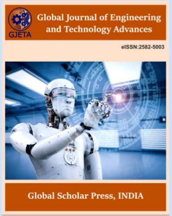
Global Journal of Engineering and Technology Advances

eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/


Global Journal of Engineering and Technology Advances
GJETA
eISSN:2582-5003
Global Scholar Press, INDIA

(Research Article)

Revisiting the performance of PCA versus FDA versus Simple Projection for image recognition

Fahad Bin Mostafa [1,*], Md Sakhawat Hossain [1] and Md Easin Hasan [2]

[1] *Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX-79409, USA.*
[2] *Department of Mathematical Sciences, The University of Texas at El Paso, El Paso, TX-79968, USA.*

Global Journal of Engineering and Technology Advances, 2021, 08(01), 084–095

Publication history: Received on 02 June 2021; revised on 11 July 2021; accepted on 13 July 2021

Article DOI: https://doi.org/10.30574/gjeta.2021.8.1.0099

Abstract

Computer vision becomes a great area of research due to huge availability of images and videos. For enhancement of security, biomedical imaging or automation of identification, one may need useful tools to recognize images. One main problem of image data set is high dimensional, and it is very expensive to work with huge dimensions. In this paper, our main aim is to show a better dimension reduction process of high dimensional image data sets from several existing techniques. To verify it we start with most useful singular value decomposition to reduce the dimensionality of data to incorporate principal components. On the other hand, we classify data in advance to work out Fisher's discriminant analysis. From many real-world examples, we set a very well-known paradigm of analysis using Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) or Fisher Discriminant Analysis (FDA) and Simple Projection (SP) to recognize people from their facial images. We consider that we have some images of known people that can be used to compare and recognize new images (of the same set of face images). Moreover, we show graphical and tabular representation for average performance of correct recognition as well as analyze the effectiveness of three different machine learning techniques.

Keywords: SVD; Orthogonal linear transformation; Orthogonal projection; PCA; LDA; Eigenfaces

1. Introduction

By the advancement of technology, people are using internet for interchanging millions of photos every day from one to another part of the world, where data reductions are used to send file long distance within minimum period of time. In the medical science, physicians are detecting body organs, tumor cells and complex physical phenomena by optical fibers and where image processing is quite useful. The other important sectors of image processing is meteorology where it processes images sent by satellite to do daily weather forecasting or finding climate change. There are many other sectors such as military surveillances, underwater search, satellite navigation etc. Suppose there are millions of images in the database of NSA, but they do not have clear image of the suspect (or suspects). Their main aim is to find the image (or multiple images) of particular scene and identify object of interest in the image (or images). In the paper of digital image processing [1] showed example on medical radiology. It has been quite a while that researchers are finding better method to show images more clearly in different sectors. Here we introduced three existing methods and we gave a consultancy to the researcher that which method ought to apply. One of the mostly used methods is fisher-faces which derives from fisher's discriminant analysis [2] (R.A. Fisher 1936) or simply we say LDA [3]. Apart from FDA/LDA we use dimensionality reduction technique which produces projection directions that maximize the total scatter across all classes. Image recognition has very important applications in numerous ways, so the identification and authentication of an image is very crucial. For example, face recognition techniques applied by the facial recognition machinery which can identify the facial expressions using many machine learning algorithms like PCA, LDA and Kernel

* Corresponding author: Fahad Bin Mostafa
Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX-79409, USA.

LDA [4]. Those algorithms use grayscale or RGB images without in-depth details of object for classical image recognition techniques even though PCA based on the multi view depth information can improve the image classification accuracy [5]. Hence, for obtaining the in-depth details of an image people can use unsupervised image depth estimation. We used dimensionality reduction techniques to obtain the multi-view depth characters of images through several unsupervised machine learning techniques like PCA, FDA etc. since understanding the structure of high dimensional data is difficult. PCA is very popular among researchers to reduce the dimension of a dataset and uncovering the underlying structure of that dataset. Using PCA we can derive the complex representation of image data by employing the algebra of matrices in the context of image analysis [6]. Data visualization is also an important factor for image data analysis. We can reduce the high dimensional data to low dimensional data for data visualization since the reduced form of the data shows the similar intuition as original data because we are transforming the correlated variables to uncorrelated variables by linear combinations of uncorrelated to the original variables by applying PCA [7]. We need to maintain the main features of images (distribution of color, edge shapes, texture features, and region shapes) by replacing the original features of images in case of dimension reduction and PCA can reduce the dimension by eliminating the components with low variance and keeping the components with high variance [8]. LDA is another machine learning technique for image recognition and very useful method for features extraction and dimension reduction with little computation time and the improvement of LDA method can be done by discrimination vector selection, statistical uncorrelation of discriminator vector and by improving the PCA [9]. Moreover, in the projected feature space LDA find out a set of projection vectors by supplementing the between class scatter matrix for extracting more discriminant information [10]. The most popular LDA method for image recognition is Fisherface/Eigenface which is based on Fisher's linear discriminant analysis and can generate in depth separated classes in a low-dimensional subspace even under the extreme variation in facial expressions and lightning [11]. Another technique is Eigenface which projected the image space to a low dimensional subspace but Fisherface was better than Eigenspace when tested Harvard and Yale face databases [11]. That is why we considered the fisher discriminant analysis technique in our study.

