International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Home Security using Digital Image Processing

Mr. Madhusudhan M V[1], Gowtham Vinjamuri[2], Shaik Nawaz Sharif[3], T. Venkat Rahul[4], Vishwanath[5]

[1]Assistant Professor, Department of Computer Science and Engineering, Presidency University Bengaluru, Karnataka, India
[2, 3, 4, 5]Department of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

---

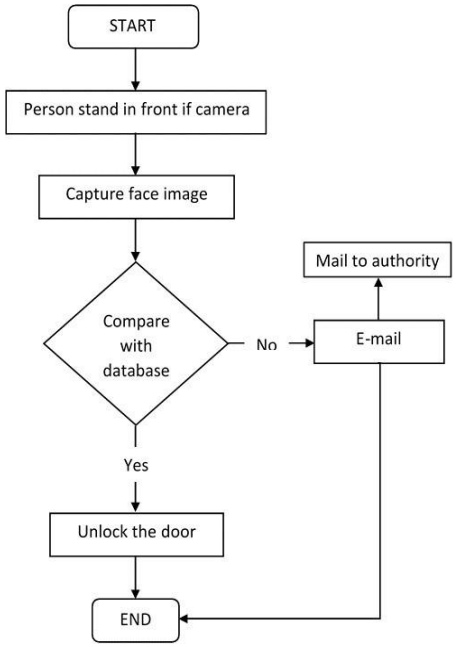
Abstract - Face Recognition is becoming a brand new trend within the security authentication systems. Modern FR systems will even observe if the person is real or not and by doing face recognition we can prevent the systems from being hacked by showing the image of a true person. I am sure, everybody questioned when Facebook enforced the tagging technique. It identifies the person and tags him/her whenever you transfer a picture. It's thus economical that, even when the person's face is occluded or the picture is taken darkly, it tags accurately. The eminent face recognition systems is backed by powerful deep learning algorithms. Let us explore one in all such algorithms and see how we are able to implement a true time face recognition system. FR can be done in 2 ways. Imagine you're building a FR system for associate enterprise. One way of doing this is often by coaching a neural network model, which may classify faces accurately. As you recognize, for a classifier to be trained well, it desires legion input data. Assembling that a lot of pictures of employees, isn't possible. The simplest manner of finding this drawback is by opting one-shot learning technique. It aims to find out info concerning object categories from one, or solely a couple of, training pictures. The model still must be trained on legion knowledge, but the dataset may be any, however of identical domain. The pre-trained model that we have a tendency to are progressing to use in Haarcascade with opencv.

Key Words: Face recognition, Auto tagging technique, KNN, One shot learning technique

1. INTRODUCTION

In these modern times, home security is the need of the hour for the development of society as a whole which in turn will help make our cities smart, so the concept of facial recognition to gain access of the house is an idea which is used to make our place of living more secure. A facial recognition system is a system which captures facial images and verifies the identity of a person using a digital camera. The human face assumes an essential part in our social association, passing on individuals' character. Utilizing the human face as a key to security, biometric confront acknowledgment innovation has gotten tremendous consideration in the previous quite a while because of its potential for a wide assortment of utilizations. A facial acknowledgment framework is a framework which gets facial pictures and confirms the character of a man using a propelled camera. It is an application fit for distinguishing or checking a man from a computerized picture. One approach to do this is by looking at chose facial components from the picture and a face database. As stood out from other diverse biometrics frameworks utilizing unique mark/palm print and iris, confront acknowledgment has unmistakable favorable circumstances due to its noncontact handle. Face pictures can be caught from a separation without touching the individual being recognized, and the ID does not require participating with the individual. It is normally utilized as a part of security frameworks and can be contrasted with different biometrics. It has additionally turned out to be main stream as a commercial recognizable proof and
advertising instrument

2. SYSTEM ARCHITECTURE



2.1 Generating Selfie training data using webcam

Writing a python script that captures images from Webcam and it detects face. Also the script draws a bounding box around each face.

If there are multiple faces, we are going to take the biggest face, crop it, then save it in the numpy array. We will ask the user to give the name of the person of whose face it is. We will create a 2dimensional matrix, which is an image (picture) itself and we are going to flatten it in the form of linear array and save it as a numpy file.

We are going to do it for multiple persons. First we will take images of person A, person B, person C, then we are going to store all these data in the numpy array and this will help us to get the training data.

