Face Recognition

Dongyeon Joo

February 19, 2021

Abstract

This paper explores the Principal component analysis (PCA) used in human face recognition. PCA is one of the mathematical concepts used in data analysis, here we apply it to eigenfaces. This method was first developed by Sirovich and Kirby (1987) and used by Matthew Turk and Alex Pentland in face classification. The goal of this paper is to study the mathematics of concept of eigenface, and then implement the method classifying the pictures of four celebrities. Once the method is trained, we test it by feeding new images to the algorithm. The tests are analyzed in three different ways. First, new image will show unique pattern of eigenfaces when new image is projected onto the eigenspaces. Second, reconstructing image using eigenface. Third, classifying the new image by comparing a score. If the score is low, we conclude that new image and training image are of the same celebrity. If the score is high, we conclude that testing image and training image belong to different celebrities.

Contents

1 Introduction

Amazon is the one of the companies which lead the future in technology. We can see how they are leading the future by looking at examples such as Amazon Go store. An Amazon Go store is a brick and mortar store where you don't need to bring your wallet or wait in any lines to make a transaction because there is no cashier. All that's needed is for you to download the store app and log into your Amazon account, which assigns you a personal bar-code, and a quick phone scan at the entrance turnstile beforehand. Then you are free to shop and leave at your own convenience. This type of transaction at the Amazon Go store is made possible through the use of technology, by using sophisticated image recognition software along with artificial intelligence. Hundreds of tiny cameras create a three dimensional image of your face in order to recognize your face. In addition to tracking and analyzing you, it examines your every move to determine what groceries you are going to purchase so that you can just grab what you need and leave. When you walk out, you are charged only for what you purchased on your credit card, which is preregistered. As Amazon plans to create 3,000 new stores by 2021, this raises concerns regarding individual privacy and the utilization of facial recognition [9]. In the future this will be the new normal; you won't even have to take out your phone to check in with your ID when you go to shop. With facial recognition, your face alone is going to be the ID that gets you in the door. You may wonder how a computer can determine a person's identity. Is there any possibility that the computer will make a mistake and misidentify someone? We are going to talk about how computers recognize human faces and what types of technology allow them to do so.

2 Background

Humans recognize other human faces via the connection system between neural networks and memory. When we look at a human face our perception network system detects the standard features of human faces such as hair, eye color, wrinkles, etc. How then can a computer detect the features of a face and recognize people? With the development of technology, there are many methods to perform face recognition but methods are classified into four different base methods in large frame: Knowledge-Based, Feature-Based, Template Matching, and Appearance-Based [7]. First, the Knowledge-Based utilizes a set of rules similar to what humans use to detect faces. However, there are constraints within that set of rules that can lead to many false positive results. False positive is incorrect result in binary classification. For example, doctor examine you and determine you have a cold even though you do not have cold. Second, the Feature-Based method extracts facial features. After extracting the facial features, they are classified by a computer. However, it is difficult to define where the feature matches occur and feature pairs are very sparse in general. Third, Template Matching uses pre-defined or parameterized face templates to locate or detect faces through correlation between the templates and input images. However, this approach is inadequate for face detection because there is a hole in the data if part of an image does not exist in the template. If this occurs, it will think that two pictures are different when they are not. Fourth is Appearance-Based. This method uses a set of delegate training face images to determine face models. This approach is better than the other approaches that we talked about earlier. This approach relies on statistical analysis and machine learning to find the relevant features of facial images. However, this method takes extensive amounts of time to execute because it utilizes machine learning techniques or deep learning techniques. In this capstone, we will perform one of the sub-methods of the Appearance-Based model, eigenface. The eigenface method has several advantages with facial recognition. First, raw intensity data are used directly for learning and recognition without any significant low- or mid-level processing. Second, we do not need to understand geometry and reflection for face recognition. Third, eigenface method is simple and efficient compared to other methods because eigenface algoritm takes less time to train the dataset than other algorithms [3]. For these reasons, we are going to use the eigenface method to perform facial recognition in this capstone.

3 Linear Algebra

Linear algebra is a extended form of mathematics. As technology is growing, linear algebra is applied throughout science and engineering because the amount of data that computer has to process has increased. Linear algebra allows you to model natural phenomena and to compute them efficiently. [2]

Definition 1 (Linear algebra)**.** *Linear algebra is a branch of mathematics that is concerned with mathematical structures closed under the operations of addition and scalar multiplication and that includes the theory of systems of linear equations, matrices, determinants, vector spaces, and linear transformations*

Also, linear algebra is central to almost all areas of mathematics such as geometry and functional analysis. We are not going to talk about all details about the linear algebra. We are going to discuss some concepts of linear algebra to understand how face recognition works and how linear algebra is used to implement face recognition in machine learning technique. Before we jump into the topic, let's give a little background about linear algebra. First, scalar and vector.

Definition 2 (Scalar)**.** *Scalar is a physical quantity that is completely described by its magnitude; examples of scalars are volume, density, speed, energy, mass and time.*

Definition 3 (Vector)**.** *A vector is a quantity that has magnitude and direction and that is commonly represented by a directed line segment whose length represents the magnitude and whose orientation in space represents the direction.*

A scalar is an element of a field which has only magnitude in vectorspace. Vector is a quantity described multiple scalars, such as having both direction and magnitude in vectorspace.

Definition 4 (Matrix)**.** *A matrix is a collection of numbers arranged into a fixed number of rows and columns. Matrices can contain real and complex numbers.*

The formation of 2 by 3 matrix looks like down below.

Example 3.1. *2 × 3 matrix*

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

3.1 Eigenvectors and Eigenvalues

Eigenvectors and eigenvalues have many important applications in computer vision and machine learning in general. Well known examples are PCA (Principal Component Analysis) for dimensionality reduction or EigenFaces for face recognition. You will explore gentle introduction into eigenvalues and eigenvectors in this section. Eigen means "latent" or "characteristic" in German word. We can translate eigenvetors and eigenvalues as characteristic vector and characteristic values.

Definition 5 (Eigenvalues and Eigenvectors)**.** *Eigenvalues and Eigenvectors of a Matrix Suppose that A is a square matrix of size $n, x \neq 0$ is a vector in $\mathbb{C}^n$, and λ is a scalar in $\mathbb{C}$. Then we say x is an eigenvector of A with eigenvalue λ if [1]*

$$A\vec{x} = \lambda\vec{x}. \tag{1}$$

Let's take an example to understand meaning of characteristic in linear algebra. We have a matrix A and vector $\vec{x}$. A matrix stretches or rotates a vector by multiplying it. We can represent equation as $A\vec{x}$. However, we do not say any random vector is characteristic. Characteristic means special or is distinguished from other vectors. Most vectors points in some different direction but there are certain vector of $A\vec{x}$ point same direction of $\vec{x}$. We call this is eigenvectors. Since $A\vec{x}$ is parallel to $\vec{x}$, we can state this circumstance in equation. λ is the number that satisfies eigenvectors and we call this is eigenvalues. This equation is true for special vectors of x and special numbers of λ. Let's take an example of eigenvectors and eigenvalues to understand deeply with two different situation.

