AN EFFICIENT METHOD FOR FACE RECOGNITION USING PRINCIPAL COMPONENT ANALYSIS (PCA)

First A. Gunjan Dashore, M.Tech student in Computer Science and Engineering,
Dr.MGR Educational and Research Institute Chennai, Tamilnadu, INDIA, gunjandashore@hotmail.com [1];
Second B. Dr. V.Cyril Raj, Head of the Computer Science & Engg. Department,
Dr.MGR Educational and Research Institute Chennai, Tamilnadu, INDIA, cyrilraj@hotmail.com [2];

Abstract

An Efficient method for face recognition using Principal Component Analysis (PCA). The PCA has been extensively employed for face recognition algorithms. It is one of the most popular representation methods for a face image. It not only reduces the dimensionality of the image, but also retains some of the variations in the image data. The system functions by projecting face image onto a feature space that spans the significant variations among known face images. The significant features are known as “Eigen faces”, because they are the eigenvectors (Principal Component) of the set of faces they do not necessarily correspond to the features such as eyes, ears, and noses. The projection operation characterize an individual face by a weighted sum of the Eigen faces features and so to recognize a particular face it is necessary only to compare these weights to those individuals.

Key Terms: Face recognition, Principal Component Analysis, Eigen faces, Eigenvectors.

Introduction

The face is our primary focus of attention in social intercourse, playing major role in conveying identity and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize faces is Remarkable. We can recognize thousands of faces learned throughout our lifetime and identify familiar faces at a glance even after years of separation. This skill is quite robust, despite large changes in the visual stimulus due to viewing conditions, expression, aging, and distraction such as glasses, beards or changes in hairstyle. Face recognition has become an important issue in many applications such as security systems, credit card verification and criminal identification. For example, the ability to model a particular face and distinguish it from a large number of stored face models would make it possible to vastly improve criminal identification. Even the ability to merely detect faces, as opposed to recognizing them, can be important. Detecting faces in photographs for Automating color film development can be very useful, since the effect of many enhancement and noise reduction techniques depends on the image content. Although it is clear that people are good at face recognition, it is not at all obvious how faces are encoded or decoded by the human brain. Unfortunately developing a computational model of face recognition is quite difficult, because faces are complex, multidimensional visual stimuli. Therefore, face recognition is a very high-level computer vision task, in which many early vision techniques can be involved.

The first step of human face identification is to extract the relevant Features from facial images. Research in the field primarily intends to generate sufficiently reasonable familiarities of human faces so that another human can correctly identify the face. The question naturally arises as to how well facial features can be quantized. If such a quantization if possible then a computer capable of recognizing a face had given a set of features. Certain facial characteristics are used by human beings to identify faces. There are three major research groups, which propose three different approaches to the face recognition problem. The largest groups have dealt with facial characteristics, which are used by human beings in recognizing Individual faces. The second group performs human face identification based on feature vectors extracted from profile silhouettes. The third group uses feature vectors extracted from a frontal view of the face. Although there are three different approaches to the face recognition problem, there are two basic methods from which these three different approaches arise. The first method is based on the information theory concepts, in other words, on the principal component analysis methods. In this approach, the most relevant information that best describes a face is derived from the entire face image. Any particular face could be economically represented in terms of a best coordinate system that they termed "eigenfaces". These are the Eigen functions of the averaged covariance of the ensemble of faces. Another possible approach would be to take the face image as a whole identity. Statistically, faces can also be very similar. Walking through a crowd without glasses, blurry vision can often result in misidentifying someone; one person may have a greater distance between his or her eyes then another, so two regions of pixels will be correlated to one another differently for image sets of these two people. The eigenface technique finds a way to create ghost-like faces that represent the majority of variance in an image database. The scheme is based on an information theory approach that decomposes face images into a small set of characteristic feature images called 'eigen-faces',which are actually the principal components of the initial training set of face images. Recognition is performed by projecting a new image into the subspace spanned by the eigenfaces ('face space') and then classifying the face by comparing its position in the face space with the positions of the known individuals.