We are going to generate real time training data by taking selfies from webcam using the python script. #Read video stream, show and capture images

import cv2 Import numpy

#Initialize camera (0 is the device id) Cap=cv2.VideoCapture(0)
#We are going to load haarcascade which is used for face detection
face_cascade=cv2.CacadeClassifier("haarcascade_frontalface_alt.xml")

For face detection, we need this object face cascade and we load the file Cascade Classifier. We will have a loop which will break when user presses q. #loop until user presses q while True:
key_pressed=cv2.waitKey(1)&&0xFF

cv2.waitKey(1) is a 32 bit integer and we are doing bitwise AND with an eight bit integer.
So, we get an 8 bit integer and we check if key pressed is equal to the ASCII value of button, in that case we have to break the loop, otherwise, we will get the return value and the frame by using the read method. We read what information we are getting from the webcam and now we have to check if this return value is false, we are going to continue, if due to any reason the frame is not captured.
When we have the frame then, we will call cv2.imshow("Frame",frame)

2.2 Detect faces and show boundary box

We are going to make use of face cascade object, and this object has one method which is called detect MultiScale Faces=face cascade detect MultiScale (frame, 1.3, 5) Faces will be a list of face.

One face will be a tuple containing (x, y, w, h) and the faces will be a list of tuples, tuples that are face.

x, y are the coordinates of the first point of the bounding rectangle. W is the width and h is the height. The list can be like [(10, 20, 30, 40), (20, 30, 40, 60)] where each tuple is a face.
#we are going to store every tenth face #to draw a bounding box for face in faces:

x,y,w,h=face cv2.rectangle(frame(x,y),(x+w,y+h), (0,255,255), 2) We are drawing a bounding rectangle at these coordinates of the face.

2.3 Storing every tenth frame from the video stream

For storing every tenth frame we will make an array

skip=0
Face data= [] #Here we are creating an array for storing the face data
We are going to store the frames at some particular location Dataset
path="./data/" if
(skip%10==0): pass

If there are multiple faces, then there would be sorting based on width and height

F= x,y,w,h indices 0,1,2,3 key= f[2]*f[3]
We will use lambda function to sort the faces based upon this key. We are going to do sorting based upon the area of the face. So, we write:

Faces=sorted(faces,key=lambda f:f[2]*f[3])

We are going to do sorting so that the largest face comes to the front of the list. While iterating through the faces, we can pick the last face, which is the biggest also. We can start from -1 to the end. We will pick the last face because it is the largest according to the area. #Pick the largest face For face in faces[-1:]:
x,y,w,h= face cv2.rectangle(frame(x,y),(x+w,y+h),(0,255,255),2)

#Extract (Crop out the required face, i.e the region of interest)

Offset=10 face_section=frame[y-offset: y+h+offset,

x-offset:

x+w+offset]
face_section is some part of the frame

face section = cv2.resize(face section, (100,100))

In frame by convention the coordinates are frame[y,x]

if skip%10===0: Face data.append(face section) print(len(face data) cv2.("Frame", frame) cv2.imshow("Face section", face section)

After every tenth frame we have increased the counter by one.

2.4 To flatten the faces.

We need to flatten our face image and save it in a numpy array.

#Convert our face list into a numpy array

face_data=np.asarray(face_data)

face_data=face_data.reshape(face_data.shape[0],-1)
Number of rows should be the same as number of faces. Number of columns should be figured out automatically.
print(face_data.shape)

Now we are going to save this data into file system np.save(dataset_path+filename+".npy",face_data) File name should be taken as input. We should ask the user to give the name of the person, whose face we are scanning.

There would be multiple files created in the data folder every time we run the script, one file will be created.

We save the file and we print one statement

print("data successfully saved at"+dataset+file_name+ ".npy")

3. WORKFLOW

You might be sensible at recognizing faces. You presumably notice it a cinch to spot the face of an admirer, friend or acquaintance. You're reception with their facial expression their Eyes, nose mouth and the way they're available along. That's however a biometric authentication system works, however on a grand, recursive scale. Wherever you see a face, recognition technology sees information. That information is also hold on and accessed. As an example, 1/2 all yank adults have their pictures hold on in one or additional facialrecognition databases. That enforcement agencies will search, in line with a Georgetown University study. So however will biometric identification work?
Technologies vary, however here area unit the essential steps:

- An image of your face is captured from a photograph or video. Your face may seem alone or in a very crowd. Your image might show you wanting straight ahead or nearly in profile.
- Biometric identification computer code reads the mathematics of your face. Key factors embody the space between your eyes and therefore the space from forehead to chin. the pc code identifies facial landmarks one system identifies sixty eight of them — That unit key to characteristic your face. The result: your facial signature.
- Your facial signature a mathematical formula is compared to an info of notable faces. And contemplate this: a minimum of 117 million Americans have pictures of their faces in one or additional police databases. In line with a mighty 2018 report, the law enforcement agency has had access to 412 million facial pictures for searches.
- A determination is created. Your face print might match that of an image in a very biometric identification system info.

