A Project Report
on

REAL-TIME FACIAL RECOGNITION FOR AUTOMATED ATTENDANCE APPLICATION

Submitted for partial fulfillment of award of

BACHELOR OF TECHNOLOGY

Degree in

INFORMATION TECHNOLOGY

By

Prateek Mudgal (2200640139003)
Priyanshi Chaudhary (2200640139004)
Shivam Sorot (2200640139005)

To

Under the Supervision of
Mr. Ajay Parashar

Department of Information Technology
Hindustan College of Science & Technology, Farah, Mathura

DR. A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW, INDIA
May, 2025

HINDUSTAN COLLEGE OF SCIENCE & TECHNOLOGY

FARAH, MATHURA -281122

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

Date:

CERTIFIED THAT **PRIYANSHI CHAUDHARY, PRATEEK MUDGAL & SHIVAM SOROT** HAVE CARRIED OUT THEIR PROJECT WORK PRESENTED IN THIS REPORT ENTITLED **"REAL-TIME FACIAL RECOGNITION FOR AUTOMATED ATTENDANCE APPLICATION"** FOR THE AWARD OF BACHELOR OF TECHNOLOGY FROM DR. A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW UNDER MY SUPERVISION. THE REPORT EMBODIES RESULTS OF ORIGINAL WORK AND STUDIES CARRIED OUT BY THE STUDENTS THEMSELVES AND THE CONTENTS OF THE REPORT DO NOT FORM THE BASIS FOR THE AWARD OF ANY OTHER DEGREE TO THE CANDIDATE OR ANYBODY ELSE.

MR. AJAY PARASHAR
PROJECT GUIDE

MRS. DEEPTI MITTAL
PROJECT COORDINATOR

DR. SHANKAR THAWKAR
HOD, IT

ABSTRACT

The manual participation handle in instructive educate and working environments is frequently time-consuming, error-prone, and vulnerable to control. To address these challenges, this extend presents a real-time facial acknowledgment framework for robotized participation. Leveraging progressed computer vision strategies, the framework coordinating Deep Face for Face recognition and OpenCV for facial acknowledgment. The application is planned to identify and recognize enrolled people in real-time, recording their participation effectively and accurately.

The framework captures and forms facial information each 10 minutes for a length of 30 minutes, guaranteeing reliable observing. By utilizing a strong database of pre-registered facial highlights. The application diminishes reliance on manual confirmation, streamlines participation workflows, and minimizes errors.

The venture holds critical potential for arrangement in instructive teach, corporate workplaces, and other spaces where solid participation following is basic. Future enhancements may incorporate versatility for bigger datasets and integration with cloud capacity for upgraded openness. This framework illustrates the practicality of manufactured insights and computer vision in robotizing schedule assignments, subsequently moving forward efficiency and precision.

ACKNOWLEDGEMENT

We, the members of the group working on the **"Real-Time Facial Recognition for Automated Attendance Application,"** would like to extend our sincere gratitude to everyone who has contributed to the successful completion of this project.

First and foremost, we would like to thank our **Project Mentor and Guide**, **Mr. Ajay Parashar**, for their invaluable guidance, constant support, and for providing us with the necessary resources and insights throughout the development process. Their feedback and encouragement have been instrumental in the successful implementation of this project.

We also extend our heartfelt thanks to our **Project Coordinator, Mrs. Deepti Mittal**, & our **Head of Department, Dr. Shankar Thawkar**, for their unwavering support, mentorship, and motivation, which inspired us to strive for excellence.

Our gratitude goes to **Hindustan College of Science & Technology** for providing the infrastructure and a conductive learning environment that allowed us to carry out the research and development for this project.

Last but not least, we are profoundly grateful to our **family & friends** for their unwavering support, patience, and encouragement throughout this journey. Their belief in us gave us the strength and confidence to persevere through challenges and complete this project successfully.

PRATEEK MUDGAL (2200640139003)
PRIYANSHI CHAUDHARY (2200640139004)
SHIVAM SOROT (2200640139005)

PAGE INDEX

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	Explanation
CNN	Convolutional Neural network
MTCNN	Multi-Tasking Cascaded Convolutional Neural Network
GPU	Graphics Processing Unit
I/O	Input Output
CSV	Comma-Separated Values
SQL	Structured Query Language
API	Application Programming Interface
DFD	Data Flow Diagram
RELU	Rectified Linear Unit
IDE	Integrated Development Environment
TF	TensorFlow
NP	NumPy
PIL	Python Image Library
SDLC	Software Development Life Cycle
RAM	Random Access Memory
UPS	Uninterrupted Power Supply
UI	User Interface
SSD	Solid State Drive
USB	Universal Serial Bus
RFID	Radio Frequency Identification
GPS	Global Positioning System

CHAPTER – 1
INTRODUCTION

1. INTRODUCTION

- The Real-Time Facial Recognition for Automated Attendance Application is an innovative solution aimed at detecting and identifying individuals using real-time video feeds captured by a webcam.
- The system begins by recognizing the faces of the humans and drawing bounding boxes around each detected individual.

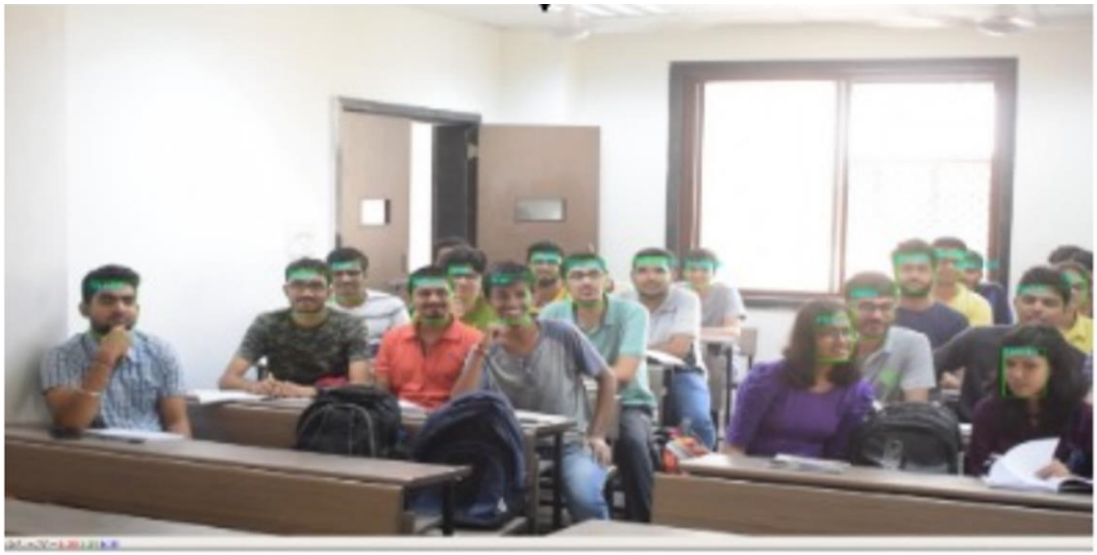


Figure 1: Real time Automated Attendance

(Reference: https://ieeexplore.ieee.org/document/9029001 Aashish Rai, Rashmi Karnani, Vishal Chudasama, Kishor Upla)[1]

- Furthermore, the system records the attendance of the detected individual and stores the information in a centralized database. This project aims to eliminate manual attendance tracking, reduce human effort, and automate the process of identification in various settings such as educational institutions and security systems.
- If a match is found, the application automatically labels the bounding box with the corresponding name of the individual.

- An end-to-end face identification and attendance approach using Multitask Cascaded Convolutional Neural Networks (MTCNN)[2], which processes the CCTV footage or a video of the class and marks the attendance of the entire class simultaneously.

1.1. INTRODUCTION TO PROJECT

Problem Statement-

- In the traditional system, the teacher calls out each student's name or roll number and makes a physical attendance sheet, which is very time consuming.
- Students who arrive late to class receive full attendance, which is unfair to those who attend the entire lecture.
- The traditional system is prone to human error.
- A major issue in the traditional system is proxy attendance.
- Teachers lose valuable class time while taking attendance.
- The traditional method involves a lot of paperwork, contributing to environmental.

Solution-

- Real-time facial recognition-based attendance.
- Eliminates manual input and improves accuracy.
- Contactless attendance marking.
- Secure, efficient, and scalable.
- Data is stored digitally in an Excel sheet for easy management.
- The system captures and processes images periodically (e.g., every 30 minutes).

1.2. OBJECTIVES

1. Eliminate the need for paperwork by digitizing attendance records.
2. Automate the attendance marking process by uploading attendance data directly to the system.
3. Minimize the wastage of time that occurs during manual attendance.
4. Face dataset collection: Gathering a diverse dataset of facial images to train the recognition model.
5. Real-time face recognition: Implementing the trained model to perform face recognition in real time using live video feeds or static images.
6. Provide real-time information about individuals detected using a webcam.
7. Ensure accurate and reliable identification of individuals using advanced AI techniques.

1.3. SCOPE

- **Security:** Ensuring the protection of personal data and compliance with privacy regulations
- **Ease of Taking Attendance:** The system simplifies attendance processes in crowded environments by allowing quick, real-time recording without disrupting the flow of the class.
- **Seamless Integration:** Ensure compatibility with existing educational management systems and centralized databases for efficient data handling.
- **Reporting:** Automated generation of attendance reports and analytics for administrative use.

1.4. FEASIBILITY STUDY

1.4.1. Technical feasibility- Technical feasibility assesses if the required technology is available and capable of supporting the project's development and

goals. It involves determining whether the necessary tools and algorithms can be implemented effectively for real-time facial recognition.

Technologies:

- **OpenCV** [3]**:** A robust computer vision library that provides tools for face detection and preprocessing tasks, such as image manipulation and scaling.
- **DeepFace** [4]**:** For deeper face recognition capabilities, DeepFace we are using. This model enables high-accuracy facial recognition by mapping faces into a high-dimensional space, allowing precise identification and differentiation between faces.
- **Python:** The primary language for development due to its rich libraries and support for machine learning, image processing, and server-side programming.
- **Streamlit (Web Framework):** Streamlit, an intuitive and lightweight Python framework, is used to build the user interface of the system. This interface enables administrators to efficiently manage attendance data, view records, add or delete users, and monitor attendance—all within an interactive and user-friendly dashboard that enhances the overall user experience.
- **MTCNN (Multi-Task Cascaded Convolutional Neural Networks):** MTCNN is used in the face detection pipeline to locate and align faces within captured images, enhancing the quality of data fed to DeepFace for recognition.

Hardware Requirements:

- **Cameras:** A high-definition webcam with at least 720p resolution for clear image capture; an IP camera can also be used if a wider area needs to be covered.

- **Computer (PC):** Preferably with a multi-core processor, such as Intel i5 or above, or AMD Ryzen 5, to handle real-time video processing and facial recognition tasks.
- **RAM:** Minimum 8GB RAM for smooth operation; 16GB or higher is recommended for faster processing.
- **Storage:** 256GB SSD for faster data read/write; SSD storage improves system speed, especially when handling large image datasets.
- **Wi-Fi/Internet Connection:** Required if data is stored or accessed on a remote server, or for remote access to the application.
- **UPS (Uninterruptible Power Supply):** For System Continuity and Stability.

1.4.2. Operational Feasibility

- Operational feasibility examines if the project will function effectively in its intended environment, meeting user needs and being sustainable over time with minimal intervention.
- Automation and Ease of Use:
    - The system eliminates the need for manual attendance, reducing errors and ensuring reliability. By setting regular scans (e.g., every 30 minutes), the system automates attendance logging with minimal human oversight.
- User Interface (UI):
    - To facilitate administrators' access to attendance logs, configuration of settings, and record management, the system should have an easy-to-use interface.
- Integration with Current Frameworks:
    - A database may be used to hold the attendance data, making it simple to export to Excel and other formats. This makes the

system appropriate for offices or educational institutions by facilitating reporting and integrating with other administrative tools.

1.4.3. Economic Feasibility

- Economic viability assesses if the project's financial advantages exceed its drawbacks. This covers the startup fee up front, continuing operating expenses, and long-term savings and advantages.
- Initial Setup Costs:
  - Hardware Investment: In order to operate the system, servers and cameras are necessary. The size and needs of the institution might affect these upfront expenses.
  - Software and Licensing: Open-source libraries, such as OpenCV, lower software expenses.
  - Development and Deployment Costs: It is important to account for the expenses related to software development, model training, and system deployment.
- Ongoing Costs:
  - Maintenance: In addition to routine software upgrades for security and feature enhancements, regular maintenance is required for cameras and hardware.
  - Operational Costs: Electricity and server hosting expenses should be taken into account, particularly for bigger organizations.
- Long-Term Savings:
  - Decrease in Manual Labor: The method saves instructors or administrators who are in charge of attendance a significant amount of labor costs by doing away with manual attendance monitoring.

CHAPTER – 2
LITERATURE REVIEW

2.1. LITERATURE REVIEW

- A game-changer in contemporary systems, facial recognition technology is widely used in industries including corporate settings, education, healthcare, and security. This technology offers a non-intrusive and effective substitute for conventional techniques by allowing the identification and verification of people based on their distinctive face traits.
- Conventional methods such as cards-based systems, fingerprint scanners, and human roll-calling are still widely used in the context of attendance systems. These approaches do, however, have a number of drawbacks. For instance, manual attendance takes a lot of time and is prone to mistakes like unintentional omissions or dishonest methods like proxy attendance. However, card-based systems frequently experience problems with lost, stolen, or damaged cards, which adds to the expenses and administrative strain. Similar to this, fingerprint scanners create hygiene issues despite their relative reliability, particularly in the post-pandemic era when touchless solutions are becoming more and more important. Additionally, these scanners are prone to deterioration over time, which might affect their effectiveness and longevity.
- By providing a contactless, automated solution, facial recognition—powered by developments in artificial intelligence and machine learning— has overcome several of these issues. Deep learning algorithms are used by contemporary face recognition systems, including those constructed using the DeepFace library, to evaluate and encode facial traits into distinct data points. To verify a person's identification in

real time, these data points are compared to profiles that have already been saved in a database. This method is more sanitary and practical since it not only improves accuracy and speed but also does away with the requirement for physical contact.

- These systems can adjust to a variety of situations, including variable lighting, perspectives, and minor changes in a person's look, thanks to the usage of neural networks and sophisticated algorithms. Some difficulties nevertheless exist in spite of these skills. The efficacy of the system can occasionally be diminished by changes in facial expressions, changes in look brought on by age or accessories, and environmental variables like dim lighting or background clutter. These problems show how important it is to do ongoing research and development in order to increase system resiliency and optimize algorithms.
- The use of face recognition technology for tracking attendance has grown in popularity among businesses in a variety of industries in recent years. While businesses use technology to increase operational efficiency and streamline personnel management, educational institutions seek to improve accountability and decrease human mistakes. This increasing use highlights face recognition's promise as a dependable, scalable, and cutting-edge attendance system solution, opening the door for more advancements in the industry.

2.2. REQUIREMENT ANALYSIS

2.2.1. Hardware Requirement (Minimum Requirement)

Component	**Specification**
Processor (CPU)	Intel i5/Ryzen 5 or above

Camera	720p or above (preferably HD or higher for better accuracy)
RAM	8GB or above
Storage (SSD)	256GB SSD or above
Keyboard	Standard USB or wireless keyboard
Mouse	Standard USB or wireless mouse
Cables	USB cables for camera and peripherals, network cables
Network	Stable Wi-Fi or Ethernet connection for data transfer

2.2.2. Software Requirement

Component	**Specification**
Programming Language	Python 3.x
Libraries	OpenCV, NumPy, Pandas, DeepFace, MTCNN, Matplotlib
Framework	Streamlit (for web application development)

Operating System	Windows 10 or above, Linux, macOS

2.2.3. Human Resources

- System development software engineers.
- Specialists in machine learning for optimization and training.
- IT assistance with hardware configuration and upkeep.

2.3. EXISTING ATTENDANCE SYSTEMS AND THEIR . CHALLENGES

- In businesses, educational institutions, and workplaces, managing attendance is an essential responsibility. Numerous solutions have been created over time to improve the effectiveness and dependability of attendance tracking. Every system, though, has a unique combination of drawbacks and difficulties. A thorough explanation of the current systems and their issues may be found below.

2.3.1. Manual Attendance Systems (Paper-Based Roll Calls)

- Among the oldest and most conventional ways to track attendance are manual attendance systems, sometimes known as paper-based roll calls. This method, which is popular at schools, universities, and small businesses, involves a manager, instructor, or teacher manually calling out names from a list to verify that people are present. The official attendance record is created by recording the replies on paper or in a register.

How It Works:

- The person in charge of attendance (such as a teacher or supervisor) starts the procedure by creating an attendance sheet or roll call list with everyone's names on it. One by one, names are called out during a session or at the start of a workday, and participants vocally acknowledge their

attendance. On the sheet, each response is indicated as "present," while those that don't reply are indicated as "absent."

Challenges of Manual Attendance Systems:

- **Time-Consuming:** Time is lost during sessions when attendance is manually marked since it is slow, especially in big groups.
- **Prone to Errors:** Missed names and duplicate entries are frequent mistakes.
- **Fraudulent Practices:** In manual systems, proxy attendance—where one person logs attendance for another—is a serious problem.
- **Record Management:** Keeping attendance records on paper is laborious, necessitates physical storage space, and is prone to damage or loss over time.

2.3.2. Biometric Attendance Systems

By identifying people by their distinct physical or behavioral characteristics, biometric attendance systems are commonly employed in workplaces, schools, universities, and enterprises to track attendance. To verify a person's identity and record their attendance, these systems use technology such as speech patterns, iris recognition, face recognition, and fingerprint scanning.

How It Works:

- Unique bodily characteristics, such fingerprints, retinal patterns, or palm prints, are used by biometric attendance systems to identify people and track their attendance. This is how they operate:

1. **Enrollment:** A biometric scanner is used to collect each user's biometric information (such as a fingerprint or retinal scan), which is then saved in a database as a digital template.
2. **Verification:** The device reads a user's biometric trait and transforms it into a digital representation when they interact with it.

3. **Matching:** The database's stored template is compared to the scanned data.

   The system verifies the identification if a match is discovered.

4. **Attendance Logging:** The system automatically records the user's attendance along with a timestamp when it has been validated.

Challenges of Biometric Attendance Systems:

- **Hygiene Concerns:** Touching shared devices is a big problem in the post-COVID-19 world, which makes fingerprint-based solutions less appealing.
- **Hardware Wear and Tear:** Due of their susceptibility to wear and tear, biometric scanners require regular maintenance or replacement.
- **Accuracy Issues:** The accuracy of fingerprint scanning can be impacted by things like age, accidents, and damp or filthy fingers.
- **Cost:** Smaller firms are less able to use these systems due to their high installation and maintenance expenses.