In this current paper, we discuss the mathematical background of PCA, LDA and SP and incorporate with statistical analysis. We show few proofs to explain the projection and how it helps to reduce dimensionality. Moreover, we explain how to apply MATLAB for all mentioned techniques. Finally, a comparison is shown between three techniques and described the performances for face recognition.

2. Methodology

As images are very high dimensional, it is not easy to analyze them directly. A few common approaches are to reduce their diminution using principal component analysis (PCA), Fisher's discriminant analysis (FDA), and other similar methods. Both PCA and LDA are linear transformation methods. PCA yields the directions (principal components) that maximize the variance of the data, whereas LDA also aims to find the directions that maximize the separation (or discrimination) between different classes, which is applied in pattern classification problem including image recognition.

Our main problem is arising from $Y = X\beta + \varepsilon$ where $Y\ is\ n \times 1$ column matrix, $X\ is\ n \times p$ matrix and ε is $n \times 1$ matrix. If we ignore the error term ε and compare it with system of linear equation of the standard form $AX = b$ then for $n > p$, the system is overdetermined and we consider it as linear least square problems (to solve it, we use Gram Schmidt process, QR factorization, Householder, etc), but when $n < p$ then the system becomes underdetermined and we use singular value decomposition shortly SVD to reduce the dimension of columns of matrix X from p to a smaller dimension (say,'d' for our case). PCA is one of the central uses of SVD.

Here, PCA analysis is to identify patterns in data; PCA aims to detect the correlation between variables. If a strong correlation between variables exists, the attempt to reduce the dimensionality only makes sense. In a sense PCA is all about finding the directions of maximum variance in high-dimensional data and projecting it onto a smaller dimensional subspace while retaining most of the information. Mathematically, PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by the projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate and so on. Let's say $\boldsymbol{X}$ as,

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$$

For any matrix $\boldsymbol{X} \in R^{n\times p}$, there exists an orthogonal matrix $U \in R^{n\times n}$ and $V \in R^{p\times p}$ such that $U^TXV = \sum$, Where, sigma, $\sum$ = diagonal $(\sigma_1, \sigma_2 \dots, \sigma_n) \in R^{n\times p}$, $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_n \geq 0$. The $\sigma_i's$ are called the singular values of X and the columns of U and V are called left and right singular vectors of X respectively. For n singular values, $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_d \geq \sigma_{d+1} = \sigma_{d+2} \dots \sigma_n = 0$. Where d is the number of positive singular values, others are zero. Now the rank of our original matrix X is d (Figure-1). Range of X is the span of $u_1, u_2, \dots, u_d$ which is equal to d. Originally, we had the dimension of X was $n \times p$ and now it is $n \times d$. Statistically, number of data is reduced from p to d. Now, $U^TXV = \sum$ becomes $X = U\sum V^T$, that's meaning, in original data we had p columns and now after SVD, we obtain d columns.

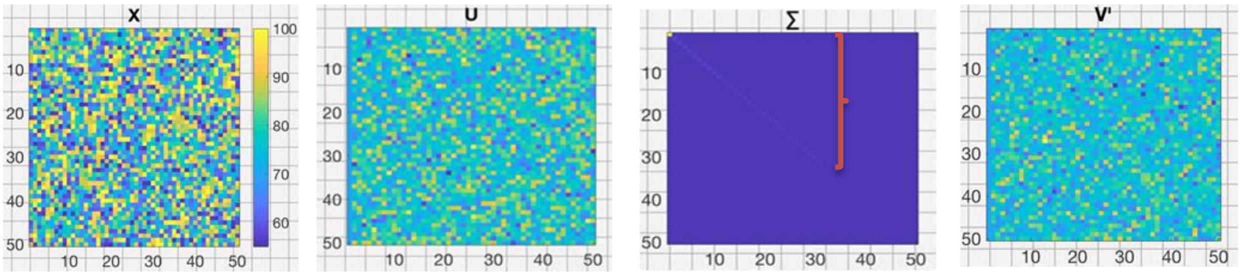


Figure 1 Full singular value decomposition to reduced singular value decomposition. Randomly generated matrix was used

From **Figure 1**, SVD is visualized using **imagesc** command from MATLAB library. Consider,

```
X = randi([55 100],50,50);

imagesc(X)

[U,S,V] = svd(X);

imagesc(U); imagesc(S); imagesc(V)
```

Now we can use principal component analysis (PCA) using this SVD technique. Originally, our X is like a column form

$$X = \begin{pmatrix} X_1 \\ X_1 \\ \vdots \\ X_n \end{pmatrix} \in R^p$$

With mean zero and covariance K. But now it is reduced by $X \in R^d$ which is our principal component. Please note, if X is not 0 then we should use $X - E(X)$ for centering.

