Bangladesh University of Business and Technology

Face Recognition Attendance System

Project Report By

Name:	**Id:**
Fairuz Tasnim	21225103460
Md Ridwanur Rahman	21225103079
Md. Rakib Hassain	21225103076
Md. Emran Hossen	21225103064
Sajib Ahmed	21225103053

Supervised by

Md. Mahbub-Or-Rashid
Assistant Professor,
Department of CSE, BUBT
Rupnagar, Mirpur-2, Dhaka-1216

Submission Date:
Date: 03 December, 2023

ACKNOWLEDGEMENTS

We would like to take the opportunity to thank and express our deep sense of gratitude to our mentor and project guide **Md. Mahbub-Or-Rashid** for his immense support and valuable guidance without which it would not have been possible to reach at this stage of our minor project.
We also express our appreciation to the entire team involved in this project. Their dedication, hard work, and collaborative spirit were instrumental in overcoming challenges and achieving our goals. Each member's unique skills and commitment significantly contributed to the project's success.

ABSTRACT

With every passing day, we are becoming more and more dependent upon technology to carry out even the most basic of our actions. Facial detection and Facial recognition help us in many ways, be it sorting of photos in our mobile phone gallery by recognizing pictures with their face in them or unlocking a phone by a mere glance to adding biometric information in the form of face images in the country's unique ID database (Aadhaar) as an acceptable biometric input for verification.

This project lays out the basic terminology required to understand the implementation of Face Detection and Face Recognition using Intel's Computer Vision library called 'OpenCV'.

It also shows the practical implementation of the Face Detection and Face Recognition using OpenCV with Python embedding on both Windows as well as macOS platform. The aim of the project is to implement Facial Recognition on faces that the script can be trained for. The input is taken from a webcam and the recognized faces are displayed along with their name in real time. This project can be implemented on a larger scale to develop a biometric attendance system which can save the time-consuming process of manual attendance system.

CERTIFICATE OF ACCOMPLISHMENT

This is to certify that,

Fairuz Tasnim	21225103460
Md Ridwanur Rahman	21225103079
Md. Rakib Hassain	21225103076
Md. Emran Hossen	21225103064
Sajib Ahmed	21225103053

Have successfully completed the project report titled

"Face Recognition Attendance System"

This project report showcases an in-depth exploration and analysis of the implementation of face recognition technology integrated with a real-time database. The comprehensive report demonstrates a profound understanding of facial recognition algorithms, database management, and their cohesive integration for real-time applications.

The project report encompasses:

- Detailed analysis of face recognition algorithms utilised
- Methodical development and implementation strategies
- Integration techniques with a real-time database system
- Thorough testing, evaluation, and validation processes

The innovative application of technology, meticulous documentation, and the systematic approach to problem-solving displayed in this report are commendable.

This certificate is awarded in recognition of the diligent effort, technical proficiency, and successful completion of the project report, which contributes significantly to the field of Machine learning, artificial intelligence, etc.

Md. Mahbub-Or-Rashid
Supervisor
Assistant Professor, Department of CSE, BUBT

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

A face recognition system could also be a technology which is very capable of matching a personality's face from a digital image or a video frame which it has or use it as a reference to map and identify against an info of faces. Researchers area unit presently developing multiple ways throughout that face recognition systems work. the foremost advanced face recognition methodology, that is to boot used to manifest users through ID verification services, works by pinpointing and mensuration countenance from a given image.

While at first a kind of laptop application, face recognition systems have seen wider uses in recent times on smartphones and in alternative kinds of technol ogy, like artificial intelligence. as a result of computerised face recognition involves the measuring of a human's physiological characteristics face recognition systems area unit classified as bioscience. though the accuracy of face recognition systems as a biometric technology is a smaller amount than iris recognition and fingerprint recognition, it's wide adopted because of its contactless and non-invasive method.Facial recognition systems area unit deployed in advanced human -computer interaction, video police work and automatic compartmentalisation of pictures. We have a created a face recognition technology capable of identifying faces.

1.2 PROBLEM STATEMENT

"**Face Recognition Attendance System"** using Intel's open source Computer Vision Library (OpenCV) and Python dependency"

There are various scripts illustrated throughout the project that will have functionalities like detecting faces in static images, detecting faces in live feed using a webcam, capturing face images and storing them in the dataset, training of classifier for recognition and finally recognition of the trained faces.

All the scripts are written in python 3.6.8 and have been provided with documented code. This project lays out most of the useful tools and information for face detect ion 2 and face recognition and can be of importance to people exploring facial recognition with OpenCV. The project shows implementation of various algorithms and recognition approaches which will be discussed on later in the project report. Face Recognition can be of importance in terms of security, organization, marketing, surveillance and robotics etc. Face detection is able to very immensely improve surveillance efforts which can greatly help in tracking down of people with ill criminal record basically referring to criminals and terrorists who might be a vast threat to the security of the nation and the people collectively.The Personal security is also greatly exacerbated since there is nothing for hackers to steal or change, such as passwords.

1.3 BACKGROUND

Traditional attendance systems often rely on manual methods such as paper-based registers or card-swiping systems. These methods can be time-consuming, prone to errors, and lack the ability to adapt to changing environments. The Face Recognition Attendance System addresses these issues by leveraging advanced technologies to streamline the attendance tracking process.

1.4 OBJECTIVES

- Automate the attendance recording process.
- Improve accuracy and efficiency in attendance tracking.
- Provide a secure and reliable system for identity verification.
- Create a user-friendly interface for system administrators.

1.5 METHODOLOGY

The Project utilizes various libraries of Python such as

1.5.1 OpenCV

“OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library”. The main purpose of this was to provide a common infrastructure for computer vision applications and it was also built specifically for such purposes not to mention it also accelerated the use of machine perception inside the business product. “Being a BSD-licensed product, OpenCV makes it straightforward for businesses to utilize and modify the code”.

In total we can say that The library has about 2500 optimized algorithms which is really insane, “These algorihms contain a comprehensive set which comprises of each classic and progressive laptop vision and machine learning algorithms. These algorithms area unit usually accustomed sight and acknowledge faces, determine objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, manufacture 3D purpose clouds from s tereo cameras, sew pictures along to produce a high resolution image of a full scene, realize similar pictures from an image info, take away red eyes from pictures taken exploitation flash, follow eye movements, acknowledge scenery and establish markers to overlay it with increased reality, etc”. The amazing thing about this library is that it has quite about forty seven thousand individuals of user community and calculable variety of downloads olympian eighteen million. The library is utilized extensively in corporations, analysis teams and by governmental bodies.

Along with well-established corporations like “Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota” that use the library, there area unit several startups like “Applied Minds, VideoSurf, and Zeitera”, that create in depth use of OpenCV. OpenCV’s deployed wide array spans vary from sewing streetview pictures along, police work intrusions in police work video in Israel, watching mine instrumentality in China, serving to robots navigate and devour objects at “Willow Garage, detection of natatorium drowning accidents in Europe, running interactive art in Espana and New York , checking runways for scrap in Turkey”, inspecting labels on product in factories around the world on to fast face detection in Japan.

1.5.2 NumPy

“The Python programming language earlier wasn't originally designed for numerical computing as we know it to be , however it also attracted the attention of the scientific and engineering community early” . “In 1995 the interest (SIG) matrix-sig was based with the aim of shaping associate array computing package; among its members was Python designer and supporter Guido van Rossum, WHO extended Python's syntax (in explicit the compartmentalization syntax) to make array computing easier”. “An implementation of a matrix package was completed by Jim discoverer, then generalized[further rationalization required by Jim Hugunin and known as Numeric (also diversely observed because the "Numerical Python extensions" or "NumPy").Hugunin, a collegian at the Massachusetts Institute of Technology (MIT), joined the Corporation for National analysis Initiatives (CNRI) in 1997 to work on JPython,leaving Paul Dubois of Lawrence Livermore National Laboratory (LLNL) to need over as supporter.Other early contributors embrace David Ascher, Konrad Hinsen and Travis Oliphant”.

“A new package known as Numarray was written as a additional versatile replacement for Numeric.Like Numeric, it too is currently deprecated.Numarray had quicker operations for large arrays, however was slower than Numeric on tiny ones, thus for a time each packages were utilised in parallel for varied use cases. The last version of Numeric (v24.2) was discharged on St Martin's Day 2005, whereas the last version of numarray (v1.5.2) was discharged on twenty four August 2006”.

There was a want to urge Numeric into the Python customary library, however Guido van Rossum determined that the code wasn't reparable in its state then. “In early 2005, NumPy developer Travis Oliphant needed to unify the community around one array package and ported Numarray's options to Numeric, cathartic the result as NumPy one.0 in 2006.This new project was a region of SciPy. To avoid putting in the large SciPy package simply to urge associate array object, t his new package was separated and known as NumPy. Support for Python three was other in 2011 with NumPy version one.5.0”.

In 2011, PyPy started development on associate implementation of the NumPy API for PyPy.It is not nevertheless absolutely compatible with NumPy.

1.5.3 Pillow

The Python Imaging Library (PILLOW) or generally called as PIL is very useful for adding image processing capabilitiy to your Python interpreter which you have or have installed when working with Pyhon.

