**ECE 532 - DIGITAL IMAGE ANALYSIS**

**Term Project - Fall 2009**

# Comparative Study of PCA, ICA and LDA for Face Recognition

**Mahesh Bharath Keerthivasan**

**CONTENTS**

# 1. INTRODUCTION

Face recognition is an image analysis problem that involves the identification of human faces from a digital still image or a video sequence. The process involved consists of face segmentation, feature extraction and finally recognition or identification. The task of face recognition is complicated by factors such as variations in facial expressions, changes in illumination and the orientation of the face of the subject. Also, affecting the accuracy of identification is the background and the inherent noise present during image acquisition.

Numerous algorithms have been presented to solve this detection – recognition problem. At the face detection level, the neural network based approach [1] and the example based learning approach [2] have been proposed to detect faces from the image. For extraction of feature points, the edge based approach as given in [3] and the Gabor wavelet decomposition method [4] have been popular choices. To implement the final recognition problem methods such as the eigen face [5], Independent Component Analysis, Incremental PCA, Support Vector Machines (SVM) and the Linear Discriminant Analysis (LDA) [6] have been developed. Accompanying the algorithms, there are numerous Face Image Databases that provide the images for testing the algorithms. These databases contain images taken under various conditions such as varying lighting, varying facial expressions and varying face orientations.

In this project, a comparative study of three different statistical techniques for face recognition has been presented. The algorithms compared are: Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Linear Discriminant Analysis (LDA). Experiments have been carried out to evaluate the performance of these algorithms under varying conditions and based on the observed results a conclusion on the effectiveness of the algorithms has been presented.

# 2. PROJECT METHODOLOGY

## 2.1 Face Database Used

In this project two face databases have been employed for comparison of performance. The ATT Face Database [7] and the Indian Face Database (IFD) [8]. These two databases were chosen because the ATT database contains images with very small changes in orientation of images for each subject involved, while the IFD contains a set of 10 images for each subject where each image is oriented in a different angle compared to the other. These two databases provide a comprehensive dataset for testing the performance of the algorithms chosen.

The faces from each of the databases are pre-processed using the tool developed by the CSU Face Identification Evaluation System which is available on the internet [9]. This pre-processing consists of converting the files to JPEG format, resizing them to a smaller size to speed up computation and renaming the files.

A set of images from the IFD are shown below in Figure 1 and those from the ATT database are in Figure 2. As seen the images from IFD have a higher change in orientation angle than compared to that from the ATT database.

**Figure 1:** Images of a subject from the ATT Database

**Figure 2:** Images of a subject from the IFD

### 2.2 Software and Source Code Used

- The implementation has been carried out using MATLAB using the Image Processing Toolbox.

- The PCA based face recognition algorithm and the LDA based face recognition algorithm were self coded.

- The source code for the ICA based face recognition algorithm was obtained from the internet [10]. This source code was slightly modified in accordance with the analysis framework used.

### 2.3 Performance Analysis Framework Used

The quantitative measure used to evaluate the algorithms is the Recognition Accuracy and the Computation Time involved. This evaluation has been carried out using the Face Recognition Evaluator, an open source MATLAB interface [11].

The qualitative evaluation has been performed by testing the algorithms on the following:

1. Input images corrupted by Gaussian noise. This has been performed for three different noise standard deviations.
2. Input images blurred by a Gaussian filter. This has been performed for three different intensities of blur.
3. Modifying the algorithm parameters, specifically the number of eigenvectors used for computing the principal components and independent components, number of images used for training and testing the algorithms, and the results obtained from each of these

evaluations are tabulated and also indicated graphically.

## 3. ALGORITHM DESCRIPTION

The PCA, ICA and LDA algorithms form the class of "View-Based" face recognition approaches in which the face images are represented as a higher dimensionality vector space [12]. These algorithms use " statistical techniques to analyze the distribution of the object image vectors in the vector space, and derive an efficient and effective representation of the feature space ". The recognition process consists of comparing the test face data in the feature space with the set of feature space data obtained from the training face images. The similarity between these two is then measured to determine if the test face is present in the trained set of images. This measure is seen as the recognition ability of the algorithm.

These feature data are the weights that are obtained by projecting the face vector onto the basis vectors of the higher dimension face vector space. The process of determining these basis vectors is what differentiates these three algorithms. They use different statistical criteria for determining these basis vector.

Thus these three algorithms can be expressed in the general case as " a linear transformation of the image vector to the projection feature vector " as given by [12]

$$Y = W^T X \quad - (1)$$

where, W is the transformation matrix having dimensions K x1, Y is the K x N feature vector matrix and X is the higher dimension face vector obtained by representing all the face images into a single vector

$$X = \{ x_1, x_2, x_3, \ldots\ldots x_N \} \quad - (2)$$

where each $x_i$ is a face vector of dimension "n" obtained from the M x N dimension face image.

From each of the face image vectors the average face is subtracted. The average face is given by

$$AvgFace = (1 / N) \sum_N x_i \quad - (3)$$

$$X' = X - AvgFace \quad - (4)$$

For all the algorithms the initial step involves calculating the average face vector.

**Figure 3:** The average face obtained for a subset of the ATT database

### 3.1 PCA Face Recognition

The PCA face recognition algorithm as proposed by Turk and Pentland [13] and as discussed in [12] and [19] is described below. This method attempts to provide the features that determine how the face images differ from one another. These features are used to identify the face from a database of images.

The basis vectors for the PCA are given by the eigenvectors of the scatter matrix $S_T$ (covariance matrix) defined by

$$S_T = \sum_N (x_i - AvgFace)(x_i - AvgFace)^T \quad - (5)$$

The transformation matrix W is defined as the eigenvectors of $S_T$ corresponding to the K largest eigenvalues. This is so because the eigenvectors corresponding to the smaller eigenvalues represent noise [15] and so can be left put. The original image can be reconstructed by these K eigenvectors alone.

The input faces are represented in PCA in terms of the weights Y obtained by projecting them on the basis vectors using the representation of Equation (1).

The steps involved in implementing the algorithm are [19]:

Training the Faces:

1. Represent the faces in the database in terms of the vector X as shown in Equation (2).
2. Compute the average face AvgFace and subtract the AvgFace from the vector X as in Equation (4).
3. Compute the scatter matrix using Equation (5).
4. Compute the eigenvectors of the scatter matrix. Retain only the K eigenvectors corresponding to the K largest eigenvalues.
5. The first few eigenvectors are discarded as they represent illumination variations in the images [15]
6. The face image vector X' is then projected onto the eigenvectors using $Y = W^T . X'$. The values of Y are the feature vectors or weights of the images. Each of the face images can be represented in terms of these feature vectors.

