Volume 2, Issue 3 (2014) 667-675 ISSN 2347 - 3258
International Journal of Advance Research and Innovation

An Improved Face Recognition Approach using Principal Component Analysis

Deepak Gaur *, Raj Kumar Sagar
Department of Computer Science & Application, ASET, Amity University, Noida, Uttar Pradesh, India

Article Info

Article history:
Received 20 July 2014
Received in revised form
30 July 2014
Accepted 20 August 2014
Available online 15 September 2014

Keywords

Affect Computing,
Face recognition,
Principal Component Analysis,
Open CV,
Cohn-Kanade DataSet

Abstract

Due to digitization and for security purpose a lot of research has been going on in the wide area of affect computing. One of the field under this affect computing is to recognize the human faces with maximum accuracy. There are still large numbers of difficulties to recognize the accurate facial expression. In this research paper we are going to represent our experimental results for facial recognition by using Principal Component Analysis (PCA) algorithm. So in the first section of this paper we discussed some algorithm for facial recognition, than compare our results of research with these algorithm. We took Extended Cohn-Kanade Dataset(CK+) for experimental results. Our experimental results are implemented in OpenCV.

1. Introduction

Face recognition is becoming an active research area spanning several disciplines such as image processing, pattern recognition, computer vision, neural networks, cognitive science, neuroscience, psychology and physiology. It is a dedicated process, not merely an application of the general object recognition process. It is also the representation of the most splendid capacities of human vision.

Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars. Early face-detection algorithms focused on the detection of frontal human faces, whereas newer algorithms attempt to solve the more general and difficult problem of multi-view face detection. That is, the detection of faces that are either rotated along the axis from the face to the observer (in-plane rotation), or rotated along the vertical or left-right axis (out-of-plane rotation), or both. The newer algorithms take into account variations in the image or video by factors such as face appearance, lighting, and pose. In this paper we provide a accurate method to recognize facial expression by using PCA algorithm.

2. Literature Survey

Face recognition is a biometric which uses computer software to determine the identity of the individual. Face recognition falls into the category of biometrics which is ―the automatic recognition of a person using distinguishing traits‖ [6]. Other types of biometrics include fingerprinting, retina scans, and iris scan.

2.1 Eigenface-Based Recognition

2D face recognition using eigenfaces is one of the oldest types of face recognition. Turk and Pentland published the groundbreaking ―Face Recognition Using Eigenfaces‖ in 1991.

Corresponding Author,
E-mail address: India.dgaur@amity.edu

The method works by analyzing face images and computing eigenfaces which are faces composed of eigenvectors. The comparison of eigenfaces is used to identify the presence of a face and its identity.

There is a five step process involved with the system developed by Turk and Pentland. First, the system needs to be initialized by feeding it a set of training images of faces. This is used these to define the face space which is set of images that are face like. Next, when a face is encountered it calculates an eigenface for it. By comparing it with known faces and using some statistical analysis it can be determined whether the image presented is a face at all. Then, if an image is determined to be a face the system will determine whether it knows the identity of it or not. The optional final step is that if an unknown face is seen repeatedly, the system can learn to recognize it [1].

The eigenface technique is simple, efficient, and yields generally good results in controlled circumstances [1]. The system was even tested to track faces on film. There are also some limitations of eigenfaces. There is limited robustness to changes in lighting, angle, and distance [6]. 2D recognition systems do not capture the actual size of the face, which is a fundamental problem [4]. These limits affect the technique's application with security cameras because frontal shots and consistent lighting cannot be relied upon.

2.2 3D Face Recognition

3D face recognition is expected to be robust to the types of issues that plague 2D systems [4]. 3D systems generate 3D models of faces and compare them. These systems are more accurate because they capture the actual shape of faces. Skin texture analysis can be used in conjunction with face recognition to improve accuracy by 20 to 25 percent [3]. The acquisition of 3D data is one of the main problems for 3D systems [2].

2.3 How Human Perform Face Recognition

It is important for researchers to know the results of studies on human face recognition [8]. Knowing these

results may help them develop ground breaking new methods. After all, rivaling and surpassing the ability of humans is the key goal of computer face recognition research. The key results of a 2006 paper ―Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About‖ are as follows:

1. Humans can recognize familiar faces in very low-resolution images.
2. The ability to tolerate degradations increases with familiarity.
3. High-frequency information by itself is insufficient for good face recognition performance.
4. Facial features are processed holistically.
5. Of the different facial features, eyebrows are among the most important for recognition.
6. The important configurable relationships appear to be independent across the width and height dimensions.
7. Face-shape appears to be encoded in a slightly caricatured manner.
8. Prolonged face viewing can lead to high level aftereffects, which suggest prototype-based encoding.
9. Pigmentation cues are at least as important as shape cues.
10. Color cues play a significant role, especially when shape cues are degraded. Contrast polarity inversion dramatically impairs recognition performance, possibly due to compromised ability to use pigmentation cues.
11. Motion of faces appears to facilitate subsequent recognition.
12. The visual system starts with a rudimentary preference for face-like patterns.
13. Motion of faces appears to facilitate subsequent recognition.
14. The visual system starts with a rudimentary preference for face-like patterns.
15. Motion of faces appears to facilitate subsequent recognition.
16. The visual system starts with a rudimentary preference for face-like patterns.