The main use of this library is that this library has a wide array of extensive file format support which allows it to run and store different files in different formats, an efficient representation which is further boosted by very powerful image process ing ability.The important point is that the core image library is meant for faster access to data stored during a few basic pixel formats.All this enable it to be a very commanding and powerful tool for image processing general, which is probably also the reason why it is used so heavily.

Let’s see a couple of possible things in which we can use this library. Image Archive-The “Python Imaging Library” is right for image archival and execution applications. you'll use the library to make thumbnails, convert between file formats, print images, etc.

From the information available to us currently about the current version is that it identifies and reads an outsized number of file formats. Write support is intentionally restricted to the foremost commonly used interchange and presentation formats.

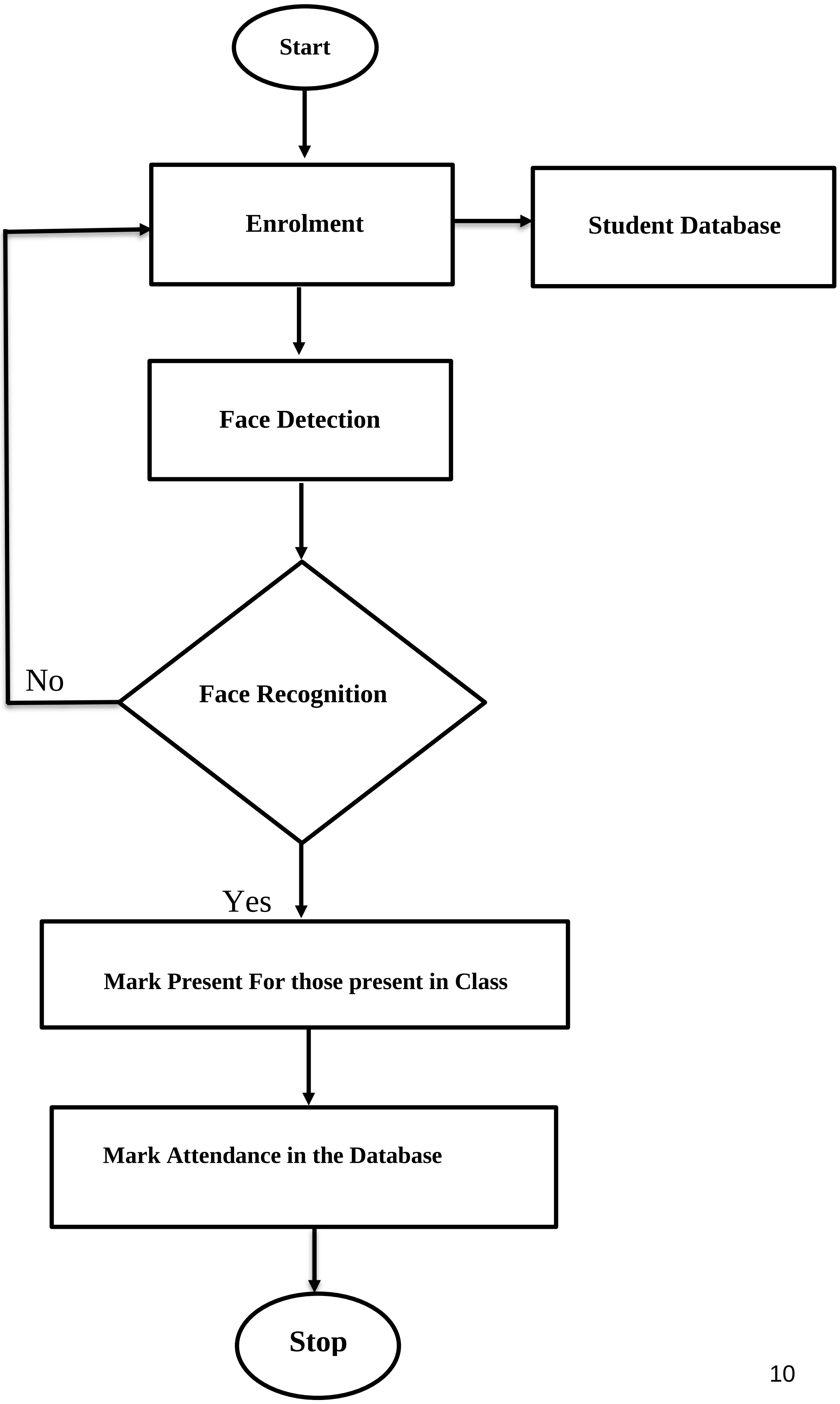
“Image Display-The current release includes Tk PhotoImage and BitmapImage interfaces, also as a Windows DIB interface which will be used with PythonWin and other Windows-based toolkits. Many other GUI toolkits accompany some quite PIL support”.“For a purpose such as debugging there are a lot of functions like the show() method which saves a picture to disk, and callsit an external display utility. Image Processing-The library contains basic image processing functionality, including point operations, filtering with a group of built-in convolution kernels, and colour space conversions”.Basically in short the “Python Imaging library” is a free additional open source library which you have to install first by using the command such as pip so that you can use it to run various python functions in the module.In regards to the main function of this library is to add functionality in opening, manipulating and storing of different file format images which can later be used to perform operations on accordingly to the needs of the programmer using the library.

1.5.4 FACE RECOGNITION

The “Face recognition” library in python is a library which helps in recognizing and manipulating the faces by using the programming language python or from the command line with the simplest face recognition library after importing the module and accessing the required functions.The “Face recognition” library was built using dlib’s “state-of-the-art face recognition” and was further enhanced and built with deep learning. The model has an accuracy of 99.38%.

CHAPTER 2
LITERATURE SURVEY

2.1. Flow Chart



Start
Enrolment
Student Database
Face Detection
Face Recognition
No
Yes
Mark Present For those present in Class
Mark Attendance in the Database
Stop

2.2 Face Tracking

Face tracking refers to identifying the features which are then used to detect a Face In this case the example method includes the receiving or we can say that it gets the first image and the second images of a face of a user who is being taken into consideration, where one or both of the images which were used to sort of look for a match have been granted a match by the facial recognition system which also proofs the correct working of the system. “The technique includes taking out a second subimage coming from the second image, where the second sub-image includes a representation of the at least one corresponding facial landmark, detecting a facial gesture by determining whether a sufficient difference exists between the second sub - image and first sub-image to indicate the facial gesture, and determining, based on detecting the facial gesture, whether to deny authentication to the user with respect to accessing functionalities controlled by the computing”

2.3 Mechanisms of human facial recognition

Basically what we see in this paper is that it presents an extension and a new way of perception of the author's theory for human visual information processing, which The method includes extracting a second sub-image from the second image, where the second sub-image includes a representation of the at least one corresponding facial landmark. “In turn detecting a facial gesture by determining whether a sufficient difference exists between the second sub-image and first sub-image to indicate the facial gesture, and determining, based on detecting the facial gesture, whether to deny authentication to the user with respect to the human recognition system and same was applied”.Several indispensable techniques are implicated: encoding of visible photographs into neural patterns, detection of easy facial features, measurement standardization, discount of the neural patterns in dimensionality .

“The logical (computational) role suggested for the primary visual cortex has several components: size standardization, size reduction, and object extraction”. “The result of processing by the primary visual cortex, it is suggested, is a neural encoding of the 12 visual pattern at a size suitable for storage. “(In this context, object extraction is the isolation of regions in the visual field having the same color, texture, or spatial extent.)”It is shown in detail how the topology of the mapping from retina to cortex, the connections between retina, lateral geniculate bodies and primary visual cortex, and the local structure of the cortex itself may combine to encode the visual patterns. Aspects of this theory are illustrated graphically with human faces as the primary stimulus. However, the theory is not limited to facial recognition but pertains to Gestalt recognition of any class of familiar objects or scenes.

2.4 Eye Spacing Measurement for Facial Recognition

Few procedures to computerized facial consciousness have employed geometric size of attribute points of a human face. Eye spacing dimension has been recognized as an essential step in reaching this goal. Measurement of spacing has been made by means of software of the Hough radically change method to discover the occasion of a round form and of an ellipsoidal form which approximate the perimeter of the iris and each the perimeter of the sclera and the form of the place under the eyebrows respectively. Both gradient magnitude and gradient direction were used to handle the noise contaminating the feature space. “Results of this application indicate that measurement of the spacing by detection of the iris is the most accurate of these three methods with measurement by detection of the position of the eyebrows the least accurate. However, measurement by detection of the eyebrows' position is the least constrained method. Application of these strategies has led to size of a attribute function of the human face with adequate accuracy to advantage later inclusion in a full bundle for computerized facial consciousness”.

CHAPTER 3
TRAINING AND TESTING

3.1 TRAINING IN OPENCV

In OpenCV, training refers to providing a recognizer algorithm with training data to learn from. The trainer uses the same algorithm (LBPH) to convert the images cells to histograms and then computes the values of all cells and by concatenating the histograms, feature vectors can be obtained. Images can be classified by processing with an ID attached. Input images are classified using the same process and compared with the dataset and distance is obtained. By setting up a threshold, it can be identified if it is a known or unknown face.Eigenface and Fisherface compute the dominant features of the whole training set while LBPH analyses them individually.

To do so, firstly, a Dataset is created. You can either create your own dataset or start with one of the available face databases.

- Yale Face Database
- AT & T Face Database

The .xml or .yml configuration file is made from the several features extracted from your dataset with the help of the FaceRecognizer Class and stored in the form of feature vectors.