2.3.3. RFID-Based Attendance Systems

In order to expedite the attendance process, schools, colleges, companies, and other institutions frequently utilize RFID (Radio Frequency Identification) attendance systems. These systems make use of RFID technology, which automatically recognizes and tracks tags affixed to items or people using electromagnetic fields. RFID tags, which are usually integrated in ID cards or key fobs, are given to specific persons in the context of attendance. RFID scanners positioned at specified entrance points may read the unique identifying information included in these tags.

How It Works:

1. **Tag Assignment:** Every person is given an RFID tag with an antenna and microchip. This tag contains a special ID that is connected to the person's database attendance record.

2. **Scanning Process:** The tag is scanned by radio waves as the person gets close to an RFID reader. The unique ID included in the tag is read by the scanner.
3. **Database Integration:** The ID of the tag is transmitted by the RFID reader to a central system, where it is compared to the relevant database record.
4. **Attendance Marking:** The system records the person's attendance and a timestamp once the ID has been verified.

Challenges of RFID-Based Attendance Systems:

- **Tag Loss or Damage:** Attendance tracking may be disrupted if RFID tags are lost, stolen, or broken.
- **Unauthorized Use:** It is possible for someone to exchange tags, which might result in phony attendance.
- **Initial Setup Cost:** The initial setup cost, which includes readers and tags, might be costly even though operating expenditures are minimal.

2.3.4. Smart Card Attendance Systems

One popular technique for controlling attendance in workplaces, educational institutions, and other organizational settings is the use of smart card attendance systems. These systems make use of smart cards that include a magnetic strip or integrated microchip that holds each person's unique identifying information. To record attendance, the smart card is swiped or scanned on a suitable reader. This system's simplicity and convenience of use have made it popular, but like other systems, it has drawbacks and restrictions.

How It Works:

- **Card Issuance**: Every person receives a smart card with a special identification number on it. Usually, the card is connected to a centralized database that houses attendance and personal data.

- **Attendance Marking**: The person must swipe or scan their card on a card reader in order to record their attendance. The reader instantly registers the attendance after recognizing the distinct ID on the card.
- **Data Synchronization**: The central database, which stores the attendance data for later use or reporting, is synced with it. Administrators can use this data to create reports, handle payrolls, and track attendance patterns.

Challenges of Smart Card Attendance Systems:

- **Card Dependency:** Having the smart card is a requirement for attendance.

  Attendance problems arise from lost or forgotten cards.

- **Maintenance Costs:** Operating costs are increased when lost or damaged cards need to be replaced.
- **Fraudulent Practices:** Cards may be abused or exchanged between people, much as RFID.
- **Hardware Reliability:** Over time, card readers may experience hardware malfunctions that are inconvenient.

2.3.5. Cloud-Based Attendance Systems

Cloud-based attendance systems are cutting-edge, contemporary solutions that effectively manage attendance by utilizing cloud computing technologies. Instead, then using local devices or conventional systems, these solutions store, process, and manage all attendance records on cloud servers. To record attendance data in real time, these systems may interface with a variety of gadgets, including RFID readers, biometric scanners, and smartphone apps. Authorized users can then access the data remotely for analysis, reporting, and storage after it has been uploaded to a centralized cloud server.

How It Works:

The attendance information in a cloud-based system is gathered via mobile applications, RFID, QR codes, or biometric techniques. Following its acquisition, the data is sent to the cloud for processing and storage. Administrators and HR staff, for example, may simply manage and track attendance by using a web portal or mobile application to view the data from any place. Features like real-time reporting, analytics, and the capacity to automatically issue attendance certificates or reports are frequently included in these systems.

Challenges of Cloud-Based Attendance Systems:

- **Internet Dependency:** Any network outage can affect the functioning of these systems, which depend on a steady internet connection.
- **Data Security:** Concerns regarding data breaches and illegal access arise when sensitive data is stored on the cloud.
- **Recurring Costs:** Over time, cloud infrastructure maintenance and subscription expenses can add up.
- **Hardware Compatibility:** Purchasing suitable hardware may be necessary for integration with several devices.

2.3.6. Mobile-Based Attendance Systems

Because smartphones and mobile applications are so widely used, mobile-based attendance systems have become quite popular in recent years. By automating attendance marking, these systems take use of mobile devices' capabilities, increasing accessibility and efficiency. Depending on the use case, mobile attendance systems can operate in a variety of ways, including via Bluetooth, QR code scanning, and GPS tracking.

How It Works:

- Using mobile devices, such as smartphones or tablets, to record a person's attendance is the fundamental purpose of a mobile-based attendance system.

1. **GPS-based Attendance:**
    - When a user checks in, GPS-based technologies track the mobile device's position. The user's physical presence at the designated place (such an office or classroom) is verified by the system.
    - When the system finds the user's GPS coordinates inside a predetermined geofenced region, the attendance is recorded.
2. **QR Code Scanning:**
    - In systems that employ QR codes, the user scans a code that is presented in a certain spot, causing the system to record the user's attendance.
    - The user simply opens the mobile app, scans the code, and the system records its presence in real time.
3. **Bluetooth-based Attendance:**
    - Bluetooth technology can be used to track a mobile device's proximity to a Bluetooth-enabled device (like an attendance beacon placed in the classroom or office).
    - Once the mobile device comes within range of the beacon, attendance is automatically recorded, reducing the need for manual intervention.

Challenges of Mobile-Based Attendance Systems:

- **Device Dependency:** Users must have a smartphone with sufficient battery and storage, which may not always be feasible.
- **Location Spoofing:** GPS-based attendance systems are vulnerable to location spoofing, leading to fraudulent entries.

- **Privacy Concerns:** Tracking location or requiring access to personal devices raises concerns about user privacy.
- **Network Requirements:** A reliable internet connection is necessary for real-time synchronization.

2.4. NEED OF THE PROJECT

The Real-Time Facial Recognition-Based Attendance System solves several problems that are associated with traditional attendance management systems, marketing a modern and effective alternative.

Manual Logging Challenges:

- **Time Consuming:**
    - Conventional attendance recording, either on paper or digitally via biometrics, might prove to be a lengthy process for staff and students/employees.
- **Attendance Excess Issues:**
    - In frameworks like our college's SIM, un-updated participation records inside a particular time allotment result in excesses. In this manner, powers instructors to physically oversee and upgrade participation, driving to wasteful aspects and inaccuracies.
- **Inaccurate Rate Calculations:**
    - Current frameworks translate participation supported by an ingeneral rate: understudies with destitute participation within the past few months moreover have issues recovering, indeed on the off chance that they have been going routinely for the most recent month.

Problems with the Existing System:

- **Time Delays:**
  - Manual section leads to noteworthy delays in recording and overhauling attendance.

- **Inaccuracy and Intermediary Attendance:**
  - Errors and deliberateness control, such as intermediary participation, compromise the unwavering quality of manual systems.
- **Data Administration Challenges:**
  - Managing, recovering, and analyzing participation information for long periods or expansive numbers of people is awkward and inclined to errors.
- **Lack of Scalability:**
  - Current frameworks battle to productively handle participation over bigger organizations or numerous locations.

Why the Proposed Framework is Needed:

- The Real-Time Facial Recognition-Based Participation Framework overcomes these issues by:
- Automating participation stamping to spare time and progress accuracy.

Storing information carefully for consistent administration and analysis.

Providing a versatile arrangement reasonable for expansive organizations.

- Enhancing security by killing intermediary participation and guaranteeing solid identification. This framework altogether diminishes the regulatory burden, guarantees precise participation, and makes a user-friendly.

2.5 LANGUAGE & TOOLS USED

To guarantee the specialized achievability and viability of the Real-Time Facial Recognition-Based Participation Framework, a combination of programming

dialects, libraries, and systems has been utilized. These instruments give the essential capabilities for confronting location, acknowledgment, and the administration of participation data.

2.5.1. Technologies Used

OpenCV

- OpenCV, an effective computer vision library, is indispensable to the systems confronting discovery and preprocessing assignments. It empowers operations such as picture control, scaling, and highlight extraction, laying the establishment for solid real-time facial recognition.

DeepFace

- For improved facial acknowledgment capabilities, DeepFace is utilized. This show maps faces into a high-dimensional space, empowering exact distinguishing proof and separation between people. DeepFaces vigorous calculations guarantee tall exactness in recognizing faces, indeed in challenging conditions.

Python

- Python serves as the essential programming dialect for this framework, chosen for its broad library biological system and flexibility. Its bolster for machine learning, picture preparation, and server-side advancement makes it perfect for building a comprehensive real-time facial acknowledgment system.

Streamlit

• Streamlit, an intuitive and lightweight Python framework, is used to build the user interface of the system. This interface enables administrators to efficiently manage attendance data, view records, add or delete users, and monitor attendance—all within an

interactive and user-friendly dashboard that enhances the overall user experience.

2.5.2. MTCNN [5] (Multi-Task Cascaded Convolutional Neural Networks)

MTCNN is executed within the confront discovery pipeline to find and adjust faces inside captured pictures. Its capacity to identify faces with guarantees that high-quality information is nourished into DeepFace for precise acknowledgment, improving the systems by and large execution.

2.6. PROPOSED SYSTEM

- The proposed model puts forward the automation of participation marking using facial recognition technology as a secure, efficient, and scalable solution. This system eliminates manual input, reduces errors, and guarantees contactless checks, thus making it apt for modern digital setups where contactless processes are the norm.
- Key Features:
    - Real-Time Facial Recognition: Spot and Recognize faces through live camera feeds instantly. Contactless Stamping: Does away with the need for physical contact, hence ensuring cleaner and more convenient logging.
    - Accurate and Secure: Improves the accuracy of attendance records while ensuring data security.
    - Digital Data Management: All attendance records are stored in a structured Excel sheet, enabling easy access, review, and control.
    - Periodic Image Training: The system captures an image periodically and processes it (e.g., every 30 minutes) to improve the attendance record.

2.6.1. System Workflow:

- **Face Detection:**
    - The system applies the camera to detect faces from the video stream in real time.

- **Face Recognition:**
    - It matches the detected faces with those already stored in a database to identify them.
- **Attendance Marking:**
    - On successful identification, the attendance for that person is automatically recorded in an Excel sheet.
- **Data Storage:**
    - To facilitate simple retrieval and analysis, attendance records are time-stamped and arranged in an Excel file. This solution streamlines attendance management; market it as a cutting-edge method of recording attendance for workplaces, educational institutions, or any other industry where precise attendance monitoring is required.

CHAPTER – 3
PROJECT DESIGN

- The project's goal is to use facial recognition technology to automate the attendance procedure. Through real-time facial scanning, this system will enable users to indicate their attendance with ease and accuracy, offering a frictionless and effective solution. The program is intended to work in settings like offices, conference rooms, and schools where attendance is often recorded by hand or using other ineffective techniques.
- The system will consist of several modules working together, including:
- **Facial Recognition Module**: In order to verify identification, this will take and process face photos in real-time and compare them to database entries that have been recorded.
- **Database Management**: This module allows for the tracking of user presence by storing user information, attendance records, and face data.
- **User Interface (UI)**: a user-friendly interface that makes it easy for administrators and users to monitor attendance records, change user information, and interact with the system.
- **Attendance Log**: Real-time tracking and management will be possible since the attendance data will be updated immediately.

3.1. WORKFLOW

To guarantee accurate and efficient attendance marking, the Real-Time Facial Recognition for Automated Attendance Application [6] adheres to a methodical approach. A thorough description of each stage in the procedure is provided below:

1. Capture User Image

- The system starts by taking a picture of the user's face. A linked camera automatically recognizes a user when they enter a specified space, such a conference room, workplace, or school. High-definition cameras (720p or higher) are advised to guarantee that the image quality is adequate for precise facial recognition; the procedure is completely contactless, removing the need for manual intervention; and the camera is positioned carefully to guarantee that it takes a clear picture of the user's face.
- Since it supplies the raw data needed for facial recognition, this stage forms the basis for the next procedures.

2. Pre-Processing

- The system pre-processes the face picture once it is taken to improve its quality and analytical appropriateness. To guarantee that the algorithm can reliably recognize distinct face characteristics in every setting, preprocessing is an essential step.
- Key actions in this stage include:

- **Lighting Adjustment**: Normalized brightness and contrast levels provide constant image quality even in overexposed or low-light conditions.

- **Noise Reduction**: To enhance clarity, unwanted noise is eliminated, such as grainy textures.

- **Face Detection**: The face, or region of interest, is separated from the rest of the picture by the system.

- **Image Resizing**: To ensure consistency across all processed images, the image is scaled to meet the input specifications of the facial recognition algorithm.

By doing these steps, the system reduces mistakes brought on by changes in background interference, camera quality, or illumination.

3. Facial Feature Extraction

- Following pre-processing, the system uses sophisticated face recognition algorithms, such DeepFace, to extract distinctive facial traits from the image. In this stage, many features of the user's face are analyzed and encoded, such as:

- **Facial Landmarks**: Characteristics such as the separation between the eyes, nose shape, jawline, and lip contour.
- **Textural Patterns**: Skin tone, hairline, and other minute characteristics that contribute to the face's individuality.

- **Pose and Orientation**: To guarantee precise recognition, the algorithm takes into consideration minute changes in head tilt or facial expression.

- The user's distinct digital signature is created by converting the retrieved features into a feature vector, which is a numerical representation. After then, this feature vector is compared to the current database.

4. Matching with Database

- This stage involves comparing the feature vector produced by the acquired image with the face data that has already been saved in the database. The feature vectors and associated information (such as names and IDs) of registered users are contained in the database.

- **Similarity Check**: A similarity score between each feature vector in the database and the collected feature vector is determined by the system.

- **Threshold-Based Matching**: The system recognizes the user as a match if the similarity score is higher than a certain threshold. If not, the face has the label "unrecognized."

- **Efficient Search**: Using methods like cosine similarity or Euclidean distance, modern algorithms make sure that the matching process is quick, even for enormous databases.

Accurately recording attendance and confirming the user's identification depend on this step.

5. Attendance Marked

- The system automatically logs the user's attendance in the database if a match is discovered. By doing this, human entry is no longer necessary, and the possibility of mistakes or inconsistencies is decreased.
- Information like the user's name, ID, date, and time of entrance are all included in the attendance record.

- The system may optionally alert managers to take additional steps, such adding a new user or manually confirming the person's identity, if no match is discovered.

This step's automation simplifies attendance management and saves administrators and user’s time.

6. Real-Time Update

- Teachers, supervisors, and human resources staff are among the approved persons who have access to the real-time updated attendance log. Transparency and instant access to data for evaluation or reporting are therefore guaranteed.

- **Database Update**: A central database that is accessible from several devices safely houses the attendance data.

- **Report Generation**: The system may provide attendance reports for a department, person, or time period, offering important information about attendance trends.

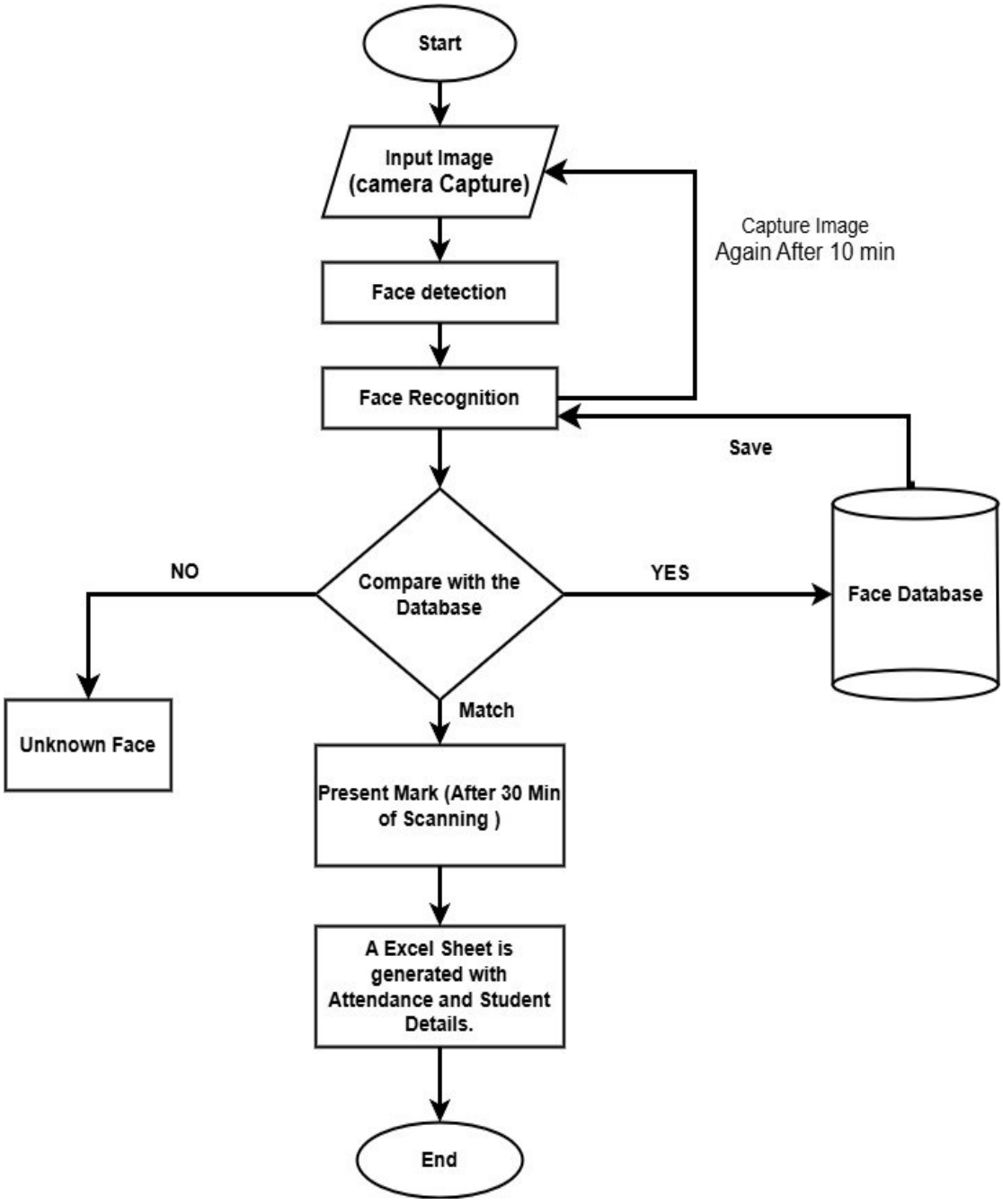


Figure 2: Workflow of the project

3.2. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) MODEL

An organized method for creating, testing, and implementing software applications is the Software Development Life Cycle (SDLC). By breaking down the software development process into discrete stages, it guarantees that the project will be finished effectively, satisfy the specifications, and be delivered on schedule with the fewest possible risks. Every stage is essential to the project's overall success and entails particular tasks to produce the intended results.

We have selected the Iterative Waterfall Model [6] to direct the development of our Real-Time Facial Recognition for Automated Attendance Application. The steps of the SDLC are explained in full below, along with why this model is appropriate for our project.

