INDEPENDENT PROJECT

Face Recognition Using Python

A PROJECT REPORT

Submitted by

Arpit Kumar	(21BCS 1815)
Shreya Pandey	(21BCS 8286)
Ayushi	(21BCS 2849)
Chaitanya vatsyayan	(21BCS 2414)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Chandigarh University

May 2022

INDEPENDENT PROJECT

BONAFIDE CERTIFICATE

This is to certify that the project work entitled as **" Face Recognition using Python "** is being Submitted by Ram Ashish Yadav, Sujal Sharma , Neeraj Verma , Rishikant Singh , Manish Kumar, in the partial fulfillment for the award of the Degree of Bachelor of Engineering in "COPUTER SCIENCE & ENGINNERING" in the academic during 2021-2025.

Signature of the HOD

Ms. Ravneet Kaur

SUPERVISOR

Computer Science & Engineering

Signature of the Supervisor

Dr. Gaurav Bathla

HEAD OF THE DEPARTMENT

Computer Science & Engineering

Submitted for the project viva-voice examination held on _____ ___

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I thank the almighty for giving us the courage and perseverance in completing the main project. This project itself is acknowledgement for all those people who have given us theirheartfelt co-operation in making this project a grand success.

I am greatly indebted to project guide **Ms. Ravneet Kaur**, Assistant Professor, Computer Science & Engineering, for providing valuable guidance at every stage of this project work. I am profoundly grateful towards the unmatched services rendered by her. Our special thanks to all the faculty of Computer Science & Engineering and peers for their valuable advises at every stage of this work.

Last but not least, we would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation indoing the main project.

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

INDEPENDENT PROJECT

ABSTRACT

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisherface method. Facial image recognition Eigenface method is based on the reduction of facedimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project face detection system with face recognition is Image processing. The software requirements for this project is matlab software.

Most of these systems use a Passive Infrared (PIR) motion sensor for motion detection. Although affordable, such a system still has many limitations. For example, false alarms triggered due to an abnormal condition such as rapid heating from sunlight exposure. In this work, a vision-based home security system using OpenCV on Python 3 model B was developed to improve the effectiveness of motion detection. This system applied the Haar-Cascade algorithm coupled with background subtraction as well as considered the Histogram of Oriented Gradients (HOG) during the development stage.

The developed prototype was tested under a few conditions to determine the accuracy of motion detection and compare the results with a system that uses a PIR motion sensor for motion detection. From the results obtained, the developed vision-based home security system using OpenCV has 100% of detection rate compared to the PIR motion sensor-based security system with 76% of the detection rate.

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

INDEPENDENT PROJECT

INTRODUCTION

Face recognition is the task of identifying an already detected object as a known or unknown face.Often the problem of face recognition is confused with the problem of face detectionFace Recognition on the other hand is to decide if the "face" is someone known, or unknown, using for this purpose a database of faces in order to validate this input face.

The system is based on security that combines the functions of smart phone and home network system. It enables the users to monitor visitors in real-time, remotely via the IoT based doorbell installed near the entrance door to a house. Face Recognition is a technology in computer vision. In Face recognition / detection we locate and visualize the human faces in any digital image. It is a subdomain of Object Detection, where we try to observe the instance of semantic objects.

These objects are of class such as animals, cars, humans, etc. Detection technology has importance in many fields like marketing and security. This system makes security as further autonomous by capturing the images automatically and processing the image for facial matching and uses mail communication to the server to confirm the intruder is known or unknown.

SOFTWARE & HARDWARE IMPLEMENTATION

Open CV

Open-Source Computer Vision Library is an open-source computer vision and machine learning software library which is built to provide a common infrastructure for machine learning algorithms and computer vision. It has thousands of optimized algorithms which can be used different purposes like detecting and recognizing faces, identifying objects and many more. We need it to take pictures using our webcam and some manipulation needed to be done in the image. Although Intel started OpenCV, the library is and always intended to promote commercial and research use for free. OpenCV can take advantage of multicore processors because it is written in optimized C. There is active development on interfaces for Python and other languages. The OpenCV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics. Based on OpenCV first release 1.0, the most mature version was on Windows using 32-bit Intel architecture (IA32), followed by Linux on the same architecture. Once Apple startedusing Intel processors, portability to Mac OS X then became a priority as well, but this OS X port was still not mature compared to Windows and Linux versions. Support was also added to 64-bit Intel architecture (IA64) and 64-bit support on extended memory (EM64T).