3.2 TESTING RESULT

The system underwent rigorous testing to ensure:

- Accurate face detection and recognition in various conditions.
- Reliable attendance logging and data storage.
- User interface functionality and ease of use.

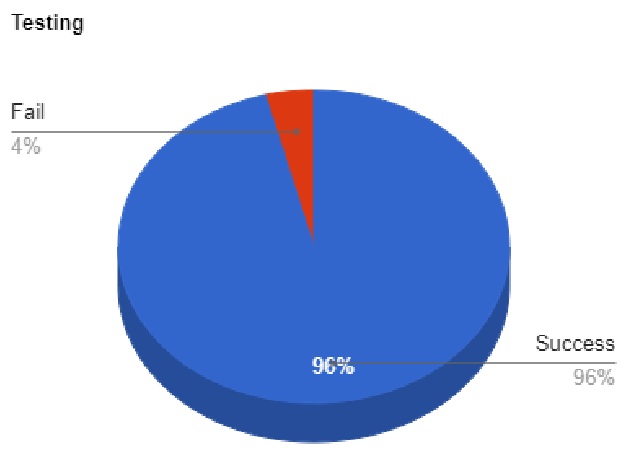
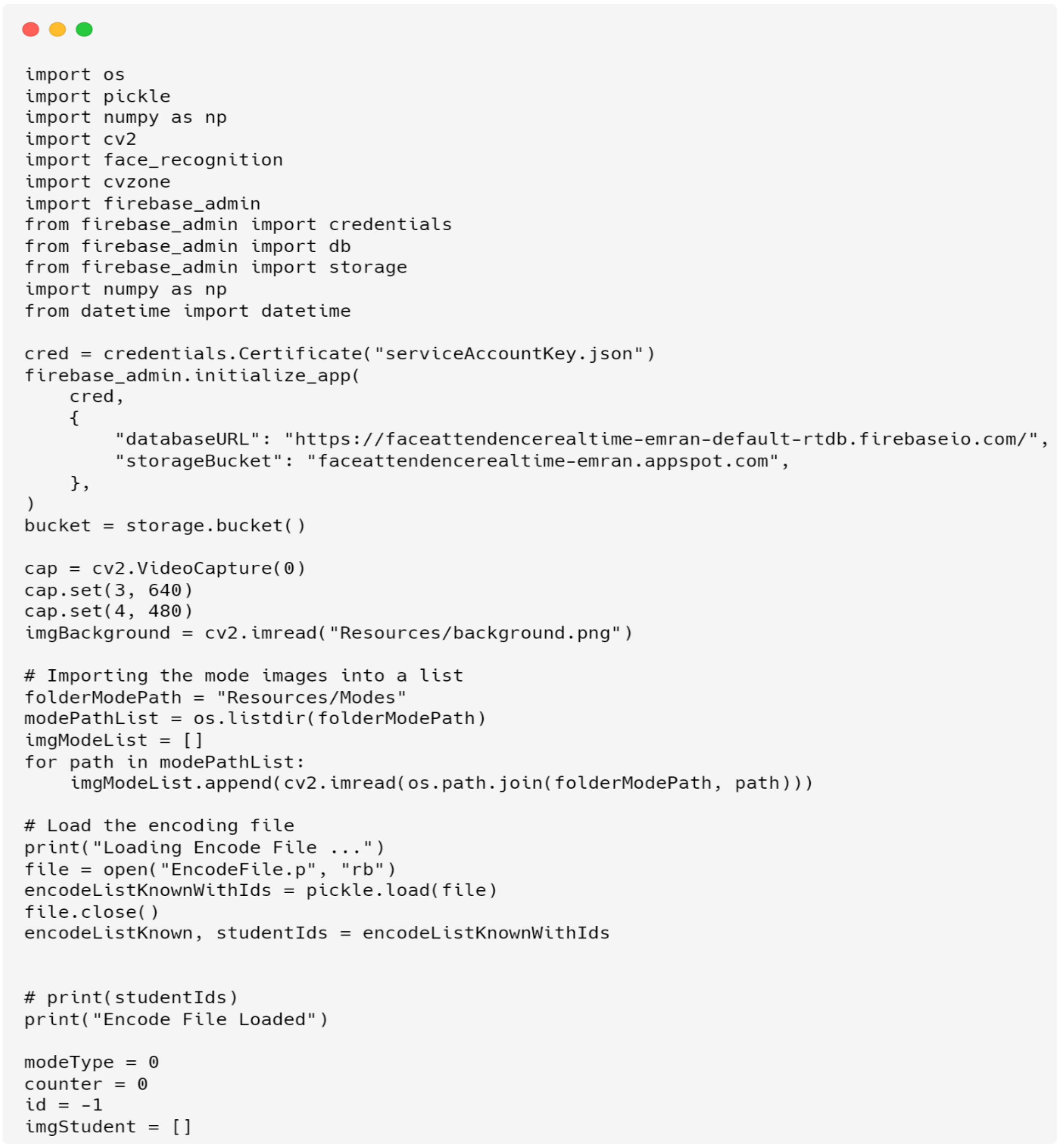


Figure No. 3.0

3.3Experimental Code

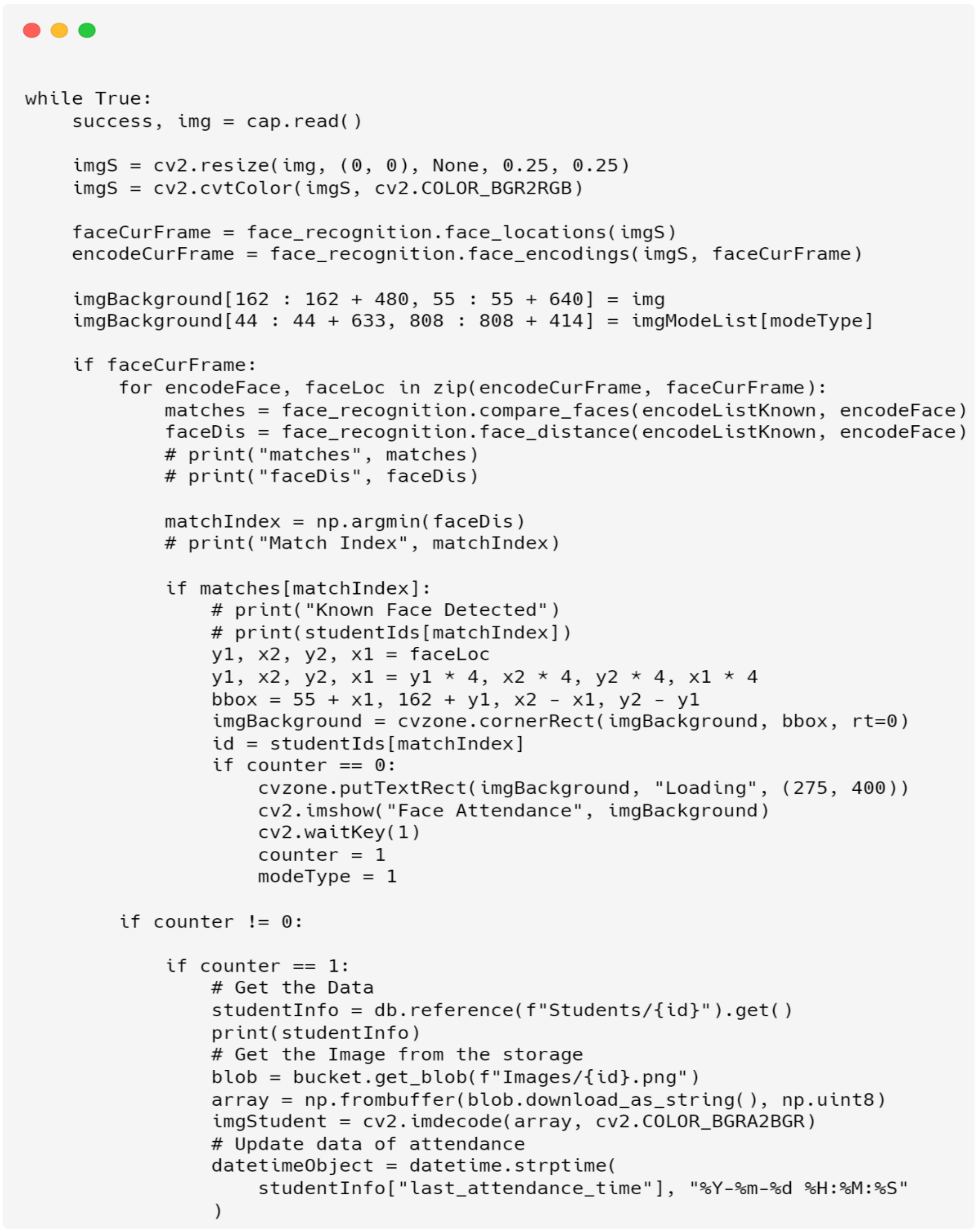
This code appears to be a Python script that utilizes various libraries like OpenCV, Face Recognition, and Firebase to perform real-time face attendance recognition and management.

