FACE DETECTION AND RECOGNITION USING OPENCV

Suganthi.A, Poongothai.K, Prashanthini.K, Sangeetha.C

Assistant Professor, Department of Computer Science and Engineering,
KGISL Institute of Technology, KGISL Campus,
Saravanampatti, Coimbatore – 641035, Tamilnadu, INDIA.
Ph. No: +91-7092745697, E-mail: sugantit@gmail.com
Ph. No: +91-9080959132, E-mail: poovi3521@gmail.com
Ph. No: +91-9994556707, E-mail: prashakrish14@gmail.com
Ph. No: +91-9789791905, E-mail:sanguit.8@gmail.com

ABSTRACT:

In recent time, Face detection and recognition from an image or a video could also be a popular house in natural science analysis. Face recognition technology has wide attracted attention because of its monumental application , like network security, criminal identification, smart advertising, etc. whereas face recognition has been around in one kind or another since the Sixties, recent technological developments have led to an oversized growth of this technology. We've got a bent to vogue face detection and recognition system supported Haar cascade classifier and LBPH face recognizer severally, by manner of OpenCV and Python programming. Our system includes four parts: Training data preparation, Face Detection, Train Face Recognizer and Prediction.

Keywords: Face Detection; Face Recognition; OpenCV; Classifier.

1. INTRODUCTION

Face is most ordinarily used biometric to acknowledge individuals. Face recognition has received substantial attention from researchers because of human activities found in varied applications of security like field, criminal detection, face following, rhetorical etc. Compared to utterly completely different biometric traits like palm print, Iris, finger print etc., face statistics unit non-intrusive. area unit going to be taken even whereas not user's information and extra are aiming to be used for security primarily based applications like criminal detection, face following, field security, and rhetorical investigation systems. Face recognition involves capturing face image from a video or from an investigation camera [1]-[2]. They are compared with the hold on information. Face natural science involves training data legendary footage, classify them with legendary classes then they are hold on inside the data. Once a take a glance at image is given to the system it's classified and compared with hold on information. Face natural science could also be a tough field of research with varied limitations obligatory for a machine face recognition like variations in head cause, modification in illumination, facial expression, aging, occlusion because of accessories etc,. Varied approaches were schooled by researchers in overcoming the constraints express. Automatic face recognition involves face detection, feature extraction and face recognition. Face recognition algorithms unit typically classified into a pair of classes as image guide primarily based [3] and geometric feature based [4]. The guide primarily based ways total correlation between face and one or further model templates to look out the face identity. Principal component analysis, linear discriminate analysis, kernel ways etc. unit used to construct face templates. The geometric feature primarily based ways unit used to analyze specific native choices and their geometric relations

(elastic bung graph method), except for these ways, multiple algorithms is used to discover and acknowledge countenance [8]-[9] and this technique is implemented in varied frameworks like Microsoft's .NET[5]

2. PROBLEM DEFINITION

The techniques utilized in face detection and face recognition systems would possibly rely on the appliance of the system. To say in abstract manner, it's classified in to a pair of broad categories. First, to look out a private within associate degree outsized datasets of faces (e.g., throughout a police database). This technique typically returns a list of the foremost likely people inside the data and returns one image as a result. Secondly, to identify express person in amount (e.g. throughout a security observation system, location pursuit system, etc.), or to allow access to entirely a bunch of people. (e.g., access to a building, computer, etc.) Multiple footage per person unit generally gettable for training data and amount recognition is required. In this paper, we've got a bent to own associate degree interest in first case and targeted on choices of OpenCV and to point what proportion this approach is easy and clear-cut inside the appliance areas of face detection and recognition.

3. FACE RECOGNITION

Face recognition system is comparable the tactic of but human brain identifies and acknowledges a private from a photograph or at streams. The sequences unit investigating face by perceptive the image, Extracting distinctive characteristics of the face, compare those choices with all people legendary and recognition of person. Based on this technique, our face recognition system consists of four modules: Training data preparation, Face detection, Train Face Recognizer and Prediction.

