

An OpenCV Tool for Data Logging by Face Recognition

Manimala K

Assistant Professor, Government College of Engineering, Salem

Abstract

The human face contains various features and characteristics, which is not unique for two humans. Each human has different identity. Face recognition is an identification system that uses the characteristics of a person's face to obtain the person's identity. The human face recognition process consists of two stages: face detection, here the camera detects that there is a face in front of it; this process is rapid in humans and the next is recognizing or identifying that face among a group of people. The recognition stage is mostly worked upon to develop models for facial recognitions. This paper presents a new method used to detect and recognize the face in real-time environment using Raspberry Pi and Python. The system uses a camera that will be able to detect the human face, auto counter and log into the system. A single portal is used for a human's data Entry and Exit. These Entry and Exit functions are triggered by GUI. The information of the members are stored in a Spreadsheets are used in this paper. In this paper LBP histograms is used for face recognition and a modified algorithm of Haar's Cascades is used for face detection.

Keywords: Face Recognition, Local Binary Pattern Histograms, OpenCvPython, Haar's Cascade, Raspberry Pi

I. INTRODUCTION

The use of biometrics has led to the automation of verification of a person's identity. The verification is done based on a person's physiological or behavioral characteristics [6]. Facial recognition, one of the methods having the highest accuracy and low intrusiveness, has been used in this project for the automation of data entry. Face Recognition is an identification system that uses an individual's facial characteristics to obtain their identity. Face recognition and detection are two different phenomena on a computer. Detection refers to the objective of locating the faces on an image, crop and extract them so that they can be used by the recognition algorithm, whereas recognition refers to the objective of finding characteristics that describe or that are unique to an image.

To encapsulate, the purpose of this paper is to provide a clear understanding of the methods of achieving facial recognition and implementing it into a hardware prototype hence introducing a relatively cheap, highly reliable, and user-friendly automated authentication system that relies on the facial features of a human.

The problem statement is described in section II of this paper. Section III shows how we ended up choosing the algorithm, classifier, and the processor required for executing the project. The next section illustrates the hardware setup and the user interface. Section V elucidates the proper workflow of the program. The results obtained have been discussed in section VI, and the conclusion is drawn in the last section.

II. PROBLEM STATEMENT

To build a system which can detect a member's face and monitor their time of entry and exit from the specific confidential lab or room for concern. There should be a single portal where any activity at the door will be auto countered and logged.

Any confidential room has a single entryway so that monitoring the activity at the door becomes easier. For both entry and exit had to be logged, we used two separate cameras for the two functions, both of which would be logged in a single portal. Both the cameras will be installed at specific angles, so that they easily capture the face of the person. A member can enter and exit the premise multiple times a day so every cycle of entry and exit will be logged separately. Spreadsheets will be generated every time the program is run, which would consist of the date and time of entry and exit, and other details such as registration number and phone number of the member. The operator would be responsible for the registration of a new member, and running and stopping of the program.

III. SOLUTION

To execute the project, we had to settle on three things. First was which algorithm to apply, the second was which classifier to use and the third was which processor to choose. Every algorithm and processor has their own merits and demerits depending on the requirement. The selection criteria shown in Fig. 1.

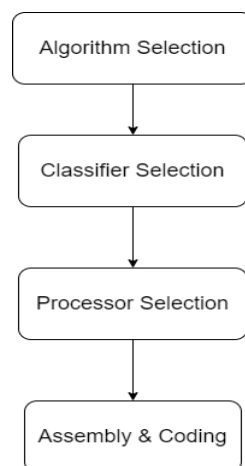


Fig. 1. Methodology Flowchart

A. Algorithm Selection

Facial recognition is an effortless task for a human being, but it was found to be very complex to achieve the same feat on a computer as many variables can inhibit the performance and accuracy of the algorithm. Numerous methods have been proposed to create facial recognition models and algorithms. Each of these methods can be used to recognize faces. Selection of any of these methods depends on a plethora of constraints [4] that impact the performance and accuracy of the models. Some of the algorithms are mentioned in [6]:

- 1) Eigenfaces
- 2) Local Binary Patterns Histograms (LBPH)
- 3) Fisherfaces
- 4) Scale Invariant Feature Transform (SIFT)
- 5) Speed Up Robust Features (SURF)

