Face Recognition Using Eigenfaces, Geometrical PCA Approximation and Neural Networks

Alina L. Machidon, Octavian M. Machidon, Petre L. Ogrutan
Department of Electronics and Computers
Transilvania University of Brasov
Brasov, Romania
Email: {alina.machidon, octavian.machidon, petre.ogrutan}@unitbv.ro

***Abstract*—The human face exhibits a high level of complexity when it is regarded as a multidimensional visual model, leading to face recognition systems that require difficult and extensive computations for coding and decoding the face images. A well-established approach in this regard is based on using principle component analysis (PCA) for both feature extraction and face recognition, known as the eigenface approach. This technique, despite a good recognition rate, suffers from the disadvantage of high computation cost due to the complexity of the PCA algorithm. In this paper, we use a geometrical approximated PCA (gaPCA) algorithm for computing the eigenfaces for three different datasets. The face recognition task is performed using a similarity score based on the inverse Euclidean distance for the first two datasets and using a nerual network in the third case. All the results are compared to the case where standard PCA is used. These accuracy results show that gaPCA represents a viable alternative to the classical statistical approach for computing the principal components.**

***Keywords*—eigenfaces; face recognition; feature extraction; neural networks; principal component analysis**

I. Introduction

Principal Components Analysis (PCA) [1] is a statistical procedure widely used in exploratory data analysis, as a dimensionality reduction technique and as a tool for making predictive models [2]. A well-known application field of PCA in computer science is the Eigenfaces project [3]; Kirby and Sirovich [4] were the first that showed that principal component analysis could be used on a collection of face images to form a set of basis features. They also introduced an algebric manipulation which made it easy to directly calculate the eigenfaces, and showed that fewer than 100 were required to accurately code carefully aligned and normalized face images [5]. Turk and Pentland [3] then demonstrated that the residual error when coding using the eigenfaces could be used both to detect faces in cluttered natural imagery, and to determine the precise location and scale of faces in an image for both face representation and recognition [5]. The PCA and eigenfaces approach has since been widely used in face recognition applications [6][7].

The main idea of the eigenfaces method consists of extracting the characteristic features of the face and representing the face as a linear combination called "eigenfaces". These eigenvectors are derived from the covariance matrix of the probability distribution of the high-dimensional vector space of all the database face images. The maximum number of eigenfaces is equal to the number of the training faces. When all the eigenfaces are computed, each face in the training set is projected onto the lower dimensional face space to find all of weights [3] that characterize the contribution of each vector in the face space.

It has been shown that using only the top 100-150 principal components is enough to represent almost all of the images with high accuracy [8]. Each of these principal components can be viewed as an image and in many cases interpreted directly. Being able to correctly retrieve this subspace is crucial in many applications such as face recognition and alignment. However, realistic face images often suffer from self-shadowing, specularities, or saturations in brightness, which make this a difficult task and subsequently compromise the recognition performance [9].

Despite being a viable method with good recognition accuracy results, PCA has the disadvantage of a relatively high computational cost and technical difficulties in algorithm parallelization [10]. In this context, we have previously proposed a geometric construction-based approach for PCA approximation (gaPCA) based on the observation that the direction given by the furthest points is, depending on the correlation of data, relatively close to the one given by the first principal component [11].

In this paper, we apply gaPCA for face recognition and compare the recognition accuracy with the one obtained by using standard PCA, on three different face databases. We describe the methodology used to perform the experiments and the accuracy metrics used, present and analyse the experimental results and draw the final conclusions.

II. Methodology

The geometrical approximated PCA (gaPCA) method approximates the n principal components of a dataset as a set of basis vectors $V = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_n}\}$. Given a set of points represented as n-dimensional vectors $P_1 = \{\mathbf{p_{11}}, \mathbf{p_{12}}, ...\} \subset R^n$, the first step is to identify the two elements $\{\mathbf{e_{11}}, \mathbf{e_{12}}\}$ in the set separated by the maximum (Euclidean) distance.

