Principal Component Analysis and K-Means Clustering – Biometrics

Mohit Mewara
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, USA
Email: mohitmewara@ufl.edu

***Abstract*—Biometrics plays uses physical characteristics like face, fingerprint, and other features for identification of a person. Face recognition system is a computer application which extracts the features of image and compares them with the database to return the closest match. The focus of this paper is to analyze the Principal Component Analysis(PCA) algorithm and K-Means Clustering for face recognition. PCA is used for important feature extraction and dimensionality reduction of face image which can be stored in database and can be compared with new faces for recognition or authentication. Experimental results show that the performance of Face Recognition System with PCA is equivalent to the system without PCA. It is also observed that the K-means clustering performs well on the face images after PCA reduction.**

***Index Terms*— Face Recognition, PCA, K-Means**

I. Introduction

Face Recognition has been extensively researched and used in applications like access control, identity authentication, camera surveillance, face-based video indexing and browsing since last decade. Face recognition lacks performance and uniqueness like fingerprint analysis, iris scanning, and other biometrics, however its ability to identify person without his cooperation, makes it suitable for law enforcement purposes. Feature extractor is the most important component of Face Recognition system. Some of the feature extractors used in Face Recognition system are Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), Fourier Transform (FT), and Principal Component Analysis (PCA) [1]. The extractor which extracts the most important features and retains the maximum properties of object is regarded as the efficient extractor [2].

Developing a computational system to recognize face is difficult because the images of face are multi-dimensional, complex and varies a lot depending on the lighting conditions or expression of the subject. There are broadly two challenges while developing Face Recognition system – extracting the appropriate features and modification of existing and new image based on extracted feature.

Out of the above specified feature extractors, we have used Principal Component Analysis (PCA) to build Face Recognition system. It uses orthogonal principal components. We can choose the number of principal components as per our needs. The first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components [2]. PCA reduces the dimensions of face image and retains the features with maximum variance. Reduction of dimensions is important because with the advancement of technology, the dimensions of images are increasing drastically. A normal cell-phone can capture a full HD image of resolution 1920 × 1080. The images with such high dimensions are computationally very expensive and are very hard to compute in real-time. Thus, to curb this problem, PCA is used widely.

The 2-D image is represented in 1-D space vector and then using principal components analysis, eigenvectors are computed. The eigenvectors with maximum variance are selected and are known as eigenfaces. The face images are then projected on the eigen space composed of eigenfaces. The algorithm is thoroughly explained in the next section.

II. Principal Component Analysis

A. *Algorithm*

The face image is represented in the 2-dimensional coordinate system as N × N. The image is converted into a column vector of N^2 to form a matrix of M rows, where each row represents a face image. The M images of training set can be represented as P_1, P_2, ... P_N.

$$x_i = [p_1 \dots\dots . p_n]^T, i = 1, \dots\dots ., M \quad (1)$$

The images are mean centered by subtracting the mean image from each image vector. Let m represent the mean image [3].

$$m = \frac{1}{M}\sum_{i=1}^{M} x_i \quad (2)$$

And let w_i be defined as mean centered image

$$w_i = x_i - m \quad (3)$$

We have to find a set of eigenvectors e_i which have the largest projection on each mean centered image w_i. We need to discover M eigenvectors for which the eigenvalues are maximum. The eigenvalue λ_i and eigenvector e_i can be calculated using the covariance matrix

$$C = WW^T \tag{4}$$

Where W is, the matrix created using M mean centered images w_i and each row of the matrix M represents an image. However, the size of the covariance matrix C is enormous as images have large dimensions. M images of size 50 × 50 will make a covariance matrix of size 2500 × 2500. Solving the matrix of such a huge size is computationally very expensive and we must find a different approach to calculate the covariance matrix. Another approach is to solve the covariance matrix W^TW of size M × M which is computationally feasible and then calculate e_i and λ_i. Let d_i and μ_i be the eigenvectors and eigenvalues of W^TW respectively.