The algorithm chosen out of the aforementioned was LBPH. LBPH is one of the simplest face recognition algorithms that can exhibit local features in an image. It is robust against monotonic grayscale transformations and is possible to get results, mostly in a controlled environment [5]. Facial recognition using LBPH involves the comparison of two histograms, one from the recorded input, which checks with all the saved histograms searching for a match. The match is decided by a ‘confidence’ value. This value is based on the distance calculated between the two histograms. The following points were necessary in our algorithm choice:

- The OpenCv library is equipped with LBP functions [7]. This library can be availed by programming software such as C++, Python, Matlab etc. making it highly portable.
- LBP is capable of extracting patterns and contours from specific image samples. These patterns are used to exhibit the regional characteristics of the image.
- Compared to LBP, the storage space required by the Fisherfaces algorithm is less. LBP also uses a lesser time to process the image.
- Eigen faces method when processing images or data, it looks at the entire dataset at once, whereas in LBP every image is evaluated individually.

B. Classifier Selection

- 1) Artificial Neural Network
- 2) Support Vector Machine
- 3) Haar’s Cascading Classifier

All other classifiers require high computational power. Haar-like features are digital imaging features used in object recognition. They owe their name to their intuitive similarity with Haar wavelets and were used in the first real-time face detector. The requirement for this classifier was to be fast in order to be implemented on low-power CPUs, such as cameras and phones. Hence for small datasets, this is the best classifier as it provides fast computational operations through the integral image technique [8].

C. Processor Selection

- 1) Raspberry Pi 4 (4GB variant)
- 2) NVIDIA Jetson Nano

From the Table I, it can be seen that we will achieve the same computational powers from both the processors as their specifications are similar, hence the price of both the processors was the main criteria for our processor selection so we opted for Raspberry Pi

IV. INTERFACING AND HARDWARE

A. User Interface Window

A user interface is provided to ease and animate the registering process of a member. The window poster’s widgets such as input fields, ‘CLEAR’ buttons and a series of control buttons. The project code uses the standard Tkinter GUI library which when complemented with Python renders a simple, rapid and efficient way for GUI creation.

AutoCounter And Logging

Serial Number	1	Clear
Enter ID	RA1611005010379	Clear
Enter Name	HASHIM	Clear
Enter Department	EEE	Clear
Enter Contact No	9176322315	Clear
Notification :	Images Saved for ID : 1 Name : HASHIM	

Take Images	Train Images	Entry	Exit	Quit
-------------	--------------	-------	------	------

Status :	
----------	--

Fig. 2. GUI Window

The steps listed below provide a fundamental understanding for implementing the Tkinter module to obtain the GUI window shown in Fig. 2.

- Import the Tkinter module using the “import” command.
- Then create the main window of the GUI application. The size of the window can be varied.
- Add and create widgets on the main window and set its attributes.
- To ensure continuous dynamic running of the GUI, create a loop for each action desired by the user.

A. Hardware Setup

The hardware required by the on-site system shown in Fig. 3. utilizes the following components:

1) Web camera

A total of two web cameras are to be installed. Each camera will be assigned entry and exit function tasks to be performed. The webcam possesses a resolution of 720p recorded at 30 FPS.

2) Raspberry Pi

A single Pi 4B module is sufficient enough for handling the database and running the instruction set. It is recommended to provide casing to the raspberry Pi to safeguard it from the hazardous external stimulus. Cooling fans can also be provided to keep the system cool.

3) Peripherals

Input peripherals such as keyboard and mouse are required to update the database and provide the operational control of the program physically. A display monitor is also installed at the site.

4) SD Card

A storage device that stores the Raspbian OS and the Database of members registered. A 32 GB SD card is used in the hardware. With each captured image having a size of 4kb, an approximate of 17,500 people can be identified.