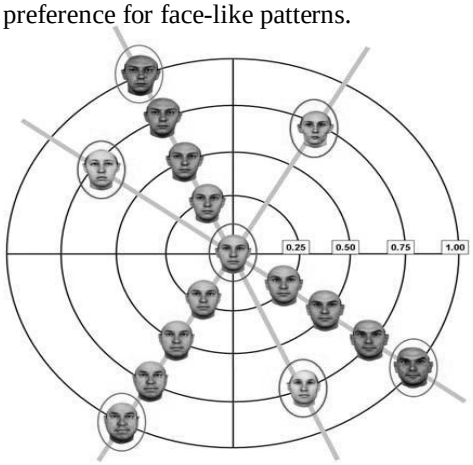

Fig: 1. Staring at the faces in the green circles will cause one to misidentify the central face with the faces circled in red. This is an example of face aftereffects [8]

17. The visual system progresses from a piecemeal to a holistic strategy over the first several years of life.
18. The human visual system appears to devote specialized neural resources for face perception.
19. Latency of responses to faces in infer temporal (IT) cortex is about 120 ms, suggesting a largely feed forward computation.
20. Facial identity and expression might be processed by separate systems.

Fig: 2. Photograph during the recording of ―We Are the World. This figure demonstrates how polarity inversion effects face recognition in humans. Several famous artists are in the picture including Ray Charles, Lionel Ritchie, Stevie Wonder, Michael Jackson, Tina Turner, Bruce Springstein, and Billy Joel though they are very difficult to identify

2.4 Face Recognition From A Law Enforcement Perspective

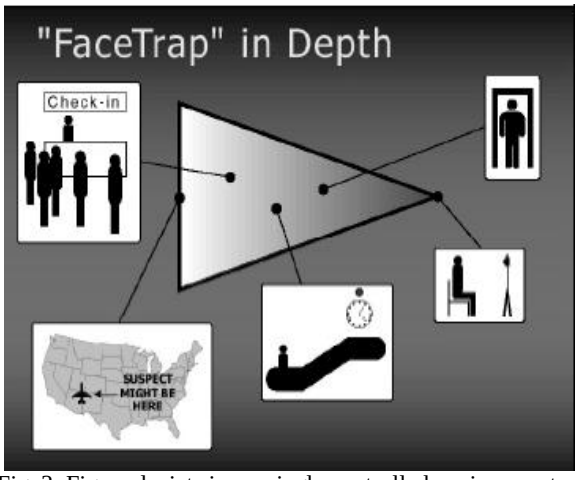

Fig: 3. Figure depicts increasingly controlled environments from left to right. From left to right: suspect on a plane (no control), subject at a check-in counter, subject on an escalator staring at a flashing red bulb, subject passing through a doorway, subject sitting in front of a camera (perfect control) [6]

Facial recognition is attractive for law enforcement. It can be used in conjunction with existing surveillance camera infrastructure to hunt for know criminals. Face recognition is covert and non intrusive, opposed to other biometrics such as finger prints, retina scans, and iris scans [6]. This is especially important in conjunction with the law because faces are considered public. Comprehensive photo databases from mug shots or driver's licenses already exist.

Because of difficulties face recognition has with respect to lighting, angle, and other factors, it is advantageous to attempt to get as high quality images with regard to these factors. Facetraps are a concept where cameras are strategically placed in order to obtain relatively controlled photographs [6]. Examples are placing cameras facing doorways, at airport check-ins, or near objects people are likely to stare at these traps would aid face recognition software by helping to capture a straight frontal image which allow for higher accuracy of the system[4]. Despite their potential benefit, there appears to be very little research done on facetraps.

Some have questioned the legality of face scanning and have argued that such systems which are used to hunt to criminals in public places are an invasion of privacy. From a legal perspective, in the United States, one does not have a right to privacy for things shown in public [6]. ―What a person knowingly exposes to the public is not a subject of Fourth

Amendment protection,‖ United States v. Miller, 425 U.S. 435 (1976). ―No person can have a reasonable expectation that others will not know the sound of his voice, any more than he can reasonably expect that his face will be a mystery to the world,‖ United States v. Dionisio, 410 U.S. 1 (1973). These excerpts from Supreme Court decisions help to establish that face recognition is constitutional.

Face recognition must be improved further before it becomes a useful tool for law enforcement. It remains to be seen what the right balance is, socially speaking, between maximizing public safety and respecting individual rights.

2.5 Algorithm Analysis for Face Recognition

There are large numbers of algorithm for face recognition. Some of these algorithms are discussed as below:

2.5.1 2 D HMM Algorithm

2D Hidden Markov Model algorithm uses following two assumptions for face recognition [15]:-

Assumption 1 *The transition probability of state s*(*i, j*) *in the model depends on its adjacent neighboring states in vertical, horizontal and diagonal directions.*

Suppose there are *M* states *{*1, 2, ...,*M}*, and for each block (*i, j*), *i* = *{*1, 2, ..., *I}*; *j* = *{*1, 2, ..., *J}*, where *I* and *J* are the numbers of row and column blocks in the original image, the feature vector is *o*(*i, j*), the corresponding hidden state is *s*(*i, j*), and the class of the block is *c*(*i, j*). We define the transition probability of state *s*(*i, j*) and it depends on its adjacent neighboring states in vertical, horizontal and diagonal directions. and is stated as follows:

$$P\{s(i,j) = l | s(i-1,j) = m,\ s(i-1,j-1) = n,\ s(i,j-1) = k\} = a_{m,n,k,l}. \quad (1)$$

where *m, n, k, l* ∈ *{*1, 2, ...,*M}* are actual values of the state.