Face Recognition :

1. For a test face image to be recognized, initially the normalization is performed by subtracting the average face from the image : $x' = x - AvgFace$
2. Then the face is projected on the basis vector : $Y = \sum_K W_i^T x' W_i$ . The Y gives the weight of the test image.
3. Compare the weight obtained Y with the values of weights recorded from the training phase. This comparison is performed using distance metrics such as the Euclidean distance, the L2 nor or the Mahalanobis distance. In this project, the Euclidean distance is used to determine the variation of the test image weight from the training set weights.
4. Based on the Euclidean distance the training face with the nearest weight is determined as

the best match for identification.

**Figure 4:** The first 5 eigenvectors obtained using a subset of the the ATT database.

### 3.2 LDA Face Recognition

In the PCA based recognition method the feature weights obtained identify the variations among the face images and this includes variations within the faces of the same subject too, under differing illumination and facial feature conditions. However, when we consider the different images of the same subject, these images have a lot of information in common and this common information is not utilized by the PCA technique which de-correlates the face images.

Hence, a class based approach has been proposed in which the faces of the same subjects are grouped into separate classes and the variations between the images of different classes are discriminated using the eigenvectors at the same time minimizing the covariance within the same class [6]. This method is called the Linear Discriminant Analysis. Here the low-energy discriminant features of the face are used to classify the images.

The LDA algorithm described below is the summary of the work published by [16] and from the discussion of [12] and [14].

The transformation matrix W for the LDA is given by the eigenvectors corresponding to the largest eigenvalues of the scatter matrix function ($S_W^{-1} \cdot S_B$), where $S_W$ is the within-class scatter matrix and $S_B$ is the between-class scatter matrix. These scatter matrices are defined as:

$$S_B = \sum_C N_i \,(\, ClasAvg_i - AvgFace \,)\,(\, ClasAvg_i - AvgFace \,)^T \quad - (6)$$

$$S_W = \sum_C \sum_{Xi} (\, x_k - ClasAvg_i \,)\,(\, x_k - ClasAvg_i \,)^T \quad - (7)$$

where, C is the number of distinct classes, $N_i$ is the number of images for each class i, $ClasAvg_i$ is the average face image of faces in class i, $X_i$ represents the face images that are in class i, AvgFace is the average face image for all images in the dataset.

The steps involved in implementing the algorithm are [16] :

Training the Faces:

1. Represent the faces in the database in terms of the vector X as shown in Equation (2).
2. Compute the average face AvgFace and subtract the AvgFace from the vector X as in Equation (4).
3. Classify the images based on the number of unique subjects involved. So the number of classes, C, will be the number of subjects who have been imaged.
3. Compute the scatter matrix using Equation (6) and (7).
4. Use PCA to reduce the dimension of the feature space to N – C. Let the eigenvectors obtained be $W_{PCA}$.
5. Project the scatter matrices onto this basis to obtain non-singular scatter matrices $S_{WN}$ and $S_{BN}$.
4. Compute the generalised eigenvectors of the non-singular scatter matrices $S_{WN}$ and $S_{BN}$ so as to satisfy the equation $S_B*W_{LDA} = S_W*W_{LDA}*D$, where D is the eigenvalue. Retain only the C-1 eigenvectors corresponding to the C-1 largest eigenvalues. This gives the basis vector $W_{LDA}$.
5. Then the image vector X is projected onto this basis vector and the weights of the image are computed.

Recognition of a new face is done similar to that of the PCA method. Here the given face is mornalized by subtracting the AvgFace and then the weights are calculated by projecting the image onto the basis vectors. Then the Euclidean measure is used as the similarity measure to determine the closest match for the test image with the face in the trained database.

### 3.3 ICA Face Recognition

The Independent Component Analysis can be viewed as a generalization of the PCA where " the higher order dependencies in the input data are minimized " [17] and the a " set of statistically independent basis vectors are determined " [18]. ICA for face recognition has been proposed under two architecture by Barlett et. al. [18]. The architecture 1 aimed at finding a set of statistically independent basis images while the architecture 2 finds a factorial code.

In this project, the architecture 1 has been used for evaluation of ICA based recognition. This process involves the following two initial steps :

1. The face images in the database are organized as a matrix X in which each row corresponds to an image.
2. The face database is processed to obtain a reduced dataset in order to reduce the computation efficiency of the ICA algorithm. The reduced dataset is obtained from the first m principal component (PC) eigenvectors of the image database. Hence the first step is applying PCA to determine the m PCs. The matrix containing these PCs is referred to as $P_m$.

Then the ICA algorithm is performed on the principal components using the mathematical procedure described in [18]. The source code for this implementation has been provided by the author, its is available at [10] and has been used for this project.

The process of ICA creates a set of statistically independent and non-Gaussian components and hence considers higher order moments instead of just the second order moments as in PCA. The basis vectors of a PCA representation are uncorrelated, however in the case of ICA the basis vectors are independent of each other. The ICA representation equals that of PCA only for the case of Gaussian distributed data, since then the conditions of uncorrelatedness and independence are both satisfied.

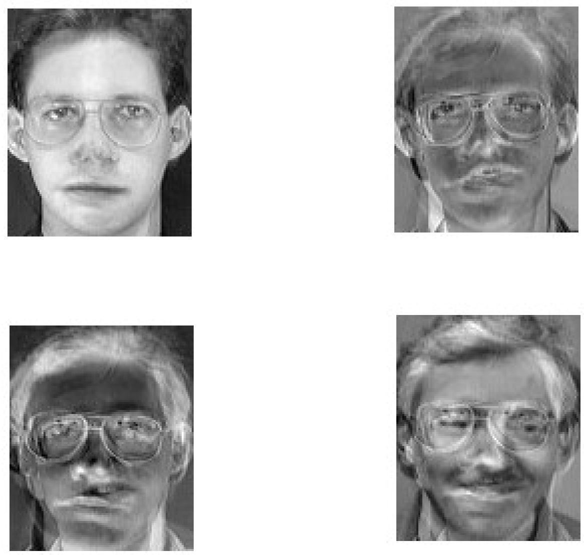


**Figure 5:** The ICA basis vectors for a subset of the ATT image database

## 4. EXPERIMENTAL RESULTS

The performance of the algorithms have been measured in terms of the recognition rate (accuracy) and the total time taken for implementation; this includes both the time for training and the time for testing the images. The algorithms have been tested on the ATT face database [7] and the IFD [8].

- *These evaluations were carried out using the Face Recognition Evaluator, an open source MATLAB interface [11]. The PCA and LDA algorithms were self-coded. The source code for ICA algorithm was obtained from [10 ].*

The results have been organized by testing the algorithms under the following criteria:

4.1. Results obtained on testing the three algorithms using the IFD dataset.
4.2. Results obtained on testing the three algorithms using the ATT database.
4.3. Comparative results obtained by testing the three algorithms on both the IFD and

the ATT database.

