NEAR FIELD HUMAN FACE IDENTIFICATION USING PRINCIPAL COMPONENT ANALYSIS (PCA), LINEAR DISCRIMINANT ANALYSIS (LDA), AND FUSION OF PCA AND LDA

Dr K. T. Deepak[1], Dr. Sibasankar Padhy[2], Kuppa Venkata Krishna Paanchajanya[3], Karthik Sajjan[4], Karusala Deepak Chowdary[5], Vanga Manoj Sahith Reddy[6]

[#1] HOD, Department of ECE, Indian Institute Of Information Technology, Dharwad, Karnataka, India

[#2] Asst. Professor, Department of ECE, Indian Institute Of Information Technology, Dharwad, Karnataka, India

[#3] Student, B. Tech III Year, CSE, Indian Institute Of Information Technology, Dharwad, Karnataka, India

[#4] Student, B. Tech III Year, CSE, Indian Institute Of Information Technology, Dharwad, Karnataka, India

[#5] Student, B. Tech III Year, CSE, Indian Institute Of Information Technology, Dharwad, Karnataka, India

[#6] Student, B. Tech III Year, CSE, Indian Institute Of Information Technology, Dharwad, Karnataka, India

Abstract- We implemented face detection and recognition systems using linear dimensionality reduction algorithms such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) wherein we computed feature spaces namely eigen space and fisher space respectively which ultimately indicate intrinsic characteristics of every subject of a training set. After computing the feature space, we computed the weight matrix of the training set which is nothing but the projection coefficients found while projecting each and every image of the dataset on to the feature space. This marks the end of the training stage. While testing an image whether it is identified as a person whose faces have been trained in the training stage, we first perform some pre-processing and then project the face vector on to the feature space of the algorithm chosen (PCA or LDA) thereby computing the projection coefficients. The projection coefficients found here shall be compared with the weight matrix to obtain the least-distant feature space column constituent. This distance, if less than the specified threshold limits, shall indicate a face amongst the trained ones has been matched. In addition to the classic PCA and LDA algorithms, we also implemented the fusion of PCA and LDA wherein we combined the distance vectors obtained from both the approaches to achieve better results. The accuracies obtained from each of the three approaches applied on 12 labelled datasets (2 of which were created by our team) are tabulated in the Results and Discussion section. Based on the results, it is found that PCA has shown greater accuracy in majority cases followed by PCA+LDA and LDA.

Keywords: Principal Component Analysis; Linear Discriminant Analysis; Fusion of PCA and LDA, Threshold limits, Euclidian Distance, feature space, eigen space, fisher space

1. INTRODUCTION:

In a navigating humanoid, computer vision plays a pivotal role in perceiving information about the surrounding environment. For object detection, the robot is expected to scan an image and detect a target object which has certain features, such as a human face. Humanoid employs a face recognition system that matches a human face from the live video feed through the vision systems against a database of faces, typically used to authenticate users or to recognize them for various human-computer interactive functions to be performed. There are two important stages to be passed through in order to perform face recognition viz., face detection and face recognition. While face detection helps us focus on the key parts of an image containing faces, face recognition recognizes individuals based on their unique facial characteristics.

An image is considered a high-dimensional vector as it takes numerous pixels to present a natural visual experience. Statistical methods are the ones frequently used in face recognition systems as many of them constitute linear dimensionality reduction algorithms which aid us in operating with large image sets with great resolutions without the problem of overfitting[1]. We obtain a feature space from the training set after performing the reduction which basically indicates the unique characteristics of each of the human faces trained. The face vector of a sample image when projected to the basis vectors of the feature space gives projection coefficients which can later be analysed using a nearest neighbour algorithm or Euclidian distance to identify the human face.

We shall discuss the implementation of a face detection and recognition system for a humanoid using linear dimensionality reduction algorithms such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) wherein we perform data reduction and feature extraction. We shall also see the working of a fusion of the PCA and LDA algorithms. Lastly, we will perform a compare and contrast study on the accuracies in human face identification using all the three algorithms.

