International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Face Recognition Utilizing Single Query Picture

Sukant Tiwari[†*] and A.K. Shukla[‡]

[†]Department of Computer Science and Engineering, SHIATS, Allahabad, India
[‡]Department of Information Technology and Engineering, SHIATS, Allahabad, India

Accepted 31 May 2015, Available online 08 June 2015, **Vol.5, No.3 (June 2015)**

Abstract

In this paper, we propose a system of face recognition from single sample image, which is robust to pose, illumination, expression and background changes. It is fast to deploy and performs at a good rate. In the method, we extract the normalized face from input image using the Viola-Jones object detection method, eliminating the background effects. The face is divided into regions. and observe the different kinds of patterns observed for each face region. Based upon the patterns observed, a codebook for each region is generated. Illumination variations are overcome using the robust and popular image encoding and compression function, the DCT, is used for encoding the data. Histograms of the feature patterns provide robustness against pose variations of the face features. Performing weighted sum of the differences and having low weight for the mouth region incorporates robustness to expression changes. The distinctiveness of each face is taken into account, thus making the system similar to human beings. The system is robust and fast in a similar manner.

Keywords: *Face recognition, Single sample image, Violoa-Jones object detection, DCT*

1. Introduction

The advances in the face recognition technology have vast and unforeseen applications and implications. It is a powerful technology with huge potential of changing the way we live. As such, it has attracted the attention of lots of researchers. Face recognition has been a challenge in the field of pattern recognition for a long time. In real-life, a lot of situations are encountered where only one image per person is available. It is an extreme case of the Small Sample Per Person (SSPP) problem of pattern recognition. Most of the methods for multiple samples fail in this case. So, specialized methods for this case need to be developed. Majority of the proposed solutions for the problem are much less robust than the multiple sample methods. The problem is an active research topic.

Machine recognition of faces from still and video images is an active research area spanning several disciplines such as image processing, pattern recognition, computer vision and neural networks. In addition, face recognition technology (FRT) has numerous commercial and law enforcement applications. These applications range from static matching of controlled format photographs such as passports, credit cards, photo IDs, driver's licenses, and mug-shots to real-time matching of surveillance video images presenting different constraints in terms of processing requirements. Although humans seem to recognize faces in cluttered scenes with relative ease, machine recognition is a much more daunting task.

Face recognition is typically used in biometric applications. It is supportive in providing security, surveillance and granting authorized access. Amongst the various biometric measures, it is the most attractive for a user due to its ease of use, and inexpensive hardware. It may be used with or without the knowledge of the subject. That is its major advantage. It allows for such a technology to be used in places where other biometrics will not survive, an example being the surveillance at a bus-stop. Thus, the area remains an active research field due to its potentially numerous areas of application.

Implementation Methodology

Methodology of Evaluation & Metrics

The face recognition algorithms are evaluated based on their performance upon the same testing image database. We are going to use the following two performance metrics

Precision

It refers to the accuracy of the algorithm in correctly recognizing the faces in the database. It is represented in the form of percentage of correct recognition. It may be calculated as follows:

*Corresponding author: **Sukant Tiwari**

$$P = \frac{CR \times 100}{TF}$$

where P represents the Precision,
CR represents the number of Correctly Recognized faces, and TF represents the Total Faces *in the testing database.* This metric is important as it gives the success rate of the algorithm. The area of application of the algorithm, or whether the algorithm is ready for applications is mostly decided based upon this metric.

Execution Time

It refers to the time that the algorithm spends after getting the input till the generation of the desired output result. It is represented in the form of units of time spent. It places an upper limit on the time taken by the algorithm to arrive at a result. This metric is an important consideration if the algorithm is going to be deployed in real-time or similar critical systems.

2. Design and Analysis

Our face recognition system has the following modules:

(a)Face detection and normalization
(b)Features' codebook generation
(c)Finding the mean face
(d)Generating the face recognition database
(e)Input face image matching

This module marks the beginning of the various transformations which are performed upon the image for face recognition. In order to develop an intelligent system that can recognize faces from given input images or videos, it is absolutely essential for it to be able to locate the human faces in the given input. This property of the system is known as the automated face detection.

