Journal of Advanced Computer Science and Technology, *1 (3) (2012) 134-166*
©Science Publishing Corporation
www.sciencepubco.com/index.php/JACST

DFT domain Feature Extraction using Edge-based Scale Normalization for Enhanced Face Recognition

K Manikantan[1,*], **S Ramachandran**[2,†]

[1]Department of Electronics and Communication Engineering,
M S Ramaiah Institute of Technology, Bangalore, Karnataka, India 560054

[2]Department of Electronics and Communication Engineering,
S J B Institute of Technology, Bangalore, Karnataka, India 560060

Abstract

This paper proposes a novel preprocessing technique in order to improve the performance of a Face Recognition (FR) system. The proposed Edge-based Scale Normalization (ESN) process involves the use of scale normalization along with edge detection as a preprocessing technique in order to eliminate unwanted background details in face images. Feature extraction is performed on the preprocessed image using Discrete Fourier Transform (DFT). The DFT spectrums of these images extract the low frequency coefficients required for face recognition. These important features are selected through a rhombus-shaped mask around the center of the DFT spectrum. Further optimization in feature selection is achieved through Binary Particle Swarm Optimization (BPSO) technique. Experimental results, obtained by applying the proposed algorithm on Cambridge ORL and Extended YaleB face databases, show that the proposed system outperforms other FR systems. A significant increase in the recognition rate and a substantial reduction in the number of features is observed. Significant dimensionality reduction by more than 98.5% and improved recognition rate of 98% are achieved for both datasets.

[*]Corresponding author, E-mail: kmanikantan@msrit.edu
[†]E-mail: ramachandr@gmail.com

Keywords: *Binary Particle Swarm Optimization, Discrete Fourier Transform, Face Recognition, Feature Extraction, Scale Normalization.*

1 Introduction

Face Recognition (FR) is an automated process of matching a given face image with the corresponding person's image stored in a database. FR finds application in areas like surveillance and security, digital libraries and human-computer interfaces. FR is a fast emerging field in computer vision and pattern recognition, and new algorithms which are computationally efficient are being developed. Refs. [1, 2, 3, 4, 5] provide an exhaustive survey of various face recognition techniques in vogue currently.

Training and Recognition form the major stages in the FR process. Training involves preprocessing, feature extraction, feature selection and creation of a face gallery. Recognition on the other hand involves preprocessing of the test images and classification/identification of these images. One of the FR methods is by preprocessing the image to discard the redundant information, extracting the features from it, selecting the adequate features and then comparing them with the stored images in a database. Feature extraction and feature selection significantly reduce the amount of information to represent an image, thereby reducing computational time and cost. Further, feature selection improves classification accuracy of classifier and results in selection of more interpretable features [6, 7, 8, 9].

Successful, speedy and practically feasible FR method depends heavily on the choice of feature vectors used for classification and addressing the curse of image dimension. Dimension reduction and subspace analysis have been major research issues in learning based image analysis, detection and recognition [12, 13].

All face images possess high information redundancy and correlation which results in inefficiency in recognition. Transformation of facial images in frequency domain can be used to reduce image information redundancy because only a subset of transformed coefficients are necessary to preserve the most important features required for recognition. Earlier studies [14, 15] concluded that information in low frequency bands play a dominant role in face recognition as low frequency components contribute to the global description, while the high frequency components contribute to the finer details [16].

For enhanced face recognition, this paper proposes the following new ideas:

1. *Edge-based Scale Normalization* (ESN): This proposed preprocessing technique reduces the redundancy in images by removing a significant portion of background details and thereby results in an efficient feature extraction.

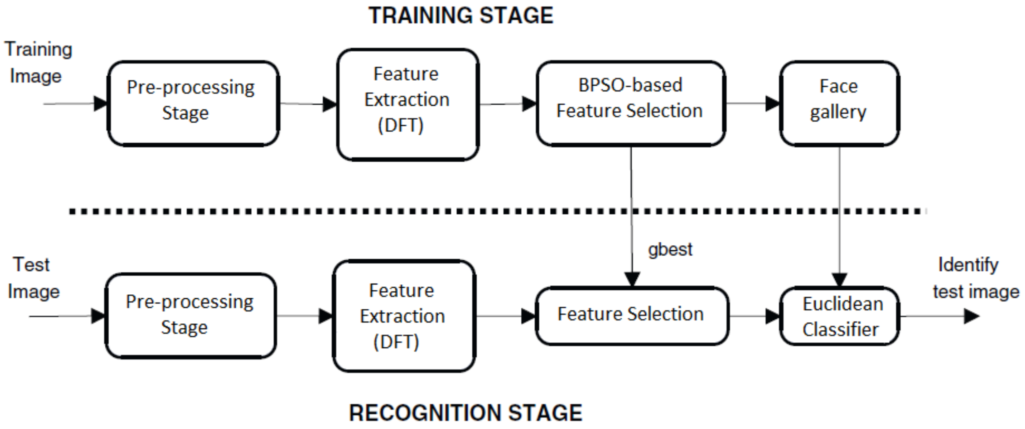


Figure 1: General block diagram of DFT based face recognition system.

2. *DFT Based Feature Extraction*: To address the challenges of *dimension reduction* and choice of *minimum size feature vector* for enhanced FR, we propose the Discrete Fourier Transform (DFT) based facial feature extraction. DFT spectrum of a face image has components of all frequencies, but the low frequencies concentrate around the center (after shifting), and as we move away from the spectrum-center, the magnitude gets smaller for higher frequencies. In order to evaluate the extent to which these centered features contribute towards face recognition rate, we propose a distinct selection method: *rhombus mask around the spectrum center*. This method extracts the best minimum feature vector from DFT spectrum. This method adequately represents the facial features required for better recognition. Moreover, this method improves the dimension reduction for better computation speed. The general block diagram of the proposed DFT based FR model is shown in Fig. 1.

The rest of the paper is organized as follows: Section 2 describes the basic image preprocessing methodologies which provide better image representation for further processing during recognition. Proposed Edge-based Scale Normalization (ESN) for background removal is described in Section 3. Discrete Fourier Transform for feature extraction is described in Section 4 along with rhombus extraction method. Feature selection using BPSO is explained in Section 5. Euclidean Classifier to determine Euclidean distances is briefly described in Section 6. Section 7 discusses Experimental Results obtained from the proposed FR model for the two benchmark datasets, ORL and Extended YaleB. Concluding remarks are presented in Section 8.

2 Fundamental Image Preprocessing Methodologies

2.1 Image resolution reduction

In face images, resolution reduction [17, 18] is one of the major contributors towards retaining suitable feature vector for recognition and dimension reduction, which enhances the computation speed and provides better image representation for further processing during recognition. In the proposed face recognition model, image resolution reduction is achieved through image *decimation*. Decimation algorithm is a novel form of obtaining desired resolution wherein the sub-sampling under the best circumstances is harmless to image as it scans through lines of pixels, averaging together pair of pixels or group of pixels according to the value of the decimation factor.

In decimation, a new value is calculated from a neighborhood of samples and these samples are replaced by the new value in the minimized image. Decimation is the process of eliminating data points from a data set.

In this model, face image decimation is achieved by convolving the face image with an averaging filter and then decimator is used to discard pixels according to decimation down scale factor as shown in Figure 2. By varying the decimation down scale factor, a *Gaussian Pyramid* (Figure 3) of varying resolution of face images is obtained. This is used in the FR model in frequency domain to evaluate the effects of reduced resolution on recognition rate, training time and recognition time.

In Figure 2, *I(i,j)* is an input image, *H* is the convolution averaging mask

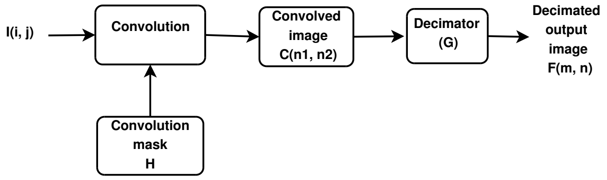


Figure 2: Image decimation process.

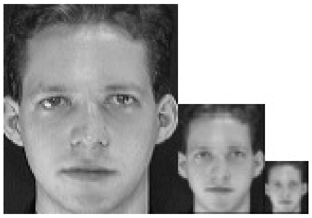


Figure 3: Original image and resized images using Gaussian pyramid with decimation factor df = 1 and 2.

and $C(n_1, n_2)$ is the convolved image without zero padding. *F(m,n)* is the decimated output image with reduced image resolution. That is,

$$F(m, n) = C(n_1G, n_2G) \tag{1}$$

$$G = 2^{df} \tag{2}$$

where,
df=*decimation down scale factor*
and $0 \leq m \leq (n_1/G)$, $0 \leq n \leq (n_2/G)$.

The resulting image is a reduced size mirror of the original image faithful in totality to the original but smaller in size. But resolution reduction is only in powers of 2 since this is a requirement for Gaussian pyramid to function properly.

2.2 Illumination normalization

Illumination is considered as an integral part of face recognition. The variance in illumination is nonlinear in nature and leads to greater complexities in face recognition. Hence it becomes necessary to neutralize the effect of illumination by illumination normalization. One of the techniques of illumination normalization is by taking *logarithmic transformation* of the image, given by Eq. 3 [9]. The logarithmic transformation helps in enhanding the low-gray levels and compressing the high-gray levels. It is well suited for shadowed and non-uniform image enhancement. It maps a narrow range of low intensity values in the input into a wider range of output levels whereas the opposite is true of higher values of input levels. It expands the values of dark pixels in an image while compressing the higher level values. Its most important characteristic is that it compresses the dynamic range of images with large variations in pixel values. The log transform is performed by the following equation.

$$s = c * \log{(1 + r)} \tag{3}$$

where r is the input image, s is the output image and c is a constant.

2.3 Edge detection

Enhancing and detecting edges form an integral part of detection algorithms as edges form the outline of an object. If the edges in an image can be identified accurately, all of the objects can be located and basic properties such as area, perimeter, and shape can be measured easily. In order to perform edge detection we use Laplacian of Gaussian (LoG)[9].

The Laplacian is a 2D isotropic measure of the 2^{nd} spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise. The LoG (Laplacian of Gaussian) operator normally takes a single gray level image as input and produces another gray level image as output.

