Face Recognition using Eigenfaces

Berk Gokberk
CMPE58Z Introduction to Biometrics

Bogazici University

In this assignment, you are going to design a classical face identification system using the Eigenface approach [1]. Eigenface technique is used to extract *features* from face images. The steps of the Eigenface feature extraction method are as follows:

1. Given a set of M face images: $I_1, I_2, \ldots I_M$, *vectorize pixel values* to obtain pixel column vectors Γ_i. If the row and column size of an image I_i is $P \times Q$, then the size of Γ_i is $(P \times Q) \times 1$. Let R be $R = P \times Q$, for simplicity. You can vectorize a face image by simply concatenating each pixel row side by side. Γ_i is referred to as a face image from now on.
2. Compute the average face Ψ using M training faces.
3. Compute difference faces Φ_i by subtracting Ψ from each face Γ_i: $\Phi_i = \Gamma_i - \Psi$
4. Form a *training matrix* $A = [\Phi_1, \Phi_2, \ldots, \Phi_M]$ containing a difference face image at every column. The size of A is $R \times M$.
5. Define the L matrix: $L = A^T A$.
6. Compute the eigenvectors v_l of L.
7. Pre-multiply v_l with A to obtain Eigenfaces u_l: $u_l = Av_l$.
8. Select M' Eigenfaces, $u_1, u_1, u_3, \ldots u_{M'}$, having the biggest corresponding eigenvalues and form the Eigenface transformation matrix $U = [u_1, u_2, \ldots u_{M'}]$. It is useful to normalize eigenvectors so that they have unit length. Normalization is simply done by dividing each eigenvector to it's norm.
9. *Eigenface Projection:* Project (or extract the feature of) a given face image, Γ_i, into the Eigenface space by: $\Omega_i = U^T(\Gamma_i - \Psi)$.

Above, steps from 2 to 5 is called *Eigenface training*. The output of Eigenface training is the Eigenface transformation matrix U. Step 6 explains how to use U to extract Eigenface features from a given face image. The maximum feature dimensionality of reduced features is bounded by the number of images in the training set. During projection phase, Step 6, mean face is subtracted from the face image. For further details on the Eigenface technique, please refer to *Calculating Eigenfaces* section (pages 73-75) in [1].

It is possible to reconstruct the original face image, up to a certain quality, from the Eigenface features. To reconstruct the face image, $\hat{\Gamma}_i$ from Eigenface feature vectors, Ω_i, (or, say Eigenface coefficients), you have to perform a slightly different matrix multiplication as found in Step 9. Please find out the answer (that is, the reconstruction equation) by doing a little research. After finding the $\hat{\Gamma}_i$, by reversing the row concatenation operation, you can convert back to original image size to obtain $\hat{I}_i$.

In this assignment, you are a given a face database, the ORL face database, containing 40 subjects with 10 face images per subject. You will use the first five images of each person to form training matrix A in order to compute U. Using U, you will compute Eigenface features for all the images in the ORL face database. The remaining five images of every subject will be used as a *test set*. You are going to find the identity of images in the test set. Each subject will have the first five images as their *enrolled images*. Therefore, in our system, each subject is represented by their five Eigenface feature vectors. In order to compute the dissimilarity of two feature vectors, Euclidean distance will be used: $d(\Omega_i, \Omega_j) = \|\Omega_i, \Omega_j\|$, where $\|.\|$ denotes Euclidean distance.

In your report, you're going to provide the following:

1. Plot the first 20 Eigenfaces, as shown in Figure 1.

Fig. 1. First 20 Eigenfaces.

