Enhanced and Fast Face Recognition by Hashing Algorithm

M. Sharif*, K. Ayub, D.Sattar, M. Raza, S. Mohsin

Department of Computer Science,
COMSATS Institute of Information
Technology Wah Cantt Pakistan.
*muhammadsharifmalik@yahoo.com

ABSTRACT
This paper presents a face hashing technique for fast face recognition. The proposed technique employs the two existing algorithms, i.e., 2-D discrete cosine transformation and K-means clustering. The image has to go through different pre-processing phases and the two above-mentioned algorithms must be used in order to obtain the hash value of the face image. The searching process is increased by introducing a modified form of binary search. A new database architecture called *Facebases* has also been introduced to further speedup the searching process.

Keywords: DCT, K-means, Facebases, database, hashing, recognition.

1. Introduction

There are two basic concerns in face recognition systems namely face identification and face authentication [1][2][3][4]. For face identification, a system has to determine whether an input image matches a person's data in the database. For authentication, a system has to make sure that an input image has an exact match with a particular individual's data in the database records.

A typical face recognition system has generally three modules, i.e., preprocessing, feature selection criteria and classification module. The preprocessing step includes the segmentation process for finding face data in an image. The area of interest, the face in this case, is extracted in this step and normalization is performed on it. Sometimes, a preprocessing step also includes the feature extraction from the segmented face data like eyes, nose, lips, etc.

Several face recognition techniques have been proposed and developed based on extracting facial features and calculating geometrical distances in facial parts including distance from nose to eyes, distance from nose to forehead, distance between eyes, etc. Unfortunately, these techniques are not able to maintain the accuracy rate in recognizing facial parts. Moreover, neural networks came in market to maximize the recognition rate by improving the training set of images.

The proposed technique is related to face hashing and this section tries to present the existing techniques

which are related and can sometimes be helpful to face hashing. The idea of face hashing has been extracted from early graphical feature mapping techniques. For example, Noor Zaman et al. [5] presented the concept of face bar code in their paper but the problem is that they did not provide enough results to prove the validity of the technique.

V.V. Starovoitov et al. [6] in their paper described facial recognition using geometric features. In this technique the face is marked as different segments. These segments are created by marking different points on the face, as a result of which fifteen segments and twenty eight features of face are created between the points. These features are then measured by the distances between the points. Feng Yang et al. [7] used the geometrical mapping on the face which helped in pose variation and face identification. King Hong Cheung et al. [8] introduced the secrets of face hashing by using false assumption. There are a number of face recognition systems available for commercial use namely Cognitec [9], Eyematic [10], Viisage [11], Identix [12],[20], [21] and [13], among others.

2. Proposed algorithm

The proposed technique is based on calculating numerical values of a face image for fast recognition. These values should be unique for each face in the database. For this purpose, 2-D Discrete Cosine Transformation (2-D DCT) has been selected for creating the face image. The calculated hash value of the face is then saved into the database with the corresponding face ID. Storing the database with respect to hash values will help to apply modified binary search to recognize the faces from the database. To further decrease the searching time, the database containing the hash values is separated into ten files (i.e. 0, 1, 2… 9) corresponding to the first character of hash values. The whole process consists of two different phases: database generation phase and binary search for face recognition phase.

2.1 Database Generation Phase

The training database for images consists of two things, i.e., the pre-processed or normalized images and the database files having information about the saved images. The images to be stored in the database must follow the preprocessing step. Figure 1 depicts this scenario.

2.1.1 The input Images

A number of train face images were used for each individual to be stored in the database. Each train image for a person is different depending upon the conditions like pose, illumination, background and other facial expressions, among others. This will help to find the faces more accurately. All the images in the database should be in grayscale hence the very first step after image acquisition is to convert the image to grayscale (See Figure 2).

