iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.43393

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Electronic Attendance Recorder and Confirmation System Using Facial Identification Modules in Python

Zenith Nandy[1], Indrani Sengupta[2], Shuvashish Mondal[3], Sk Sahil[4], Mr. Soumen Pal[5]
[1, 2, 3, 4]*Student, Department of Electronics and Communication Engineering, Narula Institute of Technology, Kolkata*
[5]*Assistant Professor, Department of Electronics and Communication Engineering, Narula Institute of Technology, Kolkata*

Abstract: ***The following paper demonstrates about a project that is basically an automated pre-programmed attendance management system. The system implements various image processing libraries/modules of the Python programming language. The main dependency is the face_recognition library, contributed by Adam Geitgey. The other dependencies used here are openCV, xlsxwriter, dlib, datetime, numpy and smtp service. This system analyzes camera roll frame by frame, looks for the known faces if found any, records the corresponding name and the time of detection. An email is sent to that person confirming the attendance with a mail id to report any unusual activities. After the system closes, the attendance details (name and first detection time) is saved in an excel sheet.***

Keywords: **Attendance system, face recognition, facial identification modules, python, opencv, email service using python**

I. INTRODUCTION

To give a proper introduction, the designed program records attendance based on the faces it knows; the known face encodings to be exact. If it is able to detect one or more known faces, the corresponding name and time is recorded and an attendance confirmation mail is sent to the person whose face has been detected at that moment. The email holds the name, time and an email id to report is there is an issue with the attendance. The email id to report the issue is a dummy email and its only used as a portrayal to how it can be upgraded and implemented in real life scenarios. All the programming has been done using Python programming language with the version 3.10.

The program requires manual intervention to stop. The ‘q’ key of the keyboard is pressed to exit and complete the running system and save the collected data. This data may be useful for analysis in future or to cross check and detect any foul play.

II. FLOW OF THE PROGRAM

As discussed above, whenever a known face is detected in the frame, the model automatically identifies it with the corresponding name and a confirmation mail is sent to the concerned person’s mail id. This process slows down the frame rate a bit as the face detection process undergoes quite a lot of data processing and validation.

The flow of the program is as given:

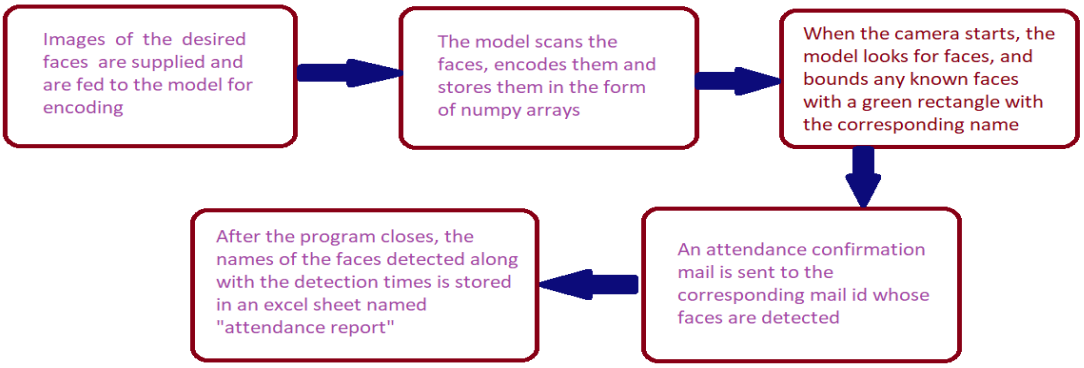


Fig. 1 Flowchart of the entire process of the program

©IJRASET: All Rights are Reserved

III.MODULE DESCRIPTIONS

A. Face Recognition

It is a python library contributed by Adam Geitgey having its initial release on March 4, 2017. This library has several inbuilt methods that facilitates various graphical operations related to images of human faces. The library has to be installed in the local system and then accessed using the command "**import face_recognition**". The several methods of the library that are used here are as follows:

1) face_recognition.face_encodings: This method detects and encodes the faces detected in a given image in the form of numpy arrays. If there are multiple faces in an image, an array of encoding arrays is formed with the default encoding order from left to right.
2) face_recognition.face_locations: This method detects the faces and returns the four co-ordinates i.e pixel locations of an imaginary rectangle. This imaginary rectangle encloses the faces from forehead to chin and the cheeks on the left and right. In case of multiple faces, an array of tuples of face location co-ordinates, is stored.
3) face_recognition.face_distance: This method calculates the difference between two face encodings and returns a single decimal value between 0 and 1. The less the value, the more matching are the encodings. Practically 0 or 1 is never returned for any two different face encodings.
4) face_recognition.load_image_file: This method helps to load the image file from the local system.
5) face_recognition.compare_faces: This method internally runs the **face_recognition.face_distance** algorithm and returns true if the numerical value is less than 0.6. For our convenience, we have set this limit or tolerance as 0.56.