main.py



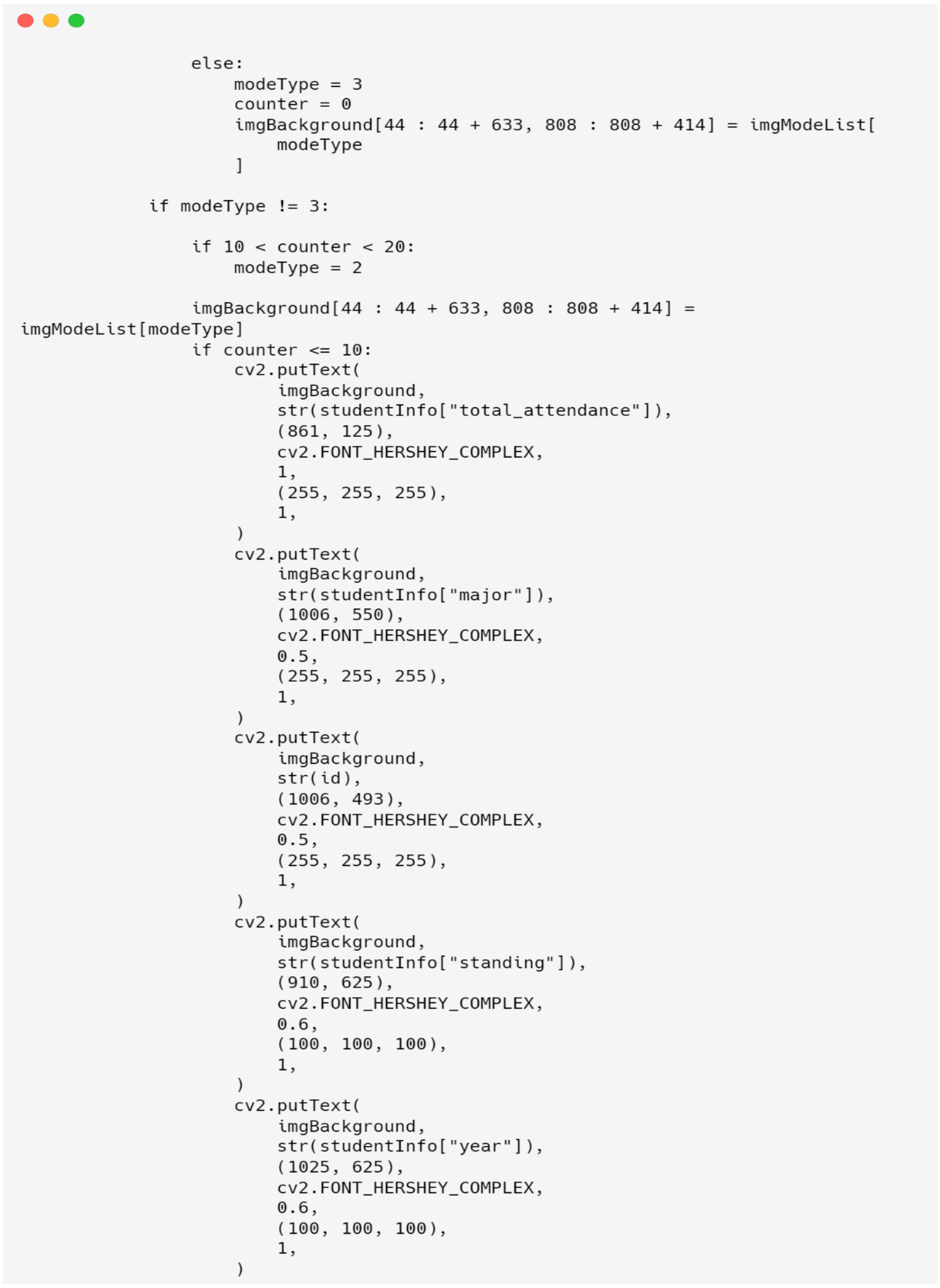
```python
import os
import pickle
import numpy as np
import cv2
import face_recognition
import cvzone
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from firebase_admin import storage
import numpy as np
from datetime import datetime

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(
    cred,
    {
        "databaseURL": "https://faceattendencerealtime-emran-default-rtdb.firebaseio.com/",
        "storageBucket": "faceattendencerealtime-emran.appspot.com",
    },
)
bucket = storage.bucket()

cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)
imgBackground = cv2.imread("Resources/background.png")

# Importing the mode images into a list
folderModePath = "Resources/Modes"
modePathList = os.listdir(folderModePath)
imgModeList = []
for path in modePathList:
    imgModeList.append(cv2.imread(os.path.join(folderModePath, path)))

# Load the encoding file
print("Loading Encode File ...")
file = open("EncodeFile.p", "rb")
encodeListKnownWithIds = pickle.load(file)
file.close()
encodeListKnown, studentIds = encodeListKnownWithIds


# print(studentIds)
print("Encode File Loaded")

modeType = 0
counter = 0
id = -1
imgStudent = []
```

Figure No. 3.1



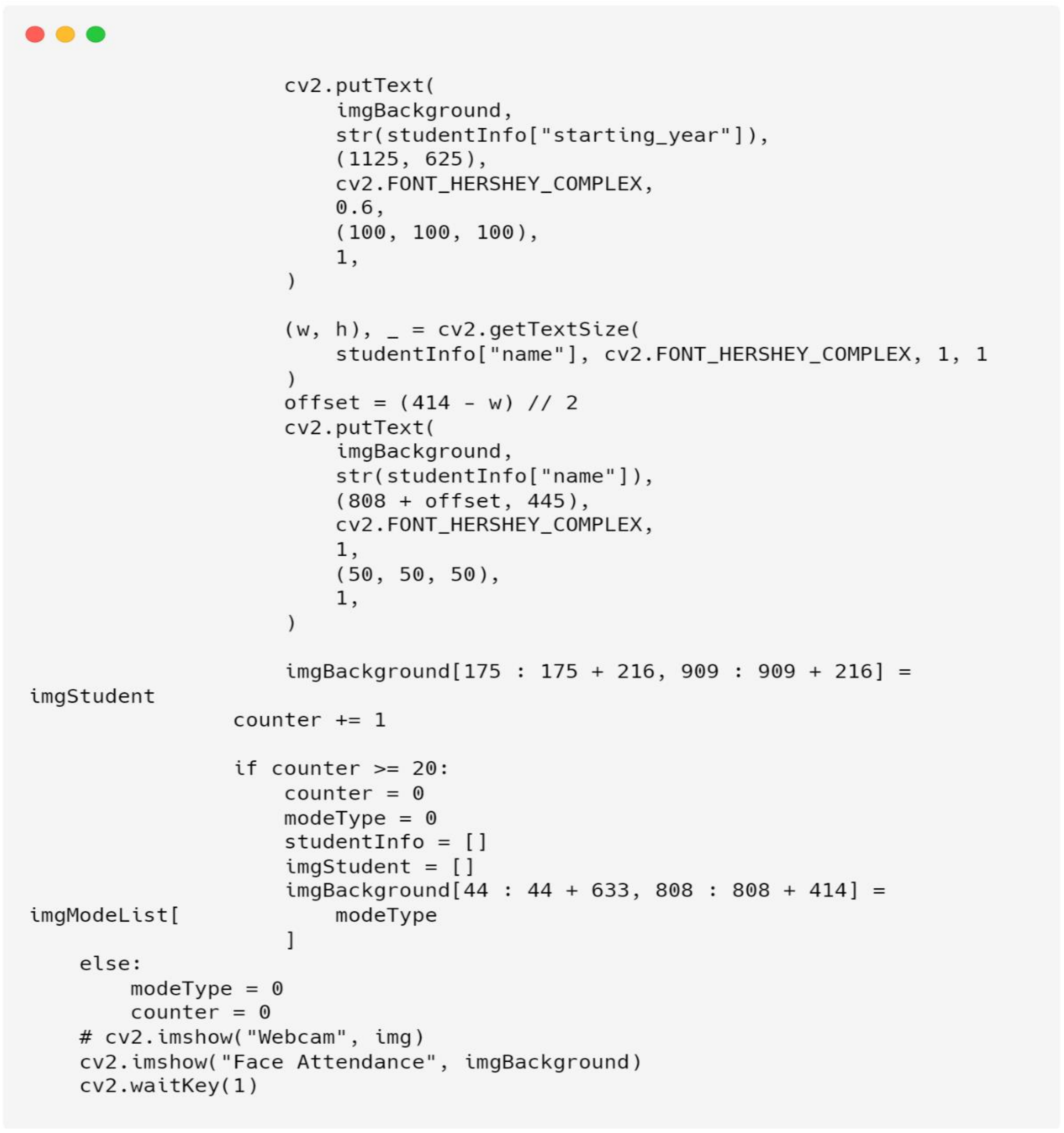
```python
while True:
    success, img = cap.read()

    imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

    faceCurFrame = face_recognition.face_locations(imgS)
    encodeCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)

    imgBackground[162 : 162 + 480, 55 : 55 + 640] = img
    imgBackground[44 : 44 + 633, 808 : 808 + 414] = imgModeList[modeType]

    if faceCurFrame:
        for encodeFace, faceLoc in zip(encodeCurFrame, faceCurFrame):
            matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
            faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
            # print("matches", matches)
            # print("faceDis", faceDis)

            matchIndex = np.argmin(faceDis)
            # print("Match Index", matchIndex)

            if matches[matchIndex]:
                # print("Known Face Detected")
                # print(studentIds[matchIndex])
                y1, x2, y2, x1 = faceLoc
                y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
                bbox = 55 + x1, 162 + y1, x2 - x1, y2 - y1
                imgBackground = cvzone.cornerRect(imgBackground, bbox, rt=0)
                id = studentIds[matchIndex]
                if counter == 0:
                    cvzone.putTextRect(imgBackground, "Loading", (275, 400))
                    cv2.imshow("Face Attendance", imgBackground)
                    cv2.waitKey(1)
                    counter = 1
                    modeType = 1

        if counter != 0:

            if counter == 1:
                # Get the Data
                studentInfo = db.reference(f"Students/{id}").get()
                print(studentInfo)
                # Get the Image from the storage
                blob = bucket.get_blob(f"Images/{id}.png")
                array = np.frombuffer(blob.download_as_string(), np.uint8)
                imgStudent = cv2.imdecode(array, cv2.COLOR_BGRA2BGR)
                # Update data of attendance
                datetimeObject = datetime.strptime(
                    studentInfo["last_attendance_time"], "%Y-%m-%d %H:%M:%S"
                )
```