V. WORKING

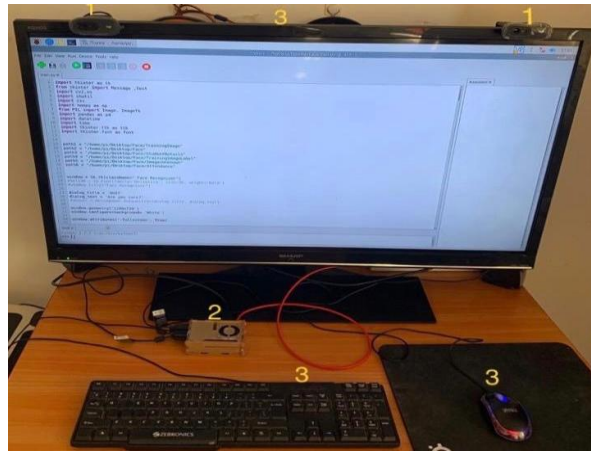


Fig. 3. Basic Hardware Setup

The program directory of the project is illustrated in Fig. 4. The folders shown in the image need to be created before the program is run for the very first time. This step is vital to the running of the program. The ‘haarcascade_frontalface’ is the classifier file which helps in detecting the frontal face of a person [1].

The working of the code is mainly divided into four sections. Each section is activated by the respective buttons as explained below:

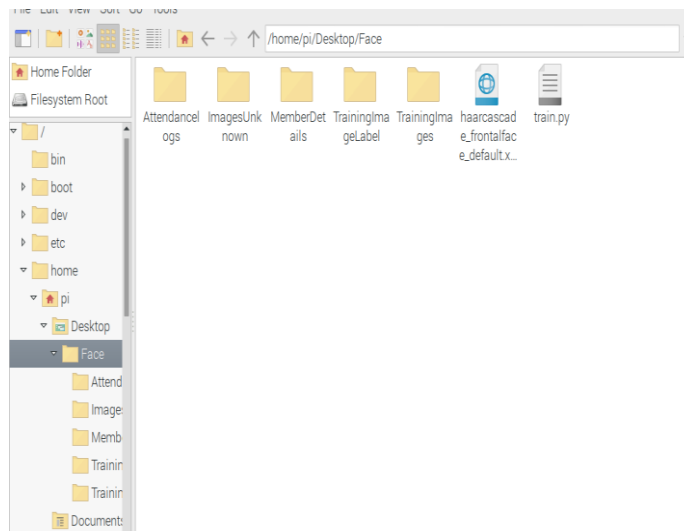


Fig. 4. Program Directory

Take Images

This section of the code performs two major tasks. Firstly, it does the registration process of the member by storing the inputted data from the user into an excel sheet and saves it in the folder “Member Details” as illustrated in Fig. 5. The second task is Image Capture, wherein a minimum of 100 images are captured and stored in the folder “Training Image”. Each image is saved using their corresponding serial number and name.

Id	Reg Number	Name	Department	Contact Number
1	RA1611005010379	HASHIM	EEE	917632315
2	RA1611005010315	akshay	eee	8876749526
3	RA1611005010347	ANANTA	EEE	9969203965
4	201464	Vidhya	EEE	8220538151

Fig. 5 Member Details



Fig. 6. Training Image folder

Train Images

In this section, all the images from the “Training Image” folder is trained. After the training is done a ‘yml’ file is created in the “TrainingImage Label” folder, which contains all the image descriptors of the trained images. These descriptors are then used to recognize faces.

C. ENTRY

This section of the program performs the following tasks.

1) Recognition

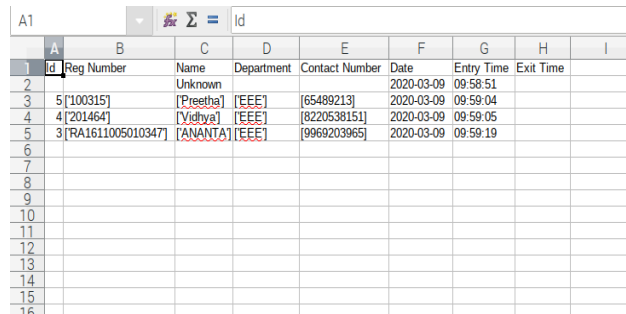
Once the ‘Entry’ button is clicked, a camera window showing the live video feed pops up. The camera scans the environment for faces. When a face is detected, it verifies the detected face with ‘yml’ file created in the previous step.

