Analysis of Eigenface and Correlation Methods of Facial Recognition

Mary M. Keenan

I. Introduction

The first facial recognition software was developed in 1964 [1]. Since then, it has become increasingly accurate to the point where now some algorithms are better at recognizing faces than humans [1].

Fig. 2. Second most prominent eigenface in the 2015-16 QEA class. It vaguely resembles a woman, which makes sense given almost half the class is female.

Facial recognition software is used to identify people in pictures. It uses algorithms to compare and match the patterns in a new face to an existing face and it returns the identity of the closest match -- in other words, it matches a new picture to a picture in the database, and the person in the new picture is (theoretically) the person in the database picture [1].

All facial recognition methods require training sets -- large databases of pictures of individuals, sometimes under various conditions (lighting, orientation, etcetera) [2]. Training sets can be used to teach a facial recognition program what's important or common in faces. They're used when testing facial recognition software with new faces -- the programs compare the new faces to the training set faces [2]. The pictures used to test a program are called a testing set.

There are a number of challenges involved with facial recognition. Many methods of facial recognition require huge training sets to compare new pictures against, which can be very memory-intensive and computationally expensive if the pictures aren't condensed in some way [2]. This makes it difficult to do real-time facial recognition. The accuracy of some algorithms can also be heavily affected by changes in lighting, position, or expression, and training programs with training sets can take time if the process isn't automatic [2]. These challenges are addressed differently by the various methods of facial recognition.

In this paper, we compare the advantages and disadvantages of the Eigenface and Correlation methods of facial recognition. We examine everything from computational speed to required memory, to accuracy under different conditions. We start by introducing the basic concepts behind each method, then begin working through the math. We briefly touch on the theoretical pros and cons of each method, before moving on to the implementation and testing of the two programs. Finally, we compare the advantages and disadvantages of the Eigenface and Correlation programs we wrote and tested.

There is a tradeoff between accuracy and efficiency in the Eigenface method of facial recognition. The more efficient the program is – that is, the higher the computational speed and lower the memory-intensiveness of the program – the less accurate it becomes. The Correlation method is less dynamic in the sense that it is always relatively computationally slow and memory-intensive, but it is also always equally or more accurate than the Eigenface method under any condition.

II. Algorithms and Justification

A. Eigenface Concept

Eigenface is the name for an eigenvector that describes a face pattern [3]. The eigenfaces of a set of images are the most common patterns found in that set of faces -- they're a set of "standardized face ingredients." [3] Figure 1 shows two reshaped eigenfaces taken from the set of eigenfaces that describe the faces of the 2015-16 QEA class.

Fig. 1. These are two eigenfaces of the 2015-16 QEA class. The left image shows the most prominent class eigenface, and the right image shows one of the least prominent eigenfaces.

The eigenface on the left of Figure 1 is associated with the highest eigenvalue of the class's covariance matrix, which means it's the most prominent face-pattern for the class. The eigenface on the right of Figure 1 corresponds with one of the lowest eigenvalues, which means it's one of the least prominent face-patterns. Looking at the eigenfaces, this makes sense. The left eigenface, the more common face-pattern, looks like a blurred photo of an average looking man, whereas the right eigenface, the less common face-pattern, looks like an etching of an odd sex-less face with a faint outline of glasses.

There are slightly more men than women in the QEA training set, so that explains why the most prominent face-pattern in the class looks like a man. The second most prominent eigenface in the class, pictured in Figure 2, looks more like a woman.

B. Eigenface Math

Covariance matrices, Principle Component Analysis (PCA), Singular Value Decomposition (SVD), and correlation matrices are used to calculate and compare eigenfaces. For now, the training set consists of all 344 of the class pictures.

First, we need to reshape the class pictures so that each picture is a column in a massive face matrix. Each picture is 360 x 256 pixels and there are 344 pictures, so the face matrix will be 92160 x 344 pixels. We need to subtract the mean of each face (each column) from that column to standardize the training set. Next, we need to find the covariance matrix that represents the class pictures.