4. RESULTS AND DISCUSSION

Fig -1: Test result

Among the various biometric techniques, biometric authentication might not be the most reliable and efficient but it has several advantages over the others: it's natural, easy to use and doesn't require aid from the test subject. Properly designed systems installed in airports, multiplexes, and other public places can detect presence of criminals among the crowd. Other biometrics like fingerprints, iris, and speech recognition cannot perform this type of mass scanning. However, questions are raised on the effectiveness of biometric authentication software in cases of railway and airport security. Critics of the technology complain that the London Borough of Newham scheme has never recognized a single criminal, despite several criminals in the system's database living within the Borough and also the system having been running for several years. "Not once, as far because the police know, has Newham's automatic biometric authentication system spotted a live target." Despite the successes of the many systems, many issues remain to be addressed. Among those issues, the subsequent are prominent for many systems: the illumination problem, the pose problem, scale variability, images taken years apart, glasses, moustaches, beards, low quality image acquisition, partially occluded faces, etc. Figures below show different images which present a number of the problems encountered in face recognition. A further important problem, on top of

the pictures to be recognized, is how different face recognition systems are compared.

5. PERFORMANCE ANALYSIS

There is no training involved. All the time is spent within the query time. So, all the work happens at query time.

Training Time is O(1).

All work is at query time. Total query time: Time in computing distance from all the points = O(n) time in sorting distances from all the points = O(log n). Therefore total query time = O(n log n) If we've got q number of queries, then time complexity is = O(N*q). KNN could be a non-parametric algorithm, we are not visiting learn any parameter because learning is 0. Problems to notice: Image Intensity This image is utilized initially.

However, the face_location() operate cannot appear to retrieve any faces. In this case, there square measure two choices that you'll be ready to use to reinforce the situation: Change the image‘s filter intensity.

At first, I try to regulate the image filter to vapor bath with the intensity fifty that the operator will later be accustomed to extract face descriptors. The effects square measure solely applicable with our dataset, and this technique isn't general for every things.

6. CONCLUSION AND FUTURE WORK

This project describes home security using digital image processing algorithms. It explains the technologies utilized in the project and also the methodology used. Finally, it shows the results, discuss the challenges and the way they were resolved followed by a discussion. Using Haar-cascades for face detection worked extremely well even when subjects wore spectacles. Real time video speed was satisfactory as well empty of noticeable frame lag. Considering all factors, LBPH combined with Haar-cascades can be implemented as a value effective face recognition platform. An example may be a system to spot known troublemakers in a very mall or a supermarket to supply the owner a warning to stay him alert or for automatic attendance taking in a very class.

Future work

1. If a blacklisted person tries to open the door, the system will send a message to the admin using GSM module regarding the identical.

2. A real time speaking assistant may be deployed to form the system more user -friendly and efficient.

3. Highly secure protocols like TLS may be deployed to confirm there's no security breach.

7. REFERENCES

1. Januzaj, Y., Luna, A., Ramaj, V. 2015 Real time access control based on Facial Recognition.
2. Lwin, H., Khaing, A., Tun, H. 2015.Automic door access system using face recognition.
3. Chowdhury, M., Nooman, S. 2013. Access Control of Door and Home Security by Raspberry Pi through Internet.
4. Senthikumar, G., Gopalkrishnan, K., Sathish Kumar, V. 2014 Embedded Image Capturing System Using Raspberry Pi System.
5. Çarıkçı, M., Özen, F. 2012 A Face Recognition System Based on Eigen faces Method.
6. Jogdand, S., Karanjkar, M. 2015Implementation of Automated Door Accessing System with Face Design and Recognition.
7. Sowmiya, U., shafiq mansoor, J. 2015 Raspberry pi based home door security through 3g dongle.
8. Kartik J. Srimadhavan V. 2013 SMS Alert and Embedded Network Video Monitoring Terminal. [9] Sahani, M., Nanda, C., Sahu, A., Pattnaik, B. 2015Web Based Online Embedded Door Access Control and Home Security System Based on Face Recognition,.
9. Mulla,,M., Patil, R. 2015 .Facial Image Based Security System using PCA.
10. Gubbi, Jayavardhana,. (2013) Internet of Things (IOT): A vision, architectural elements, and future directions. Future Generation Computer Systems 29.7: 1645-1660. [12] Raspberry Pi Foundation [Online] http://www.raspberrypi.org/downloads/.