4.4. Comparative results obtained from the three algorithms tested on the IFD by varying the percentage of face images from the database used for training and testing phases.

4.5. Comparative results obtained by testing the three algorithms on a Gaussian noise corrupted version of the IFD for three different values of noise standard deviation.

4.6. Comparative results obtained by testing the three algorithms on a Blurred version of the IFD for two different intensities of blurring.

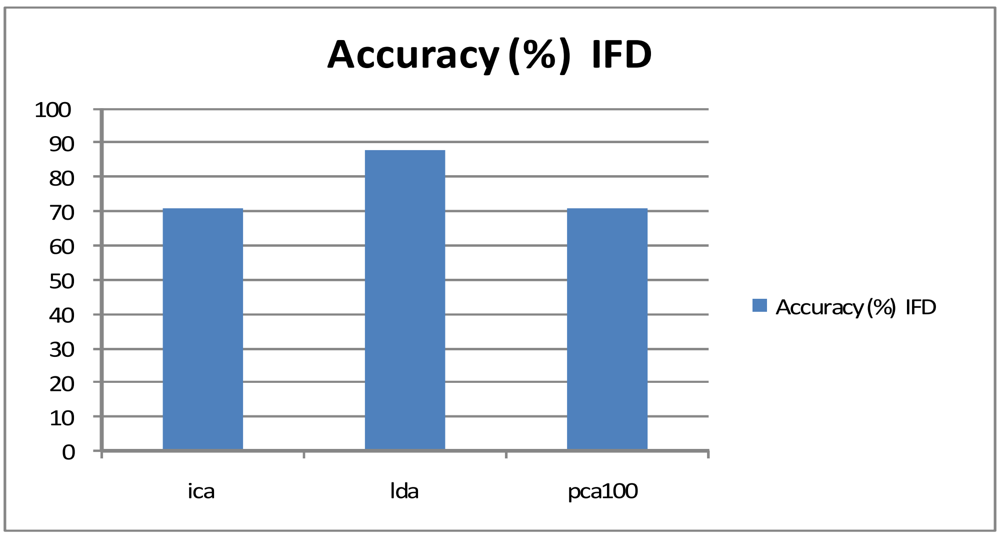
4.7. Results obtained on testing the PCA algorithm on the IFD by changing the number of eigenvectors used for determining the basis vectors.

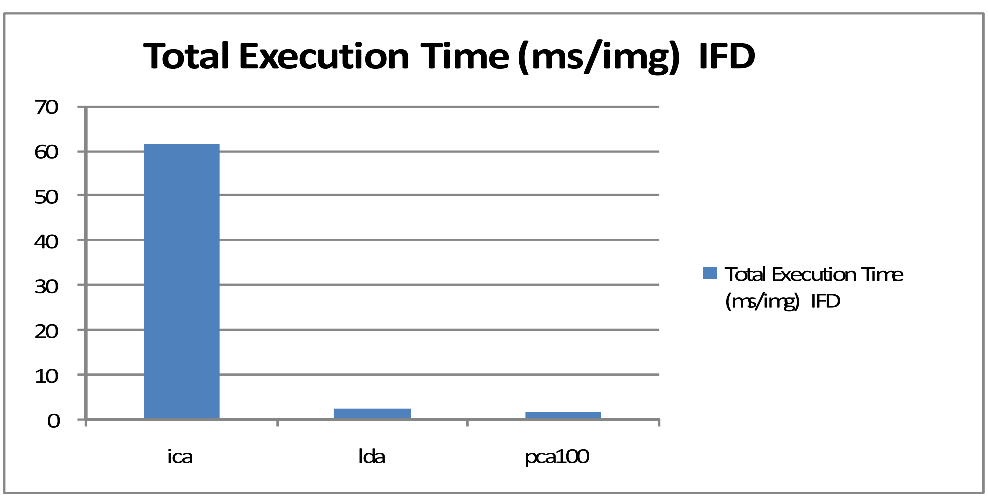
4.8. Results obtained on testing the PCA algorithm on the IFD by varying the number of initial eigenvectors that have been set to zero.

4.9. Results obtained on testing the ICA algorithm on the IFD by varying the size of the reduced dataset obtained by applying PCA.

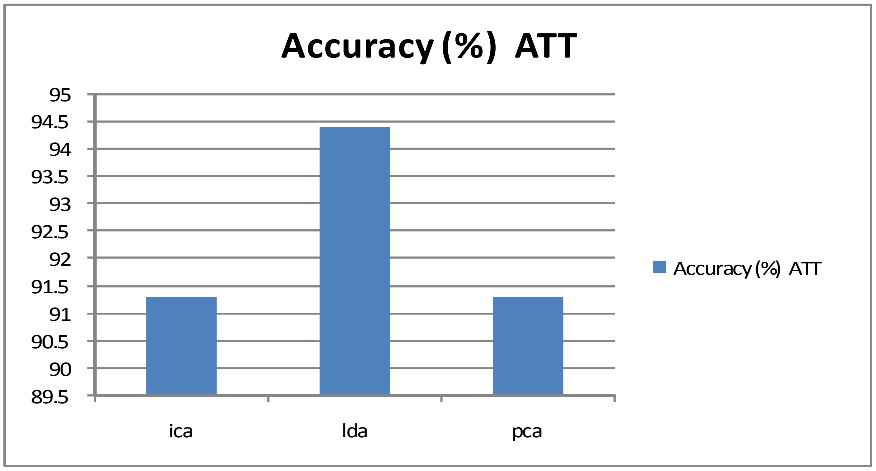
Note: All numerical results are shown in this report in graphical form.