3 Proposed Edge-based Scale Normalization (ESN)

Scale normalization is one of the fundamental steps in image preprocessing. It has been handled through different methods which include stretching algorithm [10, 11], where locations of several feature points like eyes, nose, or mouth is used and unnecessary part of the image is discarded. In our proposed face recognition system, ESN technique is used which extracts only the facial part of the image and discards the unnecessary and redundant portion including considerable amount of the background, and thereby contributes to a significant improvement in the recognition rate. Edge detection lays the ground for ESN as the edges of an image are the most important information which are detected and enhanced in order to increase the recognition rate significantly.

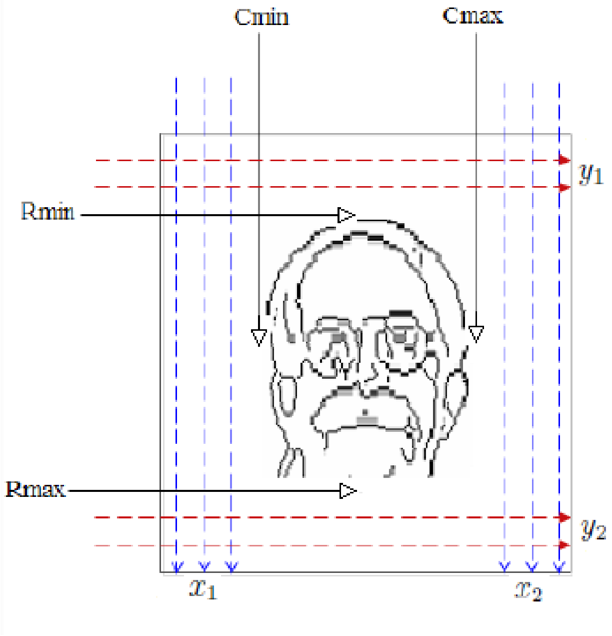
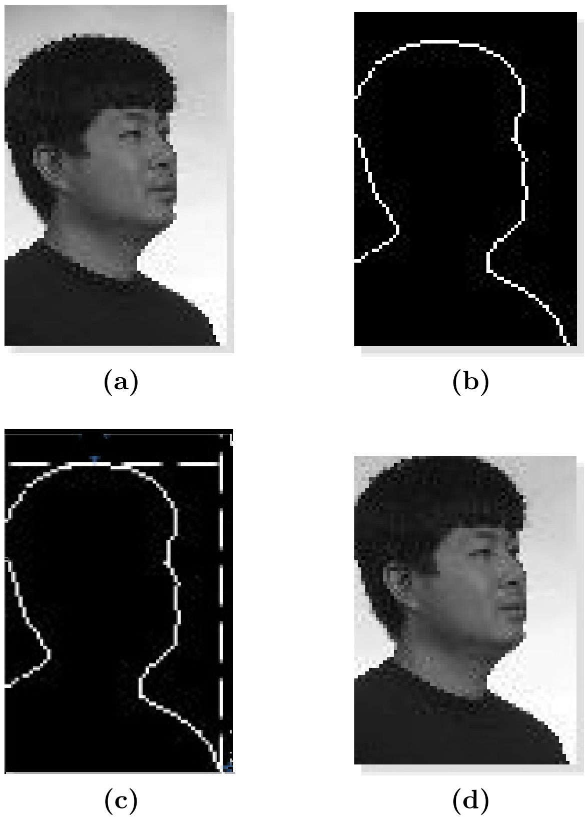


Figure 4: Classic Image scan for ESN.



(a) (b)

(c) (d)

Figure 5: (a) Original image (b) After edge detection (c) Location of extremal points (d) After ESN.

The image is first subjected to edge detection, the output of which is a binary image consisting of only the outer edge of the image. Edge detection is implemented using LoG operator which calculates the second spatial derivative of an image; this means that in areas where the image has a constant intensity (i.e. where the intensity gradient is zero), the LoG response will be zero. In the vicinity of a change in intensity, however, the LoG response will be positive on the darker side, and negative on the lighter side.

Once the binary edge detected image is obtained, it is traversed in a particular pattern in order to locate the *extremities* (Cmin, Cmax, Rmin, Rmax) of the image. For example, to find Cmin, we traverse the binary image pixel by pixel from top to bottom starting from the leftmost column, incrementing the column until we locate Cmin. Similarly, in order to find Cmax, we traverse the image in a similar manner starting from the rightmost column proceeding left. Fig. 4 shows the tracking of extremal points for ESN.

Let I denote the binary image of size M × N with (x,y) being the pixel values in the binary image. The horizontal pixel value x ranges from 1 to M and the vertical pixel value y ranges from 1 to N. The pixel by pixel traversal of (x,y) of the image can be shown using the following steps.

1. The first vertical traversal of the image begins from the leftmost pixel value of 1≤x<N, the x value is shifted to the right until it meets an edge to obtain x_1 = Cmin.
2. The second vertical traversal begins from the rightmost pixel value of x = N, the x value is shifted to left until it meets an edge to obtain x_2 = Cmax.
3. The first horizontal traversal starts from the topmost pixel value 1≤y<M, y value is shifted downwards until it meets an edge to obtain y_1 = Rmin.
4. The second horizontal traversal begins from the bottommost pixel value of y = M, y value is shifted upwards until it meets an edge to obtain y_2 = Rmax.
5. The coordinates x_1, x_2, y_1, y_2 form the extreme points of the binary image I.

These coordinates are then used to crop the sides of the image resulting in a scale normalized image. Fig. 5 shows the sequence of steps involved in edge-based scale normalization. This technique is applied to all the images in the database and a scale normalized database is obtained. Since the faces in different images vary in their position, orientation and size, the resulting scale normalized database would contain images of different sizes. Therefore these images are resized to a common size and then used in the training and testing sets.

4 Feature Extraction through Discrete Fourier Transform

The Fourier Transform [9] is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image. The Fourier Transform is used in a wide range of applications such as image analysis, image filtering, image reconstruction, image compression and recognition.

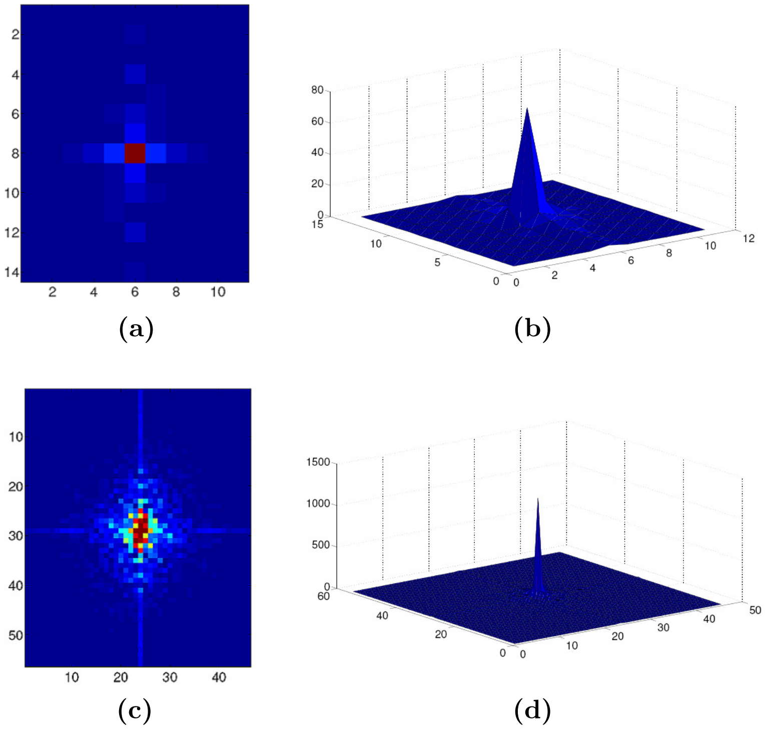
The *Discrete Fourier Transform* (DFT) is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image, that is, the image in the spatial and Fourier domain are of the same size.

The DFT equation is as follows:

$$G(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} g(m,n)e^{-j2\pi(\frac{mu}{M}+\frac{nv}{N})} \tag{4}$$

where *g(m,n)* is the image in the spatial domain and the exponential term is the basis function corresponding to each point *G(u,v)* in the Fourier space. The value of each point *G(u,v)* is obtained by multiplying the spatial image with the corresponding base function and summing the result.

The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the real and imaginary part or with the *magnitude* and *phase*, where *magnitude* determines the contribution of each component and *phase* determines which components are present. In the proposed face recognition model, only the *magnitude* of the Fourier



(a) **(b)**

(c) **(d)**

Figure 6: Spectrum with surface plots for the resized ORL image with a decimation factor of 3 and 1. (a)Spectrum with df = 3, (b)Surface plot with df = 3, (c)Spectrum with df = 1, (d)Surface plot with df = 3.

Transform is employed as it contains most of the information of the geometric structure of the spatial domain image. However, if we want to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, we must make sure to preserve both magnitude and phase of the Fourier image.

In this model, we propose the DFT based feature extraction since we want to access the geometric characteristics of a spatial domain image. Because the image in the Fourier domain is decomposed into its sinusoidal components, it is easy to examine or process certain frequencies of the image, thus influencing the geometric structure in the spatial domain.

One difficulty is that the nature of the Fourier transform produces an image which, at first, is difficult to interpret. The Fourier transform of an image gives the frequency components.

