Eigenface for face detection

Christopher Kline

Department of Mathematics

Georgia College

Christopher.kline@bobcats.gcsu.edu

Under the supervision of

Dr. Simplice Tchamna.

Abstract

The Eigenface approach is considered by many to be the first working facial recognition technology, and it served as the basis for one of the top commercial face recognition technology products. Eigenface refers to an appearance-based approach to face recognition that seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic (as opposed to a part or feature-based) manner. For this work, we have shown that by using properties and rules of mathematics, in particular matrices, we can manipulate a set of images (matrices) in such a way as to extract the relevant facial information necessary for detection and recognition.

Keywords: Matrix, Eigenvalues, Eigenvectors,

1 Introduction

Due to the practical importance of the topic and theoretical interest from scientists, facial recognition software is almost as old as computer vision itself. Some of the key reasons face recognition has always remained a major focus of research, even though other forms of recognition are capable of being more accurate, is because of its non-invasive nature and because it is people's primary method of personal identification.

Perhaps the most famous early example of a face recognition system is due to Kohonen, who demonstrated that a simple neural net could perform face recognition for aligned and normalized face images. The type of network he employed computed a face description by approximating the eigenvectors of the face image's autocorrelation matrix; these eigenvectors are now known as `eigenfaces.' Kohonen's system was not a practical success, however, because of the need for precise alignment and normalization. In following years, many researchers tried face recognition schemes based on other neural net approaches such as edges. While several were successful on small databases of aligned images, none successfully addressed the more realistic problem of large databases where the location and scale of the face is unknown.

The method for using principal components to represent human faces was first developed by Lawrence Sirovich and Michael Kirby in 1987. Kirby and Sirovich later introduced an algebraic manipulation which made it easy to directly calculate the eigenfaces and showed that fewer than 100 images were required to accurately code carefully aligned and normalized face images. The use for this method came about when Kirby and Sirovich began applying linear algebra to the problem of facial recognition and realized that problems arose when performing recognition in a high-dimensional space. However, it wasn’t until four years later that Matthew Turk and Alex Pentland first used this idea for face detection and recognition. This method would later be referred to as the Eigenface approach.

Turk and Pentland hypothesized that significant improvements could be achieved by first mapping the data into a lower dimensionality space. Turk and Pentland then demonstrated that the residual error when coding using the eigenfaces could be used both to detect faces in cluttered natural imagery, and to determine the precise location and scale of faces in an image. They then demonstrated that by coupling this method for detecting and localizing faces with the eigenface recognition method, one could achieve reliable, real-time recognition of faces in a minimally constrained environment. This demonstration that simple, real-time pattern recognition techniques could be combined to create a useful system sparked an explosion of interest in the topic of face recognition. Therefore, the goal of this paper is to demonstrate the process required to find the eigenfaces of a set of images as well as to show how these eigenfaces are used to detect faces in a set of images.

2 Background

2.1 Definition:

Let M be an n-by-n matrix. A real number λ is called eigenvalue for the matrix M if there exists a nonzero vector v such that $Mv = \lambda v$. This is equivalent to $(M - \lambda I_n)v = 0$, where I_n represents the identity matrix. In other words, the equation $M - \lambda I_n = 0$ has at least one solution in $\mathbb{R}^n$.

2.2 Remark:

Each eigenvalue of a matrix M is a solution of the equation $\det(M - XI_n) = 0$, where X is the unknown.

