Face attendance check*

Yaohan Teng
Department of Computer Science
Binghamton University
yteng1@binghamton.edu

ABSTRACT

Nowadays, to let Students attend classes actively, many professors choose to do attendance check during class. But calling the roll is not an efficient way to do it, which will waste a lot of class time. So, I have this idea: Implement an app which do attendance check efficiently by face recognition. This app is used for professors to do attendance check much easier.

KEYWORDS

Face recognition, Attendance check

1 Introduction

This app is based on face recognition for attendance check. Professors can use this app to do attendance check efficiently.
Firstly, it's much easier to manage students' information. They can upload students' face which binded with their IDs and names. Secondly, it's fast to do attendance check by camera of the phone in a few seconds. Third, it's easy to search attendance check records, even records in specific time. Fourth, it's convenient to delete students' info in the database when they drop the class.

2 Design

2.1 Hardware components

1. Telephone with Android system

2. Camera of Telephone

2.2 Software components

1. Android Studio (The Latest version)

2. Java language

3 Implementation

3.1 Register && Login && Forget Password

(1) Username + Password + Tel-number = Register,

(2) Username + Password = login

(3) Username + Tel-number = Find Password Back.

3.2 Database: SQLite

SQLite Database is a build in light-weight database, which is famous for fast, convenient and concise. So, I choose to use SQLite to store these INFO.

3.2.1. In LoginHelper.java: I created a log_table in database and exec in onCreate function. InsertLogin (String username, String password, String phone) for insert new user; function selectLogin () for searching in the table by SQL statement select and cursor.

3.2.2 In Myhelper.java: I created two tables: name-id table and time-id table, and exec. And respectively, there are two functions: insert_name_id and insert_time_id for insert operation in these two tables. Besides, there is another function for deleting tables.

3.3 Upload students' faces.

Students have two different ways to upload their faces after filling in their IDs and Names: (1) Use camera to take a picture. (2) Use Photos already in the Albums. Actually, Students just need to upload their faces at the first time they take the class. And these pictures will be used for attendance in the future. For the rest of the time, they just need to do attendance check.
In this step, we need to use Baidu Api: every time we need to use Client ID and Client Secret to get access token. Then we transfer the image to base64 and use map to store other info (include student ID) which need to be uploaded to the server. Then we use http to post the info to Baidu server. Here we use the student ID as the face image's ID to store in the server for convenience in checking step. At last, we insert this info into the name-id table.

3.4 Attendance Checking

Different from uploading faces, students can only choose to use their camera to do attendance check, which means they cannot use photo in their albums. Similar to the uploading step on upload students' picture for checking. Then we can get face_list in return result, then we can check all the face in the face_list to see if there

is one face which score is more than the threshold value that we set, which means this current face is in the face_list. So, according to the ID of the picture in the face_list, we know which student just finished the face check.

3.5 Search Operation

Search operation is pretty easy, just use SQL statement select and cursor to search in our database. Return info when found, return error message when not found.
Several functions implemented for Searching in database.
(1) Show all saved faces in Database: ID + Name;
(2) Show all records of attendance check: ID + Check time;
(3) Search Check INFO by Time: Show all records in selected time slot: ID (all students) + Check time;
(4) Search Check INFO by ID and Time: ID (students with selected ID) + Check time;

3.6 Delete Operation

Some of students may drop this class if they don't want to take this course anymore, So Professors can delete these students' faces by ID easily. Similar to uploading step and attendance check step, when we delete the student info, we still need to use our client ID and client Secret to get access token. And delete the face by ID. Here the student's ID is the same as the ID of the face picture.

3.7 Exit

Exit the app.

4. Evaluation:

In this project, I learned how to develop an app by Android Studio the first time, but it's good for me that I'm kind of familiar to Java language. But due to the time limited, there is still something need to be improved. Firstly, because in the face recognition part, we need to upload pics and receive the error code from the server, so it takes a few seconds depends on the quality of the internet. Secondly, for convenience, I choose to use SQLite as database. But SQLite is a lightweight database, so it's not very suitable for too much data. And I will modify this in the future.

REFERENCES

[1] Baidu api for face recognition:
     https://cloud.baidu.com/doc/FACE/s/7k37c1ucj
[2] Android Studio Tutorial:
     https://developer.android.com/training/basics/firstapp