Unfortunately, the arrangement of the 2D Fourier transform places the low frequency components at the *corners* of the transform [19]. *A spatial transform is easier to visualize if the DC component is in the center with frequency increasing towards the edge of the image. This can be arranged either by rotating each of the four quadrants in the Fourier transform by* 180°*, or by reordering the original image to give a transform which shifts or translates the transform to the center. This is done using* **fftshift**. *This is aimed to improve visualiza-*

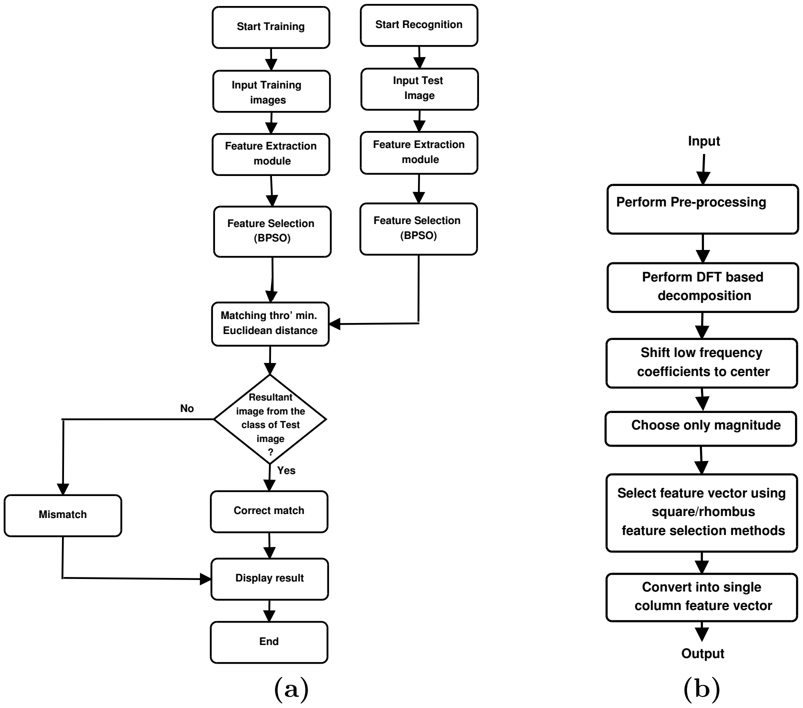

Start Training
Input Training images
Feature Extraction module
Feature Selection (BPSO)
Matching thro' min. Euclidean distance
Resultant image from the class of Test image ?
No
Yes
Mismatch
Correct match
Display result
End
Start Recognition
Input Test Image
Feature Extraction module
Feature Selection (BPSO)
(a)
Input
Perform Pre-processing
Perform DFT based decomposition
Shift low frequency coefficients to center
Choose only magnitude
Select feature vector using square/rhombus feature selection methods
Convert into single column feature vector
Output
(b)

Figure 7: Flowchart for DFT based face recognition algorithm. (a)Training and Testing module, (b)Feature extraction module.

tion and does not change any of the frequency domain information; changes only the way it is displayed. In face recognition, low frequency coefficients have the most important information which decrease as we move away from center of the translated Fourier spectrum of face image.

The transformed image shows that there are two dominating directions in the Fourier image, one passing vertically and one horizontally through the center. These originate from the regular patterns in the background of the original image. From the spectrum, we observe that the components with larger magnitudes, i.e., coefficients with lower frequencies are around the center. *In order to evaluate the extent which these centered features contribute to the face recognition rate, we propose a distinct extraction method: varying* ***rhombus*** *masks around the spectrum center. This will be compared at each stage with the conventional* ***square*** *mask around the spectrum center. The rhombus method extracts the best minimum feature vector from the DFT spectrum. BPSO further optimizes the feature vector. This method adequately represents the facial features required for better recognition. Moreover, this method improves the dimension reduction for better computation speed.* The algorithm developed for DFT based face recognition model may be summarized as shown in the flowchart of Figure 7.

4.1 Conventional square extraction method

As low frequency components of image spectrum provide maximum desired facial information for recognition, a square of k dimension around the center

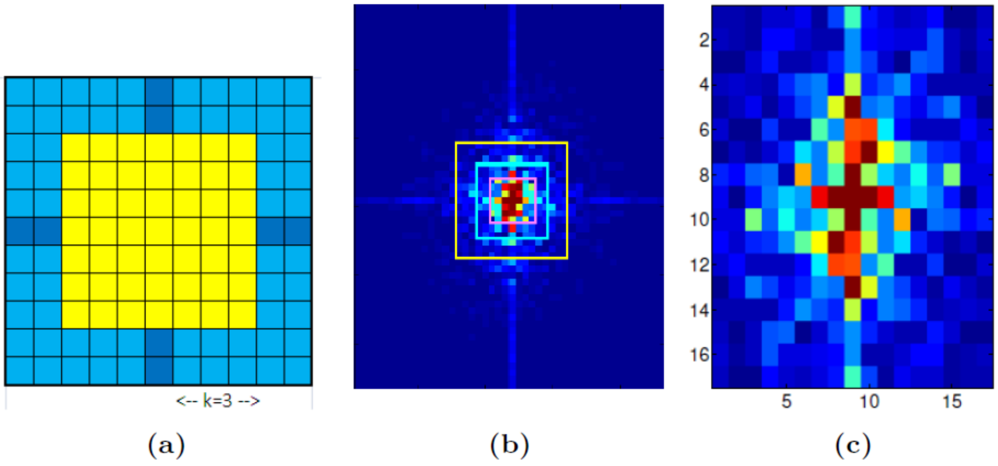


Figure 8: Square selection for ORL database. (a)Square selection with dimension 3 (k=3, No. of pixels = 49), (b)DFT spectrum with square selection around center, (c)Magnified Square selection with dimension 8 (k=8, No. of pixels = $(2k+1)^2$ = 289, before BPSO).

of the spectrum is selected and low frequency coefficients falling within this square are taken as feature vectors such that:

$$\begin{gathered} k \in (M, N) \\ 0 < k < M, \\ 0 < k < N \end{gathered}$$

where k=dimension of square (k is the no. of pixels from spectrum center to the square edge along horizontal and vertical directions as shown in Figure 8(a)), and $M \times N$ is the order of DFT spectrum matrix. Feature vectors with different square dimensions are retained against each image in dataset.

Figure 8(b) represents the square selection of three different dimensions and a corresponding plot for k=8 is shown in Figure 8(c). Here, coefficient selection is made using low frequency components falling within the square of specified length.

All the images in ORL database are of resolution 112 × 92 pixels. By applying square selection method with dimension k=8, only $(2k + 1)^2 = 289$ pixels are selected which are further reduced to 161 by BPSO, and only these are used for recognition purposes, rest being discarded, *which is a considerable dimension reduction by* 98.4%.

It is desired to achieve best recognition rate (RR) with smallest value of dimension k and largest value of decimation factor df. Experiments were carried out to find these optimum values of k and df which reveal that *for each dataset Recognition Rate reaches maximum for a specific df and k.* Images with a specific image resolution are loaded as input to the FR system. Dimension of square is defined as second input. A square with a specific length is marked around the center of DFT spectrum of face image. Coefficients falling within this square are calculated and after passing through BPSO feature selection, are retained as a feature vector. This single column feature vector against each image used in the training process of model is stored as a reference. Same procedure is repeated with the next dimension of the square.

4.2 Proposed rhombus extraction method

In this proposed method, coefficient selection is made through a *rhombus-shaped mask* around the center instead of the square. The two diagonals of each rhombus are equal. A rhombus has a specific diagonal length (2k+1), where k is the number of coefficients from the center of the DFT spectrum to each of the rhombus corners. Coefficients falling within this rhombus are calculated and after passing through BPSO feature selection, are retained as a feature vector. Here coefficient selection is made by using the low frequency components falling within the rhombus of the specified length. Dimension reduction is achieved in this process as all coefficients except within rhombus are discarded (See Figure

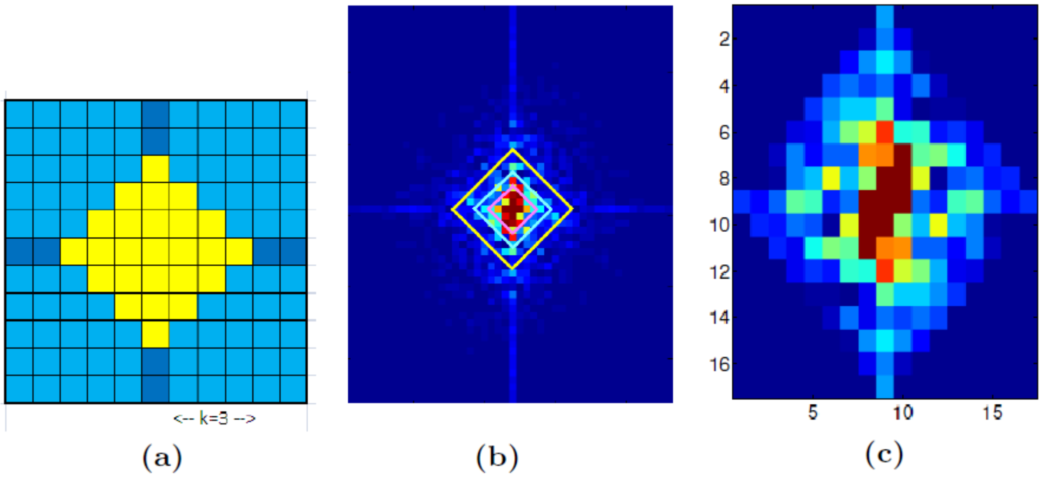


Figure 9: Rhombus selection for ORL database. (a)Rhombus selection with dimension 3 (k=3, No. of pixels = 25), (b)DFT spectrum with rhombus selection around center, (c)Magnified Rhombus selection with dimension 8 (k=8, No. of pixels = $(2k+1)^2/2 = 145$, before BPSO).

9(a)). Feature vectors with different rhombus dimensions are retained for later use for each image in the dataset.

Figure 9(b) shows the rhombus selection with three different diagonals. Figure 9(c) depicts the plot of rhombus selected coefficients.

In this procedure of feature extraction, face images with specific resolution are loaded as input to FR model and rhombus dimension k is defined as second input. A rhombus with specific diagonal length around center of transformed spectrum is identified and low frequency coefficients falling within this rhombus are extracted as feature vector.

5 Feature Selection

The feature selection process involves determination of a feature subset which can best represent a given feature set. This is accomplished by the use of binary version of the particle swarm optimization.

5.1 Binary particle swarm optimization (BPSO)

In BPSO [20, 21, 22, 23, 24], each particle's velocity is represented as the probability of bits to change from 0 to 1. The velocity updation equation is same as that of the continuous PSO [25]. The velocity is constrained by using a sigmoid function as given below :

$$sig(v_{id}) = 1/(1 + e^{-v_{id}}) \tag{5}$$

The new position of the particle is obtained as follows:

$$x_{id} = \begin{cases} 0, rand() \geq sig(v_{id}) \\ 1, otherwise \end{cases} \tag{6}$$

where rand() is a uniform random number in the range 0 to 1.

