

Face Recognition using Open CV

Pranav Verma-

**Department of Computer Science
and Engineering SRM Institute of
Science and Technology
(pd1395@srmist.edu.in)**

**P. Renukadevi-Assistant Professor
Department of Computer Science and
Engineering SRM Institute of
Science and Technology
(renukadp@srmist.edu.in)**

HiraSohail-

**Department of Computer Science
and Engineering SRM Institute of
Science and Technology
(hm4707@srmist.edu.in)**

Abstract— Face recognition by far has been coded using java and MATLAB. Face recognition has been using a 2-D structure by far which makes it easy to crack security. So, nowadays the 3-D Structure has been trending. Earlier times we used a photograph to crack passcodes, but now it's not possible anymore. The Face recognizing pattern involves - Face detection, Data gathering, Data comparison, and Face recognition. Python is a platform-independent language which made it easy to integrate and collaborate with this as well as other software such as Raspberry pie. The software has used open CV which is being used for other facial recognition software. This software takes several minutes and captures multiple photos until a certain specified amount of time. We used the HAAR cascade classifier to detect faces, different types of HAAR cascade classifiers are available to recognize different facial features such as to detect eyebrows, smile, nose, Side-face, Frontal face, etc. HAAR classifiers are available in the XML file which is used in our project to detect the frontalface. In this project we used LBPH recognizer to train our photos, this data is converted to a YML file which is saved in the same folder using PICKLE. With of help of the LBPH recognizer, we extract this YML file which we have drained to detect the face. The recognizer has CONFIDENCE MATRIX. The CM has defined 60% – 90% as the boundary value for face matching, or else the face is labeled as 'unknown'.

KEYWORDS: OPENCV, HAAR CascadeClassifier, LBPH Recognizer

INTRODUCTION-

The face detection and the face recognition have been common in the research field like computer vision and artificial intelligence. It is the

application used to analyse the picture and to create an understanding towards algorithm. A general statement on face recognition can be expressed as:

given the images or video images, it identifies and verifies one or more than one faces in the image using a database that contains different images of faces stored in it.

The face recognition involves four stages while completing the process:

Detecting the Face- The technology used to identify human faces in digital images. It is used in various applications. In this process, an image is searched to find a face, then that is processed to crop the facial part and select the person's face for better recognition.

Gathering the data- is the systematic way to collect and measure information on variables of interest, from the dataset.

Comparing the data- it is used to calculate and display the similarities and differences between the data objects.

Recognizing the Face- it is used where the detected face is found and is compared to the database which contains images of the faces which are known to the database and recognize the person.

Face detection is performed using OpenCV. It is an open-source framework developed by Intel. The framework works in 90-95% of clear photos. The clear photos are taken of the person who is in front of the camera using a face detector which is inbuilt. However, if we view the person from an angle, it is tough to detect the face.

OpenCV- is a multi-platform framework. Windows, Linux, and Mac OS X. The key to good results is the way this method works. It also helps in understanding the outcomes. The functionality of OpenCV is contained in several modules and used for face recognition.

CV Namespace- methods such as image processing and camera calibration are included in this module. It also includes computational geometry functions.

HAAR Cascade- it is a classifier which uses machine learning approach to train the images as positive and negative.

- **Positive images** – it contains the part of images which we the classifier identifies as recognized.

- **Negative Images** – it contains the remaining part of images which the classifier does not recognize.

LBPH-

The recognizer is the solution of the face recognition. It recognizes the front face as well as the side face. The LBPH algorithm defines a recognition rate which is decreased under the circumstances of heterogeneity, the variation in expression, and the transformation. For the solution, a pixel gray

median is proposed. The median value of the neighbour's sampling value is replaced by the value of the pixel. The sub blocks are used to extract the value of the feature and the histogram is formed, which is used to recognize the face with the input image.

Confidence Matrix- CM has defined 60% – 90% as the boundary value for face matching, or else the face is labeled as unknown.

GUI Namespace- is used to implement basic input-output interfaces and gives multi-platform tasks.

RELATED WORK-

Most of the paper used for the reference purpose in this project is based on how to handle the databases containing the images so the faces can be detected easily.

The following are the results of the research carried out for face detection and recognition.

[1] A popular open-source relational data base, known as MariaDB Server used by MYSQL developers. It is used in distributions of Linux. This database is built on stability and performance. It includes clustering and features compatibility using OracleDB.

[2] 3-D face model is used in computer vision. It helps in providing the invariance illumination with the help of modeling of faces and the images which are processed. Using this 3D model, the images can be fit to 2D and 3D images. In the result, the parameters highlight lighting, pose and imaging features.

