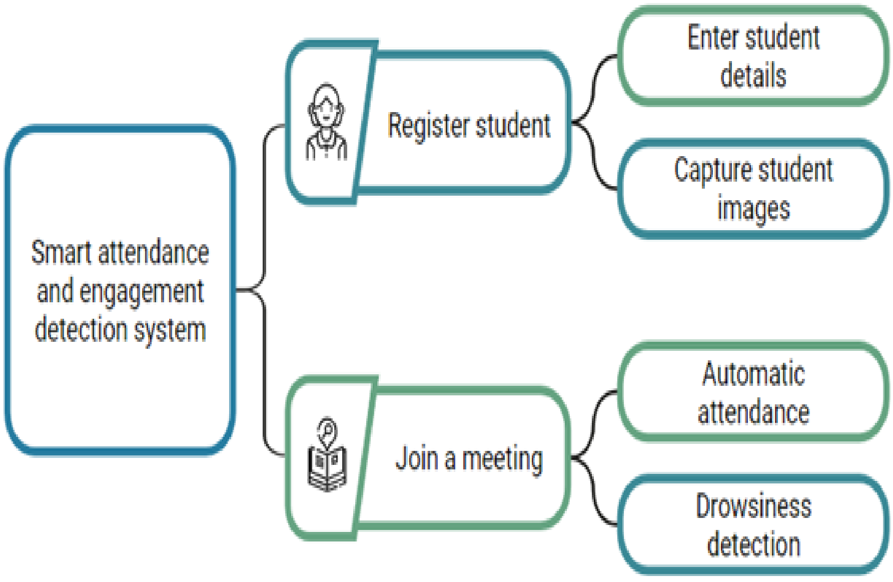


Figure 3.1: project overflow

that the student is drowsy. The smart attendance and engagement detection system can be used in a variety of settings, such as schools, universities, and workplaces. It can help to improve student attendance and engagement, and it can also help to identify students who may need additional support.

3.2.2 Dataflow Diagrams

Registering a student

The diagram shows how the webcam can be used to identify the student, track their attendance, and monitor their engagement.The webcam is connected to a computer, which is connected to a server. The server stores a database of student images and attendance records.When the student turns on their webcam, the system takes a picture of the student's face and compares it to the database of student images. If the student is identified, the system logs their attendance and begins monitoring their engagement.The system monitors the student's engagement by tracking the student's eye movements and head movements. If the student looks away from the screen or nods off, the system will generate a warning.The system can be used to improve student attendance and engagement by providing feedback to students and teachers. For example, the system can generate reports for teachers that show which students are attending class regularly and which students are paying attention.Overall, the diagram shows how a webcam can be used to