2.2 Example:

Consider the following matrix

$$M = \begin{pmatrix} 8 & 0 & 8 & 0 \\ 10 & 4 & 2 & 0 \\ 3 & 0 & 3 & 10 \\ 5 & 0 & 5 & 6 \end{pmatrix}$$

For each real number λ, we have $\det(M - \lambda I_4) = \lambda^4 - 21\lambda^3 + 84\lambda^2 - 64\lambda$. The equation $\lambda^4 - 21\lambda^3 + 84\lambda^2 - 64\lambda = 0$ has 4 solutions $\lambda_1 = 0,\ \lambda_2 = 1\ ,\ \lambda_3 = 4,\ \ \lambda_4 = 16$. Thus $0,\ 1,\ 4,\ 16$ are eigenvalues of the matrix M

2.3 Definition:

The transpose of a matrix $\boldsymbol{M}$, denoted M^t, is the matrix obtained by interchanging the rows and columns of $\boldsymbol{M}$

2.4 Example:

If $\boldsymbol{M} = \begin{pmatrix} \mathbf{3} & \mathbf{-1} & \mathbf{0} \\ \mathbf{2} & \mathbf{1} & \mathbf{-2} \\ \mathbf{4} & \mathbf{6} & \mathbf{11} \end{pmatrix}$, then $\boldsymbol{M^t} = \begin{pmatrix} \mathbf{3} & \mathbf{2} & \mathbf{4} \\ \mathbf{-1} & \mathbf{1} & \mathbf{6} \\ \mathbf{0} & \mathbf{-2} & \mathbf{11} \end{pmatrix}$

2.5 Properties of the transpose of a matrix

Let $M,\ N$ be two n-by-n matrices. Then

1. $(M^t)^t = M$
2. $(MN)^t = N^t M^t$
3. $(M + N)^t = M^t + N^t$
4. $\det(M^t) = \det M$

Proof:

Let $M = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{pmatrix}$ and $N = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kk} \end{pmatrix}$

1. The (i,j)-entry of M^t is the (j,i)-entry of M. Hence the (i,j)-entry of $(M^t)^t$ is the (j,i)-entry of M^t which is the (i,j)-entry of M. Therefore, all entries of $(M^t)^t$coincide with the corresponding entries of M, so these two matrices are equal.

2. The (i,j)-entry on MN can be written as

$$(MN)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Moreover, when we transpose a matrix, we switch the rows and the columns. Therefore,

$$\left(\left(MN \right)^t \right)_{ij} = \left(MN \right)_{ji} = \sum_{k=1}^{n} a_{jk} b_{ki}$$

But

$$\left(N^t M^t \right)_{ij} = \sum_{k=1}^{n} \left(N^t \right)_{ik} \left(M^t \right)_{kj} = \sum_{k=1}^{n} b_{ki} a_{jk}$$

This shows that $\left(MN \right)^t = N^t M^t$

3. The (i,j)-entry of A^T+B^T is the sum of $(i,j)-$entries of A^T *and* B^T, which are (j,i)-entries of A and B, respectively. Thus the (i,j)-entry of A^T+B^T is the (j,i)-entry of the sum of A and B, which is equal to the (i,j)-entry of the transpose $(A+B)^T$.

4. To show that $\det(M^t) = \det M$ for any $n \times n$ matrix, we proceed by induction on n. For the case $n = 1$, it is clear that $M = M^t$. Thus $\det(M^t) = \det M$.

Now suppose that the result is true for any $n = k - 1$ and let M be a $k \times k$ matrix. Write

$$M = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{pmatrix}$$

Then

$$M^t = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{k1} \\ a_{12} & a_{22} & \cdots & a_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1k} & a_{2k} & \cdots & a_{kk} \end{pmatrix}$$

Using the cofactor expansion formula, we have

$$\det M = a_{11} \det M_{11} - a_{21} \det M_{21} + \cdots + (-1)^{k+1} a_{k1} \det M_{k1}$$

where M_{ij} is the matrix obtained from M by removing the i^{th}row and the j^{th}column. But the cofactor expansion of M^t is

$$\det M^t = a_{11} \det(M^t)_{11} - a_{21} \det(M^t)_{12} + \cdots + (-1)^{k+1} a_{k1} \det(M^t)_{1k}$$

But $(M^t)_{ij} = \left(M_{ji}\right)^t$. Since M_{ji} is a $(k-1) \times (k-1)$ matrix, we can use the induction hypothesis to have

$$\det(M^t)_{ij} = \det\left((M_{ji})^t\right) = \det M_{ji}$$

It follows that

$$\det M^t = \det M$$

2.6 Theorem:

A matrix $\boldsymbol{M}$ and its transpose $\boldsymbol{M^t}$ have the same eigenvalues.