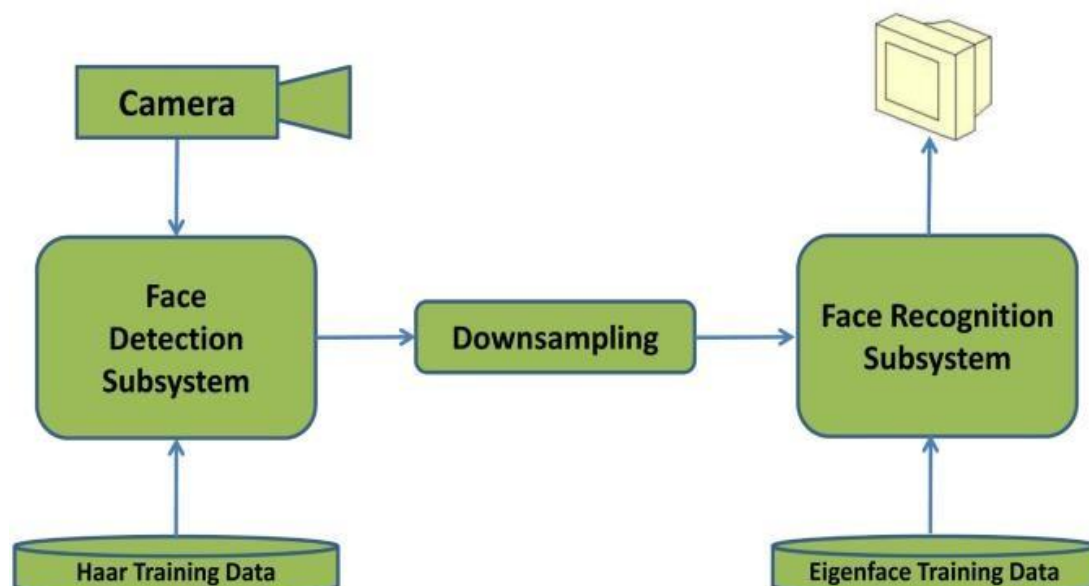
[3] The conversion of scanned documents to the encoded machine level text, processed electronically is termed as Optical Character Recognition. It is used to produce data from books, first they are digitized then the resulting images are processed by OCR engine. It solves the following problems-

Recognize portraits on the page, extract people's names from OCR results, correlate extracted names with portraits, predict people's age.

[4] Cross Platform Computer Vision Applications, uses a software, which is implemented on various platforms of multiple computing. These platforms are divided into two types- One is building and compiling the individual platforms which the application supports. Second is the software can be implemented on any platform without any kind of special precautions. The platforms can be a programming environment or an operating system.

[5] Complex Environment and Computational Technology. In today's computer life, the general purpose of computing in units processing of graphics. A drastic change can be seen when programming of computers is done in parallel ways.

ARCHITECHTURE DIAGRAM-



IMPLEMENTATION-

When an image is taken for detection and recognition, a plenty of factors are identified that prominent the accuracy of the system. The various image pre-processing techniques are applied to regulate images which are being supplied to the system.

OpenCV uses a facial detector type known as the HAAR Cascade classifier. It takes an image coming from a file or a live video, and it examines the location of each and every image and classifies it as a recognized face or not. The data stored in the XML file is used by the classifier to decide the location of the images.

ALGORITHMS USED-

The algorithm used is **PRINCIPAL COMPONENT ANALYSIS USING**

EIGENFACES. It consist of four sub algorithms. The two algorithms are used for implementing facial features for face detection, which are characteristics of color and geometrical behavior. The next algorithm contains an arrangement which has predefined features, and when it is searched it will give the image and compares the relation between the predefined arrangement and the image to be searched for. The last algorithm is used to train a classifier which is using samples of different faces, and then detects the face using it.

These different algorithms are used for face detection.

The images which are used as input and the face resolution found is around 200×200 pixels in the images. The face color of s taken as the input in the form of a binary image, then the counting of the

number of pixels that belongs to the familiar domain is done using the labeling algorithm. The domain acquires an indicating signal that the face present in the domain is similar to face given as input in size.

In the second layer, PCA with SVM is used which is based on machine learning. This helps in enabling the layer to locate the face domain directly and SVM is used for detecting the face.