B. OpenCV

It is one of the most popular python library for image processing. OpenCV provides a lot of methods that enable various operations related to graphical changes in an image. Here the OpenCV library is used to change the color channel, draw a rectangle, put a text on it and show the final output image

C. Smtplib, email.message

These libraries contain methods that methods that help us to send email from our python program using the SMTP protocol.

D. Xlsxwriter

This is a python module used to write Excel files from 2007 onwards and save them in .xlsx format. Any form of text can be inserted into the excel file. Here the attendance report is being prepared using this module to record the name and time of the faces detected.

E. Datetime

This python module helps to work with the date and time of the current system, and use them in the way we require. We can easily manipulate them and change the time zones or add days, hours, seconds or any amount of time with the current date and time.

F. Numpy

This module supports various numerical operations other than the ones defined in "math" module. In this program, the face encodings are explicitly converted to numpy array so that matrix operations can be performed on them.

IV. DETAILED INSIGHT OF THE PROGRAM

Already the brief overview of the project is discussed in the abstract and introduction section. Here the steps are a bit elaborated for better understanding and further clarity.

A. Importing the libraries

The required dependencies for the above program is mentioned in the "module descriptions" sections are imported. They contain the methods that are used to perform the required operations of the project. The libraries imported are shown in the below image.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

```
import cv2
import face_recognition
import smtplib
from email.message import EmailMessage
import dlib
import os
import xlwt
import datetime
import numpy as np
```

Fig. 2 Required prerequisites for the project

B. Supplying the Input Images

The project is fed with the facial images of four personalities; MS Dhoni, Sachin Tendulkar, Rohit Sharma, and Lionel Messi. While the program is running, if the image detected on camera contains the faces of any of the above personalities, the program will detect the face and record their attendance only once.



Fig. 3 Reference images to load the encodings in the system

C. Testing the Model

After the required coding is done, the model is finally run and put in action. Once started, the webcam of the local system is opened. When some people or a picture with some people is brought in front of the camera, the unknown faces are bounded with a red rectangle and the known ones are bounded with a green rectangle and the name below the rectangle. As soon as the known faces are detected, their attendance with the current time is recorded and saved in an excel file.

An attendance confirmation mail is sent to the corresponding email address featuring the name and time. The 'q' key of the keyboard is used to close the camera, which eventually completes the run and closes the program.

D. Observing the output

With all the coding done and a successful running of the program, following output is recorded:



©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |





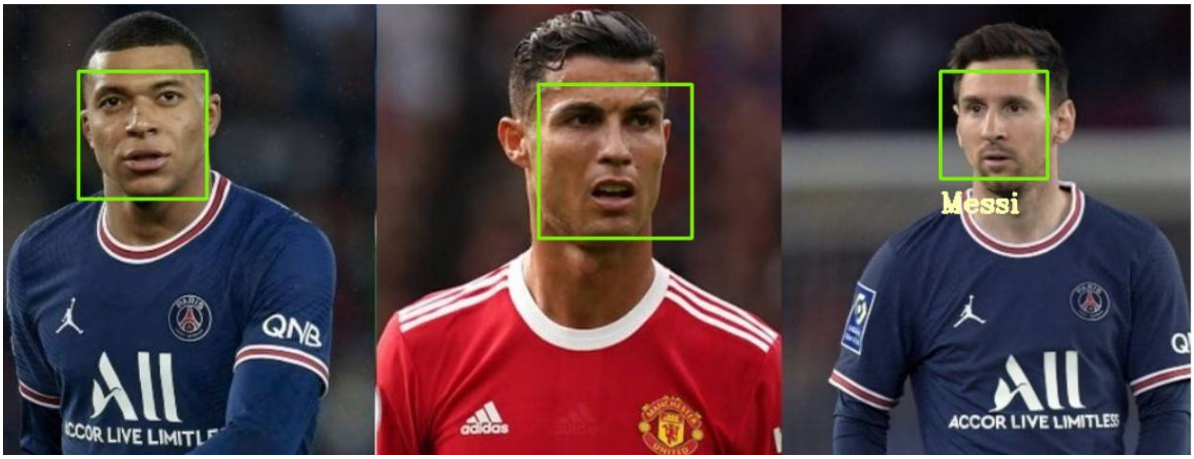


Fig. 3, 4, 5, 6 Output images when given as input to the program

	A	B
1	**Names**	**Time**
2	Rohit	00:11:40
3	MS Dhoni	00:12:37
4	Messi	00:13:09
5	Sachin	00:14:25
6		