2. METHODOLOGY

2.1 Face Databases

Numerous face databases are available for face recognition researchers. These databases differ in size, scope, and purpose. Most of the images in these databases are acquired particularly to study face recognition. Table 2.1 shows the important features of different face recognition databases.

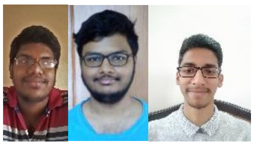
S.No	Name	Color	Individuals	File Format	Cropped to face (Y/N)
1	2D Face Dataset	RGB	23	JPG	N
2	Faces94	RGB	152	JPG	N
3	FriendsDB	RGB	3	JPG	N
4	FriendsDB_FacialHair	RGB	3	JPG	N
5	MUCT	RGB	199	JPG	N
6	NIT Rourkela – 1	RGB	10	JPG	Y
7	NIT Rourkela – 2	RGB	47	JPG	N
8	Olivetti – ATT – ORL	Gray	40	PGM	N
9	Pain	RGB	21	JPG	N
10	Pain_Cropped	Gray	12	JPG	Y
11	Yale Face Database	Gray	15	GIF	N
12	Yale Face Database B	Gray	28	PGM	N

Table 2.1

2D Face Dataset

Faces94

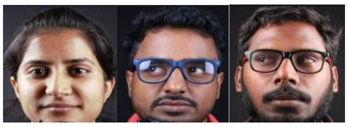
FriendsDB



FriendsDB_FacialHair

MUCT

NIT Rourkela – 1



NIT Rourkela – 2

Olivetti – ATT – ORL

Pain

Pain_Cropped

Yale Face Database

Yale Face Database B

2.2 Principal Component Analysis (PCA)

Principal Component Analysis is a standard statistical pattern recognition algorithm used for data reduction and feature extraction [2]. Here are the steps for training the face dataset using PCA i.e., generating a feature space that basically indicates unique characteristics of each of the faces trained.

- **Step 1:** Consider that the training database consists of M images. Convert every image to grayscale
- **Step 2:** Resize each of those images into 100 x 90 pixel matrices
- **Step 3:** Flatten the 100x90 pixel matrices of each of the images in the training dataset. Flatten in the sense, the pixel matrix of an image unrolls all its values into a vector. Arrange them column-wise to prepare another matrix known as the image matrix i.e., $\Gamma = [\Gamma_1, \Gamma_2, \ldots \Gamma_M]$
- **Step 4:** Compute the mean face vector as $\Psi = \frac{\sum_{i=1}^{M} \Gamma_i}{M}$ i.e., the average of individual face vectors.
- **Step 5:** The face vector space freed from common features among all the images in the training set is computed by $A = [\Phi_1, \ldots \Phi_M]$ where $\Phi_i = \Gamma_i - \Psi$. Here Φ_i is also referred to as a deviation vector
- **Step 6:** Next, we need to compute the covariance matrix given by $C = A.A^T$. This leads to a matrix with huge dimensions 9000 x 9000 wherein computing eigen vectors otherwise known as eigenfaces becomes difficult. Thus, we consider a lower dimensionality subspace here i.e., we calculate $C = A^T.A$ instead resulting in a M x M dimension matrix which is smaller and easier to compute compared to the prior covariance matrix. [3]
- **Step 7:** The set of eigen vectors for C is given as $V= [V_1, V_2, V_3, \ldots V_M]$. Now, to obtain the eigen vectors for the original higher order 9000 x 9000 matrix back, we need to multiply respective eigen vectors with the difference matrix A. Thus, the eigenfaces are $U = [U_1, U_2, \ldots U_M]$ where $U_i = AV_i$
- **Step 8:** Instead of using M eigen vectors obtained in the preceding step, the certain value $m' \leq M$ is chosen to create the eigen space U. The eigen space shall consist of m' dominant eigen vectors amongst the M eigen vectors as its columns. Dominant eigen vectors indicate the ones associated with high eigen value.
- **Step 9:** Then the weight of each eigenvector ω_i to represent the image in the eigenface space, is given by: $\omega_i = U_i^T(\Gamma - \Psi)$, i = 1, 2, 3, …m'. Weight matrix $\Omega = [\omega_1, \omega_2, \ldots \omega_{m'}]^T$
- **Step 10:** The threshold value is the value of Euclidian distance between the weight matrix column vectors and the projection of a test image on the eigen space U within which a test image can be termed as recognised or matched with a trained image. Generally, threshold value is chosen arbitrarily or taken as some factor of maximum value of minimum Euclidian distances of each image from other images. Therefore, we decided to compute the PCA threshold in two ways:
  - **Adaptive Threshold:** In order to obtain best results, we dynamically keep floating the factor between 0.75 and 0.90 and check at which value shall we be getting greater accuracy in recognition
  - **Non-Adaptive Threshold:** It is observed that best results are obtained usually with the factor being 0.8. Thus, we keep the factor static at 0.8 [4]

