

Development of Smart School ID Card System for Maintaining Attendance Record

Shalini S⁽¹⁾ Varshni Kanna M⁽²⁾ Kirubakaran D⁽³⁾

^{(1) & (2)} students / EEE. St. Joseph's Institute of Technology, Chennai, India.

⁽³⁾ Professor/ EEE . St. Joseph's Institute of Technology, Chennai, India.

yeshali64@gmail.com

Abstract: This research presents the design of a Smart ID Card System for automating student attendance in schools. By leveraging advanced technologies, the system aims to eliminate the inaccuracies and inefficiencies associated with traditional manual attendance methods. The system utilizes facial recognition technology to identify students upon entering the classroom. If the student's face is detected within a predefined timeframe before the start of class, the system automatically marks the student as "present". This automated system eliminates the need for manual attendance marking, reducing human error and improving efficiency. Furthermore, it eliminates the need for physical contact, enhancing hygiene and minimizing the spread of germs. By integrating with existing school databases, the system can seamlessly record and manage student attendance data, providing valuable insights for school administrators. This innovative solution has the potential to revolutionize attendance management in educational institutions, improving efficiency, accuracy, and overall school operations.

Keyword Section: Facial Recognition, Smart Attendance System, Biometric Authentication, Encoding and Data Encryption, Performance Optimization, Machine Learning in Facial Recognition.

1.INTRODUCTION

The Smart School ID Card system brings numerous benefits as compared to the usual ID cards: it rationalizes administration, cuts the circle of human errors, and enhances the efficiency of school

functioning as a whole. Another important thing to point out about such systems is the enhanced creation of a more secure school environment by virtue of allowing access to particular areas of the school only to those who are authorized, thereby guaranteeing students' and staff's safety.

Facial recognition is a biometric technology used in identifying or verifying individuals by comparing facial features obtained from images or video frames. It finds its application in security systems, authentication of a person, and unlocking personal devices. This system detects a face in an image or any other video frame through algorithms that identify the eyes, nose, and mouth. Alignment of the detected face depends on key facial landmarks to ensure consistency in analysis. The system then extracts unique facial features by creating a numerical representation or encoding of those features. Comparing the extracted feature against a database of known face encodings, it searches for a match. The biometric sensor of the prevailing Smart Attendance System scans the ridges and analyzes the patterns in our fingerprints. Once it is registered, the user has to place the finger which he or she previously used to register, and the current scan will compare the two images. If they match, the user gets marked present, and the time he or she logged in will be noted as well.

The key objective of this system is to replace the biometric sensor-based system and manual attendance-based system with an advanced and more user-friendly system. The task of keeping track of the attendance of the students becomes much easier with the use of recent technologies, relieves a lot of loads off the teacher or admin in charge of attendance. This system could be employed in various wide areas,

including office spaces, schools, being the main purpose of the project, government buildings, etc. By automating this menial process of taking attendance, we are saving so much time when we can use it more productively, and the time at which a student logs in can also be precisely noted down for better accuracy. The project will improve the efficiency and maintenance of these.

2. PROPOSED SYSTEM

This system can be optimized for performance by optimizing the algorithms used in image processing and recognition. High-performance libraries such as OpenCV significantly reduce latency, while optimizing models for runtime can be achieved by using pre-trained, compressed models or by deploying on hardware with GPUs. Parallel processing and caching of frequently accessed encodings can improve response time to ensure that the system can handle multiple faces in real-time with minimal delays.

Multilayer authentication can be further augmented reliability with this attendance system that comes with facial recognition integration of an ID card reader scan. The software does feature robust error handling thereby interrupting it as minimum times as possible. Regular updations of the facial recognition model along with the database entry about the students will surely build its reliability through enhanced facial recognition accuracy and changes occurring within the student's faces in due course of time.

The accuracy of facial recognition needs to be improved to reduce false negatives, which are the failure to recognize a student, and false positives, which is marking attendance incorrectly. High-quality datasets of student faces under different lighting conditions and angles are required to increase accuracy. Training a robust recognition model on these diverse samples using techniques like data augmentation helps the model generalize better. Tuning parameters such as distance thresholds for recognition comparison further refines accuracy.

Facial feature identification is identifying and extracting unique landmarks of a face, such as the eyes, nose, mouth, and jawline. Using deep learning models like CNNs for facial recognition, trained on facial data, it detects these features within the input image that can encode accurately. Using pre-trained models such as dlib's face landmark detector or other facial recognition libraries identify these key points,

and a feature map is created to help precisely compare facial data.