Assumption 2 *The feature vector for each image block follows a Gaussian Mixture distribution, given its corresponding state, and it is independent of other feature vectors and their corresponding states.*

However these algorithms have following strengths:-

1. The HMMs can be used for generating alignments, with each state of the machine corresponding to one column in the alignment.. HMMs are a bit more powerful than alignments, since the same state can be used repeatedly in a path, but each column can only be used once in an alignment. This results in ambiguous alignments if a column alignment model is used, but can be quite convenient for describing phenomena like random numbers of repeats of a short subsequence.
2. Separate HMMs built for recognizing particular structures can be merged to create HMMs that recognize sequences of structures. Unfortunately doing this cleanly requires a slightly different version of HMMs which allows *null states*--states that don't match any characters in the input sequence. The current version of my HMM code cannot handle HMMs with null states, but the extension is planned and should be straightforward.

Limitation of HMM algorithm:-

1. The HMM needs to be trained on a set of seed sequences and generally requires a larger seed than the simple Markov models.
2. For a given set of seed sequences, there are many possible HMMs, and choosing one can be difficult. Smaller models are easier to understand, but larger models can fit the data better
3. HMM is quite expensive to implement in term of memory and compute time.

2.5.2 Minimum Average Correlation Energy Filter (MACE) Algorithm

Suppose we have *N* facial images from a certain person. We consider each 2-dimensional image as a *d*×1 column vector **x***i* (*i* = 1, 2. . . *N*) by lexicographically reordering the image, where *d* is the number of pixels. The discrete Fourier transform (DFT) of **x***i* is denoted by **X***i*, and we define the training image data matrix in frequency domain as *X* = [**X1 X2** . . . **XN**]. *X* is a *d* × *N* matrix [13].

Let the vector **h** be the correlation filter (correlation template) in the space domain and **H** be its Fourier transform. The correlation result of the *i*th image and the filter could be written as

$$ci(m,n) = \mathbf{h}(m,n)\ o\ \mathbf{x}i(m,n) = (\mathbf{h}, \mathbf{x}_{im,n}) \quad \ldots\ldots\ldots(2)$$

Where o denotes correlation, and (,) denotes inner *product* of two vectors. Here **x***m,n i* is a vector obtained by circularly shifting the *i*th training image by *m* pixels horizontally and *n* pixels vertically, and reorder it to a 1-dimensional vector. Keep in mind that **h** and **x***i* are both 1-dimensional vectors obtained by reordering a 2-dimensional array. Since the correlation actually operates on a 2-dimensional plane, here we use two indices, *m* and *n*, to indicate the elements in these vectors.

The objective of the MACE filter is to minimize the average correlation energy over the image class while simultaneously satisfying an linear constraint that the correlation values at the origin due to training images take

on pre-specified values stored in vector **u**, where **u** = [*u*1 *u*2 · · · *uN*]*T* .

i.e. *ci*(0, 0) = **X***i*†**H** = *ui* (3)

The average correlation energy over all training images is *Eavg* = **H**†*D***H** where *D* is summation of all i values.

Minimize *Eavg* subject to the constraint, *X*†**H** = **u**. The solution can be obtained using Lagrange multipliers:

H = *D*−1*X*(*X*†*D*−1*X*)−1**u** (4)

2.5.3 Linear Discriminant Analysis Algorithm

In Linear discriminant analysis following steps are used to discriminant the input images:

1. First of all we need a training set composed of a relatively large group of subjects with diverse facial characteristics. The appropriate selection of the training set directly determines the validity of the final results. The database should contain several examples of face images for each subject in the training set and at least one example in the test set. These examples should represent different frontal views of subjects with minor variations in view angle. They should also include different facial expressions, different lighting and background conditions, and examples with and without glasses. It is assumed that all images are already normalized to m × n arrays and that they contain only the face regions and not much of the subjects' bodies [16].
2. For each image and sub image, starting with the two dimensional m × n array of intensity values I(x, y), we construct the vector expansion Φ_ R m× n. This vector corresponds to the initial representation of the face. Thus the set of all faces in the feature space is treated as a high-dimensional vector space.
3. By defining all instances of the same person's face as being in one class and the faces of different subjects as being in different classes for all subjects in the training set, we establish a framework for performing a cluster separation analysis in the feature space. Also, having labeled all instances in the training set and having defined all the classes, we compute the within-class and between-class scatter matrices. So finally we synthesized the face recognition.

3. Proposed System

3.1 Principal Componant Analysis Algorithm

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components. The principal components are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric. PCA is sensitive to the relative scaling of the original variables. In the case of Face Recognition system based on PCA, it seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic manner. When all the face images are converted into vectors, they will group at a certain location in the image space as they have similar structure, having eye, nose and mouth in common and their relative position correlated. This correlation is the main point to start the Eigen-face analysis.

3.2 Detail Description of PCA

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some 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[8].