Fig.1 Data Flow Diagram (DFD) of the proposed Face Recognition System

In order to check whether the given face image is vertical or not, we can compare the location of the eyes. If they both are at the same height, it means that the face is vertical. For comparing, we must make orientation of all the faces same. Using this method we can check for the vertical orientation of the face at a fast speed. So, we chose to make the face orientation in all the images vertical.

In order to compare the face images, their dimensions have to be same. The dimensions of all the face images in the database must be the same. The dimensions must be chosen in such a way that they should be neither too small, nor too large.

Thus, every face must be resized to fixed dimensions for the database. The face of different people is of different size.

(a) Face detection and normalization

The dimensions of the face of a person, especially the lower part keep changing with expressions as well. So, while generating the standard normalized face images it is important to overcome this variation. To achieve this, we try to find the location of the centre of the upper lip of the person. Below that point, a lot of variations with expressions occur. We stretch or shrink the face image in order to have the eyes and the upper lip centre at the same position in all the images. This way, we are able to measure the difference in the sizes of various features of the faces of different people.

Other variations to be overcome are that of illumination variation, variation in colour depth. Here we are using three bytes per pixel, for example the RGB colour images. Here the three values respectively indicate the intensity values of the Red, Green and Blue colours for each pixel. The complexity and the colour depth of images may thus vary a lot. For our database, we need to define standards such that the colour depth variation problem is solved and all the images fulfil the required specifications. We know that colour images can be easily converted to their greyscale counterparts as well. We are able to distinguish only about 1000 different shades of a colour. Thus we do not need more than 10-bit(2^{10} = 1024) values for encoding each colour intensity. Thus, we choose a minimum depth greyscale standard for our database. And, in order to overcome the illumination variation problem, revealing the hidden information in the image, we equalize the intensity histogram of the greyscale image.

(b) Features' codebook generation

(Timo Ojala *et al*,1996), authors proposed the Local Binary Patterns for texture analysis. The same approach was applied to face recognition by the authors (T. Ahonen *et al*,2004). The complete face image was divided into many equally sized cells. For each cell, the LBP operator was applied to each pixel and its neighbourhood. It assigned labels to each pixel. The histogram of the labels was used as the feature descriptor of the cell. They found that on using a 3×3 size operator, 90% of the patterns were encoded.

Authors (Ce Zhan *et al*, 2009) extended this approach. They proposed to divide the image into regions such that each region contain a feature. They used the Discrete Cosine Transform (DCT) in order to assign the labels. The DCT was applied on each pixel and its 8×8 neighbourhood. We know that the first DCT coefficient (DC coefficient), which represents the average pixel value of block is highly susceptible to variation in illumination(E. Ifeachor *et al*,2001). They experimented with various row-wise subsets of the DCT coefficient matrix and chose one based upon results.

The authors (C.C. Chang *et al*, 2002) proposed to use some significant DCT coefficients in order to apply it for image authentication, as they would easily overcome minor modifications made to the image. They proposed to use the first ten DCT coefficients in a zigzag scan order for image authenticity. It is so because they represent the low frequency components of the image and contain most of the energy of the image.

Combining these two approaches, we experiment upon two subsets of the DCT coefficients. The first subset being that of the first nine DCT coefficients in a zigzag scan order, excluding the first coefficient. The other being that of the first five DCT coefficients in a zigzag scan order, excluding the first coefficient.

So upon implementing this method, we would have either nine or five coefficient values per pixel of the region. This actually, is the data of the kind of textures that are observed for that feature in the face. We gather the information of all the textures that might possibly occur for that feature in any face. In order to achieve that, we do this same processing for each region on all the images in the training database. Thus we g*et al*l the possible variations.

Upon gathering the data, we need to group the data into a fixed number of distinct clusters. Here each cluster represents a class of the textures which occur in the feature under consideration. For forming groups, we use the k-means clustering algorithm. It forms a fixed 'k' number of clusters from the given data. Each data node is assigned to the nearest cluster, based upon the Euclidean distance metric. The cluster-centres are updated after each distribution. Clustering stops when the cluster-centres stop changing. The cluster-centres form the code of the codebook.