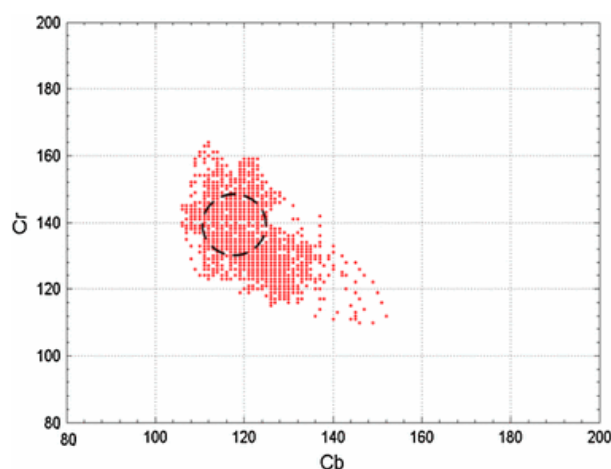
RGB to YCbCr Conversion-

The formula for converting RGB to Y, Cb, Cr is based on a linear equation containing-

$$Y=(66R+129G+25B+4096)\gg 8 \quad Cb=(112B-38R-74G+32768)\gg 8 \quad Cr=(112R-94G-18B+32768)\gg 8$$

The R, G, and B are the components of RGB888. RGB565 is transformed to RGB888 using $\gg 8 \gg 8$, which is a shift operation of 8 bits rightwards.

A face skin color model is obtained. We can also use the Gauss Model or the Statistical HistogramModel. A skin color model is used to explain the cluster characteristics of skin samples-



Eyes' Graylevel Complexity Calculation:

The sum of the gradient values of the pixels which are connected in a direction is known as Gray level complexity. It reflects the degree change of the image in adirection.

To calculate gray level complexity for horizontal direction-

$$\Delta x = \sum_{y=1}^{M-1} |p(x, y+1) - p(x, y)| \quad x \leq N, y \leq M$$

is used. Where,

$p(x, y)$ - gray pixel which is located in the xth row and yth column.

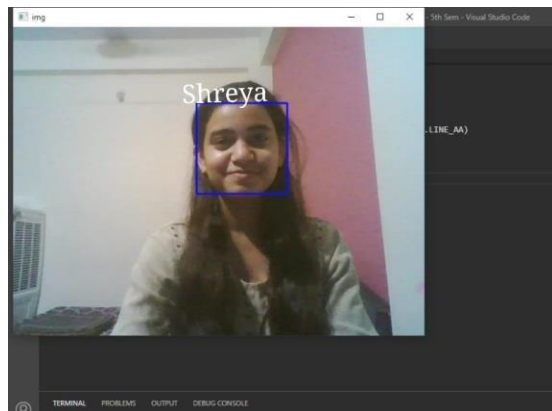
N- the number of rows

M- the number of columns

PSEUDO CODE-

```
import pickle import cv2 import numpyface_d =  
cv2.CascadeClassifier(r"F:\Project\haarcascade_frontalface.xml")  
recog = cv2.face.LBPHFaceRecognizer_create() recog.read(r"F:\Project\Trainer.yml")  
  
lab = {}  
with open(r"F:\Project\labels.pickle", 'rb') as f: lab1 = pickle.load(f)  
lab = {i:j for j, i in lab1.items()}  
rec = cv2.VideoCapture(0) while True:  
pic, image = rec.read() gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
f1 = face_d.detectMultiScale(gray_img, 7, 6.5) for (a,b,c,d) in f1:  
g1 = gray_img[b:b+d,a:a+c] g2 = image[b:b+d,a:a+c]  
c_id, confere = recog.predict(g1) if confere >= 45 and confere <= 85: print(c_id)  
print(lab[c_id])  
font = cv2.FONT_HERSHEY_SIMPLEX k = lab[c_id]  
color = (255, 255, 255)  
stroke = 3  
cv2.putText(image, k, (a, b), font, 1, color, stroke, cv2.LINE_AA)  
cv2.rectangle(image, (a, b), (a+c, b+d), (255, 0, 0), 3) cv2.imshow('image', image)  
if cv2.waitKey(33) & 0xFF == ord('q'): break  
rec.release() cv2.destroyAllWindows()
```

RESULT





To improve the performance of facial detection and recognition, we can improve many things from which some are easy to implement.

Example- adding processing of color, detecting the edges.

For the classifier for recognizing person's face in various lighting conditions and different angles, a lot of variation is important, instead of specific conditions. But the set of facial pictures found for the same person should not vary a lot. For example, if we rotate some images by 90 degrees. This will give bad results as it would make the classifier to be too generic.

So, if we take a set of images which are too varied, then we have to create a separate set of training images for each person. And the images should be aligned properly, because in many cases, we can use low resistance images which gives better results than high resistance images. There is not much of a match if the images are aligned properly, and the images that are to be detected are brighter than the images that are trained.

