Project Report: Eigenfaces for Recognition

Yuchen Dang (yd1008), Zian Jiang (zj444), Yihang Zhang (yz2865)

May 2021

1 Introduction

In this project, we will investigate how Eigenfaces works in the face recognition problem, its connection to numerical linear algebra, and its drawbacks. In Section 2 we will summarize the main results from the original paper [3], and then we will implement this algorithm and show the results of three applications on the Yale Faces Dataset b [2] in Section 4.

2 Eigenfaces

2.1 Notations

Suppose we have n grey-scale images I_i of width n and height m, which can be represented as a matrix M_i of size $n \times m$. We can then represent each image I_i as a column vector x_i by reshaping M_i into a vector of length $d := mn$. Thus, the mean face vector Ψ:

$$\Psi = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{1}$$

Now, we want to center the data around the mean Ψ. Thus, offset each x_i by the mean face vector:

$$\Phi_i = x_i - \Psi. \tag{2}$$

Finally, stacking these column vector Φ_i's side by side, we obtain data matrix X of size $d \times n$:

$$X := \begin{bmatrix} \Phi_1 & \Phi_2 & ... & \Phi_n \end{bmatrix} \tag{3}$$

2.2 Main Idea

The main ideas of Eigenfaces are image compression and dimensionality reduction. As images are usually high dimensional (in our notations they are in $\mathbb{R}^d$), computation involving them directly may be expensive. The goal is to approximate any image vector Φ by a linear combination of a small set of useful features. In other words, to represent image vectors in a low-dimensional space $\mathbb{R}^r$ instead of the original space $\mathbb{R}^d$ for $r << d$. As suggested, this is a dimensionality reduction problem involving Singular Value Decomposition (SVD). Given data matrix X of size $d \times n$, we can perform full SVD to decompose it to into

$$X = U\Sigma V^T, \tag{4}$$

where columns of

$$U := \begin{bmatrix} u_1 & u_2 & \dots & u_d \end{bmatrix} \tag{5}$$

are called Eigenfaces, which can be regarded as the basis so that we can represent any new image as a linear combination of these vectors. Each u_i is of length d, and we can reshape it back to $n \times m$ and visualize it as facial feature, hence the name Eigenfaces. Since we want a low-dimensional yet still good representation and Eigenvectors are sorted decreasingly by singular values in Σ, we can approximate X by taking only the first r columns of U (U_r), first r elements in diagonal matrix Σ (Σ_r), and the first r rows of $V^T (V_r^T)$. This is also called economical SVD.

$$X \approx U_r \Sigma_r V_r^T. \tag{6}$$

Finally, any new image Φ can be represented as a linear combination of Eigenface basis:

$$\Phi \approx \hat{\Phi} := \sum_{i=1}^{r} w_i u_i, \tag{7}$$

for $w_i = \Phi^T u_i$. $\hat{\Phi}$ is called the reconstruction image since it is reconstructed under the new Eigenfaces basis. Also, note that Eq. (7) can be rewritten as $\hat{\Phi} = U_r U_r^T \Phi$.

3 Dataset

The dataset we use to demonstrate Eigenfaces is the Yale Faces Dataset b [2], which consists of facial images of 38 distinct individuals, each of whom has 64 images captured under different lighting conditions. Each image is of size 192×168 and the face is centered. For training we will use only the images of the first 36 people and leave the rest for evaluation. Figure 1 shows some examples from the dataset. Part of our implementation follows the examples from [1].

4 Applications and Results

4.1 Face Reconstruction and Detection

Suppose we have Person A and Person B and they both have the Eigenfaces matrix U_r stored locally on their device. Person A wants to send a large image Φ, which has been offset by the average face vector Ψ, to Person B. Recall from Eq. (7) that $\hat{\Phi} = U_r U_r^T \Phi$. If U_r captures most of the variance from the data, Eq. (7) will give a good approximation of Φ. Instead of sending over Φ directly,

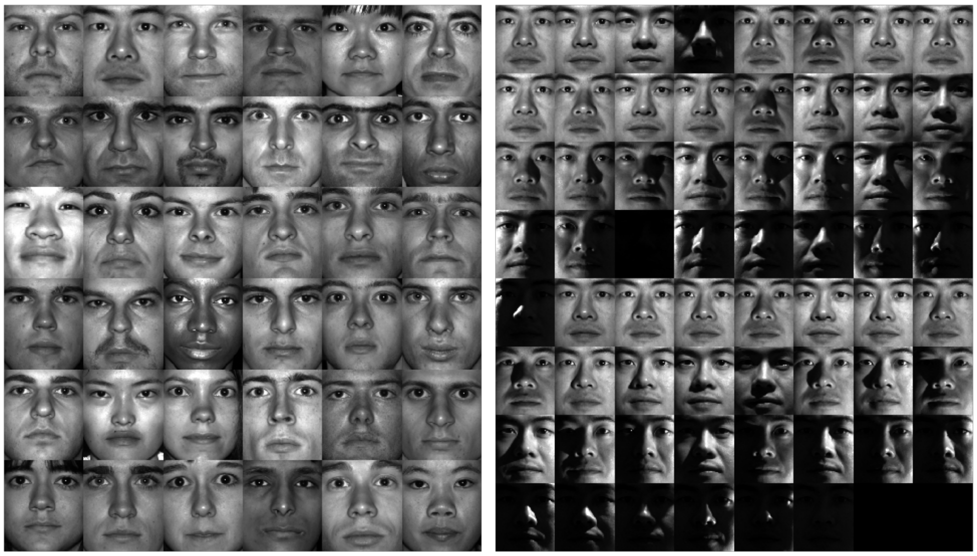

Figure 1: Left: 36 out of 38 distinct individuals in the dataset. Right: One individual under different lighting conditions

