

Face Recognition with Single Sample per Person using Haar Cascade and LBPH

(Implementation Paper)

A.L.Amutha

Assistant Professor (O.G)

S.P.Kanmani

Assistant Professor (O.G)

G.Sivagami

Assistant Professor (O.G)

Department of Computer Science and Engineering
 SRM University, Kattankulathur, Chennai-603203, India

Abstract -- In most real-world situations like in law enforcement, e-passport, ID card verification, etc. there is only one sample per person. Face Recognition (FR) with One Sample per Person is a very challenging problem due to the lack of information to predict the variations in the query image this results in the failure of most face recognition systems. This is because most current algorithms cannot recognize the face due to the extremely limited representation of training sample. The first step is to detect the face in each image. Once, we get the region of interest containing the face in the image, we will use it for training the recognizer. For the purpose of face detection, we will use the Haar Cascade Classifier and Local Binary Patterns Histograms to train and recognize the faces from the YALE DATABASE.

Index Terms—Single training sample per person, face recognition, Haar Cascade Classifier, Linear Binary Pattern Histograms.

I. INTRODUCTION

The aim of face recognition is to identify or verify one or more persons from still images or video images of a scene using a stored database of faces. Many research efforts have been focused on how to improve the accuracy of a recognition system, however, it seems that most of them ignore the potential problem that may stem from the face database at hand, where there may be only one sample image per person stored, possibly due to the difficulties of collecting samples or the limitations of storage capability of the systems, etc.

The performance of the learning mechanism of face recognition systems is quite heavily affected by the number of training samples for a particular face [7]. The current face recognition techniques use several samples of the same person.

Unfortunately, we face the problem of having far less number of training samples than what is actually required. In real world cases especially in identification applications, such as law enforcement, driver license or passport card identification, etc.

There is usually only one training sample per person in the database.

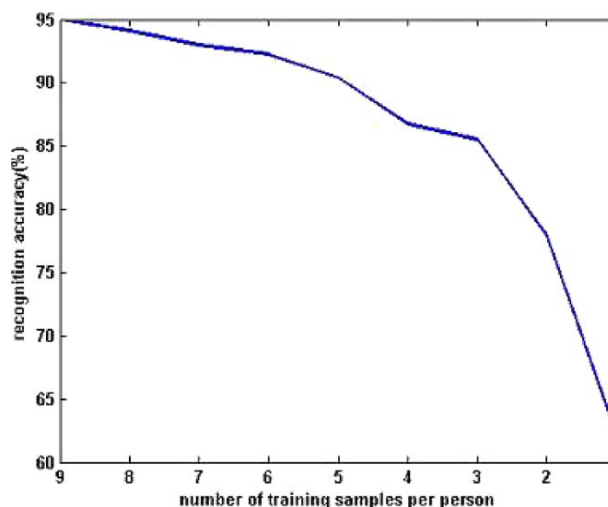


Figure 1: Graph for number of training samples per person vs. recognition accuracy.

The process of Face Recognition Systems can be primarily divided into-

1. Find a good database of faces like YALE Face Database or AT & T Face Database.
2. Extract the features from the training samples in the database using the classifier.
3. Detect the faces in the database images and then use them to train the face recognizer.
4. Test the face recognizer to recognize the faces it was trained for. [1]

II. ADVANTAGES OF ONE SAMPLE PER PERSON

There are several advantages in keeping just one sample per person in a database.

1. Samples are easy to collect- Constructing a face database is very time-consuming work thus collecting one sample per person is easier as compared to multiple samples per person.
2. Saving the cost of storage-the cost for storing in a database depends on the amount of information being stored.
3. Saving cost of computation-the number of training samples per person is directly proportional to the cost of computation thus the cost can be quite less for large scale databases with only one sample per person.