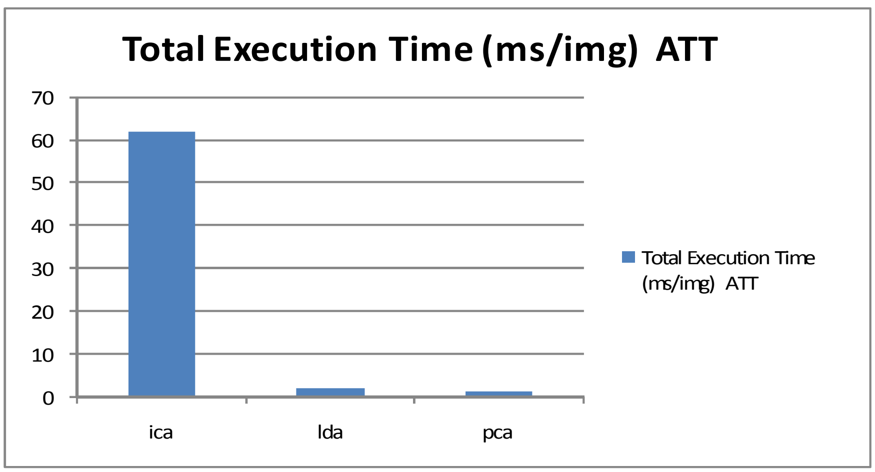
### 4.1 Results obtained by testing on the IFD dataset



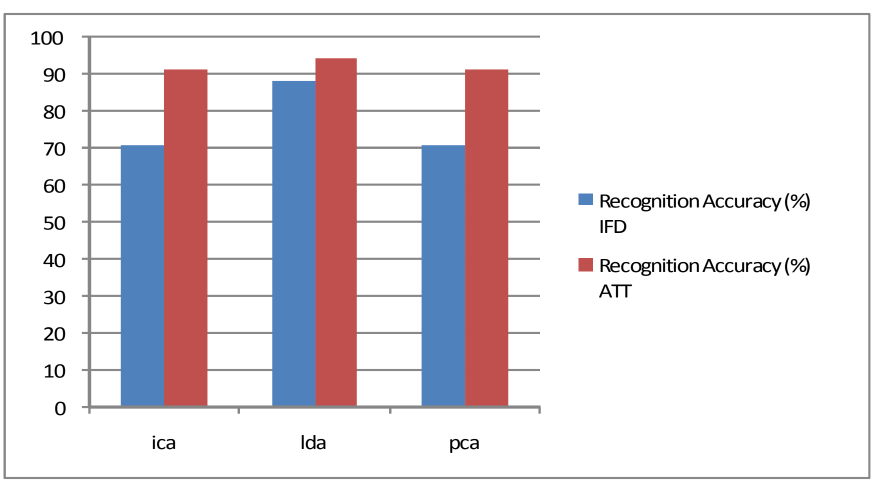



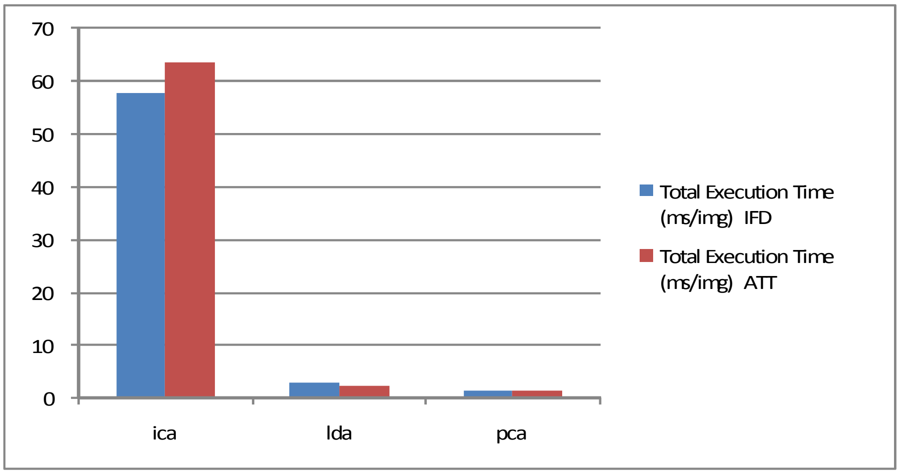
### 4.2 Results obtained by testing on the ATT database