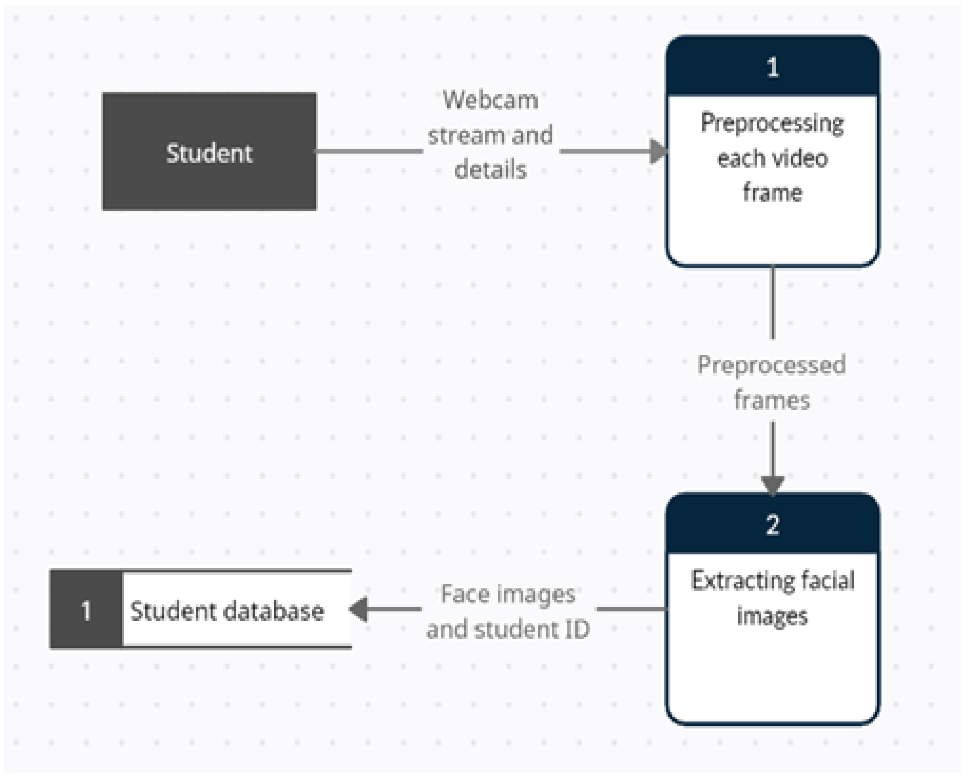
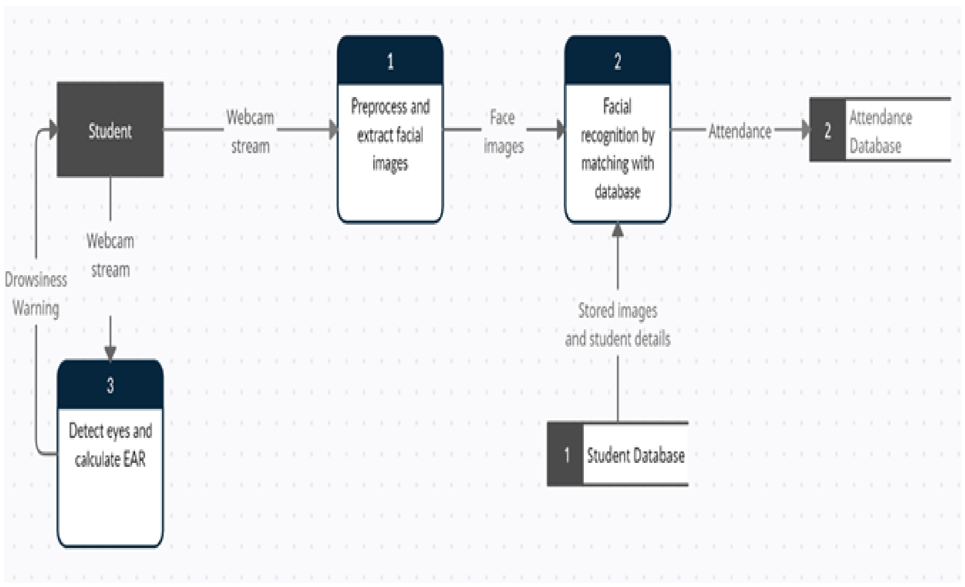

Student
Webcam stream and details
1
Preprocessing each video frame
Preprocessed frames
2
Extracting facial images
Face images and student ID
1
Student database

Figure 3.2: dataflow Diagrams

create a smart attendance and engagement detection system. This system can be used to improve student outcomes and to create a more engaging learning environment.

3.2.3 Automatic attendance and drowsiness detection



Student
Webcam stream
1
Preprocess and extract facial images
Face images
2
Facial recognition by matching with database
Attendance
2
Attendance Database
Webcam stream
Drowsiness Warning
3
Detect eyes and calculate EAR
Stored images and student details
1
Student Database

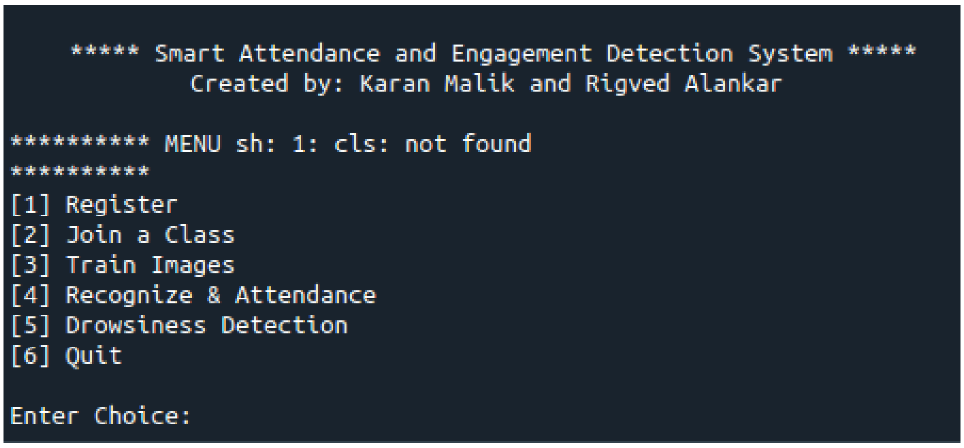
Figure 3.3: attendance system

The photo you sent shows a diagram of a student's learning process. The diagram shows how a student can access a website through a webcam. The student's webcam streams live video to the website, where it is processed to extract facial features. The facial features are then used to identify the student and to detect any signs of drowsiness. If the student is drowsy, the system will warn the student.The diagram also shows how the system can be used to collect attendance data. The system can store the student's facial images and match them to the student's database record. This allows the system to track the student's attendance and to generate attendance reports.The photo is likely from a research paper or a presentation on the topic of student engagement or learning analytics. The diagram is a clear and concise illustration of how a system can be used to monitor student engagement and to provide feedback to students.