$$W^T W d_i = \mu_i di \tag{5}$$

Multiplying W at both sides

$$W W^T (W d_i) = \mu_i (W di) \tag{6}$$

Let $e_i = W d_i$

$$C e_i = \mu_i e_i \tag{7}$$

Looking at the equation 7, it can be deduced that WW^T and W^TW has the same eigenvalues and their eigenvectors are related as $e_i = Wd_i$. We need to normalize Wd_i in order to equal it to e_i. The eigenvectors represent a subspace within which most of the information of the image is represented with nominal error. The maximum variance in the image is represented by the maximum eigenvalue and the least variance by least eigenvalue. To determine the appropriate eigenvectors, eigenvalues are sorted from larger to lower values and eigenvectors corresponding to higher eigenvalues are selected. The eigenvalues decrease exponentially, and by picking 10% of the top eigenvalues, roughly 90% of the variance is captured [3].

The facial image can be represented into M` dimensions by projecting the image on the face space. M` is less than or equal to M.

$$\Omega = [v_1\, v_2 \,.....\, v_{M`}]^T \tag{8}$$

Where $v_1 = e_i^T w_i$. The projection of first image is v_1, the projection for second image is v_2 and so on. The eigenvector e_i is also known as eigenfaces or eigenimages [4]. This name convention is given to these eigenvectors because they look like faces. So, **Ω** describes the contribution of each eigenface in representing the facial image by treating the eigenfaces as a basis set for facial images.

B. *Recognition*

Whenever a new face image is fed into the Facial Recognition system, Euclidean distance between the projected images stored in the database and the projection of new image on the face space is calculated. The Euclidean distance ε is defined as

$$\varepsilon_k^2 = \|\, \Omega - \Omega_k \,\|^2, \; K = 1,2 \,....\, M \tag{9}$$

Where **Ω** is the test image projection over the face space, and Ω_k is the projection of every image stored in the database.

The test face image is recognized, if the Euclidean distance is below a pre-defined threshold θ. The threshold is defined as

$$\theta = \frac{1}{2} \max_{i,k} ||w_j - w_k|| \tag{10}$$

III. K-Means Clustering

K-Means is an unsupervised learning algorithm that performs clustering by partitioning n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster [5].

Steps for K-Means Clustering

1. For a data set X = {x_1, x_2, …………., x_n}, randomly select some seeds which represents 'c' cluster centers.
2. Compute the Euclidean distance between cluster mean and each data point.
3. Assign each data point to the cluster with minimum Euclidean distance.
4. Calculate the mean of each cluster after the iteration using
$$M_i = \frac{1}{c_i} \sum_{j=1}^{c_i} x_i \tag{11}$$
Where C_i represents the number of points present in the cluster.
5. Recalculate the distance between new cluster means and each data point.
6. Repeat the process from 2 to 5 again, until no reassignment of data point is obtained.

IV. Implementation And Results

For this project, we have used 2 data sets – Gallery Set and Probe Set. The system is trained over gallery set and the testing has been done over probe set. The training set consists of 100 grayscale images of dimensions 50 × 50. And the testing set consists of 200 grayscale images of dimensions 50 × 50.

A. Face Recognition Algorithm

The steps involved in face recognition are summarized below:

1. Using the training face images, compute the mean.
2. Subtract each image with the mean to make a set of mean-centered images.
3. Select the eigenfaces from the eigenvectors of training set.
4. Project the mean-centered image on the face space.
5. For a new testing image, project the mean-centered testing image onto the face space.
6. Compute the Euclidean distance between the projected testing image and each projected training image and select the image with minimum distance.
7. If the minimum distance is below the threshold, then the system recognizes the face.

B. EigenFace Selection & Recognition Rate

We are selecting a subset of eigenfaces from the eigenvectors to make the computation less expensive. But we have to check, whether there is a tradeoff between performance and computation. The performance of Facial Recognition system is tested by selecting eigenfaces incrementally from 10 to 100.