Fig. 3. (left) Reconstructed face using only the first 100 of 344 eigenfaces, (center) reconstructed face using the first 200 of 344 eigenfaces, (right) original picture; increasing the number of eigenfaces increases the quality of the reconstructed picture

Covariance Matrix

Covariance is a measure of how similarly two variables vary [4]. If the variables increase and decrease at similar rates and intervals, then the covariance will be positive, and if the opposite is true, the covariance will be negative [4]. We want to create a covariance matrix, a matrix that contains the covariance values for each pixel relative to every other pixel in the face matrix. To do this, we use the following formula, where A is the face matrix and N is the length of the columns in the face matrix.

$$R = \frac{1}{\sqrt{N-1}} * A \quad (1)$$

Now, we have a choice. If we want to know how every pixel corresponds to every other pixel in the training set, we multiply R by the transpose of R, as shown in Equation 2.

$$B = R * R^T \quad (2)$$

If we want to know how each picture as a whole corresponds to every other picture, we multiple the transpose of R by R (Equation 3).

$$D = R^T * R \quad (3)$$

Eigenfaces are the eigenvectors of the matrix B in Equation 2, but matrix C is a massive matrix, 92160 x 92160, and finding its eigenvectors would be computationally expensive. Matrix D in Equation 3 has more manageable dimensions – 344 x 344, but its eigenvectors aren’t the ones we need. As we’ll see in the next subsection, matrix B and D have a special relationship which allows us to get around this computational obstacle.

Singular Value Decomposition

To find the eigenfaces, we use Singular Value Decomposition (Equation 4), where R (92160 x 344) is the covariance matrix, U (92160 x 92160) holds the eigenvectors of Equation 2, Σ (344 x 344) consists of the eigenvalues of the covariance matrix, and V (344 x 344) is the eigenvectors of Equation 3.

$$R = U * \Sigma * V^T \quad (4)$$

As mentioned in the previous subsection, it’s easy to calculate V^T using Principle Component Analysis, but it’s very difficult to do so for U, because finding the eigenvectors of a 92160 x 92160 matrix would take considerable time and resources. However, multiplying matrix R by matrix V^T produces an approximation of matrix U.

$$U = R * V^T \quad (5)$$

Multiplying a 92160 x 344 matrix and a 344 x 344 matrix is a much quicker solution than finding the eigenvectors of a 92160 x 92160 matrix. However, it’s good to note that the eigenvectors in U are not normalized and should be before continuing.

Now, we have our eigenfaces in matrix U. Next, we need to find the linear combinations that relate the pictures to the eigenfaces.

Linear Combination

Using the calculated eigenfaces (matrix U) as the basis, you can represent faces with a linear combination by multiplying the transposed eigenface matrix and face matrix (Equation 6).

$$C = U' * A \quad (6)$$

This is the equivalent of taking the dot product, and it finds how much each face projects on each eigenfaces. Matrix C will be 344 x 344, assuming all of the eigenfaces in the class training set are being used.

Finding the linear combination of the pictures and eigenfaces makes it easier to store pictures in the training set, because they can be represented as a list of linear combination coefficients instead of a list of every pixel in the picture. For example, if a training set has 100 pictures with 360 x 256 pixel resolution, the pictures’ pixels can be stored in a 92160 x 100 matrix or the pictures’ linear combination coefficients could be stored in a 100 x 100 matrix. The picture can be reconstructed using the coefficients and eigenface matrix, and if there are enough eigenfaces in the basis set, the reconstructed picture will look very similar to the original.

A set of n pictures will have n eigenfaces; however, not all of the eigenfaces in a set of pictures are necessary to reconstruct the faces to a point where they’re recognizable, and working with fewer eigenfaces for facial recognition is less memory-intensive and computationally expensive. The tradeoff is that reducing the number of eigenfaces used for linear combination does reduce accuracy.

Figure 3 shows three versions of the same picture with the original on the right and two reconstructed images, one with 100 eigenfaces and one with 200 eigenfaces -- the quality of the reconstructed pictures increases as the number of eigenfaces used is increased. However, the picture is easily recognizable when 200 out of 344 eigenfaces are used.

Correlation Matrix

At this point, you can represent your training set with two matrices: a matrix of linear combination coefficients and a matrix of eigenfaces. To identify the people in the testing set, the linear combination coefficients for the new pictures are calculated and then compared to the coefficients of the pictures in the training set using a correlation matrix. The program returns the closest match and assuming the person in the testing image has a picture in the training set, the closest match should be another picture of that person.

Correlation is a measure of how similar two things are. In a correlation matrix, the correlation coefficient of two pictures i and j is located at position (i,j) in the matrix. We use the following equation (7) to find the correlation between the linear combination coefficients of a new picture and each picture in the training set. The formula to find E is shown separately in Equation 8 for the sake of space and clarity. In this case, E consists of the adjusted linear combination coefficients for two pictures we want to compare.

$$S = \frac{1}{N-1} * E^T * E \quad (7)$$

$$E = \begin{bmatrix} \frac{x_1 - \mu_x}{\sigma_x} & \frac{y_1 - \mu_y}{\sigma_y} \\ \frac{x_i - \mu_x}{\sigma_x} & \frac{y_j - \mu_y}{\sigma_y} \end{bmatrix} \quad (8)$$

The Eigenface method uses correlation to compare the linear combination coefficients of pictures in the training set and pictures in the testing set to make matches. The Pixel-by-Pixel Correlation approach uses similar math.