Figure No. 3.2



```
                else:
                    modeType = 3
                    counter = 0
                    imgBackground[44 : 44 + 633, 808 : 808 + 414] = imgModeList[
                        modeType
                    ]

            if modeType != 3:

                if 10 < counter < 20:
                    modeType = 2

                imgBackground[44 : 44 + 633, 808 : 808 + 414] =
imgModeList[modeType]
                if counter <= 10:
                    cv2.putText(
                        imgBackground,
                        str(studentInfo["total_attendance"]),
                        (861, 125),
                        cv2.FONT_HERSHEY_COMPLEX,
                        1,
                        (255, 255, 255),
                        1,
                    )
                    cv2.putText(
                        imgBackground,
                        str(studentInfo["major"]),
                        (1006, 550),
                        cv2.FONT_HERSHEY_COMPLEX,
                        0.5,
                        (255, 255, 255),
                        1,
                    )
                    cv2.putText(
                        imgBackground,
                        str(id),
                        (1006, 493),
                        cv2.FONT_HERSHEY_COMPLEX,
                        0.5,
                        (255, 255, 255),
                        1,
                    )
                    cv2.putText(
                        imgBackground,
                        str(studentInfo["standing"]),
                        (910, 625),
                        cv2.FONT_HERSHEY_COMPLEX,
                        0.6,
                        (100, 100, 100),
                        1,
                    )
                    cv2.putText(
                        imgBackground,
                        str(studentInfo["year"]),
                        (1025, 625),
                        cv2.FONT_HERSHEY_COMPLEX,
                        0.6,
                        (100, 100, 100),
                        1,
                    )
```

Figure No. 3.3



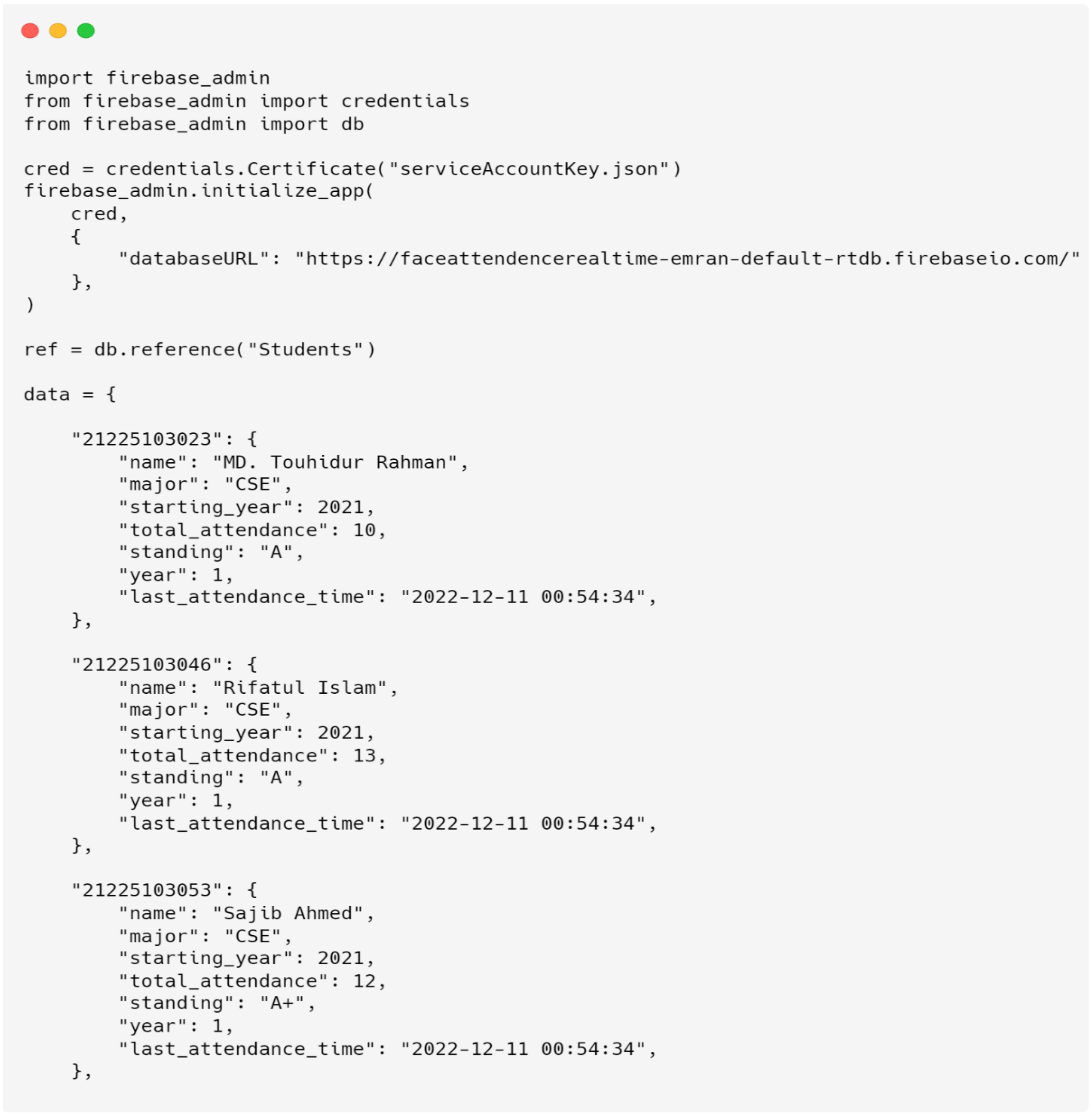
```
                    cv2.putText(
                        imgBackground,
                        str(studentInfo["starting_year"]),
                        (1125, 625),
                        cv2.FONT_HERSHEY_COMPLEX,
                        0.6,
                        (100, 100, 100),
                        1,
                    )

                    (w, h), _ = cv2.getTextSize(
                        studentInfo["name"], cv2.FONT_HERSHEY_COMPLEX, 1, 1
                    )
                    offset = (414 - w) // 2
                    cv2.putText(
                        imgBackground,
                        str(studentInfo["name"]),
                        (808 + offset, 445),
                        cv2.FONT_HERSHEY_COMPLEX,
                        1,
                        (50, 50, 50),
                        1,
                    )

                    imgBackground[175 : 175 + 216, 909 : 909 + 216] =
imgStudent
                counter += 1

                if counter >= 20:
                    counter = 0
                    modeType = 0
                    studentInfo = []
                    imgStudent = []
                    imgBackground[44 : 44 + 633, 808 : 808 + 414] =
imgModeList[            modeType
                    ]
    else:
        modeType = 0
        counter = 0
    # cv2.imshow("Webcam", img)
    cv2.imshow("Face Attendance", imgBackground)
    cv2.waitKey(1)
```