It is observed that the accuracy of system improves as number of eigenfaces increases. With more eigenfaces we are including more features in the Facial Recognition System which can be visualized by the below graph.

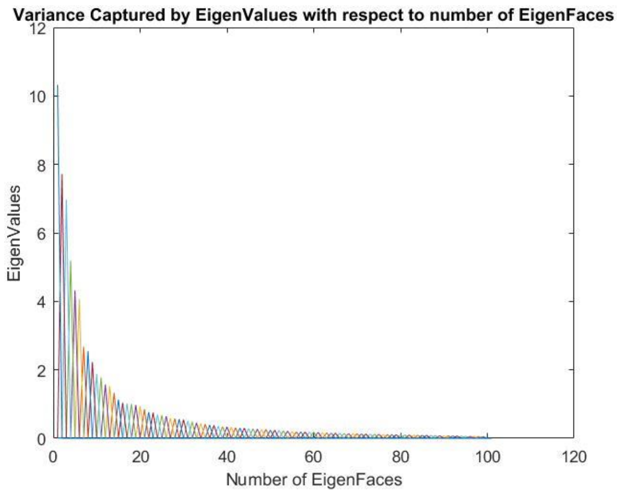


Figure 1: EigenValues with respect to number of Eigenfaces

Figure 2 shows that if we include more than 90 eigenfaces in our system, then the system will behave most efficiently. Also, it is observed that the performance increases rapidly for initial increase in eigenfaces but after 70 eigenfaces, the performance reaches 76.5% and the improvement in performance diminishes. And after 80 eigenfaces, the performance of the system stagnates at 77%, indicating that the system has reached the state of maximum performance. This observation is synonymous with the eigenvalues because more than 95% of the variance is included in the system by selecting 70 eigenfaces and almost 100% by including more than 80 eigenfaces. There is increase of only 0.5% in performance by increasing eigenfaces from 70 to 80. Thus, it can be concluded that including any number of eigenfaces above 70 will make the system perform better.

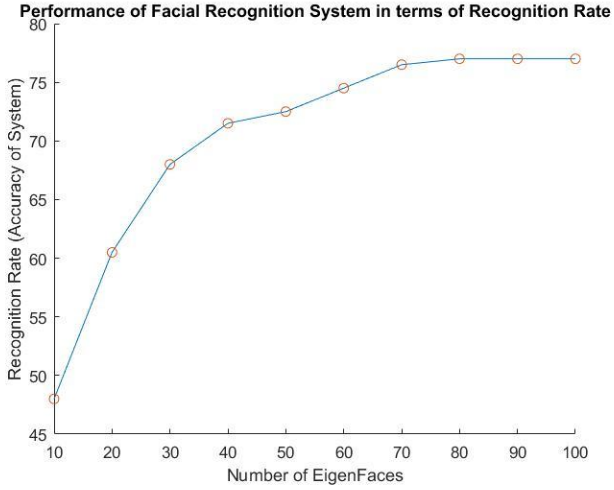
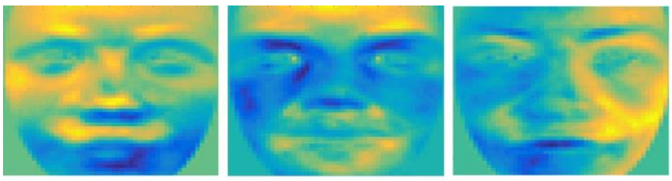


Figure 2: Performance of Facial Recognition System

The first three eigenfaces represents the maximum variance in the data. These eigenfaces can be seen in figure 3. They look like a ghostly face. The first eigenface with the maximum variance represents the illumination in terms of facial recognition. The second eigenface represents the brightness. For better performance of facial recognition system, sometimes the first eigenface is not considered because it does not contain the facial features. But, in our project the first eigenface contains the facial features apart from illumination and discarding it reduces the performance from 77% to 67%. Thus, for best performance, the eigenface is included in our Face Recognition System.