Related Work

Automated face recognition is a relatively new concept. Developed in the1960s,the first semi-automated system for face recognition required the administrator to locate features (such as eyes, ears, nose, and mouth) on the photographs before it calculated distances and ratios to a common reference point, which were then compared to reference data. In the 1970s, Goldstein, Harmon, and Lesk used 21 specific subjective markers such as hair color and lip thickness to automate the recognition. The problem with both of these early solutions was that the measurements and locations were manually computed. In 1988, Kirby and Sirovich applied principle component analysis, a standard linear algebra technique, to the face recognition problem.

In 1991, Turk and Pentland discovered that while using the eigenfaces techniques, the residual error could be used to detect faces in images Although the approach was somewhat constrained by the environmental factors, the nonetheless created significant interest in furthering automated face recognition technologies.

The technology first captured the public's attention from the media reaction to a trial implementation at the January 2001 Super Bowl, which captured surveillance images and compared them to a database of digital mug shots. This demonstration initiated much-needed analysis on how to use the technology to support national needs while being considerate of the public's social and privacy concerns.

Today, face recognition technology is being used to combat passport fraud, support law enforcement, identify missing children, and minimize benefit/identity fraud.

Methodologies

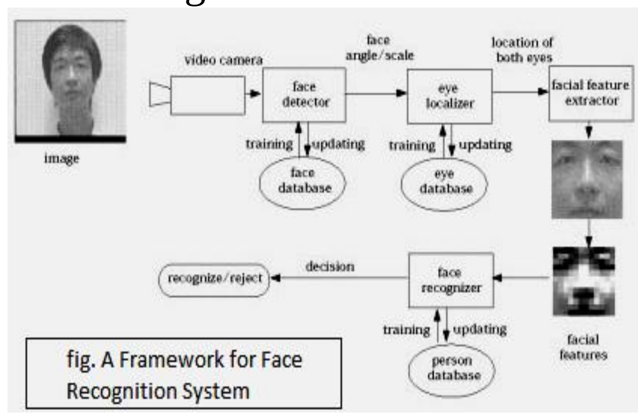


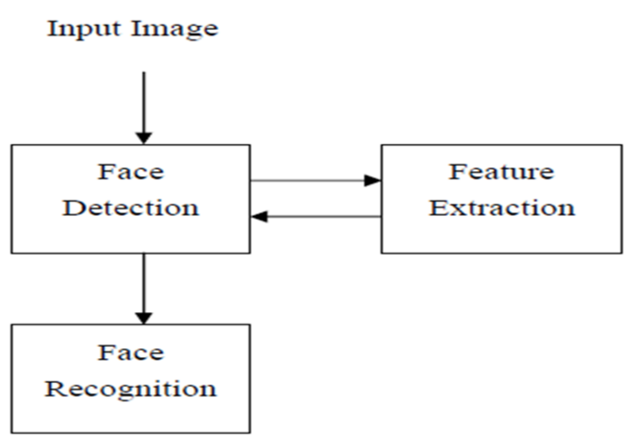
fig. A Framework for Face Recognition System

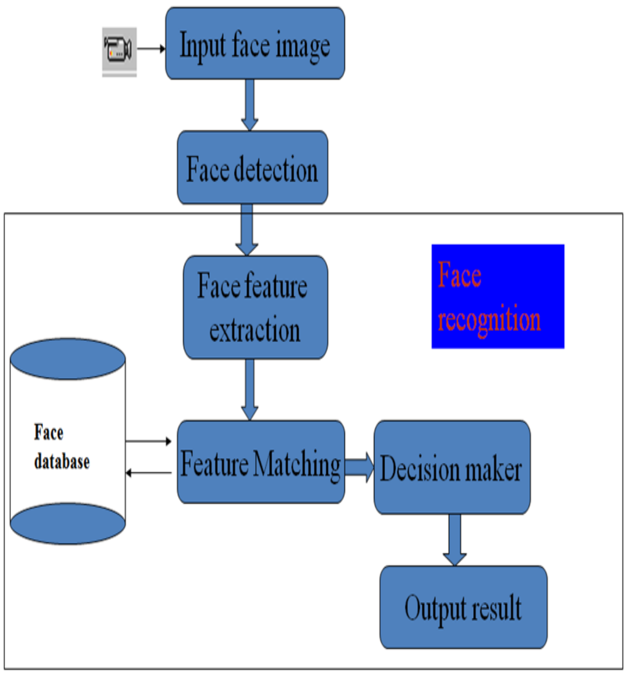
"Face Recognition" is a very active area in the Computer Vision and Biometrics fields, as it has been studied vigorously for 25 years.