Proof: Let M be $n \times n$ matrix. Then

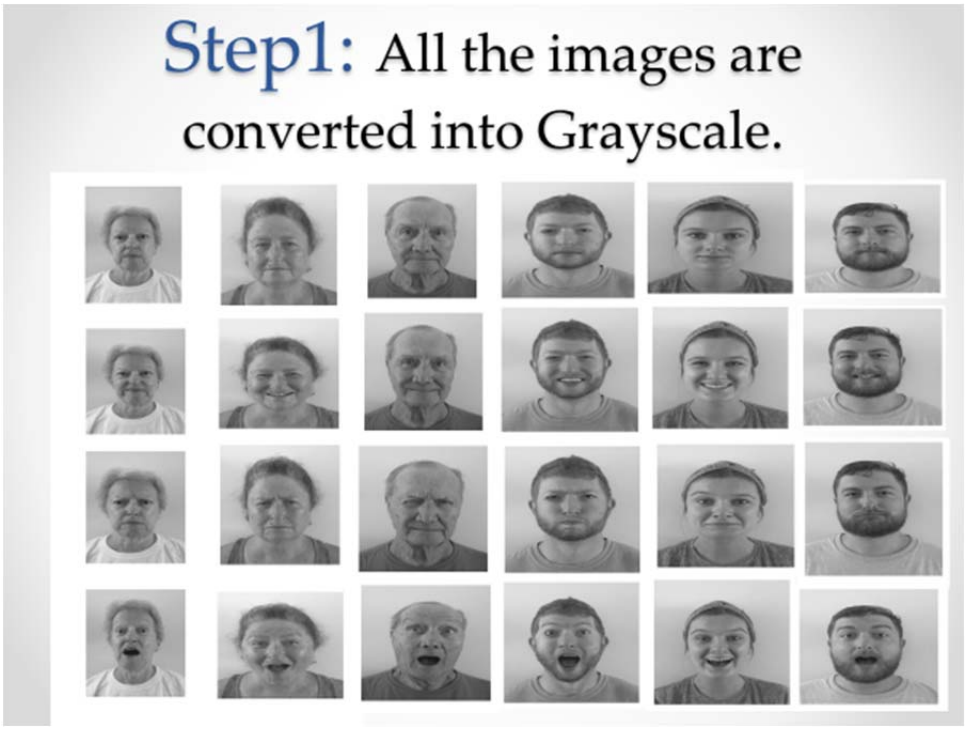
$$\det(M - XI_n) = \det(M - XI_n)^t = \det(M^t - (XI_n)^t) = \det(M^t - XI_n)$$

