Computer Vision

Assignment 2:
Face recognition using Eigen Faces

Maastricht University

Department of Data Science and Knowledge Engineering

Authors

Georgios Patsiaouras, i6198785

1 Eigen Analysis

As it was requested we performed the eigen analysis based on the data from the dataset. In order to calculate the eigen vectors and values we used the principal component analysis algorithm. The process followed was:

- Standardize data by dividing with 255
- Calculate the mean of all faces
- Subtract all faces from mean to find the unique characteristics
- Calculate the covariance matrix for all features
- Calculate eigen values and eigen vectors
- Sort principal components by the high one based on the eigen values
- Reduce dimensionality by taking the k first

The result of printing the first 25 eigen faces is shown in figure 1.1.

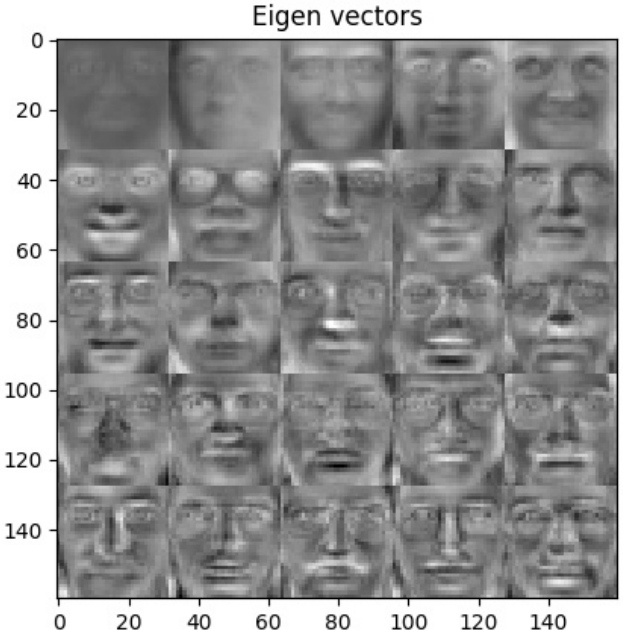

Figure 1.1: Eigen Vectors

2 Projection of training and test data

Training and test data were projected on the new space by taking the dot multiplication of the eigenvectors and the subtracted data from the mean image. In this way we retrieve the w values (projections) for each image. We did that for k=10,20,30 and plotted the results. In figure 2.1 we can see that starting from the mean face by adding the multiplication of the projected values and the eigenvectors we can augment the mean face with the special characteristics of a person. Thus adding this special features we can go from mean face to a specific person. As it was expected higher k values are giving more detail and a face closer to the actual one.

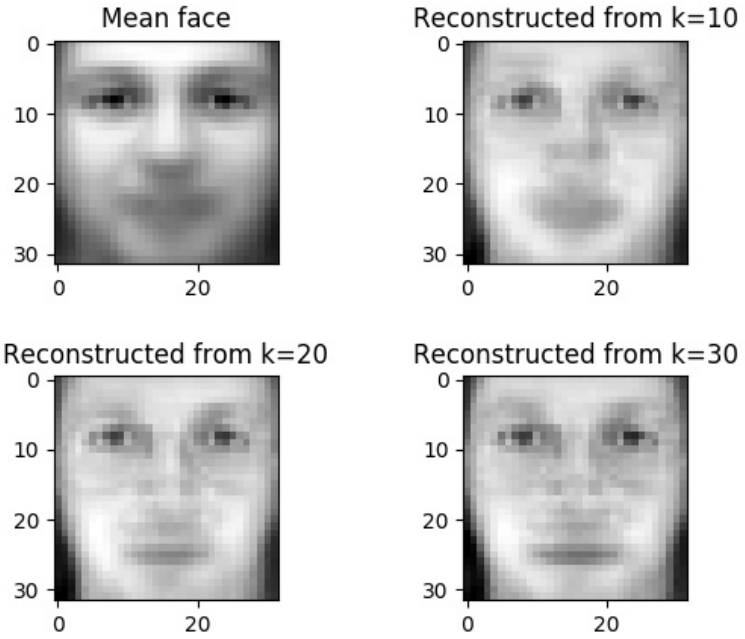

Figure 2.1: Reconstruction of the first image on the dataset by adding the mean face and the projected values of the first 10, 20 and 30 features.

As a result the k used in the next experiments will be 30 to achieve promising results.

3 Nearest Neighbors Algorithm to perform face recognition

In this section we are requested for a certain test set of faces to find which face from the training set is closest. We do this procedure by running pca for both the training and test set to project the images to the new space and then we take the distances of each component of the test set and the training set. After we find the distances we sort and we keep only the ids of the images

of the training set that are closer. We do that for every element of the test set and we return a vector of the same length containing which id of the training set is closer to the specific element of the test set.

Afterwards, we are calculating the accuracy by checking the identities of the people in the images from the labels dataset. Unfortunately, this part of the program doesn't work as it should thus we are achieving accuracy of 15-20%, meaning that most of the predictions are wrong. As we can also see in figure 3.1 almost all of the pictures are predicted wrong.

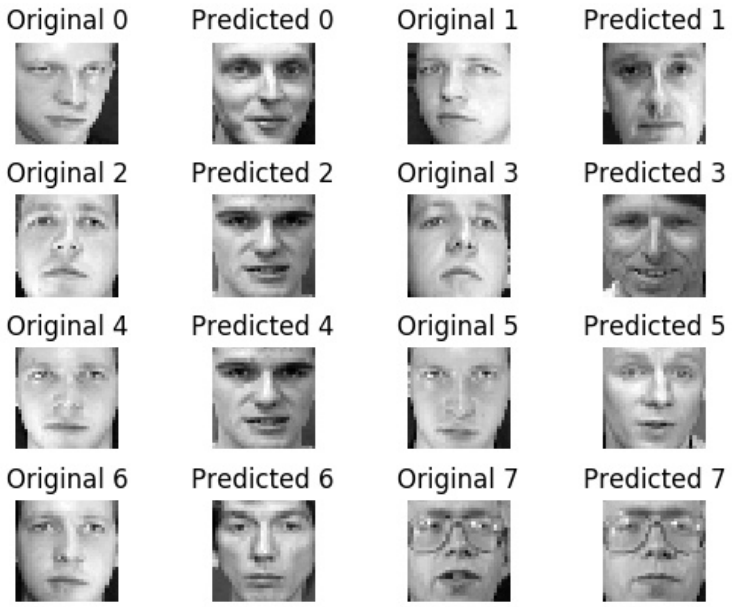

Figure 3.1: Test set images and their corresponding recognized image based on the Nearest Neighbor algorithm. These are the 8 first