3.2.4 project interface

Key features of the program:

Facial recognition:The system can identify students by their faces. Attendance tracking: The system can track students' attendance and generate



```
    ***** Smart Attendance and Engagement Detection System *****
            Created by: Karan Malik and Rigved Alankar

********** MENU sh: 1: cls: not found
**********
[1] Register
[2] Join a Class
[3] Train Images
[4] Recognize & Attendance
[5] Drowsiness Detection
[6] Quit

Enter Choice:
```

Figure 3.4: Interface

reports. Drowsiness detection: The system can detect drowsiness in students and issue warnings. Additional details: The program is designed for use in schools, universities, and workplaces. The program can be used to improve student attendance and engagement. The program can be used to identify students who may need additional support. Overall, the program is a valuable tool for educators and administrators who want to improve student outcomes.

Chapter 4

IMPLEMENTATION

4.1 Technologies

The implementation of the Smart Attendance System relies on the following technologies to ensure a seamless and efficient solution i am using the programming language is python

4.1.1 Python

Python is a versatile, high-level, and dynamically-typed programming language that emphasizes readability and simplicity, making it an ideal choice for a wide range of applications. Created by Guido van Rossum and first released in 1991, Python has gained immense popularity due to its ease of learning, clear syntax, and expansive standard library. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, providing developers with flexibility in structuring their code. It has become a prominent language in various domains, including web development, data science, machine learning, artificial intelligence, automation, and more. Python's vibrant community and extensive ecosystem of libraries and frameworks, such as NumPy, Pandas, Django, and TensorFlow, contribute to its versatility and enable developers to efficiently tackle complex tasks. With its straightforward syntax and robust capabilities, Python continues to be a go-to language for both beginners and experienced programmers alike.

4.2 Algorithms

4.2.1 Local Binary Pattern(LBP) Algorithms

The Smart Attendance System employs several algorithms to achieve its objectives, including real-time attendance tracking, facial recognition, drowsiness detection, and gaze detection. Here's an overview of the key algorithms used

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel and considers the result as a binary number. When LBP is joined with histograms of oriented gradients (HOG) descriptor, it improves the efficiency and accuracy considerably for some datasets. Moreover, utilizing LBP joined with histograms allows us to represent facial images using simple data vectors, thus making it more suitable for face detection tasks The LBPH algorithm works in 5 steps as given below:

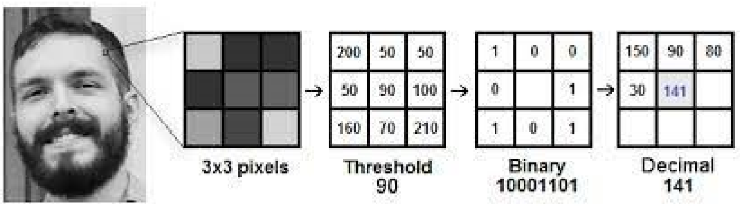
Parameters: LBPH uses 4 parameters:

- Radius: the radius is used to create a local binary pattern and represents the radius around the centre pixel. The frequency is set to 1.
- Neighbours: the number of sample points to form the circular binary pattern. Higher the number of neighbours, the higher the computational cost. The frequency has been set to 8.
- Grid X: the number of cells in the horizontal direction. Using larger values for Grid X i.e., a more precise and finer grid allows higher dimensionality for the resulting vector. The frequency is set to 8.
- Grid Y - the number of cells in the vertical direction. Using larger values for Grid Y i.e., a more precise and finer grid allows higher dimensionality for the resulting vector. The frequency is set to 8.

Creating the dataset:

The first step is to train the model. To do so, we use a database that contains images of people who are to be recognized i.e., our training data. Next, an identifier, such as a name or an identification number, is added to all the images to allow the algorithm to identify the input image. Multiple photos of the same person must have the same ID.

Applying the LBP operation

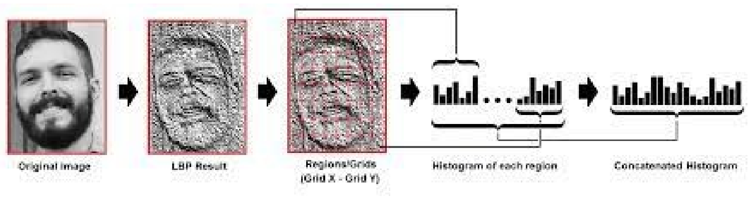


Next thing is to create an intermediate image that highlights the facial features of the original image. To do so, the algorithm law uses a sliding window which varies with change in the the parameters – radius and neighbours.