Figure 2: Left: mean face Ψ. Middle: the first Eigenface. Right: the second Eigenface

Person A may send over only $\alpha := U_r^T \Phi$, which is a vector of size r only. α serves as an "encoder" and can be interpreted as the weights for r Eigenfaces. Then, Person B can simply "decode" α and obtain a good approximation of Φ by computing $U_r \alpha$.

Comparing any original image Φ and its reconstruction image $\hat{\Phi}$ may also serve as a face detector. Suppose we have decided on a threshold T_d and an image Φ where we are not sure if it contains a face. Then, if $|\Phi - \hat{\Phi}| > T_d$ then it is not a face. The reason is that $\hat{\Phi}$ is a linear combination of the Eigenfaces; if Φ is not a face image at all, $\hat{\Phi}$ will be far away from Φ in the column space of U_r, which is also called the "face space".

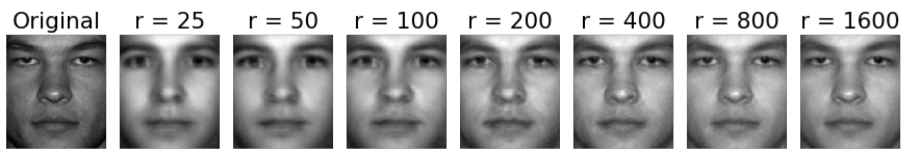

Figure 3: Reconstruction image $\hat{\Phi} = U_r U_r^T \Phi$ with varying r. As r increases, the reconstruction image offers a better approximation to the original image.

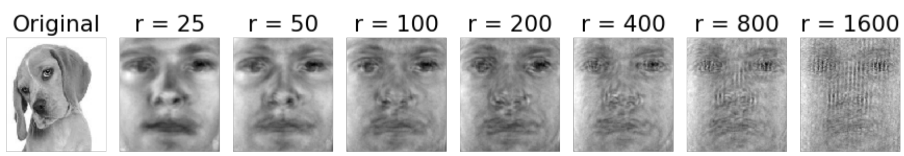

Figure 4: Reconstruction image $\hat{\Phi} = U_r U_r^T \Phi$ with varying r. However, as r increases, the resconstruction fails since the original image does not contain our Eigenfaces.

4.2 Face Classification

The idea of face detection from the previous section can be applied to face classification. Suppose there is a new face Γ. It can be transformed into its Eigenface components with the following operation:

$$\omega_r = U_r^T (\Gamma - \Psi) \tag{8}$$

where ω_r is a vector of size r describing each eigenface's contribution to representing the input face. Figure 5 illustrates a sample image and its projection onto the 5th and 6th Eigenface. After weights of the new image is obtained, they can be used to compare with the known face classes. The most straightforward method is to compute the Euclidian distance vector $d_r = [d_1, d_2, \ldots, d_r]$ between the weights of the new face and each known face classes' weights. Then the new face is classified as belonging to class k where d_k is the minimum distance in d_r.

The training and testing data for classification are chosen in a slightly different manner. The testing data are constructed by randomly selecting five faces from each individual (a total of 190 faces), and the rest goes to the training data (a total of 2220 faces). Each testing face is transformed to its Eigenface components as shown in Eq. (8). The mean weights for each individual are computed as the mean weights for all face classes. Classification of the test faces are attained by finding the face class where the Euclidean distance is minimized.

The choice of U_r^T has a significant impact on the accuracy. Figure 6 shows the first 22 Eigenfaces. The resulting accuracy on the testing data is merely 42.6%. By increasing the number of Eigenfaces to 61, testing accuracy increases to 60.5%. With 259 Eigenfaces, the testing accuracy bumps up to 73.7%. From our experiment, the accuracy will keep increasing as a larger subset of Eigenfaces are selected. Although an extremely high testing accuracy could be achieved by selecting an enormous number of Eigenfaces, one should be aware of the exponentially-growing computational cost associated with it.

Figure 5: Left: sample image. Middle: projection onto the 5th Eigenface. Right: projection onto the 6th Eigenface.

4.3 Drawbacks

However, there are some issues with Eigenfaces. One issue is its sensitivity to scale. If the size of the face from the testing images varies much from those on from the training images, the recognition performance would deteriorate. Therefore, a low-level pre-processing may be necessary for scale normalization. Also, during our analysis, all images used to train and test have a same orientation and no background. It also requires the face to be in the center of the image and point directly towards the camera. However, this condition is difficult to be satisfied in any real-world experiment.

Figure 6: First 22 Eigenfaces used in face classification. The resulting accuracy on the testing data is 42.6%

References

[1] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[2] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[3] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, January 1991.