Example 3.2. *Two cases, different direction of two vectors and same direction of two vector.*

$$\text{Let } A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \quad , \vec{x_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\text{Then } A\vec{x_1} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

$$\text{Where } A\vec{x_1} \text{ and } \vec{x_1} \text{are not parallel.}$$

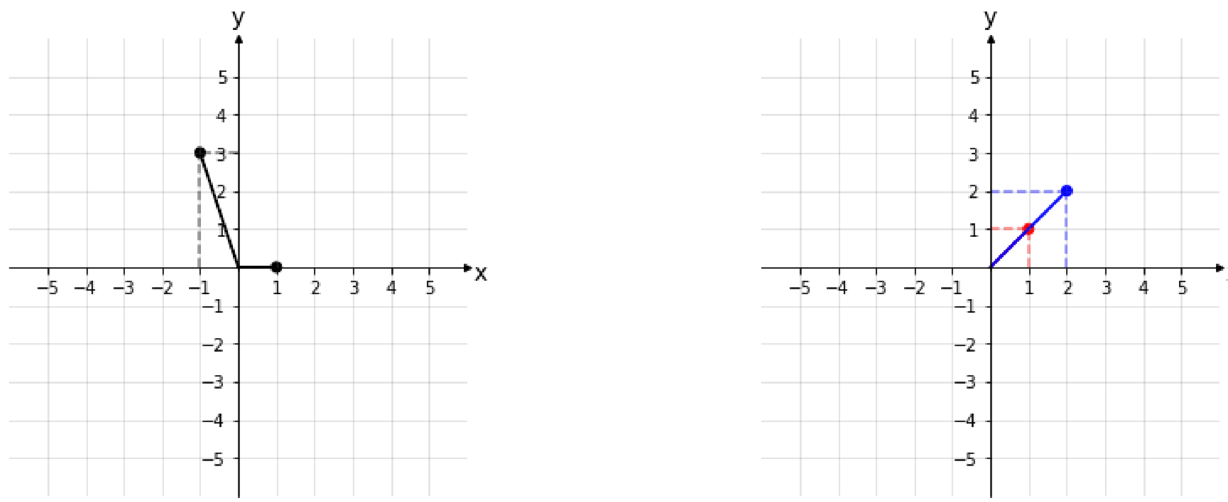
If we perform $A\vec{x_1}$ then we can see that there is no lambda by multiplying number on $\vec{x_1}$ to make $A\vec{x_1}$. Since $A\vec{x_1}$ and $\vec{x_1}$ are not on the same direction because there is no lambda to be satisfied equation $A\vec{x_1} = \lambda\vec{x}$, we know $\vec{x_1}$ is not eigenvector and there is no eigenvalues. *If $A\vec{x_2}$ and $\vec{x_2}$are in in same direction.*

$$\text{Let } A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \quad \vec{x_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\text{Then } A\vec{x_2} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Where $A\vec{x_1}$ and $\vec{x_1}$are parallel.

There is $\vec{x_2}$ and multiply matrix A. We found the vector that is on the same direction of $A\vec{x_2}$. If we multiply 2 on vector $\vec{x_2}$, we can have $A\vec{x_2}$. Since we found the values that satisfies equation $A\vec{x_2} = \lambda\vec{x_2}$, $\vec{x_2}$ is eigenvector and eigenvalue is 2. We can easily check this fact graphically in Figure 1.



(a) $\vec{x_1}$ and $A\vec{x_1}$ when two vectors are in different direction

(b) $\vec{x_2}$ and $A\vec{x_2}$ when two vectors are in same direction

Figure 1: eigenvectors and eigenvalue in Cartesian coordinate

However, we cannot find the eigenvectors and eigenvalues like plugging numbers into equation randomly because it takes much times to find the eigenvectors and eigenvalues. Eigenvectors and eigenvalues satisfy definition 1. So we can convert definition into different format. Let I be the identity matrix, let $\vec{x}$ be an eigenvector of a corresponding to eigenvalue λ then

$$A\vec{x} = \lambda I\vec{x}, \tag{2}$$

which implies that

$$A\vec{x} - \lambda I\vec{x} = (A - \lambda I)\vec{x} = 0. \tag{3}$$

Theorem 3.3. *The number λ is an eigenvalue of A if and only if $A - \lambda I$ is singular[1]*

If we move left side equation to right side and factor $\vec{x}$ out then we can have two cases to satisfy Definition 1. First case is $\vec{x}$ is equal to zero and the other case is $\vec{x}$ is not equal to zero and $A - \lambda I = 0$. Since $\vec{x}$ is non zero element then matrix $A - \lambda I$ will be singular by Theorem 3.3 and have zero determinant. We are not going to prove this property in this capstone. This two cases solve the equation. However, first case is not the case we are looking for because zero vector make any matrix into origin. Through second case, when we know an eigenvalue λ, we find an non-zero eigenvector solution.

Example 3.4.

$$\text{Let } A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \text{ and let I be the 2 x 2 identity matrix}$$

Then

$$A - \lambda I = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \quad - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$= \begin{bmatrix} 3-\lambda & -1 \\ -1 & 3-\lambda \end{bmatrix}$$

which implies

$$\det(A - \lambda I) = (3-\lambda)(3-\lambda) - 1 = \lambda^2 - 6\lambda + 8 = (\lambda - 4)(\lambda - 2) = 0$$

$$\lambda = 2, 4$$

λ is equal to 2 and λ is equal to 4 are solutions to $\det(A - \lambda I)$. 2 and 4 are eigenvalues. Since we have eigenvalues, we can compute eigenvectors. As we plug eigenvalues into the $\det(A - \lambda I)$ equation.

Example 3.5. *If* $\lambda = 2$

$$(A - 2I)\vec{x} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$x_1 = x_2$$

We can consider anything x_1 and x_2 are same that is eigenvector through example 3.5. We can take same procedure in if $\lambda = 4$. Through this calculation, we can find the eigenvalues and eigenvectors.

3.2 Singular Value Decomposition

The singular value decomposition, or SVD for short, is a matrix decomposition method for reducing matrix its constituent parts in order to make certain subsequent matrix calculations.

Definition 6 (Singular Value Decomposition)**.** *Let m and n be arbitrary; we do not require $m \geq n$. Given $A \in \mathbb{C}^{m \times n}$, not necessarily of full rank, a singular value decomposition(SVD) of A is a factorization*[10]

$$A = U\Sigma V^T. \tag{4}$$

where

$$\begin{gathered} A \in \mathbb{C}^{m\times n} \textit{ is rectangular matrix} \\ U \in \mathbb{C}^{m\times n} \textit{ is orthogonal matrix} \\ \Sigma \in \mathbb{C}^{m\times n} \textit{ is diagonal matrix} \\ V^T \in \mathbb{C}^{m\times n} \textit{ is orthogonal matrix} \end{gathered}$$

The SVD is a generalized method of eigendecomposition because SVD can be applied to non-square matrices but eigendecomposition can be performed only square matrices[10]. It helps to tailor a coordinate system or a transformation based on the matrix that we have. Easily, we can say that every $m \times n$ matrix A can be factored into three factors. We can describe the SVD in geometry interpretation. Before we observe SVD in geometry, we have to know that meaning of matrix multiplication to vector in geometry.

