AI Based Office Monitoring System using Python

[1]Dr. R Amutha , Prof. & Head, Dept Of ISE, AMCEC
Corresponding Author E-Mail ID: amutha.shruthi@gmail.com
[2]Chaitra S Patil, [2]Charitha S P, [2]Disha Shetty S, [2]Gagana R S
Department Of Information Science Engineering, AMC Engineering college, Bengaluru, Karnataka, India

ABSTRACT

In today's office environment, organizations face the challenge of effectively monitoring employee performance and ensuring workplace security. Companies need reliable systems to track employee attendance and at the same time they need to monitor employees and also ensure that employees remain productive, focused, and engaged during work hours. However, current or existing monitoring systems often focus on one aspect either security or performance without integrating both in the modern workplace. This paper discusses about the methodologies for facial attendance based on facial recognition, posture monitoring, distraction recognition, and integration of all of them in a single system.

Keywords: Facial recognition, Posture monitoring, Distraction recognition

1. INTRODUCTION

In today's fast-paced office environments, ensuring both security and productivity is essential for organizational success. Organizations are looking for ways to improve both these factors, and an integrated approach that combines both aspects security and performance into a seamless employee monitoring system. Face recognition, sitting posture monitoring, distraction recognition plays crucial roles in this transformation, offering innovative solutions to address security, productivity, and office management challenges.

Face recognition technology has become a key tool in securing office spaces by automating attendance tracking and managing access control. Using advanced machine learning algorithms, face recognition systems can accurately identify employees, ensuring attendance is tracked seamlessly. This technology identifies individuals based on their unique facial features, offering an efficient and secure method of access control. By utilizing image processing techniques and artificial intelligence (AI), face recognition systems can accurately identify employees, even in challenging conditions such as varying lighting or obstructions. These systems replace manual check-ins and physical badges, reducing administrative workload and eliminating human error. By integrating facial recognition into employee monitoring, companies can improve operational efficiency while maintaining high levels of security.

Another crucial element of employee monitoring is productivity. While traditional performance tracking often focuses on output and task completion, it is equally important to monitor how employees are working. Prolonged periods of sitting in improper postures can lead to discomfort and inefficiencies in workflow. Posture recognition systems address this challenge by monitoring employees sitting habits in real time and ensuring that they are maintaining optimal working positions. This paper proposes an approach to utilize pressure sensors to track body position and detect any deviations from the ideal posture. Using machine learning algorithms, these systems can analyze data and provide feedback to employees, prompting them to adjust their posture. By encouraging better posture and movement, these systems help employees maintain focus during long working hours.

Distraction recognition has become an essential aspect of human-computer interaction and employee productivity in workplace environments. With the increasing availability of machine learning and computer vision technologies, real-time monitoring systems have emerged as effective tools to assess user engagement. This paper presents an approach for detecting various states of distraction by analyzing facial and hand landmarks using the MediaPipe and OpenCV frameworks.

While these technologies or systems offer distinct benefits, the real challenge lies in integrating them into a cohesive system for employee monitoring. Traditionally, these systems have operated in isolation face recognition systems focus on security and identity verification, while posture monitoring systems focus on employee performance and distraction systems solely focuses on employees' distractions. However, in an increasingly complex and dynamic workplace, it is essential to combine these technologies to provide a comprehensive solution for both securing access and tracking employee productivity.

A unified system that combines real-time attendance tracking, secure access management, and performance monitoring offers several advantages. By integrating face recognition for access control with posture recognition for productivity and distraction recognition for better productivity, ensures that businesses can improve overall office efficiency and security, while providing a comprehensive approach to managing employee activity. This integrated system reduces the need for multiple, disparate tools and enhances the overall functionality of office management. Together, these technologies offer a comprehensive solution for office monitoring, addressing both security needs and performance tracking in one integrated system.