Here are the steps for testing a sample face image whether it is recognised by PCA or not:

- **Step 1:** Convert the test image to grayscale
- **Step 2:** Flatten it into a face vector
- **Step 3:** Normalize the face vector
- **Step 4:** Project the normalized face vector onto the eigen space
- **Step 5:** Compute the weight vector of the input image
- **Step 6:** Euclidian distance is one of the methods that can be used to match a new face image to the existing face image in the database. Smaller the Euclidian distance, more is the image similar to the one available in the database.
- **Step 7:** Calculate the Euclidian distance between this weight vector and the weight vectors of each of the faces already trained.
- **Step 8:** The person associated with the weight vector with which the Euclidian distance is minimum and within the threshold is identified as the person whose face is provided.

2.2 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis originates from the concept of classes while PCA does not. Class in face recognition systems mean a specific person, and elements of class are his/her face images. Here are the steps for training the face dataset using LDA i.e., generating a feature space that basically indicates unique characteristics of each of the faces trained [5].

- **Step 1:** Consider the training set consists of M images over p persons. Convert all images to grayscale
- **Step 2:** Resize each of those images into 100 x 90 pixel matrices
- **Step 3:** Flatten the 100x90 pixel matrices of each of the images in the training dataset. Flatten in the sense, the pixel matrix of an image unrolls all its values into a vector. Arrange them column-wise to prepare another matrix known as the image matrix i.e., $\Gamma = [\Gamma_1, \Gamma_2, \dots \Gamma_M]$
- **Step 4:** Compute the average of all faces as $r_m = \frac{\sum_{i=1}^{M} \Gamma_i}{M}$ i.e., the average of all individual face vectors.
- **Step 5:** Compute the average face of each person viz., r_1, r_2, …r_p
- **Step 6:** The face vector space freed from common features among all the images of the respective person in the training set is computed by $A = [\Phi_1, \dots \Phi_M]$ where $\Phi_i = \Gamma_i - r_j$. Here Φ_i is also referred to as a deviation vector and r_j ($j \in \{1 \dots p\}$) is the average face of the person associated with Γ_i
- **Step 7:** Build scatter matrices S_1, S_2, S_3, …S_p such that S_i = sum of product of the deviation vectors and its transposes associated with the i^{th} person.
- **Step 8:** Compute the within-class scatter matrix S_W which is the sum of all such scatter matrices found in the preceding step i.e., $S_W = S_1 + S_2 + \dots + S_p$
- **Step 9:** The between-class scatter matrix S_B is computed as

$$S_B = \sum_{i=1}^{p} (no.of\ images\ of\ i^{th}\ person\ for\ training) * (r_i - r_m)(r_i - r_m)^T$$