C. Pixel-by-Pixel Correlation

Pixel-by-pixel correlation is a much simpler way of doing facial recognition. It finds the correlation between each pixel in the testing set picture and each pixel in every other picture in the training set. It returns the closest match. This method doesn't condense pictures, so the training set is saved as a 92160 x n matrix, where n is the number of pictures in the training set, which is very memory-intensive.

The math behind Pixel-by-Pixel Correlation also uses a correlation matrix (Equations 7 and 8), but on a much larger scale. Instead of comparing linear combination coefficients, it compares each pixel in the picture. This means the size of the correlation matrix is much larger and the method is much slower, because we're finding the correlation between two 92160x1 matrices instead of two 344x1 matrices.

III. Comparison of Performance

When all of the eigenfaces in the training set are used, the Eigenface method is as accurate as the Correlation method [2]. The associated benefits of the Eigenface approach, however, are negated. Interestingly enough, removing the first three eigenfaces (the most prominent) has been shown to increase the accuracy of the eigenface method under variable lighting conditions, because the first three eigenfaces tend to have more to do with lighting than specific features [2], as Figure 1 shows.

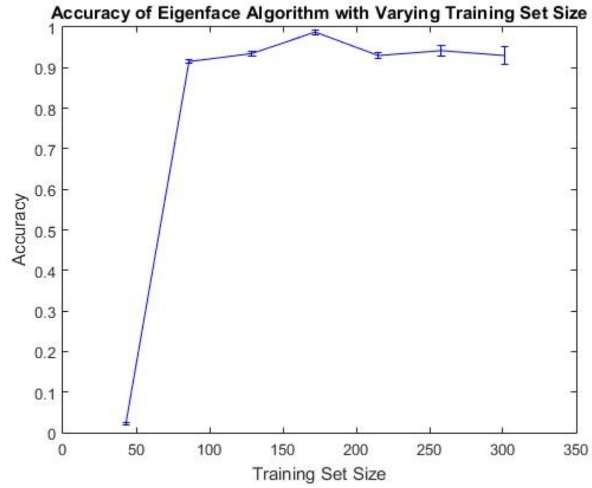

Fig. 4. As the number of pictures in the training set increases, so does the accuracy of the program (it should only ever increase, in the cases where it's not, there wasn't a large enough testing set to test it well).

Both methods tend to do badly at recognizing people under variable lighting conditions, because neither method looks at the facial features relative to each other – they both look at the whole picture. If someone shifts to the left or bleaches their hair, the Eigenface and Correlation methods don't know how to handle this other than to match the new person to the training set person who's in the closest position or has hair that's closest in color, respectively.

However, this is less true for changes in expression. People often look more like themselves than anyone else even when they're making a different face. Their hair and coloring is the same, and both the Eigenface and Correlation programs perform well in this scenario.

We tested the proficiency of the methods under both conditions. The class training set contains 344 pictures, 8 pictures per person for 43 people. Each person has pictures with different expressions in the training set and 32 out of 43 people took a ninth picture at a different time with different lighting.

For the test of how accurate the programs were at recognizing people with different expressions under the same lighting, we used only two pictures per person for the training set and used the other 258 pictures as the testing set.

When we tested how accurate the programs were at recognizing people under different lighting conditions, we used all 344 pictures of the class training set as the training set and used the 32 new pictures as the testing set. Increasing the training set size past 86 does little to improve accuracy (as you'll see later in Figure 4), but since the programs do so poorly under variable lighting conditions, it seemed like they needed every advantage, no matter how small.

When the lighting is consistent but expressions vary, both methods perform well (92%), and the eigenface method without the first three principal components (most prominent eigenfaces) performs even better (93%). However, when the lighting varies, both eigenface methods and the correlation method do very poorly as Table 1 shows. When the lighting varies, the accuracy of the methods drops about 70%, from 92% to 22%. This is line with what we expected – the Eigenface and Correlation methods look at the whole picture, not the facial features. If the testing picture as a whole is different – in brightness, for example, neither method will perform well, but when there are only slight changes, like expression, both methods are very accurate.