Phases of Iterative Waterfall Model

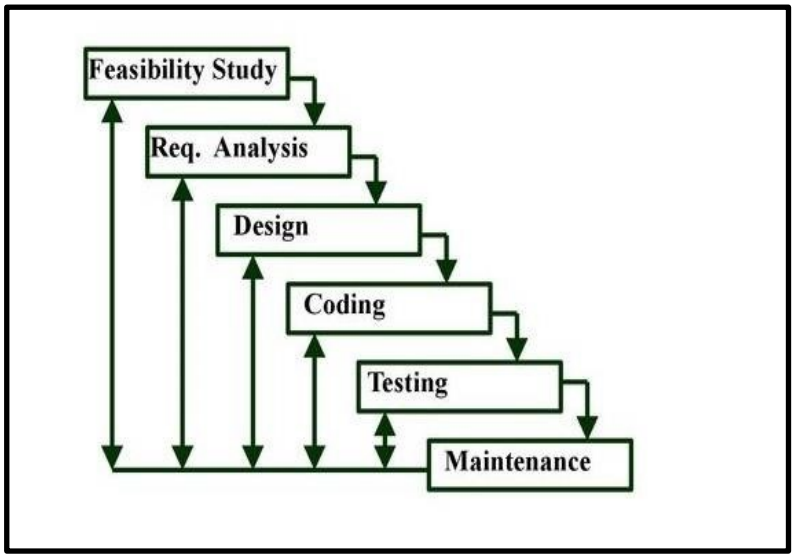


Figure 3: Iterative Waterfall Model

(Ref: Software Engineering by Prof. Rajib Mall Indian Institute of Technology, Kharagpur)

1. **Requirement Analysis**
    - We collect and record system needs from stakeholders, including managers, instructors, and administrators, throughout this phase.

      The criteria include software requirements (like Python and DeepFace), hardware specifications (like cameras and processing units), and functional expectations like real-time updates and accurate facial recognition.
    - Result: A thorough Software Requirements Specification (SRS) document outlining the capabilities of the system is produced.
2. **System Design**
    - We design the system's architecture based on the requirements, defining the hardware, software, database, and development modules.
    - Both high-level and low-level designs, such as system schematics, flowcharts, and Data Flow Diagrams (DFDs), are created.
    - Result: Explicit instructions for system builders and testers.
3. **Implementation (Coding)**
    - The application's code is written in Python by developers, who use libraries like OpenCV and DeepFace for face recognition and frameworks like Streamlit.
    - Each module is developed and tested individually (unit testing) to ensure functionality.
    - **Outcome**: Functional modules that can be integrated into the larger system
4. **Integration and Testing**
    - All individual modules are integrated, and the system is tested as a whole to identify and fix issues related to functionality, performance, and compatibility.

- Testing includes scenarios like lighting changes, partial facial occlusion, and multiple users in the camera's field of view.
- **Outcome**: A robust and error-free application ready for deployment.

5. **Deployment**
   - The system is deployed in the target environment, such as a classroom or office.
   - To make sure the system meets user demands and functions as intended, preliminary feedback is obtained.
   - Result: The application is functioning and online.
6. **Maintenance**
   - The system is checked for problems, defects, or requests for new features once it has been deployed.
   - Patches and updates are made available to enhance functionality or meet new user requirements.
   - Result: A dependable and ever-evolving application.

3.2.1. Why we choose the Iterative Waterfall Model for our project

We chose the Iterative Waterfall Model for our project, Real-Time Facial Recognition for Automated Attendance, because it blends the flexibility of iterative improvements with the structured character of the classic Waterfall Model.

Reasons for Choosing This Model:

1. **Structured Process**: The sequential completion of each step promotes attention and clarity.

2. **Early Detection of Issues**: Every stage is followed by testing, which enables issues to be found early.
3. **Iterative Improvements**: Errors may be fixed without waiting for the finished result because to the ability to go back and review earlier stages.
4. **Well-Suited for Small to Medium Projects**: The breadth of our project is manageable, and this approach gives us the right amount of structure and flexibility.

Advantages of the Iterative Waterfall Model

- **Structured Approach**: Because each phase has clear goals, the project is simpler to oversee and monitor.
- **Early Feedback and Refinement**: Early problem identification and resolution lowers the chance of more serious issues later on.
- **Improved Accuracy**: Iterative feedback guarantees that the program satisfies user requirements, such precise facial recognition.
- **Risk Mitigation**: By addressing risks in smaller cycles, the likelihood of failure is reduced.
- **Stakeholder Involvement**: Stakeholders participate in regular evaluations to make sure the system meets their needs.

Disadvantages of the Iterative Waterfall Model

- **Time-Consuming**: Reexamining phases might prolong the time needed for development.
- **Resource Intensive**: Iterations and continuous testing may call for more resources, raising expenses.
- **Complex Management**: Without enough preparation, handling several iterations and modifications might be difficult.

- **Risk of Scope Creep**: The scope of the project may increase as a result of ongoing input that generates new requirements.

3.3. DFD (DATA FLOW DIAGRAM)

A straightforward and visible method of illustrating how data moves through a system is to use a Data Flow Diagram (DFD). It offers an easy-to-understand method for comprehending a system's operations, data flow, and storage. Because DFDs simplify complicated processes into smaller, more manageable parts, they are frequently utilized in system analysis and design.

3.3.1. PURPOSE OF A DFD

- The DFD serves several key purposes:
- **Visualize the system's data flow:** It demonstrates the processing and conversion of input data into output data.
- **Understand system processes:** It helps stakeholders understand how data is handled at each step.
- **Identify inefficiencies:** By analyzing the DFD, bottlenecks and unnecessary processes can be identified and optimized.
- **Provide a foundation for system design:** It serves as a blueprint for developers to create the system.

3.3.2. Elements of a DFD

DFDs consist of four primary components, represented using standard symbols:

- **Processes:**
    - A process represents an operation or action performed on data (e.g., calculations, transformations, or decision-making).
    - It is shown as a **circle** or **rounded rectangle**.

- Example: "Process Attendance" or "Extract Features.

- **Data Flows:**
  - Data flows show how information moves between different parts of a system, such as processes, storage areas, and external entities.
  - It is shown as an **arrow** that indicates the direction of data flow.
  - Example: "Captured Image" or "Attendance Log."
- **Data Stores:**
  - A **data store** is a place where information is kept so it can be accessed or used later.
  - It is shown as an **open rectangle** or **two parallel lines**.
  - Example: "Database" or "Attendance Record Storage."
- **External Entities:**
  - External entities are the people, systems, or organizations that interact with a system from the outside. They either send data into the system or receive processed data from it.
  - It is represented using a square or a rectangle.
  - Example: "User" or "Administrator."

3.3.3. Levels of a DFD

DFDs are divided into different levels to represent increasing levels of detail.

1. **Level-0 DFD (Context Diagram)**

The Level-0 DFD provides a high-level overview of the system as a single process. It shows external entities interacting with the system and the flow of data to/from them.

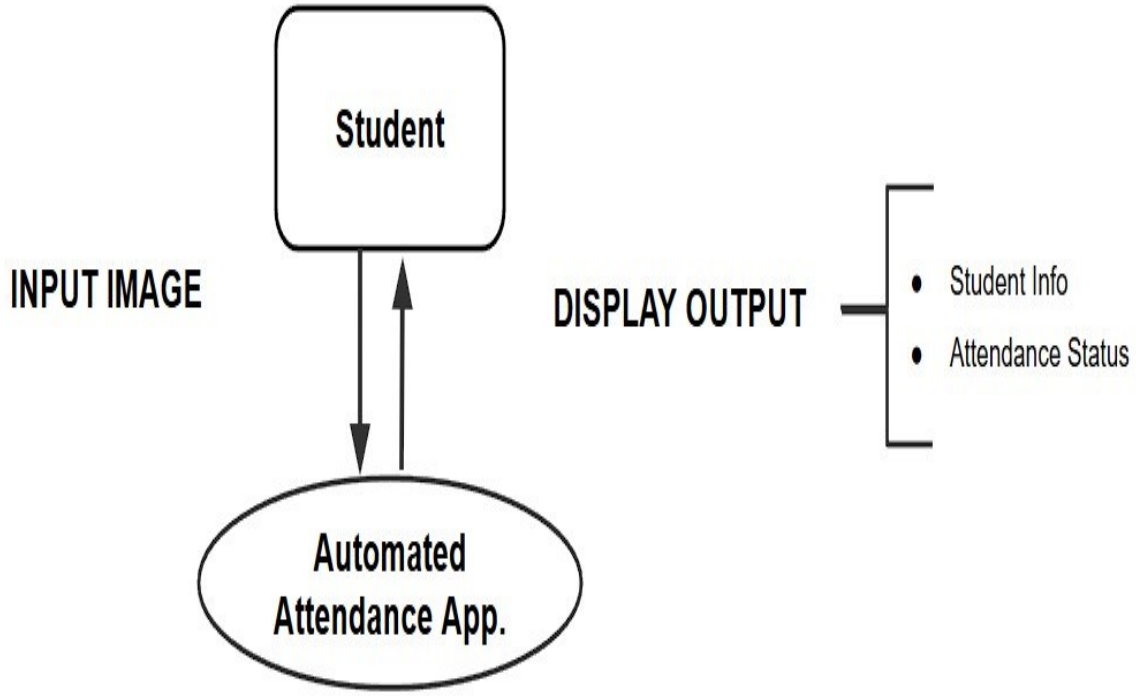


Figure 4: Level-0 DFD

2. Level-1 DFD

The Level-1 DFD breaks down the main process (from Level-0) into subprocesses. It provides more detail about what happens inside the system.

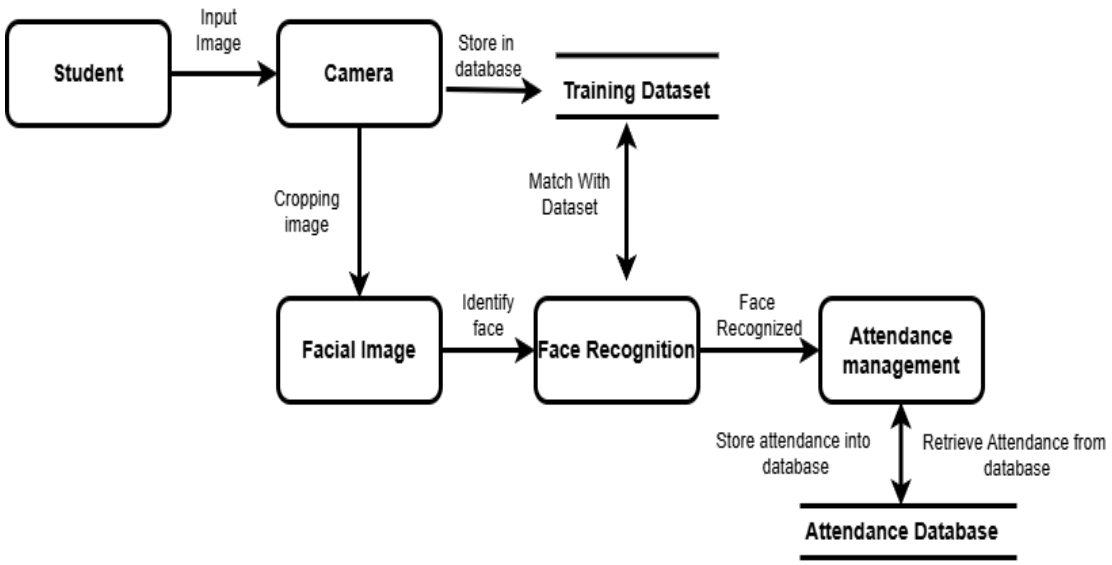


Figure 5: Level-1 DFD

3. Level-2 DFD

The Level-2 DFD further breaks down each sub-process from Level-1 into smaller, detailed processes. It provides the most granular level of detail about the system's internal workings.

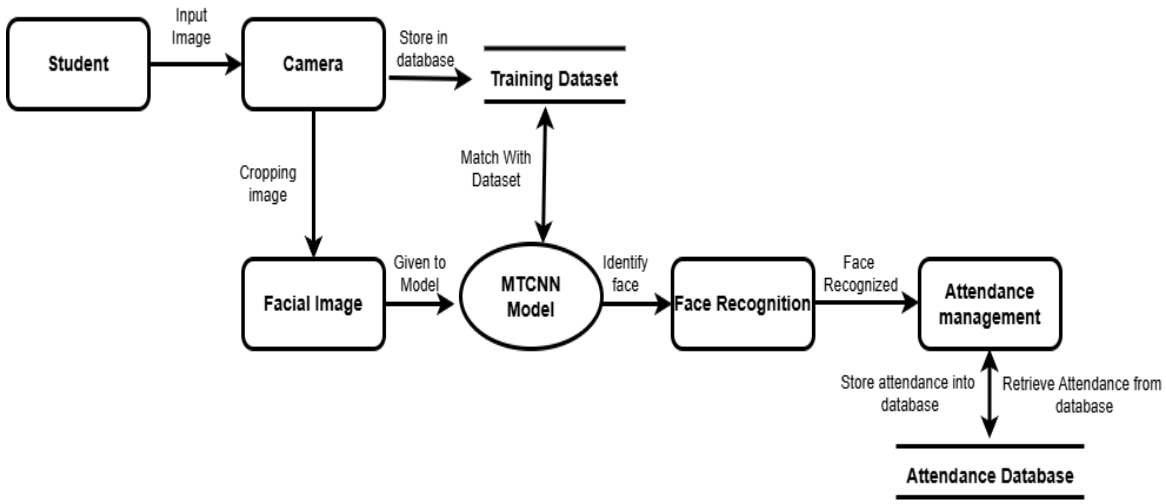


Figure 6: Level-2 DFD

3.4. SYSTEM ARCHITECTURE

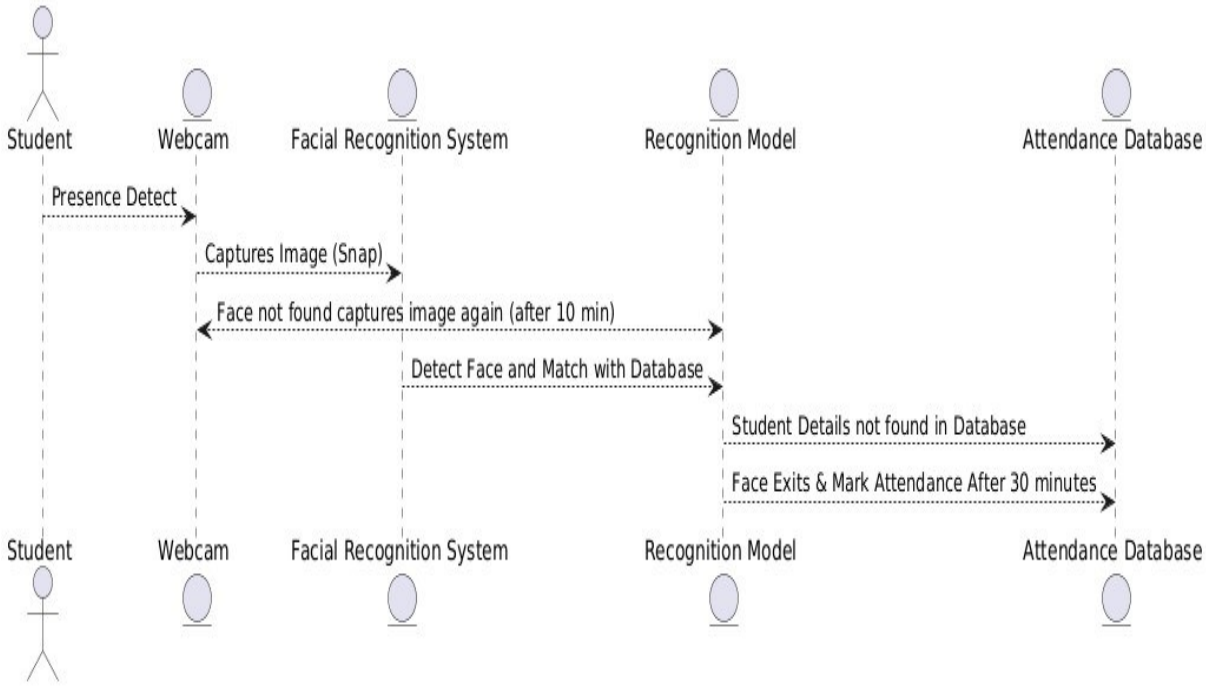


Figure 7: System architecture of real-time automated attendance application

CHAPTER – 4
ALGORITHMS USED

- Our proposed system aims to mark attendance in real-time by continuously scanning the faces (e.g. 30 minutes).
- The concept that we are using here is Neural Networks, which are majorly of two types: -

1. Artificial Neural Networks (ANN).
2. Convolution Neural Network (CNN).
3. **Multi-Task Cascaded Convolutional Neural Network (MTCNN).**

What are Neural Networks?

Neural Networks- A neural network is an artificial intelligence method that teaches computers to process data in a way that is inspired by the human brain.

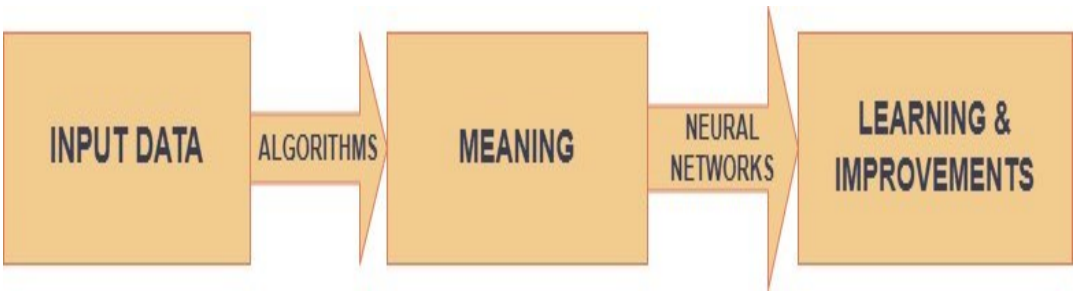


Figure 8 Neural Network Architecture

EXAMPLE- On the first day at a new office, we meet several people. as we interact, we start to recognize them based on distinct features like whether someone has a beard, appears younger or older, or has a unique hairstyle. our brain naturally picks up on these features to remember each person. when we return to the office the next day, we might notice some of these people look slightly different, maybe someone trimmed their hair or

shaved their beard. despite these changes, our brain still recognizes them by matching the core features we learned before.

A machine, however, lacks this inherent ability. It would struggle to recognize individuals based on minor adjustments unless it is trained to do so. This is where **machine learning** and **neural networks** come in. By training a machine in a way that mimics the human brain's system for recognizing faces and features, we enable it to "learn" and identify people, even if certain aspects of their appearance change.

What are Artificial Neural Networks?

1. ANNs are the simplest form of neural networks, consisting of layers of interconnected nodes or "neurons."
2. They work well for structured data (such as numeric and categorical data in spreadsheets) and on smaller datasets.