III. RECOGNIZING FROM ONE SAMPLE PER PERSON

Various existing methods dealing with robust face recognition from a single intensity image can be studied. These methods can be classified into three categories, according to the type of features used by various methods:

- (1) *Holistic methods*: The whole face image is taken as input, but the main challenge faced by these methods is how to address the extremely small sample problem.
- (2) *Local methods*: The local facial features are used for recognition. Incorporating global configurationally information into local face model can be a problem, so care should be taken.
- (3) *Hybrid methods*: Both the local and holistic features are used to recognize a face. These methods offer better performance than individual holistic or local methods, since more comprehensive information is utilized [3] [1].

We have used Local methods using local facial features for face recognition as they are suitable for handling the one sample per person problem, this is because in local methods, a set of low dimensional local feature vectors are used to represent the face which helps in better extraction of the features. Also, local methods have flexibility in recognizing a face based on its parts, thus the common features can be easily extracted.

IV. CLASSIFIER

An effective object detection method proposed by Paul Viola and Michael Jones is Object Detection using Haar feature-based cascade classifiers in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. Basically it is

a machine learning based approach using a lot of positive and negative images from which a cascade function is trained. Then it is used to detect objects in other images. Initially, a lot of positive images (images of faces) and negative images (images without faces) are required by the algorithm to train the classifier in order to find the object in the input images. Then features need to be extracted from it. The Haar Cascade Classifier is trained with a few hundred sample of a particular object, called positive examples are scaled to the same size, and negative examples which can be arbitrary images but of the same size. After the classifier is trained, an input image can be applied to it and the region of interest, which is the object (e.g. Faces) can be found in the images."1" is shown as output if the region is likely to show the object and "0" if the region of interest cannot be found. The word "cascade" in the classifier name means that the classifier consists of several simpler classifiers that are applied to a region of interest until at some stage the object is rejected or all the stages are passed.

V. LOCAL BINARY PATTERNS – LBP

This method involves the extraction of local features from images. The concept is to define only local features of an object instead of looking at the whole image as a high-dimensional vector. The features extracted this way will have a low-dimensionality implicitly. But we can observe the image representation that we are given doesn't only suffer from illumination variations. Considering things like scale, translation or rotation in images, your local description has to be at least a bit robust against those things. The main idea of Local Binary Patterns is to condense the local structure in an image by comparing each pixel with its neighborhood.

VI. MATHEMATICAL PRINCIPLES

A more formal description of the LBP operator can be given as:

$$LBP(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c)$$

With (x_c, y_c) as central pixel with intensity i_c , and i_n being the intensity of the neighbor pixel. s is the sign function defined as:[2]

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

Very fine details can be described with this. An arbitrary number of neighbors on a circle is taken in order to find the following neighborhoods.

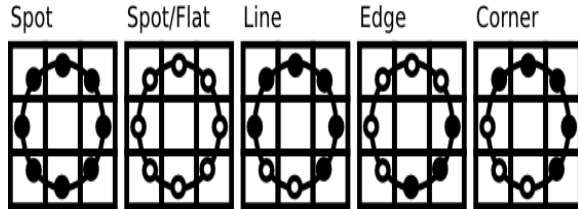


Figure 2: The above neighborhoods can be captured using LBPH

For any given Point (x_c, y_c) the position of the neighbor $(x_p, y_p) p \in P$ can be calculated by:

$$x_p = x_c + R \cos\left(\frac{2\pi p}{P}\right)$$

$$y_p = y_c - R \sin\left(\frac{2\pi p}{P}\right)$$

Where R is the radius of the circle and P is the number of sample points [2].

The LBP operator is robust against monotonic gray scale transformations which can be seen by-

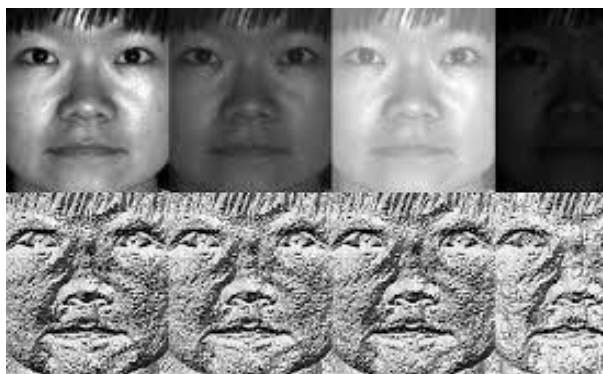


Figure 3: An artificially modified image

The LBP image is then divided into multiple local regions and histograms are extracted from each.

