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.

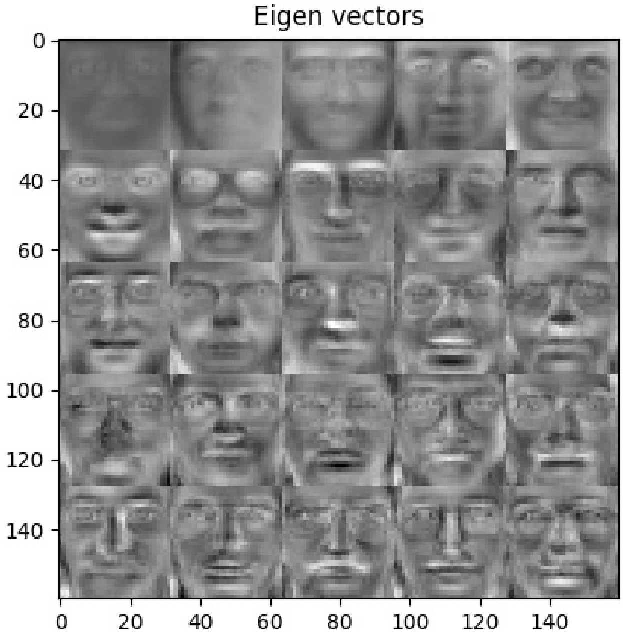

Figure 1.1: Eigen Vectors

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.

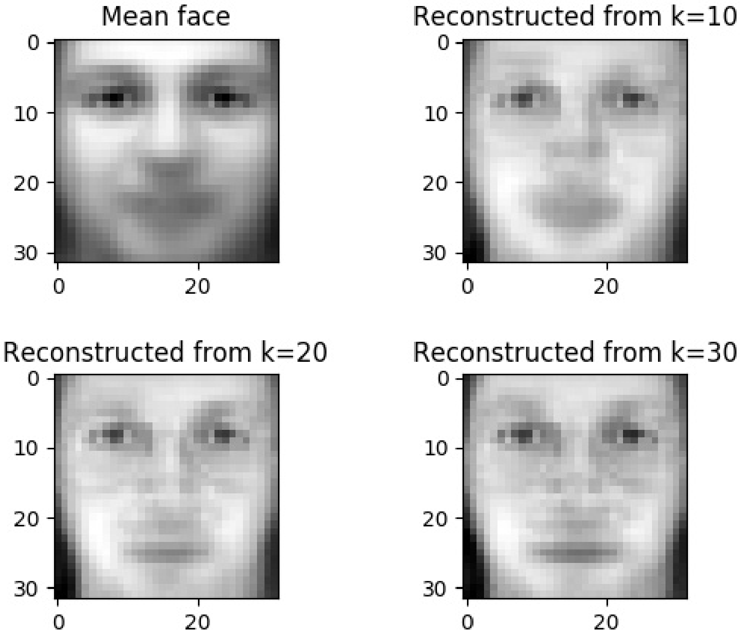

Figure 2.1: Reconstruction of the first image on the dataset by adding the mean face and the projected values of the first 10, 20 and 30 features.

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 closer. We do this procedure by running pca for the training set, receiving the eigen vectors and the mean of the trained data. We subtract the mean from the test set and then we project the test set on the space created by the trained set by taking the dot multiplication of the eigen vectors of the training set and our test set. After that by applying Nearest Neighbor algorithm, we are calculating the distances of the projected training and test set and we take the minimum ones for each element of test set.

The results can be found in figure 3.1. Original images and predicted are connected with the id {0,1,2,...,7}. For plotting we just take the first 8 example of the test set. The accuracy for this particular example was 76.79% and k 30 running on training set 3.