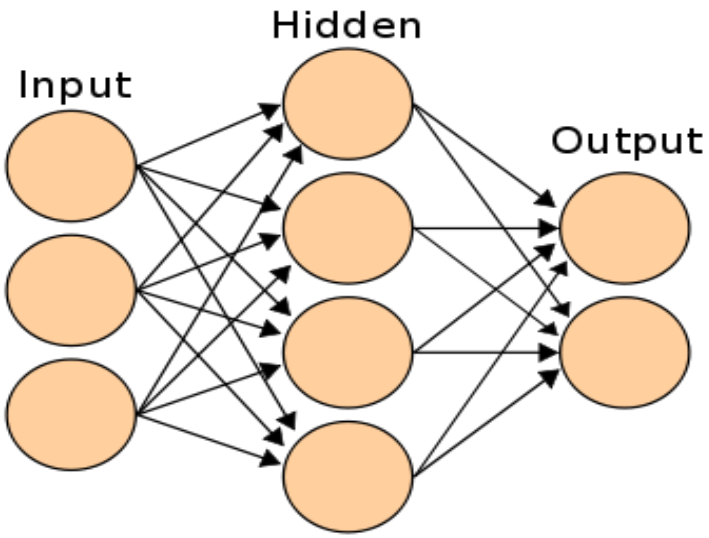


Figure 9: Artificial Neural Network Architecture

What is a Convolutional Neural Network (CNN)?

- CNNs are specifically designed for **image processing and computer vision** tasks.
- They are structured to capture spatial and hierarchical patterns in images by using **convolutional layers** that scan images in small patches.

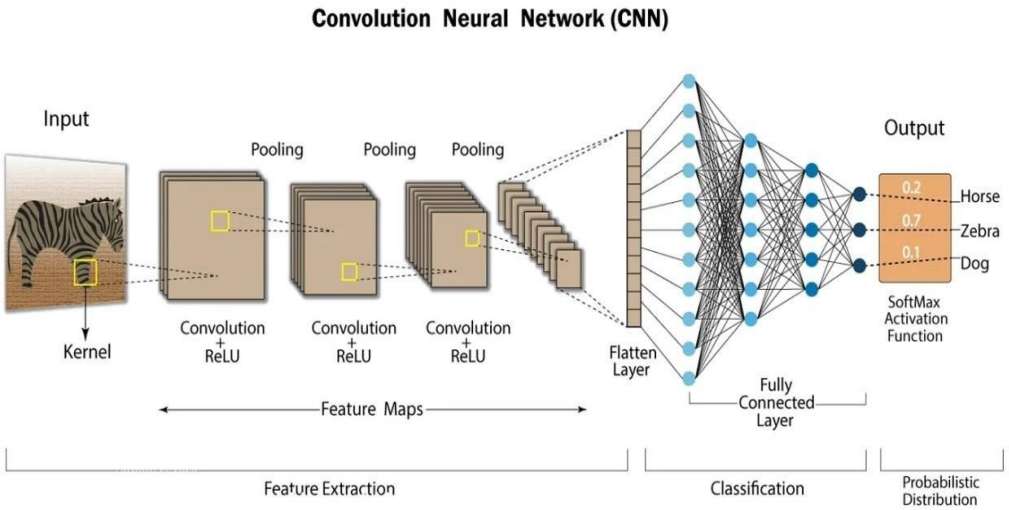


Figure 10: Convolutional Neural Network

Architecture (https://images.app.goo.gl/D1B2bti5QhUJqYRH7 **)**

➔ **The algorithm that we are using is MTCNN for face recognition because it detects better facial features as a comparison to other algorithms.**

➔ **What is MTCNN (Multi-Task Cascaded Convolutional Neural Network)?**

➔ MTCNN is a specialized CNN architecture tailored for **real-time face detection** and alignment.

➔ It consists of three stages of CNNs that work in a cascade, refining the detection results through each stage, which makes it particularly accurate.

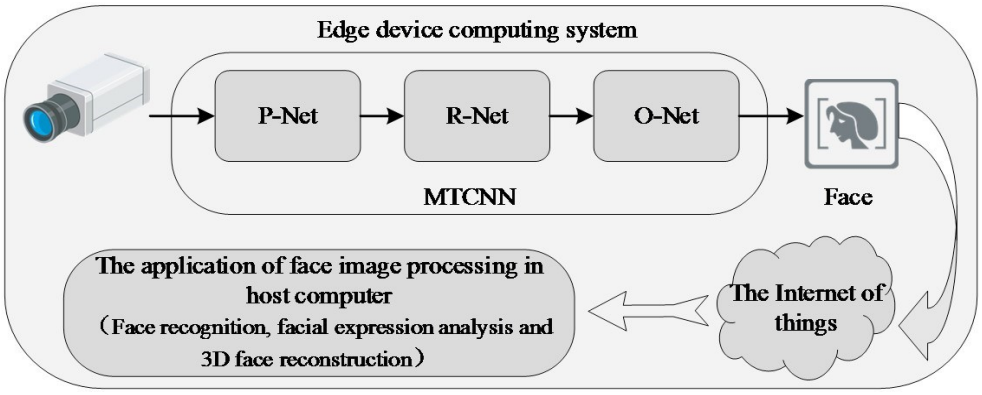


Figure 11: Multi-Task Cascaded Convolutional Neural Network

Architecture (https://images.app.goo.gl/dzeJJ6yMQzUF8hxB7)

Why MTCNN over ANN for Face Recognition

➔ ANNs are not ideal for image data because they lack a way to consider spatial information (like edges or textures) in images.

➔ Specialized design for face detection and alignment.

➔ MTCNN is an advanced CNN that provides fast, accurate face detection with landmark alignment, crucial for a real-time facial recognition system.

1) Working of an Algorithm-

Three-Stage Process:

1. **Stage 1: Proposal Network (P-Net)** - This network scans the image for candidate face regions and quickly rejects regions.

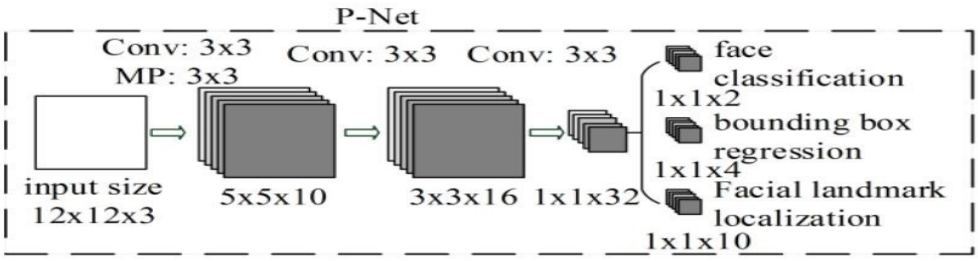


Figure 12: Architecture of P-NET

(https://towardsdatascience.com/face-detection-neural-network-structure-257b8f6f85d1)

2. **Stage 2: Refine Network (R-Net)** - The R-Net refines the candidates from the P-Net to improve the accuracy and reduce false positives.

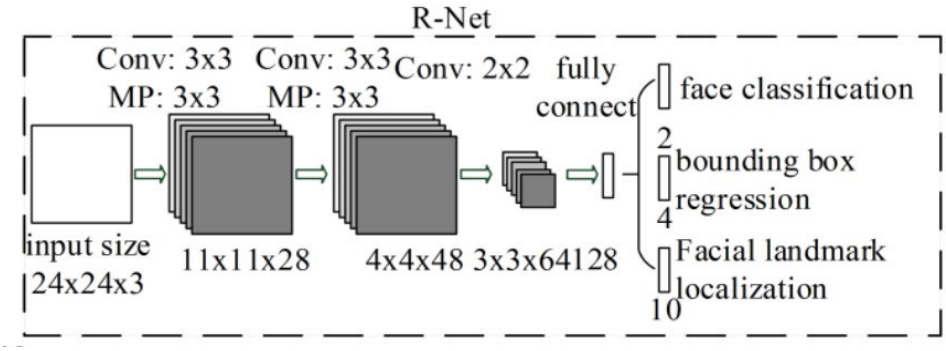


Figure 13: Architecture of R-NET

(https://towardsdatascience.com/face-detection-neural-network-structure-257b8f6f85d1)

2. **Stage 3: Output Network (O-Net)** - The O-Net further refines and identifies face landmarks (such as eyes, nose, and mouth), helping align the face correctly.

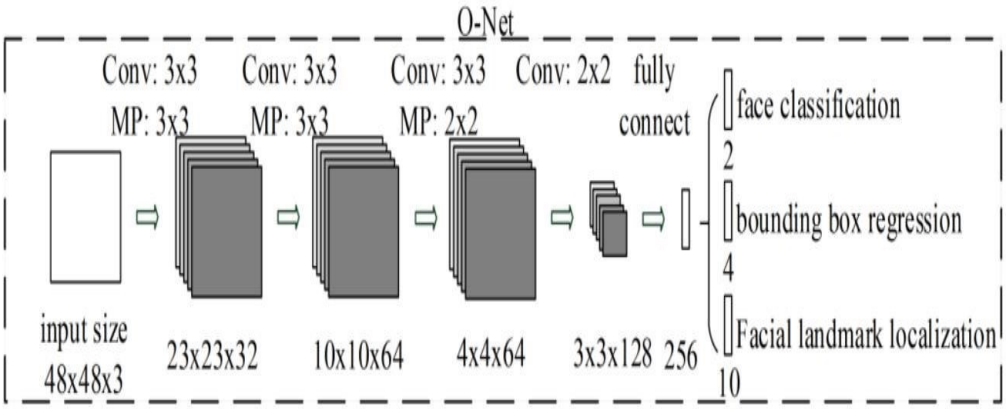


Figure 14: Architecture of O-NET

(https://towardsdatascience.com/face-detection-neural-network-structure-257b8f6f85d1)

Stage 1: Proposal Network (P-Net)- Within the, to begin with, the MTCNN makes different frames that look through the whole picture beginning from the best cleared-out corner and inevitably advancing towards the foot right corner. The data recovery preparation is called P-Net (Proposal Net), completely associated with CNN.

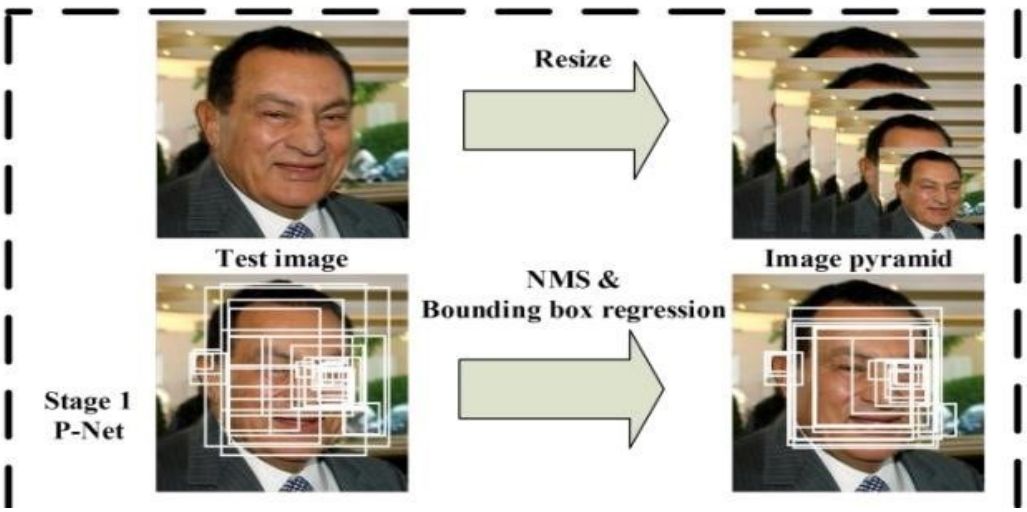


Figure 15: Pipeline for MTCNN.

- Within the P-Net, for each scaled picture, a 12x12 part runs through the picture, searching for a confrontation. Within the picture underneath, the ruddy square speaks to the part, which gradually moves over and down the picture, looking for a face.
- Within each of these 12x12 bits, 3 convolutions are run through with 3x3 parts. After every convolution layer, a prelu layer is executed (once you increase each negative pixel with a certain number alpha. Alpha is to be determined through preparing). In expansion, a max pool layer is put in after the primary prelu layer (max pool takes out each other pixel, clearing out as it were the biggest one within the vicinity).

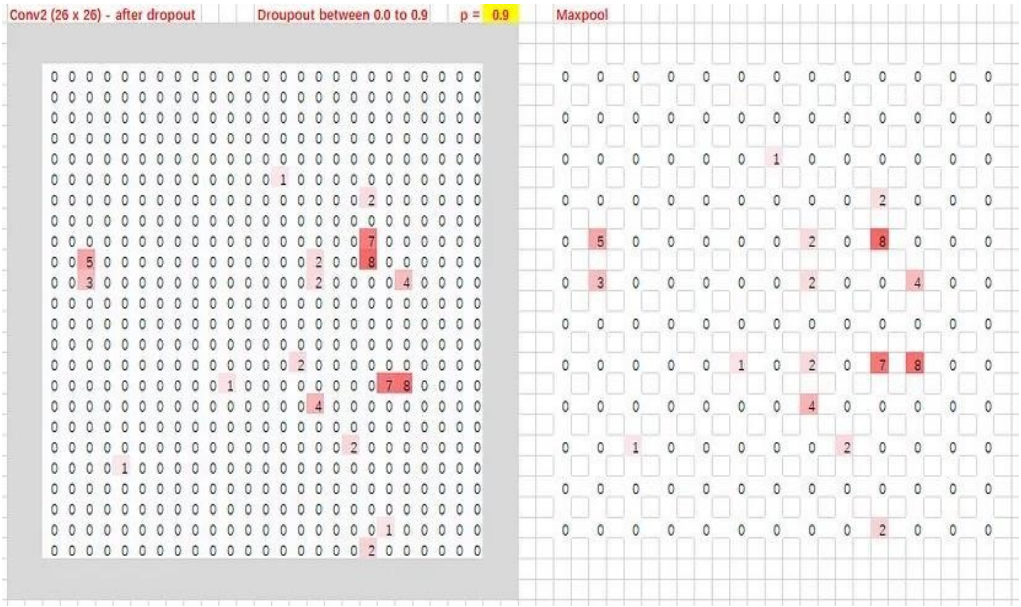


Figure 16: Max-pool // Source

- After the third convolution layer, the parts are divided into two layers. The enactments from the third layer are passed to two isolated convolution layers, and a SoftMax layer after one of those convolution layers (softmax allots decimal probabilities to each result, and the probabilities include up

to 1. In this case, it yields 2 probabilities: the likelihood that there's a confrontation within the region and the likelihood that there isn’t a face).

- Convolution 4-1 yields the likelihood of a confront being in each bounding box, and convolution 4-2 yields the arranges of the bounding boxes.

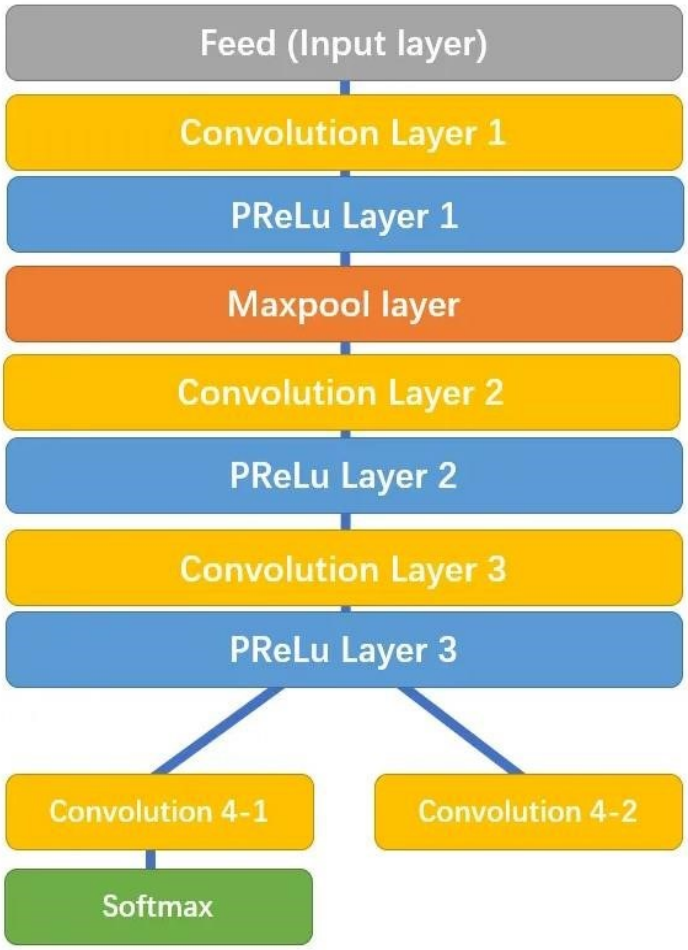


Figure 17: P-Net

- **Stage 2: Refine Network (R-Net)** - Within the moment organized all the data from P-Net is utilized as an input for the following layer of CNN called R-Net (Refinement Organize), a completely associated, complex CNN that rejects a lion's share of the outlines that don't contain faces.

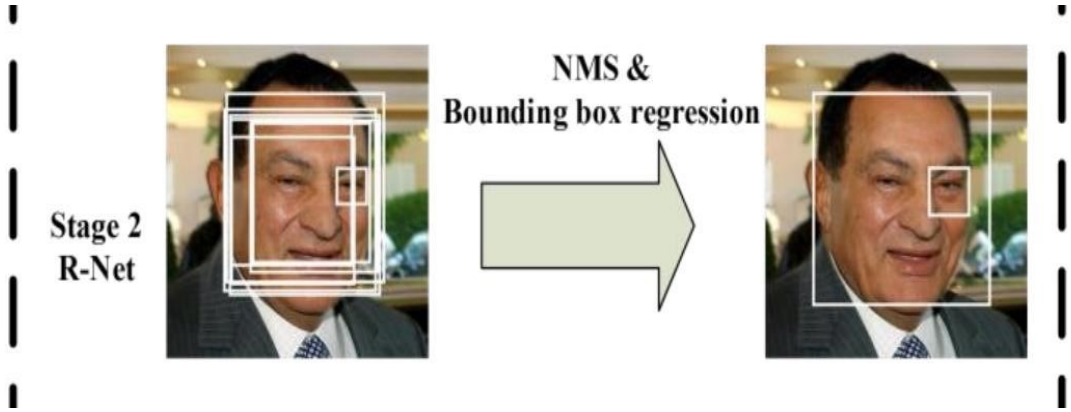


Figure 18: Pipeline for MTCNN.

o R-Net incorporates a comparable structure but with indeed more layers. It takes the P-Net bounding boxes as its inputs and refines its coordinates. o Similarly, R-Net parts into two layers within the conclusion, giving out two yields: the facilitates of the modern bounding boxes and the machine's certainty in each bounding box.