"Face Recognition" generally involves two stages:

1. Face Detection: where a photo is searched to find any face (shown here as a green rectangle), then image processing cleans up the facial image for easier recognition.
2. Face Recognition: where that detected and processed face is compared to a database of known faces, to decide who that person is (shown here as red text).

However, Face Recognition is much less reliable than Face Detection, generally 30-70% accurate. Face Recognition has been a strong field of research since the 1990s, but is still far reliable.





Work Flow Model

Principle Component Analysis (PCA)

The `Eigen Object Recognizer` class applies PCA on each image, the results of which will be an array of Eigen values that a Neural Network can be trained to recognize. PCA is a commonly used method of object recognition as its results, when used properly can be fairly accurate and resilient to noise. The method of which PCA is applied can vary at different stages so what will be demonstrated is a clear method for PCA application that can be followed. It is up for individuals to experiment in finding the best method for producing accurate results from PCA.

To perform PCA several steps are undertaken:

- Stage 1: Subtract the Mean of the data from each variable (our adjusted data)
- Stage 2: Calculate and form a covariance Matrix
- Stage 3: Calculate Eigenvectors and Eigenvalues from the covariance Matrix
- Stage 4: Chose a Feature Vector (a fancy name for a matrix of vectors)
- Stage 5: Multiply the transposed Feature Vectors by the transposed adjusted data

STAGE 1: Mean Subtraction

This data is fairly simple and makes the calculation of our covariance matrix a little simpler now this is not the subtraction of the overall mean from each of our values as for covariance we need at least two dimensions of data. It is in fact the subtraction of the mean of each row from each element in that row.

(Alternatively the mean of each column from each element in the column however this would adjust the way we calculate the covariance matrix).

STAGE 2: Covariance Matrix

The basic Covariance equation for two dimensional data is:

$$cov(x,y) = \frac{\sum_{i-1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)}$$

Which is similar to the formula for variance however, the change of x is in respect to the change in y rather than solely the change of x in respect to x. In this equation x represents the pixel value and $\bar{x}$ is the mean of all x values, and n the total number of values. The covariance matrix that is formed of the image data represents how much the dimensions vary from the mean with respect to each other. The definition of a covariance matrix is:

$$\boldsymbol{C^{n*n} = \left(C_{i,j}, C_{i,j} = cov\left(Dim_i, Dim_j\right)\right)}$$

Now the easiest way to explain this is but an example the easiest of which is a 3x3 matrix.

$$C_{mat} = \begin{pmatrix} cov(x,y) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

$$/1 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$C(/1) = \begin{pmatrix} cov(1,1) & cov(2,5) & cov(3,9) \\ cov(4,1) & cov(5,5) & cov(6,9) \\ cov(7,1) & cov(8,5) & cov(9,9) \end{pmatrix}$$

Now with larger matrices this can become more complicated and the use of computational algorithms essential.

STAGE 3: Eigenvectors and Eigen values

Eigen values are a product of multiplying matrices however they are as special case. Eigen values are found by multiples of the covariance matrix by a vector in 2 dimensional space (i.e. a Eigenvector). This makes the covariance matrix the equivilant of a transformation matrix. It is easier to show in a example:

$$\boldsymbol{Covarience\ Matrix} = \begin{bmatrix} \mathbf{2} & \mathbf{3} \\ \mathbf{2} & \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{Eigenvector} = \begin{bmatrix} \mathbf{6} \\ \mathbf{4} \end{bmatrix}$$

$$\boldsymbol{Multiplied}{:}\begin{bmatrix} \mathbf{2} & \mathbf{3} \\ \mathbf{2} & \mathbf{1} \end{bmatrix} * \begin{bmatrix} \mathbf{6} \\ \mathbf{4} \end{bmatrix} = \begin{bmatrix} \mathbf{2} & \mathbf{4} \\ \mathbf{1} & \mathbf{6} \end{bmatrix} = \mathbf{4}\begin{bmatrix} \mathbf{6} \\ \mathbf{4} \end{bmatrix}$$