First, it uses a captured image by the camera to determine if there exists a face in that particular image. Facially encoded, the face is then encoded into unique feature vectors and matched with the ones in the database, previously saved as encodings. If the predetermined threshold is met, the attendance would be marked by using the state-of-the-art facial recognition libraries so that through this system, a speedy and accurate recognition may be achieved.

The system needs to reduce variability in lighting, pose, and facial expressions to enhance precision. Preprocessing techniques could involve histogram equalization to normalize brightness and color. Higher threshold values for similarity checks can be applied in order to distinguish from similar faces with better precision. The model could also be fine-tuned with more samples and data being constantly updated.

3. SYSTEM DESIGN

3.1 ARCHITECTURE DIAGRAM

The architecture diagram for the smart attendance system involving face recognition is given in Figure 1 by ensuring secure, efficient, and automated attendance management.

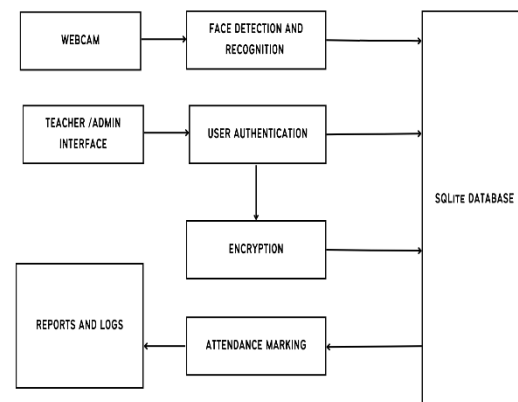


Figure1: Architecture diagram

The webcam in this system acts as the source initiator and captures a real-time video of students at the entrance. Later on, the video will be fed into and processed in the module called Face Detection & Recognition, where face representation will be done based on complex algorithms in order to identify and

extract unique facial features from the frames of the video. The identified faces along with the unique face encoding are then securely transferred to the SQLite Database, where it stores this along with student information and past attendance records. then securely transmitted to the SQLite Database, which stores this information together with student information and historical attendance records. Hence, user authentication ensures that only the authorized personnel, including administrators and teachers, are granted access to the system for sensitive operations.

Biometric data is encrypted with sensitive information, such as face encodings and attendance records, into the database for storage. It provides a layer of security to prevent unauthorized access and data breaches. The module 'Attendance Marking' will automatically update the attendance status as per the recognized faces and log into the database in real time with full accuracy, avoiding fraudulent attendance practices. Admin/Teacher Interface: easy interface to system administration-viewing and managing attendance records, enrolling new students, and other administrative tasks.

Also, the Reports/Logs component generates detailed reports of attendance and logs events occurring with the system for transparency. The arrows in the diagram indicate flow and interaction of data between the components. In this way, the entire attendance management flow is showcased in a very smooth and safe manner, while enhancing efficiency and accuracy, with protection assured for users data.

3.2 USE CASE DIAGRAM

A Use Case Diagram (fig.2) for the attendance system visually illustrates the interaction between different users and the system's functionalities. The key actors involved are the Admin, who manages and views attendance records on the dashboard, and the Student, whose attendance is marked through facial recognition. The camera/webcam plays a critical role by capturing real-time video for detecting and identifying students' faces. Once a student's face is recognized, the system marks their attendance by storing an encrypted name and timestamp in the database. This use case diagram for the Smart Attendance System incorporating facial recognition draws on the interaction of main actors, Students and

Admin/Teachers, with the functionality of the system to underline the comprehensive workflow and security measures underneath.

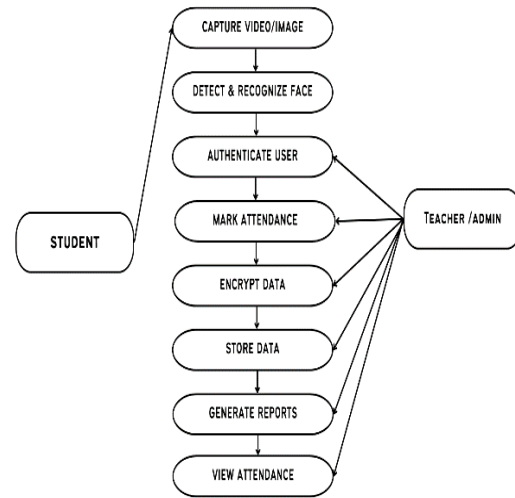
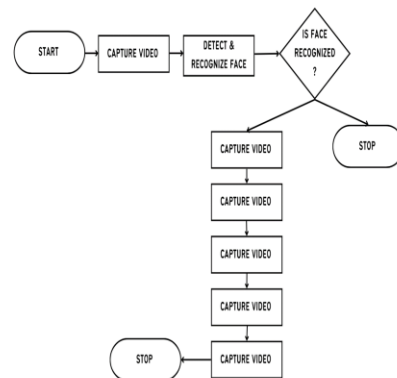