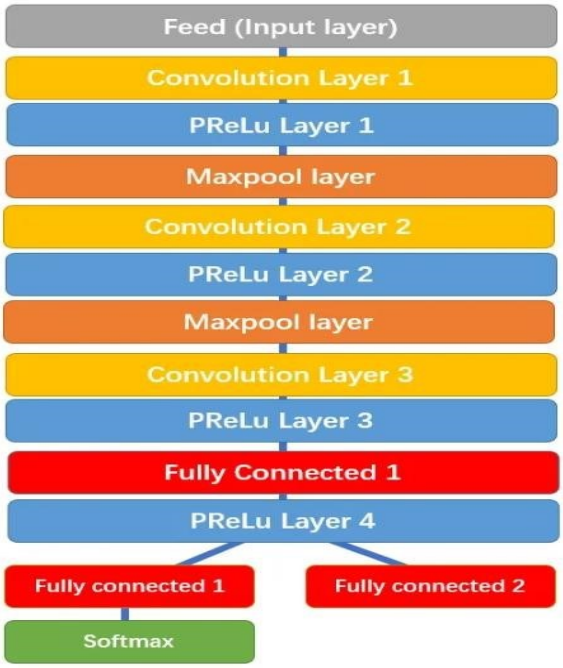


Figure 19: R-Net

- **Stage 3: Output Network (O-Net)-** Within the third and last organize, a more effective and complex CNN, known as O-Net (Output Organize), which as the title recommends, yields the facial point of interest position identifying a confront from the given image/video.

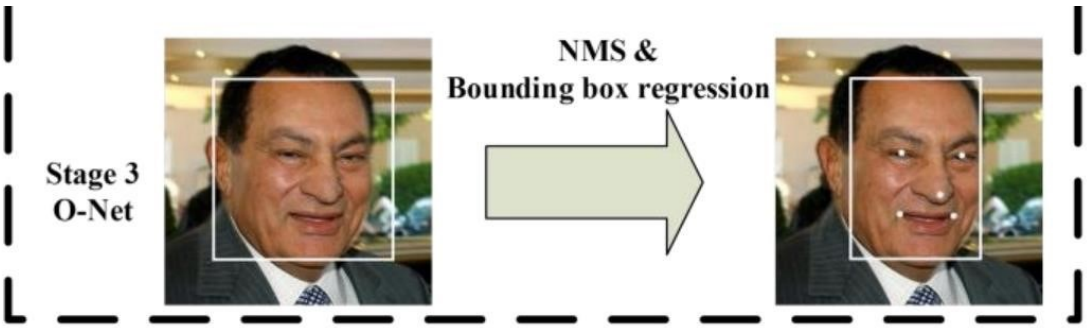


Figure 20: Pipeline for MTCNN

Figure 21: O-Net

At last, O-Net takes the R-Net bounding boxes as inputs and marks down the facilitates of facial landmarks.

- O-Net parts into 3 layers within the conclusion, giving out 3 distinctive yields: the likelihood of a confront being within the box, the arranges of the bounding box, and the arranges of the facial points of interest (areas of the eyes, nose, and mouth).

Why is MTCNN in Our System?

- **Exact Confront Detection: -** For real-time facial acknowledgment, exactness is fundamental. MTCNN's cascade engineering successfully diminishes wrong positives and recognizes faces accurately.
- **Face Alignment: -** The accurate arrangement is crucial for facial acknowledgment since minor misalignments can decrease acknowledgment exactness.

MTCNN’s point of interest location makes a difference in accurately adjusting faces, making downstream acknowledgment (e.g., utilizing DeepFace) more accurate.

- **Real-Time Processing:**
    - MTCNN’s multi-stage plan permits it to rapidly channel out insignificant locales, centering as it were on potential confront locales, which makes it proficient for real-time applications. This can be especially valuable in a computerized participation framework, where faces ought to be rapidly identified in a live environment.

- **Compatibility with Acknowledgment Models:**
    - MTCNN yields adjusted confront pictures that can effectively be bolstered into other facial acknowledgment models (like DeepFace or FaceNet) for acknowledgment. This integration progresses acknowledgment speed and precision in a real-time framework.

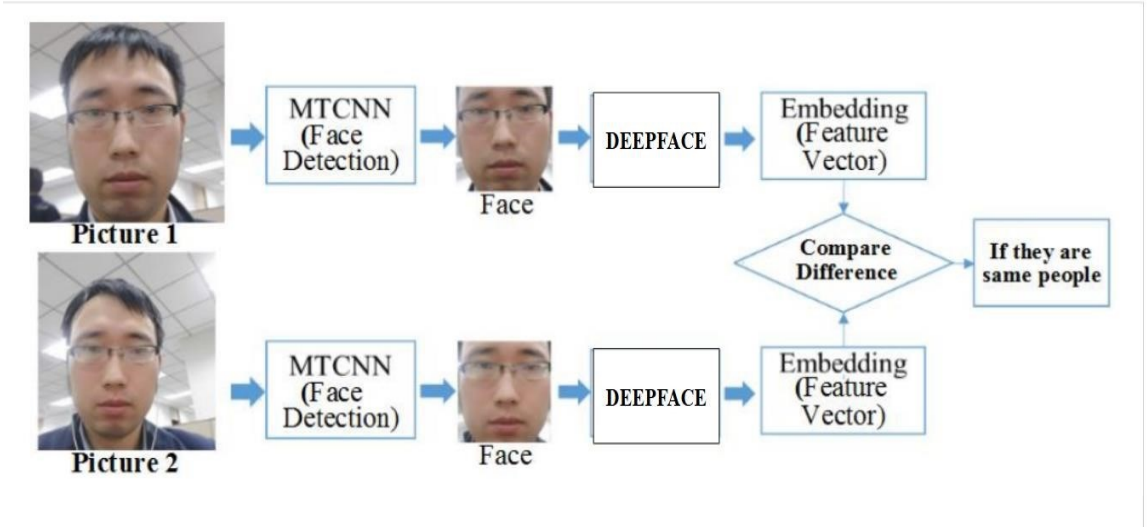


Figure 22: Why deepface?

https://iopscience.iop.org/article/10.1088/1742-6596/1518/1/012066/pdf

Why Deepface (FaceNet512) is used?

- **Disentangled Integration: -** DeepFace abstracts absent the complexities of profound learning and facial acknowledgment. It offers a high-level API, making it simple to utilize with fair a couple of lines of Python code.
- **Supports Confront Discovery, Confirmation, and Examination: -** In our application, we required not it confront acknowledgment (distinguishing an individual) but moreover confirmation (checking in case two faces have a place for the same person).
- DeepFace gives built-in capacities like confirm (), analyze (), and discover (), covering most facial acknowledgment utilizes cases.

  Cross-Platform Compatibility: - DeepFace is consistent with prevalent Python libraries like TensorFlow, Keras, and PyTorch. It can be effectively coordinated into your existing Python applications, whether we are utilizing Carafe, Django, or an information investigation device like Streamlit.

How DeepFace Differs from Other Technologies:

Feature	**DeepFace**	**Other Technologies**
Ease of Use	high-level API, quick setup	requires extensive coding/setup
Face Alignment	use MTCNN for accurate alignment	may not include built-in alignment
Open Source	open source and customize	some of them are paid

CHAPTER – 5

PROJECT MODULES

5.1. INTRODUCTION TO DATASET

- In order to preserve ethical norms and guarantee uniqueness, we used real-time data to construct a completely new dataset. This strategy puts authenticity and control over the data first, in contrast to techniques that rely on sourcing or scraping photos from other databases. Using a Python script created especially to expedite the image gathering process and satisfy the project's efficiency and scalability criteria, the procedure was automated.

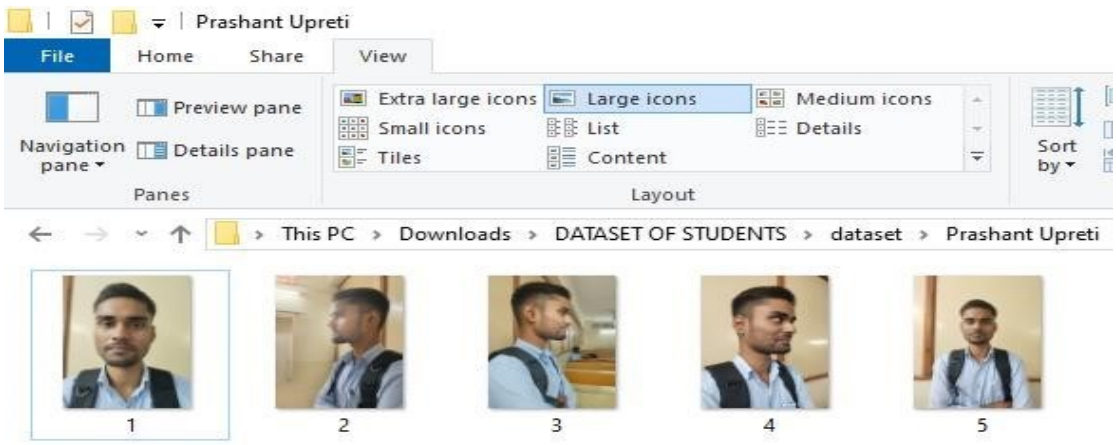


Figure 23: Initial raw data of images

The script eliminates the need for human involvement by enabling the automatic collection of photographs. The dataset is assured to be comprehensive and dependable due to its capacity to capture around 50 images per individual in a minute. To do this, the script uses real-time video stream access, where frames are extracted and stored as images. Prior to being systematically stored in a hierarchical directory structure with folder names based on unique IDs, each frame is efficiently processed.

This setup ensures organization by streamlining data management and training.

- The automatic mechanism begins capturing images as soon as the system detects a participant. This is accomplished by combining facial detection and alignment algorithms, retaining only relevant frames with unique facial traits. By concentrating on high-quality photos and removing duplicates or fuzzy frames using tools like OpenCV, the script maximizes the capture process. This ensures that every individual's dataset is varied and appropriate for model training.
- To make the process scalable, the script allows several participants to gather data simultaneously. It is suitable for situations where collecting data from large populations in a limited amount of time is required since it dynamically allocates resources to manage several video streams. Additionally, the system's architecture supports future scalability, ensuring that more users may be added to the dataset without causing any disruptions to its present structure.

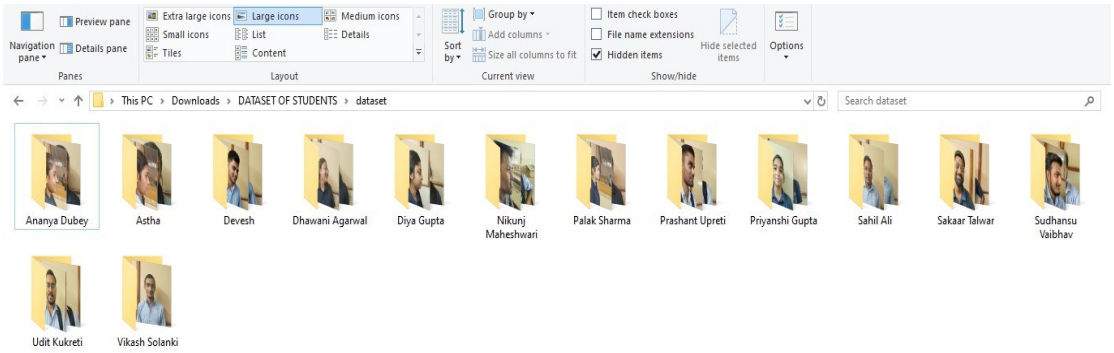


Figure 24: Prototype raw data for attendance automated via facial recognition.

- In order to guarantee uniformity in lighting and background, the first stage of the collection involves taking unprocessed pictures of pupils in

controlled settings. The foundation for additional preprocessing and augmentation operations is this raw dataset, which will increase the dataset's adaptability for machine learning model training. Combining automation, scalability, and efficiency, this method builds a solid basis for producing a high-quality dataset that is suited to the requirements of the project.

5.2. LIBRARIES USED

- The Python script utilizes several key libraries to perform real-time image collection, facial detection, and dataset creation. Below is a brief overview of each library and its role in the script:

DeepFace

- The deep face library provides powerful facial recognition and analysis capabilities, such as emotion detection, age, gender prediction, and more. Although not explicitly invoked in this script, it could be integrated later for advanced analysis or verification of collected images.

MTCNN (Multi-Task Cascaded Convolutional Networks)

The MTCNN module is utilized for facial discovery. It distinguishes bounding boxes and facial points of interest inside a picture, permitting exact discovery of faces. This script leverages MTCNN to find faces in each outline captured from the webcam. The recognized locales are at that point edited and spared to make a structured dataset.

OpenCV (cv2)

- OpenCV[7] could be a broadly utilized library for computer vision assignments. In this script, it serves numerous purposes:
- Accessing and capturing real-time video nourishes from the webcam.

- Drawing bounding boxes around identified faces for visual input amid picture capture. Converting color spaces (e.g., BGR to RGB) to preserve consistency with other libraries.
- Resizing confronts pictures to a standard determination for consistency within the dataset. OpenCV too encourages real-time outline show and client interaction during the capture process

OS

- The os module gives capacities for association with the operating system. In this script, it is utilized to form registries powerfully for putting away datasets. This guarantees that each member features an isolated envelope for their pictures, improving organization and scalability.

NumPy

- NumPy, a library for numerical computing, is utilized verifiably in conjunction with OpenCV and MTCNN. It forms picture information as clusters, empowering proficient operations like trimming, resizing, and color change.
- NumPy, a library for numerical computing, is utilized verifiably in conjunction with OpenCV and MTCNN. It forms picture information as clusters, empowering proficient operations like trimming, resizing, and color change.

5.3. SNAPSHOTS

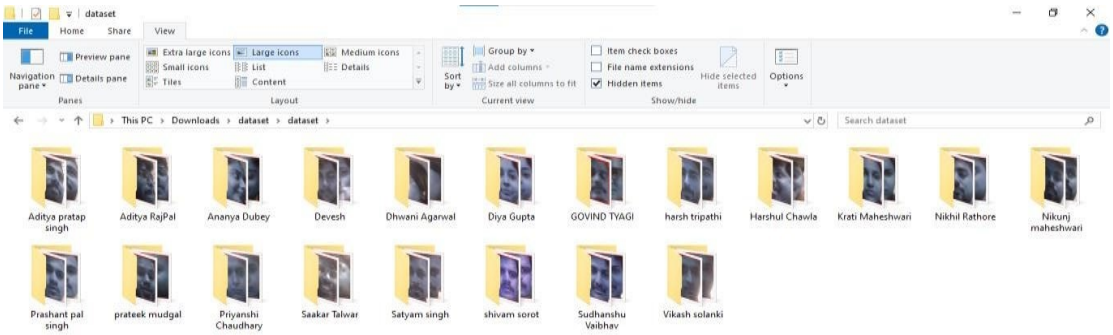


Figure 25: 50 images of students using script

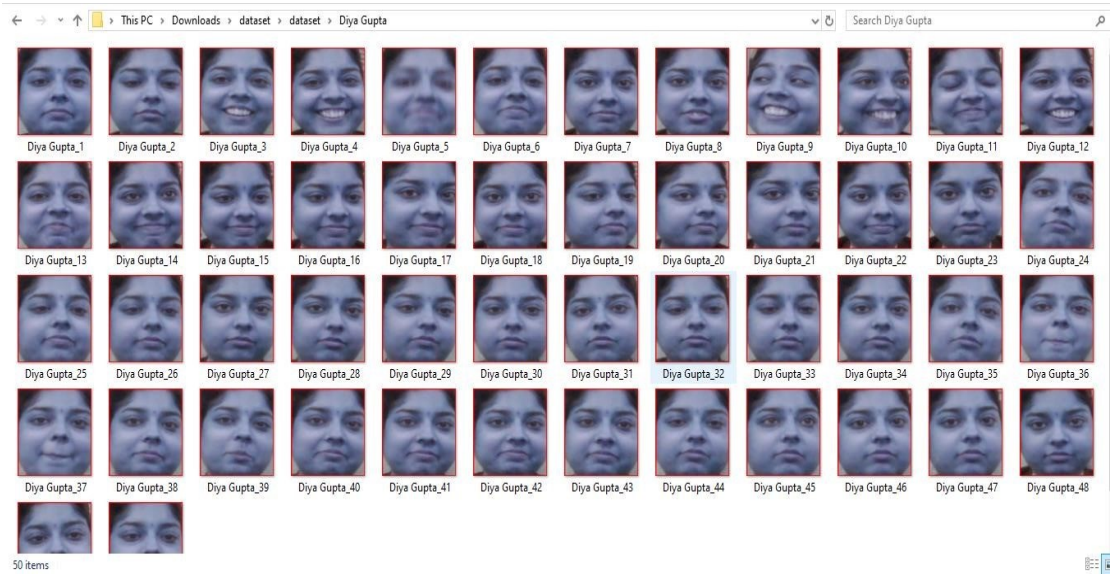


Figure 26: Folder of 50 images of students using script

5.4. MODEL TRAINING

1. Dataset Description

The description of the dataset twenty distinct students facial photos make up the dataset for this research. The full dataset comprises 1,000 photos, with 50 facial photographs in each student's allocated folder. The photos were taken under various circumstances:

- Variations in face expressions
- Diverse lighting conditions
- Modest adjustments to angles

The goal of this variety was to strengthen the model and enable it to handle actual differences in student appearances during live scanning.

2. Data Preprocessing

To get the dataset ready for use in training:

- Image Resizing: To guarantee consistent input dimensions, all images were scaled to 160×160 pixels, a standard input size for face recognition tasks.

- Normalization: To speed up training and enhance convergence, each image's pixel values were scaled between 0 and 1 by dividing them by 255.0.
- Labeling: Class labels were derived from folder names. These labels were transformed into integer indices and then one-hot encoded using Keras utilities' to_categorical() function.
- Dataset Splitting: To guarantee that the model was evaluated on unseen images during training, the dataset was divided into two parts: 80% for training and 20% for validation.

3. Model Architecture

Using the TensorFlow/Keras Sequential API, a Convolutional Neural Network (CNN) was created especially for the multi-class categorization of facial photos.

The architecture consists of:

Layer Type	Parameters & Details
Input Layer	Shape = (160, 160, 3)
Conv2D	32 filters, (3×3), ReLU activation
MaxPooling2D	Pool size = (2×2)
Conv2D	64 filters, (3×3), ReLU activation
MaxPooling2D	Pool size = (2×2)
Flatten	Converts 2D feature maps to 1D vector
Dense	128 neurons, ReLU activation
Output Dense	20 neurons (for 20 students), Softmax

4. Training Configuration

The model was compiled and trained using:

Adam, the adaptive learning rate optimization optimizer

- Loss Function: categorical_crossentropy (fit for multi-class classification with one-hot encoding)
- Accuracy is the evaluation metric.

32 is the batch size.