Figure No. 3.4

AddDatatoDatabase.py

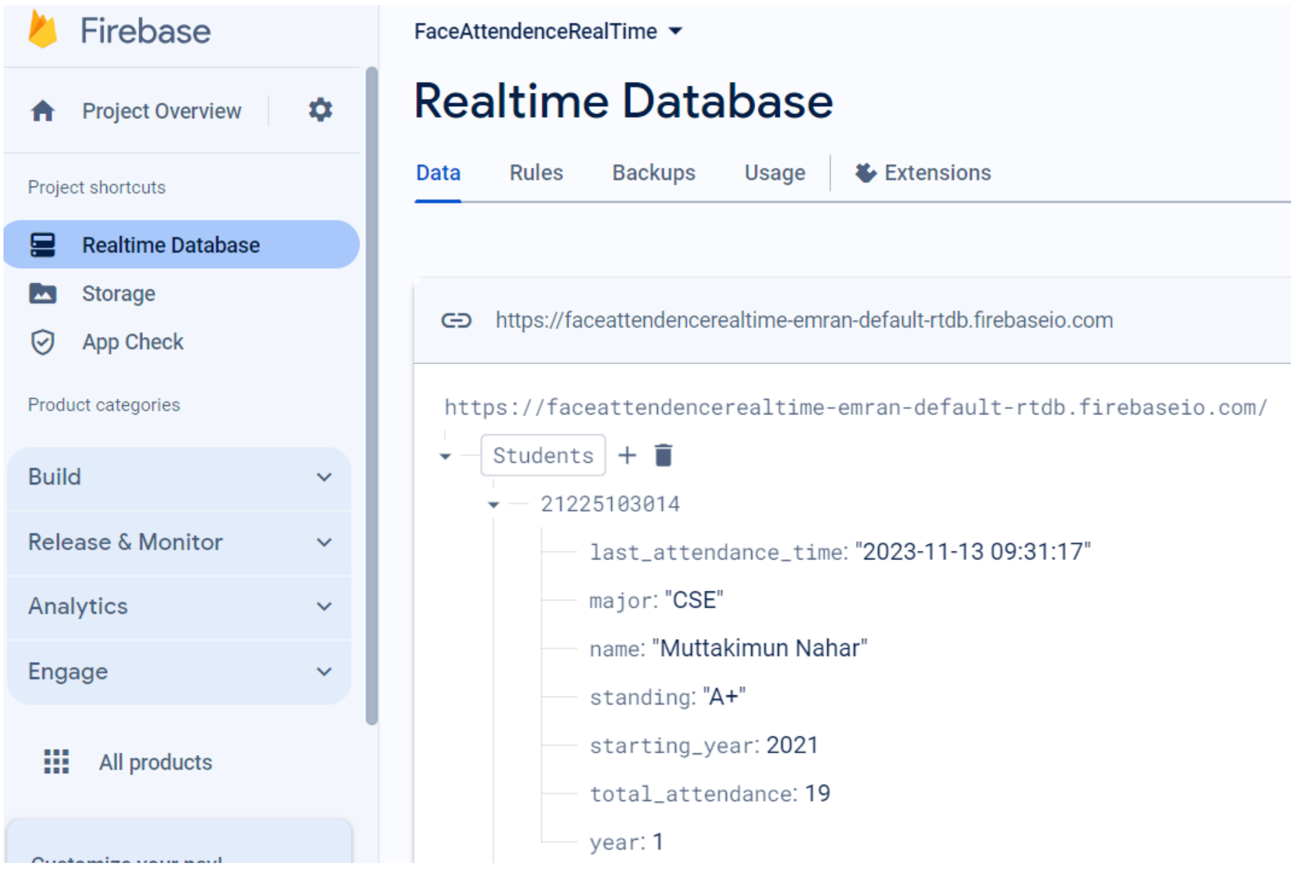
Certainly! This Python script is utilizing the Firebase Admin SDK to interact with a Firebase Realtime Database. Let's break down the script's functionality

This script would be used as a one-time execution to initialize or update student information in the Firebase database. Each time it runs, it writes or updates the student information provided in the data dictionary to the specified database location.



```python
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(
    cred,
    {
        "databaseURL": "https://faceattendencerealtime-emran-default-rtdb.firebaseio.com/"
    },
)

ref = db.reference("Students")

data = {

    "21225103023": {
        "name": "MD. Touhidur Rahman",
        "major": "CSE",
        "starting_year": 2021,
        "total_attendance": 10,
        "standing": "A",
        "year": 1,
        "last_attendance_time": "2022-12-11 00:54:34",
    },

    "21225103046": {
        "name": "Rifatul Islam",
        "major": "CSE",
        "starting_year": 2021,
        "total_attendance": 13,
        "standing": "A",
        "year": 1,
        "last_attendance_time": "2022-12-11 00:54:34",
    },

    "21225103053": {
        "name": "Sajib Ahmed",
        "major": "CSE",
        "starting_year": 2021,
        "total_attendance": 12,
        "standing": "A+",
        "year": 1,
        "last_attendance_time": "2022-12-11 00:54:34",
    },
```

Figure No. 3.5



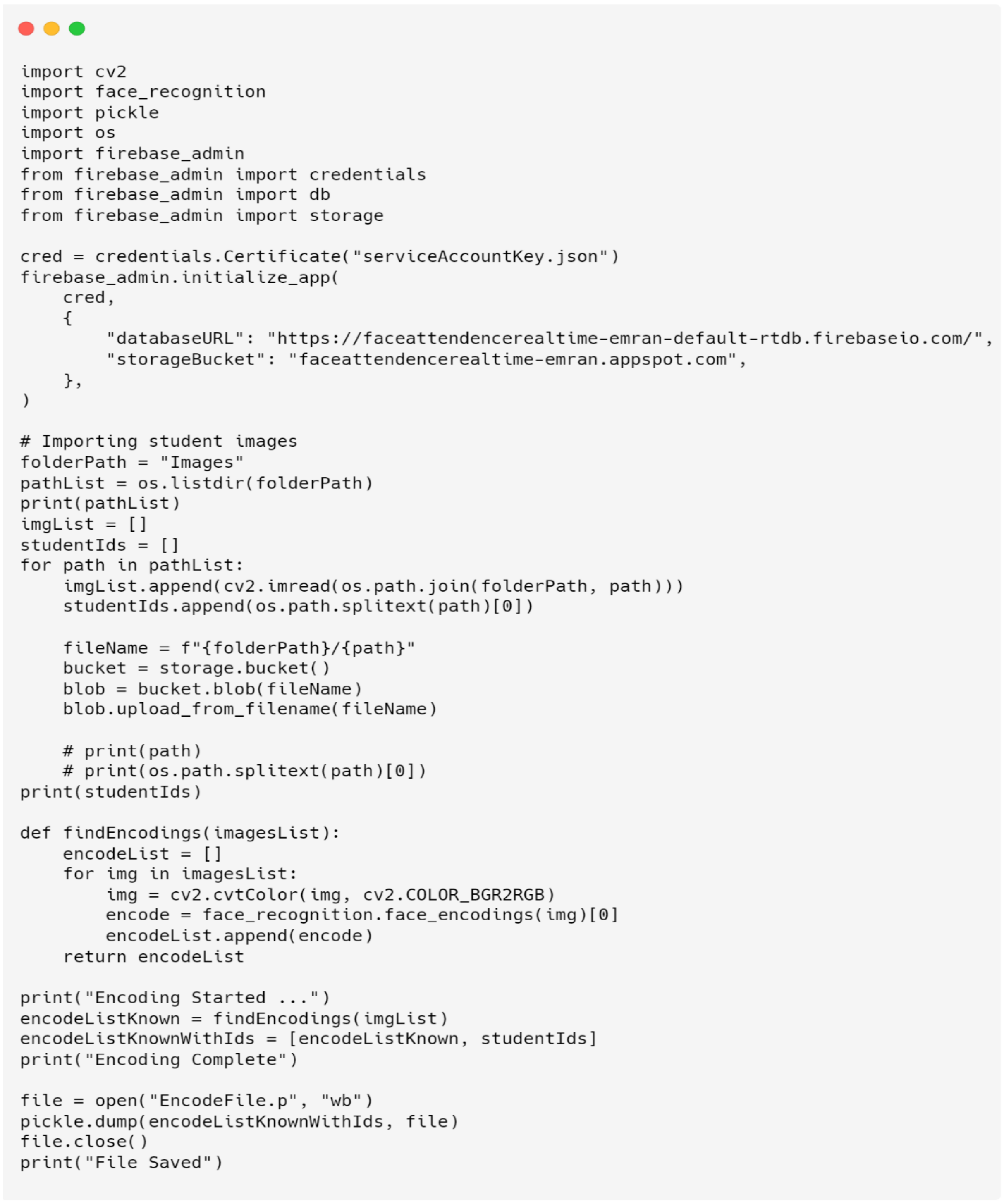
https://faceattendencerealtime-emran-default-rtdb.firebaseio.com

```
https://faceattendencerealtime-emran-default-rtdb.firebaseio.com/
  Students
    21225103014
      last_attendance_time: "2023-11-13 09:31:17"
      major: "CSE"
      name: "Muttakimun Nahar"
      standing: "A+"
      starting_year: 2021
      total_attendance: 19
      year: 1
```