3 Eigenface for face detection.

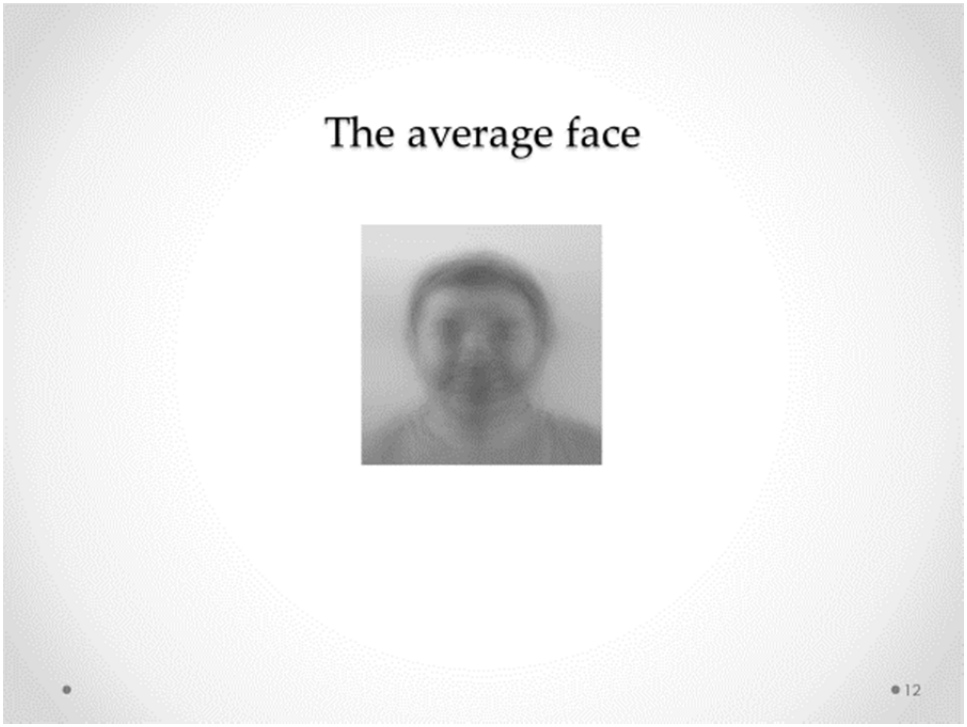
The notion of eigenface was first used by Matthew Turk and Alex Pentland in 1991, when they discovered how to detect faces within images. The method is relatively simple compared to other methods such as fisherface or Local Binary Patterns Histograms (LBPH).

We start with a set of images, denoted by P_i $(1 \leq i \leq 24)$, called training faces (or data base). For my demonstration I gathered the 24 images from a group of 6 individuals. Each individual took 4 different pictures changing their facial expression for each of the four images. The reason this is a necessary procedure is to help with the later step of finding the Eigenvectors associated with the faces. Because we are using the Eigen face approach we needed to convert our images to grayscale, which is done with a simple code in Matlab.

