

**FACE RECOGNITION
BASED ATTENDANCE
SYSTEM**

Table of contents

1. INTRODUCTION

2. REQUIREMENT ANALYSIS

3. DESIGN

4. IMPLEMENTATION

5. SNAPSHOTS

INTRODUCTION

In modern times, Automatic Face recognition have implemented with different types of algorithms. In (AFR) technologies have seen dramatic improvements in performance over the last few years. There are two reasons for this trend; the first is for saving the time in the classroom and accuracy in attendance will be maintained, and the second is availability of advanced technology it is more useful for the future generation. In simple words, it was a computer implementation for recognizing automatically whether the student is present in the classroom or not with the help of still image or video frame.

Attendance is one of the important aspects in the classroom. At the beginning and ending of the section, the teacher should check out the students presence in the class generally teachers take the attendance and note done in the register, but it may appear that miss someone and students any give attendance multiple times.so to avoid this type of

We proposed an automatic attendance management system. It was completely based on face recognition and the face detection. This both detection and recognition will efficiently mark attendance

CHAPTER 2

REQUIREMENTS ANALYSIS

The requirement analysis specifies the requirements needed to develop a graphic project. In this phase, we collect the requirements needed for designing the project.

The requirements collected are then analyzed and carried to the next phase.

2.1 SOFTWARE REQUIREMENTS:

- 1. Operating System:** Windows 10
- 2. Scripting Language:** PYTHON
- 3. Front-end Development:** TKINTER
- 4. Back-end Development:**
PYTHON & MYSQL

2.2 HARDWARE REQUIREMENTS:

1. Processor :

Pentium IV or above

2. RAM :

2 GB or more

3. Hard disk :

5GB or more,to train data efficiently

4.CAMERA:

Camera with Intel processor anGPU with at least 8GB RAM can **run face recognition** smoothly. This is the standard

CHAPTER 3

DESIGN

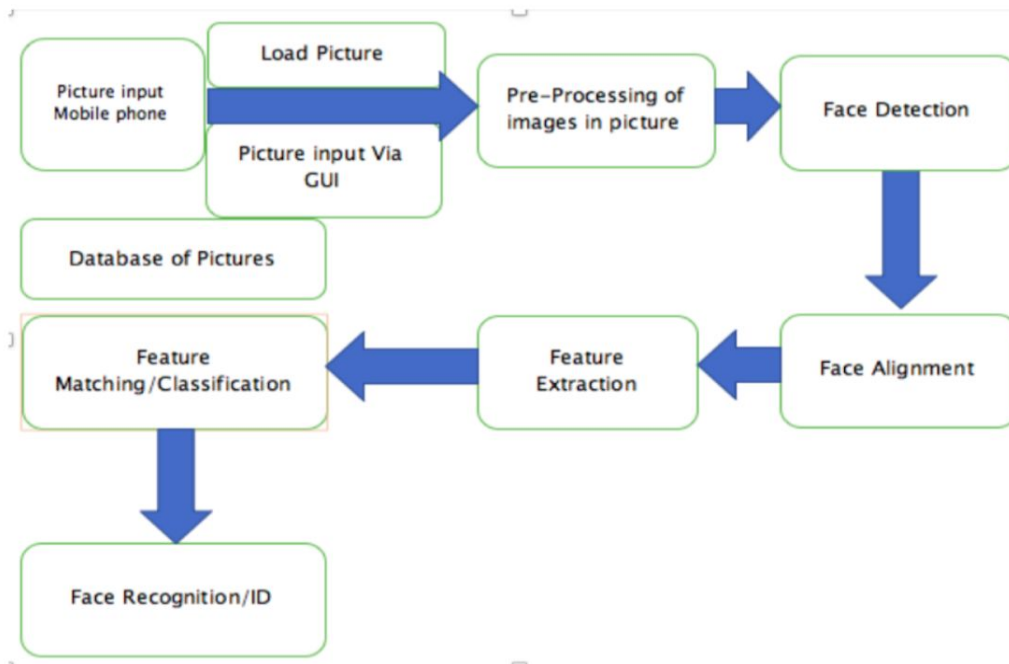


Figure 2.3 Face Detection and Recognition Flow Diagram.

Figure 2.3 Face Detection and Recognition Flow Diagram.

From the figure, above, Face Detection or face detector will detect any given face in the given image or input video. Face localization, will detect where the faces are located in the given image/video, by use of bounding boxes. Face Alignment is when the system will find a face and align landmarks such as nose, eyes, chin, mouth for feature extraction. Feature extraction, extracts key features such as the eyes, nose, mouth to undergo tracking. Feature matching and classification. matches a face based on a trained data set of pictures from a database of about 100 pictures. Face recognition, gives a positive or negative output of a recognized face based on feature matching and classification from a referenced facial image.