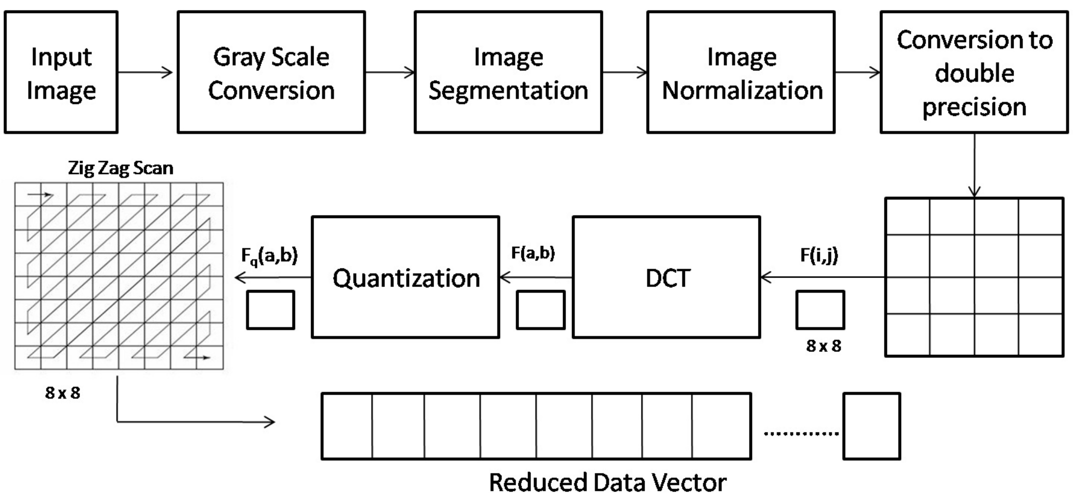


Figure1. Pre-processing and dimensionality reduction.

(a) (b)

Figure 2. a) original image, b) grayscale image.

2.1.2 Preprocessing

The preprocessing phase segments the face region from the input image. The face region is then normalized (scaling, sizing and positioning) in standard format of the stored database. The normalized image is then converted to double precision image intensity.

2.1.3 Data Reduction

The segmented faces are processed further for block-based transformation. These faces are divided into many blocks in order to apply linear transformation to them. The block size may be 8×8 or 16×16 and lie in a sequence. The linear transformation provides energy compaction (data compression) and a well-known linear transformation 2-D DCT has been selected for this purpose which provides only N number of coefficients from the block of size MxM (N < MxM).

Let, for example M=8, i.e., block size be MxM which after applying DCT is represented by N, say N=8. These eight coefficients are high energy components in that block and have low frequency. The DCT is represented by the following equation as mentioned in [14] [15]:

$$y(k) = w(k) \sum_{n=1}^{N} x(n) \cos \frac{\pi(2n-1)(k-1)}{2N}, k = 1,..., N \qquad (1)$$

where x_n is even around n=0 and even around n=N-1:

$$w(k) = \begin{cases} \frac{1}{N}, k=1 \\ \sqrt{\frac{2}{N}}, 2 \le k \le N \end{cases} \qquad (2)$$

2.1.4 Facebases Generation

The vectors created using DCT coefficients are then quantized to discard the non significant information from the image (quantization is lossy) [14] [15]. These coefficients then pass to the next step for Facebase generation.

This paper introduces a term named *Facebase* for face database. The architecture of the proposed database (Facebases) is different from the traditional face databases in that it consists of ten database files, i.e., FB0, FB1, FB2, FB3… FB9 depending upon the first digit of the hash value (described later). This means that information of a face having the hash value starting with digit 0 will be saved into the database file FB0, information of a face having the hash value starting with digit 1 will be saved into the database file FB1 and so on.

The design of the Facebase generator uses the K-means clustering [16] algorithm for generation of Facebases from the training set of vectors. For this, the mean vector is calculated from all the reduced data vectors formed from the data reduction step described previously. The K-means filter [16] is then applied to the resultant vector with k clusters. The proposed technique selects four clusters which can be varied [16].

$$\arg_s \min \sum_{i=1}^{k} \sum_{x_j \in s_i} \left\| x_j - \mu_i \right\| \qquad (3)$$

where ($\mathbf{x}_1$, $\mathbf{x}_2$,…, $\mathbf{x}_n$) is a set of vectors from the previous step.
k = number of clusters (partitions)
S = k sets i.e., $\mathbf{S} = \{S_1, S_2 \ldots S_k\}$

Depending upon the four partitions, the four centroid values are calculated. Figure 3 shows four centroid values of a face image produced after applying the 2-D DCT transform, zig-zag scanning and the Facebase generation algorithm (through K-means clustering)

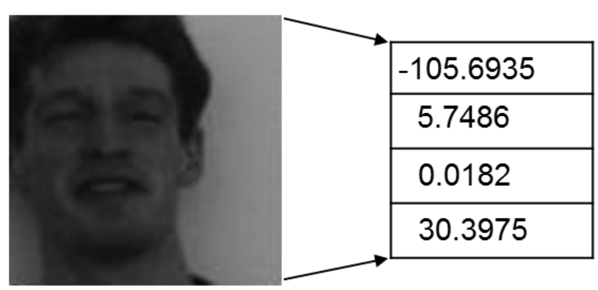


Figure 3. Four centroid values of a face image.

