International Journal of Information Technology and Knowledge Management
July-December 2011, Volume 4, No. 2, pp. 427-430

PERFORMANCE EVALUATION OF FACE RECOGNITION USING PCA

Kiran Jain[1] & Sukhvir Singh[2]

The face recognition problem is difficult by the great change in facial expression, head rotation and tilt, lighting intensity and angle, aging, partial occlusion (e.g. Wearing Hats, scarves, glasses etc.), etc. The Eigenfaces algorithm has long been a mainstay in the field of face recognition and the face space has high dimension. Principal components from the face space are used for face recognition to reduce dimensionality. In this paper, the technique PCA is applied to find the face recognition accuracy rate and Kernel PCA is described.

Keywords: Facial Expression, Intensity, Eigenfaces, PCA, KPCA

1. Introduction

Face recognition is gaining enormous interest nowadays. However, the technical challenges to "teach" a computer to recognize faces have been very difficult. Many methods and approaches have been proposed in the literature. Face recognition is a study of how machines can recognize face, a task that humans perform naturally and effortlessly throughout their lives. Currently, powerful and low-cost computing systems (in the forms of, for example, desktop or embedded systems) are widely available. These systems are powerful enough to perform face recognition tasks and are cheap enough to make such implementation economically viable. It is only natural, then, that enormous interest in automatic face recognition of digital images and videos has risen recently.

There are numerous areas in which an automatic face recognition system can be implemented. These include biometric authentication, surveillance, human-computer interaction, and multimedia management (for example, automatic tagging of a particular individual within a collection of digital photographs). The interest in face recognition application for biometric authentication, in particular, has risen rapidly mainly due to several factors. The first one is the rising public concern for security especially in public facilities such as airports. The second factor is the rising need for verifiable identity over the Internet due to the rise in internet-based e-commerce. And finally, the use of face recognition in biometric authentication is viewed to be one of the least intrusive methods while retaining a high level of accuracy.

2. Principal Component Analysis

The Principal Component Analysis (PCA) is one of the most successful techniques that have been used in image recognition and compression. PCA is a statistical method under the broad title of factor analysis. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which is needed to describe the data economically. This is the case when there is a strong correlation between observed variables. The jobs, which PCA can do, are prediction, redundancy removal, feature extraction, data compression, etc. Because PCA is a classical technique, which can do something in the linear domain, applications having linear models are suitable, such as signal processing, image processing, system and control theory, communications, etc. The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This can be called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images (vectors).

PCA computes the basis of a space that is represented by its training vectors. These basis vectors, actually eigenvectors, computed by PCA are in the direction of the largest variance of the training vectors. As it has been said earlier, we call them eigenfaces. Each eigenface can be viewed a feature. When a particular face is projected onto the face space, its vector into the face space describes the importance of each of those features in the face. The face is expressed in the face space by its eigenface coefficients (or weights). We can handle a large input vector, facial image, only by taking its small weight vector in the face space. This means that we can reconstruct the original face with some error, since the dimensionality of the image space is much larger than that of face space.

The key procedure in PCA is based on Karhumen-Loeve transformation. If the image elements are considered to be random variables, the image may be seen as a sample of a stochastic process. The Principal Component Analysis basis

[1]N. C. College of Engineering, IsranaPanipat, Haryana-132105
[2]N. C. College of Engineering, IsranaPanipat, Haryana - 132105
E-mail: [1]India kiranjainaka@gmail.com, [2]Indiaboora_s@yahoo.com

vectors are defined as the eigenvectors of the scatter matrix S_T,

$$S_T = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \qquad \text{(Eq. 2.1)}$$

The transformation matrix is composed of the eigenvectors corresponding to the d largest eigenvalues.

3. CALCULATION OF EIGENFACES WITH PCA

The eigenface technique is a powerful yet simple solution to the face recognition dilemma. In fact, it is really the most intuitive way to classify a face. Old techniques focused on particular features of the face. The eigenface technique uses much more information by classifying faces based on general facial patterns. These patterns include, but are not limited to, the specific features of the face. By using more information, eigenface analysis is naturally more effective than feature-based face recognition.

To generate a set of eigenfaces, a large set of digitized images of human faces, taken under the same lighting conditions, are normalized to line up the eyes and mouths. They are then all resample at the same pixel resolution. Eigenfaces can be extracted out of the image data by means of a mathematical tool called principal component analysis (PCA).