Example 3.6. *There are matrix A and vector $\vec{x}$*

$$\text{Let } A = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

then

$$\vec{y} = A\vec{x} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

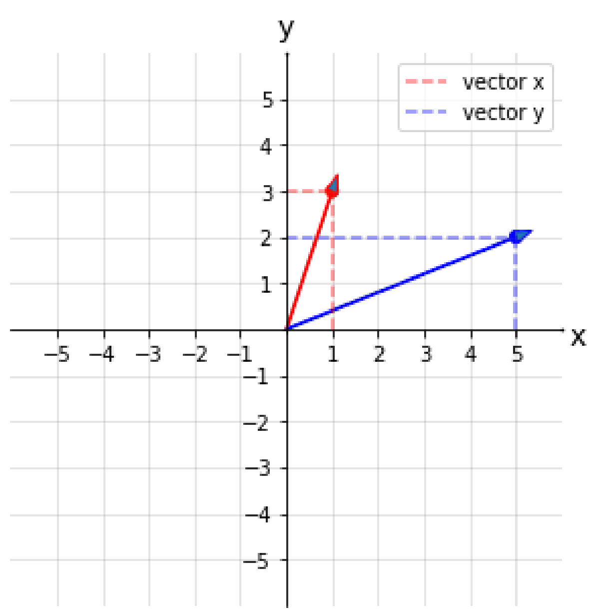


Figure 2: How vector changed when matrix A multiply to vector x

If you look at the Example 3.6, we can see that we transform one vector which is $\vec{x}$ to another vector which is $\vec{y}$. We chose vector x and rotated and then we stretched it. There is two fundamental of idea of matrix multiplication to vector is rotating and stretching. Let's expand this concepts to SVD. There is unit circle S which consisted of orthogonal vectors which are $\vec{v_1}$ and $\vec{v_2}$ in two dimension. If these vectors in three dimension, it is a sphere. If it exist in n dimension, it is called hypersphere. We have matrix $A \in \mathbb{C}^{m \times n}$. We are going to perform matrix multiplication to unit circle S. AS becomes ellipse because we rotate and stretch the circle. If we compute in higher dimension sphere or hypersphere will become hyperellipse. Let assume we have two vector component $\sigma_1 \vec{u_1}$ and $\sigma_2 \vec{u_2}$ in ellipse. $\vec{u_1}$ and $\vec{u_2}$ are unit vectors and it tell us which direction is going. Also, $\vec{u_1}$ and $\vec{u_2}$ are orthogonal because these two vectors are mapping from $\vec{v_1}$ and $\vec{v_2}$ with not changing angle. σ_1 and σ_2 tell us how much direction is going. We can see these situation in graphically in Figure 3[10].

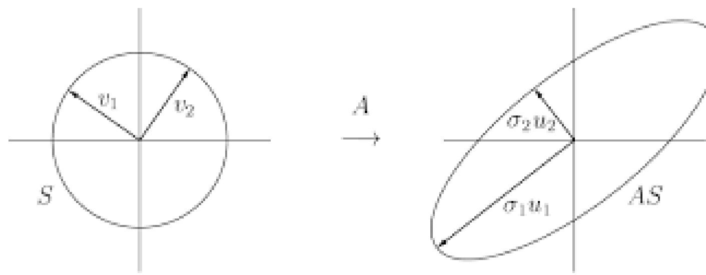


Figure 3: circle to ellipse when vector is multiplied matrix A

u_j is unit orthonormal vector and is called principal axes and σ_j is stretching factor and is called singular values. We can write this situation in formal way.

$$A\vec{v_j} = \sigma \vec{u_j}. \qquad (j = 1, 2, \cdots, n) \tag{5}$$

We can write this equation in short hand way in matrix multiply way.

$$AV = U\Sigma. \tag{6}$$

U and V are orthonormal components.There are special properties in orthogonal matrix which is inverse of the matrix is the same as its transpose matrix. So $U^T = U^{-1}$ which is the complex conjugate transpose and $V^T = V^{-1}$ is applied to same rule. Since they are orthogonal matrix, we can compute identity matrix $UU^T = U^TU = I$ and $VV^T = V^TV = I$. Through this calculation we can also check U and V are unitary matrix.

$$\begin{aligned} AV &= U\Sigma \\ AVV^{-1} &= U\Sigma V^{-1} \\ AI &= U\Sigma V^{-1} \\ A &= U\Sigma V^T. \end{aligned}$$

If we multiply V transpose on both side, we can get equation of Singular Value Decomposition which is SVD. The diagonal matrix means that outside the main diagonal such as off diagonal values are all zero. We can show this in matrix format.U is the left singular vector and V is the right singular vector. Σ is the singular value and it is hierarchically ordered $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m$.

$$A = U\Sigma V^T = \begin{pmatrix} | & | & & | \\ \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_m \\ | & | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & & & 0 \\ & \sigma_2 & & & 0 \\ & & \ddots & & 0 \\ & & & \sigma_m & 0 \end{pmatrix} \begin{pmatrix} - & \vec{v}_1^T & - \\ - & \vec{v}_2^T & - \\ & \vdots & \\ - & \vec{v}_n^T & - \end{pmatrix} \tag{7}$$

Theorem 3.7. *singular values $\{\sigma_j\}$ are uniquely determined, and if matrix A is square then σ_j is distinct.*[4]

Theorem 3.8. *$\{\vec{u}_j\}$ and $\{\vec{v}_j\}$ are also unique up to a complex sign.*[4]

The singular value which is Σ has three different cases in diagonal matrix; $m = n$, $m > n$, and $m < n$

$$\begin{pmatrix} \sigma_1 & 0 & ... & 0 \\ 0 & \sigma_2 & ... & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & ... & \sigma_m \end{pmatrix} \tag{8}$$

$$\begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{9}$$

$$\begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ & & \ddots & & & & \\ 0 & 0 & \cdots & \sigma_m & 0 & \cdots & 0 \end{pmatrix} \tag{10}$$

The expansion of A is $\sigma_1\vec{u}_1\vec{v}_1^T + \sigma_2\vec{u}_2\vec{v}_2^T + \cdots + \sigma_m\vec{u}_m\vec{v}_m^T + 0$. The 0 stands for after first m columns of U and V get multiplied by 0. In the expansion, $\vec{u}\vec{v}^T$ have the value between -1 and 1 because they are normalized. Since σ is in order from the biggest to the smallest, σ_1 is the best rank to determine size of matrix A and σ_2 is the next best rank and so on. We can calculate SVD using eigenvalue and eigenvector concepts.

Example 3.9. *Find V*

$$\begin{aligned} A^TA &= (U\Sigma V^T)^T(U\Sigma V^T) \\ &= V\Sigma U^TU\Sigma V^T \\ &= V\Sigma^2V^T \\ A^TAV &= V\Sigma^2V^TV \\ A^TAV &= V\Sigma^2 = \Sigma^2V \end{aligned}$$

We can solve this equation using eigen value problem from previous section. V is considered as eigenvector and Σ^2 is considered as eigenvalue.

Example 3.10. *Find U*

$$\begin{aligned} AA^T &= (U\Sigma V^T)(U\Sigma V^T)^T \\ &= U\Sigma V^T V\Sigma U^T \\ &= U\Sigma^2 U^T \\ AA^T U &= U\Sigma^2 U^T U \\ AA^T U &= U\Sigma^2 = \Sigma^2 U \end{aligned}$$

We can solve this equation using eigen value problem from previous section. U is considered as eigenvector and Σ^2 is considered as eigenvalue.

Through example 3.9 and 3.10, we can notice that AA^T and $A^T A$ have same eigenvalues but it depends on calculation we can get eigenvectors of U and eigenvectors of V.

Example 3.11. *Consider the SVD of A*

$$A = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 4 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 4 \end{bmatrix}$$

The eigenvalues of AA^T are clearly $\lambda = \{9{,}4\}$ and singluar values are $\sigma_1 = 3, \sigma_2 = 2$ using eigenvalue and eigenvector calculation.[4]

$$A = U\Sigma V^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

A can be factored three components using SVD.