CHAPTER 4

FEATURES:

This project can be divided into three major parts:

- 1.Registration
- 2.Attendance marking
- 3.Prediction

4.1:REGISTRATION:

step 1:

Registration in this project consists of a student entering the required details:-

- ID
- Name
- Semester
- Section

These details are then fed into our local database using pythons 'mysql.connector',this allows us to connect to the given database.

Our database name "student" contains table "studentdetails" which stores each students information in the respective column.We have made use of XAMP to create a local server and provide us with DBMS functionality

Step 2:

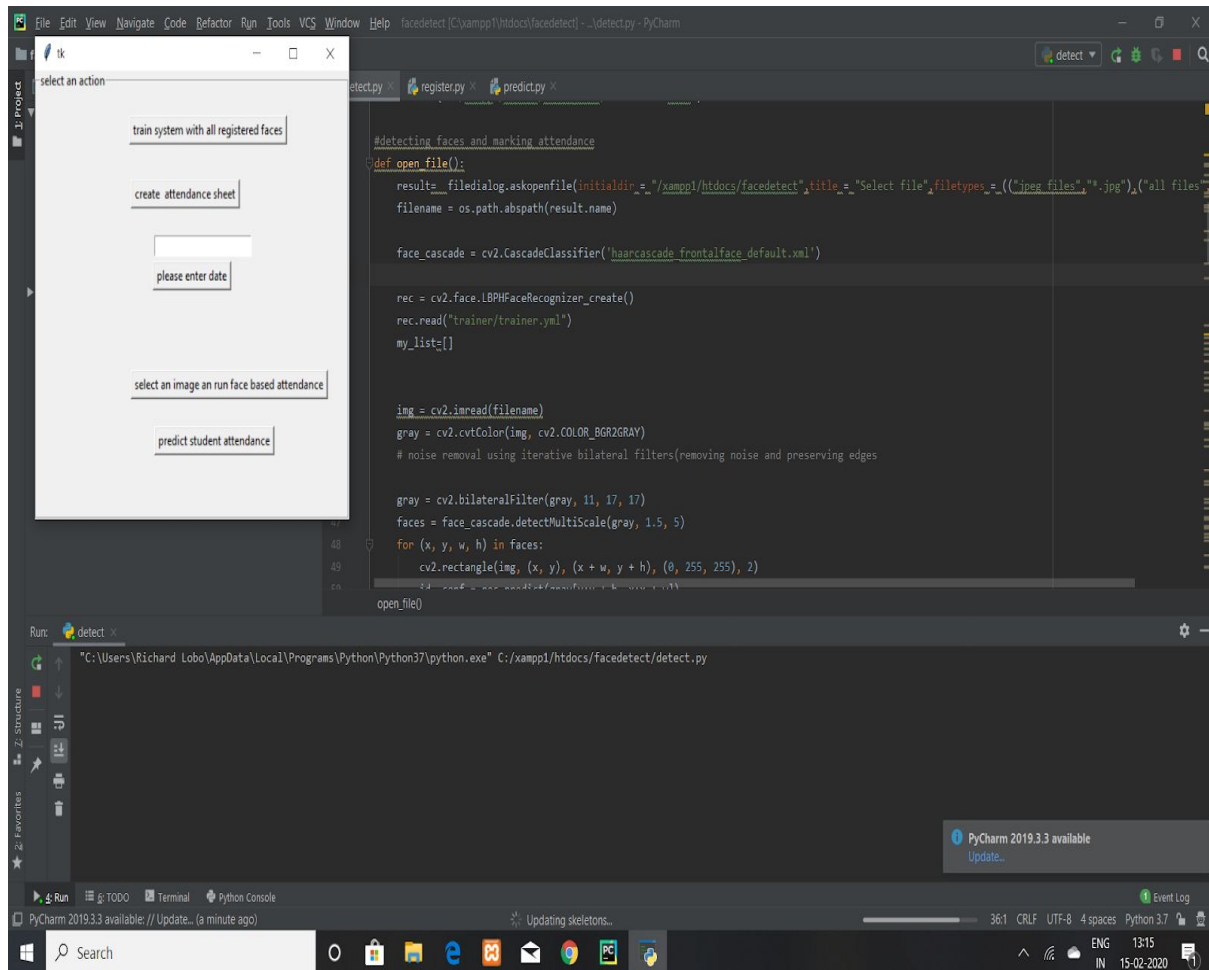
After the given steps are complete the program starts collecting samples of each students face, this is done using the library CV2

These samples are images of the students face and contain his ID and sample number, these samples are stored in the file "dataset"

All students must complete registration phase before moving ahead, we have implemented it in such a way to mimic the running of a school as, all students who must register are already known and new Students are seldom added later on

4.2: Attendance:

This feature is implemented using the program "detect.py" on running the program we get a window:



Here the first 2 buttons are used only once, and the rest are used on a daily basis

BUTTON 1: TRAIN SYSTEM WITH REGISTERED FACES

This button is used to generate the training data file from all the registered student faces in the file “dataset”

we have made use of:

```
cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
```

