Computer Vision
G22.2271-001
Assignment 3.

March 31, 2011

1 Face Recognition

In this assignment the goal is to implement a nearest-neighbor recognition algorithm, using the Eigenface representation, presented in Lecture 8.

Download the `faces.zip` file from the course webpage. This contains two Matlab files: `ORL_32x32.mat` and `train_test_orl.mat`. The former contains 400 faces from the Olivetti Face database, each being 32 by 32 pixels grayscale (in variable `fea`) and the label of each image (in variable `gnd`). The latter contains a set of indices to be used for training and testing.

Open Matlab and load the two files into memory. You can view the files with the following command: `figure; montage(reshape(fea'/255,[32 32 1 400]));`. To implement the algorithm, your code should be structured as follows:

1. Split `fea` and `gnd` into training and test sets using the indices in `trainIdx` and `testIdx`. Scale the images so that they range from 0 to 1.

2. Center the training data, so that the per-pixel mean of across all images is zero.

3. Form C, the 1024 by 1024 covariance matrix.

4. Compute the first K principal components v of C using the `eigs` function, e.g. `[v,d]=eigs(C,K);`.

5. Plot out these principal components. They should look like the Eigenfaces in the slides.

6. Now project the centered training data into the PCA space using the principal components, yielding descriptors p.

7. To get a sense of what the model has captured, form the reconstruction of the face by projecting back into the image space using p and v. Do not forget to add the mean face back on again.

8. Now center and project the test data into the PCA space, giving descriptors q.

9. Perform a nearest-neighbor search for each of the descriptors in q to find the closest Euclidean descriptor in p. Assign the test image belonging to the query descriptor the label from this closest training image.

10. Measure the fraction of test images correctly classified.

11. Now repeat the whole scheme, varying the value of K. Plot a graph of the classification rate as a function of K.

You should turn in: your Matlab code, plot of classification rate vs K, a plot of the reconstructed training faces for $K = 20$.

For bonus points, extract some 32 by 32 patches from non-face images and try projecting them into face space with different values of K. Compare the reconstruction error to that of face images. Is there some threshold you can set that enables you to reliably tell if an image is a face or not?