Assignment 2: Eigenfaces for Face Recognition

Submitted By: Jinay Dagli [20110084]

(I) Introduction

Our face plays a primary role in conveying identity and is the major focus in most human interactions. The human brain is very efficient in recognizing faces even when the conditions change, that is, even after a change in lighting, angle, or expressions. Computational models have evolved recently, that could recognize faces with considerable accuracy. Image recognition has very practical applications, such as criminal identification, healthcare, and education. In this assignment, a face recognition system based on eigenspaces has been implemented and tested.

The ATT Faces database has been used for this assignment. It consists of 40 different subjects with 10 images for each of the subjects.
The Google Colab file for the same can be found here. It consists of the final code for the assignment.

Alternatively, it has also been implemented on the Yale B database, which consists of a total of 166 images. But since it only consists of 15 subjects, it has not been used for this assignment. However, the link for the same has been attached here.

(II) Algorithm of Face Recognition Using Eigenfaces

As has been described in [1], the steps for face recognition using eigenfaces consist of the following:

1. An initial set of images is acquired (the training set). Generally, a number of images of each subject are taken in the training set.
2. A mean image is calculated and all the images in the training set are normalized by subtracting this mean image.
3. The Eigen values and Eigen vectors of these images are calculated. Only the important eigenvectors are chosen (which have the highest eigenvalues) and are used to form what is called the *face space*. This is what is called dimensionality reduction. As new faces are encountered, the eigenspace (or the face space) can be updated.
4. The weights of each of the training faces are calculated by projecting each face onto the face space.

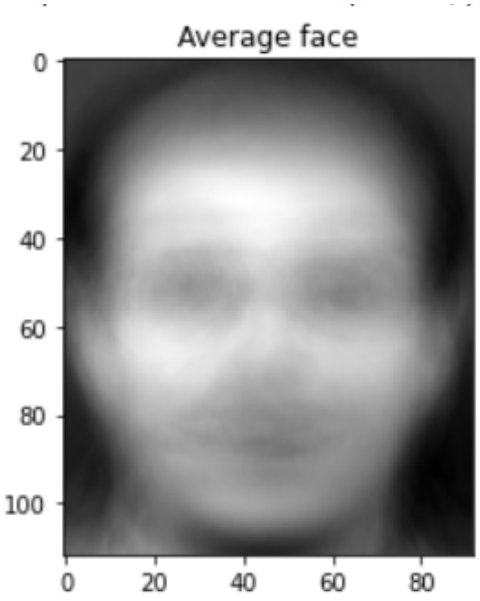
(III) Implementation of the algorithm:

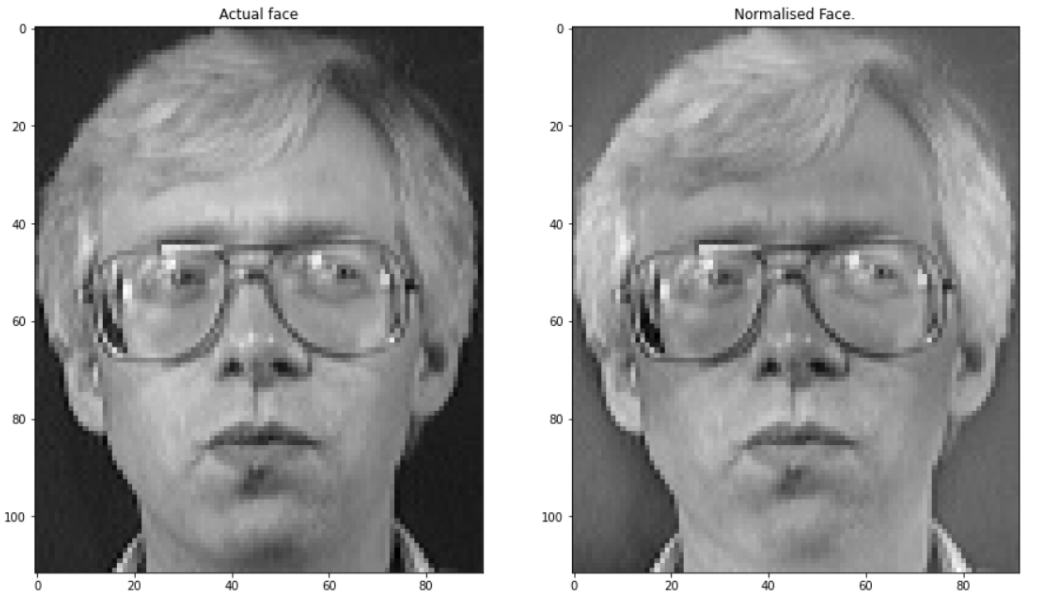
The link for the code can be found in the Introduction section above.
The following steps have been implemented for the same:

1. The database is split into training and test sets in the ratio of 60%:40%. That is, for every subject, 6 images are taken in the training set (as described in the paper ([1]), at least some images need to be taken in the training set). The first six images are taken in the training set.