Based upon the number of groups formed, we create the codebook for the region with the same number of codes.

Thus, individual codebooks are generated for each region of the face.

(c) Generating the mean face

We need to find the degree to which the features of a particular face vary from the general, commonly occurring features. In order to be able to do that, we must first have an average face to compare to. Earlier we have ensured that the location of the various features of the face in the image database is fixed. The image is greyscale as well, i.e. there is only one intensity value per pixel. It means that the information of the image is contained within one channel. We are to find the average value of the various features of the face. Since the positions of the features is fixed, we can simply find the mean of the intensity values observed for each pixel over the whole training database. This will provide us with a face image containing the most commonly occurring features, i.e. the most common face.

In the end before saving, the image intensity histogram is equalized to overcome the intensity variation problem.

(d)Generating the face recognition database

Before starting to recognize faces, we are to tell the system which subjects' images are there to be recognized. For each subject we provide only one frontal face image to the system. The images are processed and their data is kept in the database, ready for comparison.

The DCT values are generated for each pixel of the region. When we consider the problem of classification in human recognition from a pattern recognition point of view, we find that there are too many classes, and only one correct class. So, the nearest neighbour classifier is more suitable than any other complicated classifiers. The nearest matching code is found using the k-nearest neighbour strategy.

We count the occurrence of each kind of code in the given region. The histogram of the assigned codes for each region is generated. In this way, the spatial information within the region is ignored. Pose variation in the feature is overcome.

The authors (T. Ahonen *et al*,2004) based upon numerous face recognition surveys, concluded that all parts of the face do not have the same importance for face recognition. It means that the contribution of different parts towards face recognition varies. Some parts are more significant towards recognizing face than others. So, they assigned weights to the various different parts according to their general importance in face recognition. Later, this trend was followed by many other authors as well (Caifeng Shan *et al*,2005). The authors (Ce Zhan *et al*, 2009) further extended this approach. Till now the authors had proposed the application of same weights for all the persons. The authors argued this point. Based upon the studies (A. O'Toole *et al*, 1994) and (P. J. Hancock *et al* 2000), they said that the humans remember the unusual features of faces and use them for recognition.

They introduced the notion of variable region-weights for different faces. They proposed for the use of the Mahalanobis distance metric for finding out the distinctiveness of the facial features. In statistics, it is a measure of similarity of a new sample to an already known one. It involves finding out the mean values for each pixel location, finding out the covariance between the two quantities, computing the inverse of this covariance matrix, etc.

Upon empirical analysis of this system, we found that using the Mahalanobis distance is a computationally expensive task. This is observed because of the computation of the inverse of the high-dimensional covariance matrix. Due to this reason, the metric is unsuitable for the development of real-time face recognition system in future.

In order to overcome this drawback, we need another measure to find out the variation of the given input face feature from the mean face feature. Upon pondering over this problem, we found that it was quite similar to the problem we are already attempting to solve: finding the degree of variation between two faces. So we proposed to generate the histograms of codes for the two features to be compared, and later find the distance between them.

In this way, we found the weights of the features for each subject in a much faster manner. The weights and the histograms of each region for every image are stored for future matching.

(e) Input face image matching

We find the histogram of the codes from the codebook for each region of the input image.

The nearest match of the given input image is to be found from the system database by comparing the code histograms. In order to compare the histograms, some authors considered the various proposed histogram dissimilarity measures and concluded to use the weighted chi square statistic (χ_w^2) measure:

$$\chi_w^2(S,M) = \sum_{i,j} w_j \frac{(S_i - M_i)^2}{S_i + M_i}$$

where w_j is the weight of region j.

The input image data is to be compared with the data of the images in the system database. We compare the region histograms of the input image with each of the system database image histograms by the chi square statistic, using the weights of the system database image. The value finally generated is the dissimilarity measure of the input image from the system database image.