2. LITERATURE SURVEY

1.The paper, "A Review of Face Recognition Technology" by Lixiang Li, Xiaohui Mu, Siying Li, and Haipeng Peng, published in 2020 in IEEE Access, provides an insightful and detailed look into the world of face recognition (FR) technology. It traces the evolution of FR from its early days, when techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) were used, to modern advancements involving Support Vector Machines (SVM), Ada boost, and cutting-edge deep learning methods like Convolutional Neural Networks (CNN). The authors explore how face recognition has evolved to tackle real-world challenges, including changes in lighting, diverse facial expressions, varying head poses, and the difficulty of recognizing faces in low-resolution images. They highlight the use of popular datasets like LFW and Mega Face to test and compare algorithms, with performance being measured using metrics like accuracy (ACC) and ROC-AUC. Notably, the paper shows how CNNs and Generative Adversarial Networks (GANs) have pushed the accuracy of FR systems to levels that rival human recognition. The study emphasizes FR's wide range of applications, from enhancing security and streamlining financial processes to enabling smarter surveillance systems. Looking ahead, it discusses exciting opportunities, such as using 3D modelling for more accurate recognition, while also addressing critical concerns around bias, privacy, and ethics. Overall, the paper captures the remarkable progress and the challenges that remain in making FR technology more effective and inclusive.

2. The paper titled "Sitting Posture Recognition Systems: Comprehensive Literature Review and Analysis" by Muhammad Nadeem, Ersin Elbasi, Aymen I. Zreikat, and Mohammad Sharsheer, published in Applied Sciences in 2024, takes a detailed look at technologies designed to recognize sitting postures. It explores both traditional and cutting-edge approaches, focusing on their applications in healthcare, ergonomics, and human-computer interaction. The study reviews sensor-based methods, such as pressure sensors, strain gauges, and inertial measurement units (IMUs), along with vision-based techniques using RGB and depth cameras. It also highlights the role of machine learning and deep learning models like convolutional neural networks

(CNNs) and long short-term memory (LSTM) networks in posture recognition. Some systems showcased impressive accuracy, with decision trees reaching 99.51% and artificial neural networks (ANNs) achieving 97.78%. The authors address key challenges, such as integrating sensors effectively, ensuring real-time monitoring, and navigating environmental issues like lighting and privacy. The paper emphasizes the importance of combining multiple technologies (sensor fusion) to improve accuracy and reliability. It also highlights the potential of posture recognition in smart workplaces and healthcare, particularly for preventing musculoskeletal disorders, with insights into current advancements and future opportunities.

3. The paper "Deep Learning for Head Pose Estimation: A Survey" by Andrea Asperti and Daniele Filippini, published in April 2023, explores how computer vision systems estimate head orientation using deep learning. Head Pose Estimation (HPE) involves determining the orientation of a human head relative to a camera and has critical applications in areas such as human-computer interaction, driving assistance, surveillance, and targeted advertising. The authors categorize approaches into classical methods, segmentation-based models, model-based techniques, and deep learning-driven methods. Deep Convolutional Neural Networks (CNNs), 3D Morphable Models (3DMMs), and multi-task learning architectures are highlighted as significant contributors to modern HPE systems. They highlight how advanced models like Synergy Net and SADR Net use neural networks to predict head poses accurately, even in tough conditions like bad lighting or when parts of the face are hidden. These models can reconstruct 3D facial shapes and handle large variations in head angles. The paper also looks at the challenges of making HPE systems reliable in real-world situations. It showcases improvements in accuracy, with some models achieving errors of less than 5 degrees when predicting head orientation. This survey outlines the evolution of HPE and its growing integration into real-world intelligent systems.
contribute to advancing user-independent emotion recognition systems, bridging gaps in prior research.

3. PROPOSED METHODOLOGY

With respect to office monitoring system, various methodologies are used to enhance security, track employee attendance, and improve productivity. These methodologies often integrate various advanced technologies and monitoring tools. The proposed system, with the help of technologies such as face recognition, posture recognition, distraction recognition integrates artificial intelligence (AI) and computer vision technologies to monitor and enhance employee efficiency and well-being in a workplace setting. The methodology focuses on three key areas: facial attendance, posture monitoring, and distraction detection, all performed in real-time using webcam inputs.

1. Methodology for Attendance Monitoring System Using Face Recognition

Face recognition technology in office monitoring primarily focuses on security and attendance management. This project is a combination of machine learning and computer vision techniques to develop a robust attendance monitoring system.
This system is designed to capture, train, and recognize faces while storing attendance records in a structured format. This methodology involves four main stages: User Interface Design, Data Collection, Model Training, and Real-Time Face Recognition & Attendance Logging.

1. User Interface Design
The graphical user interface (GUI) is implemented using Python's Tkinter library. It serves as the primary interaction medium between users and the system. The interface includes:
Input Fields: Users can enter their ID and name, which are essential for associating facial data with unique identifiers.
Notification Areas: Text labels dynamically update to provide feedback to the user, such as successful image capture, training completion, or errors like invalid inputs.
Action Buttons: Several buttons are provided to perform key operations.
Clear: Resets the input fields.
Take Images: Captures and saves facial images of the user.
Train Images: Initiates the training of the face recognition model.