Here are the steps involved in creating a set of eigenfaces:

3.1 Prepare the Data

Prepare a training set. For training set, we first need to collect a set of face images. These face images become our database

Fig.3.1: Face Space

of known faces. We will later determine whether or not an unknown face matches any of these known faces. The faces constituting the training set should be of prepared for processing, in the sense that they should all have the same resolution and that the faces should be roughly aligned. All face images must be the same size (in pixels), and for our purposes, they must be grayscale, with values ranging from 0 to 255. Each image is seen as one vector of length N (N = imagewidth*imageheight), simply by concatenating the rows of pixels in the original image. A grayscale image with r rows and c columns is therefore represented as a vector with r x c elements. In the following discussion we assume all images of the training set are stored in a single matrixΓ, where each row of the matrix is an image. The most useful face sets have multiple images per person. This sharply increases accuracy, due to the increased information available on each known individual. We will call our collection of faces "face space." This space is of dimension N.

3.2 Subtract the Mean

The average image a has to be calculated and subtracted from each original image in Γ.

$$\Psi = \frac{1}{M} \sum_{n-1}^{M} \Gamma_n \qquad \text{(Eq. 3.1)}$$

Fig. 3.2: Average Face

We then compute each face's difference from the average:

$$\Phi_i = \Gamma_i - \Psi \qquad \text{(Eq. 3.2)}$$

3.3 Calculate the Covariance Matrix

The covariance matrix or dispersion matrix is a matrix of covariance between elements of a random vector. Covariance is a measure of how much two variables change together. (Variance is a special case of the covariance when the two variables are identical.)

$$C = \frac{1}{M} \sum_{n-1}^{M} \Phi_n \Phi_n^T \qquad \text{(Eq. 3.3)}$$

3.4 Calculate the Eigenvectors and Eigenvalues of the Covariance Matrix

The eigenvectors μ and the corresponding eienvalues λ should be calculated. The eigenvectors (eigenfaces) must be normalized so that they are unit vectors, i.e. of length 1. Eigenvalue, eigenvector, and eigenspace are related concepts in the field of linear algebra. Eigenvalues, eigenvectors and eigenspaces are properties of a matrix. In general, a matrix acts on a vector by changing both its magnitude and its direction. However, a matrix may act on certain vectors by changing only their magnitude, and leaving their direction unchanged (or possibly reversing it). These vectors are the eigenvectors of the matrix. A matrix acts on an eigenvector by multiplying its magnitude by a factor, which is positive if its direction is unchanged and negative if its direction is reversed. This factor is the eigenvalue associated with that eigenvector. An eigenspace is the set of all eigenvectors that have the same eigenvalue. Each eigenvector has the same dimensionality as the original images and can itself be seen as an image. The eigenvectors of this covariance matrix are therefore called eigenfaces. They are the directions in which the images in the training set differ from the mean image.

3.5 Select the Principal Components

Choose the principal components. The C covariance matrix will result in D eigenvectors, each representing a direction in the image space. From D eigenvectors (eigenfaces) μ, only D' should be chosen, which have the highest eigenvalues. The higher the eigenvalue, the more characteristic features of a face does the particular eigenvector describe. Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of characteristic features of the faces. After D' eigenfaces are determined, the"training" phase of the algorithm is finished.

3.6 Classifying the Faces

The process of classification of a new (unknown) A to one of the classes (known faces) proceeds in two steps.

First, the new image is transformed into its eigenface components. The resulting weights form the weight vector A_w. The Euclidean distance between two weight vectors d(i, j) provides a measure of similarity between the corresponding images i and j. If the Euclidean distance between A and other faces exceed-on average-some threshold value, one can assume that A is no face at all

4. IMPLEMENTATION AND RESULTS

There are well-known face database that can be downloadable from the Internet. They contain different images of each of different distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses).

An experiment with a database that contains only 12 subject's images has been performed to ensure how well the eigenface system can identify each individual's face. The system has been implemented by MATLAB. 12 subjects are selected as test set and each subject have 2 images as training set

