Face recognition

An Eigenface approach

Mahesh Subedi
2058 Computer

Introduction

The detection of faces and facial features from an arbitrary uncontrived image is a critical precursor to recognition. A robust scheme is needed to detect the face as well as determine its precise placement to extract the relevant data from an input image. This is necessary to properly prepare the image's 2D intensity description of the face for input to a recognition system. This detection scheme must operate flexibly and reliably regardless of lighting conditions, background clutter in the image, multiple faces in the image, as well as variations in face position, scale, pose and expression. The geometrical information about each face in the image that we gather at this stage will be used to apply geometrical transformations that will map the data in the image into an invariant form. By isolating each face, transforming it into a standard frontal mug shot pose and correcting lighting effects, we limit the variance in its intensity image description to the true physical shape and texture of the face itself.

Over the last ten years, face recognition has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding. Because of the nature of the problem, not only computer science researchers are interested in it, but neuroscientists and psychologists also. It is the general opinion that advances in computer vision research will provide useful insights to neuroscientists and psychologists into how human brain works, and vice versa.

> ***it's a high-level pattern recognition problem in which humans are very adept, whereas it can be quite challenging to teach a machine to do it.***

A general statement of the face recognition problem (in computer vision) can be formulated as follows: Given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces.

Research directions

- Recognition from outdoor facial images.
- Recognition from non-frontal facial images.
- Recognition at low false accept/alarm rates.
- Understanding why males are easier to recognize than females.
- Greater understanding of the effects of demographic factors on performance.
- Development of better statistical methods for understanding performance.
- Develop improved models for predicting identification performance on very large galleries.
- Effect of algorithm and system training on covariate performance.
- Integration of morphable models into face recognition performance.

History

Research efforts towards of faces recognition start in 1878 by English scientist **Sir Francis Galton**. His research involved the combination of people's photos, by means of superimposing face images. **Galton** proposed the photos alignment from human faces, depending on its main characteristic, putting some of them upon the other. The main difficulty was to describe the personal similarities, the types of faces and personal characteristics. To overcome this difficulty, biometrics characteristic were extracted from the face image to compare with the measures of another face in order to succeed in face recognition.

Biometrics and Face Recognition

Face Recognition is part of a larger context called Biometrics that give us the notion of life measure. Biometrics can be defined as some characteristics that can be used to verify an individual's identity. The biometrics system is essentially a pattern recognition system that makes a personal identification determining the authenticity of some characteristic of an individual. In fact, an automatic people identification system based exclusively on fingerprints or on face recognition does not satisfy all the functionality requirements. Face Recognition is natural and non-intrusive, but it is not trustworthy as compared to the fingerprint verification which is relatively more trustworthy but can be intrusive and can cause resistance to the users, depending on the application.

Why Face Recognition is Interesting?

Face recognition is interesting to study because it is an application area where computer vision research is being utilized in both military and commercial products. Much effort has been spent on this problem, yet there is still plenty of work to be done.

Basic research related to this field is currently active. For example, research searching for a fundamental theory describing how light and objects interact to produce images was recently published in April 2004. Often, practical applications can grow out of improvements in theoretical understanding and it seems that this problem will continue to demonstrate this growth.

Personally, I'm interested in this project because it's a high-level pattern recognition problem in which humans are very adept, whereas it can be quite challenging to teach a machine to do it. The intermediate and final visual results are interesting to observe in order to understand failures and successes of the various approaches.

How Computers See

Although we are able to visually recognize complex objects from an early age, visual recognition is very difficult to automate. A single object may be viewed from a number of different angles, in different lighting conditions, and with other objects partially obscuring view of the object.

Purpose of a vision system: To move from an initial digitized image to useful analysis of the scene. Goal is to recognize objects in a scene, given one or more images of that scene.

Ultimate goal: Develop a system with capabilities comparable to human capabilities.

Stages of the vision process

Begin with a digitized image, which gives the brightness at each point (pixel) in the image. The stages that follow:

Low-level Processing

Simple features are identified for example, lines or edges in the image. Output is something like a line drawing of the objects in the image, the lines separating the image into regions corresponding to object surfaces.