Eigenvectors can be scaled so ½ or x2 of the vector will still produce the same type of results. A vector is a direction and all you will be doing is changing the scale not the direction.

$$\begin{bmatrix} \mathbf{2} & \mathbf{3} \\ \mathbf{2} & \mathbf{1} \end{bmatrix} * \begin{bmatrix} \mathbf{3} \\ \mathbf{2} \end{bmatrix} = \begin{bmatrix} \mathbf{12} \\ \mathbf{8} \end{bmatrix} = \mathbf{4} \begin{bmatrix} \mathbf{3} \\ \mathbf{2} \end{bmatrix}$$

Eigenvectors are usually scaled to have a length of 1:

$$\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix} \ \boldsymbol{becomes} \ \begin{bmatrix} \boldsymbol{A}/\left(\sqrt{\boldsymbol{A}^{1}+\boldsymbol{B}^{2}}\right) \\ \boldsymbol{B}/\left(\sqrt{\boldsymbol{A}^{1}+\boldsymbol{B}^{2}}\right) \end{bmatrix}$$

Thankfully finding these special Eigenvectors is done for you and will not be explained however there are several tutorials available on the web to explain the computation. The Eigen value is closely related to the Eigenvector used and is the value of which the original vector was scaled in the example the Eigen value is 4.

STAGE 4: Feature Vectors

Now a usually the results of Eigen values and Eigenvectors are not as clean as in the example above. In most cases the results provided are scaled to a length of 1.

So here are some example values calculated using Matlab:

$$Covarience\ Matrix = \begin{bmatrix} 0.2120.345 \\ 0.6210.111 \end{bmatrix}$$

$$Eigenvector = \begin{bmatrix} 0.6271 \\ 0.3041 \end{bmatrix}$$

$$Eigenvalues = \begin{bmatrix} 0.6392 - 0.557 \\ 0.7691\ 0.3814 \end{bmatrix}$$

Once Eigenvectors are found from the covariance matrix, the next step is to order them by Eigenvalue, highest to lowest. This gives you the components in order of significance. Here the data can be compressed and the weaker vectors are removed producing a lossy compression method, the data lost is deemed to be insignificant.

$$Resultant\ Eigenvalues = \begin{bmatrix} 0.6392 \\ 0.7691 \end{bmatrix}$$

STAGE 5: Transposition

The final stage in PCA is to take the transpose of the feature vector matrix and multiply it on the left of the transposed adjusted data set (the adjusted data set is from Stage 1 where the mean was subtracted from the data).

The Eigen Object Recognizer class performs all of this and then feeds the transposed data as a training set into a Neural Network. When it is passed an image to recognize it performs PCA and compares the generated Eigen values and Eigenvectors to the ones from the training set the Neural Network then produces a match if one has been found or a negative match if no match is found. The is a little more to it than this however the use of Neural Networks is a complex subject to cover and is not the object of this article.

Results And Analysis

A. How to preprocess facial images for Face Recognition:

Most face recognition algorithms are extremely sensitive to lighting conditions, so that if it was trained to recognize a person when they are in a dark room, it probably won't recognize them in a bright room, etc. This problem is referred to as "lumination dependent", and there are also many other issues, such as the face should also be in a very consistent position within the images (such as the eyes being in the same pixel coordinates), consistent size, rotation angle, hair and makeup, emotion (smiling, angry, etc), position of lights (to the left or above, etc). This is why it is so important to use a good image preprocessing filters before applying face recognition.