Figure 2: Use Case Diagram

In this system, students mainly interact by having their real-time video captured via Webcam as they enter the classroom. In turn, it triggers the use case Detect & Recognize Face, which executes extensive face recognition algorithms against the captured video frames and identifies unique facial features against the pre-stored face encodings in the database. This seamless process makes sure that each identity is being followed through with the process of the students' verification with no manual interference.

3.3 ACTIVITY DIAGRAM

Figure 3 shows the Activity Diagram for the given system that outlines how it operates



.Figure 3: Activity Diagram

It begins with the "Start" node, the very beginning of the operation of the system. The first activity is "Capture Video," describing that the webcam has to capture the real-time video of the students whenever they enter the classroom. Then, this video feed is processed through the "Detect & Recognize Face" activity, where advanced algorithms related to face recognition detect and identify the individual faces from among the captured frames. After that, the flow goes to the decision node represented by "Is Face Recognized?", which then checks if the face detected matched any stored face encoding in the database. If the detected face matches, it proceeds to the "Mark Attendance" activity, automatically recording the attendance of the student.

To ensure that unique facial features of each student are stored and recognized precisely, the encoding process converts distinctive facial characteristics into a digital form. Using facial encoding libraries, such as the `face_recognition` library within Python, each face has been analyzed for key landmark positions, such as positions of eyes, nose, and mouth. These mapped to a 128-dimensional feature vector that represents this face uniquely. In the marking of attendance, each new image is encoded against stored encodings for speedy and secure matching.

This is further followed by the "Encrypt Data" activity to secure the data so that sensitive information in the form of biometric data is not easily hacked. The data will further be persisted in the database for which the "Store Data in Database" activity is used.

3.4 COMPONENT DIAGRAM

The Smart Attendance System facial recognition component diagram embeds the main components and their interaction, emphasizing how the system is modular and with clear separation of concerns. The User Interface shall be the main interaction with admins and teachers for authentication, viewing of records of attendance, and reporting. This module interacts with the Face Detection & Recognition module, which is intended for video capturing through a webcam, face detection, identification of persons, and marking their respective attendance. The Webcam module provides the video feed required for face detection and recognition.

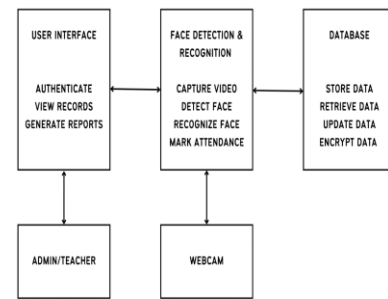


Figure 4: Component Diagram

The Face Detection & Recognition module also acts as an interface to the Database module, which itself provides basic create, read, update, and delete operations. The database uses encryption to make the data with regard to biometric and attendance information secure. Accordingly, the Database module provides methods to securely store the data, retrieve it for reporting and viewing, and update attendance based on the recognition results given from the face detection system.

Figure 4 shows the interaction between components by showing that the User Interface depends on face detection to give perfect records of attendance, and this also reflects how the database ensures stored data integrity and security. By keeping this segregation of responsibility in place, this architecture provides the capabilities for easy maintenance, scalability, and robust performance of the smart attendance system.