Consider a data matrix, X, with column-wise zero empirical means (the sample mean of each column has been shifted to zero), where each of the n rows represents a different repetition of the experiment, and each of the p columns gives a particular kind of datum (say, the results from a particular sensor).

Mathematically, the transformation is defined by a set of p-dimensional vectors of weights or loadings $\mathbf{w}_{(k)} = (w_1, \ldots, w_p)_{(k)}$ that map each row vector $\mathbf{x}_{(i)}$ of X to a new vector of principal component scores $\mathbf{t}_{(i)} = (t_1, \ldots, t_p)_{(i)}$, given by

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)}$$

In such a way that the individual variables of t considered over the data set successively inherit the maximum possible variance from x, with each loading vector w constrained to be a unit vector. Eventually maximum possible variance from X has been formed from the desired data set. According we have to calculate first component of PCA, covariance of PCA has to satisfy some equation to find the final solution of any recognition from data set.

3.2.1 First Component of PCA

The first loading vector w(1) thus has to satisfy

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i (t_1)^2_{(i)} \right\} = \arg\max_{\|\mathbf{w}\|=1} \sum_i \left(\mathbf{x}_{(i)} \cdot \mathbf{w} \right)^2 \quad (5)$$

Equivalently, writing this in matrix form gives

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \|\mathbf{X}\mathbf{w}\|^2 \right\} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \right\} \quad (6)$$

Since w(1) has been defined to be a unit vector, it equivalently also satisfies

$$\mathbf{w}_{(1)} = \arg\max \left\{ \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\} \quad (7)$$

The quantity to be maximised can be recognised as a Rayleigh quotient. A standard result for a symmetric matrix such as XTX is that the quotient's maximum possible value is the largest eigenvalues of the matrix, which occurs when w is the corresponding eigenvector.

3.2.2 K[th] Componant of PCA

The kth component can be found by subtracting the

first k − 1 principal components from X:

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^T \quad (8)$$

and then finding the loading vector which extracts the maximum variance from this new data matrix. It turns out that this gives the remaining eigenvectors of XTX, with the maximum values for the quantity in brackets given by their corresponding eigenvalues.

The full principal components decomposition of X can therefore be given as

$$\mathbf{T} = \mathbf{X}\mathbf{W} \quad (9)$$

Where W is a p-by-p matrix whose columns are the eigenvectors of XTX

3.2.3 Covariances of PCA

The sample covariance Q between two of the different principal components over the dataset is given by:

$$\begin{aligned} Q(\mathrm{PC}_{(j)}, \mathrm{PC}_{(k)}) &\propto (\mathbf{X}\mathbf{w}_{(j)}) \cdot (\mathbf{X}\mathbf{w}_{(k)}) \\ &= \mathbf{w}_{(j)}^T \mathbf{X}^T \mathbf{X} \mathbf{w}_{(k)} \\ &= \mathbf{w}_{(j)}^T \lambda_{(k)} \mathbf{w}_{(k)} \\ &= \lambda_{(k)} \mathbf{w}_{(j)}^T \mathbf{w}_{(k)} \end{aligned} \quad (10)$$

Where the eigenvalue property of w(k) has been used to move from line 2 to line 3. However eigenvectors w(j) and w(k) corresponding to eigenvalues of a symmetric matrix are orthogonal (if the eigenvalues are different), or can be orthogonalised (if the vectors happen to share an equal repeated value). The product in the final line is therefore zero; there is no sample covariance between different principal components over the dataset.

Another way to characterise the principal components transformation is therefore as the transformation to coordinates which diagonalise the empirical sample covariance matrix.

In matrix form, the empirical covariance matrix for the original variables can be written

$$\mathbf{Q} \propto \mathbf{X}^T \mathbf{X} = \mathbf{W} \mathbf{\Lambda} \mathbf{W}^T \quad (11)$$

The empirical covariance matrix between the principal components becomes

$$\mathbf{W}^T \mathbf{Q} \mathbf{W} \propto \mathbf{W}^T \mathbf{W} \mathbf{\Lambda} \mathbf{W}^T \mathbf{W} = \mathbf{\Lambda} \quad (12)$$

Where Λ is the diagonal matrix of eigenvalues λ(k) of XTX

λ(k) being equal to the sum of the squares over the dataset associated with each component k: λ(k) = Σi tk2(i) = Σi (x(i).

3.2.4 Dimensionally Reduction

The faithful transformation T = X W maps a data vector x(i) from an original space of p variables to a new space of p variables which are uncorrelated over the dataset. However, not all the principal components need to be kept. Keeping only the first L principal components, produced by using only the first L loading vectors, gives the truncated transformation

$$\mathbf{T}_L = \mathbf{X}\mathbf{W}_L \quad (13)$$

Where the matrix TL now has n rows but only L columns. In other words, PCA learns a linear transformation $t = W^T x, x \in R^p, t \in R^L$, where the columns of $p \times L$ matrix W form an orthogonal basis for the L features (the components of representation t) that are decorrelated. By construction, of all the transformed data matrices with only L columns, this score matrix maximises the variance in the original data that has been preserved, while minimising the total squared reconstruction error

$$\|\mathbf{T}\mathbf{W}^T - \mathbf{T}_L\mathbf{W}_L^T\|_2^2 \text{ or } \|\mathbf{X} - \mathbf{X}_L\|_2^2 \quad (14)$$