If the particle's position is set to 1, then the corresponding feature is selected. Otherwise, the feature is discarded. Thus, BPSO plays a major role in minimizing the overall feature set to a smaller feature subset. In this paper, the parameters of BPSO are modified to accomplish the aim of reducing the feature subset at a faster and efficient pace. The parameters of BPSO are shown in Table 1.

5.2 Fitness function

As the particles are iterated, the quality of the feature subset obtained is ensured to be optimum by checking its ability to maximize the class separation. The fitness value evaluated is the metric that is maximized as the iterations progress. Let C_1, C_2,.. C_N be N number of classes having I_1, I_2,.. I_L number of images in each respectively. Then the individual class mean is calculated as given below:

$$M_i = \frac{1}{T}\sum_{j=1}^{T} C_i^{(j)} \tag{7}$$

where
i varies from 1,2,....N.
T is the number of training images in each class.

The class means thus obtained are used to evaluate the grand mean as

Table 1: Parameter setting for BPSO.

Parameter Setting	
Swarm size	30
c_1 (cognitive parameter)	1.8
c_2 (social parameter)	2
inertia weight	exponentially decreasing function [26]
Iterations	100

given below:

$$M_o = \frac{1}{N}\sum_{i=1}^{N} M_i \tag{8}$$

The fitness function used is as follows:

$$F = \sqrt{\sum_{i=1}^{N} (M_i - M_o)^2} \tag{9}$$

6 Euclidean Classifier

The feature subset representing each image is the face gallery that is used for similarity measurement in the recognition stage. The test image is also reduced in dimension in order to have the same set of features as the images in the gallery(training images). The P-dimensional Euclidean distance [7, 8] between them is calculated as given below:

$$F = \sqrt{\sum_{i=1}^{P} (f_i - t_i)^2} \tag{10}$$

where f_i is the features of the image in the gallery, t_i is the selected features of the test image.

7 Discussion of Proposed Face Recognition Systems and Experimental Results

The schematic diagram presented earlier in Figure 1 shows the various steps involved in processing an input image. *Recognition rate* is used as a performance metric. If there are t test images and if c images are correctly recognized, then RR is c/t. For example, consider the ORL face database containing data of 40 subjects with 10 images per subject (4 images for training and remaining for testing). For each test image, let d_{min} represent the minimum Euclidean distance among the 160 distances calculated, i.e., 4 training images/subject x number of subjects. If d_{min} is found to belong to the proper subject out of 40 subjects, then it represents a correct recognition. This process is repeated for all 240 test images. Thus RR = $c/240$, where c represents the sum of all correct recognitions.

Cambridge ORL and Extended YaleB face databases are used to evaluate the changes in RR due to variation in image resolution and feature vector length.

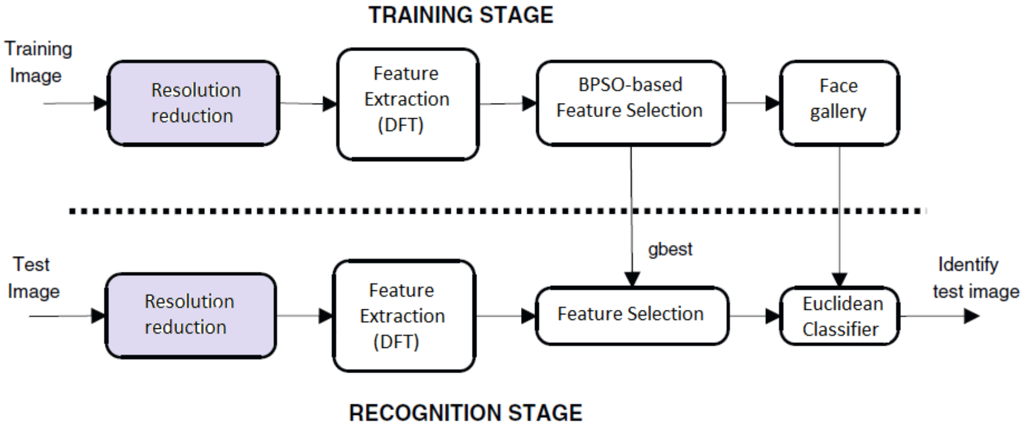


Figure 10: Block diagram of DFT based face recognition system for ORL database.

ORL [27] consists of a set of face images taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All images were taken against a uniform background with all subjects in an upright, frontal position. This is composed of 400 face images with 10 different images per person, a sample of which is presented in Figure 11. All images are gray scale and size of each image is 112 × 92 pixels. The block diagram of the proposed DFT based FR system for ORL database is shown in Fig. 10.

Extended YaleB [28, 29, 30, 31] is the most challenging face database for illumination-robust recognition as the lighting directions vary from left 130° to right 130°. This contains 16,128 images of 28 human subjects with 9 poses under various illumination conditions 12, 13, 14.

All images are gray scale and size of each image is 640 × 80 pixels.

Because of huge illumination variations, YaleB database needs illumination normalization and edge detection. The various preprocessing steps involved in handling YaleB images are shown in Fig. 15. The proposed FR model for YaleB database is shown in Fig. 16.

We have taken a 'Trainset : Testset' ratio of 3:5 in YaleB as explained below [Tables 2a, 2b, 3a, 3b].

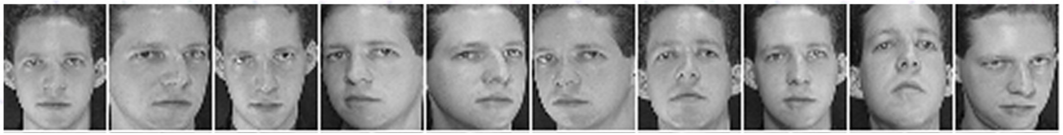
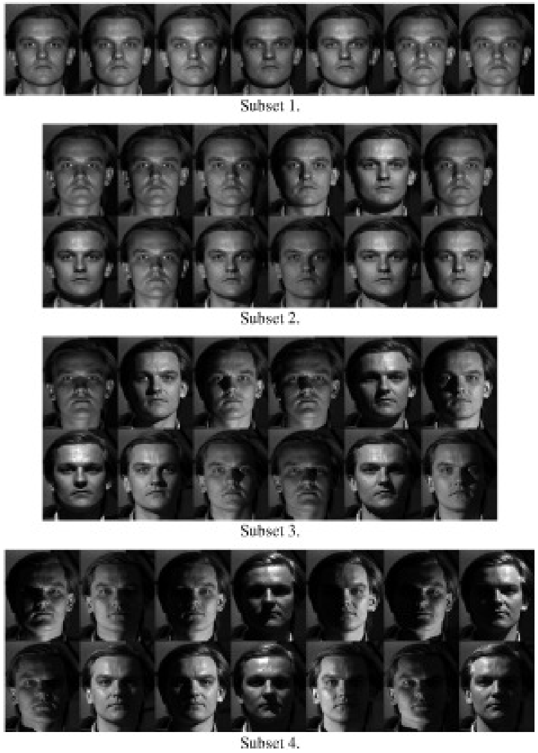


Figure 11: Sample images of a single individual from ORL Database.


Subset 1.

Subset 2.

Subset 3.

Subset 4.

Figure 12: Example images of a single individual in frontal pose from the YaleB Database showing the variability due to illumination. The images have been divided into four subsets according to the angle the light source direction makes with the camera axis - Subset 1 (up to 12°), Subset 2 (up to 25°), Subset 3 (up to 50°), and Subset 4 (up to 77°).

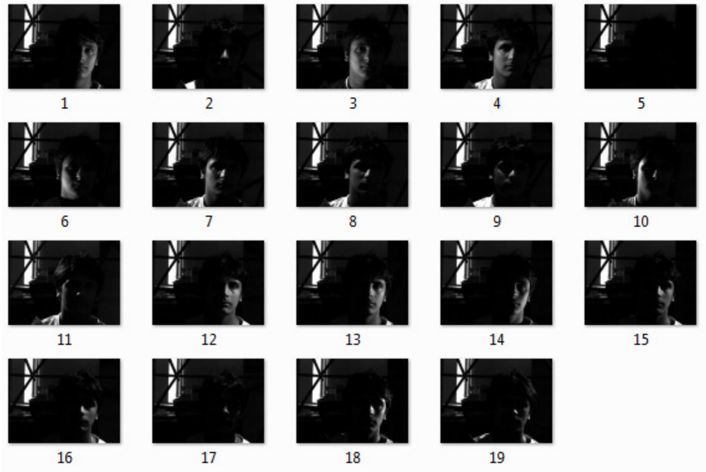


Figure 13: Example images of a single individual from the YaleB Database of Subset 5 ($>$ 77°).

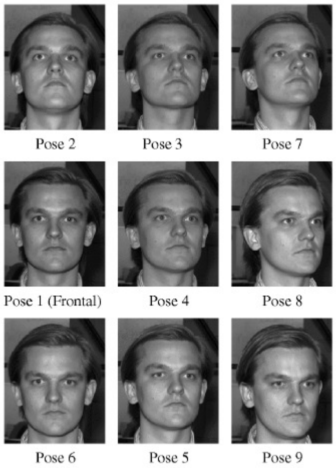


Figure 14: Example images of a single individual, one from each of the nine different poses in the YaleB Database.

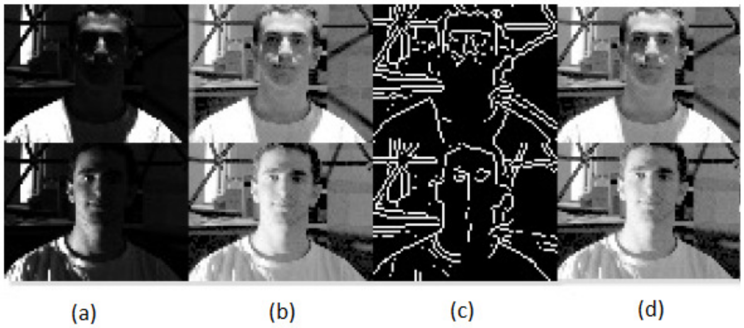


Figure 15: Preprocessing steps for YaleB images. (a)Sample image from Subset 5 Pose 0 of YaleB (b) After log transform (c) After edge detection (d) After ESN.

