IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

FACIAL RECOGNITION SYSTEM USING OPENCV

Vivek Anand[1], Vimal Singh Parihar[2], Shubham Kumar Sharma[3], Vikas Singhal[4] and Pramod Kumar Sethy[5]

[1]*Department of Computer Science and Engineering, Krishna Engineering College, India*
[2]*Department of Computer Science and Engineering, Krishna Engineering College, India*
[3]*Department of Computer Science and Engineering, Krishna Engineering College, India*
[4]*Department of Computer Science and Engineering, Krishna Engineering College, India*
[5]*Department of Computer Science and Engineering, Krishna Engineering College, India*

Abstract

The main objective behind the facial recognition system is the certitude that every person has a unique face. As we know that every person has unique fingerprint, similarly every individual face have unique features. Here we use features of face of an individual. We can stored the features of the faces of many individuals and they can be identified according to their face features. Facial testimony and facial recognition are difficult and challenging piece of work. For facial recognition systems to be authentic, they must work accurately and precisely. The facial recognition technique captured the image using the camera and mapping it for comparison to the images stored in the database. If the captured image is matched with any of the stored images then it shows face matched otherwise it shows face not matched. This paper elaborate in detail the entire process of facial recognition system using OpenCV library. We use Haar cascading algorithm using OpenCV library for the face detection.

Keywords: Face detection, face recognition, Haar cascade, OpenCV.

1. INTRODUCTION

In the last two decades, the facial recognition system has become one of the most important and interesting research areas. A facial recognition system is a software application for certifying an individual and recognizing him/her with images or videos from a source. Facial recognition can be done speedily and accurately with the open-source platform called OpenCV. A path from a face and a picture database are favorite facial features. It is usually compared to biometrics such as fingerprints and eye investigation systems, and security systems and used in thumb detection systems. The OpenCV library makes programming easy to use. This comes up with advanced proficiencies like face detection, face tracking, facial recognition, and many more methods for artificial intelligence (AI). The main advantage of the OpenCV library is, it is a multi-platform framework; it supports Windows, Mac OS, Mac OS X.

Its challenging work includes face recognition on the lowest computing cost framework such as smartphones and embedded devices. For the person's testimony, facial recognition is used. Everyone has unique features that do not share with another person.

Currently, there are many devices and application which use face detection technology to recognize and detect a face such as Facebook. Therefore, face detection is not new in the vision of computer science. In this letter, we have taken our research using Open CV. Reasons for using OpenCV are discussed further in this paper.

2. REASONS FOR USING OPENCV

(a) Speed- OpenCV uses C/C++ library functions which provide the computer with machine language code and help in faster execution, use of OpenCV results in more use of time and resources in image processing and less in explaining.

(b) Portability- As OpenCV implements on C, therefore any devices which run on C can run OpenCV. It can work well with Windows or Linux.

(c) Cost- OpenCV is free for all because it is a BSD license so it is free of cost.

3. RELATED WORK

There are five basic steps involves in a facial recognition system includes image capture, face detection, feature extraction, comparison, and face recognition. Face recognition technology analyzes the unique shape, pattern, and positioning of facial features. Face recognition is a very complex technology and is largely software-based.

It holds the record for identifying the captured image. Various features like nose shape, eye shape, lips, skin color, skin tone, etc. The face recognition system first captures an image from a camera as input and finds the face in the image for face detection. Face extraction includes obtaining the features from the face captured by the camera. Then it compares the features with the features of images stored in the database and according to which it gives result. If the features are matched with stored image features then it identified the person otherwise not identified. However facial recognition system still has few drawbacks as it cannot detect the face due to overlapping of faces or difficult recognition of two faces having the same features.

