Face Recognition using Matrix Decomposition Technique Eigenvectors and SVD

A.N.M. Rezaul Karim*, Thwarique
Dept. of Computer Science & Engineering
International Islamic University Chittagong, Bangladesh
zakianaser@yahoo.com, trq_imran@yahoo.com
doi: 10.4156/ijact.vol2.issue1.5

Abstract

Principle Component Analysis (PCA) is an important and well-known technique of face recognition, where eigenvectors are used. In this paper, we propose a face recognition technique, which combines Eigenvectors with Singular Value Decomposition (SVD) techniques to reduce size of the Eigen-matrix. The detailed theoretical derivation and analysis are presented and a simulation results on Olivetti Research Laboratory (ORL) face database has given. The simulation result indicates that the proposed approach is superior to conventional PCA with lower database size and recognition performance.

Keywords: *Eigenface, PCA, SVD, image processing, pattern recognition, face recognition.*

1. Introduction

Face recognition is an unsolved problem under the conditions of pose, illumination, and database size. Face recognition has many real world applications like human/computer interface, surveillance, authentication and perceptual user interfaces. Principal Component Analysis is a typical and successful face based technique. Turk and Pentland developed a face recognition system using PCA in 1991 [1] [2]. Belhumeur et. al proposed Fisherface technique based on Linear Discriminant Analysis (LDA) in 1997 [3]. Recently Facial feature extraction [4] [5] has become an important topic in automatic recognition of human faces. Extracting the basic features like eyes, nose and mouth exactly is necessary for most of the feature-based approaches [6]. Other Face Recognition approaches based on Discriminant Analysis and Feature Extraction also exists but the achievements in the field of automatic face recognition by computer is not as satisfactory as in other area like fingerprints recognition. Despite of the good results of PCA, it is computationally expensive and complex with the increase in database size, since all pixels in the image are necessary to obtain the representation used to match the input image with all others in the database. The main objective of our research is to combine PCA with SVD technique to enhance performance of face recognition by compressing database size for better results.

2. Literature Review

Face recognition is one of biometric methods identifying individuals by the features of face. Before face recognition is performed, the system should determine whether or not there is a face in a given image or given video or a sequence of images. This process is called face detection. Once a face is detected, face region should be isolated from the scene for the face recognition. The face detection and face extraction are often performed simultaneously. The overall process of face recognition is depicted in Fig. 1.

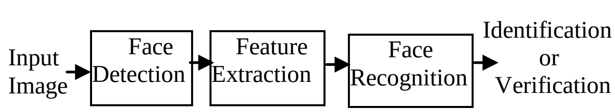

Figure 1. Process of face recognition

2.1. Existing Face Recognition Approaches

There are three categories of recognition methods: holistic matching, feature-based and hybrid methods.

2.1.1. Holistic Approaches

The first category of face recognition techniques is the holistic matching methods that use the whole face region as input to the face recognition system. The Eigenface approach by Turk and Pentland [4] is a representative of holistic analysis on faces, where it directly applies the PCA method developed by Kirby and Sirovich. Other representations of this approach include Linear Discriminant Analysis (LDA) and Independent Component Analysis (ICA).

Principal Component Analysis (PCA)/ Eigen space Projection: Principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. Depending on the field of application, it is also named the discrete **Karhunen–Loève transform (KLT)**, the **Hotelling transform** or **proper orthogonal decomposition (POD)**. PCA involves the calculation of the eigen value decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centering the data for each attribute. PCA is the simplest of the true eigenvector-based multivariate analyses. It is a statistical technique that has been widely used for face recognition and a common technique for finding patterns in images. In fact, it has been one of the major driving forces behind face representation, detection and recognition [7].

The applicability of PCA is limited by the assumptions[5] made in its derivation. These assumptions are:

- Assumption on linearity
- Assumption on the statistical importance of mean and covariance
- Assumption that large variances have important dynamics

2.1.2. Feature-based (Structural) Approaches

The second type is feature-based or structural matching methods, which include classical methods such as the Pure Geometry Methods proposed by Kelly. The first step in this approach is to extract local facial features such as the eyes, nose and mouth. Then, their locations and local (geometric) statistics are fed into a structural classifier. Other examples of this approach are Dynamic Link Architecture (graph matching methods) and Hidden Markov Models.