- Time periods: 30

Both were used to train the model:

- Locally in Visual Studio Code (with Python and TensorFlow)
- On Google Colab, accessing the dataset and saving the model via Google Drive integration

5. Training Environment

Environment	Details
Local	VS Code, Python 3.9, TensorFlow, OpenCV
Cloud	Google Colab with GPU acceleration
Dataset Location	/content/drive/MyDrive/MODEL TRAINING/dataset
Model Output	Saved as person_model.keras in Drive

6. Training Results During training:

Adam (adaptive learning rate optimization) is the optimizer.

- Loss Function: categorical_crossentropy (fit for multi-class classification with one-hot encoding).
- After around 20 epochs, accuracy stabilized after increasing rapidly.
- There was little overfitting, since validation accuracy regularly matched training accuracy or was marginally lower.
- Accuracy of final training: ~ 94%
- Accuracy of final validation: ~ 84%

7. **Model Evaluation**

After training, the model was tested on:

- Student pictures that aren't included in the training collection.
- Webcam inputs in real time to test recognition capabilities in real time.
- The model could recognize student faces with accuracy.
- Using softmax likelihood to match their identification with high confidence
- Managing little cosmetic adjustments, such as hairstyles or accessories

8. **Real-Time Face Recognition Testing**

During live testing, a webcam module was utilized to continually record frames.

- OpenCV was used to process each frame:
- Face detection (MTCNN)
- Normalization and resizing to the model that has been trained
- If the confidence level above a predetermined threshold, the anticipated student name (class label) was shown.

• Attendance was automatically recorded in a CSV or Excel sheet.

Workflow Example:

1. Let the webcam run for half an hour.
2. Take a picture and guess the identity every ten seconds.
3. Enter the timestamp and the name of the identified student in the attendance file.
4. Save the file to Google Drive or your local drive.

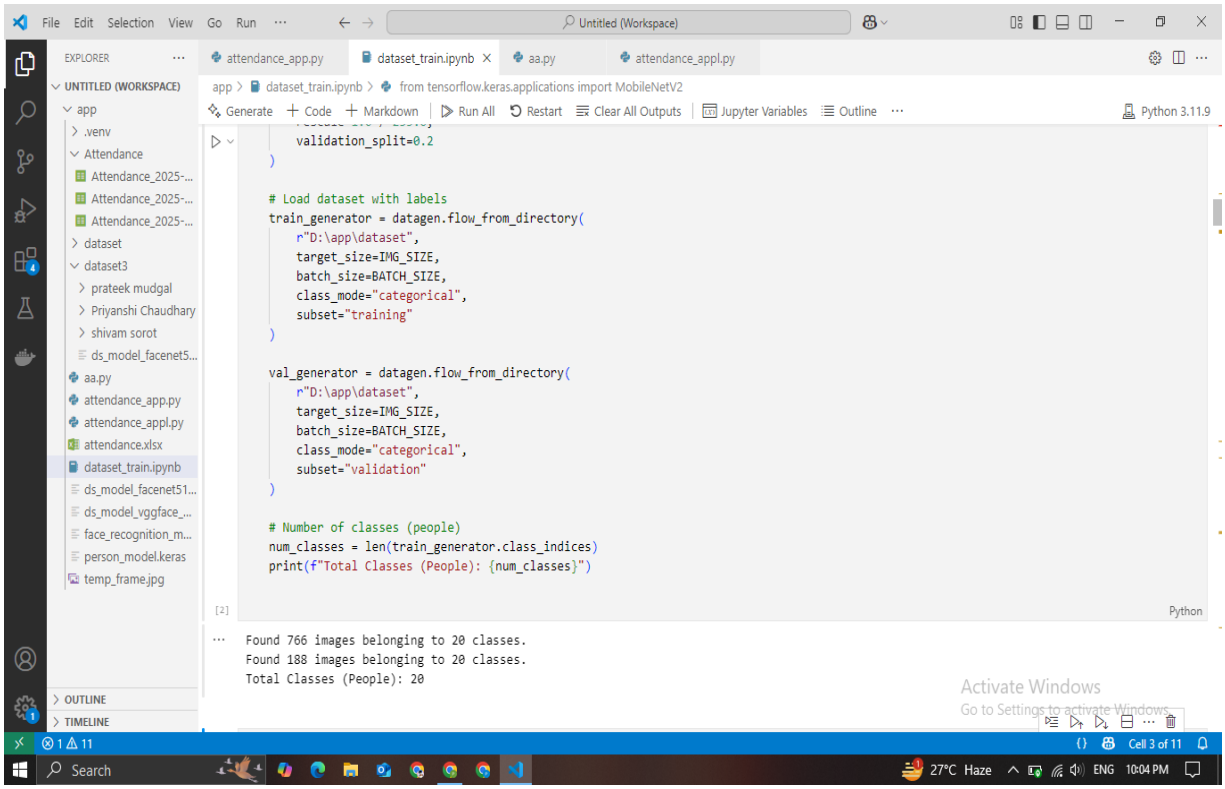


Figure 27: Train dataset on locally

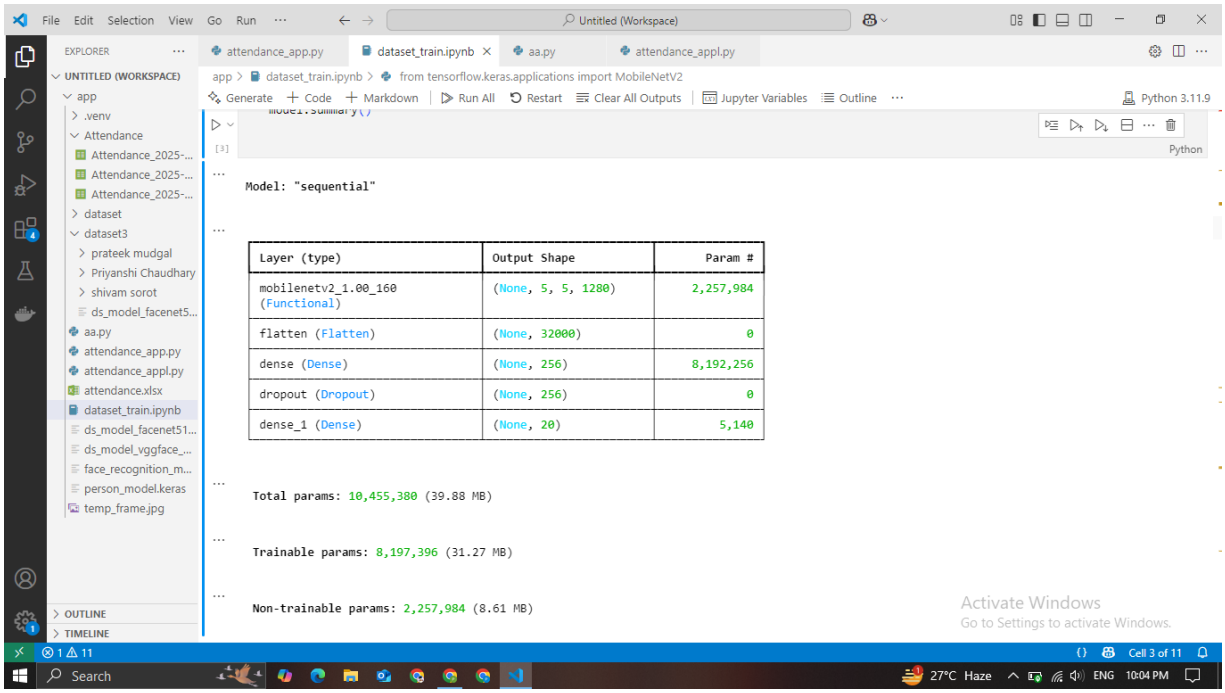


Figure 28: Train dataset locally

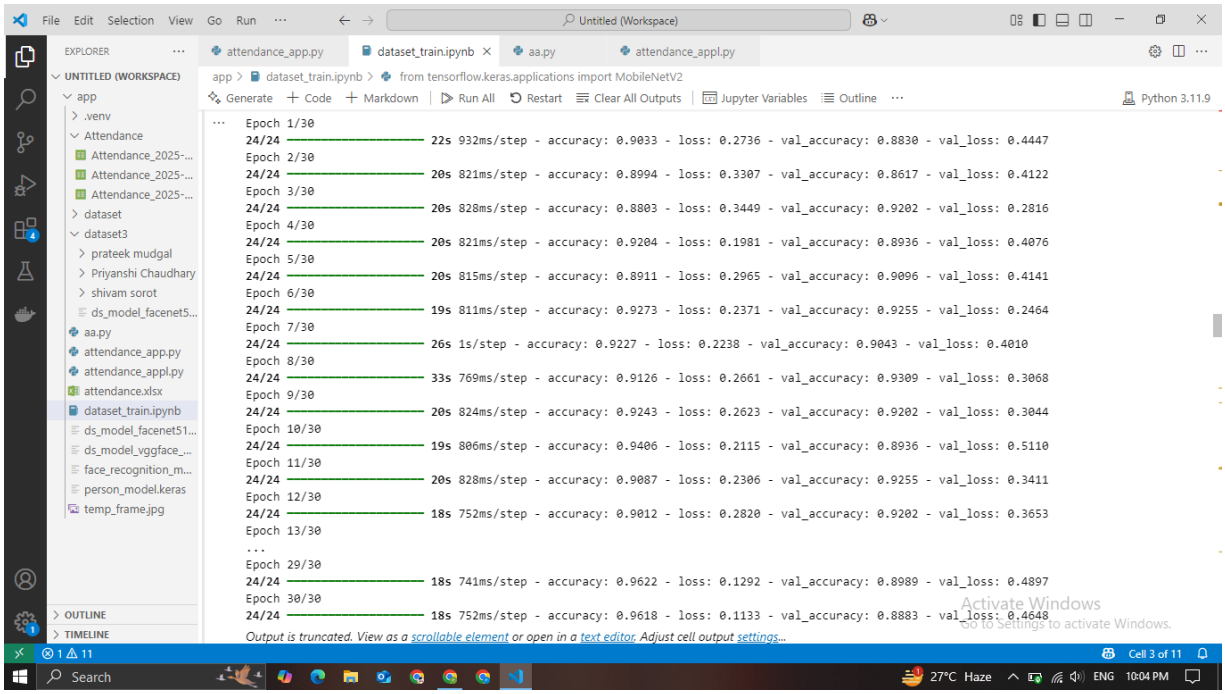


Figure 29: Train dataset locally

Model Trained Google Colab

1. Dataset Overview

A bespoke dataset of 20 distinct students' faces was produced in order to develop a strong facial recognition system. The total dataset size was 1,000 photos, with 50 photographs in each student's allocated folder. In order to replicate authentic classroom situations, these images were purposefully taken in a variety of real-world settings, including varying lighting conditions, slight perspective adjustments, and face expressions. In order for the model to learn to detect faces accurately even when there are slight differences during real-time recognition, this diversity was essential.

2. Data Preprocessing

- A number of preprocessing procedures were done before the dataset could be utilized to train the model:

All of the images were scaled to 160 by 160 pixels. Facial recognition models frequently employ this scale to strike a compromise between computational efficiency and detail.

- Normalization: By dividing each value by 255, pixel values that were initially in the 0–255 range were reduced to the 0–1 range. This ensured constant input data, which sped up and smoothed out the training process.
- Label Encoding: Class labels were automatically generated from the folder names, which reflect the identities of the students. In order to create one-hot encoded vectors that may be used to train classification models, these labels were first translated into numerical representation.
- Split Training-Validation: To assess model performance, 20% of the data was set aside for validation and 80% was utilized for training. This made sure the model was tested on data that it had never seen before.

3. Model Architecture

The first model was developed using a bespoke Convolutional Neural Network (CNN) built with TensorFlow's Keras API. This model's objective was multiclass categorization, with each class standing in for a single student. The architecture was composed of the following layers:

Layer Type	Description
Input Layer	Accepts images of shape (160, 160, 3)
Conv2D Layer	32 filters, 3×3 kernel, ReLU activation
MaxPooling2D	Pool size of 2×2
Conv2D Layer	64 filters, 3×3 kernel, ReLU activation
MaxPooling2D	Pool size of 2×2
Flatten Layer	Converts 2D feature maps into 1D vector
Dense Layer	128 neurons, ReLU activation
Output Layer	20 neurons (for 20 students), SoftMax

Because of this structure, the model was able to identify the right student identities by mapping the spatial elements it learned from the facial photos.

4. Model Training on Google Colab

All model training was done with Google Colab, which provides GPU acceleration for faster performance. The setup was customized for cloud-based operations, utilizing Google Drive for dataset access and model storage. The following was the configuration for the model training

• Adam, the optimizer, was selected due to its capacity for adjustable learning rate.

- Loss Function: Categorical Cross entropy, which works well for classification applications involving several classes.
- Evaluation Metric: Accuracy, which is monitored throughout validation and training.
- 32 is the ideal batch size for efficient GPU use.
- 30 epochs are used to provide the model adequate time to learn without being overfit.

The trained model was stored as person_model.keras in Google Drive for convenient access and deployment after the training was carried out directly in Colab notebooks.

5. Training Environment

The training environment used was entirely cloud-based. Below are the key specifications:

Component	Details
Platform	Google Colab with GPU runtime
Programming Language	Python 3.x
Framework	TensorFlow with Keras API
Dataset Path	/content/drive/MyDrive/MODEL TRAINING/dataset
Model Storage	Google Drive (person_model. keras)

This cloud architecture provided simple interaction with Google Drive for smooth dataset administration and allowed effective training without local hardware constraints.

6. Training Results

Throughout 30 training epochs, the model demonstrated consistent progress. The accuracy started to level off by the 20th epoch, suggesting that the model had successfully picked up important characteristics from the training set. With few indications of overfitting, validation accuracy exhibited a similar trend and stayed around training accuracy.

• Accuracy of Final Training: about 94%

• Accuracy of Final Validation: about 84%

According to these findings, the CNN model performed well on the validation dataset.