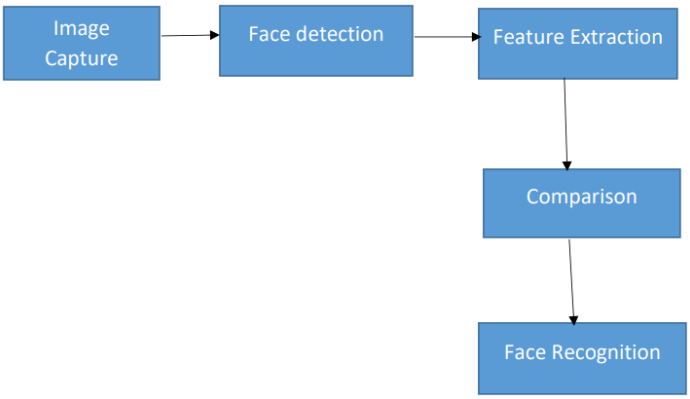


Fig 1. Steps of Facial Recognition System

3. FINDING FACES

Finding faces is the most essential part of face detection. There are different techniques by which faces can be found. In this paper, we will compare the various algorithms used before and analyzing them. Face detection is the most important step of a facial recognition system still the technique and algorithm used to implement it need to be improved. The accuracy of facial recognition systems depends on face detection due to this face detection is the main part of the entire process of the facial recognition systems

3.1 FINDING FACES VIA COLOR

A. IMAGES HAVING A DEFINITE BACKGROUD

One process is to find images in which we have a definite background containing only grayscale pixels. These images have a narrowband wavelength. While using these images when we remove the background from the foreground we get facial boundaries. This is the easiest method for face detection.

B. IMAGES HAVING A COLORED BACKGROUND

For colored image face detection is based on two procedure

B.1. BY APPLYING A SKIN FILTER

A skin filter is applied to detect skin. The texture of the image part that is being masked is defined by the skin filter process. The output generates contains distinct areas of human skin. Dilation and erosion are the techniques used to develop this kind of filter.

B.2. BY HAULING OUT THE FEATURES WHICH ARE BEING MASKED

During this step, very dark and very bright portions are removed from the image. By removing these portions, we get the most appropriate area covered by the skin for face detection. The major problem in this step is the light sources at the time of facial detection from the camera. Always the camera doesn't need to be placed under the sunlight some might have been placed in the average light or low light area.
To solve this problem, the input should be in RGB format with good intensity in the range of 0 to 255.

C. IMAGES HAVING A COMPLEX BACKGROUND

Face detection in the complex background can be done using the MUHULANOBIC metric. It is based on the detection of human faces in two-dimensional natural images. It discretely makes use of the process of a color fraction of the image that is being taken as the input. This fraction of the color is being performed by sill the image in the tone color space. It is taken care of the effects of distinguishing the color in the human skin when the lighting has been changed in the image. Then the results of the image are collected together for any other examination. At the end the difference between the faces and the remaining complex background, a multi-layer perceptron neural network is used with the invariant moments as an input factor.

3.2 FINDING FACES VIA MOTION

3.2.1 USE OF BLINK DETECTION TECHNIQUES

Blinking is a reflexive act conducted by humans. This is a very hurry process. Some humans might not even take care of blinking in daily life but blinking as a process has been proved to detect the presence of a human significantly at any frame of time. Blinking provides a casual time and space signal which is unique to every other individual. Therefore, the blinking process can identically act as a biometric means of measurement to detect the presence. An algorithm is used to make blinking make sense to a computer. This algorithm includes taking two images of a person simultaneously and removing the second image from the first image. This removal causes a discrete boundary outside the head and if in one of the images the eyes are blinked and there seems to be a little circled region at the eye portions. To this removed image a connected component procedure is represented. For the detection of the blinked image, there should be horizontal and vertical bounded regions. These regions mark horizontal and vertical segmentation.

3.3 FINDING FACES IN LIMITED AREAS OF PIXELS

In the images which are abandoned it is difficult to detect the faces but various methods have now been introduced such as edge detection orientation, weak classifier cascades. Edge orientation matching is a technique treated as a template matching procedure. This includes object modeling based on edge orientations. Various templates are created and matched to the image to detect the edges of the face. It takes approximately less than 0.08 seconds.

3.4 FACE DETECTION USING HAAR CASCADE

Haar cascade makes use of the image subtraction philological process for face detection. In this, the cascades of distinct images of the same person are taken and recorded in the database. All the pixels in the effect of the white region are removed from all the pixels in the effect of the black region. This technique of subtraction is performed on every image in the cascade but all the images might not give us the optimal results. Many of the images have a lot of errors. The image with the minimum error is selected. The result of all the images is added together and is referred to as a weak classifier. All the weak classifiers are added together to form a strong classifier. Applying the subtraction process and determining each image error is a very time and space-consuming process. Instead of applying it on each of the images, subtraction is applied to images one by one. If the last image is not useful it is discarded. Haar Cascade is an algorithm for face detection where many

positive and negative images are used to train the classifier. Positive images are those images which we want to identify. Negative Images are those images which contains useless things.