3.3 Principal Component Analysis

We are going to use sigular value decomposition to compute principal component analysis. Principal Component Analysis is dimensionality reduction technique for probability in statistic and commonly used in machine learning application.

Definition 7 (Principal Component Analysis)**.** *The principal component(PCA) is an example of feature transformation approach where the new features are constructed by applying a linear transformation on the original set of features.*

The Principal Component Analysis, or PCA for short, is a dimension reduction tool to reduce a large set of variables to a small set that still contains most of the information in the large set. Precisely, PCA helps to find another basis, which is a linear combination of the original baisis, that best re-express our data set. For instance,analyzing 192×192 dataset is easier rather than analyzing $36,864 \times 36,864$ dataset. Before we dive into PCA, we need to know variance and covariance to understand PCA because variance and covariance explain relationships among elements. Variance is statistical measurement of how the values in the range far away from the mean.

Definition 8 (Variance)**.** *Variance measures how far each number in the set is from the mean and thus from every other number in the set. Variance is often depicted by this symbol* σ^2*.*

Let's look at the Figure 4. There are two bell curves which are A and B. We can see the bell curve that B is widespread than A. Since B is widespread, it shows that B is large variance and indicates that numbers in the set are far from the mean. A small variance which is A, it indicates the opposite.Let' apply this concept into vector. We have set of vector a and b and we assumed that mean of a and b are zero. Vector $\vec{a}$ consist of $[a_1, a_2, ..., a_n]$ and vector $\vec{b}$ consist of $[b_1, b_2, ..., b_n]$. We have mentioned above to see relationship among elements especially vector in this case. We have to know variance and covariance. We are going to compute variance of vector $\vec{a}$ and vector $\vec{b}$. We can compute variance using equation down below[4].

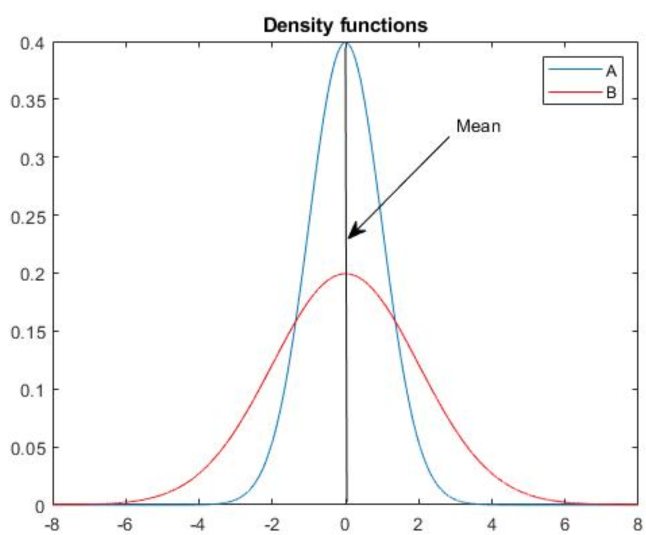


Figure 4: Gaussian distribution

Example 3.12. *Variance of set of vector a and set of vector b*

$$\sigma_a^2 = \frac{1}{n-1}\vec{a}\vec{a}^T \quad , \quad \sigma_b^2 = \frac{1}{n-1}\vec{b}\vec{b}^T$$

$$\sigma_a^2 = \begin{bmatrix} \vec{a_1} & \vec{a_2} & \vec{a_3} & \cdots & \vec{a_n} \end{bmatrix} \cdot \begin{bmatrix} \vec{a_1} \\ \vec{a_2} \\ \vec{a_3} \\ \vdots \\ \vec{a_n} \end{bmatrix} = \vec{a_1} \cdot \vec{a_1} + \vec{a_2} \cdot \vec{a_2} + \cdots + \vec{a_n} \cdot \vec{a_n}$$

$$\sigma_b^2 = \begin{bmatrix} \vec{b_1} & \vec{b_2} & \vec{b_3} & \cdots & \vec{b_n} \end{bmatrix} \cdot \begin{bmatrix} \vec{b_1} \\ \vec{b_2} \\ \vec{b_3} \\ \vdots \\ \vec{b_n} \end{bmatrix} = \vec{b_1} \cdot \vec{b_1} + \vec{b_2} \cdot \vec{b_2} + \cdots + \vec{b_n} \cdot \vec{b_n}$$

$\vec{a}\vec{a}^T$ and $\vec{b}\vec{b}^T$ are the inner product which provides the length of the vector and $\frac{1}{n-1}$ is an unbiased estimator. σ_a^2 and σ_b^2 represent how the large the changes are in the vector of a and b. Since we saw relationship among each vector, we are going to concern relationship between $\vec{a}$ and $\vec{b}$. We can check the relationship through covariance because covariance is a measure of how much two random variables vary together.

Definition 9 (Covariance)**.** *Covariance is a statistical tool that is used to determine the relationship between the movement of two asset prices. When two stocks tend to move together, they are seen as having a positive covariance; when they move inversely, the covariance is negative.*

We can compute the covariance same method as variance.

Example 3.13. *Covariance of set of vector a and set of vector b*

$$\sigma_{ab}^2 = \frac{1}{n-1}\vec{a}\vec{b}^T$$

$$\sigma_{ab}^2 = \begin{bmatrix} \vec{a_1} & \vec{a_2} & \vec{a_3} & \cdots & \vec{a_n} \end{bmatrix} \cdot \begin{bmatrix} \vec{b_1} \\ \vec{b_2} \\ \vec{b_3} \\ \vdots \\ \vec{b_n} \end{bmatrix} = \vec{a_1} \cdot \vec{b_1} + \vec{a_2} \cdot \vec{b_2} + \cdots + \vec{a_n} \cdot \vec{b_n}$$

$\vec{a}\vec{b}^T$ is the inner product and this equation tell us the score of covariance. This covariance scores tell you how much of $\vec{a}$ and $\vec{b}$ are in same direction. We assume that both $\vec{a}$ and $\vec{b}$ are length 1. If $\vec{a}$ and $\vec{b}$ are in the exact same direction, you will get the maximum covariance score as a result. If they are orthogonal, you will get zero as the minimum covariance score. It means $\vec{a}$ and $\vec{b}$ are not related. If you want to know relationship more than two vectors, you can construct covariance matrix. Covariance matrix is represented as same equation of covariance.

Definition 10 (Covariance Matrix)**.** *Covariance matrix is a square matrix giving the covariance between each pair of elements of a given random vector. Any covariance matrix is symmetric and positive semi-definite and its main diagonal contains variances*

For instance, you have set of $\vec{X}$ and it consist of $[\vec{x_1}, \vec{x_2}, \cdots, \vec{x_n}]$ then you can create down below matrix using covariance equation.