- We have a grayscale face image .
- Get a 3x3 pixel window from the image, with each number in the matrix showing the intensity of the corresponding pixel from 0-255.
- Find the central value of the matrix and use it as a threshold.
- Reassign the values of the remaining 8 neighbours – value=1 for pixels with equal or higher intensity than the threshold and value=0 for the others.
- Now, the matrix will contain only binary values. Concatenate the 8 binary values together to form a new 8-bit binary value. (e.g. 10001101).
- After that, we convert this binary number to a decimal value and set it to the center value of the matrix, which is a pixel from the first image.
- At the end of this process (LBP process), we have a new image that better represents the features of the original image

Recording Histograms:

Now, using the image created in the last step, we can use the Grid X and Grid Y parameters to split the image into multiple grids, as can be seen in the following image [5]:



- Since we have a grayscale image, each histogram (from each grid) will only contain 256 (0 255) positions representing each pixel intensity. - Further, a larger histogram is created by concatenating each histogram. If we assume we have 8x8 grids, we will have 8x8x256 = 16.384 positions in the final histogram. The final histogram represents the features of the first image.

$$D = \sqrt{\sum_{i=1}^{n} (\text{hist } 1_i - \text{hist } 2_i)^2}$$

Figure 4.1: Distance Between two histograms

Performing face recognition:

At this point, the algorithm is already trained. Each created histogram is used to represent each image from the training database. Whenever an image is input to the system, the following steps are retraced and a histogram is created for the given image.

To find the most similar image from the database to the input image, we just need to compare the two histograms and return the image with the nearest histogram.

We can use various methods to compare histograms (calculating the distance between two histograms), for example, Euclidean distance, square-chi, total value, etc. In our implementation, we use the Euclidean distance based on the following formula:

So, the calculation output is an ID from a picture with a close-by histogram. The calculated distance can be utilized as a measure of 'confidence'.

4.2.2 Detecting drowsiness

Using facial landmark detection to localize important regions of the face, including eyes, eyebrows, nose, ears, and mouth allows us the leverage to extract specific facial structures by knowing the indexes of the particular face parts: In terms of drowsiness detection, we are only interested in two sets of facial structures — the eyes. Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye and then working clockwise around the remainder of the region:

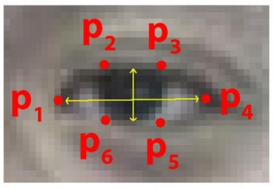


Figure 4.2: The 6 facial landmarks associated with the eye

Based on the work by Soukupová and Čech in their 2016 paper, Real-Time Eye Blink Detection using Facial Landmarks,[4] we can then derive an equation that reflects the relation between the width and height of the eye, called the eye aspect ratio (EAR):

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 4.3: The eye aspect ratio equation.

Where p1, . . . , p6 are 2D facial landmark locations.

The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, weighing the denominator appropriately since there is only one set of horizontal points but two sets of vertical points. The eye aspect ratio is approximately constant while the eye is open but will rapidly fall to zero when a blink is taking place. Using this simple equation, we can avoid image processing techniques and simply rely on the ratio of eye landmark distances to determine if a person is blinking.

To make this clear, consider the following figure from Soukupová and Čech:

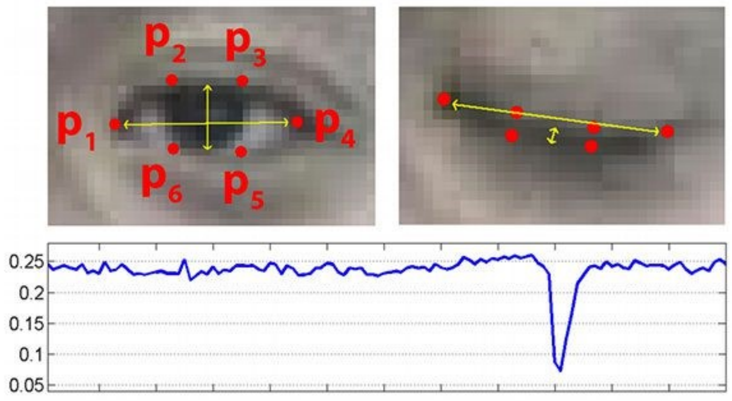


Figure 4.4: The dip in the eye aspect ratio indicates a blink

On the top-left we have an eye that is fully open — the eye aspect ratio here would be large(r) and relatively constant over time. However, once the person blinks (top-right) the eye aspect ratio decreases dramatically, approaching zero. The bottom figure plots a graph of the eye aspect ratio over time for a video clip. As we can see, the eye aspect ratio is constant, then rapidly drops close to zero, then increases again, indicating a single blink has taken place.