- **Step 10:** Now, computer the eigen vectors of the matrix $S_W^{-1} S_B$. The set of the eigen vectors computed when arranged column-wise in a matrix is the feature space formed by LDA sometimes also referred to as fisher space.
- **Step 11:** Instead of using all the 9000 eigen vectors obtained in the preceding step, the certain value $m' \leq 9000$ is chosen to create the feature space U. The space shall consist of m' dominant eigen vectors amongst the 9000 eigen vectors as its columns. Dominant eigen vectors indicate the ones associated with high eigen value.
- **Step 12:** Then the weight of each eigenvector ω_i to represent the image in the fisher space, is given by: $\omega_i = U_i^T(\Gamma - \Psi)$, i = 1, 2, 3, …m'. Weight matrix $\Omega = [\omega_1, \omega_2, \dots \omega_{m'}]^T$
- **Step 13:** Generally, threshold value is chosen arbitrarily or taken as some factor of maximum value of minimum Euclidian distances of each image from other images. Therefore, we decided to compute the LDA threshold in two ways:
  - **Adaptive Threshold:** In order to obtain best results, we dynamically keep floating the factor between 0.75 and 0.90 and check at which value shall we be getting greater accuracy in recognition
  - **Non-Adaptive Threshold:** It is observed that best results are obtained usually with the factor being 0.8. Thus, we keep the factor static at 0.8 [4]

Here are the steps for testing a sample face image whether it is recognised by LDA or not:

- **Step 1:** Convert the test image to grayscale
- **Step 2:** Flatten it into a face vector
- **Step 3:** Normalize the face vector using the average of all faces
- **Step 4:** Project the normalized face vector onto the fisher space
- **Step 5:** Compute the weight vector of the input image
- **Step 6:** Calculate the Euclidian distance between this weight vector and the weight vectors of each of the faces already trained. The person associated with the weight vector with which the Euclidian distance is minimum and within the threshold is identified as the person whose face is provided.

2.3 Fusion of PCA and LDA (PCA+LDA or PCA-LDA) [6]

The Fusion algorithm of PCA and LDA basically uses both the PCA and LDA feature spaces to obtain better results. Thus, the algorithm needs training using both PCA and LDA approaches which has already been discussed prior. In the testing stage comes the new approach where we make use of both the eigen and fisher spaces obtained while training. Here are the steps:

- **Step 1:** Convert the test image to grayscale
- **Step 2:** Flatten it into a face vector and make a copy of it for further use
- **Step 3:** Normalize the face vector using the PCA average of all faces Ψ
- **Step 4:** Project the normalized face vector onto the eigen space
- **Step 5:** Compute the weight vector of the input image.
- **Step 6:** Calculate the Euclidian distance between this weight vector and the weight vectors of each of the faces already trained. This shall form the PCA distance vector
- **Step 7:** Normalize the face vector whose copy we had made, using the LDA average of faces r_m
- **Step 8:** Project the normalized face vector onto the fisher space
- **Step 9:** Compute the weight vector of the input image
- **Step 10:** Calculate the Euclidian distance between this weight vector and the weight vectors of each of the faces already trained. This shall form the LDA distance vector
- **Step 11:** We normalise the PCA distance vector and LDA distance vector to reduce the distance range to the interval [0,1].
- **Step 12:** Compute a combined distance vector which contains both PCA and LDA information. To do so, we obtained the combined distance vector using the mean vector

$$d = \left\{ \frac{d_1^{PCA} + d_1^{LDA}}{2}, \dots, \frac{d_N^{PCA} + d_N^{LDA}}{2} \right\}$$

- **Step 13:** The person associated with the weight vector with which the Euclidian distance is minimum and within the threshold is identified as the person whose face is provided.

3. RESULTS AND DISCUSSION

We have implemented Principal Component Analysis and Linear Discriminant Analysis making use of both Adaptive and Non-Adaptive threshold limits in MATLAB. We also implemented PCA+LDA which makes use of the feature spaces generated by both PCA and LDA. Table 3.1 indicates number of images trained and tested in each of the 12 datasets chosen for study. It also illustrates the accuracies obtained in recognising the test images using dimensionality reduction algorithms i.e., PCA, LDA, and PCA+LDA.