(1) Preparing training data:

The accuracy of face recognition defends on the number of images of a person given to recognizer as training dataset. The more the images, the better the accuracy is. By taking the importance of this principle, we trained the recognizer with different images of same person with different emotions, like happy, sad, crying, laughing, and even, with glasses, without glasses, with beard, without beard, etc. So, our training data consists of total of 50 people with 20 images for each person. All our training data is inside the folder: 'training-data'. This folder contains one subfolder for each person, named with the format: SLabel (e.g. S1, S2) where the label is the integer assigned to that person. For example, the subfolder called S1 means that it contains images for person 1. The directory structure tree for training data is shown in the Figure.1.

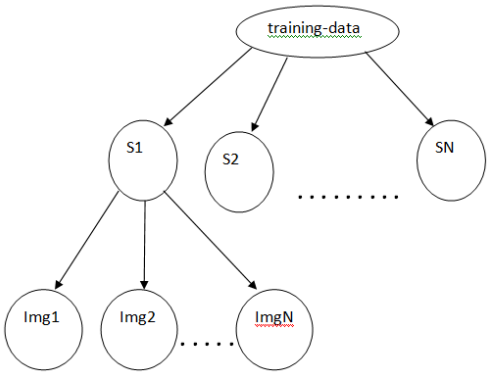

Figure.1: "Directory tree structure for training dta set"

The importance of training information preparation is to understand that face belongs to that person. OpenCV face recognizer accepts info during an explicit format. One is that the faces of all the folks. The second is that the whole number labels for every face. So, a bunch of pictures belongs to an individual given associate degree whole number label, and mapping is completed between these labels and faces.

(2) Face detection:

Face detection has been one among the favored topic in video analytics analysis. This technology has been accessible for a few years and is getting used by everywhere the place. Face detection is done by victimization classifiers. Classifier may be a computer virus that decides whether or not a picture may be a positive image (face image) or negative image (non-face image). A classifier is trained on many thousands of positive and negative pictures to be told the way to classify a replacement image properly. OpenCV provides USA with 2 pre-trained and prepared to be used for face detection classifiers: (i).Harr Classifier (ii). LBP Classifier. Each of those classifiers method pictures in grey scales, primarily as a result of we have a tendency to don't would like color data to make a decision if an image contains a face or not. As these area unit pre-trained in OpenCV, their learned information files additionally return bundled with OpenCV opencv/data/.

(i)Haar Cascade classifier:

Haar cascade classifier is an efficient object detection technique. it's a machine learning primarily based approach wherever a cascade perform is trained from heaps of positive and negative pictures. it's then accustomed sight objects in alternative pictures. Our interest is to target face detection. Initially, the algorithmic program desires heaps of positive pictures and negative pictures to train the classifier and it extract options from it. For this, Haar classifier options shown in Figure.2 are used. [7]

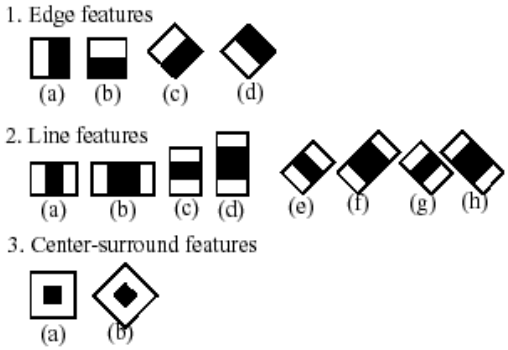

Figure.2: "Haar classifier features (input to the basic classifier)"

The feature utilized in a specific classifier is fixed by its form (1a, 2b, 3a, etc.) every feature could be a single price obtained by subtracting total of pixels underneath white parallelogram from total of pixels underneath black parallelogram

(ii) LBP Classifier

The native Binary Patterns (LBP), additionally must be trained on many thousands of pictures. LBP could be a visual/texture descriptor, and fortunately, our faces also are composed of

small visual patterns. So, LBP options are extracted to make a feature vector that classifies a face from a non-face [6].

