CS 479 Pattern Recognition
Programming Assignment 3 Dimensionality Reduction and Eigenfaces
Julia Adamczyk, Brendan Aguiar
A shared private Github repository, Git, and Google docs were used to collaborate in this project. Both team members contributed equally to the project and clarification of work allocation was documented to further show transparency where appropriate. The source code has a list of version history. A further breakdown of the division of work is shown:
Julia - matrix operations, logic
report: theory, discussion
Brendan - reading algorithm, logic
report: implementation, discussion, results, matches, Mahalanobis distance.

Principal Component Analysis

Theory

Principal Component Analysis (PCA) is a method of dimensionality reduction, which transforms large datasets to smaller ones while saving the most information of the large dataset. PCA uses eigenvectors and eigenvalues to determine the most expressive features, which have the largest eigenvalues of the large dataset and uses the corresponding eigenvectors to construct a subspace of a new, reduced dimension. This produces a new dataset which is much simpler, however, some accuracy of the data is lost in the reduction process. In order to perform PCA, the data needs to be centered at zero. Therefore the sample mean is calculated using the Eq. (1):

$$x_{bar} = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{1}$$

where M is the sample size and corresponds to every image. The mean simply calculates the average value of each pixel in the dataset. In order to center data at zero, each pixel of every image needs to be subtracted by the mean creating Φ. Then, the sample covariance is calculated according to Eq. (2):

$$\mathbf{\Sigma}_x = \frac{1}{M}\sum_{i=1}^{M} \Phi_i^T \Phi_i \tag{2}$$

where $A=[\Phi_1\ \Phi_2\ ...\ \Phi_M]$ i.e., the columns of A are the Φ_i. Then, eigenvectors and eigenvalues of the covariance matrix are computed using the Jacobi algorithm taken from Numerical Recipes in C [Press, 2002]. Lastly, the dimensionality reduction step is performed by approximating **x** (each image) using only the first largest K eigenvectors, U and eigen coefficients, Ω. This is done by reconstruction: projecting the image data onto principal components (K largest eigenvectors).

Each step of the Principal Component Analysis needs to be further developed in the context of application to images and face detection/recognition. In this assignment PCA is applied to train a large set of images. Then the performance of the reduced data set is evaluated using the facial recognition matches of an unknown image set that contain the same subjects. Steps for PCA are as follows:

1. Convert and normalize data from image files stacking rows into a single vector of size pixels, N to be used in the PCA reduction process.
2. Stack rows of the images M on top of each other to create an M x N matrix.
3. Calculate sample mean using Eq. (1).
4. Center data at zero by subtracting sample mean from each image, **x.**

5. Calculate covariance matrix for the data. Since matrix AA^T would become a N^2 x N^2 matrix and result in high computational complexity, calculate A^TA, a M^2x M^2matrix instead. It can be proven that matrices AA^T and A^TA have the same eigenvalues, and eigenvectors of AA^T can be computed from the Eq. (3), (4), and (5):

$$AA^T u_i = \lambda u_i \tag{3}$$

$$A(A^T A)v_i = \lambda A v_i \tag{4}$$

$$A v_i = u_i \tag{5}$$

In the equation shown, u_i is an eigenvector of matrix AA^T and v_i is an eigenvector of matrix A^TA.

6. Perform the dimensionality reduction step where each image is approximated by the K largest eigen coefficients, y_i using the centered image and eigenvectors shown in Eq. (6), (7):

$$y_i = \Phi^T u_i \tag{6}$$

$$\Omega^T = \left[y_1, y_2, y_3, \cdots, y_k\right] \tag{7}$$

In the equation shown, y_i is coefficient of projection, u_i is the corresponding eigenvector. The size of omega unreduced becomes M x M where the rows represent the images and the columns represent the eigen coefficients.

Implementation

The PCA is performed on a training set of images in order to produce the model that is to be reduced in the test set phase. Training mode outcomes needed in testing are the set of eigenfaces, U; coefficients, Ω; and mean, x_{bar}. After checking if reconstruction works, the training data can then be used to test facial recognition performance. In the test mode, the program uses a new set of the images with the same subjects as the training set. Like the training images, The test set images are centered around the mean and then projected onto the eigenspace, which computes their own Ω_i coefficients. The coefficients of test data and coefficients of training data are then compared using Mahalanobis distance to find the closest match. The Mahalanobis distance, S is shown applied to the coefficients in Eq. (8):

$$S = \sum_{j=1}^{K} \frac{1}{\lambda_j} (y_j - y_j^i)^2 \tag{8}$$

The distances are then normalized using their mean and standard deviation, shown in Eq. (9):

$$S_i = \frac{S_i - \mu}{\sigma} \tag{9}$$

The minimum distance S_i is then used to identify an ID that will correspond to the matching subject in the training data. If the ID and subject are a mismatch, then the recognition failed. The advantage of using minimum distances is that the N smallest distances can be used to compute a Cumulative Match Characteristic curve. To produce this curve, calculate the success rates of the recognition as the number of smallest distances N for each test image increases. The success rate should go to one hundred percent as the number of distances N for each image goes to the number of images in the training set.

Results and Discussion

The mean faces of the training sets are shown on a Fig 1. The image of the left corresponds to high resolution images and one on the right corresponds to low resolution images. Low resolution images produce an average face of worse quality, which can negatively affect the model performance and accuracy. The mean face is computed by adding each corresponding pixel from all images to each other and dividing by the number of images in a set. Therefore, the mean face is a construction of pixels that holds an average value based on the set of all images. The image is blurry, however, while still depicting the main face features like eyes, nose, or mouth.