Now the covariance matrix is K which looks like

$$K = \begin{pmatrix} cov(X_1, Y_1) & \cdots & cov(X_1, Y_n) \\ \vdots & \ddots & \vdots \\ cov(X_n, Y_1) & \dots & cov(X_n, Y_n) \end{pmatrix}$$

Covariance matrix is obtained by $cov(X) = K$.

Suppose U be the $p \times p$ orthogonal matrix such that the elements of the vectors $Z = U^TX$ are uncorrelated. The uncorrelated elements of the vector $Z = U^TX$ are called the component of X. So, our motto is to find this U which can be obtained from the SVD of covariance matrix K. Precisely, $cov(Z) = U^T cov(X) U = U^TKU = \sum$. Now we can compare this SVD.

Thus $Z_i = U_i^TX$ is the principal component. More elaborately $(Z_1, Z_2, \dots)$ and $(U_1, U_2, \dots)$ are the principal components and directions respectively. Here, Z_1has the largest variance and Z_2 has the second largest and so on. So, we can reduce the algorithm of PCA of given data as bellows where X denotes an independent observation vector $'X'$. Finding the sample covariance $K \in R^{p\times p}$ then compute the SVD of K to obtain orthogonal matrix $U \in R^{d\times n}$. Then we define principal directions by choosing first column of U. Thus, the principal component is

$$Z_{d\times 1} = U^T_{d\times n} X_{n\times 1}$$

Note that, if we have more correlated data then d will be very smaller than n.

One of the most useful demonstrations of PCA is the so-called Eigen faces example. There are two sets of images considered for this particular study. One is known as the training *set*; these images are already identified and labeled by some experts, say humans and the second one is *test images*; these are new images which need to be identified and labeled. Our aim is to use the similarities between the *test* and *training* images to label the *test* images. Linear Discriminant Analysis (LDA) will project a dataset onto a lower-dimensional space with good class-reparability in order to avoid overfitting ("curse of dimensionality") and also reduce computational costs.

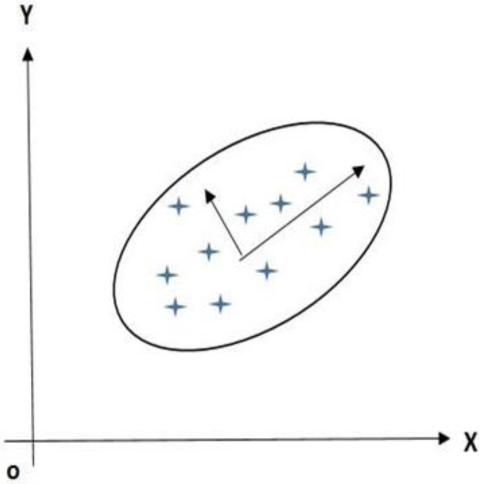


Figure 2 Principal component analysis

Ronald A. Fisher formulated the Linear Discriminant in 1936 (The Use of Multiple Measurements in Taxonomic Problems), and it also has some practical uses as a classifier. The original linear discriminant was described for a 2-class problem, and it was then later generalized as "multi-class Linear Discriminant Analysis" or "Multiple Discriminant Analysis" by C. R. Rao in 1948. The general LDA technique is congruent to a PCA, but in addition to finding the component axes that maximize the variance of our data (PCA), we are additionally interested in the axes that maximize the separation between multiple classes (LDA).

In LDA, the goal is to separate and characterize the observation after the projection. Similar observations should be closer and un-similar observations should be separated. In case of multiple class, let us take $C_1, C_2, \dots, C_m$ be m sets that partition of $\hat{R}$ into m classes.

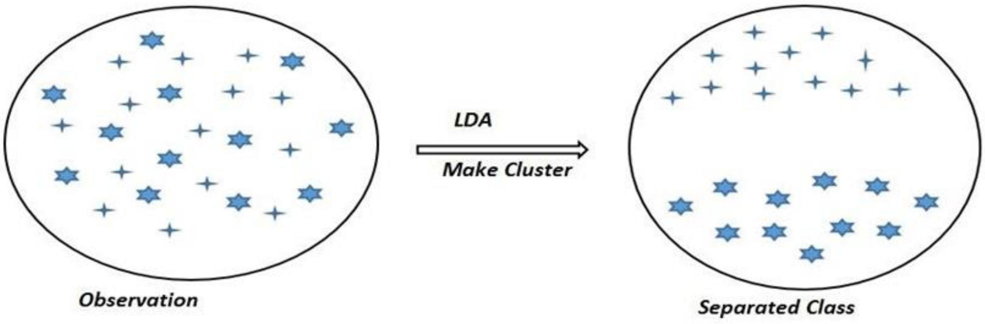


Figure 3 Linear Discriminant Analysis for two classes

We are given, N_j observations from class labeled by C_j. That is $X_i^j \epsilon C_j$ for $i = 1,2, \dots, N_j$ & $j = 1,2, \dots, m$