Figure No. 3.6

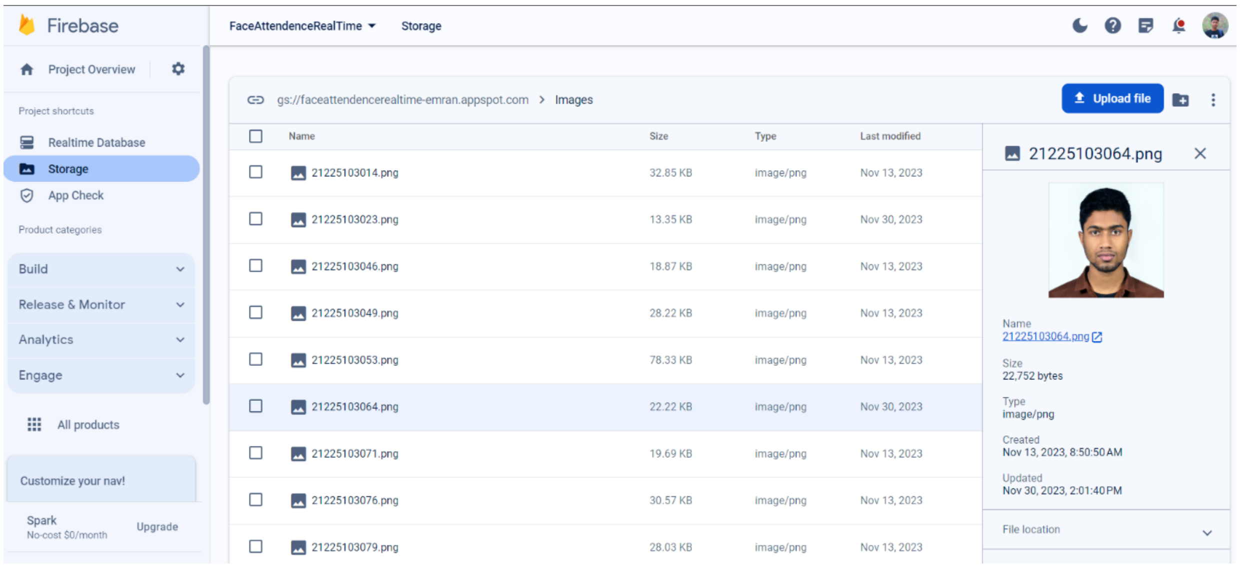
EncodeGeneratoy.py

Certainly! This Python script serves the purpose of encoding facial features from a set of student images and storing these encodings along with their respective IDs.



```python
import cv2
import face_recognition
import pickle
import os
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from firebase_admin import storage

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(
    cred,
    {
        "databaseURL": "https://faceattendencerealtime-emran-default-rtdb.firebaseio.com/",
        "storageBucket": "faceattendencerealtime-emran.appspot.com",
    },
)

# Importing student images
folderPath = "Images"
pathList = os.listdir(folderPath)
print(pathList)
imgList = []
studentIds = []
for path in pathList:
    imgList.append(cv2.imread(os.path.join(folderPath, path)))
    studentIds.append(os.path.splitext(path)[0])

    fileName = f"{folderPath}/{path}"
    bucket = storage.bucket()
    blob = bucket.blob(fileName)
    blob.upload_from_filename(fileName)

    # print(path)
    # print(os.path.splitext(path)[0])
print(studentIds)

def findEncodings(imagesList):
    encodeList = []
    for img in imagesList:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList

print("Encoding Started ...")
encodeListKnown = findEncodings(imgList)
encodeListKnownWithIds = [encodeListKnown, studentIds]
print("Encoding Complete")

file = open("EncodeFile.p", "wb")
pickle.dump(encodeListKnownWithIds, file)
file.close()
print("File Saved")
```

Figure No. 3.7



gs://faceattendencerealtime-emran.appspot.com > Images

Name	Size	Type	Last modified
21225103014.png	32.85 KB	image/png	Nov 13, 2023
21225103023.png	13.35 KB	image/png	Nov 30, 2023
21225103046.png	18.87 KB	image/png	Nov 13, 2023
21225103049.png	28.22 KB	image/png	Nov 13, 2023
21225103053.png	78.33 KB	image/png	Nov 13, 2023
21225103064.png	22.22 KB	image/png	Nov 30, 2023
21225103071.png	19.69 KB	image/png	Nov 13, 2023
21225103076.png	30.57 KB	image/png	Nov 13, 2023
21225103079.png	28.03 KB	image/png	Nov 13, 2023

21225103064.png

Name
21225103064.png

Size
22,752 bytes

Type
image/png

Created
Nov 13, 2023, 8:50:50 AM

Updated
Nov 30, 2023, 2:01:40 PM

Figure No. 3.8

3.4 Final Output

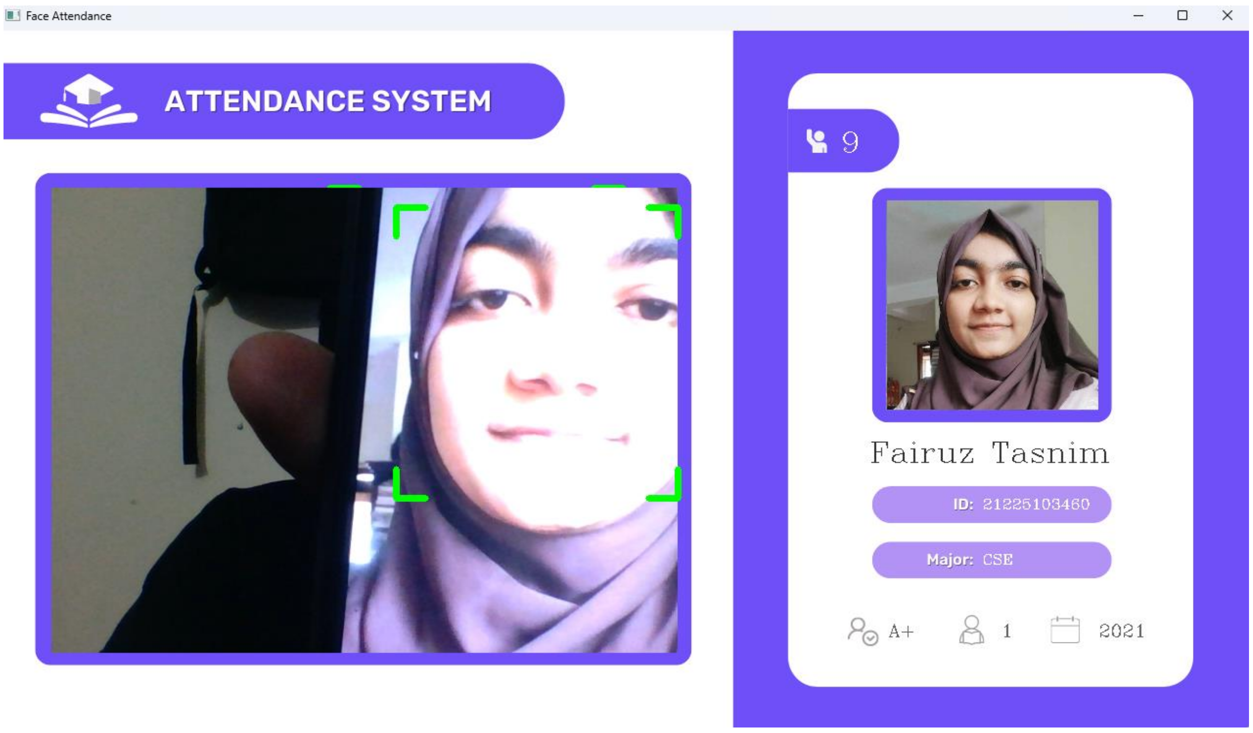


Figure No: 3.9

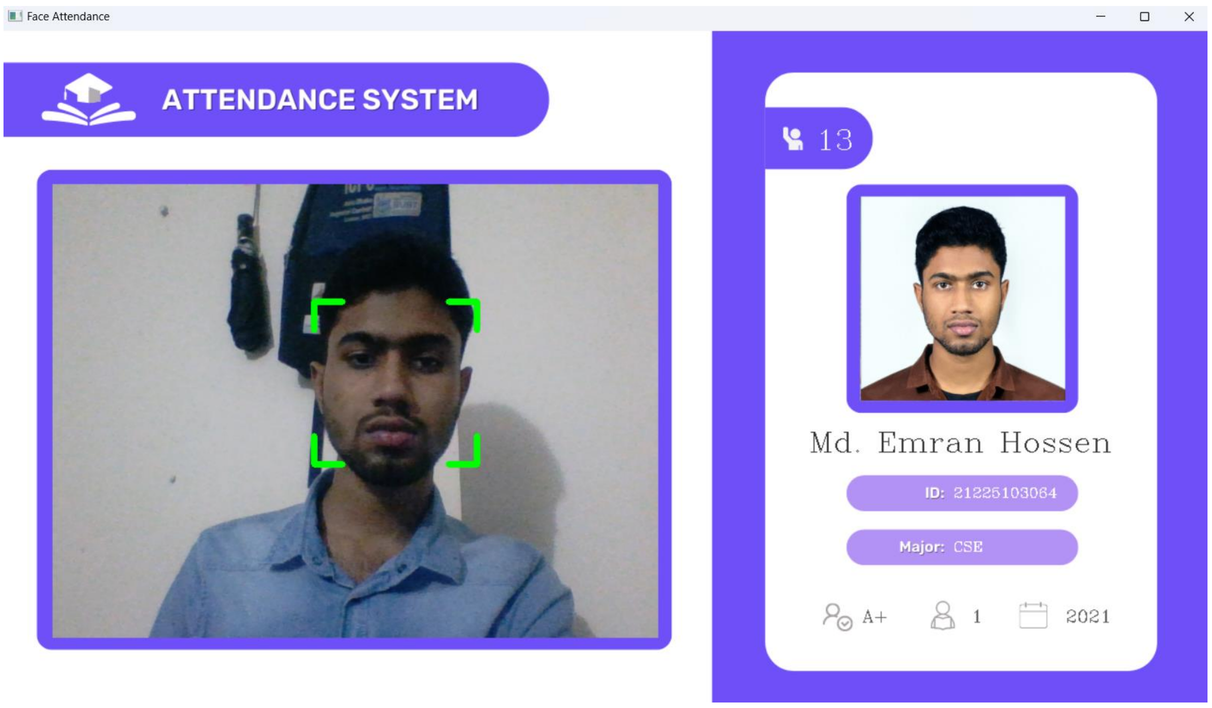
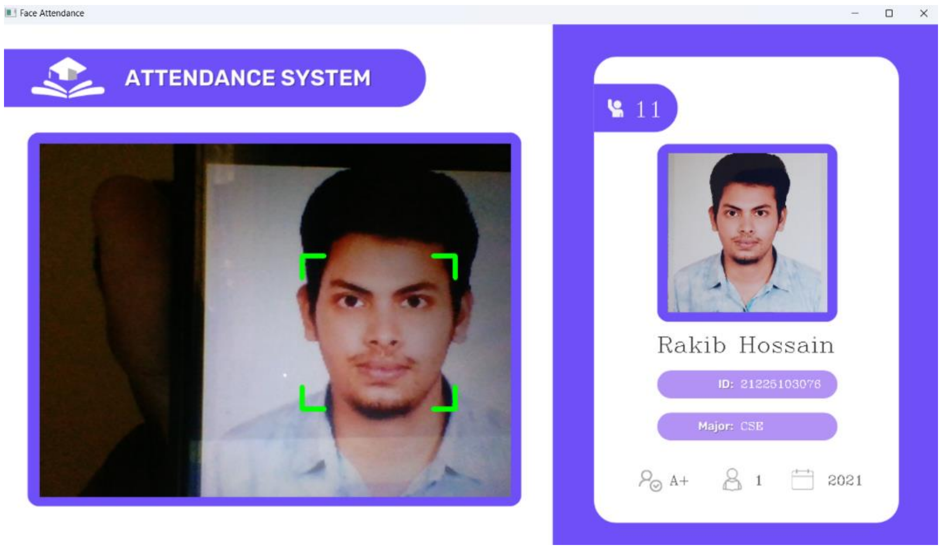