Medium-level Processing

Next, determine how far away the regions are and what their orientation is. Output referred to as "2 -D sketch."

High-level Processing

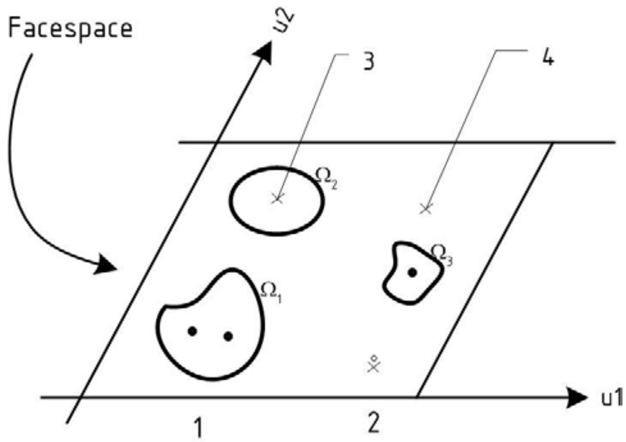
Obtain a useful high level description/representation of the scene. Work out the 3-D models of objects in the scene given the depth and orientation information above. Next, attempt is to recognize what sorts of objects occur in the scene.

Face Recognition Using Eigenfaces

Introduction

The information theory approach of encoding and decoding face images extracts the relevant information in a face image, encode it as efficiently as possible and compare it with database of similarly encoded faces. The encoding is done using features which may be different or independent than the distinctly perceived features like eyes, ears, nose, lips, and hair.

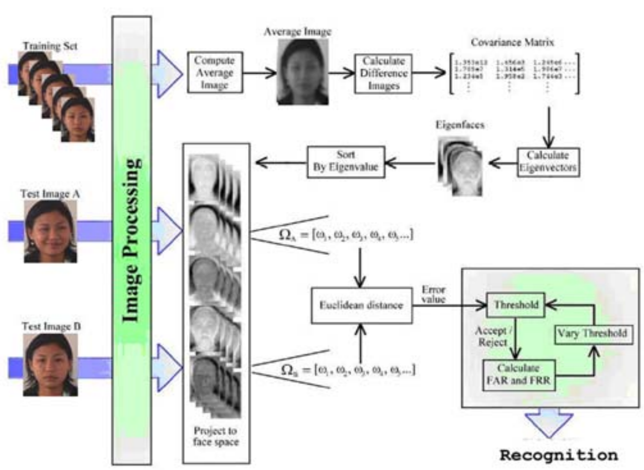
Mathematically, principal component analysis approach will treat every image of the training set as a vector in a very high dimensional space. The eigenvectors of the covariance matrix of these vectors would incorporate the variation amongst the face images. Now each image in the training set would have its contribution to the eigenvectors (variations). This can be displayed as an '*eigenface*' representing its contribution in the variation between the images. In each eigenface some sort of facial variation can be seen which deviates from the original image.



	Face Space	Known face class	Result
1	near	near	Recognized as Ω_1
2	near	far	Who are you ?
3	far	near	?False positive?
4	far	far	No face

Figure: a simplified version of face space to illustrate the four results of projecting an image into face space. In this case, there are two eigenfaces (u_1 and u_2) and three known individuals (Ω_1, Ω_2, and Ω_3).

The high dimensional space with all the eigenfaces is called the image space (feature space). Also, each image is actually a linear combination of the eigenfaces. The amount of overall variation that one eigenface counts for, is actually known by the eigenvalue associated with the corresponding eigenvector. If the eigenface with small eigenvalues are neglected, the image can be a linear combination of reduced number of these eigenfaces. For example, if there are M images in the training set, we would get M eigenfaces. Out of these, only M' eigenfaces are selected such that they are associated with the largest eigenvalues. These would span the M'-dimensional subspace 'face space' out of all the possible images (image space).