Table 1: Accuracy of Eigenface and Correlation under Different Conditions

Method	Accuracy	
	Different Lighting (with training set size 344)	Different Expressions (with training set size 86)
Eigenface	.219±.031 (7/32)	.915±.004 (236/258)
Eigenface w/o first 3	.156±.031 (5/32)	.930±.004 (240/258)
Correlation	.219±.031 (7/32)	.919±.004 (237/258)

As we briefly touched upon earlier, the performance of the eigenface method improves as the training set size increases, but a training set size of 86 out of 344 images is sufficient in this case to achieve accuracy above 90%, as Figure 4 demonstrates. Past a training size of 86, the effects of incrementally increasing the training size decreases dramatically.

The biggest advantage of the Eigenface method is that it is computationally faster and less memory-intensive, but this is only true when fewer eigenfaces than pictures in the training set are used. As fewer eigenfaces are used, the accuracy of the program decreases but its speed increases and the memory needed decreases. Table 2 supports this – as the number of eigenfaces used to test the Eigenface method under the variable expressions condition decreased, the memory and time needed to run the program also deceased.

Table 2: Computational Speed and Memory Required for Eigenface and Correlation Methods

Method	Time (seconds)	Memory (Mb)
Eigenface w/ 86	175.2	63.47
Eigenface w/ 60	167.0	45.02
Eigenface w/ 40	157.4	30.26
Correlation	179.1	63.41

Figure 3, from the previous section shows how the quality of the reconstruction of a picture is impacted by the number of eigenfaces being used – the quality decreases as the number of eigenfaces decreases. The accuracy of the program also decreases as the number of eigenfaces used decreases, as shown in Figure 5, but only 30 out of 86 eigenfaces are needed to achieve above 90% for accuracy. This is the beauty of the Eigenface approach to facial recognition. It's not the most accurate method, but it's fast and it doesn't need a lot of memory. Using 40 out of 86 eigenfaces, the Eigenface method is above 90% for accuracy (compared to 92% for the Correlation method), runs 12% faster than the Correlation method, and uses less than half the memory of the Correlation method.

IV. Conclusion

The eigenface algorithm is simple, fast, and non-memory-intensive even when facial expressions differ. However, it is highly inaccurate when the lighting conditions change and its benefits regarding computational efficiency and memory-intensiveness are negated when the number of eigenfaces is equal to the number of pictures in the training set. The eigenface method's tradeoff for efficiency is accuracy, but the program can still be very accurate with only 30 out of 86 eigenfaces.

The correlation method is more or equally accurate than the eigenface method, depending on the number of eigenfaces used, but it is computationally expensive and memory-intensive, and it does poorly at handling changes in lighting.

Next steps include investigating other approaches to facial recognition, including but not limited to the Fisherface and kernel methods.

References

[1] "Facial Recognition System." *Wikipedia*. Wikimedia Foundation, n.d. Web. 02 May 2016.

[2] Belhumeur, Peter N., Joao P. Hespanha, and David A. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection." *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* 19.7 (1997): n. pag. *The Fu Foundation School of Engineering and Applied Science*. Columbia University. Web. 2 May 2016.

[3] "Eigenface." *Wikipedia*. Wikimedia Foundation, n.d. Web. 02 May 2016.

[4] "Covariance." *Wikipedia*. Wikimedia Foundation, n.d. Web. 02 May 2016.

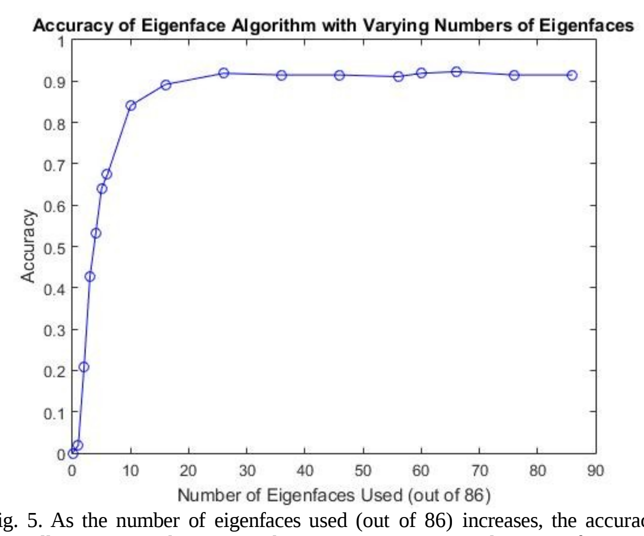

Fig. 5. As the number of eigenfaces used (out of 86) increases, the accuracy generally increases. The program hits 90% accuracy around 30 eigenfaces, and past that, increasing the number of eigenfaces does little to improve accuracy.