Track Images: Detects and identifies faces in real-time and records attendance.
Quit: Closes the application.

2. Data Collection
Data collection is a critical phase where the system uses a webcam to capture facial images. This process is triggered when the user clicks the Take Images button.
The system validates the inputs to ensure the ID is numeric and the name consists only of alphabets.
A Haar Cascade Classifier, a pre-trained model for face detection, is used to detect faces in the video feed.
Detected faces are extracted, converted to grayscale, and saved in the Training Image directory with filenames structured as <Name>.<ID>.<SampleNumber>.jpg.
Up to 60 face samples are captured per user to ensure variability and robustness during training.
This phase creates a comprehensive dataset representing different angles, expressions, and lighting conditions for each user.

3. Model Training
The Train Images button triggers the training phase. The Local Binary Patterns Histograms (LBPH) algorithm is used for face recognition due to its effectiveness in real-world conditions. The process involves:
Loading all saved images from the Training Image directory.
Converting these images to NumPy arrays.
Extracting the user IDs from the filenames and associating them with the corresponding face data.
Training the LBPH model using these inputs to create a classifier capable of recognizing stored faces.
The trained model is saved as Trainner.yml in the TrainingImageLabel directory, making it reusable for real-time recognition.

4. Real-Time Face Recognition & Attendance Logging
The Track Images button initiates real-time face recognition.
This process involves:
Video Stream and Face Detection:
The system captures live video feed from the webcam.
The Haar Cascade Classifier detects faces in each video frame.
Face Recognition:
Detected faces are compared against the trained LBPH model.
If a match is found the system identifies the user by their ID and retrieves the associated name.
Attendance Logging:
For each recognized user, the system records:
ID and Name
Date and Timestamp
The attendance is logged in a structured CSV file stored in the Attendance directory. Duplicate entries for the same user on the same day are avoided.
Handling Unknown Faces:
Faces that do not match any record in the model are labelled as "Unknown."
Images of unknown faces are stored in the ImagesUnknown directory for further analysis.
User Feedback:
Real-time notifications display the recognized user’s details or alert the presence of unknown faces.

2. Methodology for Posture Monitoring System
This project aims to develop a system that monitors employee posture in real-time using computer vision and pose detection techniques. The methodology involves capturing live video or pre-recorded footage, analyzing body landmarks, and classifying posture as either "good" or "bad" based on defined thresholds.

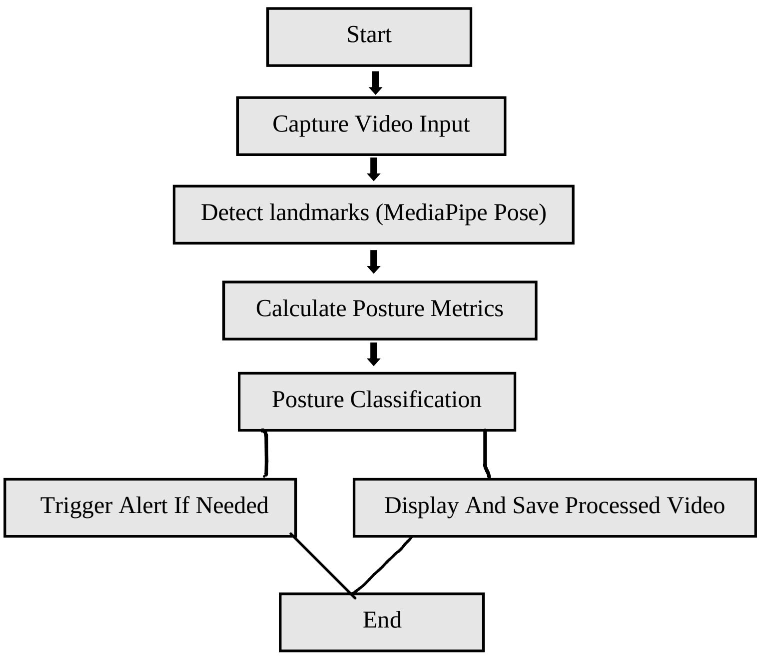

Figure 1: Flow chart of Posture Monitoring System

This system incorporates visual feedback and alerts to encourage ergonomic sitting habits, enhancing workplace productivity.