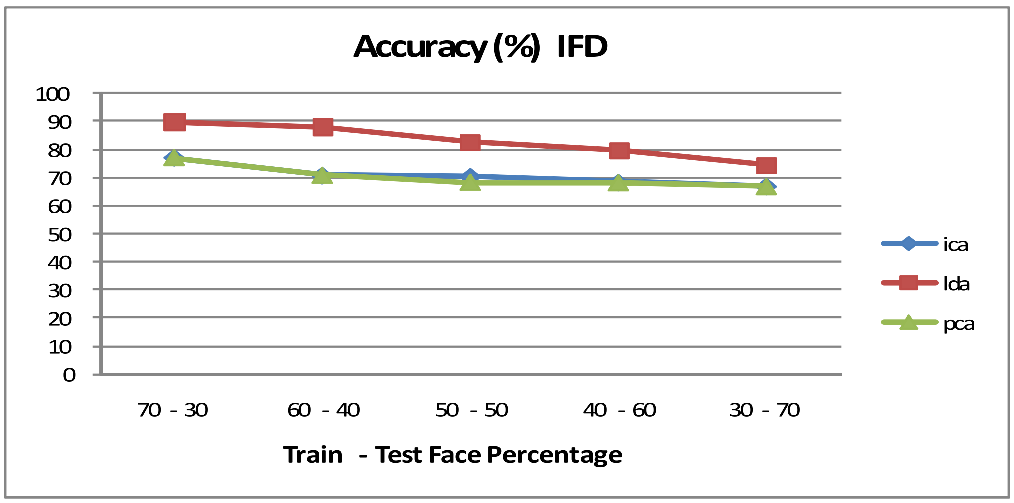



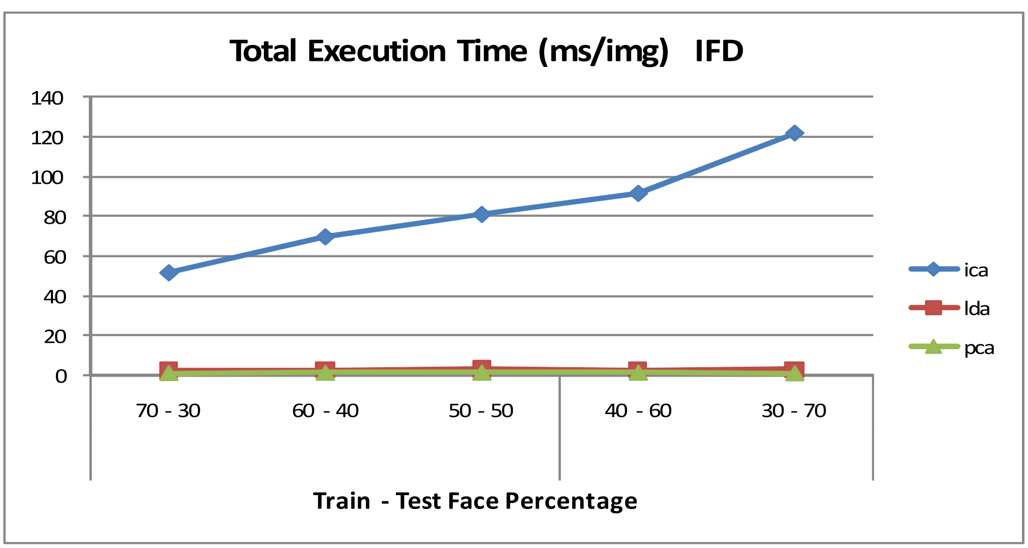

### 4.3. Results obtained by testing the three algorithms on both the IFD and the ATT database.





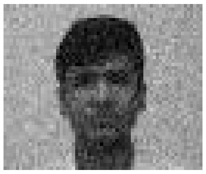

## 4.4. Results by varying the percentage of face images from the IFD database used for training and testing phases.



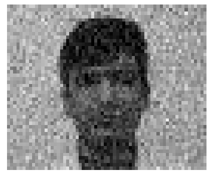



### 4.5. Results obtained from Gaussian noise corrupted IFD dataset.

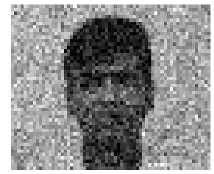
The face images were corrupted with Gaussian noise. Three different noise standard deviations of 0.002, 0.005 and 0.01 were chosen. The effect of adding this noise on the images is shown below:



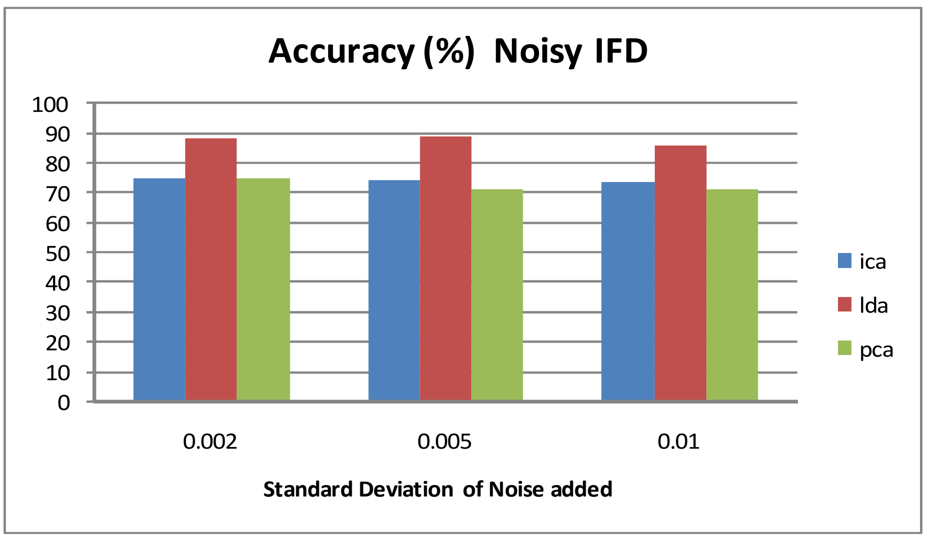
$\sigma = 0.002$

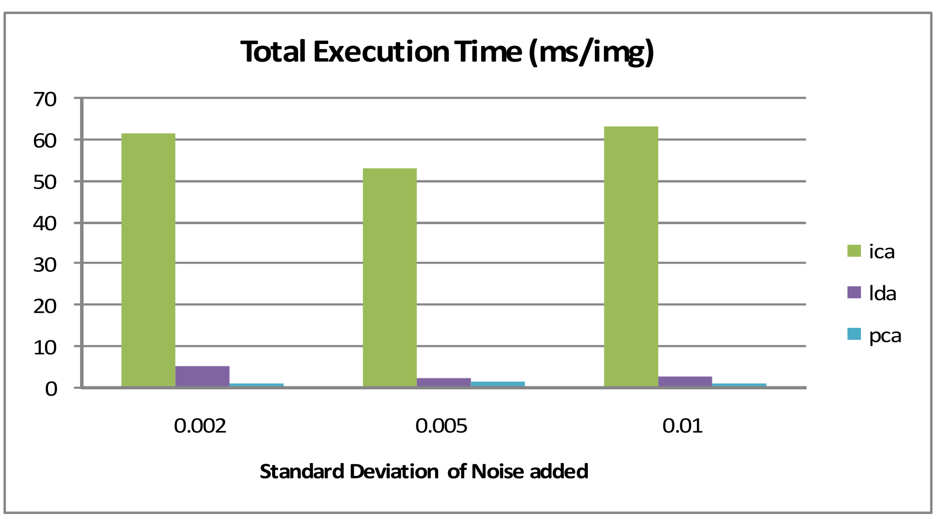


$\sigma = 0.005$



$\sigma = 0.01$



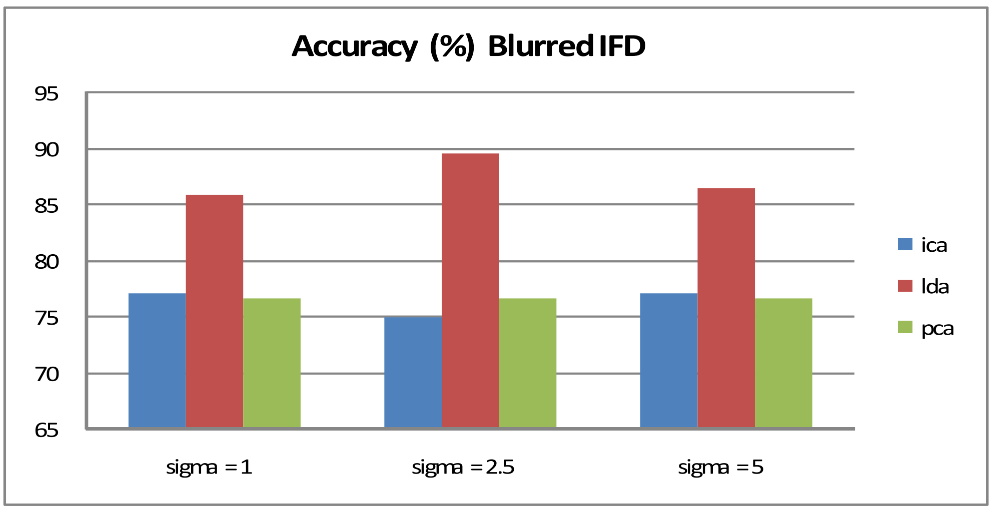



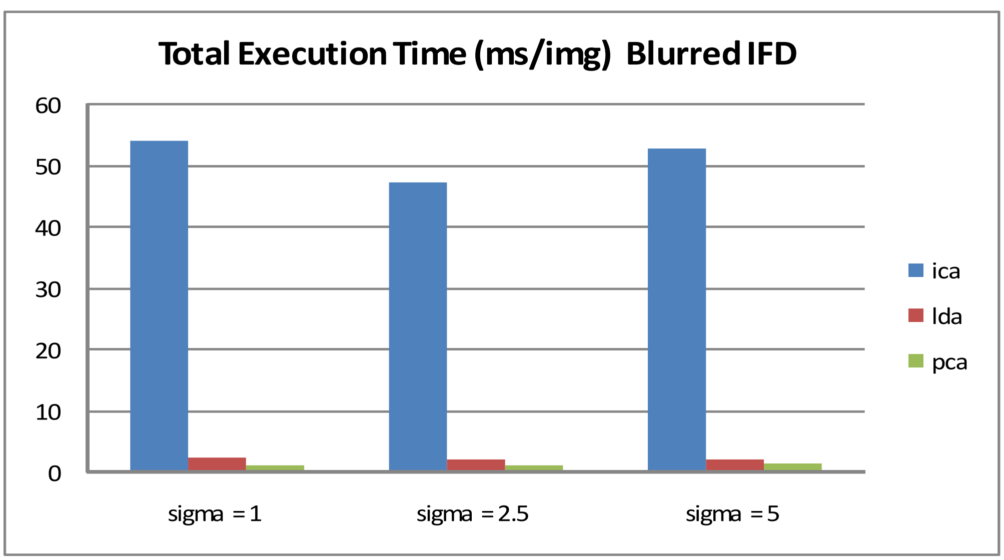
## 4.6. Results obtained from Blurred IFD dataset.

The face images were blurred using a Gaussian PSF with standard deviations of 1, 2.5 and 5. The effect of blurring on the images is shown below:

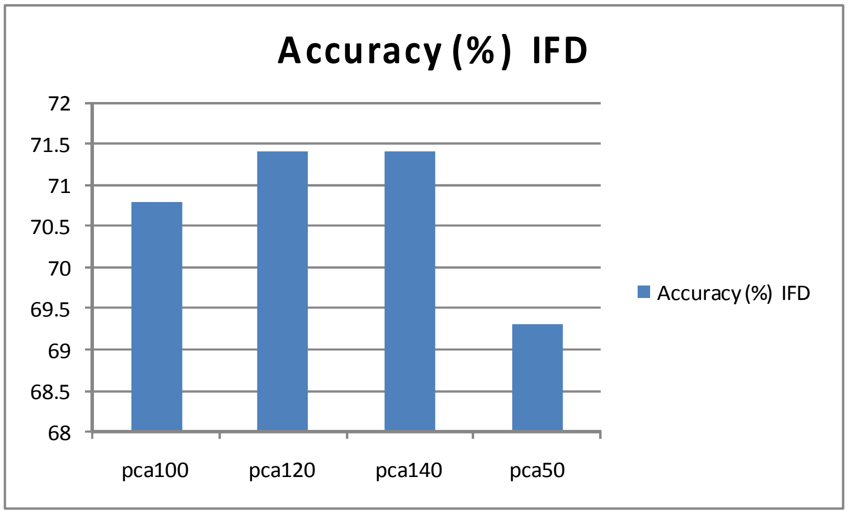


$\sigma = 1$ $\sigma = 2.5$ $\sigma = 5$

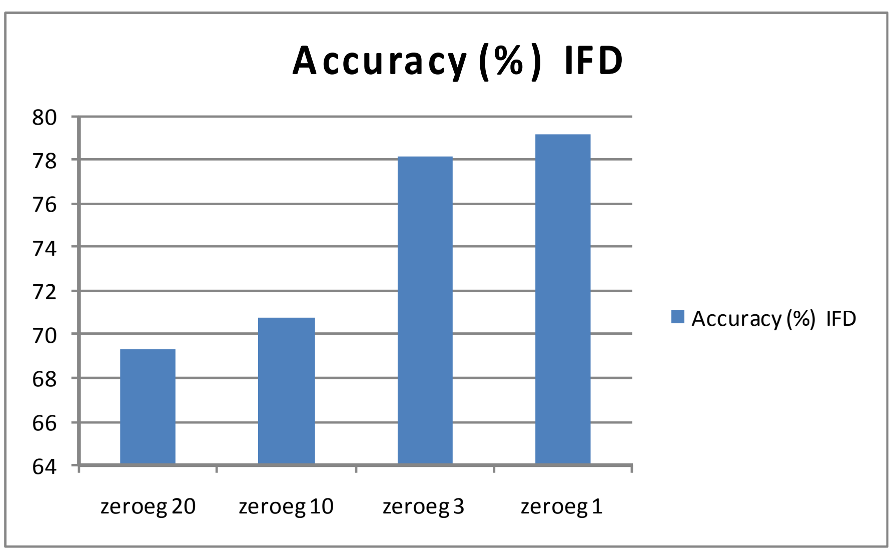


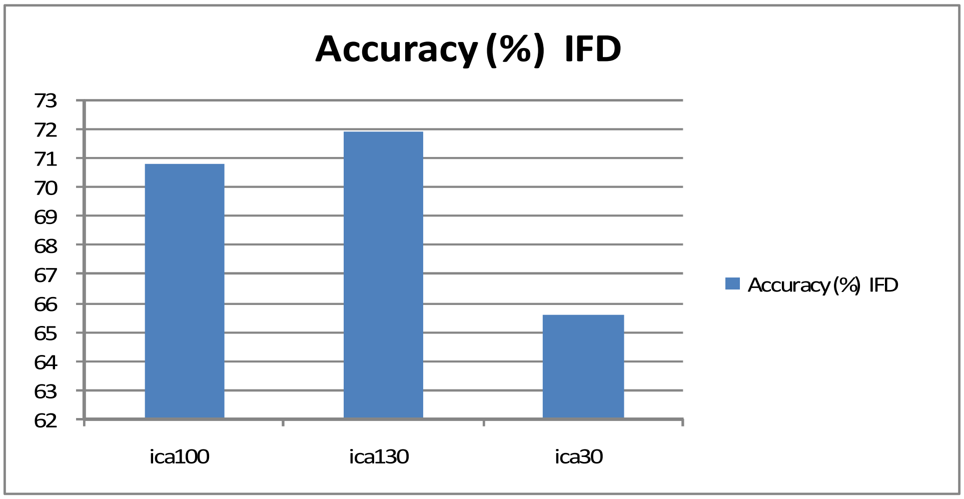
### 4.7. Results obtained from PCA on IFD for basis vectors with varying number of eigenvectors.




### 4.8. Results obtained from PCA on IFD with varying number of zero initial eigenvectors.



### 4.9. Results obtained from ICA for varying size of the reduced dimension data after initial PCA operation.




## 5. PERFORMANCE ANALYSIS

The performance of the algorithms will be analyzed based on the results shown in Section 4. These results justify the properties of these algorithms as explained earlier. The experiments were carried out so as to perform a proper study of these statistical schemes. These will be summarized in this section.