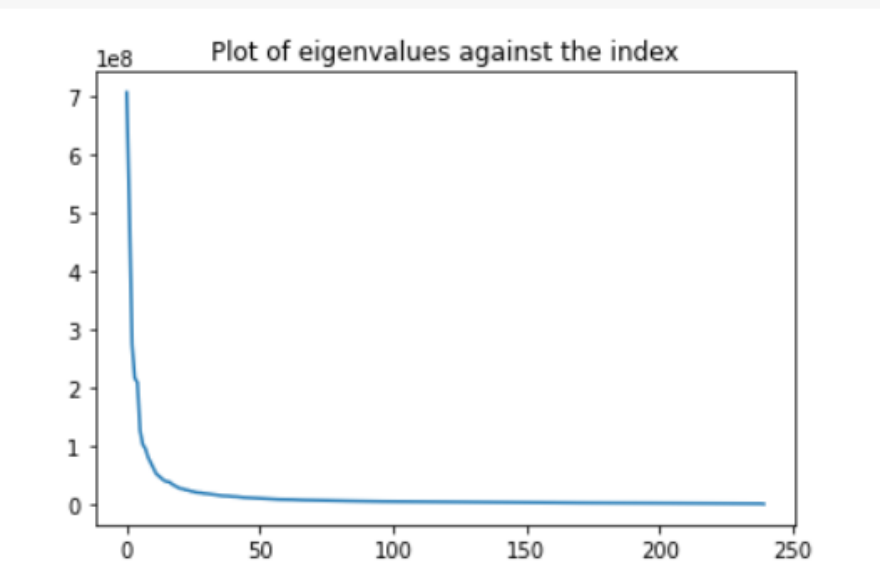
2. The training set matrix and test set matrix are being formed using the images in the training set and test set respectively.

3. After the training set is formed, an average face is calculated by summing over the faces in the training set and dividing it by the length of the training set. The faces in the training set are normalized by subtracting the average face from each of the faces. Let us call the array of these normalized faces 'A' (as mentioned in [1]). For example, the mean face and normalized face for one of the images are shown below:





4. The covariance matrix of these images is calculated by the dot product of A and its transpose. This covariance matrix is then used to calculate the eigenvalues and eigenvectors. It is sorted in decreasing order, and only the absolute value of these are taken (to take care of the complex values!)
   The plot of eigenvalues is shown below:

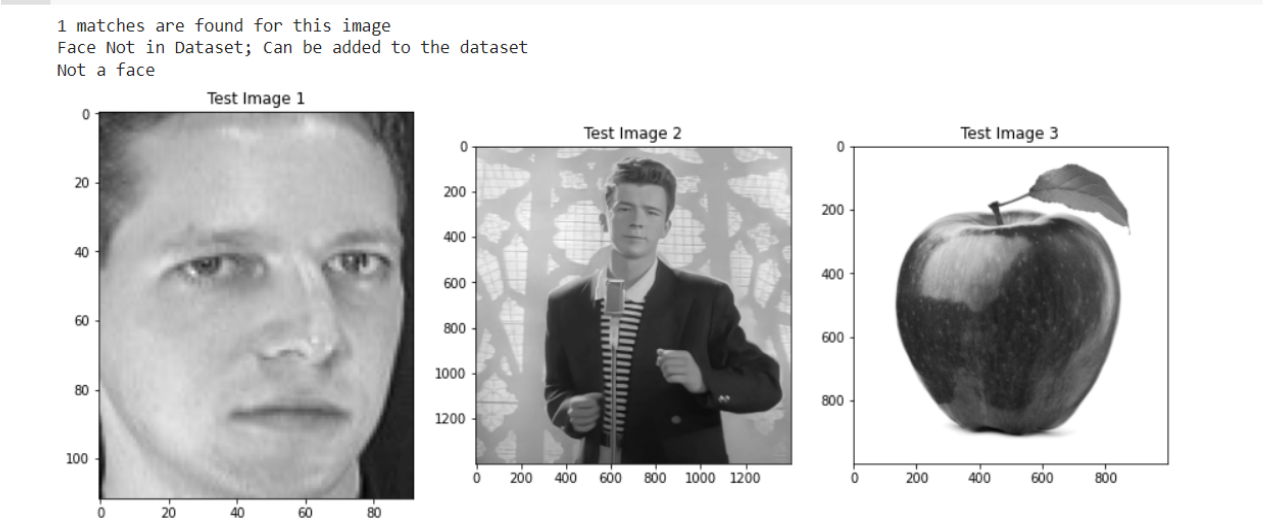


5. On plotting the eigenvalues, we observe that only some of these images (m=50 for the ATT Dataset) are important. These 50 eigenvectors are used to form the *face space*. The

images are projected onto the face space to get the eigenspace. Its dot product with the normalized faces gives us the weight matrix.

6. Now, when a new image is encountered, its corresponding weights are calculated and its Euclidean distance with the eigenspace and the different face classes is calculated. Based on two thresholds, it is classified as either of the following: Face belonging to one of the subjects in the dataset; Unknown face that could be added to the dataset; and Not a Face.

All three cases have been observed for the same thresholds (an image could be found below). It could also be found on the Google Colab file.



(IV) Results and Inference:

For the ATT dataset, when 60% of the dataset is taken as the training set, an accuracy ranging between 50%-70% is observed on the test set with changing thresholds. The ones which are predicted incorrectly are because of a mismatch between the subject names. The thresholds are taken by observing the minimum norm of any one unknown face. When observed for the mentioned thresholds, we get an accuracy of nearly 68.75 %.
For the Yale B database as well, when 80% of the dataset is taken as the testing set, we observe an accuracy ranging from 60%- 80% with changing thresholds.

We would expect that if we increase the size of the dataset, the accuracy would increase as well!

(V) References:

[1] M. Turk and A. Pentland, "**Eigenfaces** for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[2] M. Turk and A. Pentland, "Face recognition using eigenfaces," *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586-591, doi: 10.1109/CVPR.1991.139758.

[3] PCA Algorithm: Link

[4] GitHub Repo1: Link

[5] GitHub Repo2: Link