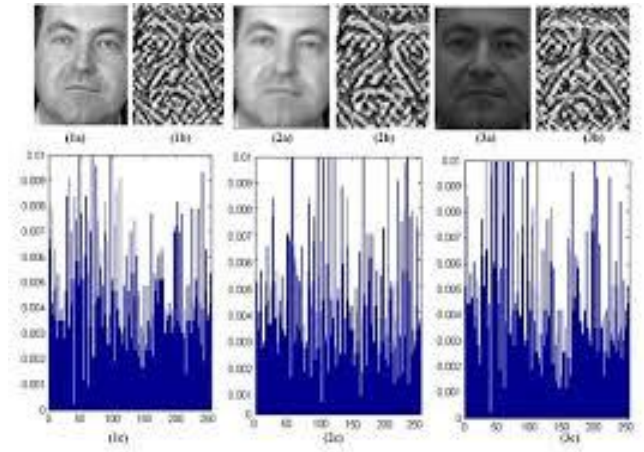


Figure 4: Histograms for the local regions of the LBP image.

VII. YALE FACE DATABASE

The Yale Face Database (size 6.4MB) contains 150 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, and sleepy, surprised, and wink. Images differ in head position, lighting conditions, beard, glasses, hairstyle, expression and age of subject [1].



Figure 5: YALE Database

VIII. EXISTING WORK

Building a deep neural network using supervised autoencoders which are considered as the building blocks for deep architectures. First the faces are treated with all types of variants (for example, illumination, expression, occlusion, or poses) as

images contaminated by noises. Experiments have been done on the AR, Extended Yale B, CMU-PIE, and Multi-PIE data sets. These experiments show that by using sparse representation-based classification, the stacked supervised autoencoders-based face representation can be used to recognize the faces with single sample per person in the data set and it achieves higher recognition accuracy compared with other deep learning models, in spite of much less training data and without any domain information. Though with a supervised auto-encoder, we can recover the face without the variant, but it has less stability and the resultant image is restricted to 32x32 pixels [3].

IX. PERFORMANCE EVALUATION

First step in our implementation is to detect the face in an image. For this we get the region of interest from the image which contains the face, this is used for training the recognizer. Haar Cascade is used in order to detect the face and it is provided by OpenCV.

CascadeClassifier.detectMultiScale is a function used to detect the faces. Next step is to create the face recognizer object. The face recognizer object has functions like *FaceRecognizer.train* to train the recognizer and *FaceRecognizer.predict* to recognize a face. For our implementation we will use Local Binary Patterns Histograms Face Recognizer. So the path to the training image is given and the images are read as a numpy array.

Next *CascadeClassifier.detectMultiScale* is used to detect the faces from the images (region of interest). From the YALE DATABASE only one sample per person has been taken for detecting the face. This function returns a list of faces. Now we train the recognizer using *FaceRecognizer.train*.

This function takes two arguments one is the list of faces and the other is the label for each face.

Next are the remaining images in the DATASET which are left for testing the face recognizer. Thus again the region of interest is found in the testing images and then *FaceRecognizer.predict* is used to recognize the faces. Each image gets a confidence level attached which tells how confident the recognizer is in recognizing the image.



Figure 6: One sample per person from Yale Database is taken as input.