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

INDEPENDENT PROJECT

Num Py

NumPy is the fundamental package NumPy is the fundamental package for scientific computing in Python which provides a multidimensional array object other mathematical operations can be performed using this but simply speaking we just need it to convert our images into some form of an array so that we can store the model that has been trained. NumPy builds on (and follows) the success of the Numeracy program. Its goal is to lay the foundation for a useful computer science environment. To better understand the people around NumPy and (its) Library-package) SciPy, we will explain a bit about how SciPy and (current) Numum it emerged. 1998, as a graduate student studying biomedical imaging at the Mayo Clinic in Rochester, MN, I discovered Python and its numerical expansion (Numbers) while looking for ways to analyze large Magnetic data sets Resonance Imaging and Ultrasound using high quality language. I quickly fell in love the Python program which is an amazing action statement about planning language. If I had not seen others with similar ideas, I would have I was very skeptical of my own intelligence. I became involved with the Numeric Python community, adding a C-API chapter to numerical texts (Paul Dubois kindly made me a writer).

Haar

A more sophisticated method is therefore required. One such method would be the detection of objects from images using features or specific structures of the object in question. However, there was a problem. Working with only image intensities, meaning the RGB pixel values in every single pixel in the image, made feature calculation rather computationally expensive and therefore slow on most platforms. This problem was addressed by the so called Haar-likefeatures, developed by Viola and Jones based on the proposal by Papageorgiou et. al in 1998. A Haar-like feature considers neighboring rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image.

Cascade

The cascade classifier consists of a list of stages, where each stage consists of a list of weak learners. The system detects objects in question by moving a window over the image. Each stage of the classifier labels the specific region defined by the current location of the window as either positive or negative – positive meaning that an object was found or negative means that the specified object was not found in the image. If the labelling yields a negative result, then the classification of this specific region is hereby complete, and the location of the window is movedto the next location. If the labelling gives a positive result, then the region moves of to the next stage of classification. The classifier yields a final verdict of positive, when all the stages, including the last one, yield a result, saying that the object is found in the image.

INDEPENDENT PROJECT

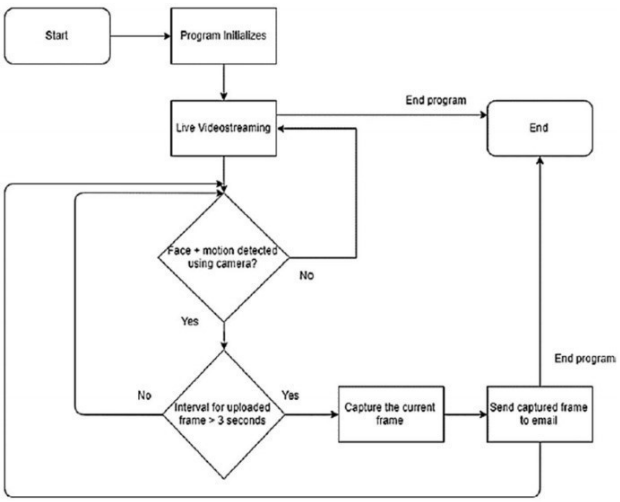
Python

Python is dynamically typed, and garbage collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020. In this article, you will learn how to build a face-recognition system using Python. Face recognition is a step further to face detection. In face detection, we only detect the location of the human face in an image but in face recognition, we make a system that can identify humans.

SYSTEM IMPLEMENTATION

Explore Different Libraries in Python. Get the insights about new technology. How to recognize face and using NumPy, open cv categorize different attributes of face. Our approach is to build a model that would sense data via camera or upload data like images or videos.

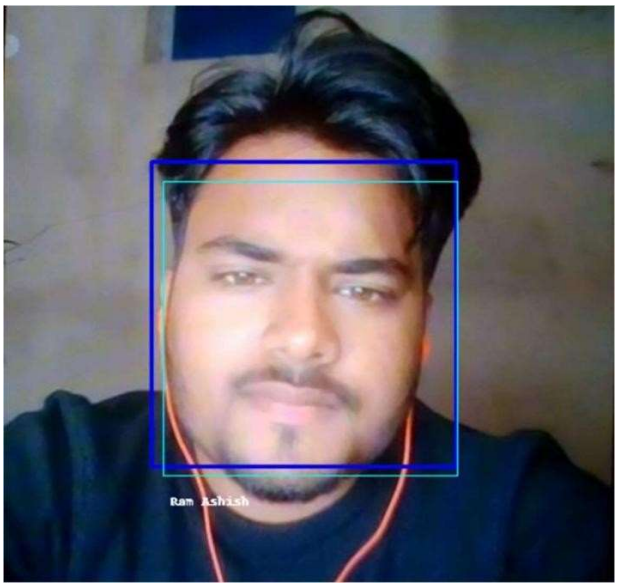


Live video streaming for capturing the original face:

(figure 3.1)

Face recognition using Haar cascade and using NumPy storing face detail to compare it with other faces.

