Facial Recognition Using Principal Component Analysis

Joey Bringley

Abstract— **This paper examines the Eigenface method used for facial recognition. It will explore related work in this area, detail the Principal Component Analysis algorithm and it's relation to face detection, and seek to discover the causes of failure or success within the method. An implementation of the algorithm was written in Matlab, and the results will be referenced throughout the paper.**

I. INTRODUCTION

The human brain is able to recognize faces easily. We do this, subconsciously, by examining a face, and extracting the particular features. Shape, texture, and complexion of the eyes, nose, mouth, cheeks, hair, etc all come into play when we try to recognize a person. However, things, like aging, lighting, distance, and other elements can make this more difficult for the brain. In short, recognizing a face is a complex and remarkable process.

Using algorithms to detect and identify faces is an important topic because of its many applications in various domains. Security, human computer interaction, and film processing are just some areas where these algorithms can be useful. While it is a difficult process for the human brain to identify faces it becomes even harder for a machine to do. As mentioned above, the human brain accomplished the task of facial recognition by examining distinct features of a face, and using all this information, it tries to identify the face. Interestingly enough, there is a algorithm already in place, that tries to accomplish this same task.

Facial recognition using Principal Component Analysis (also know as, the Eigenface method) is a widely studied and examined technique. First developed in 1987 and expanded in 1991 by two M.I.T. computer scientists, Matthew Turk and Alex Pentland. The development of the algorithm began with a idea of using PCA to find a low dimensional representation of images that could be linearly combined to reproduce the images in the original data set.

The PCA algorithm allows faces to be recognized efficiently and accurately. In this paper, I will detail this process, experiment with some of my own collected data, and hypothesize successes and failures of the algorithm.

II. LITERATURE REVIEW

A. Turk and Pentland (1991)

Matthew Turk and Alex Pentland were the first to fully develop the eigenface method, four years after the idea was suggested and examined by Sirovich and Kirby. Their paper, "Face Recognition Using Eigenfaces" is the first to detail the eigenface algorithm. Summarizing the method in four steps...

(i) Calculating eigenfaces from a training set, which characterizes the face space.
(ii) When a test image is received, project the image onto the face space, and calculate a weight vector for the image.
(iii) Determine if the image is a face, by checking its proximity to the face space.
(iv In the event that it is a face, use the weight vector to determine if the person is known or unknown.

Turk and Pentland found the method effective when the test images had subjects in front frontal view, but noted difficulties when dealing with images that differ from this description. Although the paper does explain the method behind choosing the k best eigenfaces, it fails to illustrate the process. Only stating to choose the eigen-vectors, "that have the largest eigenvalues, and which therefore account for the most variance within the set of face images." [1].

B. Neto, Jackson, And Somers (2004)

Selecting the number of principal components is tough problem with the algorithm, due to the fact that there is no definite answer. The paper entitled, "How Many Principal Components? Stopping Rules for determining the number of non-trivial axes visited" details the methods that can be used to solve the issue.

The paper explains that Kaiser's Rule, a method very popular among researchers and perhaps was used in Turk and Pentland's paper (and used in this project), is actually less effective than other available methods. The rule, which suggests keeping vectors whose corresponding eigen-values are great than 1, is based off

the idea that eigenvalues greater than 1 best summarize shared variation. However, this idea is often criticized because, "due to sampling variation, approximately one-half of sample eigenvalues from random data will exceed 1.0." [2]

The paper instead suggests different methods that are more accurate in selecting the best number of principal components, one being, *Random Average Under Permutation*. This idea involves randomization of the data matrix and computing the average eigen-value. Vectors whose corresponding eigen-values exceed the average random value are chosen as the principal components.

C. Rajesh and Sahu (2013)

In their paper, "Real Time Face Detection Under Different Conditions" Rajesh and Sahu implement the eigenface method used by Turk and Pentland in their original paper, with the goal to examine how the algorithm performs with variations to lighting and pose.

First the paper examines how a tilt in the subjects head can affect performance of the method. The data set consists of 3 images from 9 subjects where the head orientation differs in each picture. Two images are placed in the training data set, while 1 is placed in the test set. Immediately, the results show weakness in the algorithm. Of the 25 test images, 19 were correctly identified, 4 people were incorrectly identified, and 2 people were identified as an unclassified subject (when the subject was indeed classified).