1. Setup and Initialization
Capture video frames and initialize the pose detection model.
Tools Used:
Python's cv2 library for video processing.
MediaPipe's Pose module for real-time landmark detection.
Custom utility functions for calculating metrics and triggering alerts.
Video Input:
Supports webcam (real-time monitoring).
Captures frames using cv2.VideoCapture.
Output Configuration:
Generates annotated output video (output.mp4) using cv2.VideoWriter.
Offers flexibility for custom resolutions (e.g., 1280x720) to meet display requirements.

2. Real-Time Pose Detection
Pose Model Initialization:
MediaPipe’s Pose model is initialized to detect key body landmarks.
Converts video frames to RGB format (cv2.cvtColor) before processing, as required by the model.
Landmark Extraction:
Key body landmarks such as shoulders, hips, and ears are identified for posture analysis.
Coordinates are scaled to pixel dimensions for accurate visualization and calculations.

3. Feature Extraction
Key Measurements
The following measurements are calculated to determine posture:
Neck Inclination: Angle between the left shoulder and left ear to measure neck posture.
Torso Inclination: Angle between the left shoulder and left hip to evaluate torso alignment.
Shoulder Offset: Horizontal distance between the left and right shoulders to assess alignment.
Threshold Values
Good Posture:
Neck inclination: < 45 degrees
Torso inclination: < 13 degrees
Shoulder alignment:<420 degrees
Bad Posture:
Values outside the thresholds are considered bad posture.

4. Posture Analysis
Good vs. Bad Posture
Good Posture: If neck and torso inclinations fall within acceptable thresholds, the system increments the counter for "good frames."
Bad Posture: If the inclinations exceed thresholds, "bad frames" are counted.
Posture Time (seconds) = Number of Frames / Frames Per Second (FPS)

5. Visual Feedback and Alerts
On-Screen Feedback:
Displays key metrics (e.g., angles, alignment, and posture time) in the video frame using cv2.putText.
Alerts for Prolonged Bad Posture:
If bad posture persists for over 3 minutes (180 seconds), triggers a warning via the sendWarning function.

6. Output Video Generation
Frame Annotation: Annotated frames, including landmarks, angles, and feedback, are written to an output video using cv2.VideoWriter.
Real-Time Display: The annotated video is displayed in a resizable window using cv2.imshow. Supports live monitoring or playback for later review.

This methodology ensures an efficient, scalable, and interactive solution for posture monitoring in office environments.

3. Methodology for Distraction Recognition System

The proposed system detects distraction states such as yawning, eye closure, hands on face, and face distance from the camera using real-time video processing with MediaPipe and OpenCV. The methodology consists of video input capture, landmark detection, thresholding, data logging and real-time feedback display.

1. Real-Time Video Input Capture
A webcam feed is captured using OpenCV’s cv2. VideoCapture method.
The resolution of the video feed is retrieved to scale and interpret landmark coordinates accurately.

2. Landmark Detection
Mediapipe's FaceMesh model is used to detect 468 facial landmarks for one face.
Mediapipe's Hands model is used to detect 21 key points per hand for up to two hands.
Each frame is processed through these models to identify and store landmark coordinates.

3. State Detection
Four states are evaluated using the detected landmarks:
Yawning Detection:
The vertical distance between the upper and lower lip landmarks (e.g., points 13 and 14) is measured.
This distance is normalized by the horizontal distance across the mouth (e.g., between points 78 and 308).
If the ratio exceeds a predefined threshold (thresh_yawn), yawning is detected.
Eye Closure Detection:
Distances between the upper and lower eyelid landmarks (e.g., points 160 and 144 for the left eye) are calculated for both eyes.
These distances are normalized by the horizontal width of the eyes (e.g., points 33 and 133 for the left eye).
If the average normalized distance for both eyes falls below a threshold (thresh_eye), eye closure is detected.
Hands on Face Detection:
Distances are calculated between hand landmarks (e.g., palm center, point 9) and key facial landmarks (e.g., nose, cheeks).
If any of these distances are below a threshold (thresh3), it is determined that the user's hands are on their face.
Face Distance from Camera:
The vertical (e.g., landmarks 10 and 152) and horizontal (e.g., landmarks 234 and 454) distances across the face are used to determine proximity.
If these distances are below predefined thresholds (thresh1 and thresh2), the user is considered to be too far from the camera.

4. Thresholding and Decision Making
The system evaluates all states in real-time for each frame.
If any distraction state (yawning, eye closure, hands on face, or away from the camera) is triggered, the overall state is marked as "distracted."
The detected distraction is displayed on the video feed using cv2.putText.