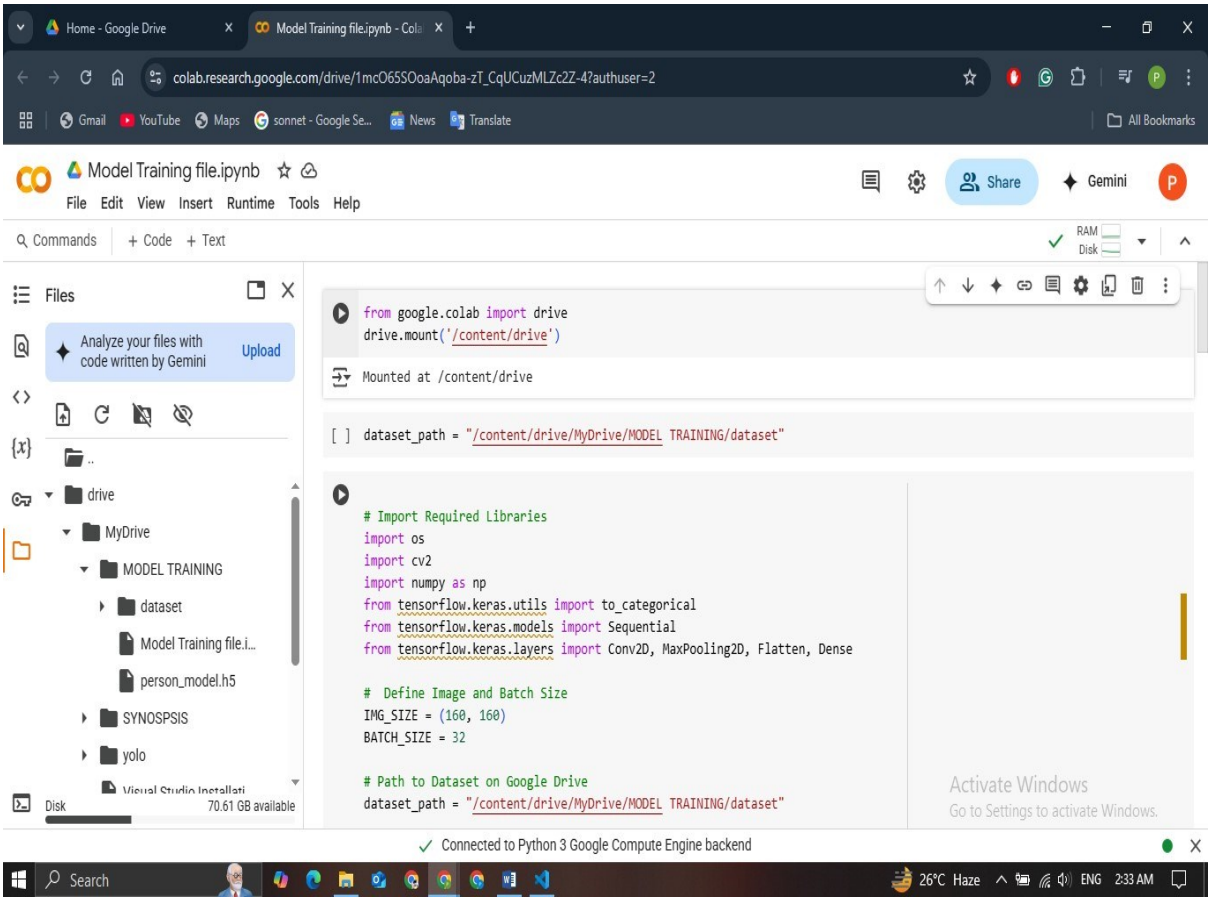


Figure 30: Train dataset on Google Colab

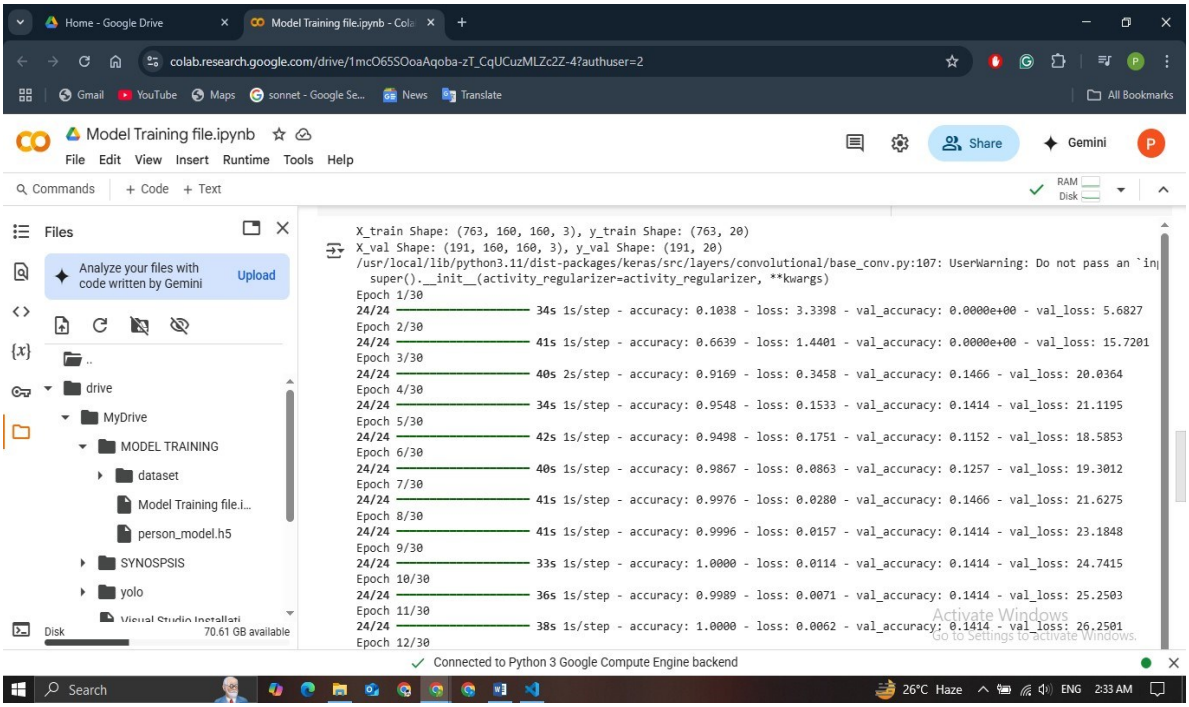


Figure 31: Train dataset on Google Colab

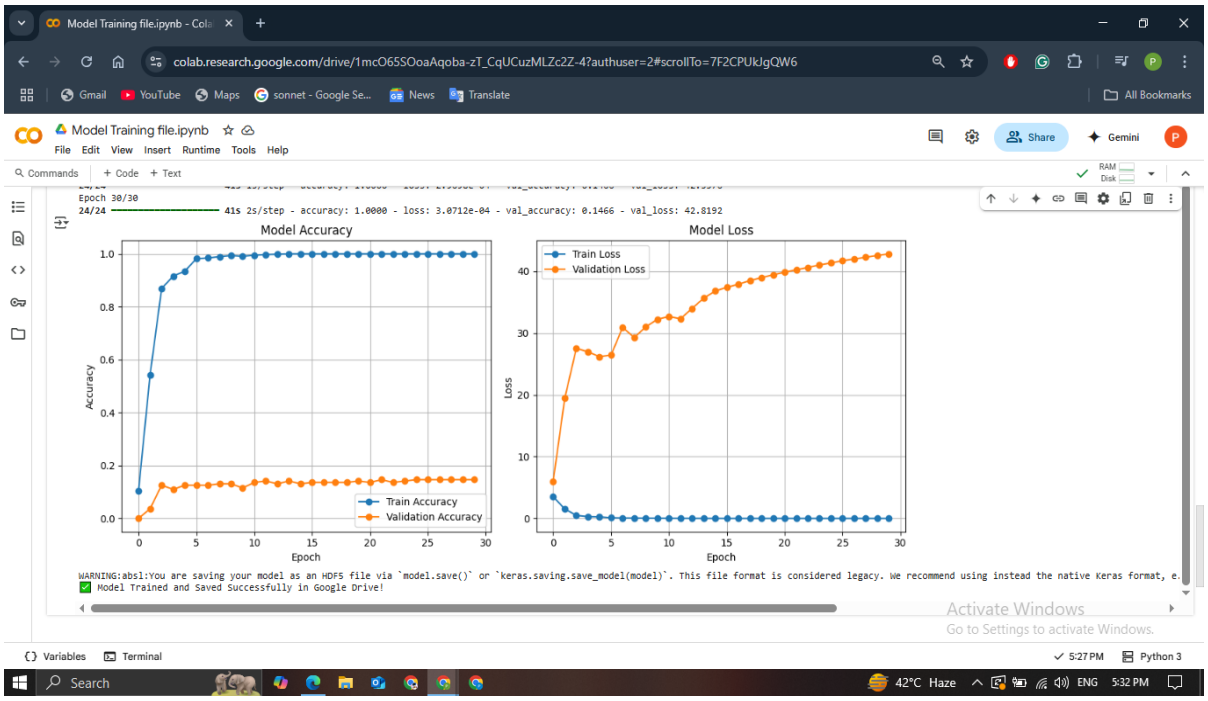


Figure 32: Graph plotting

7. Model Evaluation

The model was evaluated in two ways after training:

1. Fresh, untested photos that weren't part of the validation or training sets.
2. Real-time face capturing via live webcam input.

The model did a respectable job of confidently recognizing student faces. It could adapt to little variations in illumination, haircuts, and face accessories like spectacles. SoftMax output was used to make predictions, and only outcomes with a high degree of certainty were deemed legitimate by using a confidence criterion.

8. Real-Time Face Recognition

It was combined with a real-time recognition system that received live input from a camera. This is how it operated:

- Frame take: OpenCV was used to continually take frames from the camera.
- Face Detection: Each frame's faces were found using MTCNN.
- Preprocessing: Faces that were detected were normalized and shrunk to 160 x 160.
- Prediction: The trained model was given the processed picture to classify identities.
- Attendance Marking: The student's name was shown and entered into an Excel or CSV sheet with a date if the anticipated confidence level surpassed a certain threshold.

Example Workflow:

- For a predetermined amount of time, say thirty minutes, the camera operated.
- Every 10 seconds, a fresh frame was taken.
- For every student who was recognized, predictions were recorded.

- The attendance file was automatically stored locally or on Google Drive.

9. Initial Approach: Custom CNN Model

We started by utilizing the Keras Sequential API to create a customized CNN model. Convolutional layers with ReLU activations, MaxPooling layers for downsampling, and dense layers for classification (20 output classes) were all part of the design.

Using the labeled dataset, we trained the model for 30 epochs.

Restrictions noted:

- The validation accuracy was within 70–75%.

The model was sensitive to partial occlusions, face angles, and lighting . .conditions, but it was unable to generalize to real-time camera data.

- Misclassification occurs often, particularly when pupils have comparable characteristics.

We abandoned the CNN-based classifier because of its subpar performance in the actual world.

10. Second Attempt: MTCNN for Detection and Recognition

- After that, we employed the Multi-task Cascaded Convolutional Neural Network (MTCNN) for both:
- Face detection and identification based on embedding
- Even though MTCNN improved face detection, overall identification accuracy remained uneven.

The Reason It Didn't Work:

- Each frame takes a long time to process (around 1 second each picture).
- Needed extensive fine-tuning and lacked resilience in real-time situations; low accuracy for faces with varying emotions or partial visibility.

11. Final Solution: FaceNet 512

- The most efficient way to overcome these restrictions was to use FaceNet with 512-dimensional embedding.
- FaceNet does not categorize faces directly. Rather, it produces a distinct embedding vector for every face, which may subsequently be compared using cosine or Euclidean distance.
- Why FaceNet 512 Was Effective:
- Presented extremely precise embeddings
- Attained a % recognition accuracy
- Real-time webcam input functioned well, and it was resilient to variations in illumination, posture, and emotion.
- Scalable: just store new embeddings when a new student is added; retraining is not necessary.

FaceNet-Based Workflow

1. Face Detection: A lightweight detector, such as OpenCV, is used to detect faces.
2. Preprocessing: Faces that have been identified are scaled to 160 by 160 pixels and aligned.
3. Embedding Generation: A fixed-size embedding vector is produced using FaceNet 512.
4. Matching: A distance threshold is used to compare the embedding with previously saved embeddings of known pupils.
5. Attendance Marking: The student is marked as present if they match.

12. Outcome

Model	Accuracy	Real-Time Speed	Robustness	Scalability
Custom CNN	70%	Moderate	Low	Poor
MTCNN	80%	Slow	Moderate	Limited
FaceNet 512	**95%**	**Fast**	**High**	**Excellent**

5.5. USER INTERFACE (UI) DEVELOPMENT

1. Screen 1: Login to the system this is the system's authentication gateway.

• Fields for input:

• Teacher ID

• Password

• Practicality:

Only authorized workers are granted access, and the system is protected against unauthorized use.

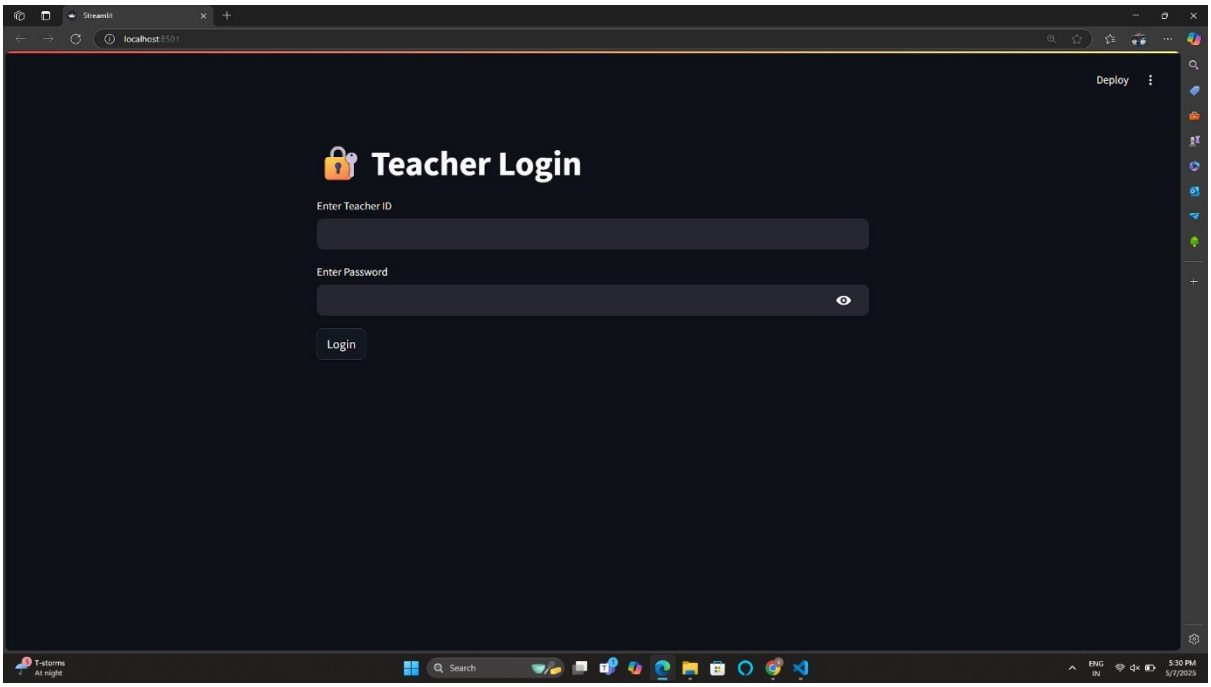


Figure 33: Teacher Login Page

2. Screen 2: Entry Screen for Class Session

The teacher is sent to a screen to enter session-related information after successfully logging in.

1. Input fields: Teacher Name, Subject Name, Subject Code, Branch, Semester,

   Date, and Time (automatically updated with the current timestamp)

2. Goal:

   • To record session-related metadata for documentation purposes

   • This information is kept for traceability along with attendance.

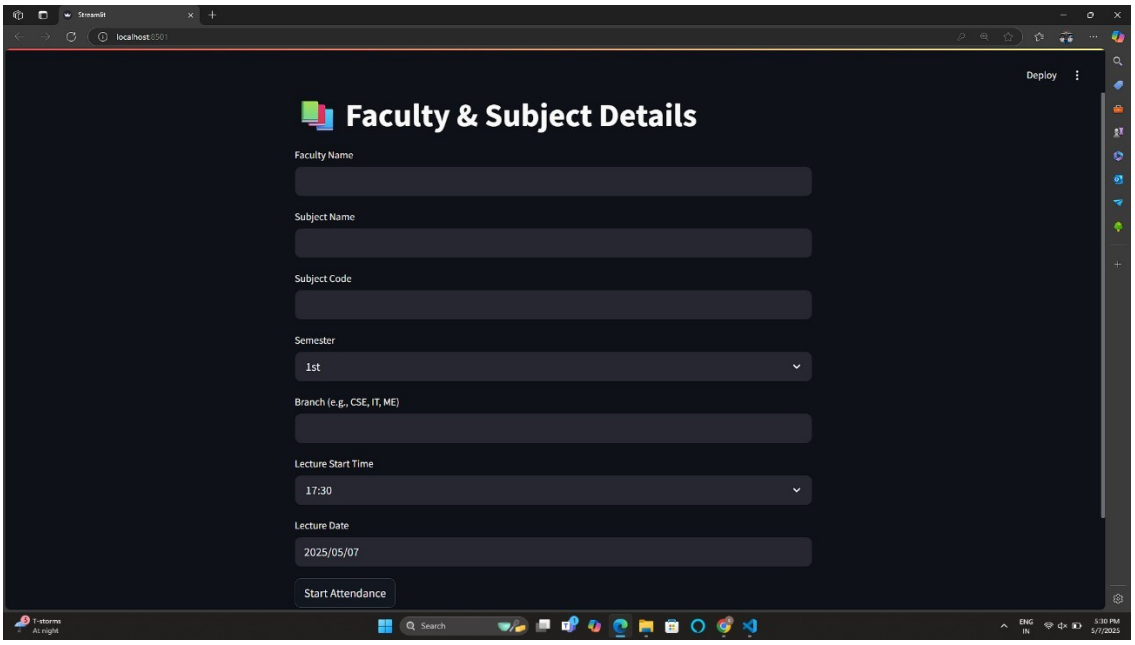


Figure 34: Faculty & Subject details.

3. Screen 3: Face Scanning & Attendance

FaceNet 512 is used on this screen for real-time facial recognition.

- Qualities:

  OpenCV is used to access the webcam; FaceNet is used for face detection and embedding extraction.

  To guarantee accurate attendance recording, a student must be matched four times in a row. Once matched, the student's name is displayed as "Present" in an Excel file. Students who are not detected are left as "Absent" by default.

  For thirty minutes, attendance is monitored, with rescans occurring every ten minutes.

