FACE RECOGNITION ATTENDANCE SYSTEM

SUBMITTED

BY

K.SURYA

22COS64

Under the guidance of

Mrs. Mary Monica

Submitted to

The Department of Computer Science (UG)

FOR THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

AWARD OF THE DEGREE OF

B.Sc., COMPUTER SCIENCE

THE AMERICAN COLLEGE

(An Autonomous Institution Affiliated to Madurai Kamaraj University)

Re-accredited with (3rd cycle) by NAAC with Grade “A+”, CGPA – 3.47 on a 4-Point scale

MADURAI - 625002

APRIL - 2025

THE AMERICAN COLLEGE

(An Autonomous Institution Affiliated to Madurai Kamaraj University)

Re-accredited with (3rd cycle) by NAAC with Grade “A+”, CGPA – 3.47 on a 4-Point scale

MADURAI - 625002

APRIL - 2025

BONAFIDE CERTIFICATE

This is to certify that the project work entitled “FACE RECOGNITION ATTENDANCE SYSTEM” is a bonafide record of work done by K.surya (22COS64) in partial fulfillment of the requirements for the award of the degree of B.Sc., Computer Science of The American College. This work has not been submitted for the award of any degree to the best of our knowledge.

Submitted for viva-voce examination, held on \_\_\_\_\_\_\_\_\_\_\_\_, in the Department of Computer Science, The American College, Madurai.

Internal Guide Head of the Department

DECLARATION

I hereby declare that the project entitled “FACE RECOGNITION ATTENDANCE SYSTEM” is a project report of the original work done by myself. This project work is submitted to The American College (Affiliated to Madurai Kamaraj University) in partial fulfillment of the degree of Bachelor of Science in Computer Science during the academic year 2020 – 2023.

I declare that this project work or any part thereof has not been submitted for getting any degree or diploma from any other university or college.

Place : Madurai

Date :

Signature of the Student

## Project Abstract: Face Recognition Attendance System

The Face Recognition Attendance System is an automated solution designed to enhance the efficiency and accuracy of attendance tracking in educational institutions using facial recognition technology. Traditional attendance methods, such as manual roll calls or RFID-based systems, are time-consuming, error-prone, and vulnerable to proxy attendance, where one student marks attendance for another. To address these challenges, this system utilizes **MediaPipe**, an advanced machine learning framework, to detect and recognize faces in real time while incorporating **proxy detection mechanisms**. The system captures students' facial images through an IP webcam and processes them using **MediaPipe’s Face Detection and Face Mesh modules**, which analyze facial features and compare them with pre-stored data. To prevent proxy attendance, the system verifies unique facial landmarks, detects duplicate attempts within a short time frame, and flags suspicious activities, such as multiple faces appearing simultaneously or the same student attempting attendance more than once for different individuals. Once a valid face is detected and authenticated, attendance is automatically marked and stored in a CSV file. The system operates through a web-based interface, ensuring a seamless, contactless, and user-friendly experience. Additionally, it supports **subject-wise attendance marking**, ensuring that students are recorded present only for the relevant lecture or session. By integrating real-time facial recognition with intelligent proxy detection, this system provides a **highly secure, efficient, and fraud-resistant** attendance management solution, improving classroom discipline and reducing administrative workload in educational institutions.

## Hardware Requirements for Face Recognition Attendance System

1. **Processor**: Intel Core i3 (minimum) or higher / AMD Ryzen 3 or higher
   * A multi-core processor is recommended for efficient real-time face detection and recognition.
2. **RAM**: Minimum **4GB** (Recommended: **8GB or more**)
   * Face detection and recognition require sufficient memory to process video frames smoothly.
3. **Storage**: Minimum **256GB HDD or SSD** (Recommended: SSD for better performance)
   * Required to store datasets, attendance records, and system files.
4. **Webcam (IP or USB Camera)**:
   * **IP Webcam (preferred)** – For real-time face capture via mobile or network-based camera.
   * **USB Webcam** – If an IP webcam is not available, a high-resolution USB camera can be used.
5. **Graphics Processing Unit (GPU) (Optional but Recommended)**
   * A **dedicated GPU (NVIDIA GTX 1050 or better)** can improve processing speed for large datasets.
6. **Internet Connection (For Web-Based System)**
   * Required if using an IP webcam or cloud-based data storage.
7. **Power Supply**
   * A reliable power source to keep the system running during attendance tracking sessions.