The sum of these four centroid values is calculated and considered as the face hash value. See Figure 4.

Vectors
-105.6935
5.7486
0.0182
30.3975

Sum
-69.5292

Figure 4. A Face hash.

The face hash value is then saved into the Facebases corresponding to the first digit of hash value (i.e., the sum of vectors is calculated and based on its first digit the corresponding Facebase is loaded for searching). Figure 4 shows that face value -69.5292 will be stored in FB6 (Facebase number 6). The whole face hashing process is explained in Figure 5.

The hashes in Facebases are stored in sorted order so that the modified binary search could be applied for fast searching. Figure 6 shows the format of a Facebase database.

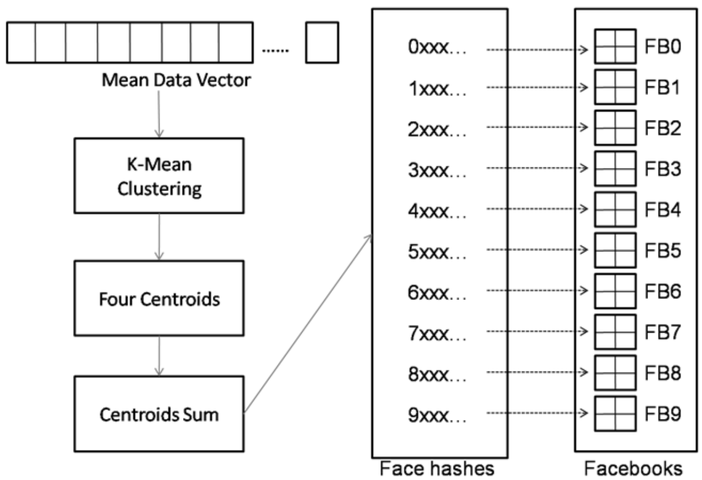


Figure 5. Face hash storing process.

Centroid Of Cluster

FB4

-11.5478	4.2350	-409.2513	-0.3392	-416.9034	36
-12.9431	9.4030	-409.6738	-0.2855	-413.4995	9
-12.4156	10.1382	-408.5586	-0.2580	-411.0940	10
48.6913	-1.5248	0.4661	-458.3303	-410.6977	40
-13.1292	10.0194	-406.4799	-0.2845	-409.8742	11
-14.7031	8.8243	-402.9926	-0.3319	-409.2034	15
-14.7031	8.8243	-402.9926	-0.3319	-409.2034	4
-14.8563	9.1825	-398.3262	-0.3291	-404.3292	16
-14.8563	9.1825	-398.3262	-0.3291	-404.3292	5
-17.1060	10.2213	-392.2459	-0.3697	-399.5003	7
16.3060	-98.7191	87.0996	0.3122	4.9987	45

Sum

Picture ID

Figure 5. Figure 6. A Facebase database.

Each entry in a database consists of six fields: four fields for the centroid of clusters (hash value), one field for the sum of centroids and one for the picture ID to keep track of pictures in the database. This proposed scheme of database arrangement stores the data in a symmetric and arranged order. The fields defined for each entry in a Facebase helps to track pictures as well as the hash values of each entry; therefore, it efficiently helps to speed up the recognition process because of requiring less searching time to find a match for test image in a database.

2.1.5 Face Recognition Process

The proposed scheme of Facebase generation and database record management helps to speed up the recognition rate and achieves the goal of automatic face recognition; it also makes the search fast and reduces bandwidth consumption. The initial steps followed in the proposed architecture for the recognition module are the same as in the database generation step i.e.:

Proposed Algorithm: Face Recognition

Take a segmented face region as an input image.
Convert the image to grayscale.
Segment the image and extract the face region.
Normalize the image.
Convert the image to double precision.
Apply 2-D DCT on the image.
Apply quantization.
Apply zig-zag scan to form the reduced data vector.
Apply the K-means clustering algorithm with k=4 and find the centroid values.
Sum the centroid values resulting in the face hash value and find the first digit of this hash value.
Depending upon the first digit, select the corresponding Facebase
Apply modified binary search to find the particular face ID in the selected Facebase. (Described later).