The first basis vector is the vector $\mathbf{v_1}$ that connects the two points: $\mathbf{v_1} = \mathbf{e_{11}} - \mathbf{e_{12}}$. All the other basis vectors belong to subspaces of R^n that contain the midpoint $\mathbf{m}$ of the segment that connects the two points: $\mathbf{m} = \frac{\mathbf{e_{11}}+\mathbf{e_{12}}}{2}$. The second basis vector is computed by projecting all the elements in P_1 onto the hyperplane H_1, determined by the normal vector $\mathbf{v_1}$ and containing $\mathbf{m}$:

$$H_1 = \{\mathbf{x} \in R^n | < \mathbf{v_1}, \mathbf{x} >=< \mathbf{v_1}, \mathbf{m} >\} \quad (1)$$

where $< \cdot, \cdot >$ represents the dot product operator. This results in a set of projections of the original points, $P_2 = \{\mathbf{p_{21}}, \mathbf{p_{22}}, ...\}$, computed using the following formula:

$$\mathbf{p_{2i}} = \mathbf{p_{1i}} + (< \mathbf{v_1}, \mathbf{m} > - < \mathbf{v_1}, \mathbf{p_{1i}} >) \cdot \mathbf{v_1} / ||\mathbf{v_1}||^2 \quad (2)$$

After the identification of the two projections separated by the maximum distance $\{\mathbf{e_{21}}, \mathbf{e_{22}}\}$, the second vector, $\mathbf{v_2}$, is computed as the difference between the two values. Consequently, every i-th basis vector is obtained by computing the projections of the points in the set P_{i-1} onto the hyperplane H_{i-1}, identifying the two projections separated by the maximum distance and computing $\mathbf{v_i}$ as the difference between the two.

The final result is the set of basis vectors $V = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_n}\}$. The representation of n-dimensional data using V is done using the dot product operator; thus, in the new representation, the i-th component c_i of a vector $\mathbf{x} = [x_1, x_2, ..., x_n]$ is computed as: $c_i =< \mathbf{x}, \mathbf{v_i} >$.

For performing the comparative assessment between standard PCA and gaPCA, we used three face datasets: Yale, Cambridge and Labeled Faces in the Wild (LFW). For all three datasets we computed the eigenfaces using both algorithms. In the first two cases, face recognition was performed by computing and a Euclidean Distance based metric, while in the last case the computed eigenfaces were used to train a neural network classifier that performed the actual face recognition.

A. Yale and Cambridge datasets

The Yale Face Database [12] contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration [12]. As in the case of the Cambridge dataset, we computed the eigenfaces and aimed to validate the gaPCA method from the perspective of face recognition performance, comparing the results with those achieved by the Standard PCA. The data set was divided in two subsets: a training set consisting of 135 images and a test set with the rest of 30 images. Due to the fact that this is a smaller database, we selected the top 20 eigenvectors, for each method.

The Cambridge Database of Faces [13] contains ten different images of each of the 40 distinct subjects [13]. This data set was also divided in two subsets: a training set consisting of 280 images and a test set with the rest of 120 images.

We implemented the face recognition task on the two datasets, using first the Standard PCA algorithm and then the gaPCA one. For all the faces in each test set, we searched, based on the eigenfaces provided by each method, the most resembling face in the corresponding training set.

To recognise faces, the first step was to project all the mean-shifted images of the training set into the subspace defined by the set of eigenvectors to generate the features vectors, which described the contribution that each eigenface has to each image of the dataset. Secondly, for each mean-shifted face image in the test set, the features vector was computed by projecting the image onto the collection of eigenfaces. This provided a set of features describing the test face. These features are then compared against all features in the training set to find the closest match.

Once the face images have been projected into the eigenspace, the similarity between any pair of face images was calculated by finding the Euclidean distance between their corresponding feature vectors; the smaller the distance between the feature vectors, the more similar the faces. The face in the training set associated with the highest similarity score was considered the most resembling with the face image provided from the test set.

We used a similarity score based on the inverse Euclidean distance:

$$S = \frac{1}{1 + d(p, q)} \quad (3)$$

Where $d(p, q)$ is the Euclidian Distance between two vectors $p = \{\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_n}\}$ and $q = \{\mathbf{q_1}, \mathbf{q_2}, ..., \mathbf{q_n}\}$, defined as the square root of the sum of squared differences between corresponding elements of the two vectors:

$$d(p, q) = \sqrt{\sum_{i=0}^{n} (p_i - q_i)^2} \quad (4)$$

For all the faces in the test sets, we searched, using the eigenfaces provided by each of the two methods, for the most resembling face in the training sets and computed the similarity score with the formula above. An overall accuracy result was computed for both gaPCA and standard PCA methods, on each of the two datasets.