5. Data Logging
For each frame, the distraction state, associated parameters (e.g., yawning, eye closure), and the timestamp are recorded in a CSV file.
This structured data allows further analysis of distraction patterns.

6. Real-Time Feedback and Notification
The system overlays feedback on the video feed (e.g., "distracted") for immediate user awareness.
A background thread monitors the elapsed time and triggers an external script (read.pyw) if a distraction persists for more than 0.3 seconds.

7. System Efficiency and Robustness
To ensure real-time processing, the image is marked as non-writable during landmark detection.
Data processing and drawing functions are executed efficiently, leveraging Mediapipe's optimized models.

This methodology provides an effective way for detecting user distraction states through computer vision. The combination of Mediapipe's robust models, OpenCV's real-time processing capabilities, and a well-defined decision-making framework ensures reliable detection.

4. Methodology for Integrated Monitoring System

Combining face recognition, posture recognition and distraction recognition methodologies into a single integrated monitoring system involves:
1. Multi-Sensor Data Fusion: Data from face recognition, posture monitoring and distraction recognition systems are merged into a centralized dashboard. This integration allows for a unified monitoring approach, where both security and productivity are managed through a single system.

2. Data Synchronization: All these systems synchronize in real-time, enabling organizations to track employee attendance while simultaneously monitoring their working habits.

3. Unified Analytics and Reporting: This integrated system can generate reports that combine data from all these systems, providing insights into both employee performance.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

4. RESULT AND DISCUSSION

1. Attendance monitoring system

This paper proposes a project that implements a Face Recognition-based Attendance System using Python, OpenCV, and Tkinter. The primary goal of this system is to track attendance by recognizing faces in real-time through a webcam and matching them with the stored records. It provides an interface where users can input their ID and name, capture face images, train the model with those images, and track attendance through face recognition.

