FaceFinder: Attendance Made Easy

Prerequisites

- A device with a camera.

- A Python environment with the necessary libraries:

1.Face Recognition

2. OpenCV

3.Flask

4.os

5.pickle

6.numpy

7.pymongo

8.datetime

9.bson

10.pandas

11.xlswriter

12.io

13.collections

14.flask\_mail

15.pathlib

16.requests

17.google.oauth2

18.google\_auth\_oauthlib.flow

19.pip\_vendor

20.google.auth

21.functools

Overview:

The main objective of our project is to revolutionize the attendance system and automate this tedious process. In our project , the user can simply pass an image and get the attendance of the whole class. Apart from this, the user can also connect the CCTV camera of class and students will simply have to look at the camera and their attendance will be marked. We have also tried to ease the user interaction by adding a lot of features in which different users can create different classes and the attendance of each class will be saved separately.

Our project uses data structures like array and hash map to store data of students and classes and we have simplify the access of data in which user can download the excel file of the required data.

We have achieved a remarkable accuracy of 94% in our project.

How It Works:

Although the process of face recognition requires very complex computing but in simple terms, it involves face detection, face encoding, face comparison, marking attendance, storing attendance.

Now let’s understand each process in detail.

1.face detection: We have used face-recognition library. To detect face, it uses HOG (Histogram of oriented Gradients) . HOG divides the images into small dells calculate gradient histograms for each cell. It uses sliding window technique to go through the whole image. It then checks if a particular region is a potential face or not. After the detection, we can get face landmarks(nose, ears, eyes) using face\_recognition.face\_location.

2.Face encoding: face recognition library uses CNN(convoluted neural network) in which we get a vector of different values for a face and these values are what we call encodings. It can have 128 values for a face.

3.Face Comparison: the next part is face recognition. We have saved the encodings of all students of the class separately. Now, we will get new encodings of the faces present in the input video or image. We will compare the face encodings of the faces in input image with each photo in the database. Then we get a particular image which is the most accurate match and his attendance is marked.

4. Storing Attendance: After a face is recognized we will store the value of that roll no in an array and that array is then saved.

DATA STRUCTURE:

We use data structures like array , hash-map, hashing, that keep attendance records organized, easily accessible, and scalable.

FEATURES:

We offer a wide variety of features.

We have added a google authorization in which only the allowed users will be able to login . Apart from this, a user can create many classes and take attendance for each class separately just like google classroom. There are also features in which a user can delete a class or edit the name of the class. After login and entering into a particular class user will see the home page of our website.

On the home page, there is a navbar in which there is a button to take attendance.

User can take attendance in two ways:

1.User can upload an image and get attendance of all the students. User can see the roll no. of all the students present in the class on the website too.

2.User can take attendance using video in which he can connect the site with a video camera and will get attendance of all the students present in the frame of the camera. The live video can be seen in a separate pp up prompt which appears after the user clicks on the button.

After taking attendance, the user can see the records of students.

There are three ways a user can access the data inn excel format:

1. The user can see the attendance record of all students on a particular day.
2. The user can see the record of a particular student and see the dates on which that student was present.
3. The user can give a prompt (like 50) and he will get the roll no’s of students who have attendance less than or equal to that that prompt(like students whose attendance is less than or equal to 50%).

Apart from this, the user can also send an email to each student whose attendance is less than or equal to the prompt(for eg. : if user gives prompt of 50, user will be able to send mail to students whose attendance is less than or equal to 50%).

MongoDB:

We have used MongoDB to implement google classroom like model in which user can add, delete, edit classes and get or take separate attendance for each class.

Let’s see the implementations we have done using Mongo DB:

1.There is a database called subjects which contains the name of all the subjects present.

2.ADD operation: We can add a subject in the database by inserting the subject name in the ‘subjects’ database .Also, a separate database of that subject is created in which all the attendance record can be stored( Please NOTE that we have implemented attendance using data structure also and all the excel files that are downloaded is implemented using data structures only).

3.EDIT operation: If a user by mistake created a subject with wrong name, he can simply edit the name of the subject. This feature is implemented by first creating a separate database with new name and then we copy the old data from old named database to new named database and then delete the previous old database. Also in the ‘subjects’ database, the old database’s name is deleted and new database’s name is inserted.

4.DELETE operation: If a user wants to delete a particular subject, he can do so easily. It is implemented by first deleting the subject’s name from the ‘subjects’ database and then the database of that particular subject is also deleted.

BACK-END:

We have used Flask framework to implement back-end in our project .All the route integrations, requests(POST,GET),passing variable to html files(websites), etc. are done using flask framework.