After comparing with the whole of the database, the match is the system database image with the minimum sum, i.e. the least dissimilarity.

Before beginning the proposed steps, some parameters need to be decided.

For generating the normalized face image, the dimensions of the generated images need to be decided. Keeping the arguments given earlier in mind, that the dimensions should be neither too large, nor too small. We decide upon the face image dimensions of 64×64.

The colour depth of the images should neither be too low nor too high, as per the discussion above. So, we decide upon the depth of 8-bit single channel (greyscale) per pixel values. It means that the intensity of each pixel can vary from 0 to 255.

The number of regions and their location and size need to be defined. From the survey (A. O'Toole *et al*, 1994) it was found that the eyebrows, eyes, nose and the mouth are features important for the human recognition of the faces. The authors of (Ce Zhan *et al*, 2009) divided the image according to those features as well. We propose to divide the face into the same regions. The location and size of the regions was decided by experimenting upon the training database.

The number of codes in the codebook (or the codebook size) need to be experimentally determined for the best possible results. Ideally, the size should be such that they encode only the inter-personal variations and not the intra-personal variations. It means that the size should be sufficiently large to store all variations of different faces, but not the expression variations.

For the application of the k-nearest neighbour strategy, we are storing four values of each cluster, including the cluster-centre. The match is the cluster with maximum occurrence amongst the nearest data nodes, with a maximum of 33 nodes.

3. Experimental Setup

The experiments for the proposed technique were done in the C language using the OpenCV library. The Code::Blocks IDE was used with the MinGW GCC compiler. The experiments have been performed on a Intel vPro dual core system with 512MB RAM and Windows XP SP3 operating system.

It has been found that the lower part of the face varies a lot with expressions, contributing least towards face recognition. So, its weight is not calculated. Instead, it is assigned the least weight of the other five regions.

We have used the BioID face database (Frischholz *et al*,2000) for the training of the system. It contains 1521 facial images of 23 subjects with varying illumination, background, poses, time and expressions.

The ORL face database contains 400 facial images of 40 subjects, with similar variations. It is used as the testing database.

For image normalization, the coordinate values of Table 1 were found from the database and used subsequently.

Table 1 Table of used coordinate positions

Feature	Coordinate position
Right eye	(16,19)
Left eye	(46,19)
Mouth centre	(33,50)

In the database images the faces were detected, and their features normalized to the given positions. The position of these features in all the normalized images were same as shown in Fig.2 The normalization process.

Fig.2 The normalization process

For finding the region dimensions, grids were applied to the normalized faces as shown in Fig 3

Fig.3 Deciding the region dimensions

The selected dimensions are shown in Table 2.

Table 2 Selected region dimensions for normalized images

Feature	Origin	Width	Height
Right eyebrow	(0,0)	32	14
Left eyebrow	(32,0)	32	14
Right eye	(0,14)	32	15
Left eye	(32,14)	32	15
Nose	(0,29)	64	14
Mouth	(0,43)	64	21

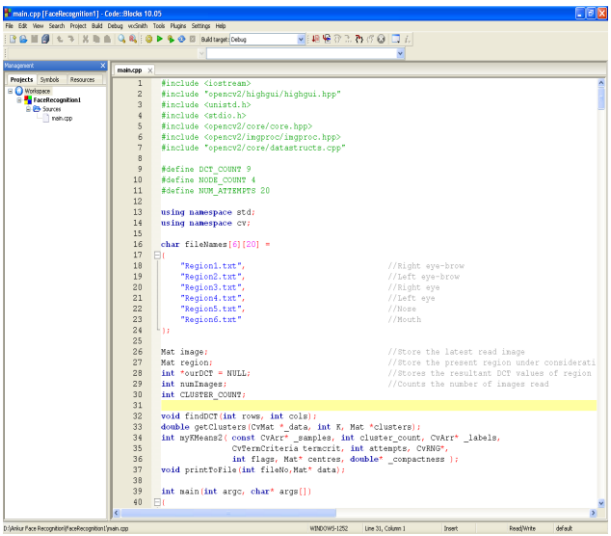

Fig.4 The used Code::Blocks IDE

4. Test Application

For testing the application, the system database was generated using the 64 cluster codebooks. The arguments were the codebook size and the folder location of subject images as shown in Fig..