4.3 Face recognition steps and working

Finding all the Faces

We start by looking out for faces in an image. Initially the image is converted to grayscale as colour data isn't required to identify or recognize faces and this allows our model to reduce its computational requirements. Then the pretrained model, haarcascade, provided by OpenCV has been used to detect all the faces in the input image.

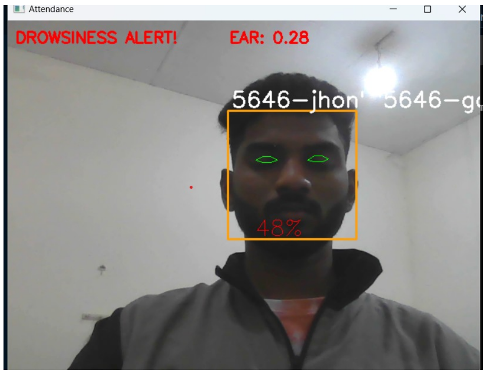


Figure 4.5: Predicting face using LBPH model

Training the model

In this step, the face image is encoded and the LBPH model trainer is used and the model is trained using the images in the dataset.

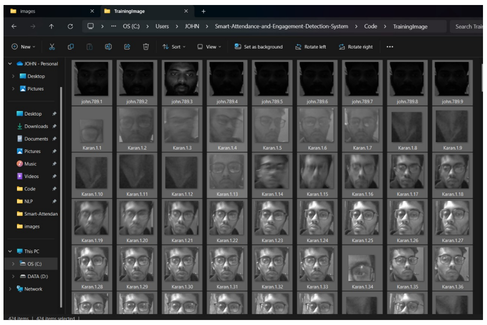
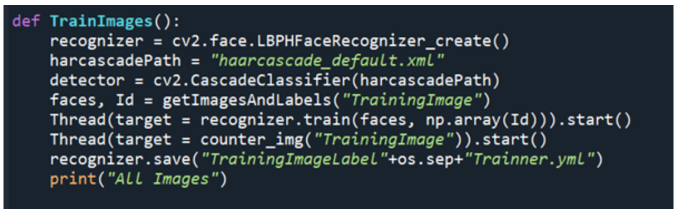


Figure 4.6: Training the model

From the figure above, it can be noticed that 30-40 images get captured and stored in the database/ local storage. These images are then used to detect and recognize the person to mark the attendance.



```
def TrainImages():
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    harcascadePath = "haarcascade_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, Id = getImagesAndLabels("TrainingImage")
    Thread(target = recognizer.train(faces, np.array(Id))).start()
    Thread(target = counter_img("TrainingImage")).start()
    recognizer.save("TrainingImageLabel"+os.sep+"Trainner.yml")
    print("All Images")
```

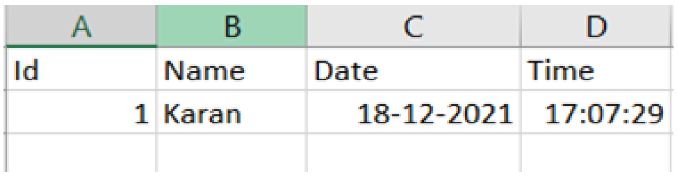
Figure 4.7: Function to train images in Python

Compare against known faces

In this step, the model trained and stored in the previous step is used. An image which is input to the model is preprocessed and face images are extracted. Then the LBPH predict function is used to identify the facial image from the dataset closest to the image which is received.

Marking attendance

The face is recognized, and the attendance is marked for the recognized person, along with a timestamp. This attendance is then stored in a CSV file



A	B	C	D
Id	Name	Date	Time
1	Karan	18-12-2021	17:07:29

Figure 4.8: Marked attendance and CSV file

4.3.1 Drowsiness detection steps and working:

step - 1: Finding all the Faces

For this, we use Dlib's face landmark estimation. The library allows us to return sixty-eight specific points (landmarks) including the upper part of the chin, the skin fringe of every eye, the inner fringe of the eyebrow, etc