2.1.3. Hybrid Approaches

The third approach is the hybrid models that use both local features and the whole face region. These methods are said to have best of both worlds and could potentially give a more robust performance as compared to the other two approaches applied individually. Modular Eigenfaces, Hybrid Local Feature Analysis (LFA) and Component-based 3D Models are examples of hybrid models.

2.2. Properties of the SVD

There are many properties and attributes of SVD; here we just present parts of the properties that we used in this paper.

1. The singular value $\sigma_{1,}\ \sigma_{2\ldots}\ \sigma_n$ are unique, however, the matrices U and V are not unique.

2. Since $A^TA = VS^TSV^T$, so V diagonalizes A^TA, it follows that the $v_j s$ are the eigenvectors of A^TA.
3. Since $AA^T = US^TSU^T$, so it follows that U diagonalizes A^TA and that $u_j s$ are the eigenvectors of $A.A^T$.
4. The rank of matrix A is equal to the number of its nonzero singular values

2.2.1. SVD Approach for Image Compression

SVD has many practical and theoretical values; special feature of SVD is that it can be performed on any real (m, n) matrix. Let's say we have a matrix A with m rows and n columns, with rank r and $r \le n \le m$. Then the A can be factorized into three matrices:

$$A = USV^T$$

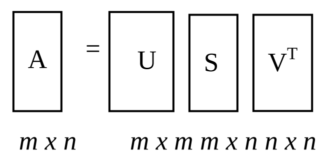

Figure 2. Illustration of Factoring A to USV^T

Image compression deals with the problem of reducing the amount of data required to represent a digital image. Compression is achieved by the removal of three basic data redundancies: [8].

1. Coding redundancy, which is present when less than optimal
2. Inter pixel redundancy, which results from correlations between the pixels
3. Psycho visual redundancies, which is due to data that is ignored by the human visual.

The property 4 of SVD in section (B) tells us "the rank of matrix A is equal to the number of its nonzero singular values". In many applications, the singular values of a matrix decrease quickly with increasing rank. This property allows us to reduce the noise or compress the matrix data by eliminating the small singular values or the higher ranks.

When an image is SVD transformed, it is not compressed, but the data take a form in which the first singular value has a great amount of the image information. With this, we can use only a few singular values to represent the image with little differences from the original.

To illustrate the SVD image compression process, we show detail procedures

$$\mathbf{A = USV^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T} \quad (1)$$

That is A can be represented by the outer product expansion:

$$\mathbf{A = \sigma_1 u_1 v_1{}^T + \sigma_2 u_2 v_2{}^T + \ldots\ldots + \sigma_r u_r v_r{}^T} \quad (2)$$

When compressing the image, the sum is not performed to the very last SVs, the SVs with small enough values are dropped.

The closet matrix of rank k is obtained by truncating those sums after the first k terms:

$$\mathbf{A = \sigma_1 u_1 v_1{}^T + \sigma_2 u_2 v_2{}^T + \ldots\ldots\ldots + \sigma_k u_k v_k{}^T} \quad (3)$$

The total storage for A_k will be $k (m + n + 1)$

The integer k can be chosen confidently less than n, and the digital image corresponding to A_k still have very close to the original image. However, the chose the different k will have a different corresponding image and storage for it. For typical choices of the k, the storage required for A_k will be less than 20 percentages.

3. Proposed Method of Face Recognition System

The task of facial recognition is discriminating input signals (image data) into several classes (persons). The input signals are highly noisy (e.g. the noise is caused by differing lighting conditions, pose etc.), yet the input images are not completely random and in spite of their differences there are patterns. Such patterns, which can be observed in all signals, could be in the domain of facial recognition - the presence of some objects (eyes, nose, and mouth) in any face as well as relative distances between these objects. These characteristic features are called eigenvectors (eigenfaces) in the facial recognition domain (or principal components, singular value decomposition generally). They can be extracted out of original image data by means of mathematical tools called Principal Component Analysis and Singular Value Decomposition (PCA and SVD). The steps/algorithm of proposed face recognition system is shown below.