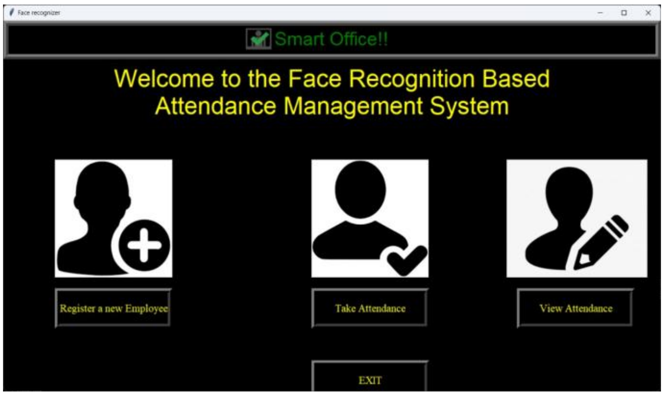

Figure 2: Dashboard For Face Recognition

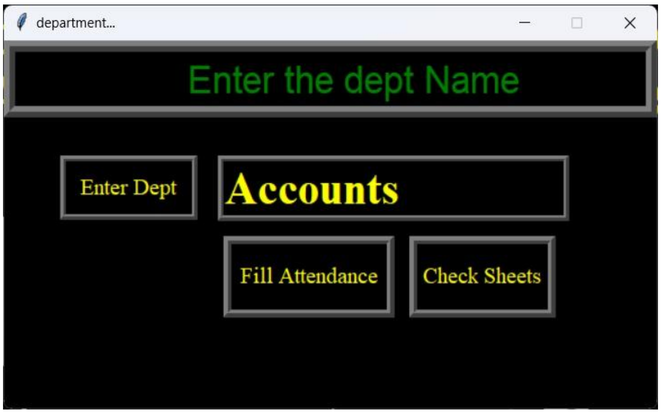

Figure 3: Fill Attendance for Department

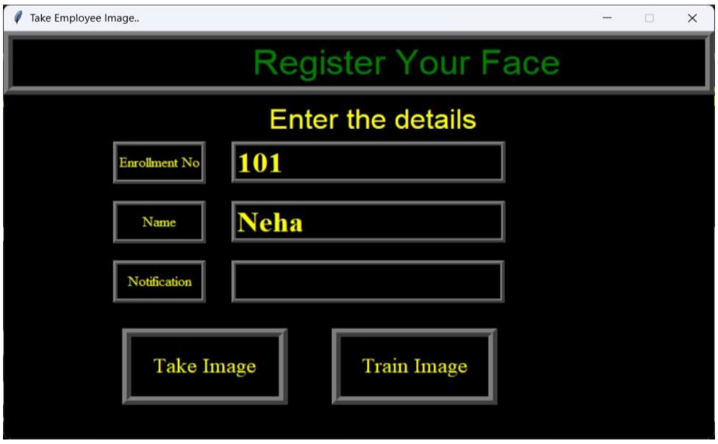

Figure 4: Face Registration

Key Features and Functionalities:
GUI Interface (Tkinter):
The user interface includes several elements: Input fields for ID and Name, labels for displaying the status and results, and buttons for operations like capturing images, training the system, and tracking attendance.
Labels and buttons are organized to guide the user through the system's operations.
Capture Images (TakeImages):
The system takes a snapshot of the user's face using the webcam, saving the captured images in a folder for training the face recognition model.
It uses OpenCV's Haar Cascade Classifier to detect faces in the captured frames.
The images are saved with a unique naming (based on ID and Name) for later use in training.
Training the Model (TrainImages):
The images captured for each person are used to train the LBPH (Local Binary Patterns Histogram) face recognition model.
Face Recognition (TrackImages):
The trained model is used to predict the identity of faces detected in real-time video feed from the webcam.
The system displays the person's name and logs attendance with a timestamp (date and time).
Attendance is saved in a CSV file, and duplicate entries are removed.
Attendance Logging:
The system prevents multiple attendance logging for the same individual in the same session. A CSV file is generated for each session to maintain a record of attendance.

Existing attendance systems that use facial recognition involve a dedicated machine with a camera, facial

recognition software, and a local database. They often do not offer an easy-to-use interface for the end user. Some face recognition attendance systems are cloud-based, where captured images are sent to a remote server for processing and recognition.
Limitations of existing methods:
1.Difficult to scale or modify.
2.Might lack a flexible or intuitive user interface.
3.Require significant setup and configuration for real-time recognition.
4.Possible privacy concerns regarding cloud storage of biometric data.
5.Potential latency issues due to the need to send data over the internet for processing.
The proposed face recognition attendance system is a significant improvement over existing methods in terms of usability, local processing, and real-time logging. Its intuitive GUI, ease of training, and flexibility make it a powerful tool for organizations or any environment requiring face-based attendance tracking. The local processing ensures that users don't face issues privacy concerns, which are common in existing systems. The provided system includes a GUI interface built with Tkinter, making it highly user-friendly, even for non-technical users.
Users can easily input their information, capture their faces, and track attendance with just a few clicks.

2. Posture monitoring system
This paper proposes a project that implements a real-time posture detection system using MediaPipe's pose estimation model. It calculates key metrics such as shoulder alignment, neck inclination, and torso inclination from detected body landmarks. The system evaluates the user's posture as either "Good" or "Bad" based on these measurements, tracks the duration of good/bad posture, and sends warnings if bad posture persists for an extended time.

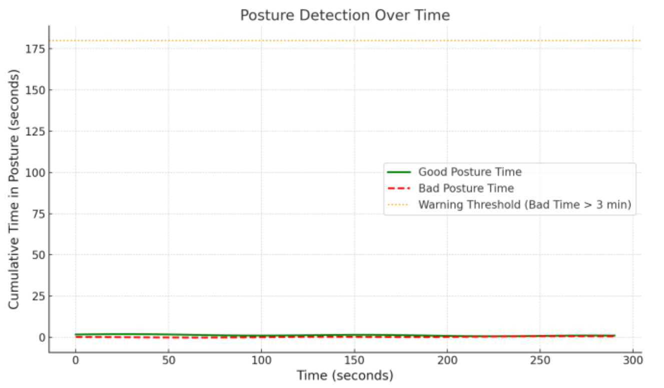

Figure 5: Graph for posture detection

The graph displays the cumulative time spent in Good Posture and Bad Posture over a 5-minute interval. The green curve represents the Good Posture Time, while the red dashed curve represents the Bad Posture Time. The orange dotted line indicates the warning threshold of 180 seconds (3 minutes) for bad posture, triggering an alert when exceeded.