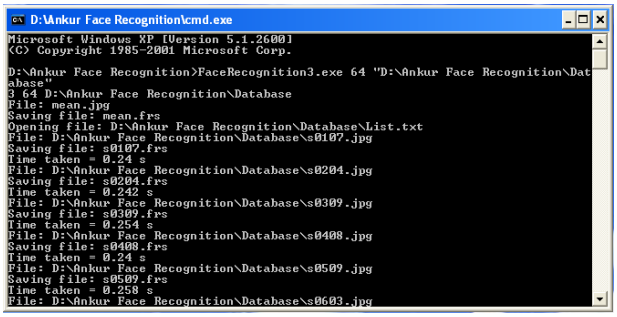

Fig.5 Generating the database for 64 cluster codebooks

The generated database was used for recognizing a given sample face image of one of the subjects. The arguments were the codebook size, database folder location and the location and name of the file to be matched. The file was compared with all the subjects and finally the correct match was found as shown in Fig.6 and Fig.7

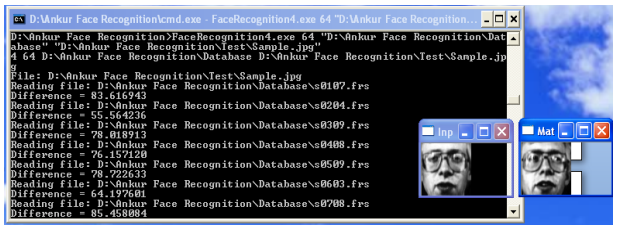

Fig.6 Finding match for the input face image – 1

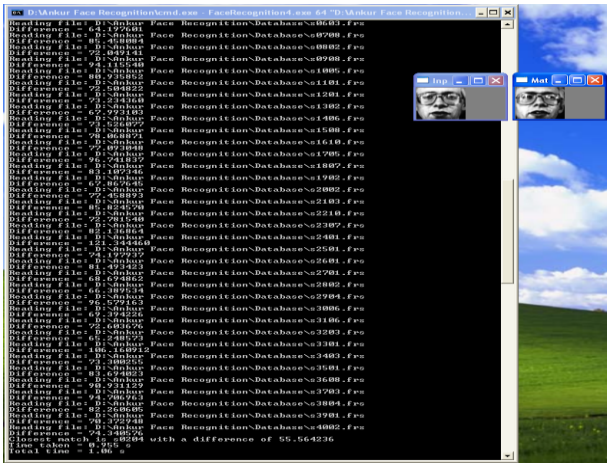

Fig.7 Finding match for the input face image – 2

5. Performance Results and Analysis

Precision Performance

The system is run for varying codebook sizes and different DCT subsets over the ORL database. The results obtained are shown in figure 8.

It can be seen that the system has improved performance over the existing approach in case of DCT subset of five zigzag coefficients. It means that it is a worthwhile encoding unit for face recognition purposes. The results also show the efficiency of the proposed weighting scheme. The performance of DCT subset of nine zigzag coefficients didn't achieve sufficient accuracy. It means that using nine zigzag coefficients includes data which varies a lot, even for the same person. That causes random recognition behavior.