Steps of Proposed face Recognition System shown below:

3.1. Training

1. Take input of different face images to create training database from ORL.
2. Create or form an image matrix (Γ_i) using the input face image.
3. Determine the average image matrix (ψ).
4. Determine the difference image matrix (ϕ_i).
5. Calculate covariance matrix (**C**).
6. Calculate Eigenvector (u_i) and Eigen values (λ_i) and determine L
7. Select the principal components

3.2. Testing

1. Image/Face classification.
2. Face recognition.

3.2.1.Take Input of Different Face Images

In this step, the number of faces constituting the training set is got as input from database of ORL (Olivetti Research Laboratory). Each image is got as a matrix.

3.2.2. Form an Image Matrix

The image matrix is converted into a single column matrix. All such images are concatenated to form a single matrix (Γ_i).

3.2.3. Determine the Average Matrix

The average matrix (Ψ) has to be calculated, as such the following:

$$\psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \tag{4}$$

3.2.4. Determine the Difference Image Matrix

The difference matrix (ϕ_i) is calculated by subtracting the average matrix (Ψ) from the original faces matrix (Γ_i) and the result stored in the variable ϕ_i

$$\phi_i = \Gamma_i - \psi \quad \text{Where} \quad i = 1 \text{ to } M \tag{5}$$

3.2.5. Calculate the covariance matrix

In the next steps the covariance matrix C is calculated according to

$$C = \frac{1}{M}\sum_{n=1}^{M}\phi_n\phi_n^T \tag{6}$$

Where ϕ_n^T denotes the normalized matrix transposition of the face image. Now, from this covariance matrix, we are interested in finding a set of orthogonal Vectors (the principal components) that best describes the distribution of the data in a least squares sense i.e the Euclidean projection is minimized Mathematically, these principal components are the eigenvectors of the covariance matrix. To find the eigenvectors u_i, we can solve

$$Cu_i = \lambda_i u_i$$

$$u_i^T C u_i = u_i^T \lambda_i u_i$$

$$u_i^T C u_i = \lambda_i u_i^T u_i$$

Where λ_i are the corresponding eigen values. The eigenvectors are orthogonal and normalized, hence

$$u_i^T u_j = 1;\ \text{if}\ \ i = j$$

$$= 0;\ \text{if}\ i \neq j$$

An alternative way of representing the covariance matrix is by taking each of image vectors ϕ_i and placing them in each column of matrix A.

$$A = [\phi_1, \phi_2, \phi_3, \ldots\ldots\ldots\ldots\phi_M]$$

$$\mathbf{C = AA^T} \tag{7}$$

Turk and Pentland then circumnavigate the problem by proposing the following scheme. Consider the eigenvectors v_i of $A^T A$ such that

$$A^T A v_i = \mu_i v_i$$

The scalars μ_i correspond to the eigenvalues of v_i. Multiplying both sides by A,

We obtain

$$AA^T A v_i = \mu_i A v_i$$

$$CAv_i = \mu_i A v_i,$$

From which we can see that Av_i are the eigenvectors of the covariance matrix C, with μ_i being its corresponding eigen values. With this treatment, the dimension of the matrix on which we have to compute is now M by M instead of N by N. This implies that the eigenvector analysis is reduced from the order of the number of pixels in the images (N2) to the order of the number of images in the training set (M).

Following this method, we first construct the matrix $L = A^T A$ of M by M dimension, where $L_{ij} = \phi_i^T \phi_j$, where $1 \le i, j \le M$ The first M eigenvectors of the covariance matrix C can be obtained by finding Av_i. Once the eigenvectors v_i are found, they are sorted according to their corresponding eigen values so that the first eigenvector has the largest variance [5].