Dataset (train, test)	PCA Accuracy (%)		LDA Accuracy (%)		PCA+LDA Accuracy (%)	
	Adaptive Threshold	Non-Adaptive Threshold	Adaptive Threshold	Non-Adaptive Threshold	Adaptive Threshold	Non-Adaptive Threshold
2D Face Dataset (184, 230)	79.130435	79.130435	84.7348485			
Faces94 (1520, 1520)	98.421053	97.565789	85.2243592			
FriendsDB (72, 72)	100.000000	98.61111	84.722222	75.000000	98.611111	
FriendsDB_FacialHair (540, 720)	99.222222	98.888889	99.027778		99.444444	
MUCT (796, 995)	70.351759	61.306533				
NIT Rourkela – 1 (127, 144)	96.527778	93.055556	56.944444			
NIT Rourkela – 2 (27700, 27977)	85.756157	82.085284				
ATT ORL (190, 181)	67.955801	67.955801				
Pain (294, 252)	92.063492	92.063492				
Pain_Cropped (48, 36)	83.333333	83.333333	66.666667			
Yale Face Database (75, 89)	77.528090	77.528090	68.544601			
Yale Face Database B (8204, 8176)	43.664384	38.992172	31.3451293			

Table 3.1

CONCLUSION AND FUTURE SCOPE

Highest recognition rate in most of the datasets is achieved using Principal Component Analysis (PCA) while the fusion of both PCA and LDA statistical approaches also gives best results in case of some datasets. Overall, when compared to other approaches, LDA has achieved lesser accuracies. Adaptive threshold has turned out to be a good approach for deciding the threshold limit thereby giving more accuracy than the one with static 0.8 factor. We believe using some advanced nearest neighbour algorithm for computing the minimum-distant faces shall give us better results in accuracy. We could also study how histogram equalization in the pre-processing stage and the dominant eigen vector selection shall affect the recognition rate.

REFERENCES

1. Face Detection and Recognition Using Viola-Jones with PCA-LDA and Square Euclidean Distance Nawaf Hazim Barnouti Al-Mansour University College Baghdad, Iraq Sinan Sameer Mahmood Al-Dabbagh Al-Mansour University College Baghdad, Iraq Wael Esam Matti Al-Mansour University College Baghdad, Iraq Mustafa Abdul Sahib Naser Al-Mansour University College Baghdad, Iraq
2. M. Turk and A. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991
3. Christian Tarunajaya1 , Oey Kevin Wijaya1 , Reinard Lazuardi Kuwandy1 , Heri Ngarianto1 , Alexander A S Gunawan1 , and Widodo Budiharto1, IPTEK, The Journal for Technology and Science, Vol. 26, No. 2, August 2015 Development of Intelligent Humanoid Robot with Face Recognition Features
4. A New Optimized Approach to Face Recognition Using EigenFaces Sheifali Gupta [1], O.P.Sahoo [2], Ajay Goel[3], Rupesh Gupta[4] [1] Department of Electronics & Communication Engineering, Singhania University, Rajasthan,India [2] Department of Electronics & Communication Engineering, N.I.T. Kurukshetra, India [3] Department of Computer Science & Engineering, Singhania University, Rajasthan, India [4] Department of Mechanical Engineering, Singhania University, Rajasthan, India sheifali@yahoo.com , opsahu_reck@yahoo.co.in , goelajay1@gmail.com, rup_esh100@yahoo.co.in
5. Face Recognition by LDA by Nawin Kumar Sharma
6. Fusion of LDA and PCA for Face Recognition Gian Luca Marcialis and Fabio Roli Department of Electrical and Electronic Engineering - University of Cagliari Piazza d'Armi - 09123 Cagliari (Italy) {marcialis,roli}@diee.unica.it