Face Recognization Attendance system

Team Members:

Rohan Parsad Gupta	: 6522040549
Adnan Ali	: 6522040531
Kyaw Maung Maung Thwin	: 6522040648

Table of Contents

Introduction

This project is designed to automate the attendance taking process in a classroom setting. The system captures live video from a camera, detects faces, compares them to a database of known faces, and marks the attendance of recognized students. The attendance data is stored in a Firebase Realtime Database, and students' profile pictures are stored in Firebase Storage.

System Overview

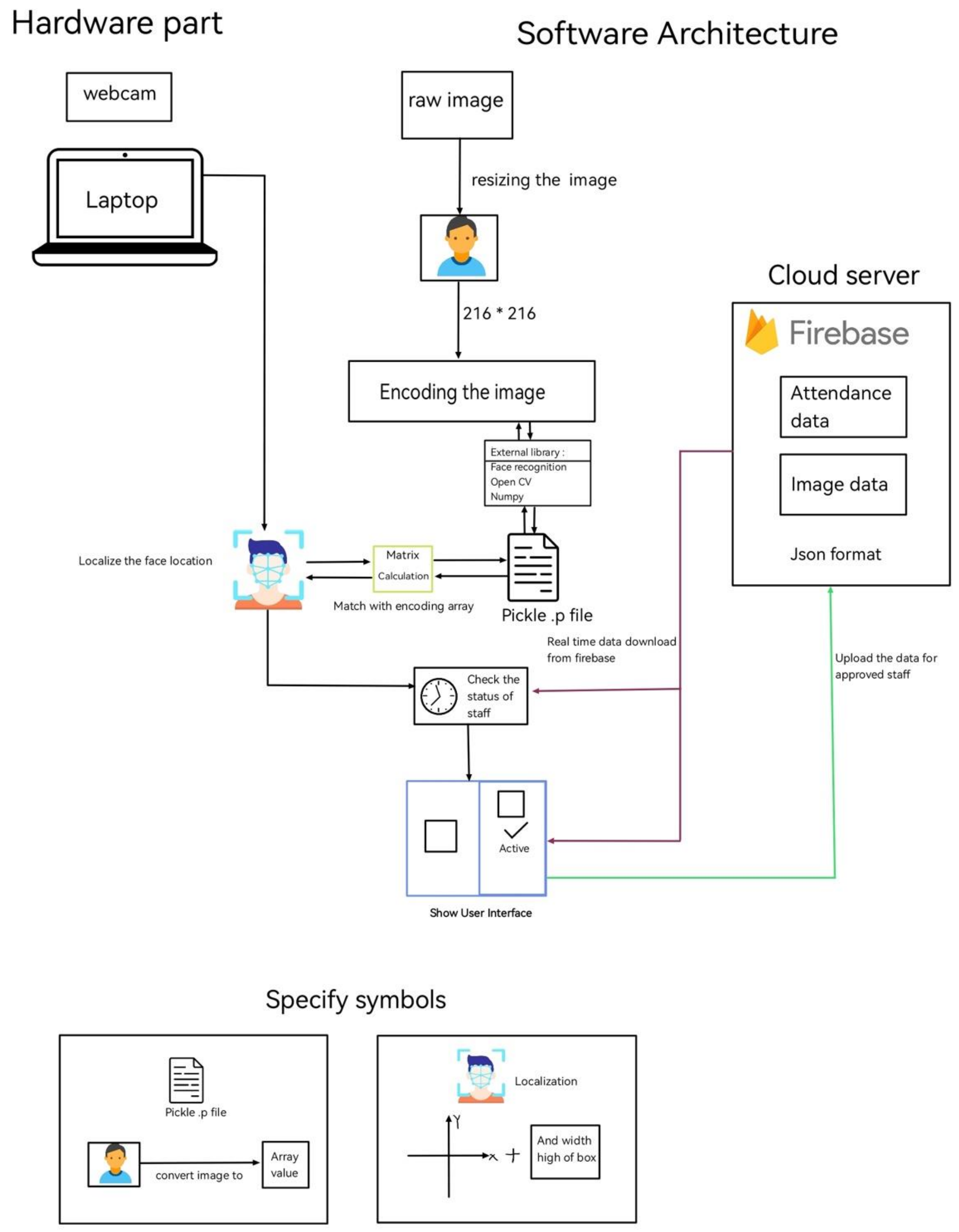


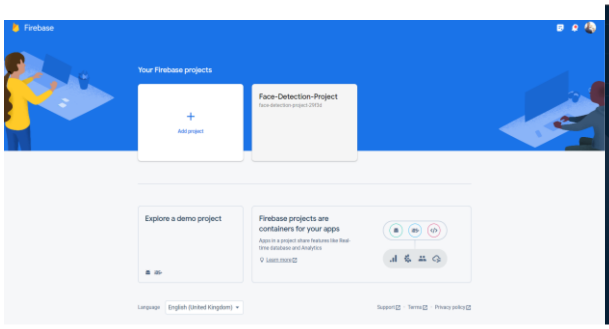
Fig (1) overview system Architecture

Project Dependencies

- **Os:** a module that provides a portable way to interact with the operating system.
- **Pickle:** A module used to convert Python objects into a stream of bytes, and vice versa.
- **Numpy:** A library for the Python programming language, adding support for large, multi-dimensional arrays and matrices.