Figure 7: Image of a person from Yale Database

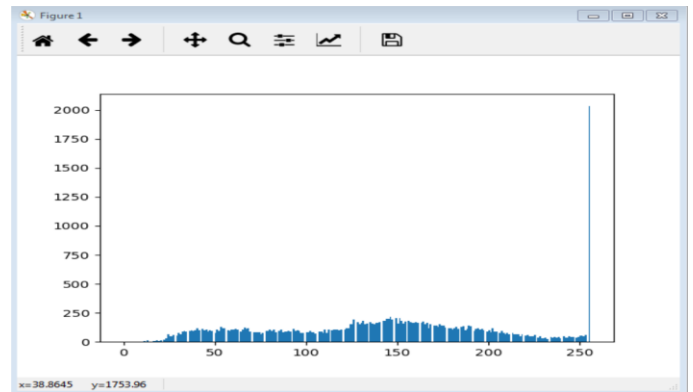


Figure 8: Histogram of the above person

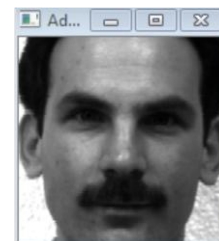


Figure 9: Image of another person from Yale Database

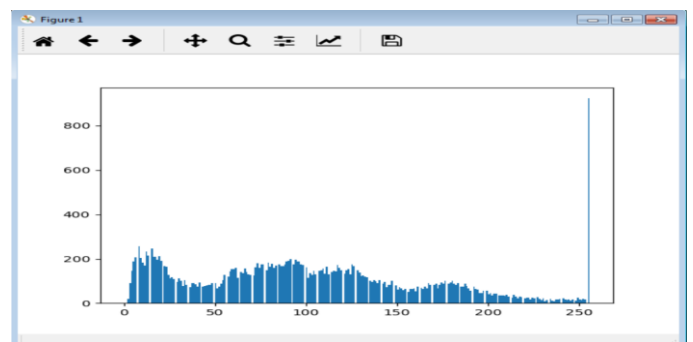


Figure 10: Histogram of the above person

12 is Correctly Recognized with confidence 68.12800:	Intell. 22 (12) (2000) 1424–1445.
12 is Correctly Recognized with confidence 50.77353:	
12 is Correctly Recognized with confidence 50.77353:	
12 is Correctly Recognized with confidence 66.01432:	
12 is Correctly Recognized with confidence 36.70300:	
12 is Correctly Recognized with confidence 55.91338:	
12 is Correctly Recognized with confidence 51.84937:	
13 is Correctly Recognized with confidence 59.41093:	
13 is Correctly Recognized with confidence 41.02300:	
13 is Correctly Recognized with confidence 46.35581:	

Figure 11 Result which shows the minimum difference between the training image and the testing image

X. CONCLUSION

In this paper, we have discussed the implementation of face recognition system using single training sample per person. This implementation has higher efficiency than most other extensively used algorithms which heavily depend on the dataset. The performance evaluation has been done by assigning the confidence level of the face recognizer for each testing image. In future, we intend to make it more efficient and more robust.

REFERENCES

- [1] Face recognition from a single image per person: A survey. Xiaoyang Tana, b, Songcan Chena, c, *, Zhi-Hua Zhou, b, Fuyan Zhang, b
- [2.] Face Recognition with OPENCV-
http://docs.opencv.org/2.4/modules/contrib/doc/face_rec_facerec_tutorial.html
- [3.] Single Sample Face Recognition via Learning Deep Supervised Autoencoders
 Shenghua Gao, Yuting Zhang, Kui Jia, Jiwen Lu, *Member, IEEE*, and Yingying Zhang
- [4.] A. Samal, P.A. Iyengar, Automatic recognition and analysis of human faces and facial expressions: a survey, *Pattern Recognition* 25 (1) (1992) 65–77.
- [5.] An extended set of Haar-like features for rapid object detection- R. Lienhart, J. Maydt
Image Processing, 2002. Proceedings. 2002 International Conference on 10 December 2002.
- [6.] M. Pantic, L.J.M. Rothkrantz, Automatic analysis of facial expressions: the state of the art, *IEEE Trans. Pattern Anal. Mach.*