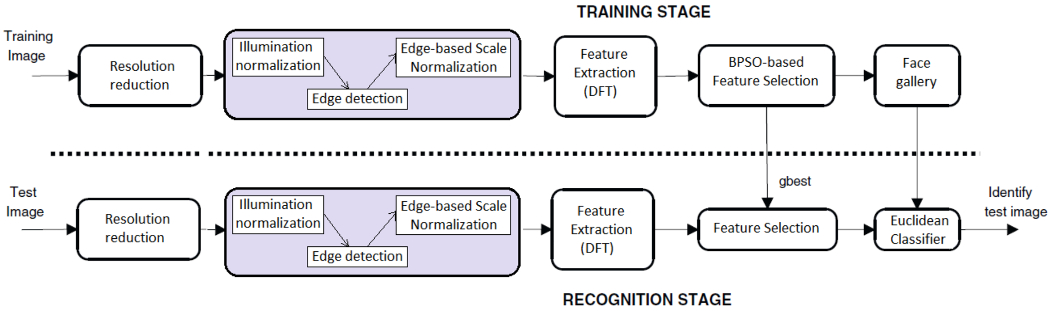


Figure 16: Proposed block diagram of DFT based face recognition system for YaleB database.

Table 2: (a)3 Training images per pose., (b)Trainset and Testset of 28 subjects.

(a)

	Train	Test
P0	3	61
P1	3	61
..	..	..
..	..	..
P8	3	61
Total	27	549

(b)

	Trainset	Testset
	P0 P1 .... P8	P0 P1 .. P8
S1	3 + 3 + .. + 3 = 27	61 61 ... 61 = 549
S2	3 + 3 + .. + 3 = 27	61 61 ... 61 = 549
S3	3 + 3 + .. + 3 = 27	61 61 ... 61 = 549
..	..	..
..	..	..
S28	3 + 3 + .. + 3 = 27	61 61 ... 61 = 549

Table 3: (a)27 Training images., (b)Total Trainset and Testset.

(a)

	Trainset	Testset
	P0 P1 .... P8	P0
S1	3 + 3 + .. + 3 = 27	61
S2	3 + 3 + .. + 3 = 27	61
S3	3 + 3 + .. + 3 = 27	61
..	..	..
..	..	..
S28	3 + 3 + .. + 3 = 27	61

(b)

	Trainset	Testset
S1	27	61
S2	27	61
S3	27	61
..	..	..
..	..	..
S28	27	61
Total	756	1708

Out of 64 illuminations per pose, we have taken *randomly* 3 images for training and the remaining 61 for testing as shown in Table 2a. The system can be sufficiently trained with 3 images. Table 2b presents the training and test set of various subjects. For training, we use all 27 images per person; but testing is done *posewise* as shown in Table 3a. For *pose 0*, Trainset and Testset will be as depicted in Table 3b. Thus training is done using *all* poses while testing is done per pose. Hence recognition rate is found per pose. Out of 61 images in the Testset, we have done testing for 5 images only (selected randomly). Hence we write the 'Trainset : Testset' ratio as *3:5*.

Presence of uneven background together with illumination and pose variances presents a challenge in face recognition for Extended YaleB database. Gaussian pyramid is first applied with a reduction factor of 3. This reduces the image resolution from 640 × 480 to 80 × 60. Then, an image enhancement technique in the form of logarithmic transformation is applied to neutralize the effect of illumination in the FR process. DFT is then applied and all the low frequency components of the spectrum are shifted to the center.

To find the tradeoff between dimension k and decimation factor df, that is, to find the optimum values of k and df for which the RR will be maximum, and training and recognition times minimum, results have been obtained for two different scenarios by using both square and rhombus feature extraction applied to both datasets as summarized in Table 4. (Note: Each of the following table contains the *total* recognition time; we can get the recognition time *per testing image* by dividing the indicated time by the number of testing images.)

Table 4: Salient features of Experiments. (Note: df = decimation factor, r = (Trainset : Testset) ratio, k = Dimension value.)

	Experiment-1a	*Experiment-1b*	*Experiment-1c*	*Experiment-2a*	*Experiment-2b*	*Experiment-2c*
	square	square	square	rhombus	rhombus	rhombus
df	Constant	Varying	Constant	Constant	Varying	Constant
k	Varying	Constant	Constant	Varying	Constant	Constant
r	Constant	Constant	Varying	Constant	Constant	Varying
ORL	Table 5, Figure 17	Table 7, Figure 19	Table 9, Figure 21	Table 5, Figure 17	Table 7, Figure 19	Table 9, Figure 21
	df = 1 k = 1 to 9 r = (4 : 6)	df = 0 to 2 k = 8 r = (4 : 6)	df = 1 k = 8 r = (1 : 9) to (9 : 1)	df = 1 k = 1 to 9 r = (4 : 6)	df = 0 to 2 k = 8 r = (4 : 6)	df = 1 k = 8 r = (1 : 9) to (9 : 1)
YaleB	Table 6, Figure 18	Table 8, Figure 20		Table 6, Figure 18	Table 8, Figure 20	
	df = 3 k = 1 to 17 r = (3 : 5)	df = 2 to 4 k = 8 r = (3 : 5)		df = 3 k = 1 to 17 r = (3 : 5)	df = 2 to 4 k = 8 r = (3 : 5)	

7.1 Experiment 1 : Square extraction method

7.1.1 Experiment-1a : Square extraction on ORL and YaleB with varying vector length and fixed resolution (df=1 for ORL and 3 for YaleB)

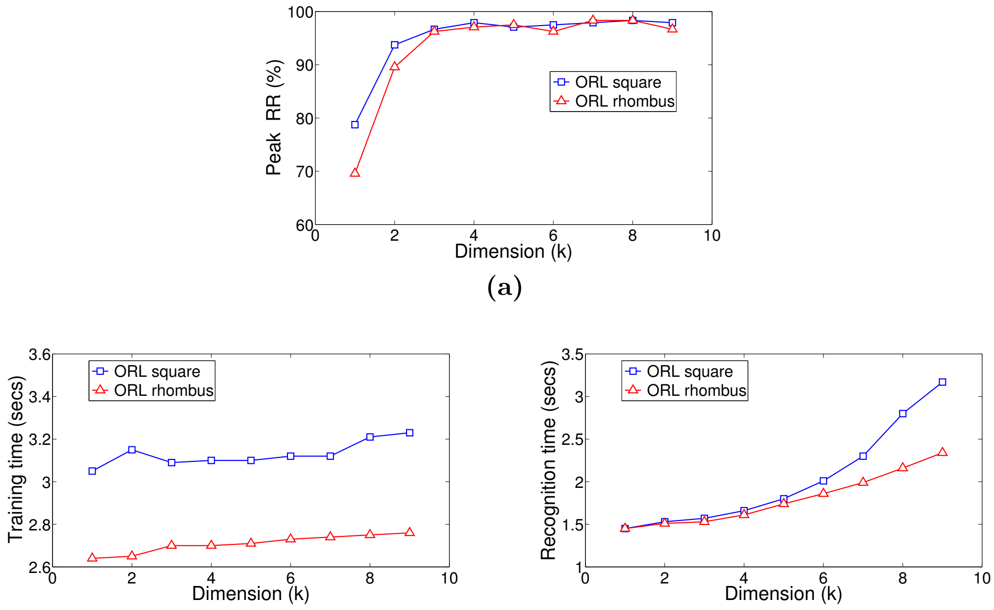
This set of experiments is based on *square* feature extraction method on ORL and YaleB datasets. Recognition results have been obtained by varying the feature vector length by changing the square dimensions. The results acquired on ORL database images (image resolution of 56 × 46 pixels obtained with df=1) against varying square dimensions are shown in Table 5 and Figure 17. Dimension length k means k pixels on right and k pixels on left of center pixel of spectrum. Hence Feature vector length = L^2, where L=(k+k+1). After BPSO, the number of features gets reduced by around 50%. Thus *significant dimensionality reduction is achieved by a combination of rhombus extraction and BPSO.*

Table 5 reveals that as dimension of feature vector increases from 1, the recognition rate improves and achieves best results at around k=8. This illustrates that a specific *minimum* number of feature coefficients are sufficient to represent the face image for recognition purposes. Similarly results have been obtained on Extended YaleB database at image resolution of 80 × 60 pixels corresponding to df=3 and are shown in Table 6 and Figure 18.