- **cv2:** OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. This library provides a wide variety of algorithms, including face detection and recognition.
- **face_recognition:** a Python library for face recognition based on dlib library.
- **cvzone:** a Python library built on OpenCV to provide some extra features, such as adding text or images to a rectangle, drawing shapes, etc.
- **firebase_admin:** a Python package that provides an interface to the Firebase Realtime Database and Firebase Storage.

Firebase Configuration

The project requires authentication with Firebase, which is provided through a service account key file (serviceAccountKey.json). The Firebase configuration is initialized using the credentials.Certificate function, and the firebase_admin.initialize_app function.



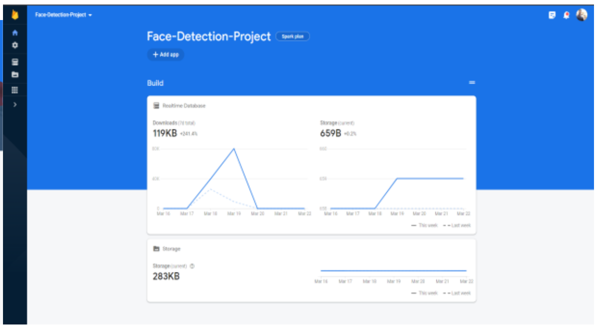
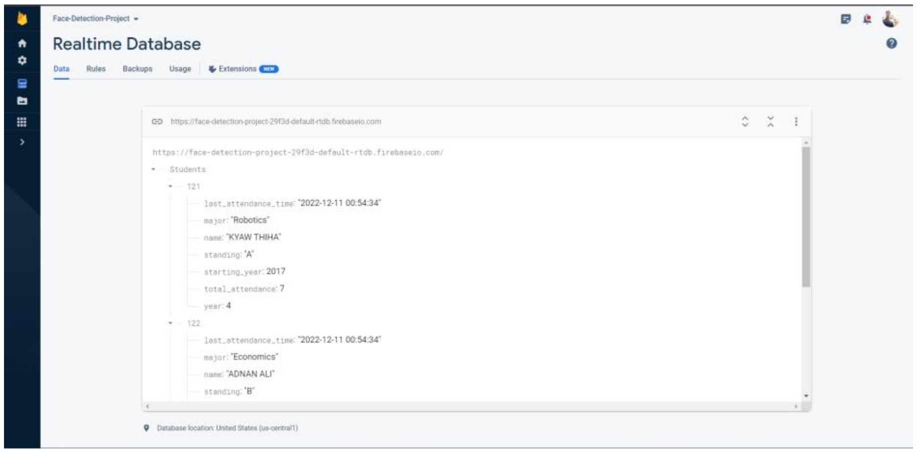


Fig (2) firebase project page

Fig (3) firebase Dashboard

- **Initializing the Firebase SDK:** The code initializes the Firebase SDK with the credentials provided in the "serviceAccountKey.json" file. The URL of the Firebase database is also specified.