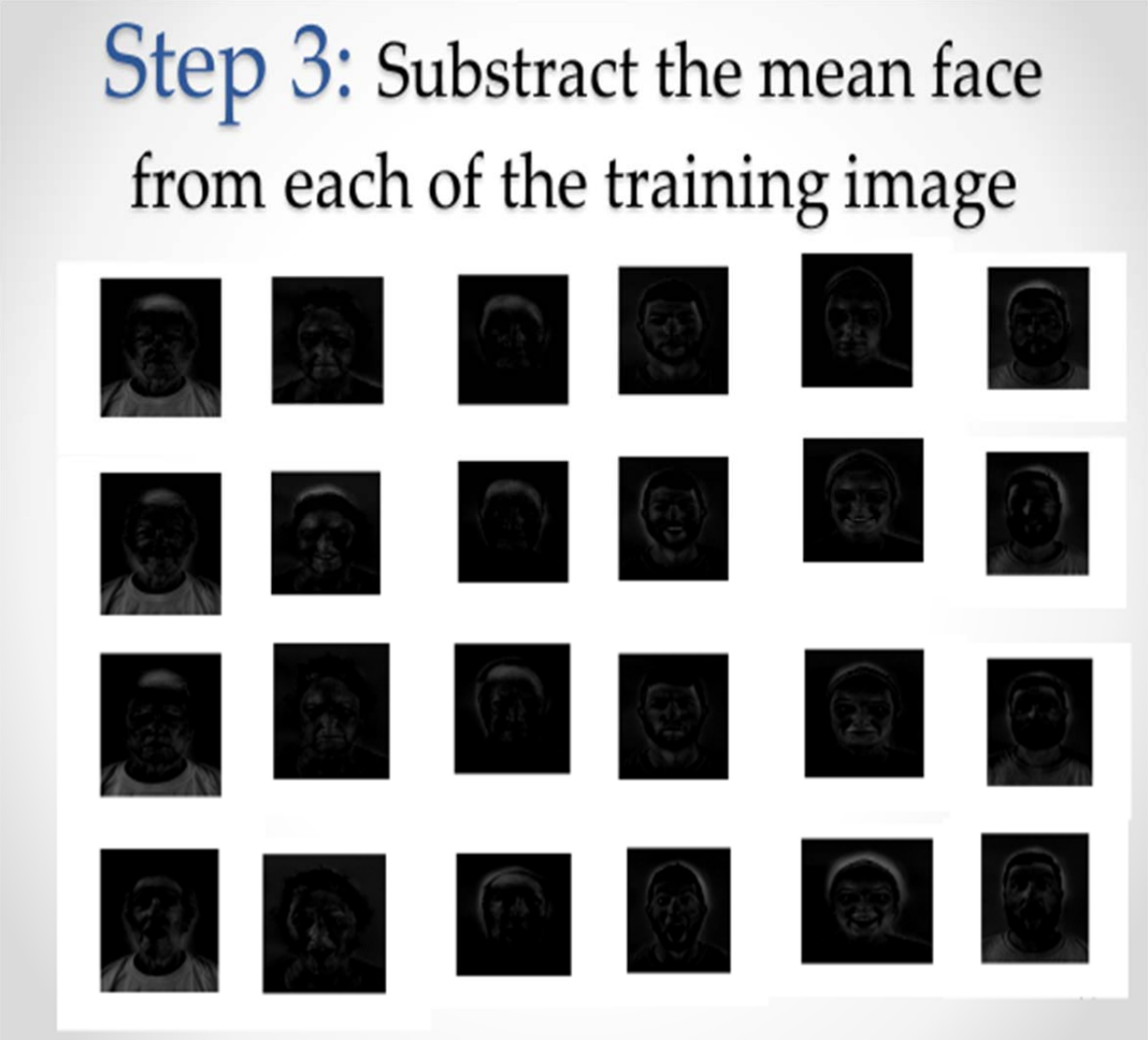

Training faces (Database)



The second step in our process is to find the average matrix relative to our data base. In my program each image is represented by a matrix $M_i\,(1 \leq i \leq 24)$. From this we create a summation formula to calculate the average matrix of the data base matrices, which is denoted by Ψ. The formula for the average is $\Psi = \frac{1}{24}\sum_1^{24} M_i$. The image associated to the average matrix is called the average face, which can be found below.


The average face
12

Once we have acquired an average face we can move on to the third step of our procedure which is to subtract the average face from each of the existing faces in our grayscale data base. The formula associated with this step is $Q_i = M_i - \Psi$. This process is done to each of the individual images to obtain the following 24 images. What this step does is that it shows how each of the original individual images differs from the relative average of the set of images. This is a highly important condition necessary for finding the Eigen faces.



After this comes our fourth step, which is to take each of the Q_i images and find its transpose. Recall that the transpose of a matrix is the matrix obtained by interchanging the rows and the columns. This operation thus corresponds to a counter clock wise rotation of the image associated to the matrix Q_i as well as a 180^o flip of the image. For example, the transpose of the image below,

is this image,

The best way to see how the transpose affects this image in particular is to focus on the way the hair slants across the forehead. You can see that the image is not simply rotated counterclockwise, but that it also has the 180^o as mentioned previously.

The next step, our fifth step, is to find the covariant image using the two previous steps. The covariant image of the set of 24 images is the matrix defined by,

$$C = (1/24) \sum_{i=1}^{24} Q_i \; Q_i^T .$$

Note that each of the images Q_i and Q_i^T has a squared matrix of size 240-by-240 which is essential for the computation of matrices. Once we compute our covariant image we can move on to step six of the process and arguably the most important step, finding the eigenvalues relative to the data base images.