## Software Requirements for Face Recognition Attendance System

1. **Operating System**:
   * **Windows 10/11 (64-bit)** (Recommended)
   * **Linux (Ubuntu 20.04 or higher)** (For better performance and flexibility)
2. **Programming Language**:
   * **Python 3.10.10** (or latest stable version)
3. **Libraries & Dependencies**:
   * **MediaPipe** (For face detection and recognition)
   * **OpenCV** (For image processing and webcam integration)
   * **NumPy** (For numerical operations)
   * **Pandas** (For handling attendance data in CSV format)
4. **Web Technologies** (For Web Interface):
   * **HTML, CSS, JavaScript** (For frontend)
   * **Flask / FastAPI** (For backend, if needed)
5. **Database (Optional, if using a structured database)**:
   * **SQLite / MySQL / PostgreSQL** (For storing attendance records instead of CSV)
6. **Development Environment**:
   * **Python IDLE / PyCharm / VS Code** (For coding and debugging)
7. **Browser Compatibility**:
   * **Google Chrome / Mozilla Firefox / Microsoft Edge** (For accessing the web interface)

## Existing System

In traditional attendance systems, students mark their presence through manual roll calls, ID card scanning, or biometric fingerprint systems. These methods have several drawbacks:

* **Time-Consuming**: Manual roll calls take significant time, especially in large classrooms.
* **Human Errors**: Mistakes in marking attendance or misplacing records can occur.
* **Proxy Attendance**: Students can fake attendance by answering for absent classmates.
* **Dependency on Physical Cards/Devices**: RFID-based systems require students to carry ID cards, which can be lost or damaged.
* **Hygiene Concerns**: Fingerprint-based biometric systems require physical contact, raising hygiene concerns, especially in post-pandemic scenarios.

## Proposed System

The Face Recognition Attendance System eliminates these issues by using **MediaPipe-based facial recognition** to mark attendance automatically. This system offers the following improvements:

* **Automated Attendance Marking**: The system captures students’ faces using an **IP webcam** and recognizes them without manual intervention.
* **Proxy Detection**: Ensures only the actual student is present by analyzing unique facial landmarks and detecting suspicious activity.
* **Contactless System**: Unlike biometric fingerprint systems, facial recognition does not require physical contact.
* **Real-time Processing**: MediaPipe enables fast face detection and recognition, ensuring quick attendance marking.
* **Web-Based Interface**: The system provides a user-friendly web interface to view attendance records.
* **CSV-Based Storage**: Attendance data is stored in CSV format, making it easy to manage and analyze.

## ****Module Design (Your Existing Program)****

### ****1. Dataset Creation Module****

* **Purpose**: Capture face images for training.
* **Tools Used**: MediaPipe
* **Functionality**:
  + Capture 40 face images per student using IP webcam.
  + Save images with name and roll number.
  + Store images in folders for each student.

### ****2. Face Recognition Module****

* **Purpose**: Identify student using camera feed.
* **Tools Used**: face\_recognition + MediaPipe
* **Functionality**:
  + Detect faces in webcam feed.
  + Compare with trained encodings.
  + Return matched name and roll number.

### ****3. Attendance Marking Module****

* **Purpose**: Mark attendance per subject.
* **Functionality**:
  + Subject buttons shown after login.
  + When a face is recognized, check if already marked for that subject.
  + If not present, mark name, roll number, subject, and time in CSV file.

### ****4. Web Interface Module****

* **Purpose**: Run everything through a browser.
* **Tools Used**: Python web framework (e.g., Flask) — no database used.
* **Functionality**:
  + Login screen (admin login only).
  + After login, show subject buttons.
  + Display live IP webcam feed with title **"Face Recognition"**.
  + Start recognition after selecting subject.
  + Automatically logs attendance without further clicks.

## ****Table Design****

### ****1. Attendance Table (CSV Format)****

This table stores attendance records for recognized students.

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| Name | String | Name of the recognized student |
| Timestamp | DateTime | Date and time when attendance was marked |

**Example CSV (e.g., attendance.csv)**:

Name,Timestamp

John Doe,2025-04-04 09:05:12

Jane Smith,2025-04-04 09:06:45

## ****Module Description****

### ****1. Dataset Creation Module****

* **Description**:  
  This module captures face images of students using an IP webcam. It uses **MediaPipe** for face detection to ensure accurate face cropping and alignment. The images are saved locally for training the face recognition model.
* **Key Features**:
  + Captures ~40 images per student.
  + Saves images in a structured directory format (dataset/Name/).
  + Ensures only one face is detected per frame.

### ****2. Face Recognition Module****

* **Description**:  
  This module performs real-time face recognition using the live webcam feed. It uses **MediaPipe** for detecting faces and **face\_recognition** (with dlib) for encoding and matching. When a student is recognized, their name is returned.
* **Key Features**:
  + Real-time face detection and recognition.
  + Matches faces with previously stored encodings.
  + Displays the recognized name on screen.

### ****3. Attendance Marking Module****

* **Description**:  
  After recognizing a face, this module marks attendance by recording the student’s **name** and the **current timestamp** in a CSV file. Duplicate entries for the same session are avoided.
* **Key Features**:
  + Records attendance only once per student per run.
  + Saves attendance in attendance.csv.
  + Stores: Name, Timestamp.

### ****4. Web Interface Module****

**Description**:  
A simple web-based interface allows the admin to log in and initiate face recognition. After login, the system displays the live webcam feed with a title **"Face Recognition"**, and starts capturing attendance automatically.