3.2.6. Calculate the eigenvectors and eigenvalues of the covariance matrix:

In this step, the eigenvectors (eigenfaces) u_i and the corresponding eigenvalues λ_i should be calculated. The eigenvectors (eigenfaces) must be normalized so that they are unit vectors, i.e., of length one. The description of the exact algorithm for determination of eigenvectors and eigenvalues is omitted here, as it is available in most of the math programming libraries.

According to PCA, since we have only M images, we have only M non-trivial eigenvectors. We can solve for these eigenvectors by taking the eigenvectors of a new M x M matrix.

So to solve this we can first compute the matrix L.

$$L=A^TA \tag{8}$$

At equation (8), if we apply singular value decomposition process on the data set matrix *A* the equation will be stood as follows:

$L = A^T A$

We have, $A = USV^T$ (from equation 1)

So, $L = (USV^T)^T USV^T$

$= VSU^T USV^T$

$= VSISV^T$ (U is orthogonal, I is Identity Matrix)

$$= VS^2V^T \tag{9}$$

Since the covariance matrix is Symmetric, the N eigenvalues obtained from C are same as M eigenvalues with remaining N-M eigenvalues equals zero.

As stated earlier in section (4) the memory required to store a matrix can be reduced by using Singular Value Decomposition method. By reducing the dimension of the S we can test the performance of this method. This method is preferable when we consider the limitation of the memory.

3.2.7. Select the principal components

From M eigenvectors (eigenfaces) u_i, only M_i should be chosen which have as the highest eigen values. The higher the eigen value, 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 M_i eigenfaces u_i are determined, the "training" phase of the algorithm is finished.

3.2.8. Classification the faces

The process of classification of a new (unknown) face Γ_{new} to one of the classes (known faces) proceeds in the following step. The new image is transformed into its eigenface components. The resulting weights form the weight vector $\Omega_{\Gamma new}$

$$\omega_k = u(\Gamma_{new_k} - \psi), \quad \mathbf{k = 1, 2, \ldots, M'} \tag{10}$$

$$\mathbf{\Omega^T}_{\mathbf{new}} \boldsymbol{\omega} = [\boldsymbol{\omega_1}, \boldsymbol{\omega_2}, \boldsymbol{\omega_3}, \ldots \boldsymbol{\omega_{M'}}]' \tag{11}$$

3.2.9. Face Recognition

The Euclidean distance between two weight vectors $d(\Omega_i, \Omega_j)$ provides a measure of similarity between the corresponding images i and j. If the Euclidean distance between Γ_{new} and other faces exceeds – on average – some threshold value θ, one can assume that Γ_{new} is no face at all, $d(\Omega_i, \Omega_j)$ also allows one to construct "clusters" of faces such that similar faces are assigned to one cluster.

4. Experiments and Results

This chapter provides the results of the experiments of the face recognition technique using eigenvectors and SVD. The proposed method describes in this paper is implemented using MATLAB v7.1.0.246 (R14) Service Pack 3 of Mathworks Incorporation. MATLAB was chosen due to its vast collection of computational algorithms and mathematical functions like matrix inverse and matrix eigenvalues as it allows easy matrix manipulation. Many built in functions are for image processing. Experiment of this system is conducted with the subset of well-known database (formerly) known as ORL face database). We have taken forty different images saved in .pgm format of four subjects with a standard size of 112 x 92 pixels as our training database set. The experiments are carried out many times to ensure the perfect results. As we don't concern with the eigenfaces, which are the feature vectors of the covariance matrix; we fix the number of eigenfaces and only conduct the experiments the number of eigenfaces and only conduct the experiments with the different number of k terms of SVs which composed of the covariance matrix, the main matrix to form the eigenfaces.