Fig. 4.1: Test Database

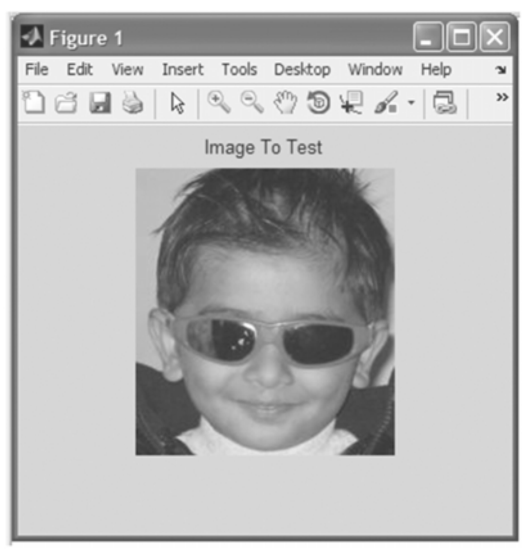

Fig. 4.2: Image to Test

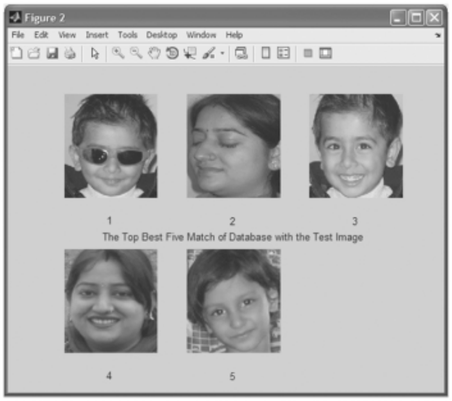

Fig. 4.3: Best Five Match of Database

5. Critical Reviews

In order to examine how well the algorithm works, I did much experiment with different sets of training images. For the accuracy of the experiments, the training set should not overlap the test set. The size of the training set was increased, i.e., 1st testing –2 subjects each of whom has 2 images total 4 images, 2nd testing –6 subject each of whom has 2 images total 12 images and last 10 subjects each of whom has 2 images total 20 images which is the maximum number of subject in the given database. The results showed me that the system gave correct identification for the known faces that is trained by the system. The noticeable fact is that the bigger the size of training set is, the less the reconstruction errors are. This is because there are more basis vectors (eigenvectors) to express the given data (faces + non-faces) in the feature space.

I would like to enumerate some advantages and limitations of the algorithm, which I found from the experiments and careful considerations.

(a) Advantages

1. Ease of implementation
2. No knowledge of geometry or specific feature of the face required
3. Little preprocessing work

(b) Limitations

1. Sensitive to head scale
2. Applicable only to front views
3. Work well only in control background.
4. The face image should be normalized.

6. Conclusion

Eigenfaces algorithm has some shortcomings due to the use of image pixel gray values. As a result system becomes sensitive to illumination changes, caling, etc. and needs a beforehand pre-processing step. Satisfactory recognition performances could be reached by successfully aligned face images. When a new face attend to the database system needs to run from the beginning, unless a universal database exists. Unlike the eigenfaces method, kernel PCA method is more robust to illumination changes.

6.1 Kernel PCA

KPCA is a development of the PCA (eigenfaces) method. PCA method is used in the feature extraction step of a face recognition system. Face image data has a very high dimensionality. Trying to perform classification on such a data would be extremely difficult. Therefore, we need to reduce the dimensionality of the data. The reduced-dimensionality data would contain only important features that we need in order to perform classification. PCA Performs very well in reducing the dimensionality of the data. However, the performance of PCA method (or other linear methods) is not completely satisfactory for problems with high nonlinearity, such as face recognition. Basically, the KPCA is used to tackle the nonlinearity of face recognition problem. By using a nonlinear kernel function, a dimensional reduction (i.e., nonlinear projection) is performed. The images are first transformed from image space into a feature space. In the feature space, the manifolds of the data become simple.

Unlike traditional PCA, KPCA can use more eigenvector projections than the input dimensionality. But a suitable kernel and correspondent parameters can only be decided empirically.

References

[1] Matthew A. Turk and Alex P. Pentland. "Eigenfaces for Recognisation". Journal of Cognitive Nerosciences, 3, Number 1, Nov 27, 2002.

[2] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pages 586-591, 1991

[3] L.I.Smith. "A Tutorial on Principal Component Analysis", Feb 2002.

[4] Hazem M. El-Bakry, "New Fast Principal Component Analysis for Face Detection".

[5] Kyungnam Kim, "Face Recognition Using Principle Component Analysis".

[6] S. Z. Li and A. K. Jain, "Handbook of Face Recognition", Springer Science + Business Media, Inc., 2005.

[7] Linlin Shen and Dr. Li Bai "A Review on Gabor Wavelets for Face Recognition" Aug 18, 2006

[8] "Face Recognition Using Kernel Methods by Ming Hsuan Yang".

[9] Stefano Arca, Paola Campadelli "A Face Recognition System Based on Automatically Determined Facial Fiducial Points" March 15, 2005

[10] Y. Zhang and A.M. Martinez. "Recognizing Imprecisely Localized Partially Occluded and Expression Variant Faces from a Single Sample Per Class", Proceedings of International Conference on Pattern Recognition (ICPR), 2004.