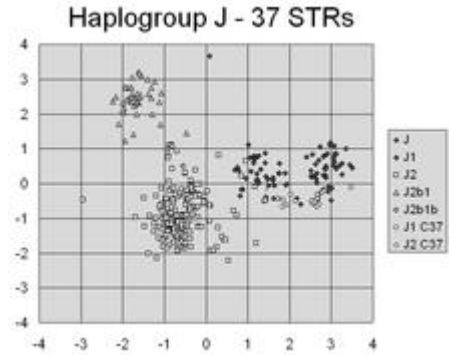

Fig: 4. A principal components analysis scatterplot of Y-STR haplotypes calculated from repeat-count values for 37 Y-chromosomal STR markers from 354 individuals. PCA has successfully found linear combinations of the different markers that separate out different clusters corresponding to different lines of individuals' Y-chromosomal genetic descent.

Such dimensionality reduction can be a very useful step for visualising and processing high-dimensional datasets, while still retaining as much of the variance in the dataset as possible. For example, selecting L = 2 and keeping only the first two principal components finds the two-dimensional plane through the high-dimensional dataset in which the data is most spread out, so if the data contains clusters these too may be most spread out, and therefore most visible to be plotted out in a two-dimensional diagram; whereas if two directions through the data (or two of the original variables) are chosen at random, the clusters may be much less spread apart from each other, and may in fact be much more likely to substantially overlay each other, making them indistinguishable[12].

Similarly, in regression analysis, the larger the number of explanatory variables allowed, the greater is the chance of overfitting the model, producing conclusions that fail to generalise to other datasets. One approach, especially when there are strong correlations between different possible explanatory variables, is to reduce them to a few principal components and then run the regression against them, a method called principal component regression.

Dimensionality reduction may also be appropriate when the variables in a dataset are noisy. If each column of the dataset contains independent identically distributed Gaussian noise, then the columns of T will also contain similarly identically distributed Gaussian noise (such a distribution is invariant under the effects of the matrix W, which can be thought of as a high-dimensional rotation of the co-ordinate axes). However, with more of the total variance concentrated in the first few principal components compared to the same noise variance, the proportionate effect of the noise is less—the first few components achieve a higher signal-to-noise ratio. PCA thus can have

the effect of concentrating much of the signal into the first few principal components, which can usefully be captured by dimensionality reduction; while the later principal components may be dominated by noise, and so disposed of without great loss.

3.2.5 Properties and Limitations of PCA

Property 1: For any integer q, $1 \le q \le p$, consider the orthogonal linear transformation

$$y = \mathbf{B}'x \quad (15)$$

Where y is a q-element vector and $\mathbf{B}'$ is a (q × p) matrix, and let $\mathbf{\Sigma}_y = \mathbf{B}'\mathbf{\Sigma}\mathbf{B}$ be the variance-covariance matrix for y. Then the trace of $\mathbf{\Sigma}_y$, denoted $tr(\mathbf{\Sigma}_y)$, is maximized by taking $\mathbf{B} = \mathbf{A}_q$, where $\mathbf{A}_q$ consists of the first q columns of $\mathbf{A}$ ($\mathbf{B}'$ is the transposition of $\mathbf{B}$).

Property 2: Consider again the orthonormal transformation

$$y = \mathbf{B}'x \quad (16)$$

With x, $\mathbf{B}$, $\mathbf{A}$ and $\mathbf{\Sigma}_y$ defined as before. Then $tr(\mathbf{\Sigma}_y)$ is minimized by taking $\mathbf{B} = \mathbf{A}_q^*$, where $\mathbf{A}_q^*$ consists of the last q columns of $\mathbf{A}$.

The statistical implication of this property is that the last few PCs are not simply unstructured left-overs after removing the important PCs. Because these last PCs have variances as small as possible they are useful in their own right. They can help to detect unsuspected near-constant linear relationships between the elements of x, and they may also be useful in regression, in selecting a subset of variables from x, and in outlier detection.

Property 3: (the Spectral Decomposition of $\mathbf{\Sigma}$)

$$\mathbf{\Sigma} = \lambda_1\alpha_1\alpha_1' + \lambda_2\alpha_2\alpha_2' + \ldots + \lambda_p\alpha_p\alpha_p' \quad (17)$$

Before we look at its usage, we first look at diagonal elements,

$$var(x_j) = \sum_{k=1}^{p} \lambda_k \alpha_{kj}^2 \quad (18)$$

Then, perhaps the main statistical implication of the result is that not only can we decompose the combined variances of all the elements of x into decreasing contributions due to each PC, but we can also decompose the whole covariance matrix into $\lambda_k\alpha_k\alpha_k'$

Contributions from each PC. Although not strictly decreasing, the elements of $\lambda_k\alpha_k\alpha_k'$ will tend to become smaller as k increases, as $\lambda_k\alpha_k\alpha_k'$ decreases for increasing k, whereas the elements of α_k tend to stay 'about the same size' because of the normalization constraints: $\alpha_k'\alpha_k = 1, k = 1, 2, \ldots, p$

Following are the limitation of PCA

1. The results of PCA depend on the scaling of the variables. A scale-invariant form of PCA has been developed.
2. The applicability of PCA is limited by certain assumptions made in its derivation.