Live video streaming for capturing the test face:



(figure 3.2)

Next, OpenCV was implemented on the camera to process the image before streaming. Two different approaches were attempted for this implementation, namely the Histogram of Gradients (HOG) descriptor algorithm and Haar- Cascade algorithm plus background subtraction. The HOG descriptor approach is more suitable for a pedestrian-based system, which has larger frame views. When the HOG descriptor was used in this system, instead of drawing a rectangle to indicate a human, it drew a large rectangle to occupy the whole frame. Haar-Cascade algorithm was then used to detect the body of a person in the frame. However, this algorithm alone was

also unsuitable for this system because of false detections due to the live video streaming. Thus, instead of using only Haar-Cascade for full body detection, Haar-Cascade for face detection was used coupled with the background subtraction method for motion detection. Figure 7 shows the captured frames using the two mentioned approaches. The background subtraction method used in the second approach was improvised and applied with Haar-Cascade algorithm to improve the effectiveness of the motion detection. The frame delta obtained from the background subtraction was compared with a threshold value of 255 to identify the difference in image regions corresponding to motion. Next, contour detection was applied on regions that passed the thresholding check. Once motion is detected, then the system will look for faces. If a face is detected, then the system will capture the current frame and send to the user through email. In addition, to prevent multiple uploads and spamming the intended email, the difference between the previous upload time and the current upload time was set to more than three seconds.

Experimental Results

In each experiment, 10 trials or tests were performed and the number of passing results were recorded. It should be noted that in our conducted experimental procedures, a true or pass reading covers both true positive and true negative cases. On the other hand, a false or fail reading covers both false positive and false negative cases. Separate positive and negative tests were not conducted and are planned as a future work to further validate the reliability of the developed systems. Table 1 summarizes the overall results obtained from the three experiments conducted.

TABLE 1. Performance evaluation of Haar Cascade motion detection and OpenCV camera-based motion detection

S. No.	Input Face	Test Face	Was able to Detect	Result
1.	Arpit	Ram	No	100%
2.	Arpit	Arpit	Yes	100%
3.	Arpit	Chaitanya	No	100%
4.	Shreya	Ayushi	No	100%
5.	Shreya	Shreya	Yes	100%
6.	Ram	Chaitanya	No	100%
7.	Ram	Ram	Yes	100%
8.	Chaitanya	Chaitanya	Yes	100%
9.	Ayushi	Ayushi	No	0%
10.	Ayushi	Shreya	No	100%
Accuracy	9/10		90%	

The model was able to detect 9 out of 10 faces. One was not detected due to insufficient clarity inthe picture. We have tried to train and compare model with other libraries and with a few additions we believe this would predict 100% result with a few major changes. With machine learning we have developed the model and adding artificial intelligence can give it a great push.

Future Scope

Face Recognition is a way where we make a machine learn and recognize things and humans with you of programming and mathematical functions. Our aim is to develop a device in future thatwould be able to be installed in homes of people and they would be able to safeguard their thingswith this device/software.

An example is a system to identify known troublemakers in a mall or a supermarket to provide the owner a warning to keep him alert or for automatic attendance taking in a class. WE would try to make an device which would take data from the IOT cloud and would lock or unlock the door as expected from the model

REFERENCES

[1] M. Menon, "Innovating to zero crime rate," in Digital News Asia, ed, (2015).

[2] M. Pugh, J. Brewer, and J. Kvam, "Sensor fusion for intrusion detection under false alarm constraints," in 2015 IEEE Sensors Applications Symposium (SAS), (2015), 1-6.

[3] R. Sampson, "False burglar alarms," in Problem-Oriented Guides for Police: Problem-Specific Guides Series No. 5, U. S. D. o. J. Office of Community Oriented Policing Services, Ed., 2 ed, (2007).

[4] P. Vigneswari, V. Indhu, R. Narmatha, A. Sathinisha, and J. Subashini, "Automated security system using surveillance," International journal of current engineering and technology, vol. 5, no. 2, 882-884, (2015).

[5] S. Suresh, J. Bhavya, S. Sakshi, K. Varun, and G. Debarshi, "Home monitoring and security system," in 2016 International Conference on ICT in Business Industry & Government (ICTBIG, 2016), 1-5.

[6] H. U. Zaman, T. E. Tabassum, T. Islam, and N. Mohammad, "Low-cost multi-level home security system for developing countries," in 2017 International Conference on Intelligent Computing and Control Systems (ICICCS, 2017), 549-554.

[7] S. Sruthy and S. N. George, "WiFi enabled home security surveillance system using Raspberry Pi and IoT module," in 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES, 2017), 1-6.

[8] R. R. Ragade, "Embedded home surveillance system with pyroelectric infrared sensor using GSM," in 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM, 2017), 321- 324,

[9] S. Prasad, P. Mahalakshmi, A. J. C. Sunder, and R. Swathi, "Smart surveillance monitoring system using Raspberry Pi and PIR sensor," in International Journal of Computer Science and Information Technologies, vol. 5, no. 6, 7107-7109, (2014).

[10] B. Kaur, P. K. Pateriya, and M. K. Rai, "An illustration of making a home automation system using raspberry Pi and PIR sensor," in 2018 International Conference on Intelligent Circuits and Systems (ICICS), 2018, pp. 439-444.