First Eigenface | Second Eigenface | Third Eigenface

Figure 3: First three Eigenfaces

C. Performance of Face Recognition System with and without PCA

We are performing PCA for dimensionality reduction, extracting the most relevant features and performing the computation fast. But, we should compare the performance of the Facial Recognition System after performing PCA with the Facial Recognition System without PCA to ensure that the performance of system does not decrease by including PCA.

The Face Recognition System with PCA performs best for eigenfaces above 80. So, comparing the best performance of Face Recognition System using PCA with the naïve Face Recognition System shows that the performance of both the systems remain same. This observation shows that it is better to use PCA with Face Recognition System because by using PCA we have decreased the dimensions of face images from 2500 to 100 (way too less) without compromising the performance of the system.

The below table shows that the maximum performance in both the systems remain same.

	Facial Recognition System With PCA	Naïve Facial Recognition System Without PCA
Performance (%)	77	77

Table 1: Performance comparison between Naïve Facial Recognition System(Without PCA) and Facial Recognition System with PCA

D. Facial Expression

The results show that the maximum performance of system is 77% instead of 100%. This means that there are some features present in the testing images which were not present in the training images. While analyzing the images present in the gallery set and the probe set, it is observed that in testing set there are 2 images of each person. Figure 4 shows that one of the image is with expression and the other is without expression. However, all the images of training set are without expressions. The facial expressions accounts for comparatively lower recognition accuracy of system.

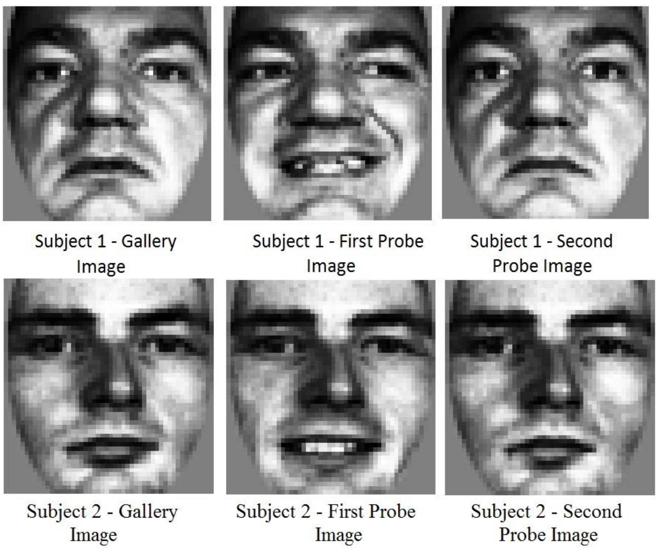


Figure 4: The First Image is from Gallery Set while second and third images are from Probe Set.

E. K-Means

Soft biometric classification can be used to improve performance of Face Recognition System by reducing the search space. Soft biometrics traits are ethnicity, weight, height, age, gender, or any other characteristics which are reliable. They are not unique but provides some information about the user. In this project, we are using the gender trait to perform soft biometric classification due to its availability in the given data set.

K-means clustering is used to classify the data into 2 classes - Male and Female. Our aim is to perform K-means clustering for the testing images projected over the face space to figure out the difference in performance of K-means clustering after performing PCA. We have previously established that using PCA reduces the dimensions of images which helps to perform computation faster. And if we get the performance of K-means using PCA equivalent to the clustering without PCA then it would be a better approach to use PCA prior performing clustering.

One of the major drawback of clustering algorithms is that the clustering initialize using random seeds. The clustering depends heavily on the selection of these random seeds. There are many other approaches like X-means, K-Means++, which uses the option of selecting initial seed, instead of random initialization for better results. However, to increase the performance of clustering, we have executed K-Means multiple times for the same projected images to minimize the cost function. The clusters are finalized when the cost function is minimum. The cost function of K -Means is the sum of the squared distance between each observation and its closest centroid.