Example 3.14. *Covariance matrix of X*

$$\text{Let } \vec{X} = \begin{bmatrix} \vec{x_1} \\ \vec{x_2} \\ \vdots \\ \vec{x_n} \end{bmatrix}, \quad \text{C be a covriance matrix of } XX^T$$

Then

$$\begin{aligned} C = \vec{X}\vec{X}^T &= \cdot \begin{bmatrix} \vec{x_1} \\ \vec{x_2} \\ \vdots \\ \vec{x_n} \end{bmatrix} \begin{bmatrix} \vec{x_1} & \vec{x_2} & \cdots & \vec{x_n} \end{bmatrix} \\ &= \begin{bmatrix} \sigma_{x_1x_1} & \sigma_{x_1x_2} & \cdots & \sigma_{x_1x_n} \\ \sigma_{2_1x_1} & \sigma_{x_2x_2} & \cdots & \sigma_{x_2x_2} \\ & & \ddots & \\ \sigma_{x_nx_1} & \sigma_{x_nx_2} & \cdots & \sigma_{x_nx_n} \end{bmatrix} \end{aligned}$$

This covariance matrix shows that all relationships among pairs in set of $\vec{X}$. If you look at the pattern of matrix you can check easily, we can find the variance values in diagonal in covariance matrix. Off diagonal in covariance matrix, we can obtain covaraince values between all pairs. Now you are going to make diagonal matrix from covariance matrix because you know that $\sigma_{x_1x_2}$ and $\sigma_{x_2x_1}$ are same covariance score. There are redundancy in covariance matrix. So we want to remove redundant values in covariance matrix. If you make diagonal matrix from covariance matrix and put the elements in order from the largest to smallest, then you can sort in order the elements affect on the set of $\vec{X}$. We can compute eigenvalue of C and eigenvector of C using concept that we have learned in previous section.

$$CV = VD. \tag{11}$$

where

$$\begin{aligned} C &: \text{Covariance matrix} \\ V &: \text{Eigenvector} \\ D &: \text{Eigenvalues} \end{aligned}$$

Here is the principal components coming. If we set new frame of reference with T as principal components. We can write equation like down below

$$\text{Let } T = XV, \quad X = U\Sigma V^T$$

Then

$$T = U\Sigma V^T V$$
$$T = U\Sigma.$$

T is a Principal component, X is a set of vectors and V is Eigenvector. Through this computation, we can get the principal components and eigenvectors directly from SVD. This is why SVD is important in principal component. Also, Singular values which is sigma can be calculated from covariance matrix using SVD method. We used SVD and make matrix to diagonal that means we remove redundant data in our covariance matrix.[4]

$$C = U\Sigma V^T$$
$$C \cdot U^T = \Sigma V^T$$
$$(C \cdot U^T) \cdot (C \cdot U^T)^T = (\Sigma V^T) \cdot (\Sigma V^T)^T = \Sigma^2.$$

The singular values in sigma give you an indication of the amount of the variance of the dataset X. If we want to high dimensional data in terms of the first two principal components or first two eigenvectors, we can use principal component technique.

$$\Sigma^2 = \begin{bmatrix} \lambda_1 = \sigma_1^2 & 0 \\ 0 & \lambda_2 = \sigma_2^2 \end{bmatrix}. \tag{12}$$

Values of λ are equal to the square of the singular values and it is equal to the variance of that principal component in the data X.

Example 3.15. *Example of PCA calculation step by step[8].*
1.Create dataset. This dataset consist of four features and five training examples.

$f1$	$f2$	$f3$	$f4$
-1	-0.63246	0	0.26062
0.33333	1.26491	1.73205	1.56374
-1	0.63246	-0.57735	-0.17375
0.33333	0	-0.57735	-1.04249
1.33333	-1.26491	-0.57735	-0.60812

X is feature matrix

$$X = \begin{bmatrix} -1 & -0.63246 & 0 & 0.26062 \\ 0.33333 & 1.26491 & 1.73205 & 1.56374 \\ -1 & 0.63246 & -0.57735 & -0.17375 \\ 0.33333 & 0 & -0.57735 & -1.04249 \\ 1.33333 & -1.26491 & -0.57735 & -0.60812 \end{bmatrix}$$

2.Calculate the covariance matrix from feature matrix. C is a covariance matrix

$$C = XX^T = \begin{bmatrix} 0.8 & -0.25298 & 0.03849 & -0.14479 \\ -0.25298 & 0.8 & 0.51121 & 0.4945 \\ 0.03849 & 0.51121 & 0.8 & 0.75236 \\ -0.14479 & 0.4945 & 0.75236 & 0.8 \end{bmatrix}.$$

3.Calculate eigenvalues and eigenvectors using $\det(A - \lambda I)v = 0$ *from example 3.5 and 3.6*

$$\det(A-\lambda I)v = \begin{bmatrix} 0.8-\lambda & -0.25298 & 0.03849 & -0.14479 \\ -0.25298 & 0.8-\lambda & 0.51121 & 0.4945 \\ 0.03849 & 0.51121 & 0.8-\lambda & 0.75236 \\ -0.14479 & 0.4945 & 0.75236 & 0.8-\lambda \end{bmatrix} \times \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = 0$$

If $\lambda = 2.51579324$, $v_1 = 0.16195986$, $v_2 =$ -0.52404813, $v_3 =$ -0.58589647, and $v_4 =$ -0.59654663. We can create more eigenvectors using different eigenvalues.

4. Sort eigenvalues and their corresponding eigenvectors(4 by 4 eigenvector).

$e1$	$e2$	$e3$	$e4$
0.161960	−0.917059	−0.307071	0.196162
−0.524048	0.206922	−0.817319	0.120610
−0.585896	−0.320539	0.188250	−0.720099
−0.596547	−0.115935	0.449733	0.654547

This table shows that eigenvectos that match to eigenvalues.

5. Pick k eigenvalues and form a matrix of eigenvectos.

If we choose the top two eigenvectors and create matix then it will look like down below(4 by 2 matrix).

$e1$	$e2$
0.161960	−0.917059
−0.524048	0.206922
−0.585896	−0.320539
−0.596547	−0.115935

6. Transform the original matrix as projecting onto the eigenvector

$$CV = \begin{bmatrix} 0.8 & -0.25298 & 0.03849 & -0.14479 \\ -0.25298 & 0.8 & 0.51121 & 0.4945 \\ 0.03849 & 0.51121 & 0.8 & 0.75236 \\ -0.14479 & 0.4945 & 0.75236 & 0.8 \end{bmatrix} \times \begin{bmatrix} 0.161960 & -0.917059 \\ -0.524048 & 0.206922 \\ -0.585896 & -0.320539 \\ -0.596547 & -0.115935 \end{bmatrix}$$

$$= \begin{bmatrix} 0.014003 & 0.755975 \\ -2.556534 & -0.780432 \\ -0.051480 & 1.253135 \\ 1.014150 & 0.000239 \\ 1.579861 & -1.228917 \end{bmatrix}$$

The result is 5 by 2 matrix. We get the 5 by 2 matrix from 5 by 4 matrix using PCA method.

The principal component analysis enables us to use reduced set of variables. In example 3.15, we have four features and five entities in each feature. After using PCA method, we get two features and five entities in each feature. If we have larger dataset than the example like 500 features and each feature has 10,000 entities, then it will take a lot of times to analyze the datset and take a lot of storage space. Since we are using PCA method in analyzing dataset, we can perform analyzing smaller dataset than original dataset so that we can save time and storage space to analyze dataset.

4 Face Recognition

We have learned some Linear algebra concepts to understand mathematics used in face recognition algorithm. In this section, we are going to learn how face recognition is working using PCA method in real life and perform experiment using twenty pictures of four celebrities.

4.1 Eigenface

Eigenface is the key concept of face recognition in PCA method. There is example of eigenface in Figure 5 [6]. You might think this picture looks like ghost or blurry image and how we can recognize human face from this picture. In this section, we are going to discuss the meaning of eigenface and how computer recognize human face using eigenface.