Overall Block diagram of Eigen Face Approach of Face Recognition

When the face image to be recognized (known or unknown), is projected on this face space (figure 1), you can get the weights associated with the eigenfaces, that linearly approximate the face or can be used to reconstruct the face. Now these weights are compared with the weights of the known face images so that it can be recognized as a known face in used in the training set. In simpler words, the Euclidean distance between the image projection and known projections is calculated; the face image is then classified as one of the faces with minimum Euclidean distance.

How Eigenface algorithm works?

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 which occur in any input signal. Such patterns, which can be observed in all signals, could be - in the domain of facial recognition - the presence of some objects (eyes, nose, mouth) in any face as well as relative distances between these objects. These characteristic features are called *eigenfaces* in the facial recognition domain (or *principal components* generally). They can be extracted out of original image data by means of a mathematical tool called *Principal Component Analysis* (PCA).

By means of PCA one can transform each original image of the training set into a corresponding eigenface. An important feature of PCA is that one can reconstruct any original image from the training set by combining the eigenfaces. Remember that eigenfaces are nothing less than characteristic features of the faces. Therefore one could say that the original face image can be reconstructed from eigenfaces if one adds up all the eigenfaces (features) in the right proportion. Each eigenface represents only certain features of the face, which may or may not be present in the original image. If the feature is present in the original image to a higher degree, the share of the corresponding eigenface in the 'sum' of the eigenfaces should be greater. If, contrary, the particular feature is not (or almost not) present in the original image, then the corresponding eigenface should contribute a smaller (or not at all) part to the sum of eigenfaces. So, in order to reconstruct the original image from the eigenfaces, one has to build a kind of weighted sum of all eigenfaces. That is, the reconstructed original image is equal to a sum of all eigenfaces, with each eigenface having a certain weight. This weight specifies, to what degree the specific feature (eigenface) is present in the original image. If one uses all the eigenfaces extracted from original images, one can reconstruct the original images from the eigenfaces *exactly*. But one can also use only a part of the eigenfaces. Then the reconstructed image is an approximation of the original image. However, one can ensure that losses due to omitting some of the eigenfaces can be minimized. This happens by choosing only the most important features (eigenfaces). Omission of eigenfaces is necessary due to scarcity of computational resources.

How does this relate to facial recognition? The clue is that it is possible not only to extract the face from eigenfaces given a set of weights, but also to go the opposite way. This opposite way would be to extract the weights from eigenfaces and the face to be recognized. These weights tell nothing less, as the amount by which the face in question differs from "typical" faces represented by the eigenfaces. Therefore, using this weights one can determine two important things:

1. Determine if the image in question is a face at all. In the case the weights of the image differ too much from the weights of face images (i.e. images, from which we know for sure that they are faces), the image probably is not a face.
2. Similar faces (images) possess similar features (eigenfaces) to similar degrees (weights). If one extracts weights from all the images available, the images could be grouped to clusters. That is, all images having similar weights are likely to be similar faces.

Calculation of Eigenfaces with PCA

In this section, I am presenting the original scheme for determination of the eigenfaces using PCA.

Step 1: Prepare the data

In this step, the faces constituting the training set ($\boldsymbol{\Gamma}$) should be prepared for processing.

Step 2: Subtract the mean

The average matrix $\boldsymbol{\Psi}$ has to be calculated, then subtracted from the original faces ($\boldsymbol{\Gamma}$) and the result stored in the variable:

$$\boldsymbol{\Psi} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{\Gamma}_i . \qquad (1)$$

$$\boldsymbol{\Phi}_i = \boldsymbol{\Gamma}_i - \boldsymbol{\Psi};\ i = 1, \ldots, M. \qquad (2)$$

Step 3: Calculate the covariance matrix

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

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{\Phi}_i\boldsymbol{\Phi}_i^T \qquad (3)$$

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

In this step, the eigenvectors (eigenfaces) u_i and the corresponding eigenvalues $\boldsymbol{\lambda}_i$ should be calculated. The eigenvectors (eigenfaces) must be normalized so that they are unit vectors, i.e. of length 1. The description of the exact algorithm for determination of eigenvectors and eigenvalues is omitted here, as it belongs to the standard arsenal of most math programming libraries.