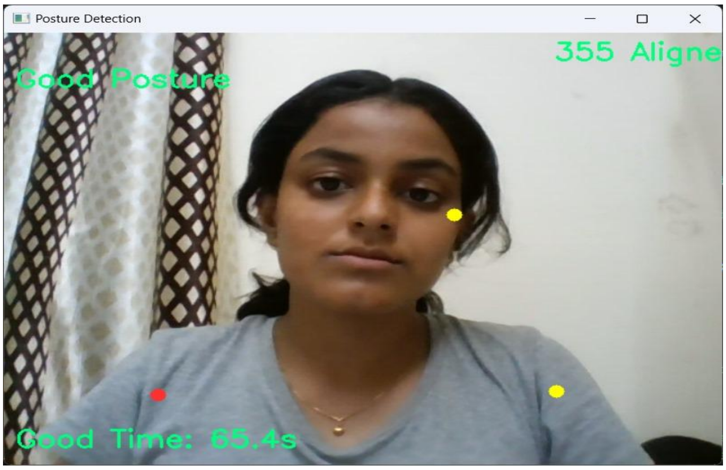

Figure 6: Good Posture

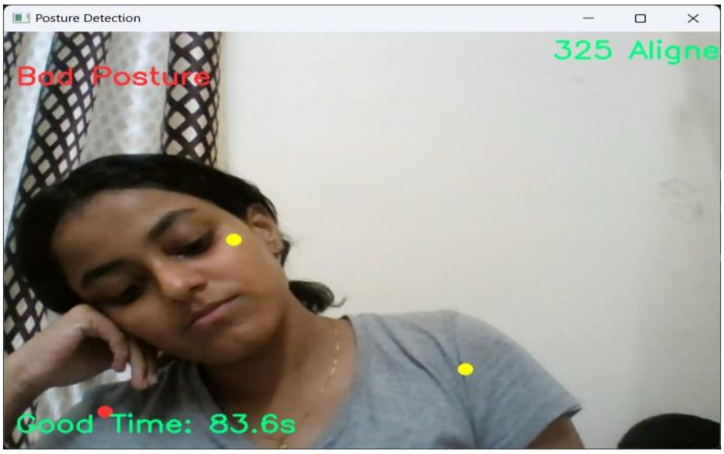

Figure 7: Bad Posture

Key Features:
Efficiency:

MediaPipe-based pose detection is highly optimized and lightweight, capable of running on low-end hardware.
The system avoids computationally heavy methods like OpenPose, making it accessible to broader users.
Customization:
Custom metrics for shoulder alignment and inclination angles offer flexibility to adapt the system to different posture standards.
Real-Time Alerts:
The addition of a warning system enhances user interaction and encourages corrective actions during prolonged bad posture periods.
Integrated Visualization:
The annotated video output with real-time metrics allows for easy analysis and understanding of posture.
Scalability:
The framework is easily extendable to incorporate additional posture metrics, improving the system's robustness for diverse use cases.

Existing posture recognition methods mostly rely on OpenPose or traditional learning models, which are computationally expensive whereas the proposed method uses MediaPipe for landmark detection, ensuring efficiency and accuracy.
Limitations of existing methods:
1.Delay in providing feedback
2.Lacks interactive alerts, reducing user engagement
3.Requires deep learning frameworks and GPU setups, making deployment challenging.

The proposed method provides immediate feedback with posture classification and alerts if bad posture persists and uses a simple python setup with MediaPipe, OpenCV and a few utility functions. The proposed posture recognition system strikes a perfect balance between efficiency, interactivity and real-time performance.

3. Distraction Recognition System
This paper proposes distraction detection system using MediaPipe and OpenCV demonstrates significant improvements over existing methods in terms of accuracy, efficiency, and real-time capability.

Key Features
Customizable Thresholds:
Fine-tuned thresholds for distraction parameters (e.g., yawning ratio, eye closure ratio, hand-to-face proximity).
Feature: Adaptable to different environments, user behaviours, and applications.
Real-Time Operation:
Processes video frames at approximately 25–30 fps on standard hardware.
Feature: High-speed performance suitable for real-world use cases like driver monitoring and productivity analysis.
Low Hardware Requirements:
Operates efficiently on consumer-grade systems without requiring GPUs.
Feature: Cost-effective solution compared to systems relying on deep learning.
Robust Landmark-Based Modelling:
Mediapipe's pre-trained models provide dense and accurate landmark detection.
Feature: Reliable performance across diverse lighting conditions, facial shapes, and hand positions.
Multi-State Detection:
Evaluates multiple states (yawning, eye closure, hand-on-face, and looking away) simultaneously.
Feature: Comprehensive assessment of distraction using multiple signals.