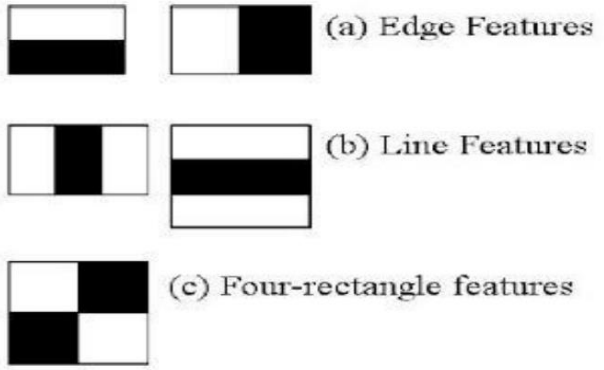


Fig 2. Haar Cascades Feature

4. OPENCV STRUCTURE AND CONTENT

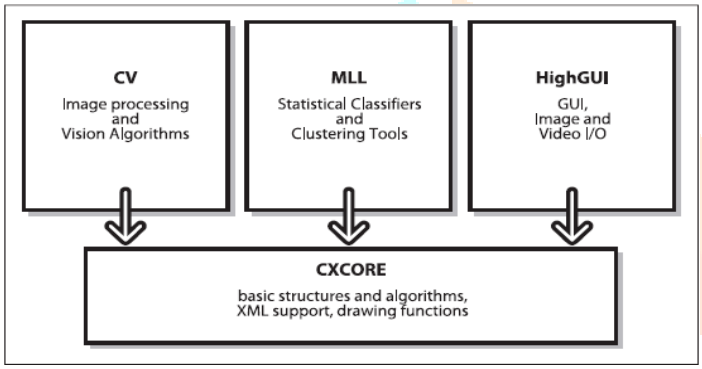


Fig 3. Structure and content of OpenCV

A. CV- This portion includes image processing and vision algorithm.

B. MLL- This portion includes statistical classifier and clustering tools.

C. High GUI- This portion includes GUI, image and video I/O.

D. CXCORE- This portion includes basic structures and algorithm, XML support and drawing functions

5. USE-CASES

A. PREVENT RETAIL CRIME- Face recognition is currently being used to instantly identify when a known thief, retail criminals, or people with a history of fraud enter retail stores. According to our data, face recognition reduces external shrinkage by 34% and, more importantly, reduces brutal events in retail stores by up to 91%.

B. FIND MISSING PERSONS- Face recognition system is used to find missing persons. In fact, once in India, approximately 3000 children were discovered in just four days with the help of a facial recognition system.

C. FORENSIC INVESTIGATIONS- Facial recognition can help in forensic investigations by recognizing persons in security images or other videos. Face recognition software can also be used to detect dead or unconscious individuals at crime scenes.

Gen with 4GB of memory running Windows 10 Home. We used Pycharm IDE and Python 3.9.1 and OpenCV 4.5.1.48 installed on my system. The Haar Cascade data file is already provided by the OpenCV library.

6. EXPERIMENTAL SETUP

6.1 REQUIREMENT

1) Any operating system that will support OpenCV and Python (Windows, Linux, MacOS)
2) Python
3) OpenCV-Python
4) Haar Cascades Data File
5) i3 or higher core processor (CPU)/ 2.1 GHz or higher
6) Photo/images for testing

We used an HP Laptop (15-bs1xx) with a CORE i5 Intel processor 1.8 GHz of 8th Gen with 4GB of memory running Windows 10 Home. We used Pycharm IDE and Python 3.9.1 and OpenCV 4.5.1.48 installed on my system. The Haar Cascade data file is already provided by the OpenCV library.

In our project, we applied face detection to some photos using OpenCV with Python. OpenCV is an open-source software library used for computer vision applications. The version we used of OpenCV for Python called OpenCV-Python because we developed our project in Python.

We implemented a system for detecting faces in digital images. These are in JPEG format only. Face detection uses classifiers algorithms that detect the faces in an image. It has been trained to detect the face using many images for more accuracy. OpenCV uses two classifiers, LBP (Local Binary Pattern) and Haar Cascade classifiers. We use the Haar Cascade classifier. We already discussed it above.