Next, the same 9 subjects were used again. This time their pictures were taken with lighting variance. Of the 3 pictures, the two with low and high lighting were placed in the training set, while the mid level lighting photo was assigned to the test set. Once again, the algorithm displayed frailty. Of the 25 test images, 21 were correctly identified, 3 were falsely identified, and 1 was identified as not belonging to training set when it in fact was.

Lastly, the 9 subjects had their photos taken from varying scales (full, medium, small). The algorithm displayed the most poor results here. Only correctly identifying 13 of the 25 test images an falsely identifying the other 12.

The paper concluded that while the algorithm is easy to understand and implement, it is very sensitive to head scale, and only performs well under controlled conditions.

D. Orczyk and Porwik (2010

With some of the difficulties in mind that previous scientists had faced, Orczyk and Porwik set out to find preprocessing methods that could improve the algorithm's efficiency. The authors suggested using a color based skin filter to eliminate non skin colored pixels, thus removing the affect of background on the algorithm. In addition, the authors also tested another technique, generating each image but with 0-255 normalized hue value. This, they believed, would "minimize the face lighting impact on a recognition process." [3]

The results of the paper turned out very well. The authors found that the luminescence and hue pre-color filtering increased the accuracy of the algorithm from 70 percent to 90 and 95 percent respectively. As thought, the color based filtering diminished the affect of the background. The authors suggest in the summary that the method could be improved further through thermal technology, but notes a lack of funding to continue with this idea.

E. Steinhoff and Smith and Gallon and Francis(2013)

This paper, authored by 4 professors at Florida Memorial University investigates other preprocessing methods that could be used to improve the accuracy of the eigenface method. Knowing that the method is very sensitive to head tit and orientation (as shown in Rajesh and Sahu's paper) the authors suggested spatial transforms to center normalize the head orientations.

The paper, although a good idea in theory, was not able to find any solid results. The difficulties were caused by obtaining control points on the face, which due to different angles, proved very hard to obtain. While the approach can work under a controlled environment, it proves hard to implement in a real world setting.

III. THE ALGORITHM

Consider training set of *P* images where each image has dimension *m x n*. Each image in the training data set can be represented as a column vector of *m* x *n* rows, where the rows detail the features of an image. Therefore, the training dataset can be represented by a set the vectors *V* where $[v_1, v_2, v_3, ..., v_p] \epsilon V$. Next compute the mean vector of this training data set, which we can consider the "average" face, denoted by $\Psi = \frac{1}{P} \sum_{i=1}^{P} v_i$ We can then say that each face γ varies from the average face by...

$$\gamma_i = v_i - \Psi$$
$$i = 1,2,..., P$$

These normalized vectors in the training data set can now be represented by a matrix *A* having dimensions *(mxn) x P*.

PCA seeks to use eigen-vectors as a way to describe the data. To find these eigen-vectors we need the covariance matrix of A, denoted as *Cov(A)* = AA^T. However it is easy to see that this will result in a matrix of dimensions *(m x n)* x *(m x n)*. If each image is *100x100* pixels, the resulting covariance matrix will be *10,000* x *10,000* dimensions, requiring a large amount of computation and making the algorithm inefficient. Using linear algebra we can reduce the dimensionality of this matrix without losing information. The original definition of the covariance matrix is linearly related to the following covariance matrix *Cov(A)* = A^TA. Due to this fact, the eigen values of this new covariance matrix will be the same as the original, but with smaller dimensions. Even better, the number of eigen-vectors in the covariance matrix will be smaller, as well as the size of each vector.

The algorithm then returns the *k* best eigen-vectors such that $k < P$. These k best eigen-vectors should be able to explain a high amount of variance in the data. How large k must be so that any error is irrelevant is a question without a definite answer. Choosing a k too low results in a loss of relevant information, while a k too high can include unwanted noise from the data set. This project uses Kaiser's Rule, to find the correct number of principal components. A simple idea based on the fact that if variables are independent, their eigenvalue is equal to 1, therefore, Kaiser's Rule states to keep only vectors whose eigen-values are greater than 1.

Once the k best eigen-vectors (or eigen-faces) are chosen we can represent each face in the training data set by a linear combination of the eigen-faces. Thus every face in the training data set can be represented by a weight vector

$$\Omega_i = [\omega_1, \omega_2, \omega_3, ..., \omega_k]$$
(where i = 1, 2,..., P).

So what happens when we receive a new face? In this implementation of the eigenface method, the algorithm will again normalize the vector and project it onto the face space. The new face will then be assigned its own weight vector. Once this is done we will compare the input images weight vector with the weight vectors in the training data set using a distance metric called the Euclidian Distance. Denoted by...

$$D(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

Where a is the weight vector from the input image and b is a weight vector from the training data. We conclude that the weight vector *b* which minimizes the distance *D* from *a* is the matching image.