There are 2 main categories of clustering validation:

1. *Internal Criteria*

The goal of clustering is to make objects within the same cluster similar and objects in different clusters distinct. Internal validation measures are often based on the following two criteria [6]:

- *Compactness*: It measures how closely the objects are related in a cluster.
- *Separation*: It measures how distinct or well-separated a cluster is from other clusters.

We have used two internal criterion – Silhouette and Calinski-Harabasz for clustering validation. Initially, the original image is clustered by K-Means using both criterion. It is observed that the optimal number of clusters are 2 for both the criterion which is synonymous with the available information. As all the features are included in these images, it can be noticed that there should be 2 clusters for the given data.

Later, to calculate the performance difference in K-Means after using PCA, PCA is performed over the training images to obtain the Face Space. All the images are included for clustering – 100 from training set and 200 from testing set. To observe the trend of number of clusters identified by K-Means, the number of eigenfaces are increased by 1 with each iteration. Figure 5 and 6 shows the clustering results using Silhouette criterion and Calinski-Harabasz criterion after 100 iterations. For low values of eigenfaces (< 2), the number of clusters are not identified as 2 but as the number of eigenfaces increases, the optimal number of clusters is found out to be 2.

This is evident from the fact that as the number of eigenfaces increases, the more features (information) is available to the K-Means algorithm to perform better clustering. But it can also be inferred that the algorithm learns early and after a point (3 eigenfaces), the performance remains same.

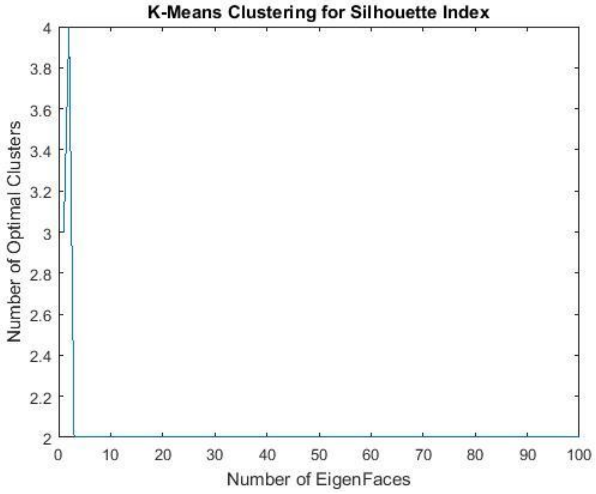


Figure 5: K-Means Clustering using Silhouette Criterion

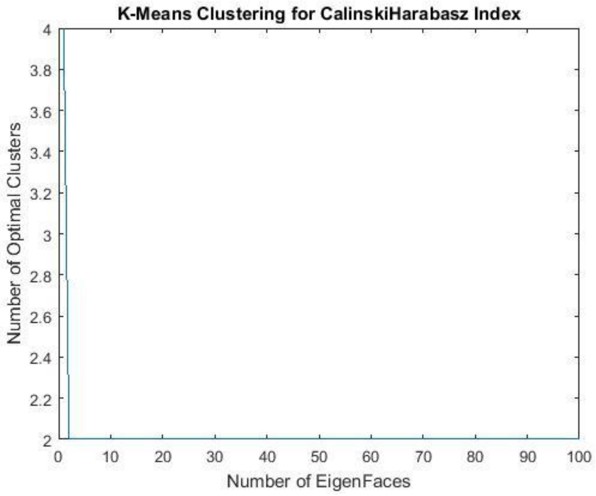


Figure 6: K-Means Clustering Using Calinski-Harabasz Criterion

2. External Criteria

External criteria use external information which is not present in the data. Mostly, class labels are used as external information because they are available. We have used the gender class labels in this application. There are four types of external criteria but we are using F-measure. The F-measure is calculated using the below equations:

$$P = \frac{TP}{TP + FP} \tag{12}$$

Where P is precision, TP is True Positive, and FP is False Positive.

$$R = \frac{TP}{TP + FN} \tag{13}$$

Where R is recall, TP is True Positive, and FN is False Negative.