Face recognition is easy nowadays, if we are training an image of a person and then processing it to recognize. The expressions and the lighting of the image and the directions in which image is being viewed will be same if the camera and background features are same.

In this case, we will get good recognition results. But if we try to recognize the images from a different direction it will not give better results. For good results, a lot of experimentation is required. The software will take several minutes and capture multiple photos in a certain amount of time. The task is to find the name of the image detected in the database and to find the location and different size of the objects belonging to the same class in an image.

Given an image, tell whether there is any human face if it detects the name, it shows a box on the face with the name of the person, if the image is not detected it simply shows an image not found.

BENEFITS OF PROPOSED SYSTEM

Filtering-

When the image is filtered, it is done with the operation, which is commonly known as Convolution. Convolution is called as Neighborhood operation. In this operation, pixel used for output is calculated using the sum of all the neighbor pixels which is present at input.

Filing-

Inserting an image and deciding what the form of this image is.

Color mode conversion and skin detection-

It deals with skin-color pixels and forms the regions in the picture. There are three parameters which helps in recognition of pixels present in a skin. These are - Red Green Blue (RGB), Hue Saturation Value (HSV) and Luminance and Chrominance (YCbCr).

Dilation-

In a morphological operation, dilation is used for enhancing facial features. It is function which contains inputs. First input is the picture which is to be dilated, second input is a structure element which is a 2-D structure. Dilation is used for exaggerating facial features too.

Erosion-

In morphological operations of processing of images, there are two fundamental operations, from

which one of them is erosion. Erosion is defined as binary images. It is extended to images for grayscale and lattices to complete.

Elimination of non-face region-

Eliminates the part that is not required in detection.

Ratio technique –

An algorithm of segmentation of an image is developed. The region which is segmented has one boundary cost associated with the exterior. And benefit associated with the interior.

CONCLUSION

To improve the performance of facial detection and recognition, we can improve many things from which some are easy to implement.

Example- adding processing of color, detecting the edges.

For the classifier for recognizing person's face in various lighting conditions and different angles, a lot of variation is important, instead of specific conditions. But the set of facial pictures found for the same person should not vary a lot. For example, if we rotate some images by 90 degrees. This will give bad results as it would make the classifier to be too generic.

So, if we take a set of images which are too varied, then we have to create a separate set of training images for each person. And the images should be aligned properly, because in many cases, we can use low resistance images which gives better results than high resistance images. There is not much of a match if the images are aligned properly, and the images that are to be detected are brighter than the images that are trained.

Face recognition is easy nowadays, if we are training an image of a person and then processing it to recognize. The expressions and the lighting of the image and the directions in which image is being viewed will be same if the camera and background features are same.

In this case, we will get good recognition results. But if we try to recognize the images from a different direction it will not give better results. For good results, a lot of experimentation is required.

The software will take several minutes and capture multiple photos in a certain amount of time. The name of the image detected in database is the task. Given an image, tell whether there is any human face if it detects the name, it shows a box on the face with the name of the person, if the image is not detected it simply shows an image not found.

REFERENCES

- [1] Wechsler H, Phillips JP, Bruce V, Soulie FF, Huang TS (eds) (1998) Face recognition: from theory to applications[M]. Springer, New York.
- [2] Rowley HA, Baluja S, Kanade T (1998) Neural network-based face detection. Pattern Anal Mach Intell IEEE Trans20(1):23–38
- [3] McCready R (2000) Real-time face detection on a configurable hardware system. Springer, Berlin

- [4] Nguyen D, Halupka D, Aarabi P et al (2006) Real-time face detection and lip feature extraction using field-programmable gate arrays. Syst. Man Cybern Part B
- [5] M. Gunther, D. Haufe and R.P. Wurtz. Face recognition with a disparity corrected Gabor phase differences. In Artificial neural networks and machine learning, volume 7552 of lecture notes in computerscience
- [6] Theodoridis T (2006) Embedded hardware face detection[D], The Pennsylvania State University.
- [7] Akrouf, S., Sehili, M. A., Chakhchoukh, A., Mostefai, M., & Youssef, C.(2014, June-July). Face Recognition using Principal Component Analysis with DCT. The International Journal of the Eurasian Journal of Science &Engineering
- [8] M. Turk and A. Pentland for a recognition journal of cognitive neuroscience,3(1):71- 86,2017.
- [9] R. Duda,P. Hart,D. Stork, “Pattern Classification”,ISBN 0-471-05669-3,pp. 121- 124.
- [10] J. Ashbourn, Avanti,V. Bruce,A. Young, “Face RecognitionBasend On Symmetric AndEignfaces