6.2 FEATURE EXTRACTION-

Process of feature extraction in the Haar Cascade algorithm

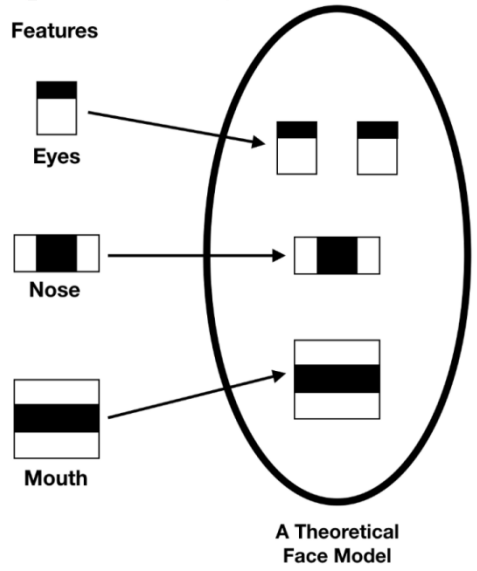


Fig 4. Feature extraction

The training data we used is an XML file called: haarcascade_frontalface_default.xml

6.3 RUNNING OPENCV

We prepared a directory where we stored all the files needed. You will need to put in this directory the following:
1) facial_recognition.py (the name we gave to our program that contains code. This name can be changed.)
2) haarcascade_frontalface_default.xml (Haar Cascade training data)
3) Images

WE ARE GOING TO USE THE

1. detectMultiscale: Module from OpenCV to create a rectangle with coordinates (x,y,w,h) around the face detected in the image.
2. scaleFactor: The value shows how much the image size is reduced at each image scale. A small value uses a smaller step for downscaling. This allows the algorithm to find the face.
3. minNeighbors: It specifies how many "neighbors" each applicant rectangle should have. A larger value results in small detections but it detects higher quality in an image.
4. minSize: The minimum image size. By default, it is (30,30). A Smaller face in the image is best to adjust the minSize value lower.

6.4 IMPLEMENTATION DETAILS

Facial recognition has three modules. Each module has been developed by a collaborative effort of all members.

1) Detector: It is responsible to detect faces from the images captured by a webcam.
2) Data Set Creator: It is responsible to create the data set of images.

3) Trainer: It is developed to train the software to detect faces.

6.5 IMPLEMENTATION FLOW

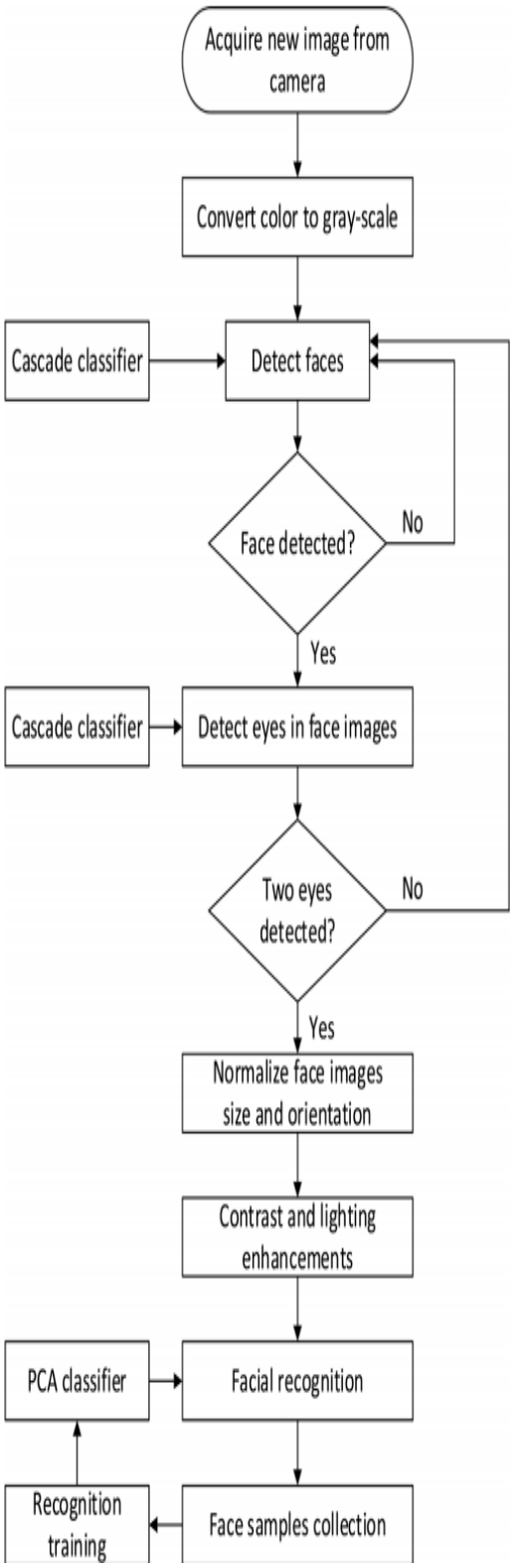
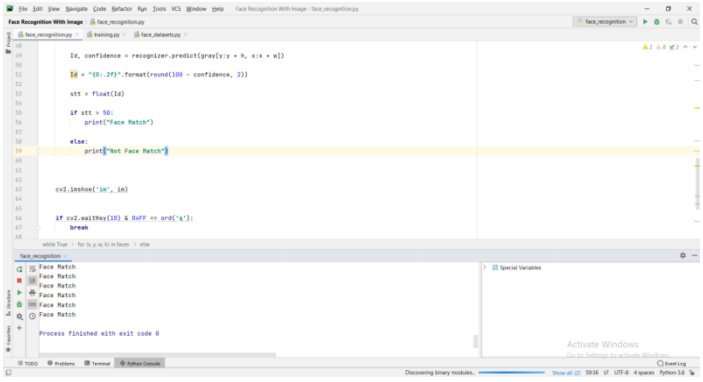