Results 4.1, 4.2 and 4.3 show the performance of the three algorithms for the two databases considered. It is observed that the recognition rate is higher in the ATT database than the IFD. This observation is due to the nature of images encompassed in the IFD. The IFD database has images where each subject is portrayed with highly varying orientation angles. Also, the IFD has images with a larger background region that the ATT images.

In terms of accuracy the LDA shows a higher recognition rate. This is because of the use of discrete classes to group the images and perform a covariance minimization within the same class. The use of this distinct class information increases then feature space used for classification.

The ICA algorithm consumes more computation time than the other two methods. The use of learning based approach and the use of the sphering matrix are steps which are mathematically complex and so involve more computer time.

Result 4.4 shows a classic shortcoming of the recognition function which is seen to be heavily dependent upon the number of images the algorithm is trained upon. Though this project considers only closed loop recognition where the images o be tested are also from the same database, the choice of a higher number of faces for training leads to a higher accuracy. For the case of equal and nearly equal training and testing images, the PCA and ICA algorithms perform equally well.

However, the improved performance of the class based LDA is evident from the results. Another noticeable property is that the execution time varies inversely with the percentage of images the algorithms are trained upon. This is because the recognition process is simplified with a larger trained weight set.

The performance of the algorithms with noisy and blurred images are analyzed in 4.5 and 4.6 which shows that the LDA is resistant to both noise and deformities such as blur. However, with blurred images both PCA and ICA give an almost equivalent performance in terms of the recognition accuracy, while LDA performs much better. In the case of noisy images the three techniques do not vary by a very significant amount in terms of the accuracy, though the LDA scores better.

A very detailed analysis of the PCA based eigenfaces method is obtained using the results of 4.7 and 4.8. They show the dependence of the algorithms performance on the selection of the eigenvectors. The choice of the number of eigenvectors to form the basis vector is seen to be not very much significant with values around 100 and above eigenvectors giving almost equal accuracy. However, if the number chosen is very low, such as the order 50 or less, then the false alarm rate increases. This observation can be attributed to the fact that the PCA projects the face images onto the basis vectors. So for an accurate projection a larger number of eigenvectors are required, in spite of the fact that the original face can be reconstructed from a smaller number of principal components.

The results in 4.8 validate the conclusions of Moon et. al. [15] that not more than 3 initial eigenvectors should be set to zero. As can be seen, for values above 3, the accuracy drops dramatically. [15] states hat since the initial eigenvectors correspond to illumination information they can be considered as noisy and discarded. But as is shown by the results too, these eigenvectors play an important role in determining the feature space for projection.

The final result, 4.9, shows the importance of dimension reduction in ICA and also the extent to which it can affect the algorithm's performance. Use of a very small basis vector for dimension reduction affects the ability of ICA to represent the images efficiently using the independent components. This step of reducing the dimension using PCA is to reduce the computation load of ICA.

## 6. CONCLUSION

In this project three different face recognition algorithms, PCA, LDA and ICA were studied. The PCA and LDA were implemented using MATLAB and the performance was determined in terms of the recognition accuracy and the execution time taken. Experiments were performed under different conditions; by varying the input face image dataset and also by varying the parameters of the individual algorithms.

The study showed that the LDA performs better than the ICA and PCA in terms of the accuracy of recognition. The computational overhead of LDA and the PCA are almost similar while the ICA has a very long execution time. Under the effects of blur in the face image, the LDA outperforms the other two schemes. For noisy images, all three methods presented almost similar results.

These algorithms fail to perform well when the input face images consist of varying orientation of the faces during image acquisition.

## 7. SHORTCOMINGS OF THE STUDY

- In this study only two of the three methods were self-coded.
- The analysis was not complete because images with varying degrees of illumination had not been considered.
- Also, the testing of the algorithms for a mixed set of train/test image set can be performed, wherein the images consist of a mixture of noisy, blurred and differing illumination faces.
- The study could also have considered faces that are partially covered as this provides an excellent scenario for real-time evaluation of these algorithms.

## 8. REFERENCES


[1] H.A. Rowley, S. Baluja and T. Kanade, "Neural Network Based Face Detection", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol 20, 1998

[2] K. Sung and T. Poggio, "Example-based Learning for View-based Human Face Detection," A.I. Memo 1521, MIT A.I. Laboratory, 1994

[3] H. Moon, R. Chellappa and A. Rosenfeld, "Optimal Edge-Based Shape Detection", IEEE International Conference on Image Processing, Canada, Sept 2000

[4] B.S. Manjunath, R.Chellappa and C.v.d. Malsburg, "A Feature Based Approach to Face Recognition", Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp.373-378, 1992

[5] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, Vol. 3, pp 72-86, 1991

[6] K. Etemad and R. Chellappa, "Discriminant Analysis for Recognition of Human Face Images", Journal of the Optical Society of America, Vol 14, pp 1724-1733, 1997

[7] AT&T Laboratories Cambridge, http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.tar.Z

[8] Vidit Jain and Amitabha Mukherjee, The Indian Face Database. http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/, 2002

[9] The CSU Face Identification Evaluation System, Version 5.0, http://www.cs.colostate.edu/evalfacerec/algorithms5.html

[10] http://www.cnl.salk.edu/~marni/icaPapers.html

[11] B.C. Becker and E.G. Ortiz, "Evaluation of Face Recognition Techniques for Application to Facebook," in Proceedings of the 8th IEEE International Automatic Face and Gesture Recognition Conference, 2008.

[12] X. Lu, Image Analysis for Face Recognition, personal notes, May 2003, 36 pages

[13] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," in IEEE CVPR), 1991, pp. 586-591.

