Face Detection and Recognition using HAAR Feature-based Cascade Classifiers and Local Binary Patterns Histograms (LBPH) algorithm

Mrigesh Kumar [1], Neeti Taneja[2] , Avneet Kaur[3]

1 U.G Student, Dept. of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, India
2 Assistant Professor, Dept. of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, India
3 U.G Student, Dept. of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, India

Abstract

This paper presents a methodology for recognizing the human face based on the features derived from the image. The proposed methodology is implemented in three modules. The first module create training data by collecting samples of face as extracted face using HAAR Feature-based Cascade Classifiers. The extracted faces will be transformed using processing techniques like Gray scaling, Histogram equalization, and Resizing then saved in specified directory with unique name. In the next module the extracted faces are used to train the model using Local Binary Patterns Histograms (LBPH) algorithm. The last module run the face recognition by passing the face through webcam to prediction model and give the confidence value for different users.The proposed methodology uses OpenCV and it's face detection methods and Python and it's packages like numpy.

Keywords: Face recognition, Gray scaling, Histogram equalization, Resizing, HAAR Feature-based Cascade Classifiers, Open CV(Computer Vision),Local Binary Patterns Histograms (LBPH) algorithm.

INTRODUCTION:

Face recognition is a major challenge encountered in multidimensional visual model analysis and is a hot area of research. The art of recognizing the human face is quite difficult as it exhibits varying characteristics like expressions, age, change in hairstyle etc[1][5].

Face recognition plays a crucial role in applications such as biometric identification, security system, credit card verification, identifying criminals in airport, railway stations etc [6].There are different types of face recognition algorithms, for example:

- Eigenfaces(1991)
- Local Binary Pattern Histograms(LBPH)(1996)
- Fisher faces(1997)

Although many methods have been proposed to detect and recognize human face developing a computational model for a large database is still a challenging task. That is why face recognition is considered as high level computer vision task in which techniques can be developed to achieve accurate results.

The proposed methodology is implemented in three modules. Since the human face can be identified by certain facial characters in the first step the relevant features from the facial image are extracted. The extracted faces will be transformed using processing techniques like Gray scaling, Histogram equalization, and Resizing then saved in specified directory with unique name,so that it will be easy to recognize the face from these features. For face detection Local Binary Pattern Histograms (LBPH) algorithm is used. To recognize the face detected HAAR Feature-based Cascade Classifiers are used to provide the confidence value for different users.

METHODOLOGY:

The proposed method implement an Face Detection and Recognition technique which is independent of variations in features like color, hairstyle, different facial expressions etc using Local Binary Pattern Histograms(LBPH) algorithm and face detection technique named HAAR Feature-based Cascade Classifiers . The process flow of the proposed approach is as shown in Figure 1.

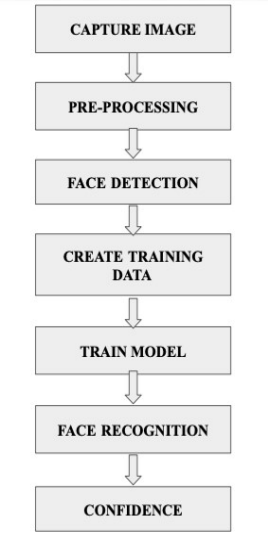

Figure 1: Step-by-step approach

I. PRE-PROCESSING:

The human face is a complicated multidimensional visual model and hence it is very difficult to develop a computational model for recognizing it. So for accurate recognition of the image, unwanted noise needs to be removed so as to smoothen the image. For obtaining a smoothened image, it needs to be pre-processed. In the Pre-processing stage after capturing image there will be further transformed using processing techniques like grayscale transformation , histogram equalization (for consistent brightness and contrast), resizing (to have the same dimension). Histogram equalization and resizing are used to reduce the recognition module's lighting and scaling sensitivity. In OpenCV, many functions grayscale images before processing. This is done because it simplifies the image, acting almost as a noise reduction and increasing processing time as there is less information in the image. Using histogram we can also visualize individual color components of any image.