2.2 The Modified Binary Search (MBS)

To speed up the search process, a modified binary search algorithm has been introduced. As the hash values are calculated for images, there may be a difference of hash values depending upon illumination, pose, etc.; because of that, a threshold limit is introduced in modified binary search scheme. This limit may vary depending upon the database size (Total database entries in a Facebase)

The algorithm for MBS is as follows:

Proposed Algorithm: MBS

Check the length of the Facebase. (say N)
Check the number of entries in the Facebase with threshold limit t (say t=30)
if (N > t) then

Apply the correlation matching or minimum distance between vectors using the following equation:

$$\gamma_k = \sum_{i=1}^{N} \left\| \alpha_i \right| - \left| \beta \right\|, k = 1,2,3,4 \quad (4)$$

Now, calculate the sum of vectors :

$$\theta = \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 \quad (5)$$

or

$$\theta = \sum_{i=1}^{4} \gamma_i \quad (6)$$

where, $\boldsymbol{\alpha}$ represents the vectors (centroid) stored in Facebase $\boldsymbol{\beta}$ represents the new vectors of the new image $\boldsymbol{\gamma}$ represents the absolute distance between stored and new vectors $\boldsymbol{\theta}$ represents the sum of absolute differences of these vectors N represents the Facebase length Check for the entry whose correlation value results greater than or equal to 0.98, otherwise the input image match will not exist in a database.

if (N > t) then

Check the first and last entry of Facebase (first entry as F_{b1} and last entry as F_{bL})

If (Fs<Fb1), then (where Fs is the hash value to be searched and Fb1is the first entry of the current Facebase) divide the threshold value by its half (say t/2 = 15) and match the hash value with the

last t/2 entries from the previous Facebase of the selected one, i.e., Fs € Fb-1 [(t/2) L]

If (Fs>Fb_L), (where the hash value to be searched is greater than the first entry of selected Facebase), divide the threshold value by its half (say t/2 = 15) and match the hash value with the first t/2 entries from the next Facebase of the selected one.

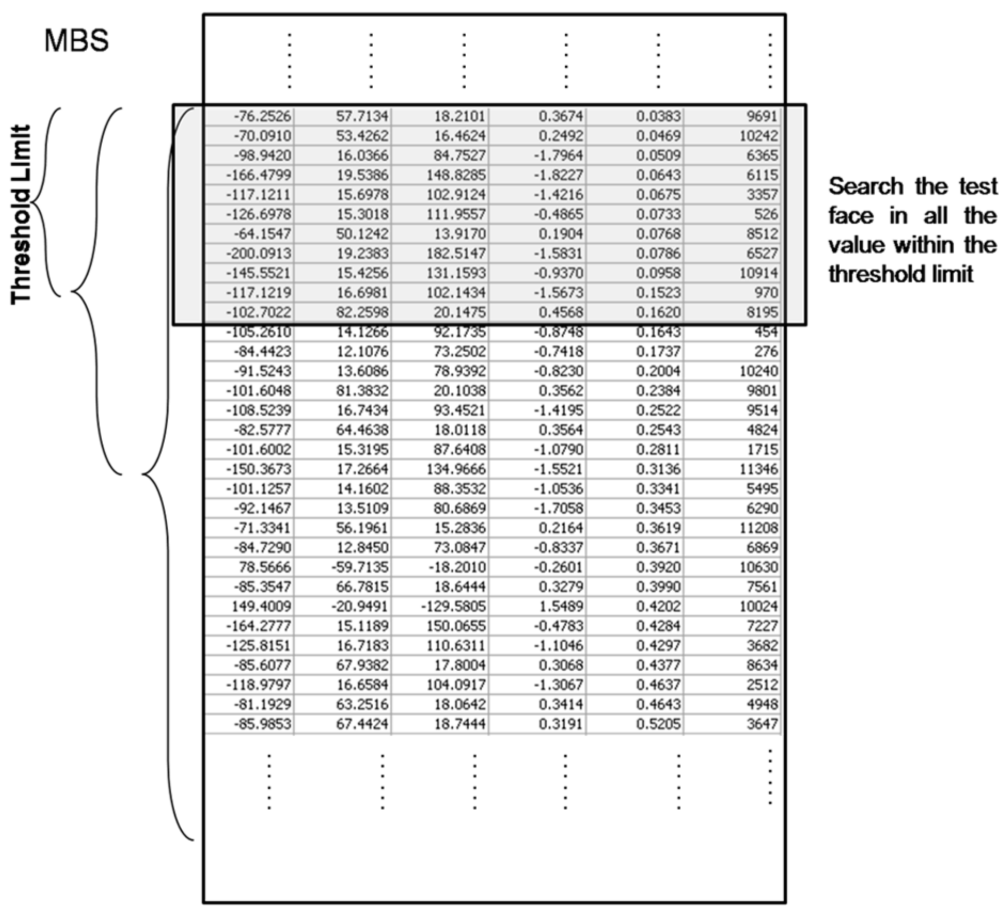
Fs € Fb+1 [(t/2) 1]
If (Fs€FbL), (the hash value to be searched is within the first and last entry of the Facebase), it means that the match can take place in the selected Facebase. Now apply the normal binary search algorithm and find the closest match.