2. Reconstruct a sample face image from the test set using Eigenface features of size: $[2, 5, 10, 20, 40, 60, 100, 150, 200]$ and plot the reconstructed images together with the original face image. See Figure 2 for an example. Do the same thing for a *non-face image*, as in Figure 3. Comment on the difference

between these two cases in terms of the reconstruction ability of the Eigenface method

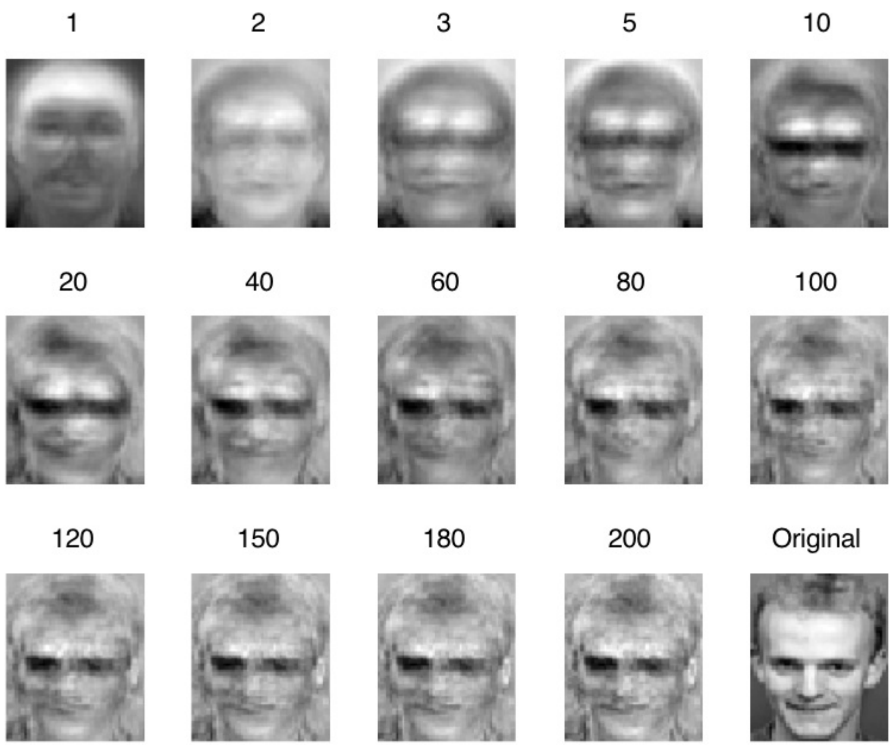

Fig. 2. Reconstructed face image.

3. Using your test set, plot the identification accuracy (y-axis) vs Eigenface dimensionality (x-axis) plot. See Figure 4 as an example. As Eigenface dimensionality (i.e., Eigenface feature vector size), you can choose $[5, 10, 15, \ldots, 200]$. In addition to this plot, provide some *identification rates* together with *Eigenface feature vector size* (i.e., selected Eigenface count) and the *total variance explained* as a table. See the format of Table 1 as an example. The total variance explained by selecting the biggest K eigenvectors is computed by $\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{R} \lambda_i}$ where R is the maximum dimensionality. Note that eigenvalues are sorted in descending order: $\lambda_i > \lambda_{i+1}$.

You are going to deliver your source code. The main script of your assignment should be named as `assignmentX.ext`, where X is the assignment number and `.ext` is the file extension of your programming language. When it runs, it should produce the 1) Eigenface images, 2) Reconstruction face example image

