Coursework 2 – Face Recognition using Eigenfaces

3. completing the code

1. Eigenface computation:

The first task required to compute and show the first 20 eigenface images, there are many possible ways to extract the top eigenvectors (matching high-to-low eigenvalues ranking), we could compute the eigen-decomposition of the covariance matrix, or manually solve the system :

$(Cov – I * eigenvalue_i) * eigenvector_i = 0$ for each eigenvector.

However the following property will allow an easier path:

1. The right eigenvectors of the svd of a matrix A are the eigenvectors of A* AT, and svd naturally orders the singular values in the correct order.

2. If the centered vectors of input image vectors are matrix A, the covariance matrix for the images is computed as A* AT , hence the SVD of A computes the eigenvectors of the covariance matrix.

This means that we can simply slice the vector V(1:20,:) to obtain the desired eigenvectors. **This is done in l46-47.**

After obtaining the eigenvectors, they need to be renormalised (which is provided in seperate renormalise.m function), the mean face added to them, then renormalised again. (taking care with the types that no information is lost due to overflow). Finally the vectors are reshapes into a 2d 28x23 image. **This is done in l50-60 for each eigenvector face.**

The output looks like slightly deformed faces, being less recognisable than in the examples in lecture since the images are very low-resolution.

2. Something went wrong in the code which has led to a 0 recognition rate. Something went w, and the result is nil **l 121-161**

3. In order to use knn to classify one has to cluster the projections according to different values of k , each cluster center would ideally correspond to one of the classes of real people.. it is logical then that the optimal k is 40, however due to technical difficulties we are unable to check this.