2) Display and Marking

If a match is found, the person’s Sr no and name are displayed below the bounding box. If no match is found the face is marked as “Unknown” and images of the unknown person is stored in the “Images Unknown” folder.

3) Logging

When a recognized face is detected, the system logs the corresponding information of the detected member stored in the “Member Details” folder into a newly created spreadsheet. This spreadsheet is saved in “Attendance Logs” folder. The spreadsheet also contains the entry time of the member. In



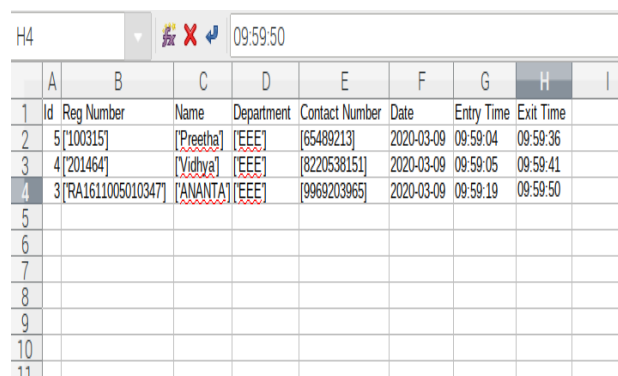
Id	Reg Number	Name	Department	Contact Number	Date	Entry Time	Exit Time
		Unknown			2020-03-09	09:58:51	
5	[100315]	[Preetha]	[EEE]	[65489213]	2020-03-09	09:59:04	
4	[201464]	[Vidhya]	[EEE]	[8220538151]	2020-03-09	09:59:05	
3	[RA1611005010347]	[ANANTA]	[EEE]	[9969203965]	2020-03-09	09:59:19	

Fig. 7. Entry Log sheet

case an unknown person is detected, the time of entry of the unknown person is logged.

A. Exit

This section of the program functions similar to the “Entry” section, with the exception of handling the logging task. In the event of an unknown person being detected, no logging of the unknown person takes place. Only recognized faces are logged in with their respective exit times. A



Id	Reg Number	Name	Department	Contact Number	Date	Entry Time	Exit Time
5	[100315]	[Preetha]	[EEE]	[65489213]	2020-03-09	09:59:04	09:59:36
4	[201464]	[Vidhya]	[EEE]	[8220538151]	2020-03-09	09:59:05	09:59:41
3	[RA1611005010347]	[ANANTA]	[EEE]	[9969203965]	2020-03-09	09:59:19	09:59:50

Fig. 8. Exit Log sheet

separate exit file is created under the same “Attendance Logs” folder.

VI. RESULTS

The program is sequenced initially by the member registration process, prompting the user to enter their requested credentials as shown in Fig. 2. Once the registration is complete the notification bars are updated accordingly. The images need to be trained before the LBPH recognizer function can be called. The train image button triggers the instructions to train the image dataset and create the .yml file. Once this is done the program is ready for recognition.

The entry and exit keys activate the respective cameras when pressed. These functions can also be made to run simultaneously by assigning a single activation command to both the exit and entry cameras depending upon the requirement.

The camera scans the environment for faces and then tests those faces with the data stored in the registry.

