5/12/2024

TANVEER AHMAD

superior univeristy lahore

Student Attendance System Using Face Recognition

1. Abstract:

* This project focuses on automating attendance systems in educational institutions using face recognition technology. By utilizing Python, OpenCV, and Firebase, the system captures student images, encodes their facial features, and maintains attendance records efficiently. The system ensures accuracy, reduces manual intervention, and offers real-time updates to the database.

1. Introduction:
   1. **Background**
   * Attendance systems are crucial for maintaining discipline and tracking student participation. However, manual methods are time-consuming and prone to errors.
   1. **Problem Statement**

* The traditional methods of taking attendance are inefficient and susceptible to inaccuracies, such as proxy attendance.

**3.3 Objective**

* To design and implement a face recognition-based attendance system that is automated, accurate, and user-friendly.

### **Literature Review**

* **Existing Systems**: Manual attendance
* **Limitations**: These systems require physical interaction, are time-consuming, or involve additional costs.
* **Proposed Solution**: Face recognition leverages existing hardware (cameras) and is more efficient.

### **5. System Design**

#### ****5.1 Architecture****

* **Input**: Captures images of students via a webcam.
* **Processing**: Processes images to extract facial features using the face\_recognition library.
* **Storage**: Saves data in Firebase Realtime Database.
* **Output**: Updates attendance records and generates reports.

#### ****5.2 Modules****

1. **Add User**:
   * Captures user information and images.
   * Encodes face data and stores it in the Firebase database.
2. **Mark Attendance**:
   * Detects faces in real-time video streams.
   * Matches detected faces with stored data.
   * Updates attendance in the database.

#### ****5.3 Workflow Diagram****

1. User inputs details and uploads images.
2. System processes the image for face encoding.
3. Real-time face detection matches stored encodings.
4. Attendance is updated in Firebase

### **6. Tools and Technologies**

* **Programming Language**: Python
* **Libraries**:
  + cv2 (OpenCV)
  + face\_recognition
  + firebase\_admin
  + cvzone
* **Database**: Firebase Realtime Database
* **Hardware**: Webcam or external camera

### **7. Implementation**

#### ****7.1 Add User Module****

Steps:

1. Capture user data (name, roll number, year, and department).
2. Capture and process an image using OpenCV.
3. Encode face data using face\_recognition and save it in Firebase with other user details.

#### ****7.2 Mark Attendance Module****

Steps:

1. Fetch user data from Firebase.
2. Detect faces in a live video stream.
3. Compare detected faces with stored encodings.
4. Update attendance records for matched users.

### **8. Code Structure**

* **AddUser Class**: Registers new users with image encoding and Firebase storage.
* **Database Class**: Handles Firebase operations (add, fetch, update).
* **Camera Class**: Detects and processes live video feeds for attendance marking.

### **9. Challenges Faced**

1. **Lighting Issues**: Solved by adjusting image preprocessing techniques.
2. **Multiple Faces**: Enhanced accuracy by fine-tuning face recognition parameters.
3. **Database Overload**: Optimized Firebase queries for real-time performance.

### **10. Conclusion**

This project successfully demonstrates a reliable and automated solution for attendance management. The system reduces manual errors, improves efficiency, and leverages state-of-the-art technology.

### **11. References**

* OpenCV Documentation: https://docs.opencv.org/
* Firebase Documentation: https://firebase.google.com/docs
* Face Recognition Library: <https://github.com/ageitgey/face_recognition>