Here you can see an example of this preprocessing stage:

B. How Eigenfaces can be used for Face Recognition:

So we should collect a group of preprocessed facial images of each person you want to recognize. For example, if you want to recognize someone from a class of 10 students, then you could store 20 photos of each person, for a total of 200 preprocessed facial images of the same size (say 100x100 pixels).

Use "Principal Component Analysis" to convert all your 200 training images into a set of "Eigen faces" that represent the main differences between the training images. First it will find the "average face image" of your images by getting the mean value of each pixel. Then the eigenfaces are calculated in comparison to this average face, where the first eigenface is the most dominant face differences, and the second eigenface is the second most dominant face differences, and so on, until you have about 50 eigenfaces that represent most of the differences in all the training set images.

In these example images above you can see the average face and the first and last eigenfaces that were generated from a collection of 30 images each of 4 people. Notice that the average face will show the smooth face structure of a generic person, the first few eigenfaces will show some dominant features of faces, and the last eigenfaces (eg: Eigen face 119) are mainly image noise.

You can see the first 32 Eigen faces in the image below.

C. Explanation of Face Recognition using Principal Component Analysis:

To explain Eigen faces (Principal Component Analysis) in simple terms, Eigen faces figures out the main differences between all the training images, and then how to represent each training image using a combination of those differences.

So for example, one of the training images might be made up of:

(Average Face) + (13.5% of eigenface0) - (34.3% of eigenface1) + (4.7% of eigenface2) + ... + (0.0% of eigenface199).

Once it has figured this out, it can think of that training image as the 200 ratios:

{13.5, -34.3, 4.7... 0.0}.

It is indeed possible to generate the training image back from the 200 ratios by multiplying the ratios with the eigenface images, and adding the average face. But since many of the last eigenfaces will be image noise or wont contribute much to the image, this list of ratios can be reduced to just the most dominant ones, such as the first 30 numbers, without effecting the image quality much.

To recognize a person in a new image, it can apply the same PCA calculations to find 200 ratios for representing the input image using the same 200 eigenfaces. And once again it can just keep the first 30 ratios and ignore the rest as they are less important. It can then search through its list of ratios for each of its 20 known people in its database, to see who has their top 30 ratios that are most similar to the 30 ratios for the input image. This is basically a method of checking which training image is most similar to the input image, out of the whole 200 training images that were supplied.

D. Accuracy Graph

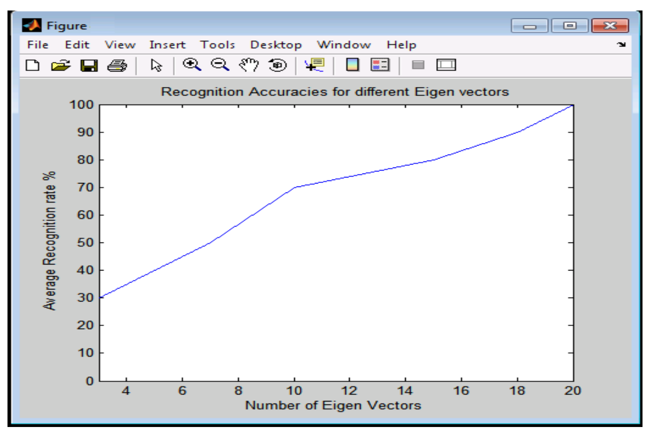


Figure 1: Graph shows the relationship between Eigen vectors and the Average Recognition Percent (%).

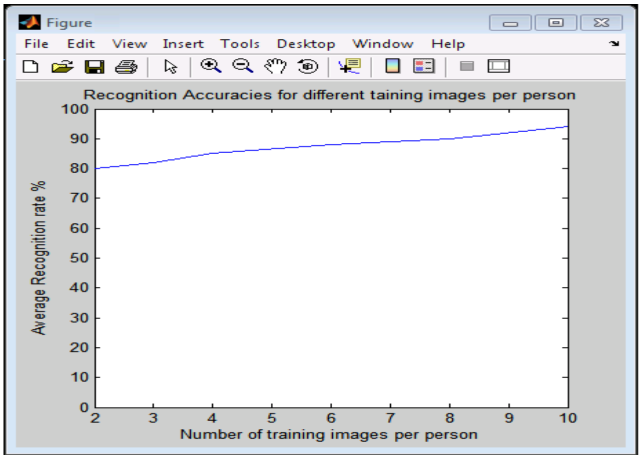


Figure 2: Graph shows the relationship between No. of Training images per person & Average Recognition rate Percent (%).