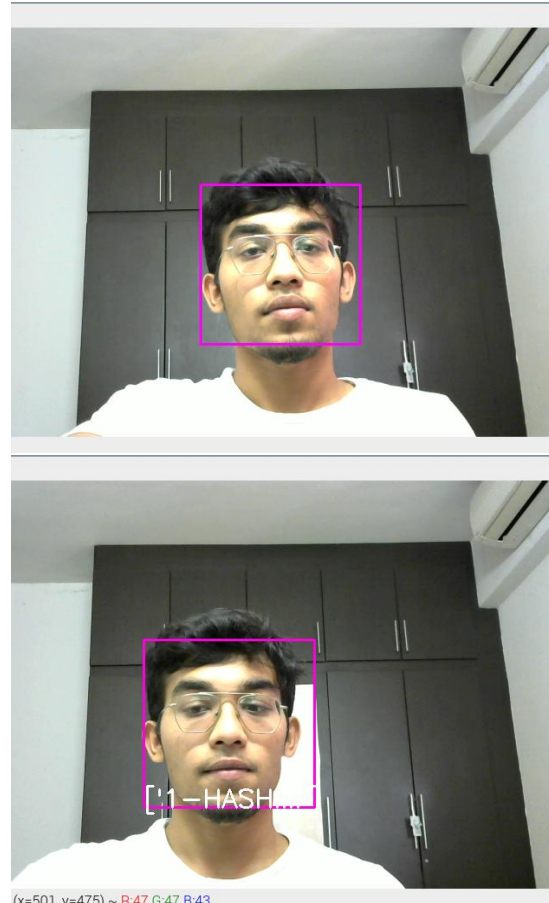


Fig. 10. Recognition

On recognition the person's credentials are logged in an excel sheet with the corresponding entry and exit timestamps. The entry file logs in the details of the people that are recognized by the camera. This case also successfully tested the ability of the system to locate any unidentified individual in the vicinity of the cameras. For handling unidentified entries, the program can also provide other kinds of alerts as required by the operator. Using a web API and python, sms alerts can be sent to the operator. To increase the security even more, door locks can be complemented with this concept that allows the door only to open when a recognized face is detected.

VII. CONCLUSION

Facial recognition in its simplicity is a very common feature in people that turns out to be quite a task on a computer. Nowadays security incentives come up in our daily lives, be it personal devices, objects, places, or basically in any kind of situation where a check is required to make sure that legitimate utilization is done. With continuing improvements in technology, the term security in itself redefines the significance of speed, accuracy, reliability and the ease by which it can be used and procured. Emplacing these factors as the foundation, we employed the LBPH method for facial recognition which provides us with a high accuracy. OpenCv serves the purpose of loading the pre packed Haar classifier files and functions on the Python platform. Embedding these features on a Pi, a simple feature such as recognition becomes possible.

With time, better cameras will be out in the market which will make security systems more accurate. As this system can count the number of persons in a room, additional features can be added such as control of lights and air conditioner, and alarm system; this is possible as Raspberry Pi has GPIO pins. Such systems can be implemented for situations that involve verification of a person, attendance management, logging and in some places it also helps in crime control. The entire system has no restrictions on size and cost making it portable and very cheap as well.

REFERENCES

1. Omkar Biradar, Anurag Bhawe, "Face Recognition Based Attendance Monitoring System using Raspberry-pi and OpenCV". International Research Journal of Engineering and Technology (IRJET). Volume: 06 Issue:01, Jan 2019, pp. 714-715.
2. Abdenour Hadid, and Matti Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition Timo Ahonen", IEEE, 5th June 2006, [draft], pp. 2-6.
3. Ashish Earnest Rahul, Face Detection and Recognition System Using Raspberry Pi. International Journal of Research in Engineering, Science and Management Volume-1, Issue-12, December-2018, pp. 323-325.
4. Shrutika V. Deshmukh, Prof Dr. U. A. Kshirsagar, "Face Detection and Face Recognition Using Raspberry Pi". International Journal of Advanced Research in Computer and Communication Engineering, Volume-6, Issue 4, April 2017, pp. 71-72.
5. Suthagar S, Augustina Shaglin Ponmalar, Benita, Banupriya, Beulah. "SMART SURVEILLANCE CAMERA USING RASPBERRY PI 2 AND OPENCV". International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Vol. 3, Special Issue 19, April 2016, pp. 178-180.
6. Shang-Hung Lin, "An Introduction to Face Recognition Technology" (Ph.D. IC Media Corporation), Informing science special issue on multimedia informing technologies part 2, Volume 3 No 1, 2000. A. J. Wood and B. F. Wollenberg, Power Generation, Operation and Control. New York: Wiley, 1996, pp. 3-5.
7. Shervin Emami, Valentin Petrușuciu, "Facial Recognition Using OpenCv". Journal of Mobile, Embedded and Distributed Systems, ISSN 2067 – 4074, vol. IV, no. 1, 2012, pp. 38-41.
8. Priya Pasurmati, P.PurnaSekhar, "Classroom Attendance Using Face Detection and Raspberry-Pi". International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03, March 2018, pp. 168-170.