Suppose, we have C_j class

$$C_1 = X_1^1, X_2^1, \dots, X_{N_1}^1 \rightarrow \mu_1 \epsilon R^n$$
$$C_2 = X_1^2, X_2^2, \dots, X_{N_2}^2 \rightarrow \mu_2 \epsilon R^n$$
$$* * *$$
$$C_m = X_1^{\mathrm{m}}, X_2^{\mathrm{m}}, \dots, X_{N_{\mathrm{m}}}^{\mathrm{m}} \rightarrow \mu_m \epsilon R^n$$

Each $X_i^j \in R^n$, the i^{th} observation in the j^{th} class. Let, μ_j be the mean of observation in class C_j.

$$\mu_j = \frac{1}{N_j}\sum_{i=1}^{N_j} X_i^j \in R^n$$

A matrix captures the separation between the classes in the between class scatter matrix.

$$S_B = \sum_{j=1}^{m}(\mu_j - \mu)(\mu_j - \mu)^T \in R^{n\times n}$$

Where $\mu = \frac{1}{m}\sum_{j=1}^{m} u_j$

A matrix captures the average separation between elements within the same class is captured by the within class scatter matrix.

$$S_w = \sum_{j=1}^{m}\left(\sum_{i=1}^{N_j}(X_i^j - \mu_j)(X_i^j - \mu_j)^T\right) \in R^{n\times n}$$

Now our goal is to find the projection, $Z = U^T X$. The between class scatter matrix becomes

$$S_B^Z = U^T S_B U$$

and the within class scatter matrix becomes

$$S_w^Z = U^T S_w U$$

Now the goal is to choose U that maximizes the following function

$$f(U) = \frac{determinant(S_B^Z)}{determinant(S_w^Z)}$$

$$= \frac{det(U^T S_B U)}{det(U^T S_w U)} \frac{\uparrow}{\downarrow}; therefore\ f(U) \uparrow$$

The optimal projection is $\widehat{U} = argmax\big(f(U)\big); with\ U \in R^{n\times d}$. Now, we have to find U that the quantity $\widehat{U}$ maximizes. $\widehat{U}$ can be solved as the generalized eigenvalue problem where we calculate $S_B\widehat{U}_i = e_i S_w \widehat{U}_i$; e_i represents the eigenvalues of the transformation matrix U. If S_B is nonsingular then we can find corresponding eigenvector by calculating $U = {S_B}^{-1}S_w$. In our case we will use built-in MATLAB code to find an eigenvector.

Now, for Simple Projection, the following ideas are used. We introduce two theorems of real analysis to present it.

In the inner product space W, if $x, y \in W\ s.t.\ \mathrm{y} \neq 0$, then the orthogonal projection of $x\ onto\ y$ is given by $proj_y x = \frac{<x,y>}{<x,x>} y$.

Theorem 1: For every vector x in the $W(Hilbert\ Space)$ and any nonempty closed convex set C in H where $C \subset W$, there exists a unique vector $p \in C$, then norm $\sqrt{< x - q, x - q >} = \parallel x - q \parallel$ is minimized thru any other vector $q \in C$.

Theorem 2: Let a projection on a vector space W is a linear operator $\mathrm{H}: W \rightarrow W$ such that $\mathrm{H}^2 = \mathrm{H}$, then H and $\mathrm{I} - \mathrm{H}$ are orthogonal projections.

Proof: For any $x, y \in W$, in the one direction

$$< Hx, (y - Hy) > \; = < (x - Hx), Hy > \; = 0$$

which is equivalent to $< x, Hy > \; = < Hx, Hy > = < Hx, y >$

Therefore, $\mathrm{H}^2 = \mathrm{H}$. Since, for being orthogonal projection, it needs to be self-adjoint.

On the other direction, or any $x, y \in W$,

consider, $Hx \in M, y - Hy \in N$, where M, N any two vector spaces and

$$< Hx, y - Hy > = < H^2x, y - H^2y > = < Hx, H(I - H)y > = < Hx, (H - H^2)y > = 0$$

Which implies, H and $I - H$ are orthogonal projections.

On the other direction, if H is orthogonal then it is self-adjoint.

So, $< x, Hy > \; = < Hx, y > \; \forall x, y \in W$ ■

Now we describe dimension reduction with orthogonal projection

For our high dimensional image data set,

$$X \in \mathbb{R}^{n \times p}, X = \{y_j\}_{j=1}^{p} \subset \mathbb{R}^n.$$

We need to map y into a lower dimensional affine subspace, $\bar{y} + N$. Note that, we keep the same notation of data set X instead of using Y to keep the similarity. In this case $\bar{y} = \sum_{i=1}^{p} \frac{x_i}{p}$, N is n dimensional linear subspace in $\mathbb{R}^p$ where $n < p$.