[14] Y. Atilgan "Face Recognition" Project report, Cankaya University, Turkey, May 2009

[15] H. Moon and P.J. Phillips, "Computational and Performance Aspects of PCA-Based Face-Recognition Algorithms", Perception, Vol.30, 2001, pp. 303-321

[16] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," in IEEE TPAMI. vol. 19, 1997, pp. 711-720

[17] K. Delac, M. Grgic, S. Grgic, Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set, International Journal of Imaging Systems and Technology, Vol. 15, Issue 5, pp. 252-260

[18] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face Recognition by Independent Component Analysis, IEEE Trans. on Neural Networks, Vol. 13, No. 6, November 2002, pp. 1450-1464

[19] Prof. G. Bebis, Case study slides on PCA, CS791Y - Fall 2003 Course Notes, University of Nevada, Reno, http://www.cse.unr.edu/~bebis/MathMethods/

## A1. APPENDIX – SOURCE CODE

```
% Comparative Study of PCA, ICA and LDA for Face Recognition

% Mahesh Bharath Keerthivasan, University of Arizona
% October - December 2009
% ECE 532 Term Project
%-----------------------------------------------------------------------

% Algorithm References:

% [1]   M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces,"
%       in IEEE CVPR), 1991, pp. 586-591.

% [2]   Prof. G. Bebis, Case study slides on PCA, CS791Y - Fall 2003 Course
%       Notes, University of Nevada, Reno.

% [3]   K. Etemad and R. Chellappa, "Discriminant Analysis for Recognition
%       of Human Face Images", Journal of the Optical Society of America,
%       Vol 14, pp 1724-1733, 1997

% [4]   P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces
%       vs. Fisherfaces:    Recognition Using Class Specific Linear
%       Projection," in IEEE TPAMI. vol. 19, 1997, pp.  711-720
```

```
% -----------------------------------------------------------------------------

% I have written the code for the PCA and LDA based recognition algorithms
% referenced above. The steps in this implementation follow that outlined
% in the project report in Section 3.1 and Section 3.2. The notation used
% is also similar to the one followed in the report.

% The code for ICA based recognition was taken from the internet at
% http://www.cnl.salk.edu/~marni/icaPapers.html

% My implementation's performance was tested using the Face Recognition
% Evaluator, an open source MATLAB toolkit developed by B.C. Becker and
% E.G. Ortiz. The source code of the Evaluator is available at
% http://www.briancbecker.com/bcbcms/site/proj/facerec/fbeval.html

% The Face Recognition Evaluator can handle different face databases and
% algorithms. It has functions to calculate the mean face vector and
% determines the recognition rate and execution time of the algorithms
% as a performance comparison measure.

% -----------------------------------------------------------------------------
% -----------------------------------------------------------------------------

% Script for performing initial preprocessing functionalities.

% Read the Face Database and index the face images

inputfiles = dir([path '/*.' ext]);

inputfaces = [];
    for i = 1:length(inputfiles)
        inputfaces(i).file = inputfiles(i);
    end



% For all face images calculate the Image Vector representation by
% arranging all images as a row vector. Each row in the ImageVector matrix
% consists of the row vector of an image.


Inp = imread([path '/' inputfiles.name]);
[M, N] = size(Inp);
Dim = M * N;
Inp = double(Inp);

for i = 1 : length(inputfiles)
    imagevector(i,:) = reshape(Inp',1,Dim);
end

% Compute the average face. This step is common to both the PCA and LDA
% schemes.

meanface = mean(imagevector,1);


% -----------------------------------------------------------------------------
% -----------------------------------------------------------------------------
```

```
%------- Principal Component Analysis for Face Recognition ----------------

% Step 1: Subtract the average face from the image vector matrix.

for i = 1:size(imagevector, 2)
    imagevector(:,i) = imagevector(:,i) - meanface;
end


% Change variable to that used in the algorithm reference [2].
A = imagevector;
At = A';

% Step 2: Compute the eigenvectors "u" of the scattermatrix A*At. But since
% this step is very memory intnsive due to the large size of the matrices
% involved, we use the alternative method of finding the eigenvectors "v"
% of At*A and then finding the required "u".

[v d] = eig(At * A);

v = v(:,1:120);  % I have taken the first 120 eigenvectors computed.
% This parameter can be varied based on the performance equirements as
% discussed in the result analysis Section 5.


% Step 3: Now the desired eigenvectors "u" can be obtained by multiplying
% the image vector matrix with the "v" vectors.

for i = 1:120
        u(:,i) = A * v(:,i);
end

% Step 4: The transformation matrix weights can be determined now

W = u' * imagevector;


%-------------------------------------------------------------------------
%-------------------------------------------------------------------------

%----- Linear Discriminant Analysis based Face Recognition ----------------

% Step 1: Subtract the average face from the image vector matrix.

for i = 1:size(imagevector, 2)
    imagevector(:,i) = imagevector(:,i) - meanface;
end

% Step 2: Classification of the images. Here I have classified them based
% on the number of subjects involved in the database. The ATT database has
% a total of 40 subjects and the IFD has 50 subjects.

% We define the variable C which is the number of classes. C can take
% values of either 40 or 50.

C = 50; % IFD
% C = 40; % ATT
```

```
% Step 3: As explained in reference [4] we reduce the dimnsion of the
% feature space to N-C using PCA.

% The code for PCA is implemented now:

A = imagevector;
At = A';

[v d] = eig(At * A);

v = v(:,1:120);

for i = 1:120
        u(:,i) = A * v(:,i);
end

% Step 4: Determine the two scatter matrices involved; SB and SW

% To determine the scatter matrices we need the class average. This is
% found out next :

for i = 1 : C
    j = 1 : 9 % J is the number of images in each class. This is 9 in the
    % two databases under consideration

    classaverage{i} = mean(imagevector(:,j), 2);
end

% Next I find the scattermatrix SW. The summation is performed by first
% initializing the scattermatrix to zero and then adding it cumulatively.

SW = zeros(Dim);
for i = 1:C
    for j = 1:9
        temp1 = imagevector(:,j) - classaverage{i};
        SW = SW + (temp1 * temp1');
    end
end

% Next the Scattermatrix SB is calculated. Again cumulative addition is
% used.

SB = zeros(Dim);

for i = 1:C
    temp2 = classaverage{i} - meanface;
     SB = SB + (temp2 * temp2');
end

% Step 5: Now we project the scatter matrix into the basis vectors obtained
% using the PCA reduction. This gives the non-singleton scattring matrices
% SWN and SBN.

SBN = u.' * SB * u;
SWN = u.' * SW * u;

% Step 6: The next step is finding the generalized eigenvectors of the SWN
% and SBN so that it satisfies the equation SB*WLDA = SW*WLDA*D.
```

```
[V,D] = eig(SBN,SWN);

% Step 7: Only the C-1 eigenvectors corresponding to the C-1 largest
% eigenvalues are retained.

% The code snippet for sorting the eigenvalues and extracting the
% corresponding eigenvectors is taken from: "Digital Image Processing Using
% MATLAB" by Gonzalez, Eddins and Woods.

Diag = diag(D);
[sorted,I] = sort(abs(Diag));
I = flipud(I);

v = V(:,I); % This contains the eigenvectors from the largest

% Only the C-1 eigenvectors are retained.

Cn = C-1;
WLDA = v(:,1:Cn);

% Step 8: Now the final step of calculating the transformation matrix
% weights by projecting the image vector matrix onto the basis vectors WLDA
% is done.

w = WLDA' * imagevector;

%--------------------------------------------------------------------------
%--------------------------------------------------------------------------
```