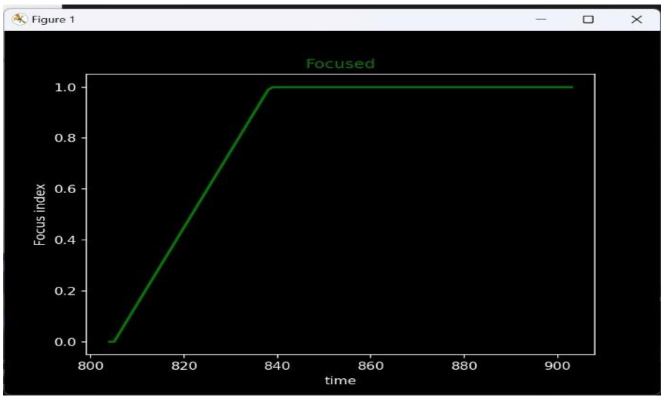

Figure 8: Focused graph

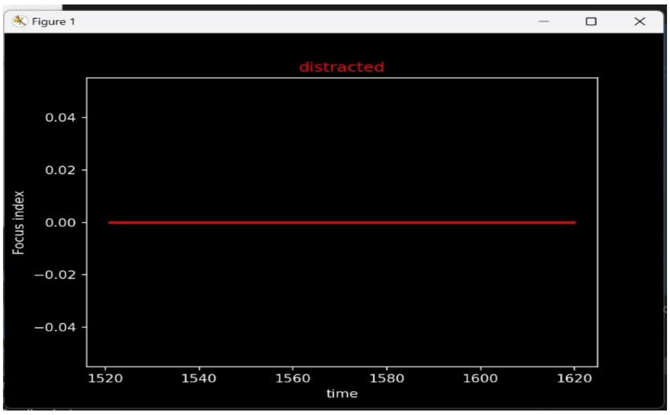

Figure 9: Distracted graph

The system displays real-time messages distracted or focused on the screen.

A real-time message "distracted" is displayed on the screen if a distraction state is detected.

Many existing traditional methods or systems rely on computationally heavy models, such as deep learning frameworks (e.g., CNNs for facial analysis) that often require dedicated GPUs to achieve comparable performance in real time.

Limitations of existing models:

1. Traditional algorithms or deep learning-based models.
2. Often focus on a single modality (face or hand).
3.Comparable accuracy but often requires more training and testing data.
4. Higher energy usage due to GPU reliance.

The proposed distraction recognition system proposes a lightweight, accurate, and real-time solution compared to existing methods. By leveraging Mediapipe's pre-trained models, it bridges the gap between the computational efficiency and high accuracy, making it ideal for practical applications like office monitoring.

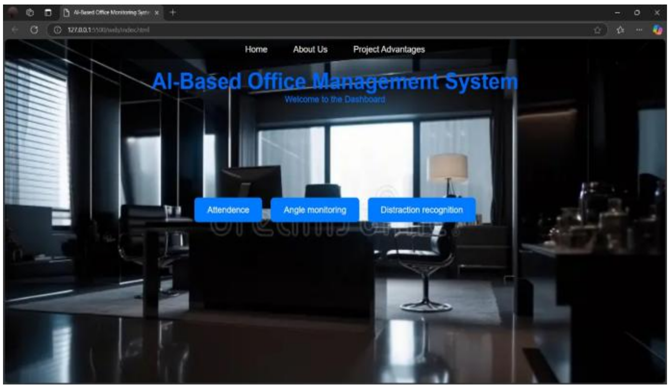

Figure 10: Home Page

5. CONCLUSION

In conclusion, the centralized dashboard integrates three new systems, attendance based on facial recognition, position monitoring system, and overall distraction monitoring providing a complete solution for a variety of business applications efficient and automated Attendance tracking using technology Streamlines, ensuring record keeping and ease of use via an intuitive GUI The position discovery system uses MediaPipe Pose to monitor and classified body posture, providing real-time feedback and continuous poor posture warnings. Overall distraction recognition involves yawning, closed eyes and hand gestures of deflections. By recognising, logging these events for the analysis, enhances the overall management of monitoring. These systems together provide robust platform for business and other environments, enhancing productivity, accuracy and user engagement improved through advanced automation and real-time analytics capabilities.

6. REFERENCES

[1611.08050] Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields (arxiv.org)

FaceNet: A Unified Embedding for Face Recognition and Clustering.

Sitting Posture Recognition Systems: Comprehensive Literature Review and Analysis (mdpi.com).

OpenCV - Open Computer Vision Library

GitHub – ageitgey /face recognition: The world's simplest facial recognition application for Python and the command line.