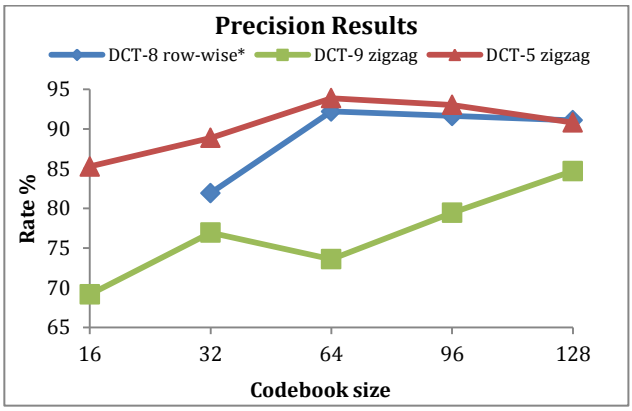

Fig.8 Precision results comparison over the ORL database

**Results as per (Zhan et al.2009)*

The precision of recognition first increases with codebook size and then decreases. It is in confirmation of the theory about the effect of encoding the inter-personal and intra-personal variation. When the inter-personal texture variations are encoded, they improve recognition performance. On the other hand, if the codebook includes intra-personal variations it results in decreased recognition precision. The ideal codebook size would balance both. The codebook sizes of 64 and 96, being quite close in terms of precision, are used for the next test.

Execution Time Performance

The system now uses existing weighting metric and the proposed one for the generation of the 40 subject system database. The time is recorded. The comparisons are made over codebook sizes of 64 and 96. The DCT subset of five zigzag coefficients is used for both the weighting metrics

Table 3 Execution time testing for database generation

	64	96
Mahalanobis weight	3260 sec	3280 sec
DCT weight	10.2 sec	19.7 sec

Time taken to process the 40 image database using DCT-5 zigzag coefficients for different weights and codebook sizes

Thus, we can clearly see from Table 3 that the proposed method is much faster. In case of the Mahalanobis weight, time taken is not affected by variation in codebook size as it does not consider it. Though, the proposed approach of using the codebook histogram variations from mean for generating weights, is definitely affected by the codebook size increase. The time taken almost doubles itself, although the time taken is itself many times lesser than the existing technique.

Fig.9 shows the reason the approach works. It works by matching pattern histograms of features, which do not vary much with pose.

Fig.9 Typical examples for regional histograms of the left eyebrow region

Conclusion

In this paper, we have proposed a method towards solving the problem of face recognition from single sample image. We studied some methods applied till now for the problem. Based upon the study, we experimented with some of our own ideas and came up with a solution. We proposed to use some significant DCT coefficients of the feature image for its encoding. The encoding was compared with corresponding encodings of same parts from other images. Thus, we proposed to match the faces based upon the matching of their features. The importance of the individual features towards recognition was taken into account as well. This idea had already an available method for it, but we proposed another method, which is much faster than the earlier method.

Later, after practical implementation and the analysis of the results, it was found that the proposed system is indeed an effective way of solving the problem. Its major advantage is its ability to develop the recognition database at a fast rate. It recognizes with an improved accuracy as well.

References

A. O'Toole.(1994) Structural aspects of face recognition and the other-race effect , Memory & Cognition ,pp. 208–224.

Beymer (1995) Face recognition from one example view, Fifth International Conference on Computer Vision Proceedings,), pp.500-507

C.C. Chang (2002),Recognition of Image Authenticity Using Significant DCT Coefficients Quantization, Informatica,pp. 359-366

Caifeng Sha (2005),Robust facial expression recognition using local binary patterns, IEEE International Conference on Image Processing, pp. II- 370-3

Ce Zhan (2009) Face recognition from single sample based on human face perception, Image and Vision Computing New Zealand,pp.56-61

E. Ifeachor (2001) Signal Processing: A Practical Approach, Prentice Hall

F. Samaria (1994),,Parameterisation of a stochastic model for human face identification, Proceedings of 2nd IEEE Workshop on Applications of Computer Vision,pp. 138–142

Gong, S., McKenna (2000), Dynamic Vision: From Images to Face Recognition, Imperial College Press & World Scientific Publishing, pp.36

He, Jiazhong, Zhang (2008),"A New Image Enhancement Method for Face Recognition with Single Training Sample, Congress on Image and Signal Processing, pp.216-219

Hongtao Yin (2006), Sampled Two-Dimensional LDA for Face Recognition with One Training Image per Person, First International Conference on Innovative Computing, Information and Control, pp.113-116.

Hung-Son Le (2005), Can the Surveillance System Run Pose Variant Face Recognition in Real Time?, Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance pp.209-216.

Hung-Son Le (2004), Identification system using single hidden Markov model and single sample image per person, IEEE International Joint Conference on Neural Networks Proceedings, pp. 459.