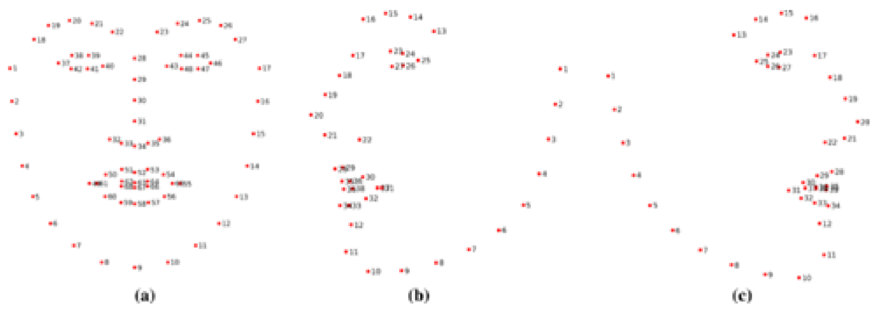


Figure 4.9: Front and side poses

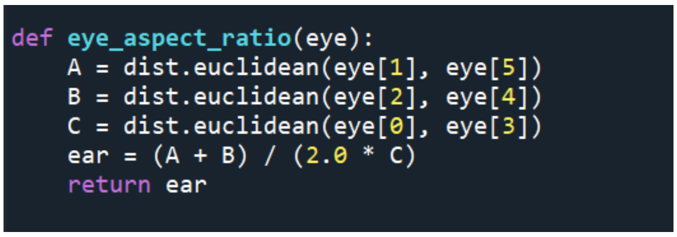
4.3.2 step - 2: Training the model

In this step, the face image is encoded and the LBPH model trainer is used and the model is trained using the images in the dataset.

From the figure above, it can be noticed that 30-40 images get captured and stored in the database/ local storage. These images are then used to detect and recognize the person to mark the attendance.

Calculating the eye aspect ratio

The above detected landmarks are then used to calculate the eye aspect ratio.



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

Figure 4.10: Calculating EAR

The Eye Aspect Ratio (EAR) is a metric used in facial landmark detection to assess the openness or closure of eyes. It is particularly useful in applications like drowsiness detection. The EAR is calculated using the coordinates of specific facial landmarks, typically the points outlining the eyes

Detecting drowsiness

If the calculated EAR is below a given threshold, a warning is displayed on the screen.

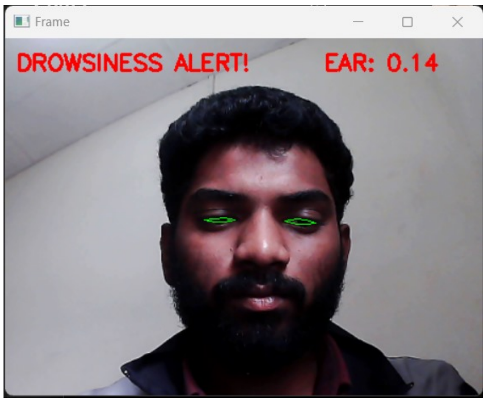


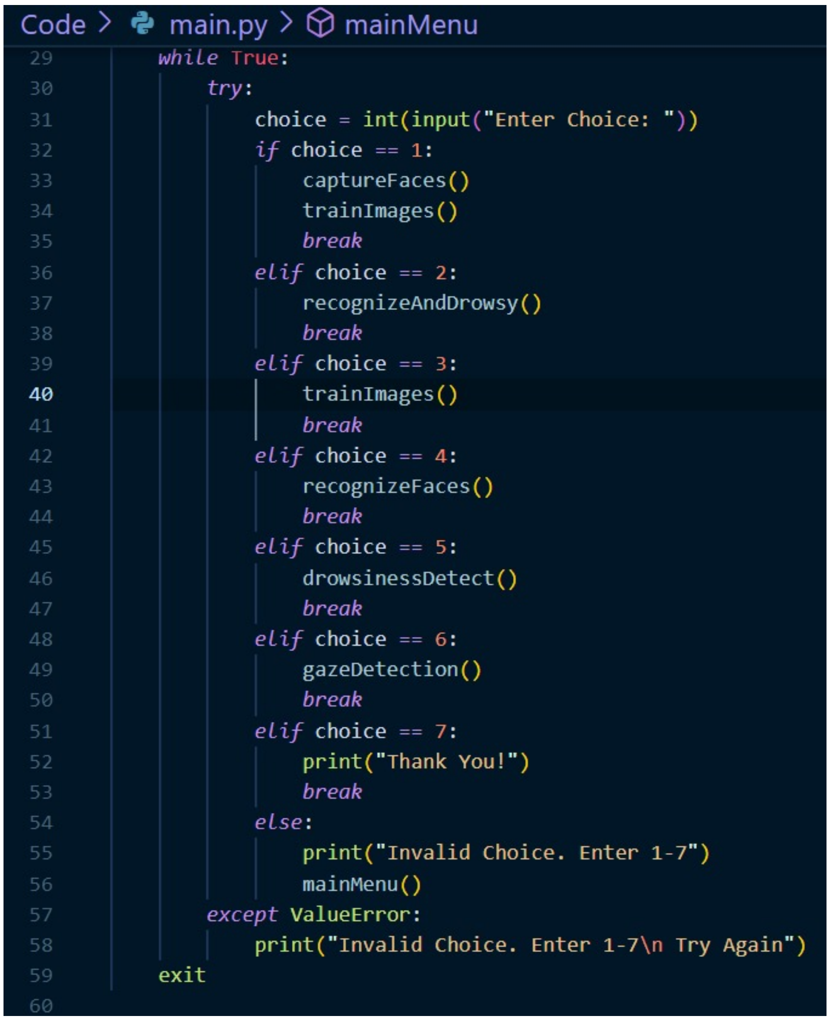
Figure 4.11: drowsiness alert

4.3.3 Task completed

We have created the models for real-time attendance and for detecting drowsiness from the captured images. All our code is written in Python. Below are the important python codes which are the core of our implementation of this software engineering project.