Results illustrated in Tables 5, 6 and Figures 17, 18 show that each database provides *best recognition rate around the square dimension of 8, but for different image resolutions.* Further, the second column of Table 5 and Table 6

Table 5: Results from ORL against varying square and rhombus dimensions and constant image resolution (for Expts. 1a & 2a). (Note: df=1, Train:Test = 4:6, No. of trials=5.)

k	Features		Peak RR (%)		Training time (Secs.)		Recognition time (Secs.)	
	square	rhombus	square	rhombus	square	rhombus	square	rhombus
1	6	4	78.8	69.6	3.1	2.6	1.5	1.5
2	16	12	93.8	89.6	3.2	2.7	1.5	1.5
3	33	19	96.7	96.3	3.1	2.7	1.6	1.5
4	54	29	97.9	97.1	3.1	2.7	1.7	1.6
5	70	40	97.1	97.5	3.1	2.7	1.8	1.7
6	98	56	97.5	96.3	3.1	2.7	2.0	1.9
7	122	69	97.9	98.3	3.1	2.7	2.3	2.0
8	161	90	98.3	98.3	3.2	2.8	2.8	2.2
9	193	103	97.9	96.7	3.2	2.8	3.2	2.3



(a)

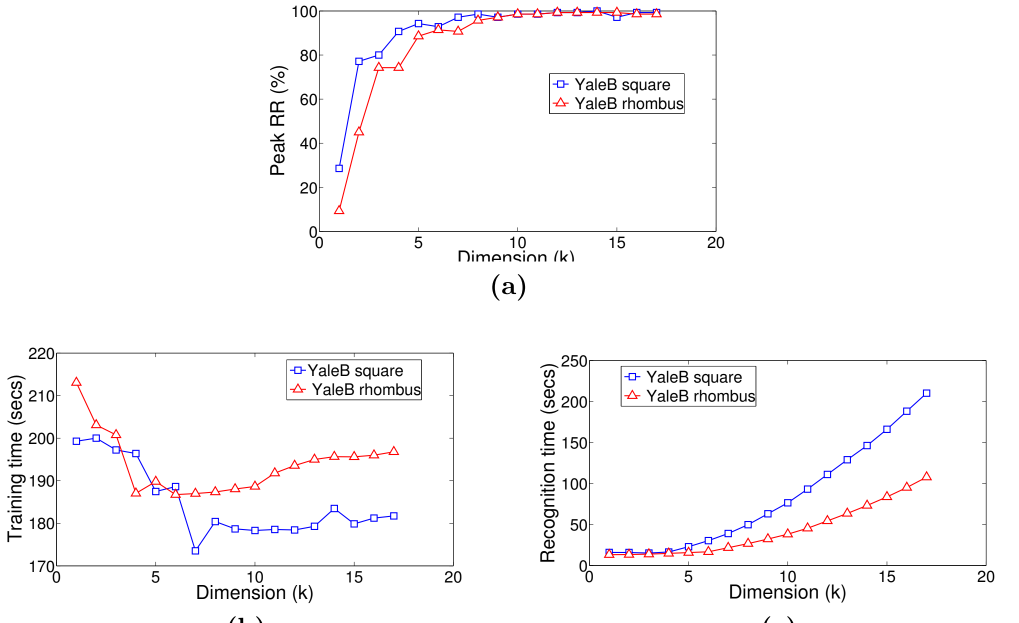
(b) **(c)**

Figure 17: Graphs showing the Results from ORL against varying square and rhombus dimensions and constant image resolution. (a)Peak RR, (b)Training time, (c)Recognition time.

indicate the total number of pixels used for recognition against a particular length of k. It shows the dimension reduction achieved through this feature extraction method. Also we observe that as k increases, the training time remains almost constant although the recognition time gradually increases.

Table 6: Results from YaleB for *pose 0* against varying square and rhombus dimensions and constant image resolution (for Expts. 1a & 2a). (Note: df=3, Train:Test = 3:5, No. of trials=1.)

k	Features		Peak RR (%)		Training time (Secs.)		Recognition time (Secs.)	
	square	rhombus	square	rhombus	square	rhombus	square	rhombus
1	7	4	28.6	9.3	199.3	213.1	15.9	13.1
2	21	11	77.1	45.0	200.0	203.1	15.8	13.3
3	37	20	80.0	74.3	197.2	200.8	15.2	13.8
4	58	30	90.7	74.3	196.4	187.0	16.4	14.7
5	81	41	94.3	88.6	187.5	189.8	22.9	15.7
6	107	59	92.9	91.4	188.6	186.7	30.3	16.7
7	137	68	97.1	90.7	173.5	187.0	39.0	21.7
8	171	89	98.6	95.7	180.4	187.4	49.7	26.7
9	211	109	97.1	97.1	178.7	188.1	63.0	32.2
10	249	126	98.6	98.6	178.3	188.7	76.4	38.1
11	294	151	98.6	98.6	178.6	191.8	93.1	45.4
12	341	184	99.3	99.3	178.4	193.6	110.0	54.4
13	394	208	99.3	99.3	179.3	195.0	128.9	63.4
14	459	237	100.0	99.3	183.5	195.7	146.2	73.2
15	511	271	97.1	99.3	179.9	195.6	166.1	83.6
16	576	296	99.3	98.6	181.2	196.0	188.1	95.0
17	642	334	99.3	98.6	181.7	196.8	210.1	107.7



(a)

(b)

(c)

Figure 18: Graphs showing the Results from YaleB against varying square and rhombus dimensions and constant image resolution. (a)Peak RR, (b)Training time, (c)Recognition time.

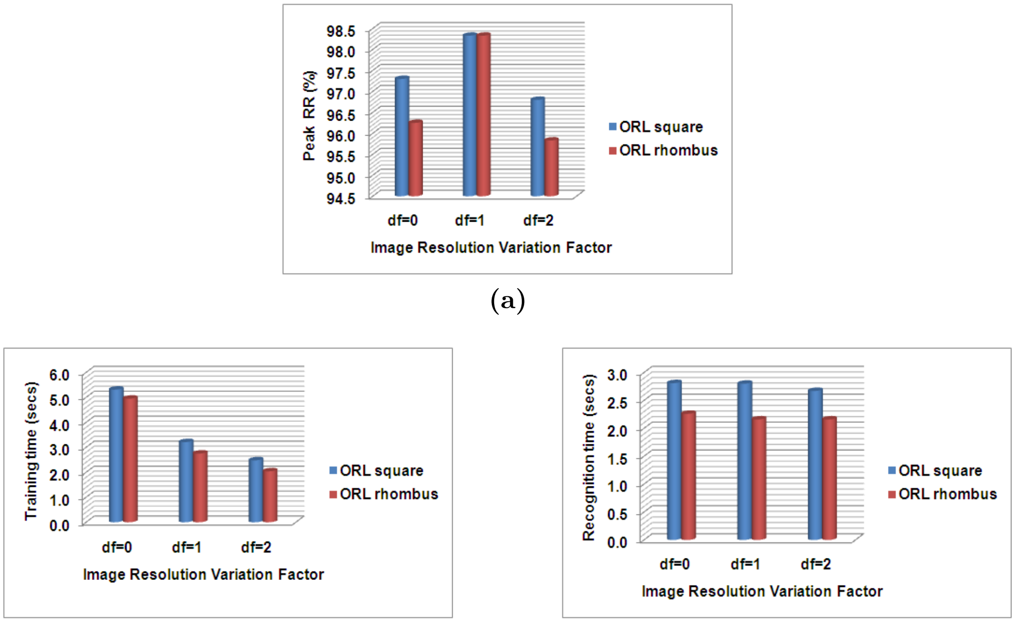
7.1.2 Experiment-1b : Square extraction on ORL and YaleB with fixed vector length (k=8 for both ORL and YaleB) and varying resolution

In this set of experiments, the feature vector dimension is kept constant at k=8 against the best recognition rate, and varying the image resolution. Results of ORL database tabulated in Tables 7 and 8, and Figs. 19 and 20, reflect that image resolution also affects the recognition rate. More importantly, the training and recognition times also *decrease* as df is increased from 0 to 2 for ORL and 2 to 4 for YaleB.

Results indicate that ORL provides best recognition rate of 98.3% at decimation down scale factor of 1 (image resolution of 56 × 46 pixels), and Ex-

Table 7: Results from ORL against varying image resolution and constant square and rhombus dimension (for Expts. 1b & 2b). (Note: k=8, ORL square features=161, ORL rhombus features=90, Train:Test = 4:6, No. of trials=5.)

df	Peak RR (%)		Training time (Secs.)		Recognition time (Secs.)	
	square	rhombus	square	rhombus	square	rhombus
0	97.3	96.3	5.3	5.0	2.8	2.3
1	98.3	98.3	3.2	2.8	2.8	2.2
2	96.8	95.8	2.5	2.0	2.7	2.2



(a)

(b)

(c)

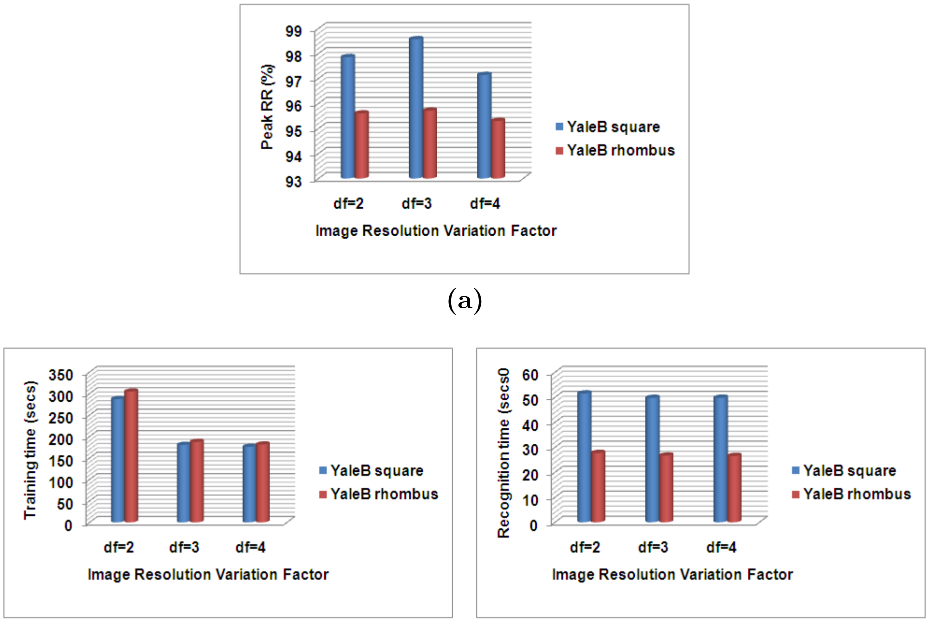
Figure 19: Graphs showing the Results from ORL against varying image resolution and constant square and rhombus dimension. (a)Peak RR, (b)Training time, (c)Recognition time.

tended YaleB achieved peak RR of 98.6% at decimation factor of 3 (image resolution of 80 × 40 pixels).

At the same time, it is evident from results that as the image resolution is reduced by increasing decimation factor df, it decreases the image dimension which ultimately reduces the training and recognition times of the model. This reduction in processing time enhances the model computation speed.