Used mapping obtained by orthogonal projection on $h \in \mathcal{G}_{n,p}$,

where Grassmannian $\mathcal{G}_{n,p} = \{h \in \mathbb{R}^{n \times p} : h^2 = h, h^T = h\}, rank(h) = n, range(h) = N$.

In this study, we compute the principal component (PCA) of training images and project them down to the smaller size. That is, each training image is now represented by a small vector size k. For the test image, we can project them down to a k vector, using the same projection and then we find the nearest image in the training set by computing k vectors. In the case of FDA, the low dimensional projection is determined by using scatter matrices. We will assume n_2 training images each for n_1 people in our database. The size of each image is $s_1 \times s_2$. These images are taken at different orientations, different facial expression etc. There are two parts, one is to analyze the training images and compute a projection to their principal k −dimensional subspace and comparing with training data. In this work we will start from the following. Let us consider vector in a matrix size $(s_1 \times s_2) \times (n_1 \times n_2)$ is the arrangement of training images and we call it Y_{train}. First n_2 columns are images of person 1, and next n_2 columns are images of person 2, and so on, with a total of $n_1 \times n_2$ columns. Then we use PCA, FDA and Simple projection and we compare these three procedures. To perform PCA from our dataset Y_{train}, we need to calculate SVD and designate U_1 be the first k columns of the orthogonal matrix U. The size of U is now $(s_1 \times s_2) \times k$. Now we will use FDA, where we still consider PCA to reduce the size of our data Y_{train} from $(s_1 \times s_2) \times (n_1 \times n_2)$ to $d \times (n_1 \times n_2)$, where $d = \frac{n_1 \times n_2}{2}$. So, we get a new matrix and let us call it $\tilde{Y}_{train}$ and call the $(s_1 \times s_2) \times d$ projection matrix U_0. Use all the vectors from the same person as observations from the same cluster. In $\tilde{Y}_{train}$, the first n_2 columns designate the first person, the second n_2 columns designate the second person, and so on. We have two scatter matrices; one is between class and another is with-in class scatter matrices and use the generalized eigen decomposition to find k eigenvectors that correspond to the largest eigenvalues. So, we get this result as a matrix form and say it submatrix V. Now we find the orthogonal columns $d \times k$ matrix. Define a $(s_1 \times s_2) \times k$ orthogonal matrix $U_1 = U_0V$.