Fig. 7 Recorded attendance with names and corresponding time

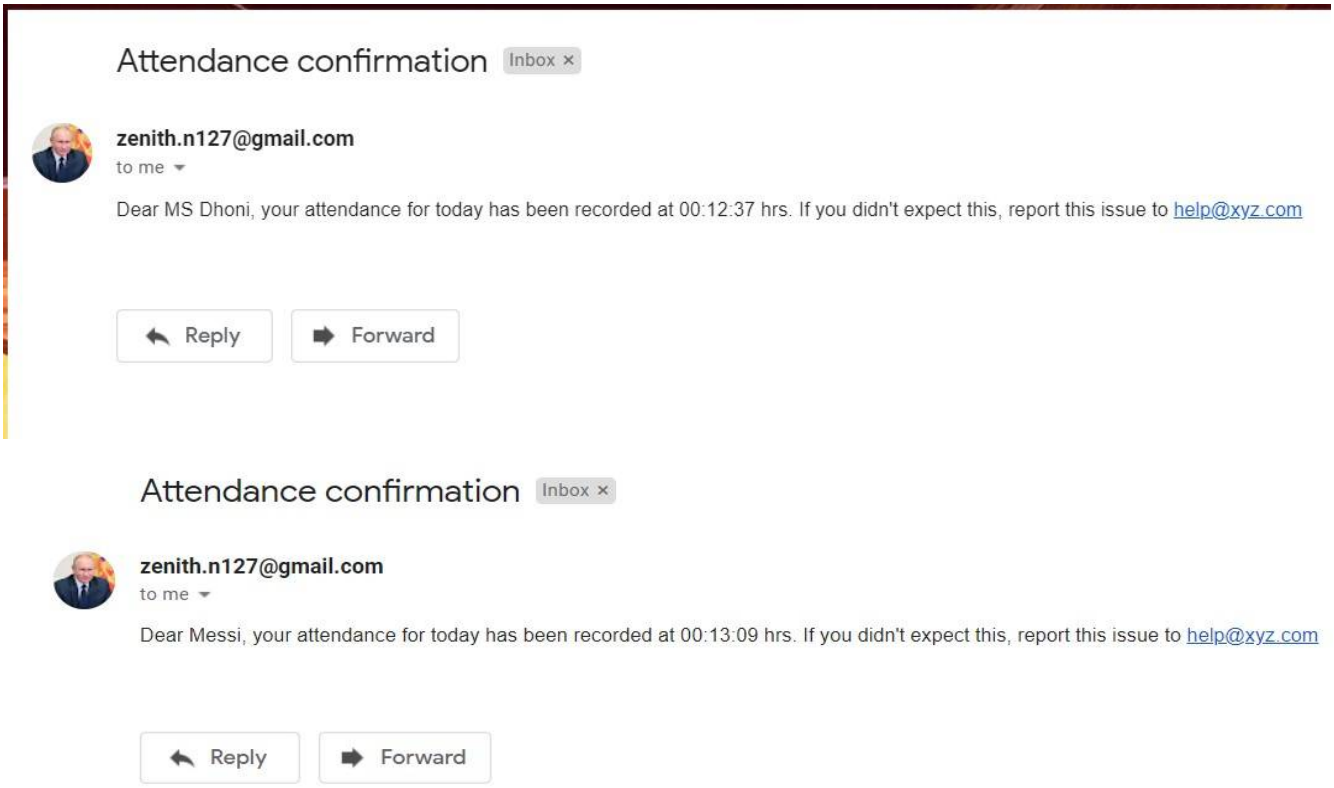


Fig. 8,9 Confirmation mails to the concerned persons (2 out of 4 shown here)

V. CONCLUSIONS

The model has been tested with various images and was found to be fruitful. The results are very accurate and this model can be relied upon for any practical use. In this project, it is used as a smart attendance system, but in real life it can be used in various other ways. Examples include: CCTV monitoring of incoming and outgoing people, finding a missing person from video footage, security purposes, etc.

Several face recognition based systems are implemented across the globe. This particular project can be further developed to a more advanced stage in order to achieve higher scalability and functionality.