After finding the closest hash value, select the previous t/2 values and the next t/2 values of the image with closest hash value in the selected Facebase (call it the threshold limit values).

Apply the correlation matching on these values (step 2, Equation 3)
Now, calculate the sum of vectors (step 2, Equations 4 and 5)

Check for the entry whose correlation value results greater than or equal to 0.98, otherwise the input image match will not exist in a database.

Figure 7 shows the way the modified binary search works.



⋮	⋮	⋮	⋮	⋮	⋮
-76.2526	57.7134	18.2101	0.3674	0.0383	9691
-70.0910	53.4262	16.4624	0.2492	0.0469	10242
-98.9420	16.0366	84.7527	-1.7964	0.0509	6365
-166.4799	19.5386	148.8285	-1.8227	0.0643	6115
-117.1211	15.6978	102.9124	-1.4216	0.0675	3357
-126.6978	15.3018	111.9557	-0.4865	0.0733	526
-64.1547	50.1242	13.9170	0.1904	0.0768	8512
-200.0913	19.2383	182.5147	-1.5831	0.0786	6527
-145.5521	15.4256	131.1593	-0.9370	0.0958	10914
-117.1219	16.6981	102.1434	-1.5673	0.1523	970
-102.7022	82.2598	20.1475	0.4568	0.1620	8195
-105.2610	14.1266	92.1735	-0.8748	0.1643	454
-84.4423	12.1076	73.2502	-0.7418	0.1737	276
-91.5243	13.6086	78.9392	-0.8230	0.2004	10240
-101.6048	81.3832	20.1038	0.3562	0.2384	9801
-108.5239	16.7434	93.4521	-1.4195	0.2522	9514
-82.5777	64.4638	18.0118	0.3564	0.2543	4824
-101.6002	15.3195	87.6408	-1.0790	0.2811	1715
-150.3673	17.2664	134.9666	-1.5521	0.3136	11346
-101.1257	14.1602	88.3532	-1.0536	0.3341	5495
-92.1467	13.5109	80.6869	-1.7058	0.3453	6290
-71.3341	56.1961	15.2836	0.2164	0.3619	11208
-84.7290	12.8450	73.0847	-0.8337	0.3671	6869
78.5666	-59.7135	-18.2010	-0.2601	0.3920	10630
-85.3547	66.7815	18.6444	0.3279	0.3990	7561
149.4009	-20.9491	-129.5805	1.5489	0.4202	10024
-164.2777	15.1189	150.0655	-0.4783	0.4284	7227
-125.8151	16.7183	110.6311	-1.1046	0.4297	3682
-85.6077	67.9382	17.8004	0.3068	0.4377	8634
-118.9797	16.6584	104.0917	-1.3067	0.4637	2512
-81.1929	63.2516	18.0642	0.3414	0.4643	4948
-85.9853	67.4424	18.7444	0.3191	0.5205	3647
⋮	⋮	⋮	⋮	⋮	⋮

Figure 7. Modified binary search.

3. Experiments and results

3.1 Results for the ORL Database

The accuracy and performance of the existing symbolic LDA (linear discriminant analysis) is reviewed in the atomizing face recognition process using the “database of faces” (formerly 'The ORL database of faces'). The ORL face database [17] consists of 400 images in such a way that it contains ten images for each of the 40 unique persons in various poses. All the images have a uniform background with the persons standing in frontal position. The spatial and gray level intensities of the images are 92□112 and 256, respectively. In our test, all images are resized to 80×80. All the 400 images from the ORL database are used to assess the face recognition performance of the proposed scheme. Six images are randomly selected from the ten images of each person for training and the remaining is used for testing. Table 1 shows the training time and recognition rate comparison of the existing and the proposed techniques using the Yale face database.

The graph in Figure 8 shows the comparison of different techniques with respect to the time required for the training of images. The x-axis shows various techniques and the y-axis represents the training time in seconds.