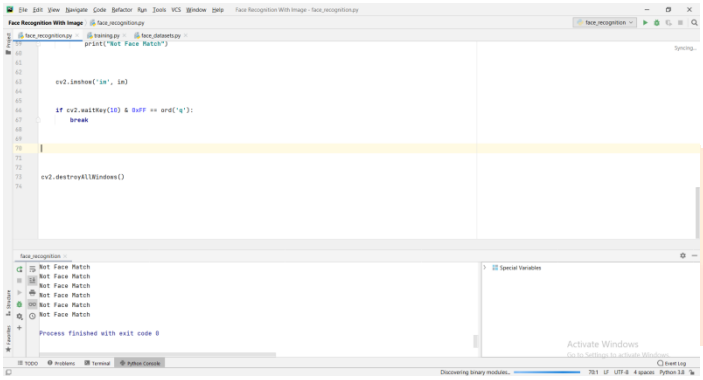
Fig 5. Implementation Flow

7. EXPERIMENTAL RESULT

7.1) If the captured image features matched with the image stored in the database then it shows 'Face Match'



7.2) If the captured image features are not matched with the image stored in the database then it shows 'Not Face Match'



8. CONCLUSIONS

This paper in detail explains the development of a facial recognition system using OpenCV. We discussed the advantages of using the OpenCV library in computer vision. The process of facial recognition with the Haar Cascade algorithm can detect and recognize the face. The facial recognition process with the Haar Cascade and can be successfully performed at a distance of more than 200 cm using a webcam. For future work, while the current system is for straight faces, it can be improved to recognize faces at different angles. Followed by the optimization of the facial recognition process for use on a small mobile device.

9. REFERENCES

[1] Learning OpenCV –Computer Vision with the OpenCV Library O'Reilly Publication.

[2] Learning OpenCV: Computer Vision with OpenCV Library, Kindle Edition. Gary Bradsk1 and Andrian Kehlar

[3] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces", IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.

[4] "KyungnamKim" Face Recognition using Principle Component Analysis"

[5] Codacus: https://youtu.be/1Jz24sVsLE4

[6] G B Huang, H Lee, E L. Miller, "Learning hierarchical representation for Face verification with convolution deep belief networks[C]",Proceedings of International Conference on Computer Vision and Pattern Recognition,pp.223-226,2012.

[7] Computer Vision Papers, http://www.cvpapers.com

[8] Learning OpenCV: Computer Vision with the OpenCV Library 1st Edition, Kindle Edition

[9] OpenCV Homepage http://opencv.willowgarage.com

[10] Recognition Homepage http://www.face-rec.org/algorithms.

[11] Paul Viola, Matthew Jones Conference paper- IEEE Computer Society Conference on Computer Vision and Pattern Recognition."Rapid object detection using a boosted cascade of simple features."