Definition 11 (Eigenface)**.** *Eigenfaces are the eigenvectors of the set of faces. Also, it is set of chateristic feature images. Each face image can be represented exactly in terms of linear combination of the eigenfaces [11].*

Figure 5: Eigenface

We talked about SVD that how we can decompose large high dimensional datasets represented in three component which are $A = U\Sigma V^T$. Also, we talked about PCA that one of the dimension reduction technique. We are going to apply these two fundamental mathematical concept into real life application which is face recognition. The key in face recognition is feature. Feature may or may not be directly related to our intuitive notion of face feature like eyes, nose, lips and hair. We want to find the principal component of the feature in face to classify who is who. Eigenvectors can be thought of as a set of feature which together characterize the variable among face images. Each image location contributes more or less to each eigenvector, so that we can display the eigenvector as a sort of face which we call an eigenface [11]. PCA method can display eigenfaces in of importance. If we look at the Figure 5, we can see the a_1v_1 to a_8v_8. The $v_n(1 \leq n \leq 8)$ represents vector in face space and $a_n(1 \leq n \leq 8)$ means how importatnt feature in face. We can generate image by adding eight eigenfaces like $a_1v_1 + a_2v_2 + ... + a_8v_8 = \textit{generated image}$. Even though generated picture is not exact same as original picture X ,we can figure out those are similar. Let we have an image that consisted of 120 by 80 pixel. Pixel is the number that represent physical point in a raster image. Each pixel can have number from 0 to 255 and 0 is taken to be black and 255 is taken to be white in gray scale. Image can be converted into data matrix from image by rearranging pixels. If we calculate 120×80, it is 9600 and this is rearrange image. The purpose of rearranging is to put one image data into one column. rearranged image is tall skinny column vector in data matrix. If we assume that we have more than one images which consited of 120 by 80 then we will have 9600 by n data matrix. We are going to perform this process with more image in experiment section. However, we are going to handle only one image in this section to understand process of eigenface. We are going to calculate covariance matrix from the rearranged image. Covaraiance matrix from A which consited of 9600 by 1 image matrix

$$C = AA^T = \begin{pmatrix} | \\ \text{REARRANGED IMAGE} \\ | \end{pmatrix} \cdot \begin{pmatrix} - & \text{REARRANGED IMAGE} & - \end{pmatrix}$$

Covariance matrix is 9600 by 9600 because 9,600 $\times$ 1 $\times$ 1 $\times$ 9,600 is 9600 by 9600 by matrix multiplication. From covariance matrix C must compute a $9,600 \times 9,600$ matrix and calculate 9,600 eigenfaces [5]. 9,600 eigenfaces are not efficient because most of eigenfaces are not useful. We saw covariance matrix create many redundancy from example 3.14. We need a few eigenfaces to recognize who is this person is rather than 9,600 eigenfaces. This is why we are using PCA method in face recognition we need important feature to recognize human face instead entire eigenfaces. We can calculate dominant eigenfaces in order of importance using eigenvalue equation. Since we know matrix of AA^T is very large, we can make new frame of reference using L. L is 1 by 1 matrix because $1 \times 9,600 \times 9,600 \times 1$is 1×1 by matrix multiplication.

$$L = A^T A = \begin{pmatrix} - & \text{REARRANGED IMAGE} & - \end{pmatrix} \cdot \begin{pmatrix} | \\ \text{REARRANGED IMAGE} \\ | \end{pmatrix}$$

v_i is the eigenface of L.

$$Lv_i = \mu_i v_i \tag{13}$$

$$A^T A v_i = \mu_i v_i \tag{14}$$

$$AA^T A v_i = \mu_i A v_i \tag{15}$$

$$CAv_i = \mu_i A v_i \qquad \text{(Where } u_i = Av_i\text{)} \tag{16}$$

$$C(u_i) = \mu_i (Av_i). \tag{17}$$

Through the calculation we can see that C and L have the same eigenvalues and eigenfaces are related as where v_i is an eigenvector of L. Av_i is an eigenface of C. To summarize what we have done so far to understand eigenface, we discussed SVD to divide image matrix in three component and used eigenvalues to find the eigenfaces in image space then we used PCA to project feature in order of importance onto eigenface space. We get regenerated image and we can use this regenerated image to recognize human face. This is one of the face recognition technique using eigenface method.

4.2 Experiment of Eigenface

Let's perform the face recognition using MATLAB. The goal of this experiment is to show how PCA can be applied to a program and how it distinguishes pictures if the information is not reflected in the dataset. The procedure of face recognition is

1. acquire an initial set of face images (training set).
2. compute average of picture (average of each celebrities images)
3. To take an image and turn it into a vector to calculate eigenvalues and eigenfaces.
4. project average face onto the eigenvector space to regenerate image.
5. classify the weight pattern as either a known person or as unknown.

We are going to perform this task with five pictures of four celebrities in Figure 6. All images have different background and in different condition because we want to know how this experiment is close to real life. In real life, it is hard to get pictures with all same condition. They are Emma Watson, Emma Stone, Tom Cruise, and Chris Evans. Images consist of pixels and each pixel has a brightness and a color. Each pixel can be represented by a number from 0 to 255. Before we perform this task, we converted all the images under the same conditions of lighting and same size to get accurate result and better performance in face recognition. All images are in the training set resized 120 by 80 in gray scale, and pictures are closed-up and front faces. You are going to calculate average face of each celebrities because all Emma Watson pictures are correlated by itself. For instance, position of Emma Watson's nose is different from that Emma Stone's nose but the position of nose is similar in all of Emma Watson's photo. Emma Watson's average face can calculate as adding up all images and divide by number of pictures.

$$\text{Emma Waton Average face} = (\text{Image of Emma Watson 1} + \text{Image of Emma Watson 2} \\ + \text{Image of Emma Watson 3} + \text{Image of Emma Watson 4} + \text{Image of Emma Watson 5}) \ / \ 5$$

Other pictures can be calculated into same way and there is average picture of celebrities in Figure 7. First picture in the first row is the average face of Emma Watson from training set. Even though Emma Watson's pictures are overlapped, you can still figure out this is Emma Watson because features of Emma Watson's pictures are correlated. Other pictures are overlapped like Emma Watson's picture, but you can distinguish celebrities who is who. we are going to use PCA method to reduce the large dimension to relatively smaller set of vectors. Each image consist of 120 by 80 pixel and reshaped into 1 by 9600 as one row vector. Reshaped image is stored in data matrix A. If we perform $A^T \cdot A$, then you can have correlation matrix that is 9600 by 9600 and result of $A^T \cdot A$ is stored into C.

$$C = A^T \cdot A.$$

The variance of an image and itself can be found in the correlation matrix. The correlation matrix shows the variance among all pairs in the dataset. In the pictures among same celebrities will show high correlation scores but among different celebrities picture will give you low correlation scores. For instance, relationship between among same celebrities picture

Figure 6: training set of celebrities image

Figure 7: Average face of celebrities

will show high correlation because they are same person's picture. However, among different person picture like Tom Cruise and Chris Evans will show low correlation scores because they

$$
A = \begin{pmatrix} - & \text{Image of Emma Watson 1} & - \\ - & \text{Image of Emma Watson 2} & - \\ - & \text{Image of Emma Watson 3} & - \\ & \vdots & \\ - & \text{Image of Chris Evans 5} & - \end{pmatrix}
$$