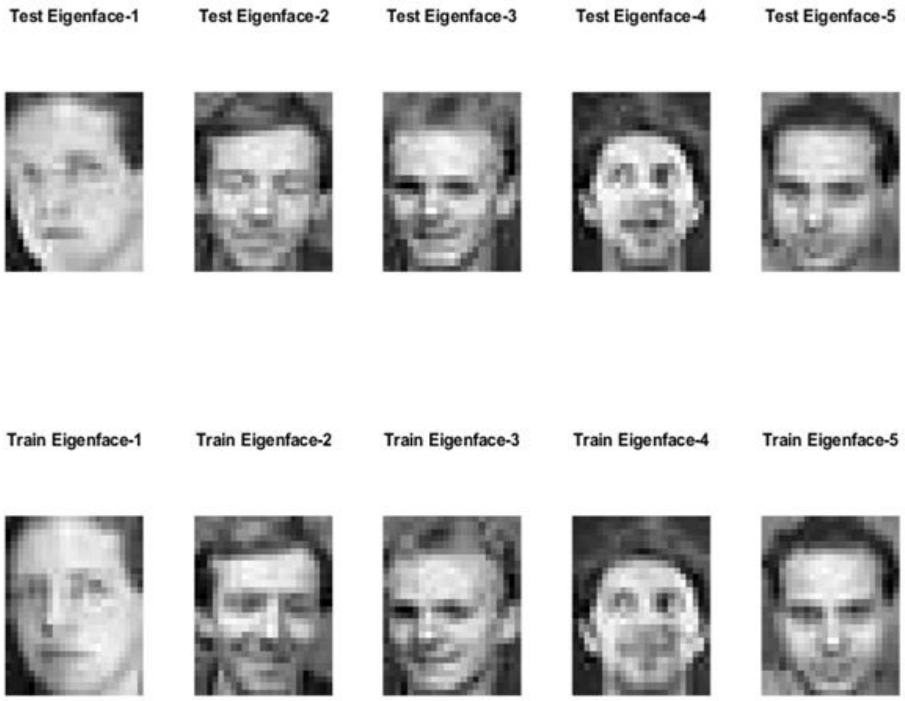


Figure 4 Examples of Eigen faces for test and train datasets

We will use simple projection for comparison with PCA and FDA. In this case, we will take another projection where U_1 is simply first k columns of $(s_1 \times s_2)$ identity matrix. Now, we will repeat the following for the upper three cases one by one. We will use the projection $Y_1 = U_1{}^T\ Y_{train}$ with size $k \times (n_1 \times n_2)$, which simplify each image in Y_{train} in a reduced form of k −dimensional vector. After using this procedure for each case, we can perform classification. We have n_2 test images per person and the test data set Y_{test} is in the same form as the training data set Y_{train}. We will take an image I randomly from the test set which represents a random column from Y_{test} actually. Projection of I can be found from $I_1 = U_1{}^T I$ where I_1 is a $k \times 1$ vector. Using l_2 norm we can find the distance between I_1 and each column of Y_1. Now, to find the label of the column that has the smallest distance to I_1. If this label matches the true label, then our recognition is successful otherwise it is a failure. For computing the percentage of successful recognition $F(k)$, we perform the above procedure 500 times, which will also give us the average performance of the upper three methods. After plotting k against each of the three projections we can easily discuss the performance. For these images the image size is $s_1 = 28$ and $s_2 = 23$ where $n_1 = 40$ which is the number of people. The number of training images per person is $n_2 = 5$. So, we have the matrix size of Y_{train} is $(s_1 \times s_2) \times (n_1 \times n_2)$, i.e. $(28 \times 23) \times (40 \times 5)$. We take 5 different training as well as test images from the data set and compare them with each other.

3. Results and discussion

We have summarized the result by using tables and graphs. Then we have shown an example of sets of images when the dimensionality parameter k changes then the performance $F(k)$ becomes improved so that the quality of image gets better. We demonstrate it by a few examples using test images and the closest images in the training set finally. Observation we wish to make from **Table 1** is that regarding our training datasets has to do with relative behavior of Simple projection (SP), LDA and PCA as the dimensionality parameter k becomes larger. The performance of these transforms gets better as the value of k increases. What's the difference between the three is that the recognition rate with PCA saturates around 4% to 89% when k varies from 1 to 40 while the performance of simple projection and LDA/FDA vary widely from PCA. For the experiments under discussion, the average performance of correct recognition best for $F_{PCA}(k) = 89.2\%$ when $k = 27$, $F_{PCA}(k) = 74.2\%$ when $k = 39$, $F_{SP}(k) = 34.4\%$ whn $k = 40$. From the table we can conclude that the values of $\boldsymbol{F(k)}$ is increasing with the increase of $\boldsymbol{k}$ where PCA shows relatively best performance and SP reflects weaker performance.

Table 1 Representation of average performance of correct recognition $\boldsymbol{F(k)}$ versus the number $\boldsymbol{k}$ for three cases

$\boldsymbol{k}$	$\boldsymbol{F_{PCA}(k)}$	$\boldsymbol{F_{FDA}(k)}$	$\boldsymbol{F_{SP}(k)}$	$\boldsymbol{k}$	$\boldsymbol{F_{PCA}(k)}$	$\boldsymbol{F_{FDA}(k)}$	$\boldsymbol{F_{SP}(k)}$
1	0.114	0.12	0.162	21	0.876	0.836	0.45
2	0.414	0.228	0.19	22	0.872	0.868	0.462
3	0.554	0.438	0.248	23	0.86	0.816	0.464
4	0.608	0.54	0.214	24	0.86	0.862	0.486
5	0.648	0.614	0.246	25	0.862	0.858	0.48
6	0.676	0.622	0.334	26	0.864	0.824	0.46
7	0.752	0.666	0.38	27	0.864	0.872	0.514
8	0.752	0.746	0.398	28	0.882	0.868	0.52
9	0.81	0.782	0.41	29	0.896	0.892	0.566
10	0.846	0.778	0.426	30	0.87	0.9	0.584
11	0.83	0.778	0.408	31	0.872	0.886	0.584
12	0.834	0.792	0.476	32	0.9	0.888	0.578
13	0.818	0.84	0.474	33	0.868	0.876	0.574
14	0.868	0.838	0.504	34	0.884	0.878	0.602
15	0.862	0.85	0.484	35	0.868	0.902	0.572
16	0.832	0.818	0.45	36	0.872	0.912	0.602
17	0.81	0.828	0.474	37	0.854	0.888	0.598
18	0.842	0.856	0.48	38	0.878	0.878	0.534
19	0.844	0.868	0.478	39	0.846	0.894	0.598
20	0.842	0.852	0.484	40	0.888	0.916	0.662

Here we calculate the average performance. After performing PCA, FDA and Simple projection 500 times and computing the percentage of successful recognition, we draw the graph of k versus $F(k)$. Figure-4 shows PCA is the most proficient technique compared to other two techniques. In fact, PCA > FDA > Simple Projection in our experiment. The performance of PCA is strictly increasing up to 35% when k runs up from 0 to 3 continuously, then again it grows continuously up to almost 60% with in the k interval 4 to 7 (app.). However, after that there are some fluctuations of performances. PCA achieves almost 90% accuracy when dimensionality parameter k becomes 26 which shows best performance in our analysis for PCA and it again declines a little and fluctuates close to 90% till k becomes 40. Performance of FDA and SimPro (SP) intersect each other when k is closed to 13, but it shows only 20% performance approximately. Then the FDA grows rapidly where SimPro does not. To validate our numerical results in table and graphical results with real images, we want to use our method (PCA, FDA and SP) to recognize new images and compare them with some images of known people. For this case, we have chosen different k values such as $4, 27, 40$ and take a person (say person 2) from the test image and compare with our experiment. **Figure6** is indicating the perfection of our study. MATLAB code is attached in **Appendix-01.**