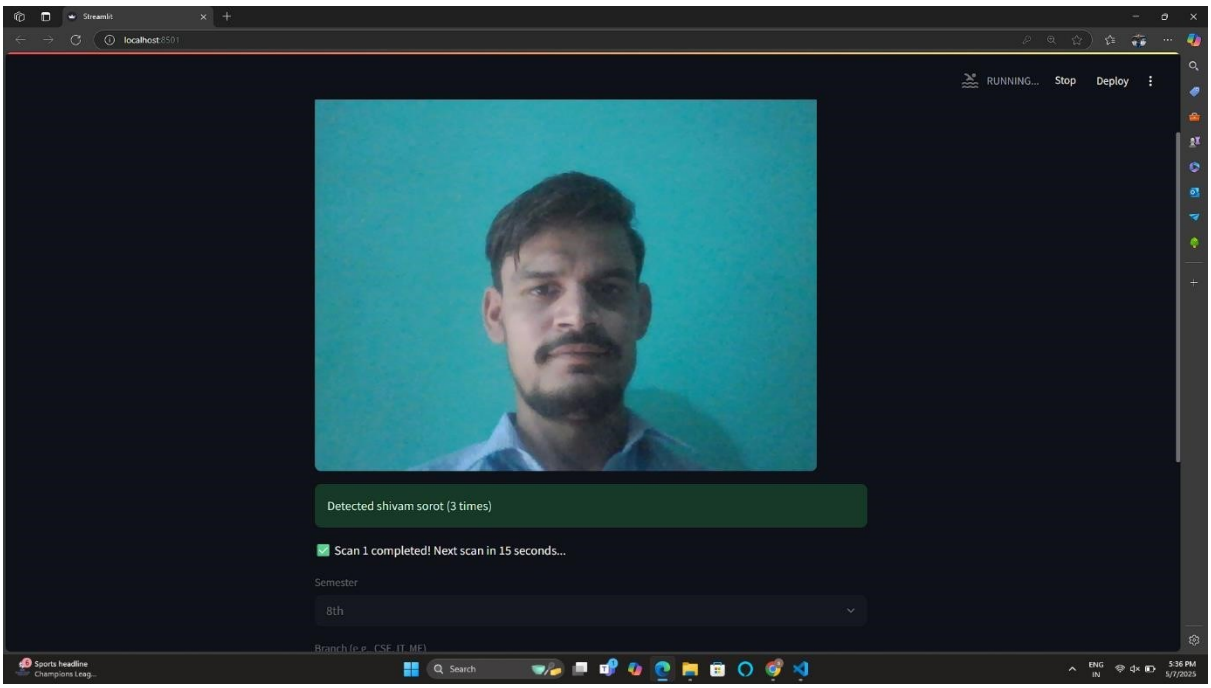


Figure 35: Student Image Capture in Frame

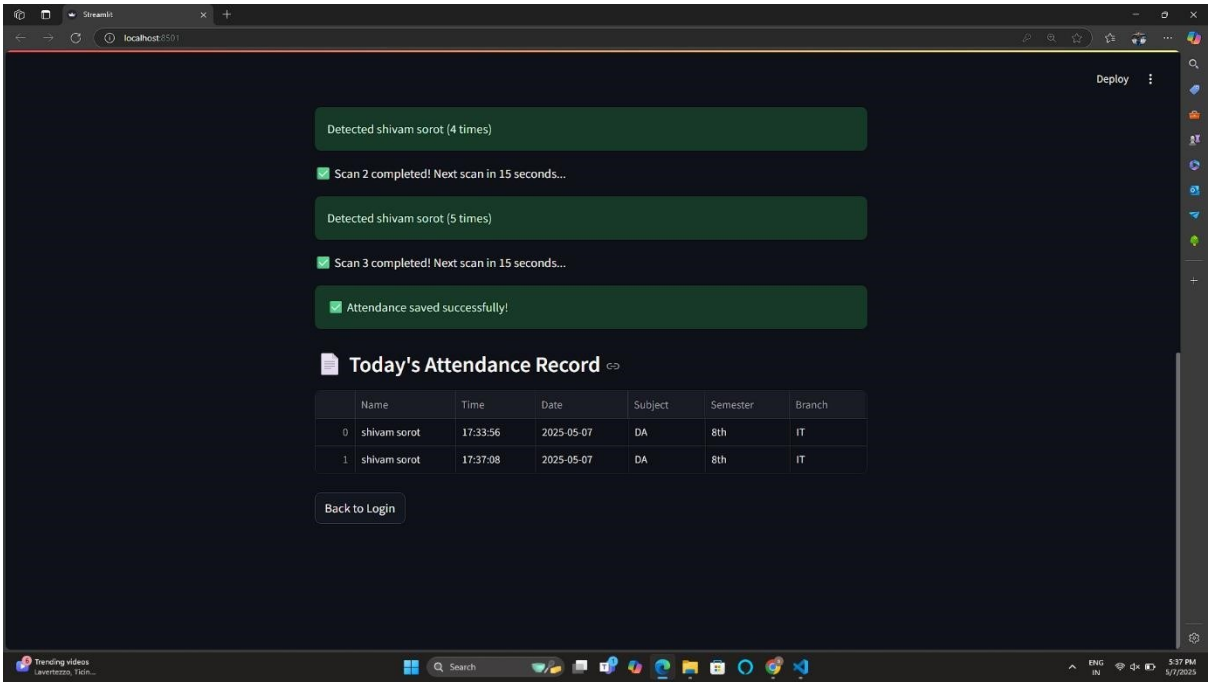


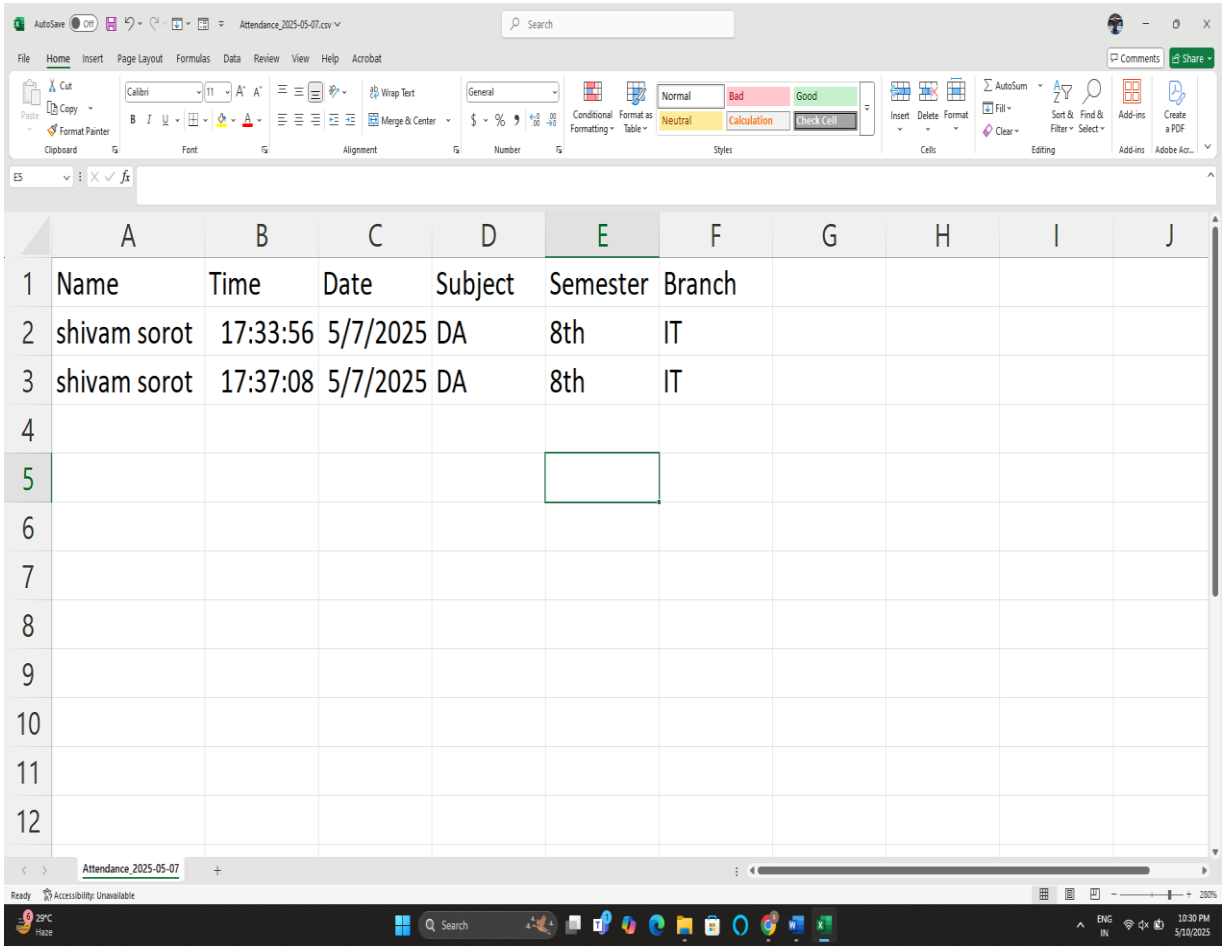
Figure 36: Scanned details of Student Attendance Record

4. Screen 4 (Attendance Summary Screen)

Following the conclusion of the scanning session:

- An Excel sheet including the names, roll numbers, present/absent status, date, time, and topic of the students is shown. Option to export for admin review
- Optional Features:

  ○ Download as.xlsx or.csv



	A	B	C	D	E	F
1	Name	Time	Date	Subject	Semester	Branch
2	shivam sorot	17:33:56	5/7/2025	DA	8th	IT
3	shivam sorot	17:37:08	5/7/2025	DA	8th	IT
4						
5						
6						
7						
8						
9						
10						
11						
12						

Figure 37: Student record in Excel sheet

Technologies Used

Component	Technology
Frontend	Streamlit
Camera Access	OpenCV
Model Inference	FaceNet 512
Data Handling	Pandas + Excel
File Storage	Local / Google Drive
Platform	Google Colab + Streamlit

Benefits of UI Design

- A smooth and methodical flow.
- Accurate attendance with confirmation for many matches.
- A simple and intuitive interface for teachers.
- Tracking and exporting records is simple.

CHAPTER – 6
TESTING

- Introduction to Software Testing- A crucial phase in the software development lifecycle, software testing entails assessing a program to find errors, confirm functionality, guarantee performance, and confirm that the system satisfies user requirements. Delivering a reliable, error-free, and user-friendly application is the ultimate objective.
- Testing was essential to confirming the quality of the machine learning model and the performance of the Streamlit-based user interface in our automatic attendance system that uses real-time facial recognition. Exact and comprehensive testing was necessary due to the sensitivity of activities including face identification, matching, and attendance marking.

Testing Methods Employed in Our Project

In our project, we conducted three main kinds of testing:

1) Manual Testing

Definition: To make sure the application functions as intended, manual testing involves carrying out test cases by hand without the assistance of automated tools.

Where It Was Employed:

We actively employed manual testing throughout the development process to verify the Streamlit interface's screen transitions, data flow, and user interaction.

Examples from Our Project:

- Tested both valid and incorrect credentials to confirm the operation of the login page.

- Teacher/session information, including topic name, code, date, and time, were manually filled out and verified.
- For student scanning, Ob functioned as the camera module's behavior.
- Verified that the student presence is updated and the Excel attendance sheet is created accurately.

2) Functional Testing

Definition- Functional testing is concerned with confirming that the system's features function in accordance with the requirements or specifications.

Where It Was Employed:

Every feature of our system, including facial recognition, scanning logic, session logging, and data export, was tested to make sure it operated as intended.

Examples from Our Project:

Accurate face recognition and matching is achieved by the trained model. The scanning method only marks a student as "present" after four successful matches.

After scanning, attendance is automatically entered into the Excel sheet.

The user is successfully sent to the session input page and subsequently the scanning screen by the system after logging in.

The Reason It Was Significant:

Functional testing made verified that no UI transitions or flawed recognition logic would result in an inaccurate attendance mark. It aided in confirming that our technology works as a trustworthy substitute for manual attendance.

3) Non-Functional Testing (Performance & Usability)

Definition- Non-functional testing is the process of assessing a system's usability, performance, scalability, and dependability in many scenarios.

Where It Was Employed:

Due to time and tool restrictions, we were unable to do sophisticated performance testing; however, we did perform basic non-functional assessments to make sure the system functions properly in real-world scenarios.

Examples from Our Project:

Confirmed that in both local and Google Colab settings, the camera opens and reacts instantly.

Made sure that up to 20 pupils could scan in real time using the technology.

During prolonged scanning sessions, observed system resource utilization (CPU, RAM).

Teachers' ability to use the user interface (UI) without requiring technological assistance was tested to ensure usability.

Why It Was Important:

A slow response time or a challenging user interface can make a system less usable even if it functions well. We were able to verify that the program is usable in real-world classroom environments thanks to nonfunctional testing.

4) Problems & Solutions in Testing

Challenge: The accuracy of our first bespoke CNN model and subsequent MTCNN model was insufficient.

Solution: We switched to FaceNet-512, which greatly increased the accuracy and performance of real-time face recognition, particularly when evaluated on a dataset of 20 students with 50 photos each.

- **Test Cases of the Project**
- **Functional**

S. No	Test Case	Description	Expected Result	Test Result
1	Enter your login credentials.	Enter the relevant instructor ID and password to log in.	Successful login; you are sent to the next screen.	Pass
2	Enter incorrect login information.	Enter incorrect login information.	Access prohibited, error message shown	Pass
3	Blank login fields	Try to log in without providing your ID or password.	Not permitted to log in, warning shown	Pass
4	Entry of session information	Input the name of the instructor, the subject, the time, date, code, semester, and branch.	After accepting the details, the user went to the scanning screen.	Pass
5	The session field is missing.	Send without completing any of the session fields.	An error notice asking the user to complete every field	Pass
6	The camera opens successfully.	The camera should initialize correctly on the scan screen.	The camera opens and displays a live video stream.	Pass

7	No face was found.	The lighting is bad or the face is out of frame.	System notifications to change the illumination or position	Fail
8	A face that is unknown was found.	An unknown individual (not in the dataset) shows up in the frame.	Attendance is not recorded by the system, and faces are ignored.	Pass
9	After four scans, the matching face was noted.	At least four times during the session, a known student's face was spotted.	Attendance is noted and recorded on the sheet.	Pass
10	Prior to four detections, attendance was not recorded.	A known student's face is only shown one or three times.	Attendance has not yet been recorded.	Pass
11	Creation of Excel reports	Create an Excel attendance report after the scan is complete.	A sheet is made with the current students' marks on it.	Pass

12	Excel opens properly.	Open the Excel file that was created.	Attendance data are accurately shown in Excel.	Pass
13	The model loads properly.	When the scan begins, load the trained FaceNet model.	The model loads and is prepared for face matching without any error.	Pass

- **Non- Functional**

S.No.	Test Case	Domain
1	Verify that the program launches and begins scanning in less than five seconds.	Performance Testing
2	Make that the user interface is accessible and responsive across a range of screen sizes.	Compatibility Testing

CHAPTER – 7
CONCLUSION

- The real-time facial acknowledgment framework utilizing OpenCV and MTCNN gives a proficient, computerized arrangement for participation administration. By leveraging progressed confront discovery and acknowledgment calculations, the framework looks at people in real-time over a 30-minute window and precisely marks participation in a consistent and error-free manner.
- This approach kills the challenges of conventional participation frameworks, such as manual blunders, buddy punching, and time wasteful aspects. The utilization of MTCNN guarantees solid confront discovery, indeed in complex scenarios like shifting lighting conditions or diverse points, whereas OpenCV encourages precise acknowledgment and information processing.
- The framework demonstrates the significant potential for use in corporate settings, educational institutions, and other settings where accurate and automated involvement is essential. Expanding the system's adaptability, coordinating cloud-based information capacity, and adding features like analytics and real-time notifications appear to be future enhancements. Generally speaking, this goes beyond grandstands to the revolutionary possibilities of computer vision and artificial intelligence in optimizing scheduling forms and advancing organizational effectiveness.

CHAPTER – 8
LIMITATION & FUTURE SCOPE

7.1. LIMITATION

- **Lighting and Natural Conditions:**

  The system's performance may suffer under poor or inconsistent illumination. For example, conditions that are too bright or too dark may make it difficult to recognize faces correctly.

- **Point and Posture Variations:**

  The system could have trouble identifying individuals if their faces are not facing the camera or if they exhibit noticeable postural changes.

- **Hardware Dependency:**

  Computationally efficient equipment is needed for real-time handling. Execution may slow down on devices with limited handling control, which might result in recognition delays.

7.2. FUTURE SCOPE

Integration with Cloud Technology:

The framework can be improved by coordinating cloud capacity for putting away and overseeing participation records. This would empower consistent getting to information from different areas and devices.

Adaptability for Bigger Populations:

Future cycles of the framework may well be optimized for adaptability, empowering it to handle huge datasets of enlisted people without compromising performance.

Portable and IoT Integration:

The framework can be conveyed on portable stages or coordinates with IoT gadgets like keen cameras, permitting adaptable and versatile participation following solutions.

Multi-Factor Authentication:

Adding multi-factor confirmation, such as ID card checking or QR code approval near facial acknowledgment, would increment security and reliability.

Real-Time Alarms and Analytics:

Features like real-time notices for inconsistencies (e.g., unregistered people recognized) and participation analytics (e.g., participation patterns) might improve the system's functionality.

Flexibility to Diverse Use Cases:

Beyond participation following, the framework may well be adjusted for utilization in get-to-control, guest administration, and other scenarios requiring secure and robotized character verification.

Security and Compliance: -

Future overhauls seem to consolidate progressed encryption and compliance with information security controls to address security concerns.

Offline Functionality:

Enhancing the framework to work offline and adjust information once an online association is accessible would make it more dependable in ranges with restricted connectivity.

Feeling and Behavior Detection:

Extending the framework to recognize feelings or screen behavioral designs may give extra experiences, particularly for applications in instruction or corporate situations.

CHAPTER- 9
REFERENCES

[1] "An End-to-End Real-Time Face Identification and Attendance System using Convolutional Neural Networks | IEEE Conference Publication | IEEE Xplore.". Available: *https://ieeexplore.ieee.org/document/9029001*

[2] C.-F. Wang, "What Does a Face Detection Neural Network Look Like?" Available:*https://towardsdatascience.com/face-detection-neural-network-structure257b8f6f85d1*

[3] "OpenCV Image Processing | Image Processing Using OpenCV." Available: *https://www.analyticsvidhya.com/blog/2021/05/image-processing-usingopencv-with-practical-examples/*

[4] J. Du, "High-Precision Portrait Classification Based on MTCNN and Its Application on Similarity Judgement," *J. Phys. Conf. Ser.*, vol. 1518, no. 1, p. 012066, Apr. 2020, doi: 10.1088/1742-6596/1518/1/012066.

[5] A. Jawabreh, "Explore the most advanced deep learning algorithm for face detection," Available: *https://medium.com/the-modern-scientist/multi-task-cascaded-convolutionalneural-network-mtcnn-a31d88f501c8*

[6] "Fundamentals of Software Engineering, Fourth Edition, Rajib Mall".

[7] "Image Processing using OpenCV — Python | by Dr. Nimrita Koul | Medium.". Available: *https://medium.com/@nimritakoul01/image-processing-using-opencv-python9c9b83f4b1ca*

Code For Dataset Collection on Real Time

```
from mtcnn import MTCNN
import cv2 import
os import numpy as
np

# Initialize the MTCNN model
detector = MTCNN()

# Create the main dataset directory dataset_dir
= "dataset"
os.makedirs(dataset_dir, exist_ok=True)

def detect_face_mtcnn(image):
    """
    Detect face using MTCNN.
    Returns the bounding box of the detected face.
    """                          results =
detector.detect_faces(image)          if
len(results) > 0:          x, y, w, h =
results[0]['box']          return x, y, w, h
return None

def capture_images():          #
Initialize webcam          cam =
cv2.VideoCapture(0)
```

```
cam.set(3,  640)    #  Set  width
cam.set(4, 480)  # Set height

    # Ask for the user's name
    name = input("Enter the name of the person: ").strip()

    # Create a folder for the user inside the main dataset directory    person_dir
= os.path.join(dataset_dir, name)
    os.makedirs(person_dir, exist_ok=True)

    print(f"Capturing  images  for  {name}.  Look  into  the  camera.")
count = 0  # Image counter
```

Student Attendance Taking code

```
import streamlit as st import
cv2 import os import time
import numpy as np from
datetime import datetime from
deepface import DeepFace from
mtcnn import MTCNN import
pandas as pd
import tensorflow as tf

# Define directories
DATASET_PATH = r"D:\project_file\maincode\dataset3"  # Student face
dataset
ATTENDANCE_DIR = "Attendance"
```

```
# Ensure attendance directory exists if not
os.path.exists(ATTENDANCE_DIR):
    os.makedirs(ATTENDANCE_DIR)

# Load MTCNN detector
detector = MTCNN()

# Initialize session state variables if "page"
not in st.session_state:
st.session_state.page = "Login" if
"attendance_data" not in st.session_state:
st.session_state.attendance_data = {} if
"scan_count" not in st.session_state:
    st.session_state.scan_count = 0

# Function to recognize faces and mark attendance def
recognize_and_mark_attendance(frame, detected_students):
    results = DeepFace.find(img_path=frame, db_path=DATASET_PATH,
model_name="Facenet512", distance_metric="cosine",
enforce_detection=False)     if results and isinstance(results, list) and
len(results[0]) > 0:
        recognized_face = results[0].iloc[0]
        name = os.path.basename(os.path.dirname(recognized_face['identity']))

        if name not in detected_students:  # Only detect once per scan
detected_students.add(name)             if name in st.session_state.attendance_data:
st.session_state.attendance_data[name] += 1             else:
                st.session_state.attendance_data[name] = 1
```

```
            st.success(f"Detected {name}
({st.session_state.attendance_data[name]} times)")

# Function to finalize attendance after multiple scans
def finalize_attendance():     now = datetime.now()
    date_str = now.strftime("%Y-%m-%d")
    file_path = f"{ATTENDANCE_DIR}/Attendance_{date_str}.csv"

    file_exists = os.path.exists(file_path)

    with open(file_path, "a") as f:
if not file_exists:
            f.write("Name,Time,Date,Subject,Semester,Branch\n")

        for name, count in st.session_state.attendance_data.items():
            if count >= 2:  # Mark attendance if detected twice
                f.write(f"{name},{now.strftime('%H:%M:%S')},{date_str},{st.sessio
n_state.subject_name},{st.session_state.semester},{st.session_state.branch}\n")

    st.success("☑ Attendance saved successfully!") if
cap.isOpened():         while
st.session_state.scan_count < 3:  # 3 scans
            detected_students = set()
scan_start_time = time.time()
            while time.time() - scan_start_time < 10:  # 15 seconds per scan
ret, frame = cap.read()                 if not ret:
                    st.error("Error accessing camera!")
break
```

FINAL REPORT MAJOR[1]

ORIGINALITY REPORT

10%	6%	5%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	towardsdatascience.com Internet Source	1%
2	Viet Tran Hoang, Khoi Tran Minh, Nghia Dang Hieu, Viet Nguyen Hoang. "Chapter 51 Monitoring Employees Entering and Leaving the Office with Deep Learning Algorithms", Springer Science and Business Media LLC, 2022 Publication	1%
3	K. Kranthi Kumar, Y. Kasiviswanadham, D.V.S.N.V. Indira, Pushpa Priyanka palesetti, Ch.V. Bhargavi. "Criminal face identification system using deep learning algorithm multi-task cascade neural network (MTCNN)", Materials Today: Proceedings, 2021 Publication	1%
4	medium.com Internet Source	<1%
5	www.ijraset.com Internet Source	<1%
6	www.ijesrt.com Internet Source	<1%
7	www.coursehero.com Internet Source	<1%
8	Submitted to University of Hertfordshire Student Paper	<1%