4. PROGRAM

4.1 ENCODE FACE PROGRAM

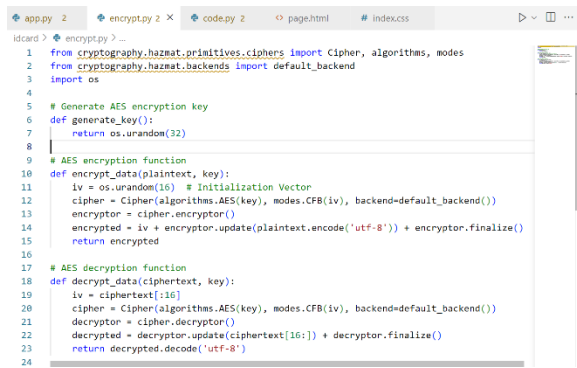
```

1 from flask import Flask, render_template
2 import sqlite3
3 from encryptor import encrypt_data, generate_key
4
5 app = Flask(__name__)
6 key = generate_key() # Generate AES key for encryption
7
8 def get_attendance_records():
9     conn = sqlite3.connect('attendance.db')
10    cursor = conn.cursor()
11    cursor.execute("SELECT * FROM attendance")
12    records = cursor.fetchall()
13    conn.close()
14
15    # Encrypt each record for display

```

Fig 5: ENCODE FACE PROGRAM

4.2 LIBRARIES AND IMPORT PROGRAM



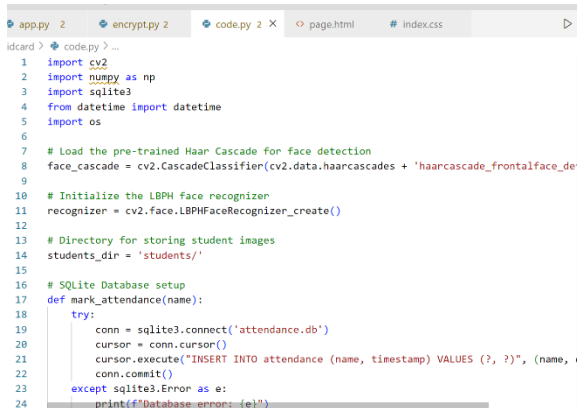
```

1 from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
2 from cryptography.hazmat.backends import default_backend
3 import os
4
5 # Generate AES encryption key
6 def generate_key():
7     return os.urandom(32)
8
9 # AES encryption function
10 def encrypt_data(plaintext, key):
11     iv = os.urandom(16) # Initialization Vector
12     cipher = Cipher(algorithms.AES(key), modes.CFB(iv), backend=default_backend())
13     encryptor = cipher.encryptor()
14     encrypted = iv + encryptor.update(plaintext.encode('utf-8')) + encryptor.finalize()
15     return encrypted
16
17 # AES decryption function
18 def decrypt_data(ciphertext, key):
19     iv = ciphertext[:16]
20     cipher = Cipher(algorithms.AES(key), modes.CFB(iv), backend=default_backend())
21     decryptor = cipher.decryptor()
22     decrypted = decryptor.update(ciphertext[16:]) + decryptor.finalize()
23     return decrypted.decode('utf-8')
24

```

Fig 6: LIBRARIES AND IMPORT PROGRAM

4.3 ATTENDANCE (KNOWN_ENCODINGS, KNOWN_NAMES) PROGRAM



```

1 import cv2
2 import numpy as np
3 import sqlite3
4 from datetime import datetime
5 import os
6
7 # Load the pre-trained Haar Cascade for face detection
8 face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_def
9
10 # Initialize the LBPH face recognizer
11 recognizer = cv2.Face_LBPHFRecognizer_create()
12
13 # Directory for storing student images
14 students_dir = 'students/'
15
16 # SQLite Database setup
17 def mark_attendance(name):
18     try:
19         conn = sqlite3.connect('attendance.db')
20         cursor = conn.cursor()
21         cursor.execute("INSERT INTO attendance (name, timestamp) VALUES (?, ?)", (name, d
22         conn.commit()
23     except sqlite3.Error as e:
24         print(f"Database error: {e}")

```

Fig 7: ATTENDANCE PROGRAM

4.4 CODE EXPLANATION :

This code is a face recognition-based attendance system that uses 'OpenCV' to capture video, 'face_recognition' to detect and match faces, and 'SQLite' to store attendance records. It begins by initializing the webcam using the 'init_camera()' function and raises an exception if the camera is inaccessible. The 'load_known_faces()' function loads face encodings and names from images stored in the "students/" folder, using the filenames as identifiers. If no face is detected in an image, a warning is shown. During execution, the 'recognize_faces()' function continuously captures frames from the video stream, detects faces, and compares them with known faces. If a match is found, it marks the person's attendance by inserting their name and timestamp into an SQLite database using the 'mark_attendance()' function. The system also labels recognized faces in the video feed with bounding boxes and names, displaying "Unknown" for

unrecognized faces. The video stream is shown in real time, and the program stops when the user presses 'q'. Proper error handling ensures that exceptions, like database issues or camera failure, are caught and reported.