Table 8: Results from YaleB for *pose 0* against varying image resolution and constant square and rhombus dimension (for Expts. 1b & 2b). (Note: k=8, YaleB square features=171, YaleB rhombus features=89, Train:Test = 3:5, No. of trials=1.)

df	Peak RR (%)		Training time (Secs.)		Recognition time (Secs.)	
	square	rhombus	square	rhombus	square	rhombus
2	97.9	95.6	287.0	304.5	51.5	27.7
3	98.6	95.7	180.4	187.4	49.7	26.7
4	97.1	95.3	175.9	181.3	49.8	26.5



(a)

(b)

(c)

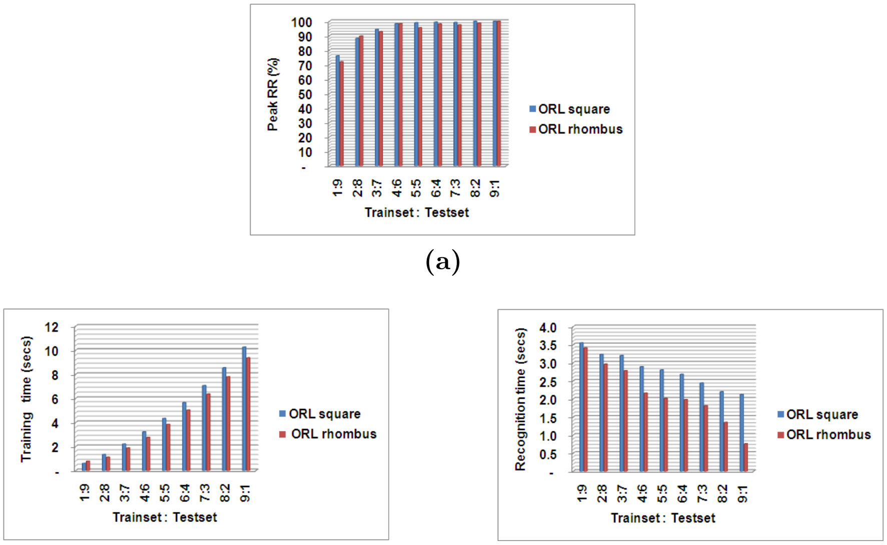
Figure 20: Graphs showing the Results from YaleB against varying image resolution and constant square and rhombus dimension. (a)Peak RR, (b)Training time, (c)Recognition time.

7.1.3 Experiment-1c : Square extraction on ORL with fixed vector length (k=8) and fixed resolution (df=1) but varying 'Trainset : Testset' ratio from 1:9 to 9:1

The effects of the ratio of Training set images to Testing set images on RR, training and testing times in ORL are shown in Table 9 and Fig. 21. These indicate that as the ratio rises to 4:6, the RR rises to 98.3%, and beyond this ratio the RR is almost constant at the maximum value.

Table 9: Results from ORL for varying 'Trainset : Testset' ratios but with constant square and rhombus dimension and constant image resolution (for Expts. 1c & 2c). (Note: df=1, k=8, ORL square features = 289, ORL rhombus features = 145, No. of trials=5.)

	Peak RR (%)		Training time (Secs.)		Recognition time (Secs.)	
Train set : Test set	square	rhombus	square	rhombus	square	rhombus
1 : 9	76.1	71.9	0.6	0.8	3.6	3.4
2 : 8	88.1	89.7	1.3	1.1	3.2	3.0
3 : 7	94.3	92.9	2.2	1.9	3.2	2.8
4 : 6	98.3	98.3	3.2	2.8	2.9	2.2
5 : 5	99.0	95.5	4.3	3.8	2.8	2.0
6 : 4	99.4	98.1	5.6	5.0	2.7	2.0
7 : 3	99.2	97.5	7.1	6.4	2.4	1.8
8 : 2	100.0	98.8	8.5	7.8	2.2	1.4
9 : 1	100.0	100.0	10.3	9.4	2.1	0.8



(a)

(b) (c)

Figure 21: Graphs from ORL for varying 'Trainset : Testset' ratios but with constant square and rhombus dimension and constant image resolution. (a)Peak RR, (b)Training time, (c)Recognition time.

Comparison with other face recognition systems

The number of training and testing images are varied in different ratios and the corresponding average recognition rates have been tabulated. To start with, an experiment is performed using ratios 4:6, 6:4, 7:3, 8:2 and, the respective average recognition rates for 10 trials have been obtained as tabulated in Table 10. The results reveal that the proposed method exhibits better performance when compared to the existing methods [32]. In the next phase, five images of each person are randomly selected for training and the remaining five for testing (ratio of 5:5). The average results for 20 trials are reported in Table 11a. We also compare our results with the previously reported results on this dataset [32, 33].

Finally 9 random images of each person are used for training and the remaining image for testing (ratio of 9:1). This is repeated for 10 trials and the average recognition rate obtained is compared with other existing techniques in Table 11b [34, 35, 36, 37, 38]. Graphs have been plotted for each of the

Table 10: Comparison of the *Average* recognition rates for 4 different ratios of training and testing images of ORL database. (Note: PCA : Principal Component Analysis, BP : Back Propagation, RBF : Radial Basis Function, LDA : Linear Discriminant Analysis.)

Ratio	PCA+BP [32]	PCA+RBF [32]	LDA+BP [32]	LDA+RBF [32]	Proposed Method (Square)
4:6	93.2%	94.8%	93.0%	93.1%	95.8%
6:4	97.0%	97.0%	97.1%	97.5%	98.3%
7:3	97.1%	97.4%	97.6%	98.0%	99.0%
8:2	98.0%	98.1%	98.2%	98.4%	99.3%

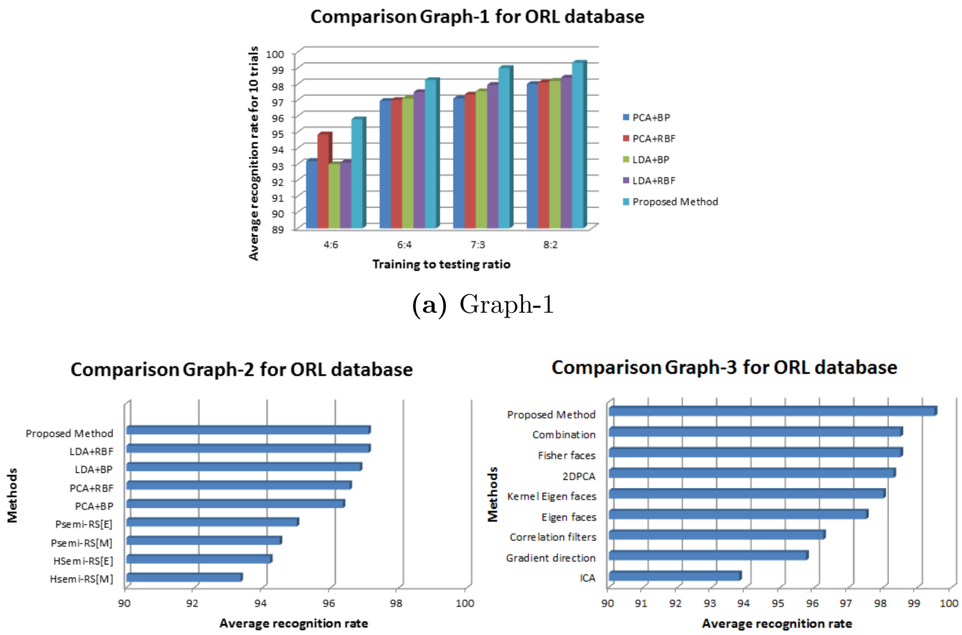
Table 11: (a)Comparison of the Average recognition rates for 5:5 training to testing ratio on ORL database. (Note: HSemi-RS : Hierarchical Semi-Random Subspace, PSemi-RS : Parallel Semi-Random Subspace, [M] : Mahcosine metric, [E] : Euclidean metric.), (b)Comparison of the Average recognition rates for 9:1 training to testing ratio on ORL database. (Note: ICA : Independent Component Analysis.)

(a)

Methods	Avg RR
HSemi-RS[M] [33]	93.4%
HSemi-RS[E] [33]	94.2%
PSemi-RS[M] [33]	94.5%
PSemi-RS[E] [33]	95.0%
PCA+BP [32]	96.4%
PCA+RBF [32]	96.6%
LDA+BP [32]	96.9%
LDA+RBF [32]	97.1%
Proposed method (Square)	97.1%

(b)

Methods	Avg RR
ICA [34]	93.8%
Gradient direction [35]	95.8%
Correlation filters [36]	96.3%
Eigen faces [34]	97.5%
Kernel Eigen faces [34]	98.0%
2DPCA [37]	98.3%
Fisher faces [34]	98.5%
Combination [38]	98.5%
Proposed method (Square)	99.5%



(a) Graph-1

(b) Graph-2

(c) Graph-3

Figure 22: Results on ORL database.

comparison tables as shown in Fig. 22a,22b and 22c.

On the whole, taking all possible ratios of training and testing images, a graph is presented in Fig. 21(a). This graph depicts top recognition rate for all the ratios (Training/testing) starting from 1:9 up to 9:1.

From Fig. 21, it is evident that the performance of the proposed FR system is not adversely affected by the change in the number of training and testing images except for the first two cases where the top recognition rates are 76.1% and 88.1%. It can be observed that the top recognition rate has reached a maximum of 100% for the last two cases. It can also be inferred that the recognition rates increase with the increase in the number of training images.

7.2 Experiment 2 : Rhombus extraction method

7.2.1 Experiment-2a : Rhombus extraction on ORL and YaleB with varying vector length and fixed resolution (df=1 for ORL and 3 for YaleB)

Here, *rhombus* feature extraction method has been used to choose the best feature vector for recognition. Image resolution is kept constant throughout at df=1 for ORL and df=3 for YaleB, and dimensions of rhombus are changed to assess their effects on recognition rate. Here, Feature vector length = $L^2/2$, where diagonal L=(k+k+1) and the two diagonals of the rhombus are assumed to be equal.

Tables 5 and 6 show the results of ORL and Extended YaleB databases. It is evident from the results that as the diagonal length of rhombus increases, data in feature vector is augmented with required coefficients for recognition. As the dimension of feature vector is increased beyond a certain level, the feature vector is polluted with the redundant data, and thereafter the recognition rate either remains the same or decreases slightly.