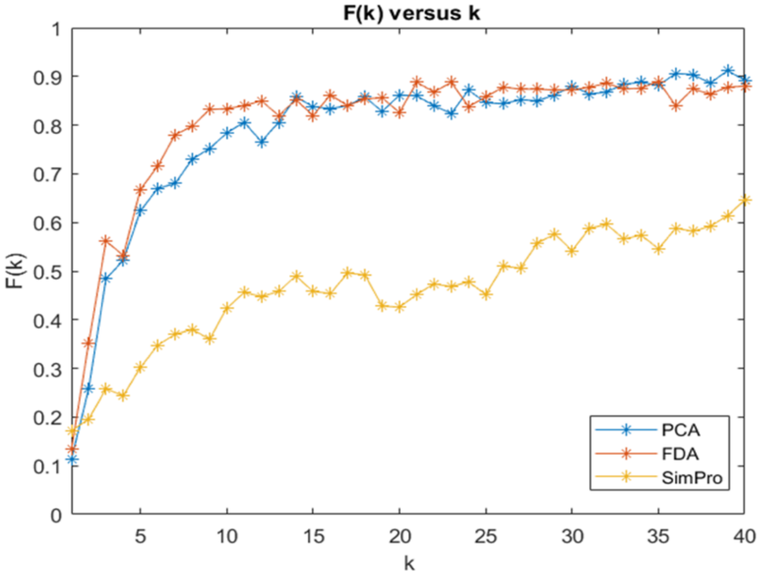


Figure 5 Variation between percentage of successful recognition $F(k)$ and the number k

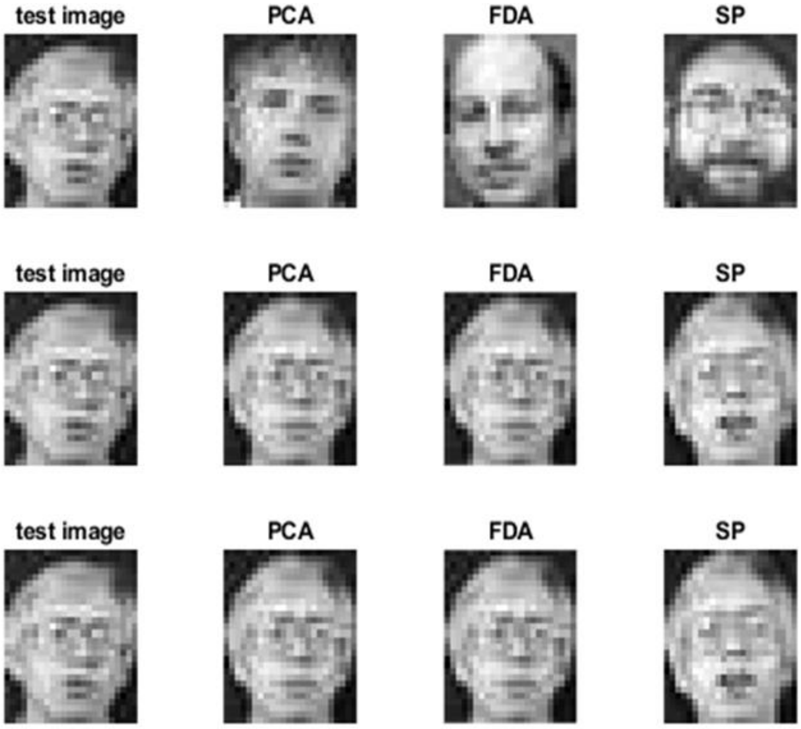


Figure 6 Comparison of Eigenfaces using test images with PCA, FDA and Simple Projection performed images

4. Conclusions

In this study, PCA, FDA and simple projection have been executed for face recognition. Although people might think that LDA always outperforms PCA since LDA deals directly with class separation, empirical evidence suggests otherwise. PCA might outperform LDA when the number of samples per class is small or when the training data non-uniformly sample the underlying distribution. For our face recognition problem, underlying distributions for different classes are unknown. So, in practice it would be difficult to assert whether or not the available training data is adequate for the job. The analysis we report validates our claim. That is, PCA is better performing than FDA. In addition, simple projection is easy to compute but which shows relatively less performance than PCA and FDA.