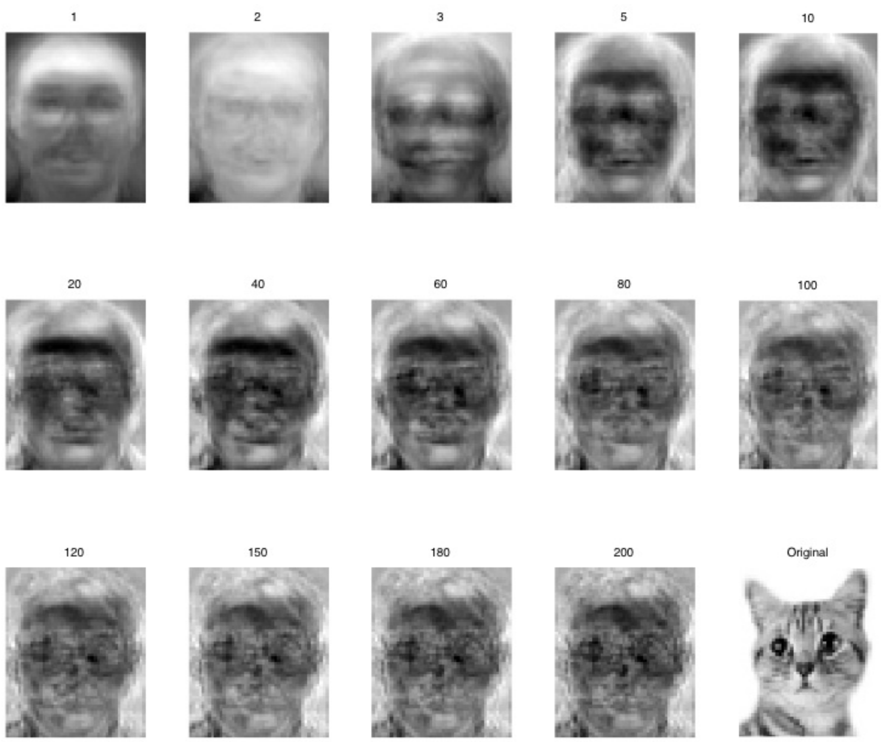

Fig. 3. Reconstructed non-face image.

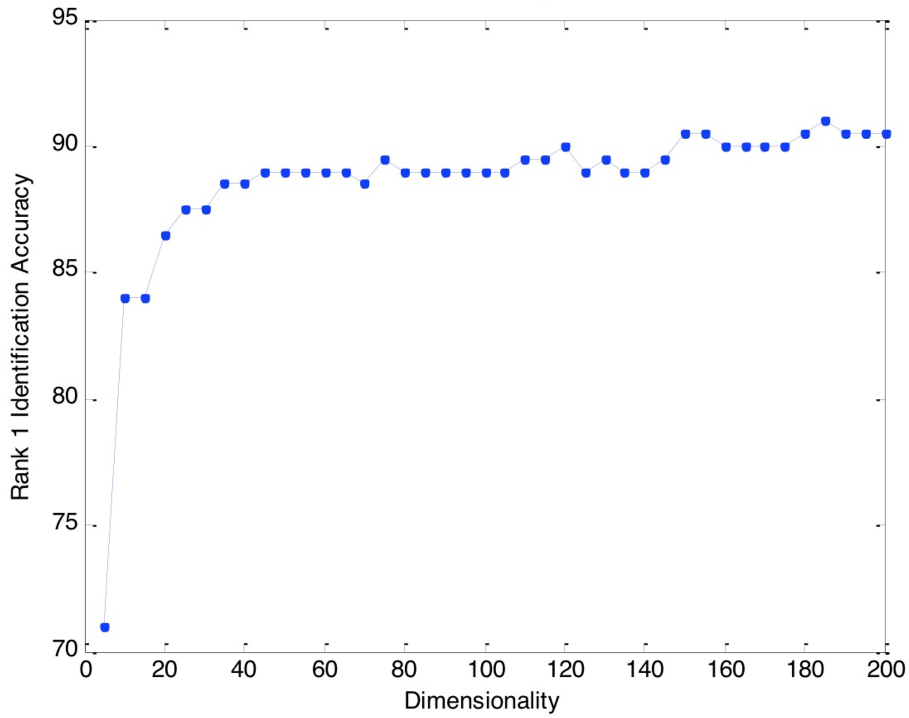

Fig. 4. Identification accuracy vs Eigenface dimensionality.

(only one face image reconstruction is sufficient), 3) identification rate for a specific Eigenface dimensionality. Your code is not expected to produce Eigenface dimensionality vs identification rate plots. You can have separate functions that are called from the main code as well. In that case, please submit all your files.

Tips

– For identification performance plots (accuracy vs Eigenface dimensionality), you do not need to re-compute eigenvectors/eigenvalues at each dimensionality. Compute them once and use the *required part* of the eigenvector matrix for feature extraction. This saves a lot of time.

References

1. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cognitive Neuroscience 3(1), 71–86 (1991)

Table 1. The performance of the Eigenface based face recognizer with respect to feature dimensionality.

Number of Eigenfaces	Variance Explained	Identification Rate
10	0.31	34.77%
50	0.53	67.23%
.	.	.
1000	0.99	78.03%