This is used to detect faces in the registration dataset images once faces have been identified

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
```

```
recognizer.train(faces, np.array(ids))
```

```
recognizer.write('trainer/trainer.yml')
```

We create and train a LBHP face recognizer which is trained with each face and links use ID to the face

This training data is stored in “trainer/trainer.yml” and will be used later on for face recognition

BUTTON 2:Create Attendance Sheet

This is a simple button the creates a simple attendance sheet in Excel called “attendance.xlsx”

This sheet is used to update daily columns and mark attendance

BUTTON 3:Please Enter date

This entry box is used to enter the day on which attendance is to be made,whatever is entered into the text box appears as a new column

Hence a teacher need only enter the Current date and it Creates a Attendance column for that day

BUTTON 4:Select An Image And Run Face Based Attendance

Using this button the following tasks are done:

- Selection on an image of the class
- Running Face Recognition
- Marking attendance in the new column of recognized faces

The teacher may upload a current image of a class to the system
the system detects all faces in image and compares it to its training data.

Using this training data in attempts to recognize all the faces in the image and marks attendance for all faces present in the image

Here we again make use of LBPH Face Recognizer to read file “trainer.yml” and make predictions of the ID of faces in the image

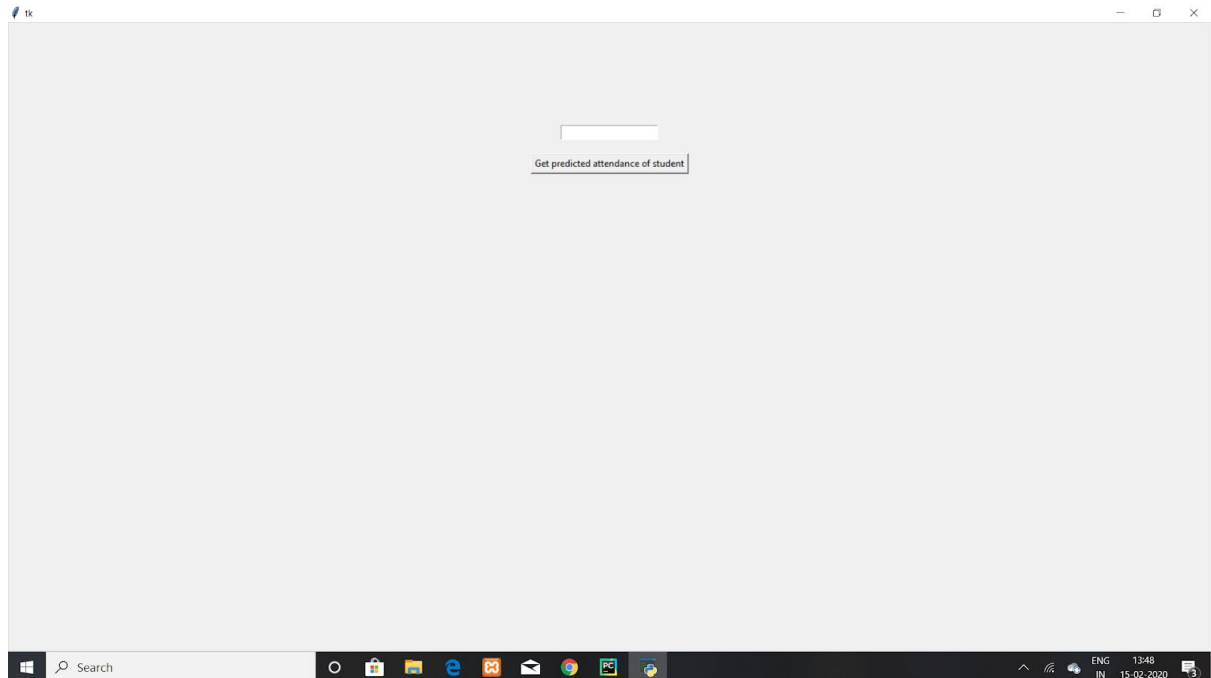
Here we make use of a variable “conf” which represents the confidence of system in the decision it made in recognizing the face and if “conf>75” it marks attendance for the given face.

This updated attendance is stored in file:

`"/facedetect/markedattendance.csv"`

BUTTON 5:Predict Student Attendance

This button runs the program 'predicty.py' which opens window:



Here we enter the student ID and it returns his predicted attendance

IF students predicted attendance is lesser than 75% it issues a warning to the student to attend more classes

We make use of a Linear Regressor to predict students future attendance

`"from sklearn.linear_model import LinearRegression"`

We train the Linear regressor with Student ID and Students current attendance percentage,using this data the linear regressor form a line of best fit and predicts students future attendance

Conclusion:

This system can be further developed with the help of IOT to send students and their parents messages regarding attendance details.

Using this system cuts shorts the time of taking attendance and allows the teacher to focus more on teaching

It can be further developed using ready made Api's such as Tensorflow to provide more accurate facial recognition