II. FACE DETECTION:

After pre-processing HAAR Feature-based Cascade Classifiers algorithm is applied for detecting the face in the image. HAAR Classifier is the face detection method implemented by OpenCV. It is version of Viola-Jones detector, a face detection technique developed by Paul Viola and Michael Jones, based on HAAR features, also known as HAAR-like wavelets. A HAAR feature or more correctly a HAAR-like wavelet is a combination of adjacent black and white rectangles, which can be used to describe face components (eyes, mouth, eyebrows). Examples of HAAR-like wavelets and how can they be used for face detection are shown in Figure 2(a) and Figure 2(b) respectively.

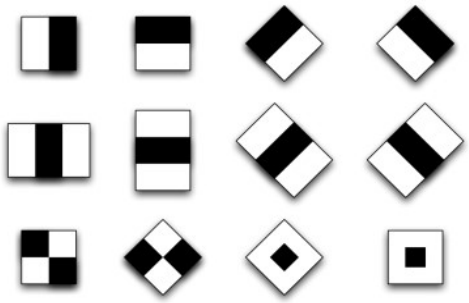

Figure 2(a): HAAR-like wavelets

Figure 2(b): HAAR feature used for face in an image

The presence of a HAAR-like wavelet [2][3] is determined by subtracting the average dark-region pixel value from the average light-region pixel value. If this difference is greater than a given threshold, we decide that this feature is present. If an image area contains a series of HAAR features, then that area will be considered a face, otherwise an element from the background.

III. CREATING TRAINING DATA:

In this step we will used the function to detect faces and returns the cropped face. If no face detected, it returns the input image. The detected face is cropped and resized [4] to a standard resolution. To create the training data we will collect 100 samples of face. We will put count on images and display live count upto 100 showing sample number as shown in Figure 3. All face as file are saved in specified directory with unique name. If there is no face detected until collection of 100 samples it will give the "Face not found" result. After successfully collection of samples it will give the "Collecting Samples Complete" result.

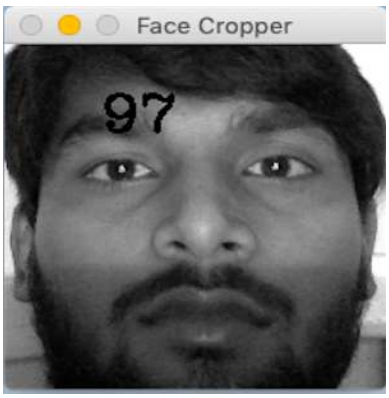

Figure 3: Collecting samples with live count

IV. TRAIN MODEL:

In this step we will train our model using Local Binary Pattern Histograms(LBPH) algorithm and by initializing facial recognizer. We will get the training data we previously made and saved in specified directory with unique name. We will open training images in our datapath.Then

numpy arrays for training data and labels are created. For modeling we will initialize the facial recognizer and used Local Binary Pattern Histograms (LBPH) algorithm. It uses Local Binary Pattern (LBP) which is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. Using the LBP combined with histograms we can represent the face images with a simple data vector. As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation [7].

Step-by-Step Algorithm:
1. Parameters: the LBPH uses 4 parameters.
2. Training the Algorithm.
3. Applying the LBP operation.
4. Extracting the Histograms.

V. FACE RECOGNITION:

In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

- So to find the image that matches the input image we just need to compare two histograms.
- We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: **euclidean distance**, **chi-square**, **absolute value**, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

- So the algorithm output is the '**confidence**' measurement. We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is higher than the threshold defined.

RESULT AND DISCUSSION:

Initialize webcam to capture image and perform some processing technique like grayscaling, histogram equalization (for consistent brightness and contrast), resizing (to have the same dimension). Pre-processing removes unnecessary noise from the image. The original image captured by webcam as shown in figure 8(a) is pre-processed using grayscaling as shown in figure 8(b) which simplifies the image, acting almost as a noise reduction and increasing processing time as there is less information in the image.

Figure 8(a): Original Image

Figure 8(b): Grayscale Image

After grayscaling we can also visualize individual color components of original image by histogram as shown in figure 8(c) .Then, we will implement sharpening as shown in figure 8(d), which has the effects of in strengthening or emphasizing edges in original image.