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{14}$$

$$\beta > 1$$

Using the above formula, the External criteria of test images obtained using PCA is computed.

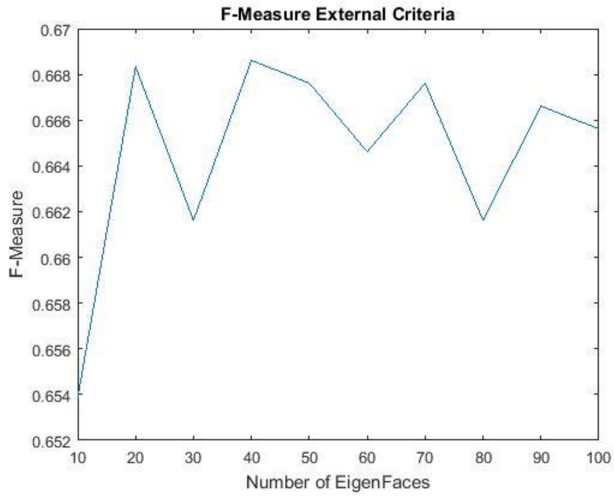


Figure 7: F-Measure External Criterion

The above graph shows that the trend is random. The value of F-Measure is changing randomly with the increase in number of eigenfaces. This means that unlike PCA, which shows better performance with increase of eigenfaces, the performance of F-Measure is not showing any trend with increase in number of eigenface. Because these values are not showing any trend, it is difficult to infer anything. The best inference would be that we can’t say anything about the value of f-measure as the number of eigenfaces (principal components) increase in the system.

Previously, it was noticed that the performance of Facial Recognition System increases with the increase in eigenfaces because the more eigenfaces included in the system, the more information is learned by system. However, in clustering it can be seen that after 20 eigenfaces, the values are fluctuating between a very less window. This happens because the clustering algorithm learns the system with less eigenfaces and it does not need more information for better performance. The values simply rattle with increase in number of eigenfaces.

The *F-measure of Face Recognition System without PCA* is **0.6656** which is almost *equal to the system with PCA*. The K-means with internal criterion and external criterion results shows that same clustering can be achieved by the system using PCA compared to the system without PCA. Thus, it is beneficial to use PCA because it significantly reduces the computation time and space.

CONCLUSION

The performance of the Face Recognition System with PCA and without PCA is compared in this report. It was experimentally concluded that the system with PCA performs as well as a system without PCA. Thus, using the system with PCA is beneficial as the computation time and space is significantly reduced by PCA. With the technological advancement, more dimensions are included in images and the staggering increase in the database size of face images, PCA would be a very helpful approach for faster search.

The performance of both systems is also compared for K-Means clustering and it is observed that the F-measure of both systems are same. These results prove that PCA is also an efficient method for K-Means clustering.

REFERENCES

[1] Surabhi A R, Shwetha T Parekh, K Manikantan, S Ramachandran. Background Removal using K-Means Clustering as a Preprocessing Technique for DWT Based Face Recognition. Proc. of ICCICT 2012 - International Conference on Communication, Information and Computing Technology, Mumbai, IEEE.

[2] Rabab M. Ramadan, Rehab F. Abdel-Kader, Face Recognition Using Particle Swarm Optimization-Based Selected Features, International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 2, No. 2, 2009.

[3] Kyungnam Kim, “Face Recognition using Principle Component Analysis”, International Conference on Computer Vision and Pattern Recognition, pp. 586-591, 1996.

[4] M.A. Turk and A.P. Pentland, “Face Recognition Using Eigenfaces”, IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.

[5] Tapas Kanungo , Nathan S. Netanyahu , Angela Y. Wu “An efficient k-means clustering algorithm: analysis and implementation”, IEEE Transactions on Pattern Analysis And Machine Intelligence, Vol. 24, No. 7, July 2002.

[6] Yanchi Liu, Zhongmou Li, Xuedong Gao, “Understanding of Internal Clustering Validation Measure”, 2010 IEEE International Conference on Data Mining.