B. LFW database

The second comparison between the standard Eigenfaces method and our gaPCA Eigenfaces Method was conducted on the Labeled Faces in the Wild (LFW) database [14].

The Labeled Faces in the Wild is a database of face photographs designed for studying the problem of unconstrained face recognition, containing more than 13000 images of faces collected from the web [14].

For this comparison, we first extracted a subset of the LFW database, containing only the individuals with at least 100 faces, resulting in 5 individuals and a total of 1140 faces. Secondly, we divided the 1140 images into a training and a testing set, the first having 70% of the images, and the second the rest of 30%. On the training set we performed the dimensionality reduction, applying the two PCA approaches separately to obtain the eigenfaces (Figure 4 and Figure 5). On this smaller subspace of faces provided by the PCA eigenvectors we applied a Multi-layer Perceptron classifier with 2 layers and 1024 neurons in the hidden layer, that takes the reduced-dimension input and produces a class label.

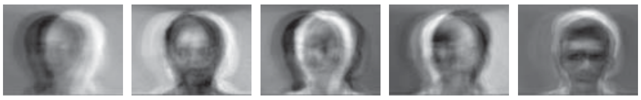
The accuracy results were computed for three cases: standard PCA, gaPCA and no PCA (the neural network was trained using the raw training data, no dimensionality reduction applied). The comparative evaluation included the number of iterations needed to successfully train the neural network classifier in each case.

III. Experimental results

A. Yale dataset

Figure 1 shows the first five principal components of the Yale faces database, after the computation of the Standard PCA algorithm (top) and the gaPCA algorithm (bottom). The recognition accuracy scores are shown in Table I.

The percentages are justified mainly due to the large variation in pose positioning of the faces, since on this data set no alignment pre-processing has been performed. However, it is interesting to point out that gaPCA's accuracy is very close to the accuracy scored by the Standard PCA (-3.33%).



(a) 1st standard eigenface (b) 2nd standard eigenface (c) 3rd standard eigenface (d) 4th standard eigenface (e) 5th standard eigenface

(f) 1st approximated eigenface (g) 2nd approximated eigenface (h) 3rd approximated eigenface (i) 4th approximated eigenface (j) 5th approximated eigenface

Fig. 1. First five eigenfaces from the Yale Database obtained with the Standard PCA (top) and the gaPCA (bottom) method

TABLE I. Face recognition accuracy on the Yale face database.

Method	Recognition accuracy
Standard PCA	76.66%
gaPCA	73.33%

B. Cambridge dataset

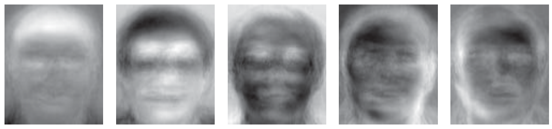
In Figure 2 we presented the first five principal components of the Cambridge database of faces, resulted after the implementation of the Standard PCA algorithm (top) and the gaPCA algorithm (bottom).

Table II shows the percent of correctly recognized faces after the implementation of the face recognition algorithm based on the eigenfaces previously computed with each of the two methods (Standard PCA and gaPCA). The results show that gaPCA scored very similar to the standard PCA. Both methods obtained high accuracy scores (over 90%) despite high variations in lighting, facial expressions and details.

TABLE II. Face recognition accuracy on the Cambridge face database.

Method	Recognition accuracy
Standard PCA	94.14%
gaPCA	93.33%

As shown in Figure 3, the accuracy of both methods exhibits unnoticeable variations when the number of eigenvectors retained is increased.