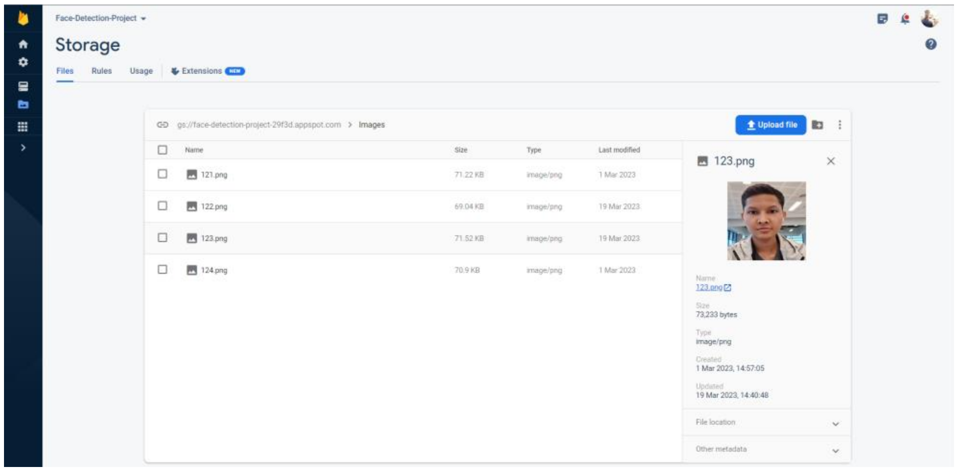
- **Defining student details:** The code defines a dictionary called "data" that contains the details of four students. The details include the student's name, major, starting year, total attendance, standing, year, and last attendance time.


Face-Detection-Project
Realtime Database
Data
Rules
Backups
Usage
Extensions
NEW
https://face-detection-project-29f3d-default-rtdb.firebaseio.com
Database location: United States (us-central1)

```
https://face-detection-project-29f3d-default-rtdb.firebaseio.com/
▾ Students
  ▾ 121
      last_attendance_time: "2022-12-11 00:54:34"
      major: "Robotics"
      name: "KYAW THIHA"
      standing: "A"
      starting_year: 2017
      total_attendance: 7
      year: 4
  ▾ 122
      last_attendance_time: "2022-12-11 00:54:34"
      major: "Economics"
      name: "ADNAN ALI"
      standing: "B"
```

Fig (4) Realtime student record database in firebase

- **Writing data to the database:** The code uses a for loop to iterate over the items in the "data" dictionary. For each key-value pair, it sets the corresponding value in the "Students" node of the database using the set function. The key represents the unique ID of each student, and the value is a dictionary that contains the student's details.



Name	Size	Type	Last modified
121.png	71.22 KB	image/png	1 Mar 2023
122.png	69.04 KB	image/png	19 Mar 2023
123.png	71.52 KB	image/png	19 Mar 2023
124.png	70.9 KB	image/png	1 Mar 2023

123.png

Name
123.png

Size
73,233 bytes

Type
image/png

Created
1 Mar 2023, 14:57:05

Updated
19 Mar 2023, 14:40:48

Fig (5) Person Image storage in firebase that showed after success face detection in first time

How it works:

The project uses a webcam to capture the video and detects faces using the face_recognition library. It then compares the detected faces with the known faces in the database. If there is a match, the program marks the attendance and displays the student's information on the screen. It also updates the attendance count in the Firebase database.

- **Video Capture:** The cv2.VideoCapture function is used to capture live video from a camera. The video capture object is then configured to set the resolution of the video.

- **Background Image:** The project has a background image (background.png) that is used to display different modes of operation, including a welcome screen, a loading screen, and a confirmation screen. The background image is loaded using the cv2.imread function.

- **Modes:** The system has three modes: welcome, loading, and confirmation. The modes are loaded from the Resources/Modes directory, and stored in a list of images (imgModeList).

- **Encoding:** The project encodes known faces using the face_recognition.face_encodings function. The encoding data is stored in a list (encodeListKnown), along with the corresponding student IDs (studentIds).

- **Face Detection:** The face_recognition.face_locations function is used to detect faces in the live video. If a face is detected, the system encodes the face using the face_recognition.face_encodings function, and compares it to the list of known face encodings.

- **Face Recognition:** The system compares the detected face encoding to the list of known face encodings using the face_recognition.compare_faces function. If a match is found, the system updates the background image with a green rectangle around the recognized face.

