Introduction

The IoT based Face Recognition Attendance Management System aims at replacing the current attendance systems and thus automating almost the entire process. By using Machine Learning and Computer Vision supported by IoT based cameras at the hardware level. At the touch of a button, pictures will be taken by all nodes present, sent to a local server for analysis and you have the attendance of the entire class, which is accessible to the teachers and the students via an Android app.

Objectives

1. Inefficiencies in Current Systems:
    a. Biometric (Fingerprint or IRIS) System:
        i. Fingerprint application and device price a lot more.
    b. RFID (Radio Frequency Identification) :
        i. Privacy is a concern with the use of RFID on products as it can be easily tapped or intercepted.
        ii. The external electromagnetic interference can limit the RFID remote reading.
        iii. The coverage range of RFID is limited which is about 3 meters.
    c. Android App Facial Recognition System:
        i. Here the picture is taken via an android phone and the onus is on the teacher to make sure that the pic is taken properly
        ii. Even after the pic is taken, there exists a possibility the camera did not cover the entire class
    d. Manual System:
        i. Manual System can be responsible for many inconsistencies and is highly unrecommended.
2. How we will address these issues:
    a. Designing a cost efficient system using a low level board

    b. Easy to use and maintain on the server using utility programs
    c. Secure and non-hackable by choosing a non-hackable board
    d. Simple setup using Android apps
    e. Provide real time monitoring the apps
    f. Increase class coverage with 4 cameras clicking from 4 different perspectives.

Deliverables

The solution consists of the following components with the corresponding features

1. Teacher app:
    a. Registration/Login
    b. New course button (year, class, department, type of course(whether compulsory or elective))
    c. Delete course button
    d. Take attendance button for each course
    e. Notification for successful picture click
    f. Status of work
    g. Manual mode of taking attendance(in case some students get missed)
    h. Change attendance manually
2. Student app:
    a. Registration/Login
    b. Upload face pic
    c. Courses list to enroll at the start of the academic year
    d. Push notifications to tell the student about his/her attendance status(present/absent) so that students in the class marked absent can report it to the faculty
    e. Calendar
    f. Course wise total attendance
3. Desktop app:
    a. Used to map static IPs to devices in each classroom

b. Form for making entries of classrooms and IPs

c. The IPs can also be reassigned using the desktop app

4. Backend:
    a. Student, Teacher, Course registration
    b. Receiving requests from faculty app(attendance, manual mode etc.)
    c. Directs the boards in that class to take pics and request pics from the boards.
    d. Given the pics it will identify students from those who are enrolled for that course
    e. Post identification, marks attendance in database, notify faculty about attendance count, notify each student enrolled in that course about status of attendance (present/absent)
    f. Serves the student dashboard (calendar counts, total counts)
5. Database:
    a. List of courses (every year new course)(total number of lectures)
    b. List of students
    c. Student course and attendance mapping
    d. Classroom and IP mapping
6. IoT:
    a. NodeMCU/MSP/DSP Boards Interfaced with an CMOS/OV Camera. 4 boards in all, two near the blackboard on either side, two in the middle of the classroom.

Benefits

1. A complete Self-Contained system with a Node provided for taking pictures
2. The Node would be a NodeMCU/MSP/DSP Board interfaced with an CMOS/OV camera which would directly communicate with our local server, due to lack of OS and pre-determined tasks and code allocated to board, the connection would be safe not hackable via network. It would also be interfaced with a (Rechargeable Lithium) battery which would provide as an independent source of power.
3. Real Time access to static database for Teachers and Students to point out irregularities (if a student is present but wasn't recognized) via Android App

4. Separate teacher and student app to avoid giving students access to the teacher app
5. Android App would function in both Manual(for editing by Teachers) and Automatic Mode
6. Facial Features are Unique and it wouldn't be possible to fake them in anyway
7. Since the Pictures would be taken on demand (via push of a button on Android App) by teachers real time data would only be provided for analysis

8. Cost benefits:
    - **a.** Cost to make**:**
        - **i.** Camera Cost: 1.5k-3k (Depending on the resolution and stuff)
        - **ii.** NodeMCU: 0.6k
        - **iii.** Effective Cost: 2.1 - 3.6k
        - **iv.** Note these are Prototype Cost .Once they are Streamlined and mass manufactured, the cost goes way down.
    - b. Cost Comparison

System	**Cost**
Finger Print Scanner	**3-5k**
RFID Scanner	**3-6k**
Iris Scanner	**8k-**
Our System	**2.1-3.6k**

Architecture and Implementation:

The apps and all the boards will be interacting with central server. This server will be running a Django backend connected to a MySQL database and will have information about the static IPs of all the boards in each class in the form of a classroom:IPs mapping as well as the attendance records.

The classroom:IPs mapping can be created easily using the desktop app which will run on the machine hosting the central server.

While initiating the attendance session the classroom and the course have to be given as a parameter. Using the classroom:IPs mapping, the server will instruct the boards in that class to take the pictures and send it to the server. Upon receiving the images the server will perform the face recognition from the students enrolled in that course. After finding the students, the server will change the attendance count in the database for that course and notify the students and teachers. The process of manual attendance marking is similar, except that upon receiving the request the server will change the count directly.