1. Dataset: Where all the face images are saved.
2. main.py: Main program file to run the program.
3. trainimage.py: Train the captured images and work on datasets.
4. recognize.py: To recognize and mark attendance
5. detectdrowsiness.py: To detect drowsiness in the captured images

4.4 Pseudocode



```
    while True:
        try:
            choice = int(input("Enter Choice: "))
            if choice == 1:
                captureFaces()
                trainImages()
                break
            elif choice == 2:
                recognizeAndDrowsy()
                break
            elif choice == 3:
                trainImages()
                break
            elif choice == 4:
                recognizeFaces()
                break
            elif choice == 5:
                drowsinessDetect()
                break
            elif choice == 6:
                gazeDetection()
                break
            elif choice == 7:
                print("Thank You!")
                break
            else:
                print("Invalid Choice. Enter 1-7")
                mainMenu()
        except ValueError:
            print("Invalid Choice. Enter 1-7\n Try Again")
    exit
```

Figure 4.12: pseudocode

Chapter 5

Manual Testing

5.1 Testing

5.1.1 Overview of testing

Manual testing for the Real-time Attendance and Engagement Detection project involves verifying the functionality, usability, and performance of the system through direct human interaction. Here is a suggested manual testing plan:

5.1.2 Installation and Setup:

The testing dimensions encompass different aspects, including functionality, performance, usability, and security. Each dimension addresses specific criteria to ensure a comprehensive evaluation. Functionality testing ensures that features like real-time attendance, drowsiness detection, gaze detection, and mobile-phone detection work as intended. Performance testing assesses the system's responsiveness and efficiency, especially in handling large datasets. Usability testing focuses on the user interface and overall user experience. Security testing examines the robustness of the system against potential vulnerabilities.

- Verify that the system is compatible with various operating systems (Windows, macOS, Linux).
- Check the installation process for dependencies and ensure it is well-documented.
- Confirm that the system runs on standard hardware configurations and does not have any unusual hardware requirements.

5.1.3 User Authentication:

- Test the user authentication process to ensure that only authorized users (administrators, teachers) can access the system.

- Check the password recovery mechanism to ensure it is secure and functional

5.1.4 Database Management:

- Verify that the system can connect to and interact with the database. Test the updating of the attendance database in real-time by manually marking a student as present or absent and confirming the database updates accordingly.

5.1.5 Real-time Attendance Tracking:

- Conduct tests in both offline and online settings to verify the accuracy of real-time attendance tracking.
- Manually compare the system's attendance records with the actual attendance to ensure consistency.

.

5.1.6 Facial Detection and Recognition:

- Test the system's ability to detect faces in various lighting conditions and angles.
- Verify the accuracy of face recognition by introducing variations in facial expressions, hairstyles, and accessories

5.1.7 Drowsiness Detection:

- Simulate drowsiness by closing eyes partially or completely to check if the system accurately detects and flags drowsiness.
- Verify that the drowsiness detection feature is sensitive enough to detect subtle signs of fatigue.

5.1.8 Gaze Detection:

- Test the gaze detection feature by simulating instances where a student may not be looking at the screen. .
- Verify that the system can accurately identify deviations in gaze direction

5.1.9 Accessibility Testing:

- Verify that the system complies with accessibility standards, ensuring it is usable by individuals with disabilities.

Chapter 6

Conclusion

6.1 Conclusion

the Smart Attendance System, leveraging deep learning algorithms and computer vision techniques, presents a robust solution for real-time attendance tracking and student engagement detection in both offline and online classroom settings. The transition from offline to online teaching, exacerbated by the pandemic, has highlighted the need for innovative solutions to address challenges related to attendance recording and classroom disruptions. The system, incorporating the Local Binary Patterns Histogram (LBPH) algorithm for face recognition, proves effective in automating attendance management, leading to a more productive learning environment.The proposed system goes beyond traditional attendance tracking by introducing engagement detection practices. Drowsiness detection, gaze detection, and mobile-phone detection enhance the system's capabilities to maintain the sanctity of the classroom. The utilization of facial landmark detection for drowsiness detection, specifically the computation of the Eye Aspect Ratio (EAR), adds an extra layer of sophistication to the system.Testing, conducted on diverse datasets, including freely available ones and real-world data from webcam live streams, has demonstrated the system's reliability and effectiveness. As the system undergoes implementation during this semester as part of the minor project, it lays the foundation for future enhancements and additions.

