Facial Recognition Using Eigenfaces

Grant Timmerman and Danny Vance

Math 136

May 20, 2013­

**Abstract**

This paper examines the algorithm of using eigenfaces in recognizing faces in static images. By performing a mathematical process called principal component analysis (PCA) on a large set of images depicting different faces, we can discover the core “ingredients” - the eigenfaces, derived from statistical analysis of many pictures of faces. Combined with the average, the set of eigenfaces allows us to compose any human face. Remarkably, it does not take many eigenfaces combined together to achieve a fair approximation of most faces. Also, since a person’s face can be recorded with a spanning set of eigenfaces plus the average face, rather than recording each individual pixel of each face, much less space is needed for each person’s face. In other word, this algorithm can be applied in image data compression. This paper aims to determine the effectiveness of this technique with regards to accuracy, computational speed, and data compressions as well as to consider its drawbacks that make the technique ineffective in recognizing faces.

**Key Words:** Eigenfaces, Eigenvectors, Principal Component Analysis (PCA)

**1 Introduction**

Human face recognition is a difficult problem in computer vision. Human faces are complex, natural objects that tend to not have easily identifiable edges and features. Because of this, it is difficult to develop a mathematical model of the face that can be used as prior knowledge when analyzing a particular image.

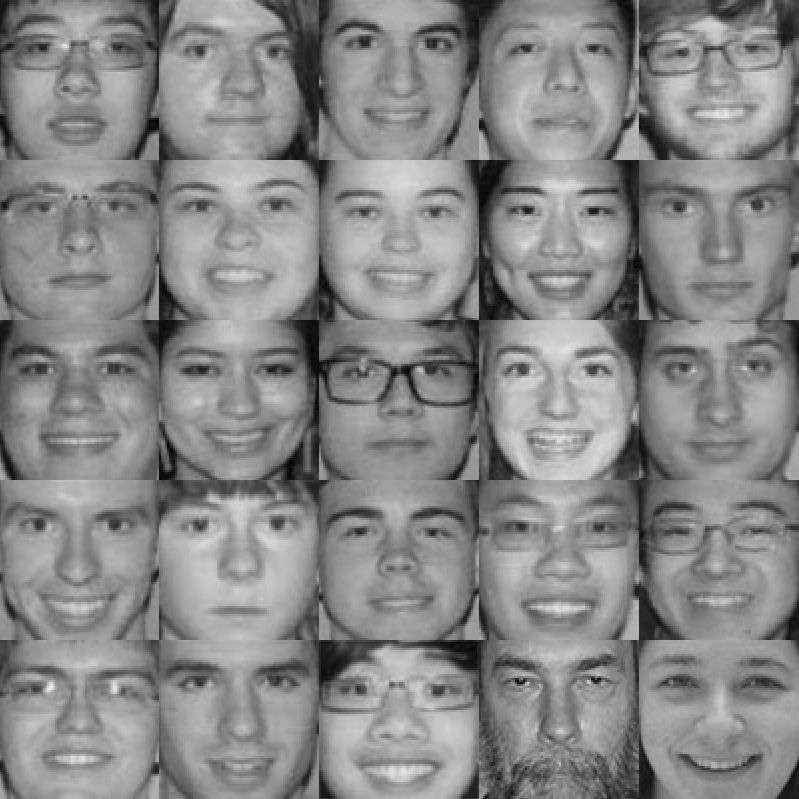
Although facial recognition is a high level visual problem, there is quite a bit of structured information that we can take advantage of. By finding the most relevant pieces of information in a group of faces, we can characterize each face into these important features, which are called “eigenfaces”. The process of recognizing a face is achieved through projecting the face’s image into the subspace spanned by the eigenfaces (“face space”) and then classifying the face by comparing its position in face space with the positions of known individuals.

**2 Procedure**

**2.1 Eigenface Initialization**

**2.1.1 Setup images**

Acquire an initial set of face images (the training set) in grayscale, cropped to only contain the face, and then rescaled to set dimensions. The photos must be centered and of the same size.