We know that the matrix M(C) associated to the image C is a matrix of size 240-by-240. Our goal is to find the eigenvalues of the Matrix (C). The Matrix M(C) has 240 eigenvalues because it is a 240-by-240 matrix. However only the 6 largest eigenvalues will be used for this demonstration.

Once we find our eigenvalues we need to find the eigen vector associated to each value. For each eigenvalue λ_i ($1 \leq i \leq 6$), we find an eigenvector v_i. It is important to note that each of the vectors v_i has 240 coordinates. In order to find the eigenfaces we write $v_i = (v_{ik})_{1 \leq k \leq 6}$ which takes each of

the six vectors and uses the first six coordinates from each vector. We then use this new value of v_i to find our eigenfaces. To calculate the 6 eigenfaces we use the formula

$$F_i = \sum_1^6 v_{ik} Q_k$$

The results are the following 6 faces.

These six faces therefore represent the common characteristic features of the faces from the data base. Therefore, one could say that an original face image could be reconstructed by some linear

combination of these eigenfaces. Recognition software could then apply a linear algorithm using these eigenfaces to find whether or not someone is in the original data base.

Further uses for this kind of software are plentiful, for example there are different facial detection and recognition software's such as, Security Systems- From the newest smartphones to top-tier securities, Human-Computer Interaction- Social media, next gen. AI, Law Enforcement/Government- F.B.I, C.I.A, etc. Facial recognition software is still relatively new and it is already advancing relatively quickly. The future possibilities for this kind of technology are endless. Facial Recognition is already being adopted by the retail industry faster than any other industry. Which some experts claim is due to the massive advantages that facial recognition software has within the industry. One such advantage is a system known as WatchList, which is a new face recognition data platform designed to help prevent shoplifting and violent crime. The system includes a managed database of known criminals that pose a safety, theft or violent crime risk, and uses feature matching technology to alert security about real-time threats.

4 Conclusion

While facial detection and recognition software is still relatively new and constantly advancing, the eigenface approach is a viable option for the future. The key benefits of the eigenface approach, when compared to other recognition methods, is that it is fast, and it is relatively simple. While these benefits may seem fairly insignificant, that couldn't be further from the truth. In the technology spectrum simplicity and speed are extremely valuable traits, especially when dealing with real-time threats. The need for this kind of technology will only increase as technology in general advances over time.

When it all boils down, eigenfaces essentially came about from the idea of basing face recognition on a set of image features that best approximate a set of known images, rather than basing it on our intuitive notions of facial parts or features. This method revolutionized recognition technology and is being adopted and altered all around the world. The accuracy of recognition software made a giant leap forward when the method of eigenfaces was introduced.

The eigenface method is therefore a very useful idea and has many real-world applications. This is merely one example of how mathematics can be applied to a seemingly unrelated field in the real world. China is already beginning to use technology like this in their police departments in order to find criminals. They have distributed facial recognition sunglasses to their officers and had them test the glasses out at train platforms. Not only have they been able to find wanted criminals they have also managed to find people that are traveling under false identities. This work has enable us to see real-word applications of linear algebra.

References:

- Axler, *Sheldon J. Linear algebra done right.* Cham: Springer, 2015. Print
- Hahn, Brian D., and Daniel T. Valentine. *Essential MATLAB for engineers and scientists.* Waltham, MA: Academic Press, 2013. Print
- Solomon, Chris and Toby Breckon. *Fundamentals of digital image processing: a practical approach with examples in Matlab.* Chichester: Wiley-Blacwell, 2011. Print
- http://www.vision.jhu.edu/teaching/vision08/Handouts/case_study_pca1.pdf
- https://www.facefirst.com/blog/brief-history-of-face-recognition-software/
- http://openbio.sourceforge.net/resources/eigenfaces/eigenfaces-html/facesOptions.html
- http://www.scholarpedia.org/article/Eigenfaces
- https://www.theverge.com/2018/2/8/16990030/china-facial-recognition-sunglasses-surveillance