Methods	Training Time (Sec)	Resolution	Recognition Rate (%)
Fisherfaces	98	80x80	92.8
Eigenfaces	102	80x80	87.65
Symbolic PCA	38	80x80	94.85
Symbolic ICA	87	80x80	89.15
Symbolic KPCA	110	80x80	95.45
Symbolic LDA	19	80x80	97.5
Proposed	25	80x80	98.2

Table 1. Comparison of the results for the ORL Database [18].

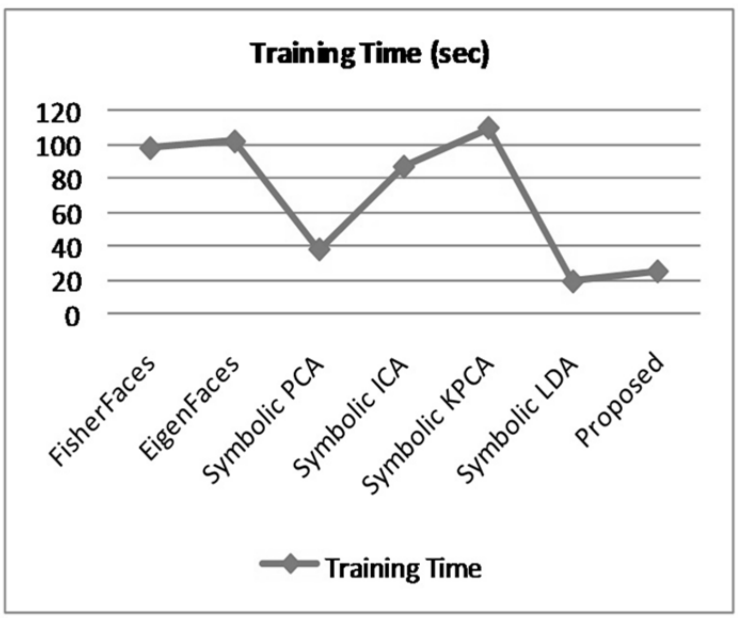


Figure 8. Training time comparison for the ORL database.

Figure 9 shows the comparison of the recognition rate of various previously defined recognition techniques with the proposed scheme of recognition. The y-axis represents the recognition rate in percentage.

3.2 Yale Face Database results

The experiments are performed using the Yale database [19] to compare the face recognition accuracy of the proposed technique with other existing techniques. In these experiments, 9 images are arbitrarily taken from each class for training and the remaining is used for recognition or testing purpose. Table 2 shows the training time and recognition rate comparison of the existing and the proposed techniques using the Yale Face Database.

Figure 10 shows the comparison of different techniques with respect to the time required for training of images. The x-axis shows various techniques and the y-axis represents the training time in seconds.

Figure 11 shows the comparison of the recognition rate of various previously defined recognition techniques with the proposed scheme of recognition. The y-axis represents the recognition rate in percentage.

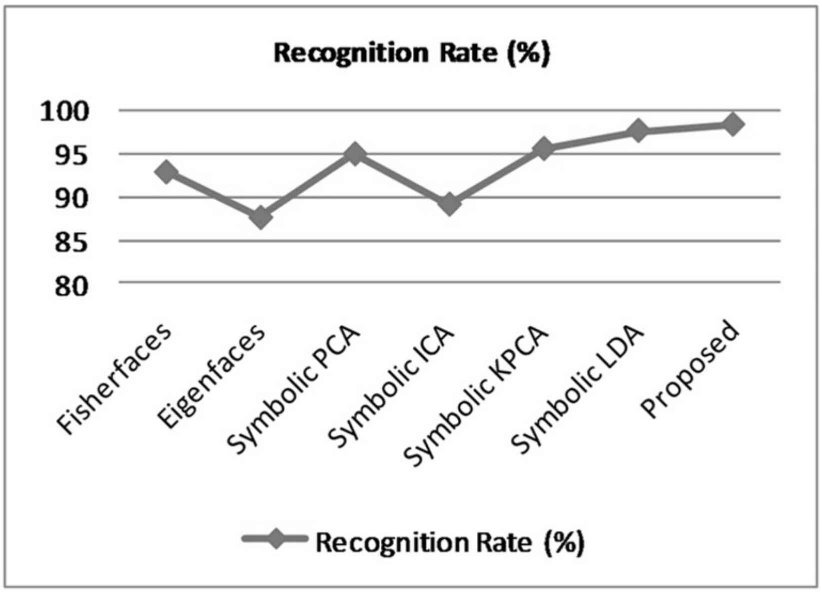


Figure 9. Accuracy comparison of different techniques for the ORL database.