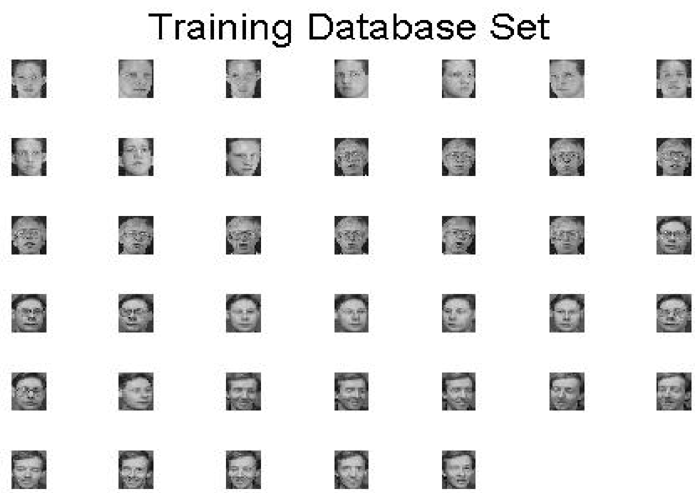

Figure 4. Images from Training Set

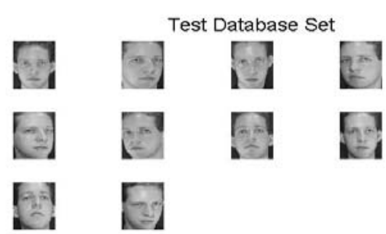

Figure 5. Images from Test Set

We use a test database set, containing ten images of different faces to conduct the experiment. The test process is going by selecting an image from the test database set and matching this image from the images of the training database set. We conduct the experiment taking the images from the test database set one by one, several times to ensure the error free result as much as possible. The usage of SVs (Singular Values) less than forty is obviously smaller in dimension than ordinary PCA process. We test the system reducing the SVs and record the recognition rate.

Table1. Recognition rate with different number of SVs

Number of SVs used (k)	Dimension (Image set) (Eigenface)	Recognition Rate (Percentage)
20	10304x20	40
30	10304x30	70
35	10304x35	100

Full	10304x39	100

Table 2. Memory Storage with different number of SVs

Number of SVs used (k)	Required Memory of proposed Technique in KB	Required memory of conventional PCA in KB
20	6.30	8.31
30	7.55	
35	7.99	
Full	8.31	

We could surely claim from the experiments that though the recognition rate falls as we truncate some SVs, the storage space of the matrix is reduced. We hope that by conducting further experiments we would able to implement the system to memory limited environment efficiently.

5. Future Plan

In future we will use Neural Network for training up instead of Euclidian distance method.

6. Conclusion

In this paper, an efficient face recognition approach called face recognition using eigenvectors and SVD. From the experimental result, it is apparent that our effort of combination of PCA with SVD, significantly improve performance of the traditional PCA method. Though it has a little complexity, the size is significantly reduced. So far we have tested this approach on a relatively medium sized and clean database only. In future work, the performance of the face recognition using eigenvectors and SVD will be further examined and we intend to gather more data in order to obtain better generalization ability, to include expression and lighting changes.

7. References

[1] A. J. Goldstein, L.D. Harmon, and A.B. Lesk, “Identification of Human Faces,” Proc. IEEE, May 1972, Vol. 59, No. 5, 748-760.
[2] L. Sirovich and M. Kirby, “A Low-Dimensional Procedure for the Characterization of Human Faces, “J. Optical Soc. Am. A,1987, Vol. 4, No.3, 519-524.
[3] M. A. Turk and A. P. Pentland, “Face Recognition Using Eigenfaces,” Proc. IEEE, 1991, 586-591.
[4] Turk, M., and Pentland, A. Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, 1 (1991), 71–86.
[5] Xiao Yee Nikki Tan ,The Eigenface Approach to Automated Face Recognition , School of Computer Science and Software Engineering, The University of Western Autralia, 2005.
[6] James Reilly, Matrix computations for signal processing. Dept. of ECE, McMaster University, 2003.
[7] Zhao, W., Chellapa, R., and Krishnaswamy, A. Discriminant analysis of principal components for face recognition. Proceedings of International Conference on Automatic Face and Gesture Recognition (1998), 336–341.
[8] Lijie Cao Singular Value Decomposition Applied to Digital Image Processing , Division of Computing Studies, Arizona State University Polytechnic Campus, Mesa, Arizona 85215