3.2.6 Computing PCA using the Covariance Method

The following is a detailed description of PCA using the covariance method. But note that it is better to use the singular value decomposition (using standard software).

The goal is to transform a given data set X of dimension p to an alternative data set Y of smaller dimension L. Equivalently, we are seeking to find the matrix Y, where Y is the Karhunen–Loève transform (KLT) of matrix X:

$$\mathbf{Y} = \mathbb{KLT}\{\mathbf{X}\} \quad (19)$$

1. Organize the Data Set

Suppose you have data comprising a set of observations of p variables, and you want to reduce the data so that each observation can be described with only L variables, L < p. Suppose further, that the data are arranged as a set of n data vectors $\mathbf{X}_1 \ldots \mathbf{X}_n$ with each $\mathbf{X}_i$ representing a single grouped observation of the p variables.

- Write $\mathbf{X}_1 \ldots \mathbf{X}_n$ as row vectors, each of which has p columns.
- Place the row vectors into a single matrix X of dimensions n × p.

2. Calculate the Empirical Mean

- Find the empirical mean along each dimension j = 1... p.
- Place the calculated mean values into an empirical mean vector u of dimensions p × 1.

$$u[j] = \frac{1}{n}\sum_{i=1}^{n} X[i,j] \quad (20)$$

3. Calculate the Deviations from the Mean

Mean subtraction is an integral part of the solution towards finding a principal component basis that minimizes the mean square error of approximating the data [12]. Hence we proceed by centering the data as follows:

- Subtract the empirical mean vector u from each row of the data matrix X.
- Store mean-subtracted data in the n × p matrix B.

$$\mathbf{B} = \mathbf{X} - \mathbf{h}\mathbf{u}^T \quad (21)$$

Where h is an n × 1 column vector of all 1s:

$$h[i] = 1 \qquad \text{for } i = 1, \ldots, n$$

4. Find the Covariance Matrix

- Find the p × p empirical covariance matrix C from the outer product of matrix B with itself:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{B}^* \cdot \mathbf{B} \quad (22)$$

Where $*$ is the conjugate transpose operator. Note that if B consists entirely of real numbers, which is the case in many applications, the "conjugate transpose" is the same as the regular transpose.

- Please note that outer products apply to vectors. For tensor cases we should apply tensor products, but the covariance matrix in PCA is a sum of outer products between its sample vectors; indeed, it could be represented as B*.B. See the covariance matrix sections on the discussion page for more information.
- The reasoning behind using N-1 instead of N to calculate the covariance is Bessel's correction

5. Find the Eigenvectors and Eigenvalues of the Covariance Matrix

- Compute the matrix V of eigenvectors which diagonalizes the covariance matrix C:

$$\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{D} \quad (23)$$

where D is the diagonal matrix of eigenvalues of C. This step will typically involve the use of a computer-based

algorithm for computing eigenvectors and eigenvalues. These algorithms are readily available as sub-components of most matrix algebra systems. Matrix D will take the form of an p × p diagonal matrix, where

$$D[k,l] = \lambda_k \quad \text{for } k = l \qquad (24)$$

is the jth eigenvalue of the covariance matrix C, and

$$D[k,l] = 0 \quad \text{for } k \neq l.$$

- Matrix V, also of dimension p × p, contains p column vectors, each of length p, which represent the p eigenvectors of the covariance matrix C.
- The eigenvalues and eigenvectors are ordered and paired. The jth eigenvalue corresponds to the jth eigenvector.

6. Rearrange the Eigenvectors and Eigenvalues

- Sort the columns of the eigenvector matrix V and eigenvalue matrix D in order of decreasing eigenvalue.
- Make sure to maintain the correct pairings between the columns in each matrix.

7. Compute the Cumulative Energy Content for Each Eigenvector

The eigenvalues represent the distribution of the source data's energy among each of the eigenvectors, where the eigenvectors form a basis for the data. The cumulative energy content g for the jth eigenvector is the sum of the energy content across all of the eigenvalues from 1 through j:

$$g[j] = \sum_{k=1}^{j} D[k,k] \quad \text{for} \quad j = 1,\ldots,p \qquad (25)$$

8. Select a Subset of the Eigenvectors as Basis Vectors

- Save the first L columns of V as the p × L matrix W:

$$W[k,l] = V[k,l] \quad \text{for} \quad k = 1,\ldots,p \qquad l = 1,\ldots,L$$

$$1 \leq L \leq p. \qquad (26)$$

Where

- Use the vector g as a guide in choosing an appropriate value for L. The goal is to choose a value of L as small as possible while achieving a reasonably high value of g on a percentage basis. For example, you may want to choose L so that the cumulative energy g is above a certain threshold, like 90 percent. In this case, choose the smallest value of L such that

$$\frac{g[L]}{g[p]} \geq 0.9 \qquad (27)$$

9. Convert the Source Data to Z-Scores

- Create an p × 1 empirical standard deviation vector s from the square root of each element along the main diagonal of the diagonalized covariance matrix C. (Note, that scaling operations do not commute with the KLT thus we must scale by the variances of the already-decorrelated vector, which is the diagonal of C) :

$$\mathbf{s} = \{s[j]\} = \{\sqrt{C[j,j]}\} \quad \text{for } j = 1,\ldots,p \qquad (28)$$

- Calculate the n × p z-score matrix:

$$\mathbf{Z} = \frac{\mathbf{B}}{\mathbf{h} \cdot \mathbf{s}^T} \quad \text{(Divide element-by-element)} \qquad (29)$$

10. Project the Z-Scores of the Data onto the New Basis

- The projected vectors are the columns of the matrix

$$\mathbf{T} = \mathbf{Z} \cdot \mathbf{W} = \mathrm{KLT}\{\mathbf{X}\}. \qquad (30)$$

The rows of matrix T represent the Karhunen–Loeve transforms (KLT) of the data vectors in the rows of matrix X.