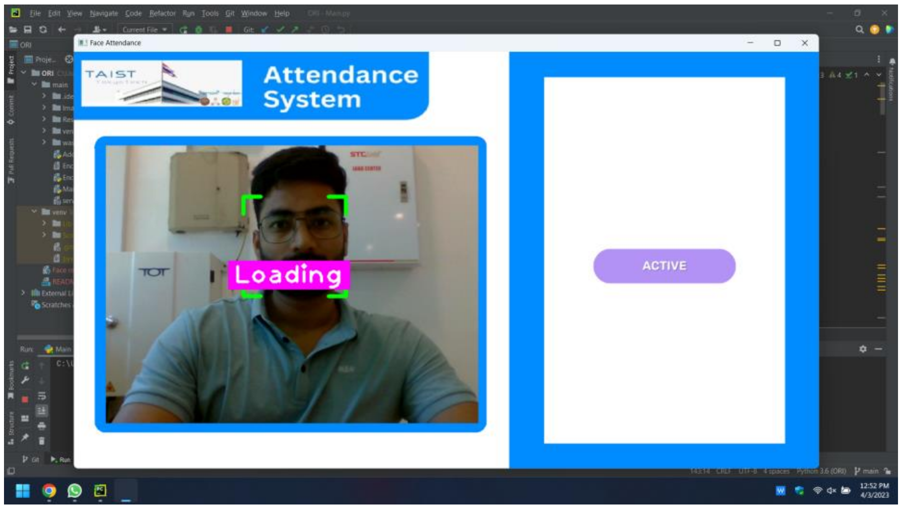
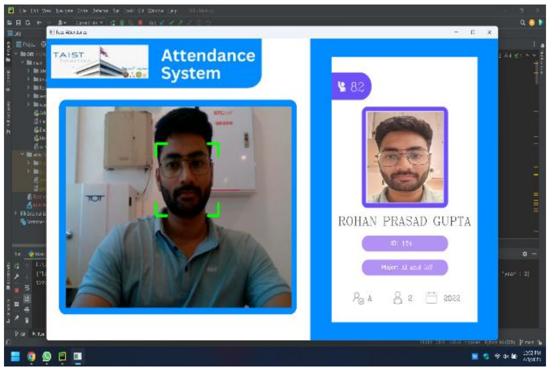


Fig (6) system demo detect face

- **Student Information:** The system retrieves student information from the Firebase Realtime Database using the db.reference function. The student's profile picture is retrieved from Firebase Storage using the bucket.get_blob function. The system also calculates the time elapsed since the student's last attendance and updates the attendance record in the Firebase Realtime Database.
-



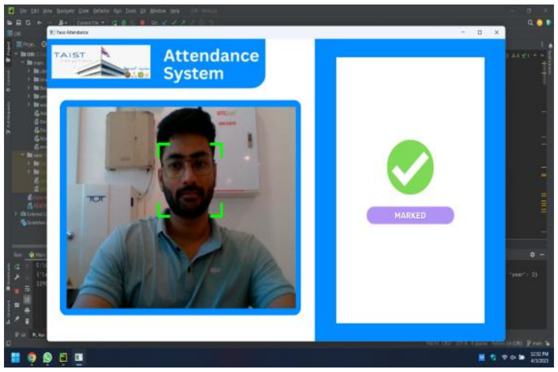


Fig (7) system demo recognized face Fig (8) attend data recorded successfully

- **User Interface:** The cvzone.putTextRect function is used to add text to the background image. The cv2.putText function is used to add text to the live video. The cvzone.cornerRect function is used to draw a rectangle around the recognized face.

IMAGE ENCODING

- **Importing student images:** The code gets a list of all the image files in the "Images" folder and reads each image using OpenCV's imread function. The student IDs are extracted from the file names using the splitext function. The images are then uploaded to Firebase storage.

- **Encoding the images:** The code defines a function called "findEncodings" that encodes the images using the face_recognition library. The images are first converted from BGR format to RGB format using OpenCV's cvtColor function. The face_encodings function of the face_recognition library is then used to encode the faces in the images.

- **Saving the encoded images:** The code calls the "findEncodings" function to encode the images and saves the encoded images along with their IDs to a file named "EncodeFile.p" using the pickle.dump function.

Conclusion

The Face Attendance proposed project is a working prototype for cutting edges attendance which more efficient and effective in classroom setting and less human interaction in recording. The system captures live video from a camera that installed in class entrance door or during lecture time, detects faces, compares them to a database of prerecorded faces if math face characteristic, and marks the attendance of recognized students.