**Figure 1:** The face images we used as our training set.

**2.1.2 Compute the Average Face**

If the photos in the training set are each by *n*, we can represent each photo as a flattened by 1 matrix (we’ll call it ) that stores values for the brightness of each pixel in one very large list. The average face is simply the arithmetic average of all of our image vectors:

Here is the result from our training set:

****

**Figure 2:** The average face of the training set.

**2.1.3 Find Difference Vectors**

Subtract the average face vector from each photo vector to get a difference vector: where is the *i*th difference vector.

We can align all of our difference vectors into an by matrix where and each difference vector represents the th column of .



**Figure 3:** The difference faces of the training set

As you can see, the luminosity of the pictures severely affects the difference vectors. Also note that Figure 3 does not fully represent the difference vectors because some values are negative yet are represented as black in this picture.

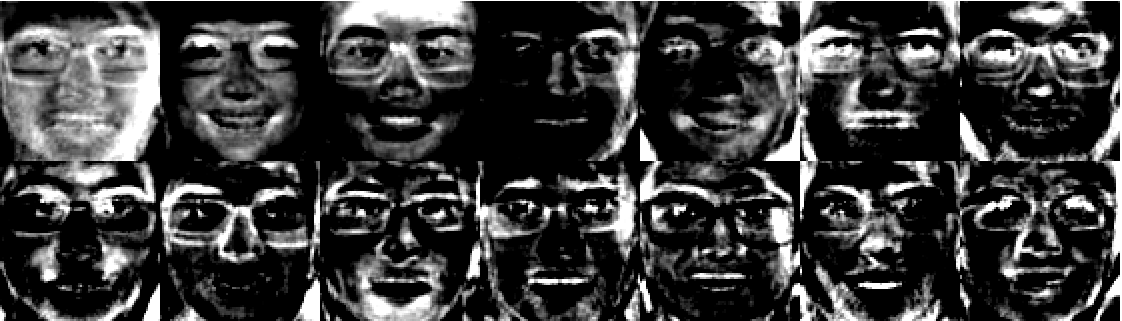
**2.1.4 Finding the Eigenfaces**

We can compute the covariance matrix for our data as follows:

The scalar value

The set of difference vectors are very large and too computationally intensive if used. This is where principal component analysis comes in. We seek a set of M orthonormal vectors and their associated eigenvalues , which best describes the data. These are the eigenvectors and eigenvalues for the covariance matrix defined by the following.

However the matrix is by , which is infeasible for large image sizes. Fortunately we can just compute the eigenvectors by first solving the much smaller by matrix and taking linear combinations of the resulting vectors. Here are the results of finding the eigenvalues for the smaller matrix:



**Figure 4:** The eigenfaces of the training set. The most important facial component is on the top left and the least important facial component is on the bottom right.

Do note that the eigenfaces shown above are actually almost completely black and we have scaled the values by 5000%. Also note that the eigenvectors can have both negative and positive values so the visual representation is not fully representative in this visual model.

**2.2 Eigenface Recognition**

**2.2.1 Project a photo to the face space**

Calculate the corresponding distribution in M-dimensional weight space for each known individual, by projecting their face images onto the “face space.”

1. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the eigenfaces
2. Determine if the image is a face by checking if the original input image is sufficiently close to its respective “free space” image.

**2.2 Another Subtopic!**

Here is more info!

Project picture, find % match to original picture

Verify picture is a face

FB Picture % match

Face that is blocked off

Include graphs

-number of eigenfaces to accuracy

-fb picture to % match of person

**3 Drawbacks**

sd

* Different levels
* Different angles
* Smiles
* Hair
* size

Center equations every so often if they are important or big. You can add a number or star (\*) if the equation or expression is referenced in the document.

**4 Summary**

Citations are important. Cite them like this. Look at the bottom of the page to see the note[[1]](#footnote-1).

**References**

[1] A book here in APA style

[2] Another book here

http://www.academia.edu/286650/FACE\_RECOGNITION\_USING\_EIGENFACE\_APPROACH

1. Here is the foot note. See [1] and [2]. [↑](#footnote-ref-1)