6.2 Future Scope

The Smart Attendance System has promising avenues for future development. In the upcoming semester, the project aims to extend its functionality by incorporating gaze detection and mobile phone detection. Gaze detection will contribute to understanding students' focus and attention during lectures, providing valuable insights into their engagement levels. Simultaneously, mobile phone detection will address potential distractions and enhance the overall classroom experience. Additionally, the future scope of the project involves the creation of a web interface, offering a user-friendly platform for administrators, faculty, and students to interact with the system. This interface will streamline the process of accessing attendance records, monitoring engagement metrics, and managing system functionalities.The continuous evolution of the Smart Attendance System reflects its adaptability to the dynamic educational landscape, offering a comprehensive solution for attendance management and classroom engagement. As the project progresses, it is poised to become an integral component of modern educational technology, contributing to the efficiency and effectiveness of classroom environments.

Bibliography

1. T. Godson 'Effects of Classroom Attendance and Learning Strategies on the Learning Outcome' Journal of International Education in Business (2018).

2. Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. "Face description with local binary patterns: Application to face recognition." IEEE transactions on pattern analysis and machine intelligence 28.12 (2006): 2037–2041.

3. P.E. Kekong, I.A. Ajah U. Chidiebere, 'Real Time Drowsy Driver Monitoring and Detection System Using Deep Learning Based Behavioural Approach' International Journal of Computer Sciences and Engineering (2021).

4. Soukupová, Tereza and Jan Cech. "Real-Time Eye Blink Detection using Facial Landmarks." (2016)

5. `https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b`

Appendix A

User manual

A.1 Introduction

A.1.1 Purpose of the System

The Smart Attendance and Engagement Detection System aims to provide an effective solution for tracking student attendance and engagement during lectures. Utilizing deep learning algorithms and computer vision techniques, the system ensures accurate attendance records and alerts for engagement issues such as drowsiness.

A.1.2 System Features

- Real-time attendance tracking using facial recognition.
- Drowsiness detection to alert users and maintain engagement.
- Gaze detection for monitoring student focus.
- User-friendly menu for easy system navigation.

A.2 System Requirements

A.2.1 Hardware Requirements

- Webcam or CCTV for capturing student faces.
- Computer system with sufficient processing power.

A.2.2 Software Requirements

- Python 3.x
- OpenCV

- Dlib
- Other dependencies (specified in the project repository)

A.3 Installation

1. Clone the project repository.
2. Set the 'code' folder as the working directory.
3. Install the required dependencies using the provided requirements.txt file.

```
pip install -r requirements.txt
```

A.4 Usage

A.4.1 Setting up the Camera

Ensure that a camera (webcam or CCTV) is properly installed in front of the students to capture their faces.

A.4.2 Running the System

Offline Mode

Run the `main.py` file to initiate the system in offline mode. Follow the on-screen instructions to navigate through different functionalities.

Online Mode

For online mode, ensure a stable internet connection. Follow the same steps as offline mode, and the system will adapt to online settings.

A.4.3 Menu Navigation

Use the keyboard to navigate through the system menu. Follow on-screen prompts for each functionality.

A.5 Attendance Tracking

A.5.1 Registering Users

If a user is not registered, capture images for recognition and register them using the system.

A.5.2 Capturing Images

Users can capture images for registration during the system setup.

A.5.3 Face Recognition

Utilizing the LBPH algorithm, the system recognizes registered users' faces for attendance tracking.

A.5.4 Exporting Attendance

Attendance details, including name, ID, date, and time, are exported and saved in a CSV file.

A.6 Engagement Detection

A.6.1 Drowsiness Detection

Facial landmarks and eye aspect ratio are used to detect drowsiness. Warnings are displayed if drowsiness is detected.

A.6.2 Gaze Detection

Gaze detection monitors student focus. Details regarding gaze behavior are displayed in the system.

A.7 Troubleshooting

Refer to the troubleshooting section in the project documentation for common issues and solutions.

A.8 Conclusion

The Smart Attendance and Engagement Detection System provides a comprehensive solution for efficient classroom management. Regular updates and improvements are encouraged.

A.9 Acknowledgments

We acknowledge the contributions of all developers and libraries that made this project possible.

A.10 Contact Information

For further assistance or inquiries, contact the developers:

- Konada Gopalakrishna
  - Email: konadajohn@gmail.com
  - Contact Number: 7780626003