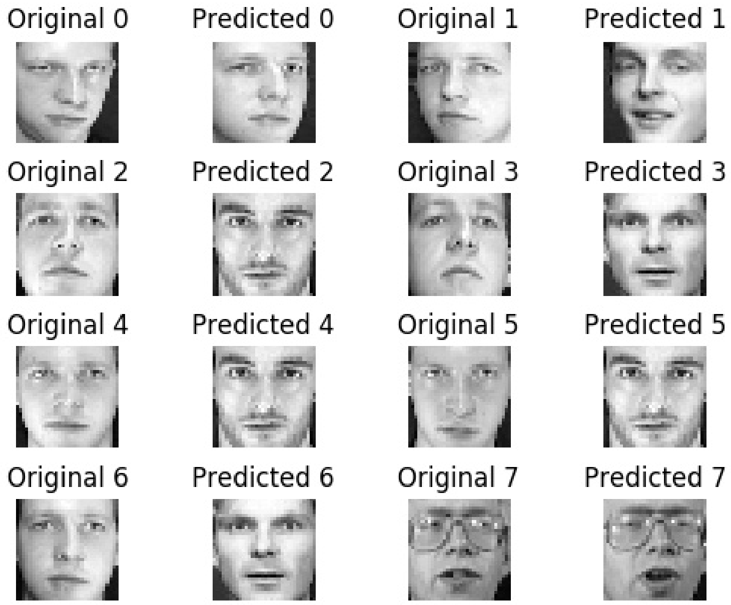

Figure 3.1: Test set images and their corresponding recognized image based on the Nearest Neighbor algorithm. These are the 8 first. k=30, train$_set = 3$

Finally, we are running also an experiment that calculates for a given set of k values the accuracy for each of the training sets 3, 5 and 7. The table 3.1 is holding the accuracy values for each of our combinations. For better understanding the figure 3.2 presents a plot with the table information.

K values	Accuracy on 3	Accuracy on 5	Accuracy on 7
2	0.260714	0.295	0.258333
12	0.689286	0.845	0.816667
22	0.742857	0.875	0.858333
32	0.764286	0.895	0.875
42	0.760714	0.89	0.883333
52	0.764286	0.895	0.875
62	0.760714	0.89	0.875
72	0.764286	0.89	0.875
82	0.767857	0.89	0.875
92	0.778571	0.89	0.875
102	0.782143	0.89	0.875
112	0.782143	0.89	0.875
122	0.775	0.885	0.875
132	0.775	0.89	0.875
142	0.775	0.89	0.875
152	0.775	0.895	0.875
162	0.775	0.89	0.875
172	0.775	0.895	0.875
182	0.775	0.89	0.875
192	0.775	0.895	0.875

Table 3.1: Accuracy achieved for k values on the 3,5,7 training sets

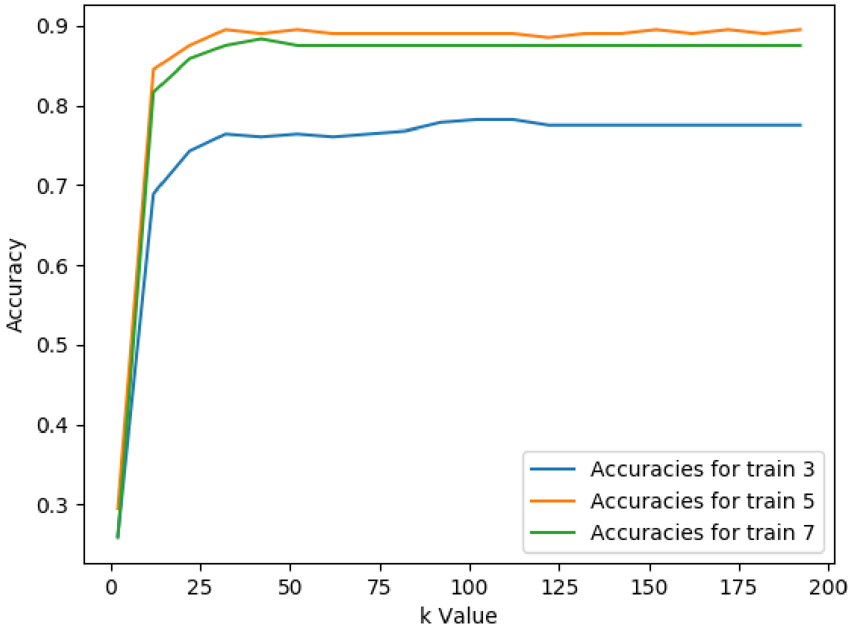

Figure 3.2: Accuracy plotted for different values of k and different sets of data