(a) 1st standard eigenface (b) 2nd standard eigenface (c) 3rd standard eigenface (d) 4th standard eigenface (e) 5th standard eigenface

(f) 1st approximated eigenface (g) 2nd approximated eigenface (h) 3rd approximated eigenface (i) 4th approximated eigenface (j) 5th approximated eigenface

Fig. 2. First five eigenfaces from the Cambridge Database of faces obtained with the Standard PCA (top) and the gaPCA (bottom) method

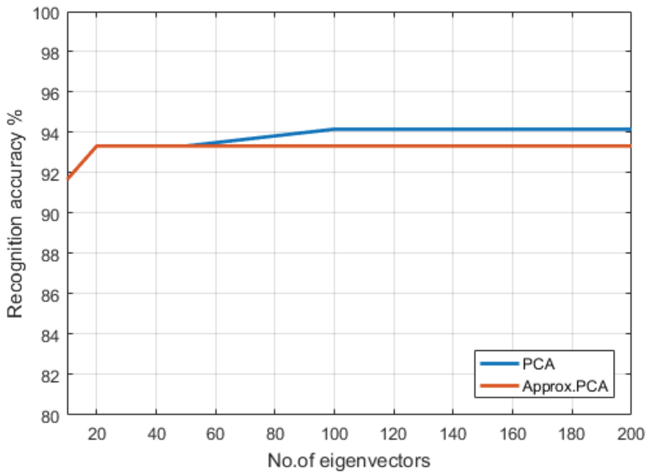


Fig. 3. Recognition accuracy vs. no. of eigenevectors for the Cambridge database of faces

C. LFW dataset

The first five eigenfaces of the LFW dataset are displayed in Figure 4 (standard PCA) and in Figure 5 (gaPCA). Table III shows the average of the precision of the classification for 5 different personalities using either no PCA, the Standard PCA or the gaPCA method.

In particular, one can notice that the Standard PCA has an overall precision of 77%, while the gaPCA approach's is very similar at 75%, as opposed to the NO PCA approach which is of only 68%. Furthermore, there is a particular class (e.g. "D.Rumsfeld") where the gaPCA scores better in labeling the faces than both the Standard PCA and the NO PCA approaches.

The results show that using a dimensionality reduction technique (either PCA or gaPCA) improves the overall precision of the classification, in addition to reducing the number of iterations needed to train the classifier, thus decreasing the time complexity (as shown in Table IV). The number of iterations needed to train the classifier is relatively high when no dimensionality reduction technique is used (48), lower with the gaPCA method (25) and even less with the Standard PCA approach (19). The neural network training process is approximately twice as faster when using the gaPCA algorithm, thus validating it as a efficient dimensionality reduction technique.

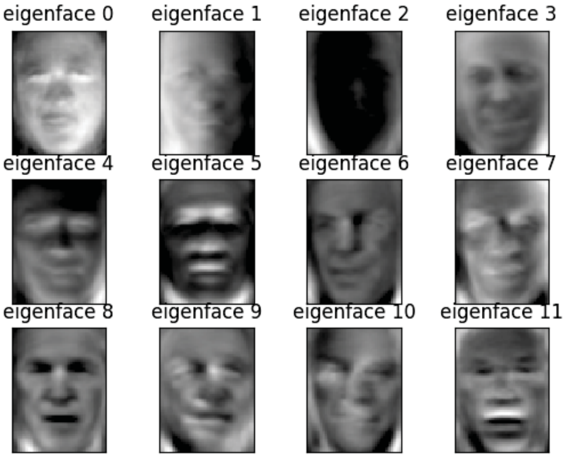


Fig. 4. Eigenfaces of the LFW database using Standard PCA.

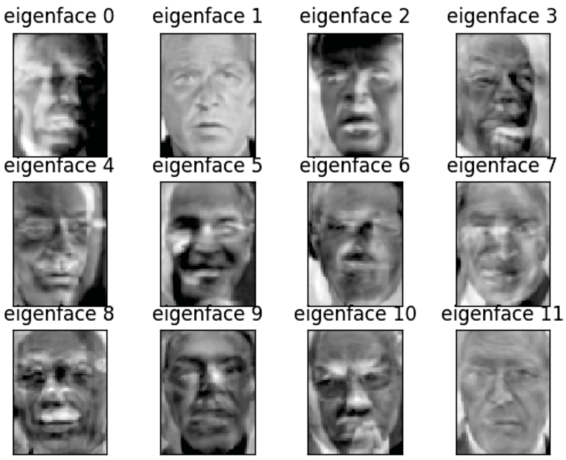


Fig. 5. Eigenfaces of the LFW database using gaPCA.