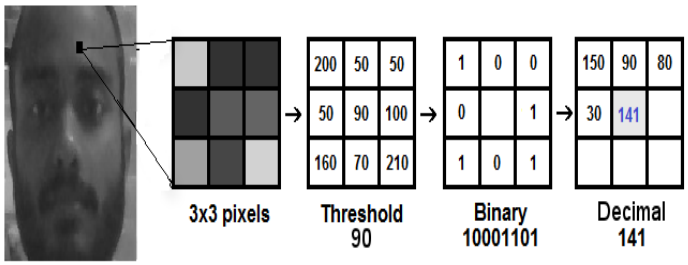

Figure.3: "LBP conversion to binary"

For each sub-window, LBP appearance at nine pixels (3×3 window) at a time and with a specific interest within the element situated within the center of the window. Then, it compares the central element price with each neighbor's element price underneath the 3×3 window. for every neighbor element that's bigger than or adequate the middle element, it sets its price to one, and for the others, it sets them to zero. After that, it reads the updated element values (which will be either zero or 1) in a very dextrorotatory order and forms a binary range. Next, it converts the binary range into a decimal range, which decimal range is that the new price of the middle element. We tend to do that for each element in a very block as shown in Figure.3.

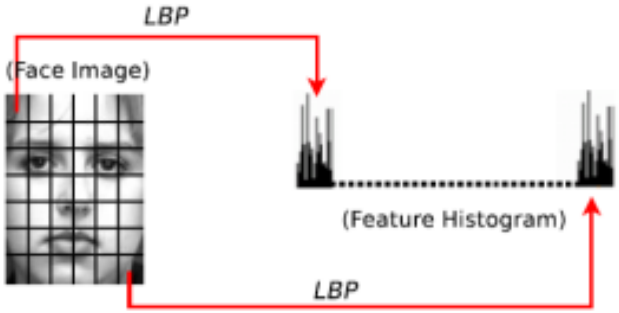

Figure.4: "Face image is split into sub windows from that LBP bar graphs are extracted and concatenated into single histogram"

LBP converts every block price in to histograms. So, the histograms are extracted from face image and eventually it concatenates these block histograms to make a 1 feature vector for one image, that contains all the options we tend to have an interest as shown in Figure.4. In our face detection work, we've chosen Haar cascade classifier to realize high detection accuracy and low false positive rate. The output obtained in our face detection module is shown in Figure.5.

Sample code:

```
# Read the image
gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
# Create the haar cascade
```

```
faceCascade=cv2.CascadeClassifier(cv2.data.haarcascades+"haarcascade_frontalface_default.
xml")
# Detect faces in the image
faces = faceCascade.detectMultiScale(gray,scaleFactor=1.3, minNeighbors=3,    minSize=(30, 30))
```

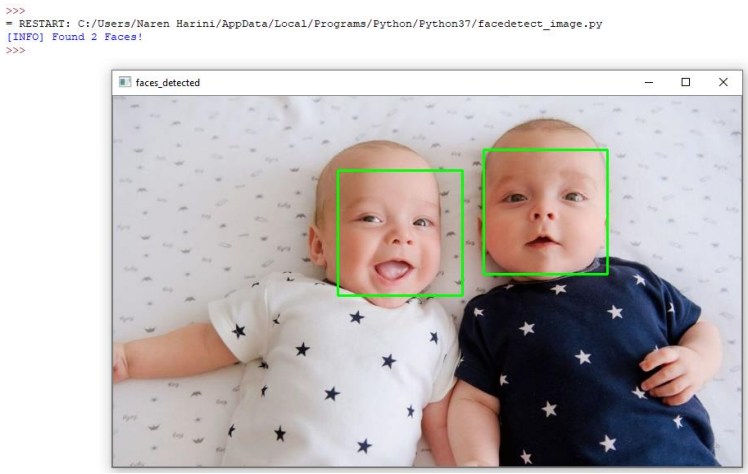

Figure.5: "The output obtained from the face detection module"

(3) Training face recognizer:

To recognize the faces of new people, the face recognizer has to be trained. OpenCV equipped with three built-in face recognizers.

i. EigenFaces:cv2.face.createEigenFaceRecognizer()
ii. FisherFaces:cv2.createFisherFaceRecognizer()
iii. Local Binary Pattern Histogram(LBPH):cv2.face.createLBPHFaceRecognizer()

We chosen to use LBPH face recognizer as Eigen faces and Fisher faces are both affected by light conditions and in real life this parameter cannot be guaranteed. The idea with LBPH is not looking at the image as a whole; instead try to find its whole structure by comparing each pixel to the neighboring pixels. Working model of face recognizer is as follows:

i. Input a new image to the face recognizer
ii. Face recognizer generates histogram for the input image
iii. It compares the histogram of input image with the histogram that already has.
iv. Finally it finds the best match, and returns the label associated with the best match.