Data: https://cam-orl.co.uk/facedatabase.html

Codes: Appendix 01

Appendix 01: MATLAB Codes

```
close all;
clear all;

load TrainImages.mat;

load TestImages.mat;

% Computing the PCA of Ytrain

[U S V]=svd(Ytrain);   %SVD of Ytrain

[~,Index]=classifier(Ytrain,Ytest); % labeling the testing dataset

  for k=1:40
    tm=0;
    fm=0;
    for m=1:500

      U1=U(:,1:k); % U1 is the 1st k columns of the orthogonal matrix U
      Y1=U1'*Ytrain;  % Ytrain reduced to a k-dimensional vector using the projection
      %randomly choosing an image from the Ytest
      idx=randsample(200,1);
      I=Ytest(:,idx);
      I1=U1'*I;   % projection of I
      T_index=Index(idx); %index of our desired image in Ytrain data
      [~,ridx]=classifier(Y1,I1); % index of the recognized image
      if ridx==T_index
        tm=tm+1;
      else
        fm=fm+1;
      end
    end
    tm;
    F_PCA(k)=tm/500;
  end
figure(1);
plot(F_PCA,'r--');
%Computing the FDA of Ytrain

d=200/2;  % define d acording to our problem given

U0=U(:,1:d); % computation of the projection matrix

Ytrain_new=U0'*Ytrain;  % constructing the new training matrix

[V,lambda]=lda(Ytrain_new); %calculation of the matrix V

V=orth(V);  %to make the column orthogonal

for k=1:40
    tm=0;
    fm=0;
    for m=1:500
```

```
         V1=V(:,1:k);

         U1=U0*V1; %orthogonal matrix

         Y1=U1'*Ytrain;  %projected reduced Ytrain data

         idx=randsample(200,1);  %randomly selecting image from test data set
         I=Ytest(:,idx);
         I1=U1'*I;   % projection of I
         T_index=Index(idx); %index of our desired image in Ytrain data
         [~,ridx]=classifier(Y1,I1); % index of the recognized image
         if ridx==T_index
            tm=tm+1;
         else
            fm=fm+1;
         end
      end
      tm;
      F_FDA(k)=tm/500;
end
hold on;
plot(F_FDA,'g');

%Computing the Simple Projection of Ytrain
%finding projections for different values of k(after reducing the dimension)

U=eye(644);

   for k=1:40
      tm=0;
      fm=0;
      for m=1:500

         U1=U(:,1:k);  %orthogonal matrix

         Y1=U1'*Ytrain; %projected reduced Ytrain data

         idx=randsample(200,1); %randomly selecting image from test data set
         I=Ytest(:,idx);
         I1=U1'*I;   % projection of I
         T_index=Index(idx); %index of our desired image in Ytrain data
         [~,ridx]=classifier(Y1,I1); % index of the recognized image
         if ridx==T_index
            tm=tm+1;
         else
            fm=fm+1;
         end
      end
      tm;
      F_Simple(k)=tm/500;
   end
   hold on;
   plot(F_Simple,'b--o');

   legend('PCA','FDA','SimPro');
   xlabel('k');
   ylabel('F(k)');
 title('Variation of F(k) versus k')
  T=table(F_PCA',F_FDA',F_Simple');
```

Compliance with ethical standards

Acknowledgments

Authors are thankful to Dr. J. Y. Su of Texas Tech University for his insights.

Disclosure of conflict of interest

There is no conflict of interests.

References

[1] RE Twogood, FG Sommer. Digital Image Processing, in IEEE Transactions on Nuclear Science. June 1982; 29(3): 1075-1086.

[2] Fisher RA. The use of multiple measurements in taxonomic problems, Annals Eugenics. 1936; 7: 179–188.

[3] Duda R, Hart P. Pattern classification and scene analysis. A Wiley-Interscience publication. 1973.

[4] A Alkandari, SJ Aljaber. Principle Component Analysis algorithm (PCA) for image recognition, 2015 Second International Conference on Computing Technology and Information Management (ICCTIM). 2015; 76-80.

[5] Y Sun, L Li, L Zheng, J Hu, W Li, Y Jiang, C Yan, Image classification base on PCA of multi-view deep representation, Journal of Visual Communication and Image Representation. 2019; 62: 253-258.

[6] M Benito, D Peña. A fast approach for dimensionality reduction with image data, Pattern Recognit. 2005; 38(12): 2400-2408.

[7] M Partridge, RA calvo. Fast Dimensionality Reduction and Simple PCA, Intelligent Data Analysis. 1998; 2(1–4): 203-214.

[8] J Ma, Y Yuan. Dimension reduction of image deep feature using PCA, J Vis Commun Image Represent. 2019; 63.

[9] Xiao-Yuan Jing, D. Zhang, Yuan-Yan Tang. An improved LDA approach," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). Oct. 2004; 34(5): 1942-1951.

[10] GF Lu, J Zou, Y Wang. Incremental complete lda for face recognition, PR. 2012; 45(7):2510-2521.

[11] PN Belhumeur, JP Hespanha, DJ Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, in IEEE Transactions on Pattern Analysis and Machine Intelligence. July 1997; 19(7): 711-720.