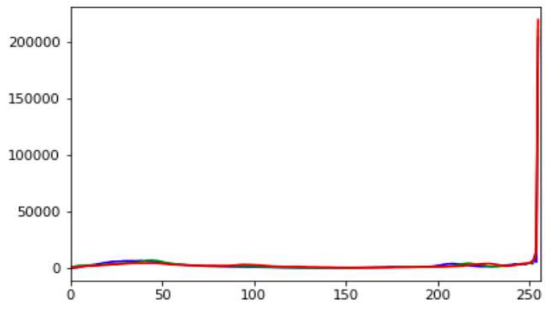

Figure 8(c): Color component of Image using Histogram

Figure 8(d): Sharped Image

After preprocessing Haar Feature-based Cascade Classifiers algorithm is applied for detecting the face in the image. It provide function which detects faces and returns the cropped face. If no face detected, it returns the input image. The detected face is cropped and resized to a standard resolution. To create the training data we will collect 100 samples of face. We will put count on

images and display live count upto 100 showing sample number as shown in figure 3. After this step we will train our model using Local Binary Pattern Histograms(LBPH) algorithm and by initializing facial recognizer. We will get the training data we previously made and saved in specified directory with unique name.

Then pass faces to prediction model it will predict the model and give the "results" as shown in figure 8(e) comprises of a tuple containing the label and the confidence value of users.

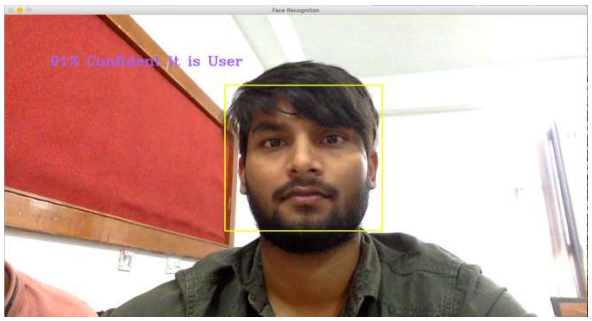

Figure 8(e): Confident Value as Result

CONCLUSION:

The paper presents an efficient approach for face detection and recognition using Local Binary Pattern Histograms(LBPH) algorithm and face detection technique named Haar Feature-based Cascade Classifiers.. These techniques can be applied in the detection of face from image and can be used as various application such as biometric identification, security system, credit card verification, identifying criminals in airport, railway stations etc Face detection and recognition plays a vital role in a wide range of applications. In most of the applications a high rate of accuracy in identifying a person is desired hence the proposed method can be considered in comparison with the existing methods.

REFERENCES:

[1]. Janarthany Nagendrajah, “Recognition of Expression Variant Faces- A Principle Component Analysis Based Approach for Access Control” IEEE International Conference On Information Theory and Information Security, pp 125-129,2010.
[2]. Maria De Marsico, Michele Nappi, Daniel Riccio and Harry Wechsler, “ Robust Face Recognition for Uncontrolled Pose and Illumination Changes”IEEE Transaction on Systems, Man and Cybernetics, vol.43,No.1,Jan 2013.
[3]. Shaoxin Li,Xin Liu, Xiujuan Chai, Haihong Zhang, Shihong Lao and ShiguangShan, “ Maximal Likelihood Correspondence Estimation for Face RecognitionAcross Pose” IEEE Transaction on Image Processing, Vol.23, No.10, Oct 2014.

[4]. Jyothi S Nayak and Indramma M, "Efficient Face Recognition with Compensation for Aging Variations" IEEE International Conference On Advanced Computing, pp 1-5, Dec2012.
[5]. Shermina J, "Illumination Invariant Face Recognition using Discrete Cosine Transform and Principal Component Analysis", IEEE International Conference On Emerging trends in Electrical Computer Technology, pp 826-830,2011.
[6]. Anil K. Jain, "Face Recognition: Some Challenges in Forensics", IEEE International Conference On Automatic Face and Gesture Recognition, pp 726- 733,2011.
[7]. Narayan T. Deshpande, Dr. S. Ravishankar, "Face Detection and Recognition using Viola-Jones algorithm and Fusion of PCA and ANN", ISSN 0973-6107 Volume 10, Number 5 (2017) pp. 1173-1189.