Sample code:

```
# using face recognizer
face_recognizer = cv2.face.LBPHFaceRecognizer_create()
```

(4) Prediction:

The actual face prediction and recognition is done in this module. We took one test image for each person and passed it to face detection module. This module identifies the face in the image passed and this face images are then passed to the face recognizer for prediction.

Sample code:

```
def predict(test_img):
        #make a copy of the image
        img = test_img.copy()
        #detect face from the image by calling face detection module
        face,rect = detect_face(img)
        #predict the image using face recognizer
        label = face_recognizer.predict(face)
        #get name of respective label returned by face recognizer
        label_text = subjects[label[0]]
        #draw a rectangle around face detected
        draw_rectangle(img, rect)
        #draw name of predicted person
        draw_text(img, label_text, rect[0], rect[1]-5)
    return img
```

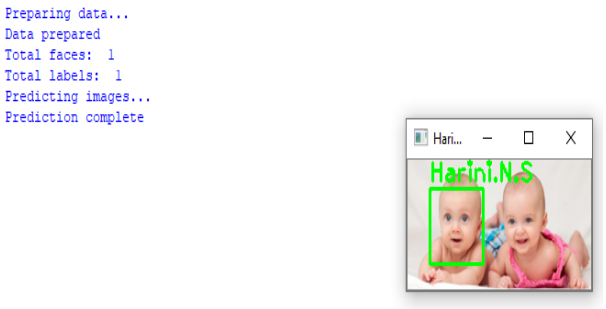

Figure.6: “Output obtained from prediction and recognition module”

Face recognizer compares the detected face with the faces in the training dataset. If the match found, face recognizer return the image with the associated labels. The output obtained from face prediction and recognition module is shown in Figure.6.

4. CONCLUSION AND FUTURE ENHANCEMENT

In this paper a method for performing face detection and face recognition is shown with different modules, sample codes and output images. OpenCV has made our work easy and straight forward. We made the comparison between three face recognizers and proved that LBPH is fine and used in our recognition module. Even the better ways are there to perform face recognition using Neural Networks. So, this work can be extended by implementing more advanced face recognition algorithms using a combination of OpenCV and Machine Learning.

In future we are in plan to extend this work to detect fake videos which is created using fake video software such as Adobe after effects by analyzing different parameters of human face.

5. ACKNOWLEDGEMENT

We would like to thank the entire anonymous user who helped us through question answering and discussion forums. We also would like to thank for the support of our family to complete this project and paper successfully.

6. REFERENCES

1. Khan, S. Chakraborty, R. Astya and S. Khepra, "Face Detection and Recognition Using OpenCV," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India,2019, pp. 116-119, doi: 10.1109/ICCCIS48478.2019.8974493.
2. P. Apoorva., H. C. Impana., S. L. Siri., M. R. Varshitha. and B. Ramesh., "Automated Criminal Identification by Face Recognition using Open Computer Vision Classifiers," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 775-778, doi: 10.1109/ICCMC.2019.8819850.
3. Archana, T & Venugopal, T.. (2015). Face recognition: A template based approach. 966-969.10.1109/ICGCIoT.2015.7380602.
4. Starovoitov, Valery & Samal, Dmitry. (1999). A geometric approach to face recognition.. 210-213.10.13140/2.1.3200.8644.
5. Emami, Shervin & Suciu, Valentin. (2012). Facial Recognition using OpenCV. Journal of Mobile, Embedded and Distributed Systems.
6. https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html
7. http://www.face-rec.org/algorithms/
8. https://www.superdatascience.com/blogs/opencv-face-recognition/
9. https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#face-recognition-with-opencv