Methods	Training Time (Sec)	Resolution	Recognition Rate (%)
Fisherfaces	59	128x156	89.85
Eigenfaces	85	128x156	82.04
Symbolic PCA	35	128x156	91.15
Symbolic ICA	43	128x156	92.00
Symbolic KPCA	98	128x156	94.55
Symbolic LDA	18	128x156	96.15
Proposed	31	128x156	96.66

Table 2. Comparison of the Yale database results [18].

Figure 10. Training time comparison for the Yale database.

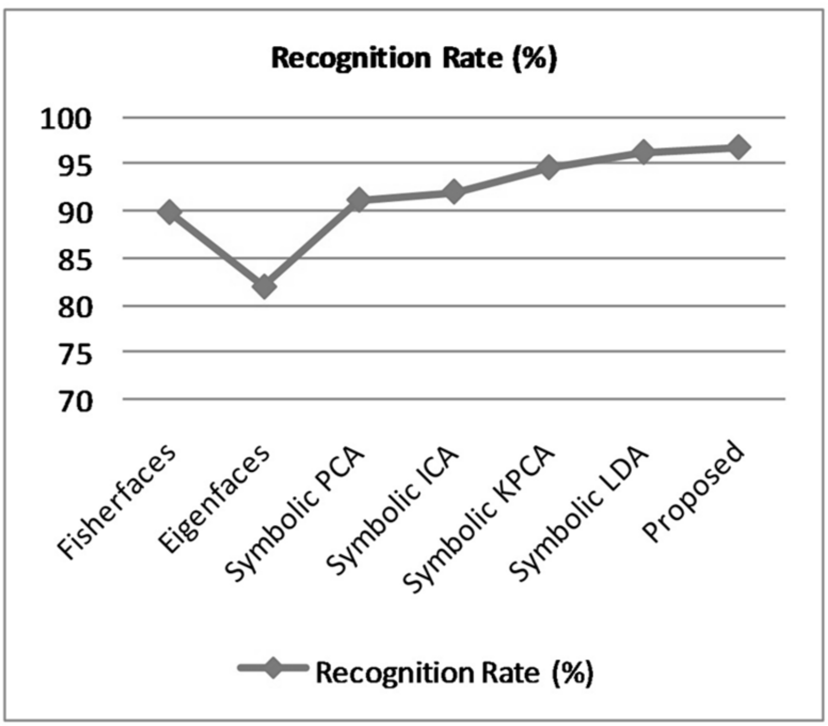


Figure 11. Recognition rate comparison of different techniques for the Yale database.

3.3 Recognition Time Analysis

The recognition time analysis of the proposed system is given in Table 3. For this purpose, a large database of 11,152 pictures was created containing the pictures from the ORL and Yale databases. Some pictures of different resolutions and backgrounds are also added to the database. The purpose of using different types of pictures is to check the system performance in different conditions. To conduct the experiment, different images were selected randomly from the database as test images which then went through pre-processing and recognition phases. Some of the results are shown in Table 3. The average time found for face recognition is approximately 142 milliseconds (0.14 seconds per image).

Conclusions and discussion

Recognition in the proposed technique is achieved using the signal compression technique. Data for building a database are first compressed and then coded in the form of vectors; therefore, each individual's data are recorded in the form of numbers or vectors. There are a number of advantages of using compression techniques for recognition systems especially in bandwidth consumption problems. Firstly, an individual's face is decomposed into blocks and then processed by block-based transforms, i.e., the DCT transform. Then transformed data is coded and used for making vectors. Afterwards, the sum of these vectors is calculated which gives a hash value against each individual's entry. A number of such vectors from an individual's data constitute a Facebase of a particular individual. The efficient use of the block-based transform reduces the original data to high energy coefficients which are very few in number as compared with the original data. Making a hash value reduces the searching time efficiently and maximizes the recognition rate.

The main purpose of the proposed technique is fast face recognition so that it can be helpful to recognize a face in a large database of faces. The technique is focused on both searching time and recognition accuracy. On the basis of the experiments and the observed results, it has been shown that the proposed algorithm reduces both the search time required by the recognition algorithm and the training time required. This algorithm can be used in the future in the areas in which fast and rapid recognition is needed.