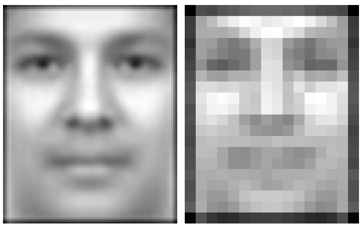

Fig 1. The mean face for high resolution (on the left) and low resolution (on the right) images is quite blurry but main face characteristics can be recognized.

The eigenfaces constructed from 10 largest eigenvectors are shown in Fig. 2. (high resolution) and Fig. 3 (low resolution). These eigenvectors capture the maximum variance of the data and bring out the most expressive features of the dataset.

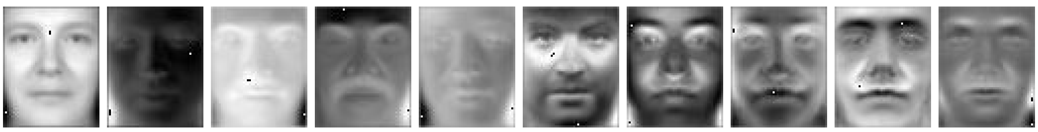

Fig 2. Eigenfaces corresponding to the 10 largest eigenvectors for high resolution images.

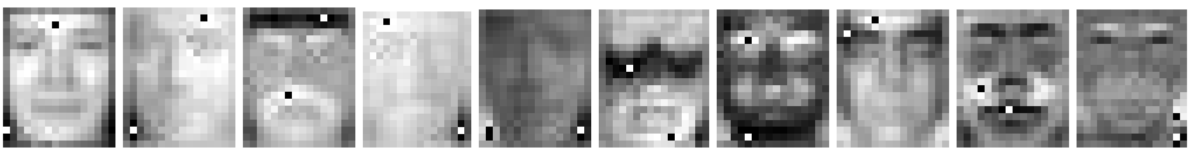

Fig 3. Eigenfaces corresponding to the 10 largest eigenvectors for low resolution images.

Fig. 4 and Fig. 5 show images created using eigenvectors corresponding to the smallest eigenvalues. These vectors correspond to the lowest variance and therefore can be rejected in the PCA process. According to PCA, the features with lower variance can be rejected because they do not convey as much data info. As shown in Fig. 3, the images corresponding to the smallest eigenvalues do not depict any features of a face. The images depict white noise that could not distinguish between objects, let alone faces, that look similar and would have an insignificant impact (eg. in the case of the same person but two different images).

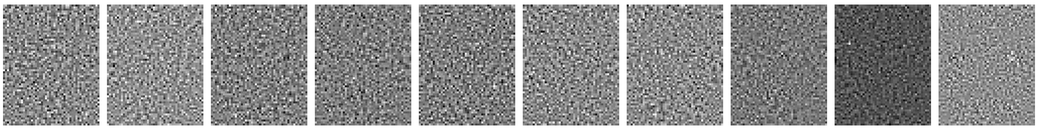

Fig 4. Eigenfaces corresponding to the 10 smallest eigenvectors for high resolution images.

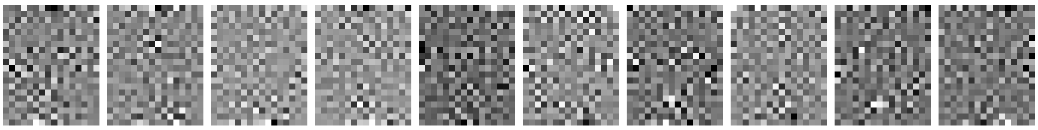

Fig 5. Eigenfaces corresponding to the 10 smallest eigenvectors for low resolution images.

Features represented by the smallest eigenvectors are less significant than features represented by largest eigenvectors. Using PCA to reduce the dimensionality of the dataset can be beneficial because it will improve the computational efficiency. The features that are rejected might influence the accuracy of recognition somehow. Applying PCA can also improve the generalization of the model.

Low resolution images produce an outcome of worse quality than high resolution images, which can negatively affect the model performance and accuracy. In the images of high resolution, the main face features can still be recognized while images produced from low resolution are blurry. Many of the low resolution images do not even look like a human face. Therefore, the test image of an object different from a human could be classified as a human face.

To test the performance of the training data, the ID of each minimum distance was recorded for each corresponding test image with a reduction of eighty percent. If the ID that corresponds to the test image matches the entry number of the training image, then a match has been found. Fig 6. shows three correctly matched images.

Fig. 6 The three matches are shown from the training set (left) and testing set (right).

Incorrect images were all matched to the first training image, which may reflect a bug in the code. Fig. 7. shows an example of an image from the testing set incorrectly matched up with the first image of the training set.

Fig. 7. The test image (left) is incorrectly matched with the first training image (right).

Although the first training image may have similar features to other subjects in the testing set, such as the subject shown in Fig. 7, all incorrectly matched images should not be matched to the first training image. Further analysis of the code and mathematical concepts behind the code will help to further understand what happened with the incorrect matches and how to make the error lower and more accurate. Regardless, the results of this experiment show that high resolution images are able to be used in recognition software. Lower resolution images may cause the results to be less accurate, but further research needs to be explored.