7.2.2 Experiment-2b : Rhombus extraction on ORL and YaleB with fixed vector length (k=8 for both ORL and YaleB) and varying resolution

In this part of experiments, image resolution is varied while keeping the feature vector dimension constant (where k=8 i.e., diagonal = 17). Figs. 19 and 20 present the results which reveal that as the image resolution is changed, the RR varies. ORL database yields the best RR of 98.3% at image resolution of 56 × 46 pixels. Similarly, experiments are carried out on YaleB database yielding peak RR of 95.7% at resolution of 80 × 60 pixels. For k=12, YaleB yields a RR of 99.3%.

7.2.3 Experiment-2c : Rhombus extraction on ORL with fixed vector length (k=8) and fixed resolution (df=1) but varying 'Trainset : Testset' ratio from 1:9 to 9:1

Table 9 and Fig. 21 show that RR reaches the maximum value once the 'Trainset:Testset' ratio approaches 4:6 for ORL rhombus. It is of importance to note that the testing time of ORL rhombus extraction is *significantly reduced* as compared to that of ORL square because of *half* the number of selected features. But the training time remains almost the *same* in both ORL square and ORL rhombus.

From the foregoing analysis of experimental results, we may derive the following conclusions.

1. From Figs. 17 and 18 for a given df, RR gradually increases as k increases and after a specific k (for both ORL and YaleB), RR either remains constant or suffers. This is because the increase in square/rhombus dimension produces information redundancy in feature vector. This illustrates that a specific number of feature coefficients are sufficient to represent the face image for recognition purposes.

2. The training time is almost independent of k and only depends on df. Of course, the training time for ORL and YaleB are different even for a given df because the original image sizes are different and the number of images in each dataset is different.

3. The recognition time of rhombus extraction is *significantly reduced* (ideally half) as compared to that of square because *half* the number of selected features only are present in the former. But the training time remains almost the *same* in both square and rhombus extractions.

8 Conclusions

For enhanced face recognition (FR) with minimal feature subset, a DFT-based feature extraction method along with feature selection using binary particle swarm optimization (BPSO) was proposed and applied to two benchmark face datasets, namely ORL and Extended YaleB. Firstly, to minimize the adverse effects of varying facial expressions and to provide improved recognition result with reduced computational complexity, a dimension reduction process using Gaussian pyramid was employed. Gaussian pyramid reduced the image resolution through image decimation and prepared the ground for DFT-based extraction. To reduce the redundancy in the images by removing significant portion of background details, a novel preprocessing technique, namely *Edge-based Scale Normalization* (ESN) was proposed, which resulted in efficient feature extraction.

After centering the low frequency coefficients in the DFT spectrum, a novel feature extraction method was proposed using *rhombus*-shaped mask around the center of the spectrum. The proposed FR system gave best performance for each dataset at a specific decimation down scale factor and a specific rhombus dimension k. The BPSO further adds to dimensionality reduction to the tune of 50%. For ORL (image size = 112 × 92 pixels), the FR system with rhombus mask (k=8) gave a peak Recognition Rate (RR) of 98.3% with only 90 features. For Extended YaleB (image size = 640 × 480 pixels), the FR system with rhombus mask (k=8) gave a peak RR of 95.7% with only 89 features. Thus, a *significant dimensionality reduction is achieved with improved recognition rate.*

Due to reduced features (almost 50%) in rhombus dimension as compared to the conventional square, the recognition time in each dataset is *substantially reduced* (by almost 50% as k increases beyond 10) for rhombus method as compared to square. For Extended YaleB, with k=8, the average recognition time (on a PC with Intel 2 Core 2.30 GHz CPU and 3 GB RAM) for rhombus was 26.7 seconds as compared to 49.7 seconds for square, when implemented using MATLAB [39]. But in both rhombus and square methods, the average training time was almost the same at around 180 seconds. These results indicate that *DFT based feature extraction with rhombus selection of the low frequency coefficients along with BPSO gave the best results, both in terms of increased recognition rate and reduced recognition time.* Thus, the proposed method evaluates the contribution of resolution reduction (through decimation) towards attaining improved recognition rate, and also a minimum feature vector dimension to reduce the computational complexity for faster processing. *The results obtained illustrate that a specific number of feature coefficients are sufficient to represent the face image for recognition purposes.*

The resolution reduction method used with rectangular images can be extended to *hexagonal* images. Hexagonal structure is expected to provide better image processing transformations and operations, like DFT, DCT and Edge detection, as compared to rectangular (and square) image processing. This work is currently under progress.

References

[1] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: A literature survey, ACM Computing Surveys 35 (2003) 399-458.

[2] Xuan Zou, J. Kittler, K. Messer, Illumination invariant face recognition: A survey, First IEEE International Conference on Biometrics: Theory, Applications, and Systems (2007).

[3] A.M. Patil, S.R. Kolhe, P.M. Patil, 2D Face recognition techniques: A survey, International Journal of Machine Intelligence 2 (2010) 74-83.

[4] Cha Zhang, Zhengyou Zhang, A survey of recent advances in face detection, Technical Report MSR-TR-2010-66 (2010).

[5] G. Shakhnarovich, B. Moghaddam, Face recognition in subspaces, Chapter 2, Handbook of Face Recognition, Springer-Verlag London Limited, 2011; 19-49.

[6] Chris Ding, Hanchuan Peng, Minimum redundancy feature selection from microarray gene expression data, Journal of Bioinformatics and Computational Biology 3 (2005) 185-205.

[7] O. Sigaud, S.W. Wilson, Learning classifier systems: A survey, Soft Computing 11 (2007) 1065-1078.

[8] Jianxin Wu, James M. Rehg, Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel, Proceedings of IEEE International Conference on Computer Vision (2005).

[9] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Third Edition, Prentice Hall, 2008; 133-135, 242-275, 485-488.

[10] D.J. Beymer, Face recognition under varying pose, Proceedings of IEEE International conference on Computer Vision and Pattern Recognition (1994) 756-761.

[11] D. Reisfeld, N. Arad, Y. Yeshurn, Normalization of face mages using few anchors, Proceedings of International conference on pattern recognition (1994) 761-763.

[12] M. Kirby, L. Sirovich, Application of the Karhunen-Loeve procedure for the charaterization of human faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1990) 103-108.

[13] Matthew A. Turk, Alex P. Pentland, Face recognition using eigenfaces, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1991).

[14] L.D. Harmon, The recognition of faces, Scientific American 229 (1973) 71-82.

[15] A.P. Ginsburg, Visual information processing based on spatial filters constrained by biological data, AMRL Technical Report (1991).

[16] J.H. Lail, P.C. Yuen, G.C. Feng, Face recognition using holistic Fourier invariant features, Pattern Recognition 34 (2001) 95-109.

[17] Jingdong Wang, Changshui Zhang, Heung-Yeung Shum, Face image resolution versus face recognition performance based on two global methods, Proceedings of Asia Conference on Computer Vision (2004).

[18] B.J. Boom, G.M. Beumer, L.J. Spreeuwers, R.N.J. Veldhuis, The effect of image resolution on the performance of a face recognition system, Proceedings of International Conference on Control, Automation, Robotics and Vision, ICARCV (2006).

[19] Mark S. Nixon, Alberto S. Aguado, Feature extraction and image processing, Second Edition, Academic Press publications, Elsevier Ltd, 2008.

[20] J. Kennedy, R. C. Eberhart, A discrete binary version of the particle swarm algorithm, Proceedings of World Multiconference on Systemics, Cybernetics and Informatics (1997) 4104-4109.

[21] Xu Jun, Huiyou Chang, The discrete binary version of the improved Particle swarm optimization algorithm, Proceedings of International Conference on Management and Service Science (2009) 1-6.

[22] C. Liu, H. Wechsler, Evolutionary pursuit and its application to face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 570-582.

[23] Xiaohui Hu, Yuhui Shi, R. C. Eberhart, Recent advances in particle swarm, Proceedings of IEEE Congress on Evolutionary Computation (2004).

[24] Maurice Clerc, Binary particle swarm optimisers: toolbox, derivations, and mathematical insights, Version 1, (2007).

[25] J. Kennedy, R. C. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks 4 (1995) 1942-1948.

[26] Guimin Chen, Jianyuan Jia, Qi Han, Geometrical profile optimization of elliptical flexure hinge using a modified particle swarm algorithm, ©Springer-Verlag Berlin Heidelberg 1 (2005) 533-542.

[27] ORL Database: `<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>`.

[28] Yale Univ. Face Database `<http://cvc.yale.edu/projects/yalefaces/yalefaces.html>`.

[29] Kuang-Chih Lee, Jeffrey Ho, David J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005).

[30] A S. Georghiades, P N. Belhumeur, D.J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 643-660.

[31] Shiguang Shan, Wen Gao, Bo Cao, Debin Zhao, Illumination normalization for robust face recognition against varying lighting conditions, Analysis and Modeling of faces and gestures, IEEE International Workshop (2003).

[32] K. Rama Linga Reddy, G.R. Babu, Lal Kishore, Face recognition based on Eigen features of multi scaled face components and an artificial neural network, Proceedings of International Conference and Exhibition on Biometrics Technology, Procedia Computer Science 2 (2010) 62-74.

[33] Yulian Zhu, Jun Liu, Songcan Chen, Y. Zhu, Semi-random subspace method for face recognition, Image and Vision Computing 27 (2009) 1358-1370.

[34] M. Yang, Kernel Eigenfaces vs. Kernel Fisherfaces: Face recognition using kernel methods, Proceedings of Fifth IEEE International Conference on Automatic Face and Gesture Recognition (2002).

[35] H F. Chen, P.N. Belhumeur, D.W. Jacobs, In search of illumination invariants, IEEE Conference on Computer Vision and Pattern Recognition 1 (2000) 254-261.

[36] B.V.K.V. Kumar, M. Savvides, C. Xie, Correlation pattern recognition for face recognition, Proceedings of IEEE 94 (2006) 1963-1976.

[37] Jian Yang, David Zhang, Alejandro F. Frangi, Jing-yu Yang, Two-Dimensional PCA: A new approach to appearance-based face representation and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 131-137.

[38] Raghuraman Gopalan, David Jacobs, Comparing and combining lighting insensitive approaches for face recognition, Computer Vision and Image Understanding 114 (2010) 135-145.

[39] `<http://www.mathworks.com>`.