Pic ID	Facebase No	Length of FB	Found at Index	Pre-Processing Time (mSec)	Searching Time (mSec)	Total Time (mSec)
1	1	3761	216	135	6.34	142
100	1	3761	659	129	6.55	136
1000	4	1015	303	130	6.67	137
2000	3	1253	162	129	8.35	137
4000	5	911	176	125	8.6	134
5000	1	3761	2263	134	7.58	141
7000	3	1253	432	135	8.35	143
8000	1	3761	2884	136	7.06	143
10000	2	1737	893	127	8.94	136
11000	1	3761	2379	160	8.78	169
11152	7	741	258	136	8.24	144
Average Time						**142**

Table 3. Recognition time analysis Total Pictures = 11,152.
Legend:
FB: Facebase
mSec: milliseconds

References

[1] R. Chellappa, C.L. Wilson, and S. Sirohey, “Human and machine recognition of faces: A survey,” Proc. IEEE, vol. 83, pp. 705–740, 1995.

[2] H. Wechsler, P. Phillips, V. Bruce, F. Soulie, and T. Huang, Face Recognition: From Theoryto Applications, Springer-Verlag, 1996.

[3] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, “Face recognition: A literature survey”, CVL Technical Report, University of Maryland, 2000.

[4] S. Gong, S.J. McKenna, and A. Psarrou, Dynamic Vision: from Images to Face Recognition, Imperial College Press and World Scientific Publishing, 2000.

[5] Noor Zaman, Abnan Alam Khan, Mansoor uz Zafar Dawood, Abdul Aziz, “ Face Recognition Techniques (FRT) based on Face ratio under controlled conditions”, IEEE Proceeding of International Symposium on Biometrics & Security Techniques 2008. PP 3-8.

[6] V.V. Starovoitov, D.I Samal, D.V. Briliuk, “Three Approaches For Face Recognition”, The 6-th International Conference on Pattern Recognition and Image Analysis October 21-26, 2002,Velikiy Novgorod, Russia, pp. 707-711.

[7] Feng Yang and Adam Krzyżak,“Face Recognition under Significant Pose Variation”, Canadian Conference (IEEE), Electrical and Computer Engineering, 2007. CCECE 2007 page(s): 1313-1316.

[8]K.-H. Cheung, A. Kong, D. Zhang, M. Kamel and J. You. Revealing the secret of Face Hashing in Intl. Conf. on Biometrics, volume 3832 of LNCS, pages 106-112, 2006.

[9] Carlos Gomez-Rodriguez, David Weir, John Carroll, “Parsing mildly non-projective dependency structures”, Cognitive science research papers, US university of Sussex.

[10] Trevor Darrell, Neal Checka, Alice Oh and Louis Phillipe Morency, “Exploring Vision Based Interfaces: How to use Your head in Dual Pointing Tasks”, AI Memo 2002-001 January 2002.

[11] Junwen Wu; Trivedi, M.M, “A Binary Tree for Probability Learning in Eye Detection”, Computer Vision and Pattern Recognition, 2005 IEEE Computer Society Conference on 25-25 June 2005 Page(s):170 – 170.

[12] Moisture Discriminating optics, “Real world advantages of Moisture Discriminating optics in live scan digital finger print imaging”, issued October 23, 2002 identix incorporated.

[13] Qinfeng Shi, Hanxi Li, Chunhua Shen, Rapid Face Recognition Using Hashing, In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 10), San Francisco, USA, June 13-18, 2010.

[14] Jain, A.K. Fundamentals of Digital Image Processing, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[15] Pennebaker, W.B., and J.L. Mitchell. JPEG Still Image Data Compression Standard, New York, NY: Van Nostrand Reinhold, 1993. Chapter 4.

[16] Hartigan, J. A. (1975). Clustering Algorithms. Wiley. MR0405726. ISBN 0-471-35645-X.

[17] “AT&T "The Database of Faces" (formerly "The ORL Database of Faces")”.

[18] P.S. Hiremath and Prabhakar C.J.,” Face Recognition Technique Using Symbolic Linear Discriminant Analysis Method”, P. Kalra and S. Peleg (Eds.): ICVGIP 2006, LNCS 4338, pp. 641 – 649, 2006.

[19] The Yale Face Database, http://cvc.yale.edu/projects/yalefaces/yalefaces.html, Last visited on 28th July, 2009.

[20] David C. L. Ngo, Andrew B. J. Teoh and Alwyn Goh , “Eigenspace-Based Face Hashing.”, Biometric Authentication, First International Conference, ICBA 2004, Hong Kong, China, July 15-17, 2004, Proceedings 2004.

[21] Ngo, D.C.L.; Teoh, A.B.J.; Goh, A., “Biometric Hash: High-Confidence Face Recognition” IEEE Transactions on Circuits and Systems for Video Technology, June 2006, Volume: 16 Issue: 6, page(s): 771 – 775.