This HTML code defines a simple Attendance Dashboard webpage that dynamically displays encrypted attendance records using Jinja2 templating (commonly used in Python web frameworks like Flask). The '<head>' section contains metadata, sets the character encoding to UTF-8, and includes a '<link>' to an external CSS stylesheet ('style.css') located in the 'static' folder, using Flask's 'url_for()' function to generate the correct path. The '<body>' section begins with a heading ('<h1>Attendance Records'), followed by a table to display the attendance data. The table has two headers: one for the encrypted name and the other for the encrypted timestamp. The '{% for record in records %}' loop iterates over the 'records' list passed from the backend (likely a Python Flask route) and populates each row of the table with the encrypted data for every attendance entry. Each entry is displayed inside '<td>' tags, representing the name and timestamp. The code ends with the closing tags for the loop and table. In summary, this webpage fetches encrypted attendance records from the server and displays them in a structured, readable format using HTML and Flask's template syntax.

5. RESULTS AND DISCUSSION

The Smart Attendance System successfully automates the process of marking attendance using facial recognition technology. Upon execution, the system captures live video from the connected camera, detects faces in real-time, and matches them against the pre-registered student images. If a student's face is recognized, their attendance is automatically recorded in the SQLite database with a timestamp, ensuring accurate and tamper-proof logging. Unrecognized faces are labeled as "Unknown" on the video feed, providing instant feedback to administrators. The system operates efficiently, offering quick recognition, real-time feedback, and ease of access to attendance records.

This eliminates the need for manual roll calls, reduces human error, and enhances the overall efficiency of the attendance management process. The outcome

demonstrates a reliable, user-friendly, and scalable solution that leverages modern computer vision technologies to streamline attendance tracking in educational institutions.

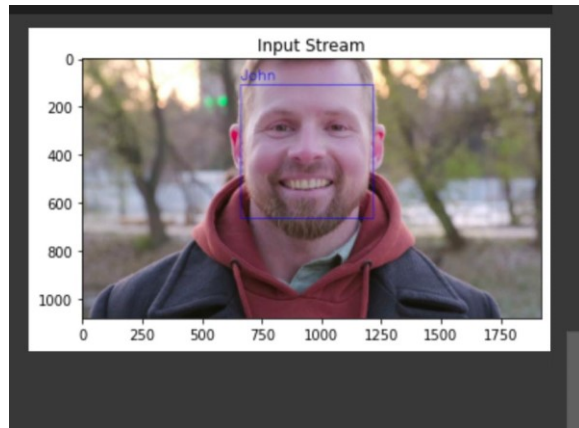


Fig 8: Sample Output

6. CONCLUSION

This research presents a novel Smart ID Card System designed to automate student attendance in schools, addressing the limitations of traditional manual methods. By integrating facial recognition technology, the system ensures accurate and efficient attendance tracking. The automated process eliminates the need for manual intervention, thereby minimizing human error and enhancing hygiene. Moreover, the seamless integration with existing school databases facilitates efficient data management and provides valuable insights for school administrators. This innovative solution has the potential to significantly improve attendance management in educational institutions, leading to increased efficiency, accuracy, and overall operational effectiveness.

6.REFERENCES:

- [1] Hapani, Smit, et al. "Automated Attendance System Using Image Processing." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.
- [2] Akbar, Md Sajid, et al. "Face Recognition and RFID Verified Attendance System." 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE). IEEE, 2018.

[3] Okokpujie, Kennedy O., et al. "Design and implementation of a student attendance system using iris biometric recognition." 2017 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2017.

[4] Rathod, Hemantkumar, et al. "Automated attendance system using machine learning approach." 2017 International Conference on Nascent Technologies in Engineering (ICNTE). IEEE, 2017.

[5] Siswanto, Adrian Rhesa Septian, Anto Satriyo Nugroho, and Maulahikmah Galinium. "Implementation of face recognition algorithm for biometrics based time attendance system." 2014 International Conference on ICT For Smart Society (ICISS). IEEE, 2014.

[6] Lukas, Samuel, et al. "Student attendance system in classroom using face recognition technique." 2016 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2016.

[7] Salim, Omar Abdul Rhman, Rashidah Funke Olanrewaju, and Wasiu Adebayo Balogun. "Class attendance management system using face recognition." 2018 7th International Conference on Computer and Communication Engineering (ICCCE). IEEE, 2018.