* **Key Features**:
  + Admin login (hardcoded or basic check).
  + Runs on local browser (no XAMPP/database).
  + Shows live feed from IP webcam.
  + No need for additional clicks after starting recognition.

## ****Types of Testing Applied in Your Project****

### ****1. Functional Testing****

* Ensures that the system performs its intended functions correctly.
* **Example in Your Project:**
  + Face detection and recognition should work correctly.
  + Attendance should be marked correctly in the CSV file.
  + The web interface should display the live IP webcam feed properly.

### ****2. Unit Testing****

* Tests individual modules or functions separately.
* **Example in Your Project:**
  + Checking if face detection works correctly using MediaPipe.
  + Testing whether the face recognition module returns the correct name.

### ****3. Integration Testing****

* Ensures that different modules of the system work together.
* **Example in Your Project:**
  + The dataset creation module should work correctly with the recognition module.
  + The attendance marking module should store data correctly after recognition.

### ****4. Performance Testing****

* Tests how well the system performs under different conditions.
* **Example in Your Project:**
  + Checking how fast the system detects and recognizes faces.
  + Ensuring the system can handle multiple faces in real-time without lag.

### ****5. Usability Testing****

* Ensures that the system is user-friendly and easy to operate.
* **Example in Your Project:**
  + The web interface should be simple and easy to navigate.
  + Subject buttons should be clearly visible after login.

### ****6. Security Testing****

* Ensures that the system is protected from unauthorized access.
* **Example in Your Project:**
  + The admin login should prevent unauthorized users from accessing the system.
  + Face data should not be stored in an unprotected location.

### ****7. Regression Testing****

* Ensures that new changes do not break existing functionality.
* **Example in Your Project:**
  + After adding face\_recognition to improve accuracy, testing to ensure MediaPipe still functions correctly.

### ****8. Acceptance Testing****

* Confirms whether the system meets project requirements and is ready for deployment.
* **Example in Your Project:**
  + Running the system in a real classroom environment to check if it successfully marks attendance.
* **Conclusion**
* The **Face Recognition Attendance System** successfully automates attendance marking using **Mediapipe and OpenCV** for face detection and recognition. The system accurately identifies students and records attendance in a CSV file while eliminating the need for manual entry.
* Security testing focused on **proxy attendance prevention** ensures that students cannot mark attendance using printed photos, digital screens, or pre-recorded videos. The system also prevents **duplicate entries**, enhances recognition accuracy in different lighting conditions, and ensures **real-time detection** using an IP webcam.
* Overall, the project provides a **secure, efficient, and user-friendly** attendance solution that can be further improved by adding **liveness detection** or **anti-spoofing mechanisms** in future versions.

## Future Enhancements

To further improve the **Face Recognition Attendance System**, the following enhancements can be implemented:

**1. Liveness Detection**

* Integrate **eye-blink detection** or **3D face movement analysis** to prevent attendance marking using printed photos, digital screens, or pre-recorded videos.

**2. Cloud-Based Attendance Storage**

* Instead of storing attendance in CSV files, integrate a **cloud database (Firebase, MySQL, or MongoDB)** to enable **real-time access and centralized storage**.

**3. Multi-Camera Support**

* support for Enable **multiple IP webcams** to cover **larger classrooms** and improve face detection accuracy.

**4. Improved Face Recognition Model**

* Replace the **LBPH recognizer** with **deep learning models** like **FaceNet or ArcFace** for better recognition accuracy.

**5. Mobile App Integration**

* Develop a **mobile app** for teachers to **monitor attendance**, view reports, and receive **real-time notifications**.

**6. Auto-Detection of Classroom Start Time**

* Implement a **schedule-based system** where the camera **automatically starts and stops** based on class timings.

**7. Improved UI for Subject Selection & Reports**

* Upgrade the **web interface** with a **dashboard** for students and faculty to view **attendance reports & analytics**.

**8. Face Recognition with Mask Detection**

* Implement **mask detection** to ensure **accurate recognition** even when students wear masks.

**9. Attendance Summary Reports**

* Generate **daily, weekly, and monthly** attendance reports in **PDF or Excel format**.

**10. Automatic Notification System**

* Send **email or SMS notifications** to students and faculty for **low attendance warnings**.

## Dataflow Diagram:

### DFD Level 0 – High-Level System Overview

Student Face Recognition System Attendance.CSV

 The student interacts with the system.

 The system recognizes the face and marks attendance.

 The attendance is stored in the CSV file.

### DFD Level 1 – Dataset Creation Process

Student Ip Webcam Feed Detection DatasetFolder

**1 Student provides face image** (via IP Webcam)

2 **Face is detected using Mediapipe**

3 **If valid, the face is saved in the dataset**

### DFD Level 1 – Recognition & Attendance Process

Attendance.CSV

Detection

IP Webcam Feed

Student

Proxy Detection?

Proxy Detected!

No Proxy

## ScreenPlates