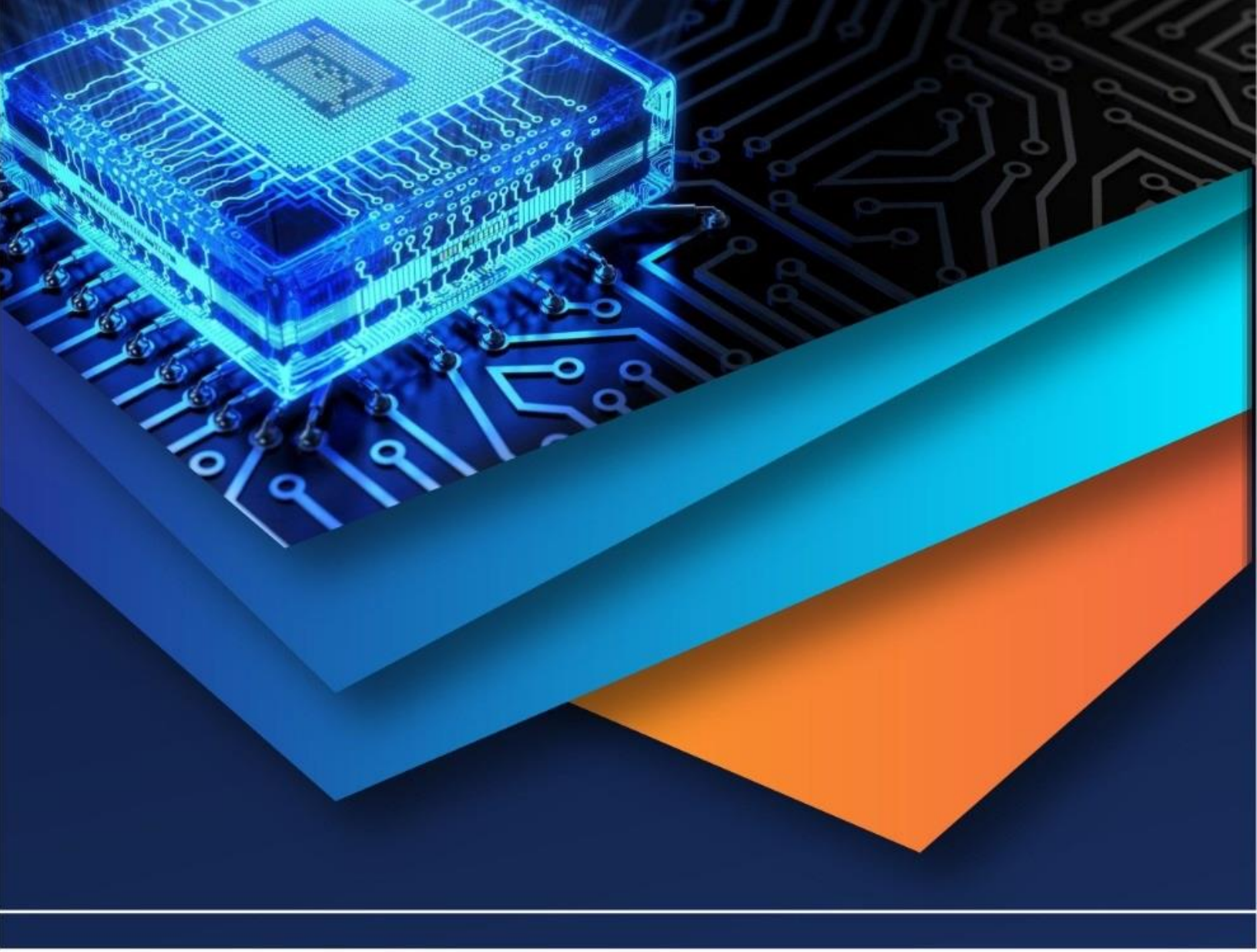
VI. ACKNOWLEDGMENT

A warm gratitude to the group members who helped me in the accomplishment, to my college Narula Institute of Technology, to our respected H.O.D (Head of Department) Dr. Anilesh Dey, our respected T.I.C (Teacher In Charge) Dr. Sangita Roy and to our mentor Mr. Soumen Pal who helped us to collect resources and build up this project to a full fledged working system. To be honest, there are many external help taken as well from internet and other learning resources. To sum up this is an assembled product of many contributions which are the key factors for this publication successful. Thank you everyone.

REFERENCES

[1] Donald B. Generry, *Object Detection and Measurement Using Stereo Vision*, August, 1979.
[2] Ahmad Tolba. Ali El-Baz, Ahmed A El-Harby, *Face Recognition: A Literature Review*, researchgate.net. January, 2005.
[3] https://www.youtube.com/watch?v=7IFhsbfby9s
[4] Xiao Han, Qingdong Du, *Research on Face Recognition Based on Deep Learning,* available at ieee.org, May 2018. DOI: 10.1109/DINWC.2018.8356995
[5] https://pypi.org/project/face-recognition/
[6] https://www.courier.com/blog/three-ways-to-send-emails-using-python-with-code-tutorials/

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)