Figure No: 3.10



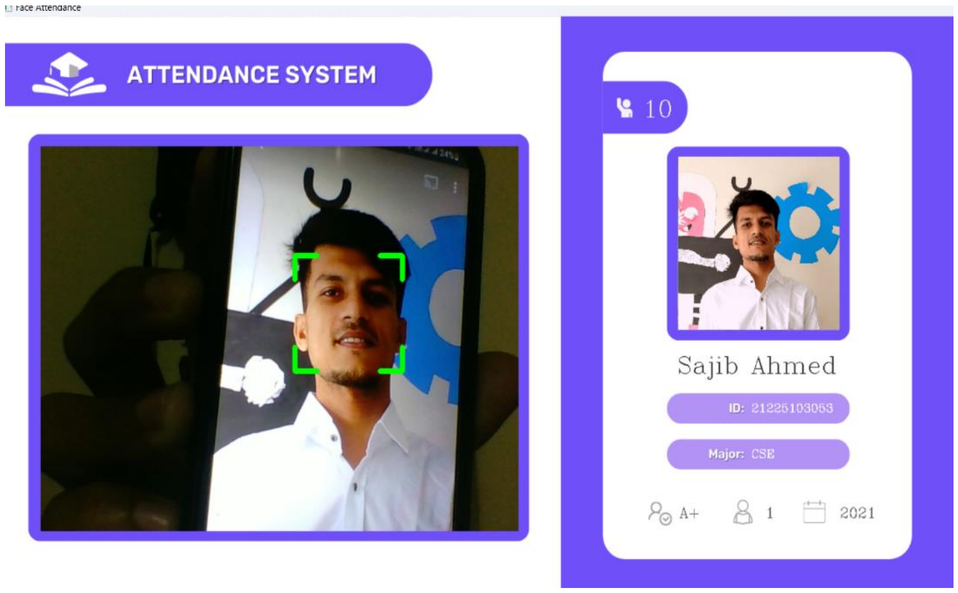
Rakib Hossain

ID: 21225103076

Major: CSE

A+ 1 2021

Figure No: 3.11



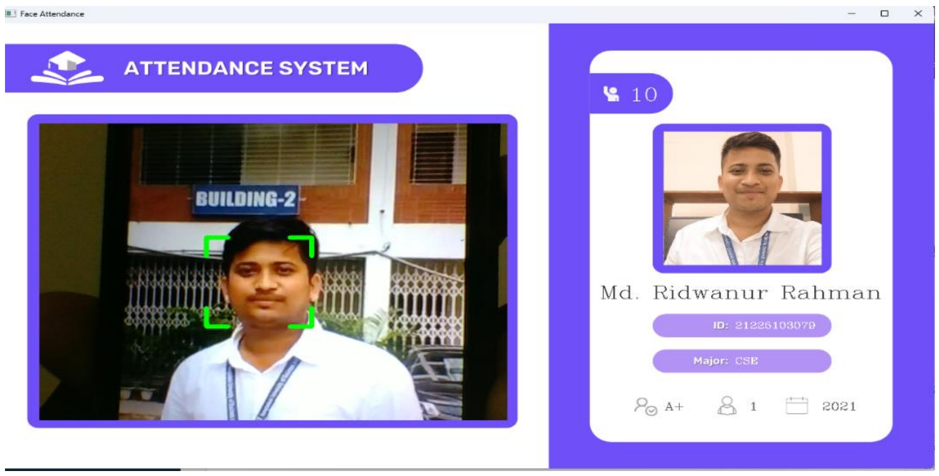
Sajib Ahmed

ID: 21225103053

Major: CSE

A+ 1 2021

Figure No: 3.12



Md. Ridwanur Rahman

ID: 21225103079

Major: CSE

A+ 1 2021

Figure No: 3.13

CHAPTER 04
CONCLUSION

4.1Activity And Time Schedule

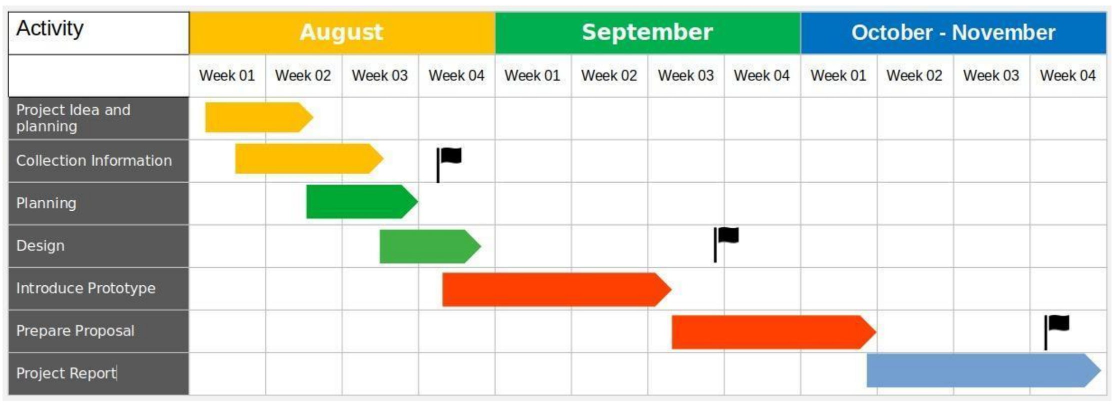


Figure No. 4.1

4.2 Future Scope

- **Government/ Identity Management:** Governments all around the world are using face recognition systems to identify civilians. America has one of the largest face databases in the world, containing data of about 117 million people**.**

- **Emotion & Sentiment Analysis:** Face Detection and Recognition have brought us closer to the technology of automated psyche evaluation. As systems now a days can judge the precise emotions frame by frame in order to evaluate the psyche.

- **Authentication systems:** Various devices like mobile phones or even ATMs work using facial recognization, thus making getting access or verification quicker and hassle free. •Full Automation: This technology helps us become fully automated as there is very little to zero amount of effort required for verification using facial recognition.
- **High Accuracy:** Face Detection and Recognition systems these days have developed very high accuracy and can be trained using very small data sets and the false acceptance rates have dropped down significantly.

4.3 Limitations

- **Data Storage:** Extensive data storage is required for creating, training and maintaining big face databases which is not always feasible.
- **Computational Power:** The requirement of computational power also increases with increase in the size of the database. This becomes financially out of bounds for smaller organizations.
- **Camera Angle:** The relative angle of the target's face with the camera impacts the recognition rate drastically. These conditions may not always be suitable, therefore creating a major drawback.

4.4 Conclusion

Facial Detection and Recognition systems are gaining a lot of popularity these days. Most of the flagship smartphones of major mobile phone manufacturing companies use face recognition as the means to provide access to the user. This project report explains the implementation of face detection and face recognition using OpenCV with Python and also lays out the basic information that is needed to develop a face detection and face recognition software.The goal of increasing the accuracy of this project will always remain constant and new configurations and different algorithms will be tested to obtain better results. In this project, the approach we used was that of Local Binary Pattern Histograms that are a part of the Face Recognizer Class of OpenCV.

4.5 References

[1] Schneiderman.United States of America Patent U.S. Patent No. 8,457,367, 2013.

[2] R. J. Baron, “Mechanisms of human facial recognition,” International Journal of ManMachine Studies.

[3] M. Nixon, “"Eye Spacing Measurement for Facial Recognition",” International Society for Optics and Photonics., vol. (Vol. 575), (19 December 1985).

[4] H. &. Y. J. Yu, “A direct LDA algorithm for high-dimensional data—with application to face recognition,” 2001.

- https://opencv.org
- https://www.geeksforgeeks.org
- https://firebase.google.com
- https://www.analyticsvidhya.com
- https://opencv.org/books