A. Methodology

For this project the above algorithm described was to be implemented in Matlab. While examining previous papers I observed that the algorithm performs incredibly well on a controlled data set. That is to say, a set of photos where the lighting, background, depth of field, and head orientation are kept the same. To verify this notion I obtained a copy of the the University of Essex Data Set, consisting of about 395 people with 20 pictures from each person. Features such as lighting, head orientation, background, and scale were kept constant for the most part, with just small variations in facial expression or lighting here and there. With the idea that I wanted to use my own data for the project, I randomly selected 10 individuals and 3 pictures for each to see how the algorithm would perform with a low number of samples. Of the 3 pictures for each individual, 2 were placed in the training set and 1 in the test set.

After I examined the first data set, I had 12 of my close friends send me three photos of their face. The participants were given no guidelines on how to take the photo, therefore, these features that were controlled in the University of Essex data set were not present here. A large variation in lighting, scale, and head orientation existed across the data set. The same ratio of test to training set images that was used for the Essex data set was used here as well.

As expected from reading others papers on the topic, the lack of uniformity across the data set would most likely lead to bad results. To try and combat this notion I applied a simple Gaussian filter to each image. The idea here is to consider the background of the face as noise, and Gaussian smoothing helps to eliminate this effect.

Fig. 1. Training Data Set with Gaussian Smoothing

IV. Results

The Eigenface method performed perfectly on the sample of the Essex data set. Of the 10 test images, all 10 were correctly identified.

For the data I collected myself, only 8 of of the 12 subjects were correctly identified. The Gaussian filter showed no effect for improving the accuracy of the algorithm, as again, only 8 of 12 matching were actually correct.

Fig. 2. Poor Results when Head Orientation Varies

V. Discussion

When all faces are located at the same position, with illumination kept and background kept constant, the algorithm performs accurately. However, when differences in lighting, pose, and depth of field are contained in the training data set, we see the method perform poorly. Upon inspection on the algorithm it can be seen that this poor performance is somewhat caused by the idea of computing the mean face of each image in the training set.

When variations in pose and depth of field are introduced, the faces in each image are not in the same position. A person holding the camera 2 feet away from their face versus someone holding it 6 inches away will produce completely different images. The first person will capture their face and the background, most likely their shirt, and whatever is behind them. For the second person however, the image contains only their face.

The problem with the eigenface method is that it does not singularly focus on the face in the image, but rather the entire image. For the two images described above, the mean vector computed will not resemble the actual average face between the two, but rather the average image. To consider the mean vector of the training data set as the average face is a poor choice in our case.

To explain further, in figure 2, you will see the algorithm identified one female subject as a male. Looking at the images from the 2 participants, the female seems to have taken the pictures at an arms length, exposing her dark black shirt in the bottom 1/4th of the image. The male subject she was identified as, took his picture from a different scale, but still exposed his dark shirt. Since both images contain these dark pixels in the bottom of the image, the method will consider these images similar, when in fact, they are very different. Obviously other features came into play here, but this coincidence shows how analyzing the whole image can lead to failure.

VI. Summary

The Eigenface approach to facial recognition is a fairly easy to grasp technique that is important to learn before going deeper into facial recognition. The algorithm shows weakness to variations in lighting, pose, and scaling due to it's nature but is efficient and accurate when dealing with constant features. The failures of the algorithm allow those studying it to look for ways of improvement, and thus learn more about image processing themselves.

References

[1] Turk M and Pentland A (1991) *Face Recognition Using Eigenfaces*. Vision and Modeling Group, The Media Laboratory Massachusetts Institute of Technology.

[2] Neto P and Jackson D and Somers K (2004) *How many principal components? stopping rules for determining the number of non-trivial axes revisited*, Computational Statistics and Data Analysis.

[3] Orczyk T and Porwik P (2010) *An Attempt to Improve Eigenface Algorithm Efficiency for Colour Images*, Journal of Medical Informatics and Technologies.

[4] Rajesh G and Sahu U (2013) *Real Time Face Recognition under Different Conditions*, International Journal of Advanced Research in Computer Science and Software Engineering.

[5] Steinhoff R and Smith C and Gallon A and Francis J (2013) *Improving Eigenface Face Recognition by Using Image Registration Preprocessing Methods*, Department of Computer Science and Mathematics, Florida Memorial University