3.3 PCA Apporach for Eigen Faces

$$C = \frac{1}{M}\sum_{n-1}^{M} \Phi_n \Phi_n^T = AA^T$$

$$L = A^T A \quad L_{n,m} = \Phi_m^T \Phi_n$$

$$u_l = \sum_{k-1}^{M} v_{lk} \Phi_k \quad l = 1,\ldots,M$$

$$\omega_k = u_k^T (\Gamma_{\text{new}} - \Psi) \quad k = 1 \ldots M'$$

$$\Omega_{\text{new}}^T = [\ \omega_1 \quad \omega_2 \quad \ldots \quad \omega_{M'}\ ] \qquad (31)$$

3.4 Flow Chart for Face Ecognition using PCA

Face recognition by using PCA has been implemented on following figure. Feature extraction from the data set has been calculated from PCA algorithm.

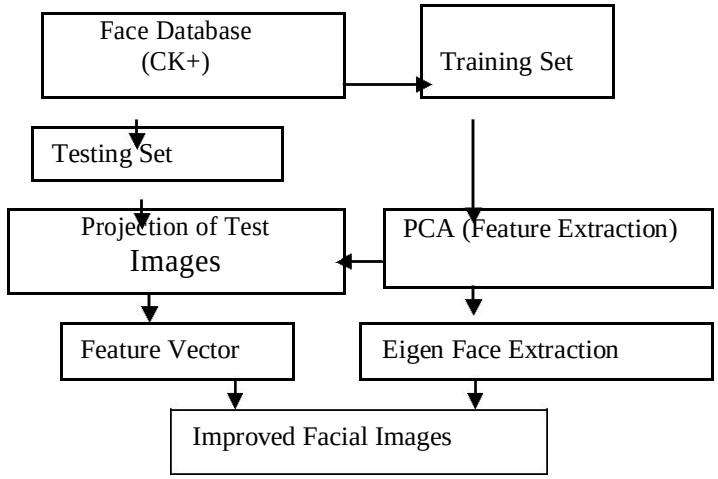

Fig: 5. Flow Chart for Face Recognition using PCA

4. Experimental Results

Our experimental result shows the following comparison chart of various algorithms. Among these we find PCA is efficient algorithm depending on eigen face value detection.

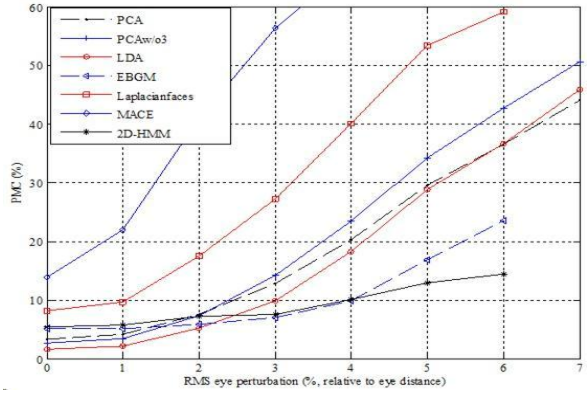

Fig: 6. Accuracy Experimental Graph between Various Facial Recognition Algorithms

Depending on the CK+ dataset following are the experimental results by PCA.

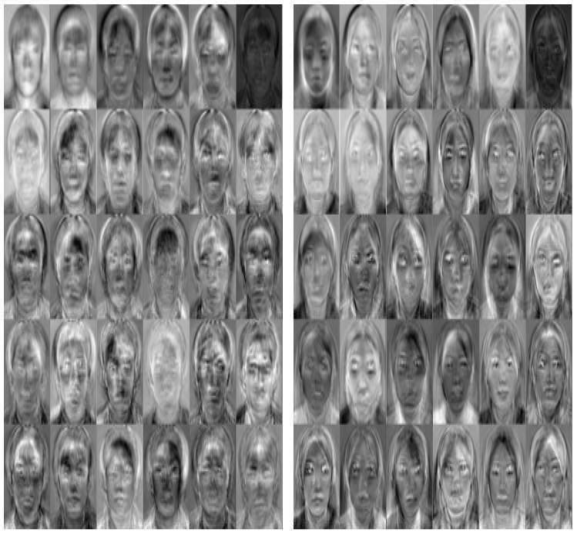

Fig: 7. Illustration of our Experimental Result Depending Upon Dataset

Now we implement the PCA algorithm on this test data base, crop the every single image and find the eigen values for every image. Compare our result with others algorithms and find the accuracy recognition for facial images.
Accuracy chart achieved by PCA is as follows:

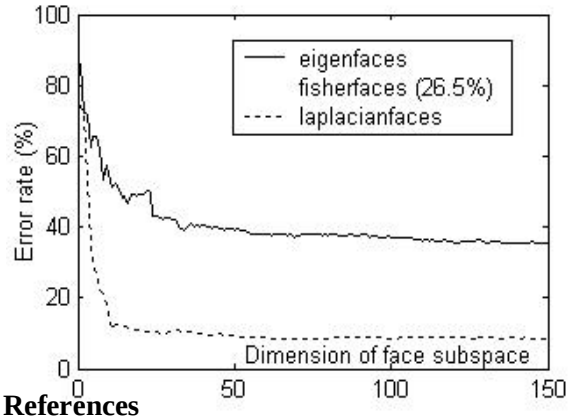

Fig: 8. Recognition Accuracy vs. Dimensionality Reduction on CK+ Data Set

5. Conclusion and Future Scope

The manifold ways of face analysis (representation and recognition) is introduced in this paper in order to detect the underlying nonlinear manifold structure in the manner of linear subspace learning. To the best of our knowledge, this is the first devoted work on face representation and recognition which explicitly considers the manifold structure. The manifold structure is approximated by the adjacency graph computed from the data points. Using the notion of the PCA of the graph, we then compute a transformation matrix which maps the face images into a face subspace. . This linear transformation optimally preserves local manifold structure. Theoretical analysis of PCA is provided. Experimental results on CK+ databases show the effectiveness of our method [18].

One of the central problems in face manifold learning is to estimate the intrinsic dimensionality of the *nonlinear* face manifold, or, degrees of freedom. We know that the dimensionality of the manifold is equal to the dimensionality of the local tangent space. Some previous works show that the local tangent space can be approximated using points in a neighbor set. Therefore, one possibility is to estimate the dimensionality of the tangent space. Another possible extension of our work is to consider the use of the unlabeled samples. It is important to note that the work presented here is a general method for face analysis (face representation and recognition) by discovering the underlying face manifold structure. Learning the face manifold is essentially an unsupervised learning process. And in many practical cases, one finds a wealth of easily available unlabeled samples. These samples might help to discover the face manifold. For example, it is shown how unlabeled samples are used for discovering the manifold structure and hence improving the classification accuracy. Since the face images are believed to reside on a sub-manifold embedded in a high-dimensional ambient space, we believe that the unlabeled samples are of great value. We are currently exploring these problems in theory and practice.

References

[1] Felix Juefei-Xu, Student Member, IEEE, and Marios Savvides, Member, IEEE,, Subspace-Based Discrete Transform Encoded Local Binary Patterns Representations for Robust Periocular Matching on NIST's Face Recognition Grand Challenge, IEEE Transactions on Image Processing, 23(8), 2014.

[2] Rabia Jafri, Hamid R. Arabnia, A Survey of Face Recognition Techniques, Journal of Information Processing Systems,5(2),2009.

[3] L. Sirovich, M. Kirby, Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America A - Optics, Image Science and Vision, 4(3):519–524, 1987.

[4] M. Kirby and L. Sirovich. Application of the karhunen-Stan Z. Li, Feature Correlation Filter for Face Recognition, loeve procedure for the characterization of human faces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(1):103–108, 1990.

[5] M. Turk, A. Pentland. Eigenfaces for recognition. Journal of Cog- nitive Neurosicence, 3(1):71–86, 1991.

[6] B. Dudley. ‖e3: New info on Microsoft's natal – how it works, multiplayer and pc versions‖. The Seattle Times, 2009.

[7] K. Massy. Toyota develops eyelid-monitoring system‖. Cnet reviews, 2008.

[8] M. McWhertor. ‖sony spills more ps3 motion controller details to devs‖. Kotaku. Gawker Media, 2009. http://kotaku.com/5297265/sony-spills-more-ps3-motion-controllerdetails- to-devs.

[9] T. Kanade, Computer Recognition of Human Faces, Brikhauser Verlag, Basel and Stuttgart, ISR-47, 1-106, 1977

[10] L. Wiskott, J. M. Fellous, N. Kruger, C. V. Malsburg, Face Recognition by Elastic Bunch Graph Matching,

IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 19(7), 775-779, 1997

[11] K. Chung, S. C. Kee, S. R. Kim, Face Recognition using Principal Component Analysis of Gabor Filter Reponses, in Proceedings of International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, 53-57, 1999.

[12] A. Pentland, B. Moghaddam, T. Starner, View-based and Modular Eigen spaces for FaceRecognition, in Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'94), 84-91, 1994.

[13] D. J. Sturman, D. Zetler, A Survey of Glove-Based Input, IEEE Computer Graphics and Applications, 14, 30-39, 1994.

[14] V. I. Pavlovic, R. Sharma, T. S. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 19(7), 677-695, 1997.

[15] T. Kenade. Picture Processing System by Computer Complex and Recognition of Human Faces. Kyoto University, November 1973

[16] Patrick Lucey1;2, Jeffrey F. Cohn1;2, Takeo Kanade1, Jason Saragih1, Zara Ambadar, The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression, Disney Research, 4615 Forbes Ave, Pittsburgh, PA 15213.

[17] Xiangxin Zhu, Shengcai Liao, Zhen Lei, Rong Liu, and Center for Biometrics and Security Research & National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, 100080 Beijing, China.

[18] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, Hong-Jiang Zhang, Face Recognition Using Laplacianfaces, Microsoft Research Asia, Beijing 100080, China.