Conclusion

The paper has proposed an algorithm for real-time human face tracking is realized. The algorithm takes the advantage not only of geometric relations between a human face, but also of a precise other feature extraction. The frame-rate is up to 10 fps on a computer with a 1.4GHz Pentium 4 CPU. The accuracy of the single-face detection is better than 92% with a simple background and sufficient light source. From experiments, the multi-face detection results in slightly higher rate of misjudgment. The face detection is accomplished regardless of the viewpoints no matter it is a front view or a side view.

A hybrid solution to frontal face detection using facial features and Eigen faces theory is presented. Using a facial feature extraction step prior to performing PCA analysis helps to address two requirements for this system. Firstly, the search for faces does not need to be carried out at every pixel location in the image since a small search space can be obtained using the detected facial feature points. Secondly, the face detection process can be carried out in one cycle over a normalized search space, thereby avoiding the requirement of processing the image at multiple scales. The final stage was to covert the code to a Graphical User interface so that the system can be used friendly. The most important component of this interface was to provide the system with a person image and identifying him using the stored data. We faced several problems in that affect the result. These include the image size and lightning of the room. A solution to this problem was to focus a white light on a person before we take his picture. Other problem was the picture's background. When a picture is taken for a person, unwanted disturbances from the surrounding that result in a low accuracy. A solution was to use only face detection

References

[1]. Ming-Hsuan Yang, David J. Kriegman and Narendra Ahuja, "Detecting Faces in Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 24, no. 1, pp. 696-706, January 2002.

[2]. A. Eleftheriadis and A Jacquin, "Automatic Face Location, Detection and Tracking for Model Assisted Coding of Video Teleconferencing Sequences at Low Bit Rates", Signal processing: Image Communication, vol. 7, no. 3, pp. 231-248, July 1995.

[3]. Karin Sobottka and Ioannis Pitas, "A Novel Method for Automatic Face Segmentation, Facial Feature Extraction and Tracking", Signal Processing: Image Communication, vol. 12, no.3, pp. 263-281, 1998.

[4]. Rein-Lien Hsu, Mohamed Abdel-Mottaleb and Anil K. Jain, "Face Detection in Color Images", IEEE Transactions on Pattern Analysis and Machine

[5]. Intelligence, vol. 24, no. 5, pp. 696-706, May 2002.

[6]. Kin Choong Yow, Roberto Cipolla, "Feature-Based Human Face Detection", Image and Vision Computing, vol. 15, pp. 713-735, 1997.

[7]. B. Menser and F. Muller, "Face Detection in Color Images using Principal Component Analysis", IEE Conference Publication, vol.2, no. 465, pp. 620-

[8]. 624, 1999.

[9]. Henry A. Rowley, Shumeet Baluja and Takeo Kanade, "Neural Network- Based Face Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, January 1998.

[10]. 8. Chengjun Liu, "A Bayesian Discriminating Features Method for Face Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, January 1998.

[11]. Cheng-Chung Lin and Wei-Chung Lin, "Extracting Facial Features by an PCA", Pattern Recognition, vol. 29, no. 12, 1996.

[12]. Matthew Turk and Alex Pentland, "Eigen faces for Recognition", Journal of Cognitive Neuroscience, vol.3, no. 1, pp. 71-86, 1991.

Biographies

[(1)]**Author** Gunjan Dashore is a M.Tech student iin computer science engineering, Dr. M.G.R. Educational & Research Institute University, Chennai-600095, Tamilnadu, India. This author can be contactedth-rough.gunjandashore@hotmail.com.

[(2)]**Author** Dr.V. Cyril Raj, is a Head of the CSE Department, Dr. M.G.R. Educational & Research Institute University, Chennai-600095, Tamilnadu, India. This author can be contacted through cyrilraj@hotmail.com.