TABLE III. AVERAGE PRECISION FOR 10 CLASSIFICATION WITH NO PCA, STANDARD PCA AND GAPCA

Face name	No PCA	Standard PCA	gaPCA
C. Powell	78	83	81
D. Rumsfeld	56	70	72
G.W. Bush	86	88	85
G. Schroeder	53	74	68
T. Blair	67	71	68

TABLE IV. MEAN NUMBER OF ITERATIONS OF THE CLASSIFIER WITH NO PCA, STANDARD PCA AND GAPCA

Dimensionlity reduction	Mean number of iterations
No PCA	48.4
Standard PCA	19.2
gaPCA	25

IV. CONCLUSION

In this paper we described the use of a geometrical approximted PCA algorithm (gaPCA) for face recognition and compared the results with the standard PCA method.

Both PCA and gaPCA were applied to compute the eigenfaces for three datasets of faces: Yale, Cambridge and LFW. In the first two cases a face recognition algorithm was implemented computing the similarity score (based on the inverse Euclidian distance) between faces in a test set and the eigenfaces of the training set computed using PCA and gaPCA. The comparative recognition accuracy results showed that gaPCA scored equal or narrowly lower (under 4% overall average difference) compared to standard PCA.

For the last face dataset, the face recognition task was implemented using a neural network trained with the eigenfaces computed by the two methods. In this case, the recognition accuracy results were also very close, with gaPCA just under 2% in average compared to the standard PCA, with better results for specific classes. Furthermore, it was shown that gaPCA decreases the number of training iterations needed for the neural network classifier.

The results described in this article show that gaPCA represents a viable alternative to the classical statistical approach for computing the principal components. Having a less complex implementation than standard PCA, future research efforts will target the parallelization of the gaPCA algorithm in order to achieve faster execution times.

REFERENCES

[1] K. Pearson, “LIII. On lines and planes of closest fit to systems of points in space”, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559-572, 1901.

[2] J. Shlens, “A tutorial on principal component analysis”, *arXiv preprint*, arXiv:1404.1100, 2014.

[3] M.Turk, A.Pentland, “Eigenfaces for recognition ”, *Journal of cognitive neuroscience*, vol. 3, no 1, pp. 7186, 1991.

[4] M. Kirby, L. Sirovich, “Application of the karhunen-loeve procedure for the characterization of human faces ”, *IEEE Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, 1990.

[5] A. Pentland, T. Choudhury, “Face Recognition for Smart Environments ” *IEEE Computer*, no. 2, vol. 33 pp. 50-55, Feb. 2000.

[6] P. Wagh, R. Thakare, J. Chaudhari and S. Patil, “Attendance system based on face recognition using eigen face and PCA algorithms,” *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, Noida, 2015, pp. 303-308.

[7] A. L. Ramadhani, P. Musa and E. P. Wibowo, “Human face recognition application using pca and eigenface approach,” *2017 Second International Conference on Informatics and Computing (ICIC)*, Jayapura, 2017, pp. 1-5.

[8] T. Roughgarden, G. Valiant. “CS168: The Modern Algorithmic Toolbox Lecture# 7: Understanding and Using Principal Component Analysis (PCA)”, *Stanford CS168 Lecture Notes* [Online]. Available: http://web.stanford.edu/class/cs168/l/l7.pdf

[9] E.J. Candès, X. Li, Y. Ma, J. Wright, “Robust principal component analysis?” *Journal of the ACM (JACM)*, vol. 58, no. 3, 2011.

[10] M. R. Kawale, Y. Bhadke and V. Inamdar, ‘implementation of eigenface on CUDA”. In *2014 International Conference on Advances in Engineering and Technology Research (ICAETR - 2014)*, Unnao, 2014.

[11] A. Machidon, R. Coliban, O. Machidon, M. Ivanovici. “Maximum Distance-based PCA Approximation for Hyperspectral Image Analysis and Visualization”. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2018.

[12] P. Belhumeur, J. Hespanha, D. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 711-720, July 1997.

[13] F.S. Samaria, A.C. Harter. “Parameterisation of a stochastic model for human face identification ”. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pp. 138-142, IEEE, 1994.

[14] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller. “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.” *University of Massachusetts, Amherst, Technical Report 07-49*, October, 2007.