Figure 8: Rearranged images in data matrix A

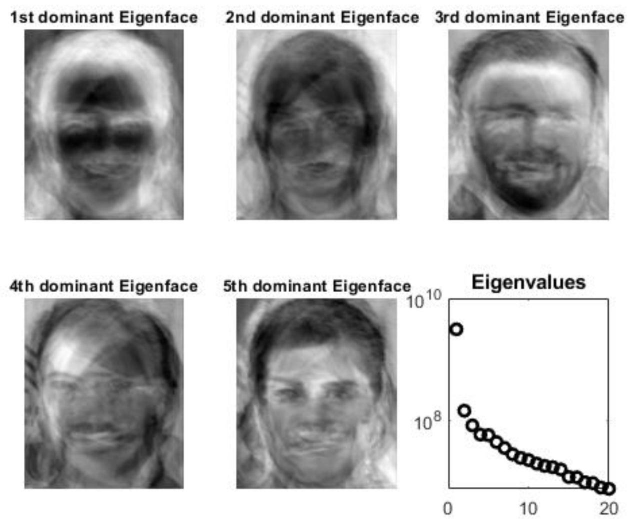


Figure 9: Twenty eigenfaces in order of dominant

are different person. You are going to find the twenty biggest eigenvalues from our image dataset C because we have total twenty of pictures so twenty is the maximum numbers that can show in eigenvalues in diagonal matrix. The first picture in the first row in Figure 9 shows that the first eigenvalue corresponds to the first dominant eigenface. The second picture in the first row of Figure 9 shows that the second eigenvalue corresponds to the second dominant eigenface, and so on. The last plotting graph in the second row shows that the biggest twenty eigenvalues affect the dominant face. In the tendency of graph, it is decay and there is huge gap of the point between first eigenvalue as almost 10^{10}, and the second eigenvalue as 10^8 This shows that the biggest eigenvalues are dominant values in recognize the face.

Figure 10 shows that projecting average face of each celebrities onto the twenty eigenvector space. For instance, first picture in the first row shows the values when average face of Emma Watson image project onto the twenty eigenface space. We can utilize this process through Emma Stone, Tom Cruise and Chris Evans. If we look at all of the projections onto twenty eigenface, we can find out the importance of pattern of bar plot. Every person has a very different pattern. These all graphs are unique because every image is different. This exactly identifies who they are. Through this process we complete training our dataset.

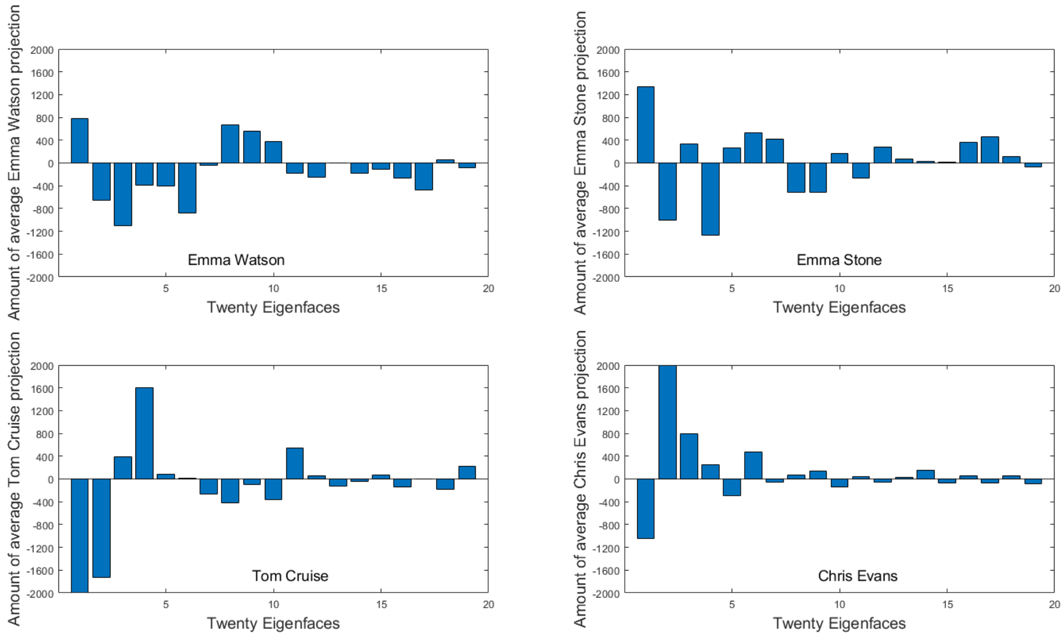


Figure 10: Project average face of each celebrities onto twenty eigenfaces

4.3 Test result

Since we have gotten the features of each celebrities, we are going to perform a three test with three different pictures. The purpose of this test is to see how computer recognize the face using PCA with score when there is new input that is not in the training set. First picture is Emma Watson from training set. Second picture is new picture of Emma Watson that is not in the training set. Third picture is new picture of Robert Downey Jr that is not in the training set and we did not train the image. We are going to take same process as experiment section and you can see the result in Figure 11. First second picture of first row is the result of projecting new image onto twenty eigenvector space from training set. Third picture in the first row is the image that is reconstructed using eigenfaces. You can see the image exact same picture of Emma Watson from training dataset. However, It is not actually exact same because we regenerate image using eigenfaces and eigenfaces include dominant features not all features. Fourth picture in the first row is the result that project new input image onto the five different Emma Watson pictures from the training set. There is no differences between input image and first picture of Emma Watson image from training set because those are same picture. However, other pictures show differences. Comparing input image with second picture of Emma Watson from training set shows 0.7 this is the biggest number in this picture of graph. This result shows that the second image is the most different when comparing input image with other Emma Watson's picture from training set. The smaller number in the graph means that the more similar between new input image and Emma Watson image from training set. Second picture in the second row is the result of twenty eigenvectors and this is unique pattern. Third picture of second row is the result of reconstructing image using eigenfaces. Fourth picture in the second row is the result of comparing new input image with all Emma Watson's images from training set. The result

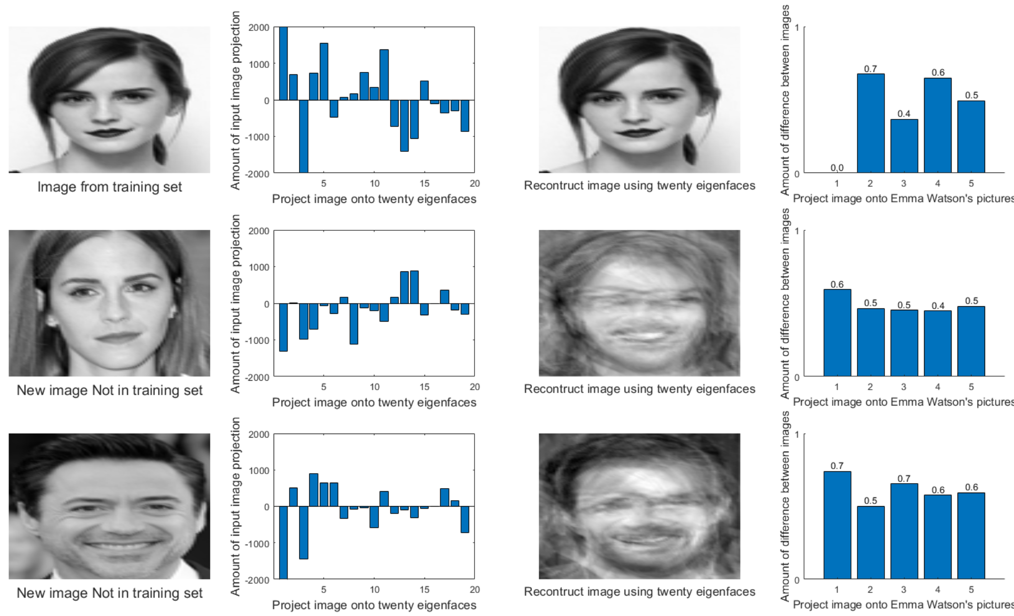


Figure 11: Test result

shows that fourth picture of Emma Watson from training set is the most similar to new input image because we trained the data. First picture in the third row is new picture and not in the training set and not trained. Second picture of third row shows the twenty eigenvectors with unique pattern of input image. Third picture in the third row is the reconstructing image using twenty eigenfaces. The fourth picture in the third low shows the score between differences between five Emma Watson pictures from training set and input image. We can see that third picture has little bit higher scores than second picture if we compare fourth graph of second row and third row. We can conclude that second picture is close to Emma Watson than third picture.