Step 5: Select the principal components

From M eigenvectors (eigenfaces) u_i, only M' 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 M' eigenfaces u_i are determined, the "training" phase of the algorithm is finished.

Improvement of the Original Algorithm

There is a problem with the algorithm described in section 2.5. The covariance matrix **C** in step 3 has a dimensionality of $N^2 \times N^2$, so one would have N^2 eigenfaces and eigenvalues. For a 256 × 256 image that means that one must compute a 65,536 × 65,536 matrix and calculate 65,536 eigenfaces. Computationally, this is not very efficient as most of those eigenfaces are not useful for our task. So, the step 3 and 4 is replaced by the scheme:

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T \qquad (4)$$

$$= \mathbf{A}\mathbf{A}^T$$

$$\mathbf{L} = \mathbf{A}^T\mathbf{A}\mathbf{L}_{n,m} = \Phi_m^T\Phi_n \qquad (5)$$

$$u_i = \sum_{k=n-1}^{M} v_{1k}\Phi_k,\ i = 1,\ldots,\ldots,M$$

where **L** is a $M \times M$ matrix, v are M eigenvectors of **L** and u are eigenfaces. Note that the covariance matrix **C** is calculated using the formula $\mathbf{C} = \mathbf{A}\mathbf{A}^T$, the original (inefficient) formula is given only for the sake of explanation of **A**. The advantage of this method is that one has to evaluate only M numbers and not N^2. Usually, $M << N^2$ as only a few principal components (eigenfaces) will be relevant. The amount of calculations to be performed is reduced from the number of pixels ($N^2 \times N^2$) to the number of images in the training set (M).

In the step 5, the associated eigenvalues allow one to rank the eigenfaces according to their usefulness. Usually, we will use only a subset of M eigenfaces, the M' eigenfaces with the largest eigenvalues.

Classifying the Faces

The process of classification of a new (unknown) face Γ_{new} 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 Ω_{new}^T

$$\mathbf{w}_k = \mathbf{u}_k^T(\mathbf{T}_{new} - \Psi),\ k = 1,\ldots,\ldots,M' \qquad (6)$$

$$\Omega_{new}^T = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \ldots, \mathbf{w}_{M'}] \qquad (7)$$

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 , 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.

For comparison, two methods are used to describe a face class in the face space. The first method, referred to as the averaging representation, calculates the class vector by averaging the projected vectors from the training images of the corresponding individual. The second method, the point-set representation, describes a face class by a set of vectors projected from all the training images of an individual.

A distance threshold, θ_c, that defines the maximum allowable distance from a face class as well as from the face space, is set up by computing half the largest distance between any two face classes:

$$\theta_c = \frac{1}{2}\max_{j,k}\left\{\left\|\Omega_j - \Omega_k\right\|\right\};\ j,k = 1,\ldots,N_c. \qquad (8)$$

In the recognition stage, a new image, Γ, is projected into the face space to obtain a vector, Ω :

$$\Omega = \mathbf{U}^T(\Gamma - \Psi) \qquad (9)$$

The distance of Ω to each face class is defined by

$$\varepsilon_k^2 = \left\|\Omega - \Omega_k\right\|^2;\ k = 1,\ldots,N_c. \qquad (10)$$

For the purpose of discriminating between face images and non-face like images, the distance, ε, between the original image, Γ, and its reconstructed image from the eigenface space, Γ_f, is also computed:

$$\varepsilon^2 = \left\|\Gamma - \Gamma_f\right\|^2, \qquad (11)$$

where

$$\Gamma_f = \mathbf{U}\cdot\Omega + \Psi. \qquad (12)$$

These distances are compared with the threshold given in equation (8) and the input image is classified by the following rules:

- IF $\boldsymbol{\varepsilon} \geq \boldsymbol{\theta}_c$
  THEN input image is not a face image;
- IF $\boldsymbol{\varepsilon} < \boldsymbol{\theta}_c$ AND $\forall k, \boldsymbol{\varepsilon}_k \geq \boldsymbol{\theta}_c$
  THEN input image contains an unknown face;
- IF $\boldsymbol{\varepsilon} < \boldsymbol{\theta}_c$ AND $\boldsymbol{\varepsilon}_{k^*} = \min_k \{\boldsymbol{\varepsilon}_k\} < \boldsymbol{\theta}_c$
  THEN input image contains the face of individual $\boldsymbol{k}^*$.

A Euclidean Distance

Let an arbitrary instance $\boldsymbol{x}$ be described by the feature vector

$$\boldsymbol{x} = [\boldsymbol{a}_1(\boldsymbol{x}), \boldsymbol{a}_2(\boldsymbol{x}), \ldots, \ldots, \boldsymbol{a}_n(\boldsymbol{x})] \qquad \textbf{(13)}$$

where $ar(\mathbf{x})$ denotes the value of the r^{th} attribute of instance $\mathbf{x}$. Then the distance between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined to be $d(\mathbf{x}_i, \mathbf{x}_j)$:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum \left(\boldsymbol{ar}(\boldsymbol{x}_i) - \boldsymbol{ar}(\boldsymbol{x}_j)\right)^2} \qquad \textbf{(14)}$$

The distance measure at a given image location is

$$\boldsymbol{\varepsilon}^2 = \left\| \boldsymbol{\Phi} - \boldsymbol{\Phi}_f \right\|^2 \qquad \textbf{(15)}$$

Conclusion

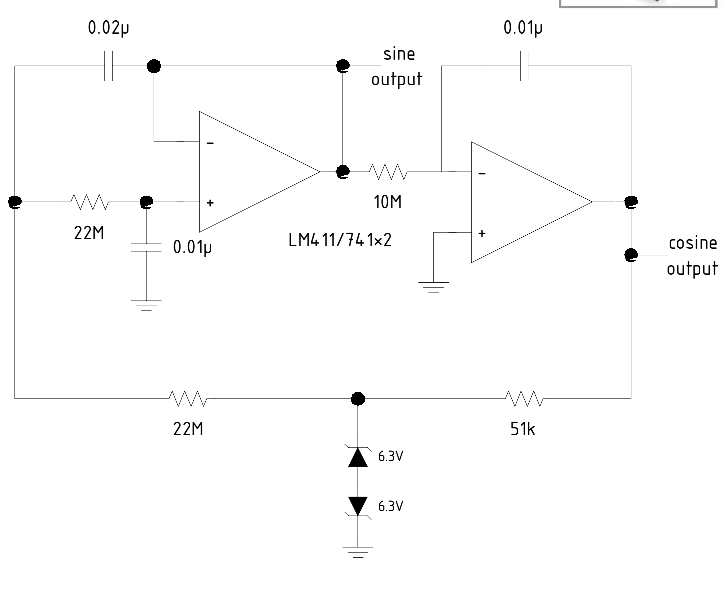
Face recognition system is a system that identifies taken or given pictures face matches or not with the faces from database. If the face is matched corresponding action is taken. Various techniques are available for face recognition process. Eigenface algorithm is faster, simple and efficient among various available algorithms.

This kind of project is applicable for variety of situations starting for security of home, offices, airports. Another approach is to automatically keep track of workers in office. When they come and leave the office can be tracked.

One limitation for eigenface approach is in the treatment of face images with varied facial expressions and with glasses. Also as images may have different illumination conditions. It is quite efficient and simple in preprocessing of image to verify the face geometry or the distances between the facial organs and its dimensions. The application of the symmetrization procedure improves significantly the Eigenface algorithm performance concerning images in unsuitable illumination conditions. ■

(Note: The writer is performing research on face recognition and its implementation issues.)

circuit ideas



1 Hz quadrature oscillator

(Source: Art of Electronics)