5 Conclusion

In this paper, the face recognition system was implemented with the use of Principal Components Analysis and the eigenface approach. The algorithm was both tested with the celebrities from Google Images and implemented with Python 3.8 and MATLAB(R2020b). We could learn the meaning of PCA and how it is used and implemented through the program. The PCA algorithm is turned to be good at classification of faces, but this experiment has limitations on the size of images and changes in color because it performed on certain condition for better performance.

6 Future Plan

This experiment was performed under certain conditions and it performed well:

1. the first pictures provided were all closed-up front faces and their dimensions were fixed. Our future goal is to develop a system with a video camera that recognizes the face in a real-time.

2. the size of recognizing the face image would overcome other issues because the size of every picture is same.

3. the colors of all image are gray, and we would like to implement the task with the pictures in various colors.

However, it is hard to apply to real life because human is moving and everyone is in different condition in real time. For example, in Figure 6 fourth row, all Chris Evans pictures have beard however we can find the Chris Evans picture which does not have beard. If Chris Evans does not have beard, computer can recognize this is Chris Evans? etc. For future paper, we are going to perform this experiment in various conditions. To see how is the face recognition rate different between specific condition and uncertain condition.

References

[1] Robert A. Beezer. *A first course in linear algebra*. Congruent Press, 2012.

[2] Niklas Donges. Basic linear algebra for deep learning, Oct 2019.

[3] Divyansh Dwivedi. Face recognition for beginners, Mar 2019.

[4] Jose Nathan Kutz. Data-driven modeling scientific computation : methods for complex systems big data, 2013.

[5] Sujay Patole and Yatin Vispute. Automatic attendance system based on face recognition.

[6] Department of computer science Princeton University. Face recognition from computer vision, fall 2007.

[7] Qaim Mehdi Rizvi, Bal Gopal Agarwal, and Rizwan Beg. A review on face detection methods. *Journal of Management Development and Information Technology*, 11(02), 2011.

[8] Gursewak Singh. Understanding principle component analysis(pca) step by step., Jan 2020.

[9] Nick Statt. Amazon reportedly planning 3,000 cashier-less go stores by 2021, Sep 2018.

[10] Lloyd N. Trefethen and David Ill. Bau. *Numerical linear algebra*. SIAM Society for Industrial and Applied Mathematics, 2000.

[11] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587. IEEE Computer Society, 1991.

A First appendix section

Image sources

Emma Watson Images

- First Emma Watson image - https://www.speakerbookingagency.com/talent/emma-watson (visited on 10th Feburary 2021).
- Second Emma Watson image - https://www.cheatsheet.com/entertainment/emma-watson-admits-she-suffers-from-imposter-syndrome.html/(visited on 10th Feburary 2021).
- Third Emma Watson image - https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.thehandbook.com%2Fcelebrity%2Femma-watson%2F&psig=AOvVaw0F1Nsb54z0Gtve7uZKfMdD&ust=1609968578314000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCOi75cfehe4CFQAAAAAdAAAAABAT(visited on 10th Feburary 2021).
- Fourth Emma Watson image - https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.thehandbook.com%2Fcelebrity%2Femma-watson%2F&psig=AOvVaw0F1Nsb54z0Gtve7uZKfMdD&ust=1609968578314000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCOi75cfehe4CFQAAAAAdAAAAABATory/life/movies/2016/05/10/emma-watson-named-panama-papers/84196712/(visited on 10th Feburary 2021).
- Fifth Emma Watson image - https://www.reddit.com/r/EmmaWatson/comments/dtuflq/beautiful/(visited on 10th Feburary 2021).

Emma Stone Images

- First Emma Watson image - https://en.wikipedia.org/wiki/List_of_Emma_Stone_performances#/media/File:Emma_Stone_at_Maniac_UK_premiere_(cropped).jpg(visited on 10th Feburary 2021).
- Second Emma Stone image - https://en.wikipedia.org/wiki/Emma_Stone#/media/File:Emma_Stone_at_the_39th_Mill_Valley_Film_Festival_(cropped).jpg(visited on 10th Feburary 2021).
- Third Emma Stone image - https://www.imdb.com/name/nm1297015/mediaviewer/rm3228157440/(visited on 10th Feburary 2021).
- Fourth Emma Stone image - https://economictimes.indiatimes.com/magazines/panache/meditating-dancing-writing-help-emma-stone-drive-away-her-blues/articleshow/75534032.cms(visited on 10th Feburary 2021).
- Fifth Emma Stone image - https://www.usmagazine.com/stylish/news/emma-stone-answers-vogues-73-questions-video-w445437/(visited on 10th Feburary 2021).

Tom Cruise Images

- First Tom Cruise image - https://www.goldenglobes.com/person/tom-cruise(visited on 10th Feburary 2021).
- Second Tom Cruise image - https://www.thefamousbirthdays.com/people/tom-cruise(visited on 10th Feburary 2021).
- Third Tom Cruise image - https://www.vulture.com/2018/08/19-unsettlingly-nice-tom-cruise-stories.html(visited on 10th Feburary 2021).
- Fourth Tom Cruise image - https://www.justjared.com/photo-gallery/2468647/cameron-diaz-tom-cruise-bordeaux-02/fullsize/(visited on 10th Feburary 2021).
- Fifth Tom Cruise image - https://www.biography.com/actor/tom-cruise(visited on 10th Feburary 2021).

Chris Evans Images

- First Chris Evans image - https://en.wikipedia.org/wiki/List_of_Chris_Evans_performances#/media/File:Chris_Evans_-_Captain_America_2_press_conference_(cropped).jpg(visited on 10th Feburary 2021).
- Second Chris Evans image - https://www.britannica.com/biography/Chris-Evans-American-actor(visited on 10th Feburary 2021).
- Third Chris Evans image - https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.tvguide.com%2Fcelebrities%2Fchris-evans%2F144489%2F&psig=AOvVaw0W5ckT-6J7OyAC7S2ZUc3Z&ust=1609969822101000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCLDjyJHjhe4CFQAAAAAdAAAAABAd(visited on 10th Feburary 2021).
- Fourth Chris Evans image - https://www.popsugar.com/Chris-Evans(visited on 10th Feburary 2021).
- Fifth Chris Evans image - https://atozhairstyles.com/articles/chris-evans-hair/(visited on 10th Feburary 2021).

Test Image

- Emma Watson - https://www.nickiswift.com/199771/emma-watsons-biggest-regret-about-harry-potter/(visited on 10th Feburary 2021).
- Rober Downey Jr - https://ew.com/article/2011/04/05/the-hunger-games-haymitch-effie-prim-who-should-play/(visited on 10th Feburary 2021).

B Second appendix section

MATLAB code - https://github.com/djoo1028/Face-Recognition