Australian National University
Research school of Engineering

ENGN4528/6528 - Computer Vision
Computer Lab 4: Eigenface Decomposition for Face Recognition

Objectives:

The goal of this computer laboratory is to help you have hands-on experiences on Principal Component Analysis (PCA) based method for the face recognition. The PCA is also called the Eigen-face decomposition in the context of image processing.

Notes:

- There is only one task in this lab.
- Your marks will be determined based on both your lab demo and your lab report. Please note that the level of difficulty for this lab (and CLAB5) is higher than the previous ones. Tutors will take this into account when they mark your results/report.

The Yale Face Database

In this work, we will have a look at some simple techniques for object recognition; in particular, we will try to recognise faces. The face data that we will use is derived from the Yale Face Database[1] (To get more information on the database, have a look at the website at UCSD. The original Yale website became obsolete). In short, the database consists of 150 images of 15 individuals, each under 10 different expressions. In this lab, these faces are included in the file ***yalefaces.zip*** in Wattle. You will find the faces are divided into 2 different folders – /trainingset and /testset. The training set contains 135 face images as some of them shown in Figure 1, and the test set contains 10 images (1 image sampled from 10 individuals) as in Figure 2.

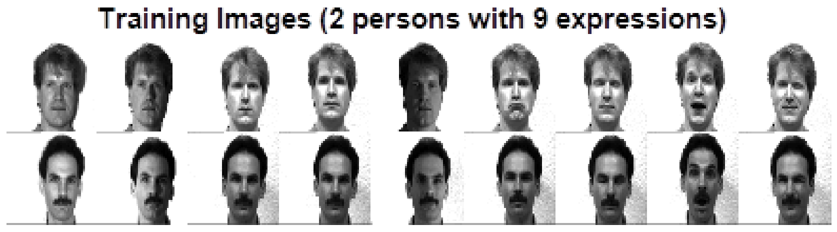

Figure 1. Training face images from 2 individuals with 9 expressions

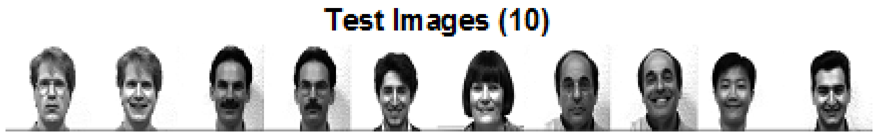

Figure 2. 10 test face images

[1] http://vision.ucsd.edu/content/yale-face-database

Task: Face recognition using the Eigenface technique

1. Dowload the Yale face dataset in Wattle.

   a) Download the Yale face dataset, ***yalefaces.zip***, from Wattle. Note that the images are in ‘gif’ format but do not have the ‘gif’ file extension, and thus not directly viewable in Windows (you can change the file name with ‘.gif’ extension). Instead, a Matlab sample code, ***viewyalefaces.m,*** is provided which can read and display them in tiled format.

   b) Read all the 135 training face images, vectorise each image as a single column vector, and collect all vectors as a big data matrix ***A***. You can use ***reshape***() to convert the matrix into a vector, or simply typing ***I***(:) will return the vectorised form of an image matrix ***I***. If you have resized the image as for example 100x100, then the matrix ***A*** would have the size of 10000x135 (the number of pixels *times* the number of images). Don’t forget to remove the mean vector from the matrix ***A***.

2. Perform Principal Component Analysis (PCA) on the data matrix

   a) For the covariance matrix $\boldsymbol{C} = \frac{1}{M}\boldsymbol{A}\boldsymbol{A}^{\boldsymbol{T}}$, with M being the number of training images, determine the top k-principal components, display the top *k*-eigenfaces on your screen (and also include in your report). *k* can be chosen at *k*=10 or *k*=15.

   b) Project each training image onto the space spanned by the top *k*-eigenfaces.

   c) Repeat this for each test image, projecting onto the space spanned by the top *k*-eigenfaces. Use this projection and Euclidian distance metric, perform a nearest-neighbour search over all the 135 faces. Find out which three images are the most similar to the test faces. Display these top 3 faces next to your test image on screen (and in your Lab report).

3. Test the face recogniser on your own face image.

   a) Read in one of your own frontal face images. Convert it to grey-scale, manually crop out the facial region out, and resize as a new image file: "my_face_cropped.jpg”. Then run your face recogniser on this new image file. Display the top 3 faces in the training folder that are most similar to your own faces.

   b) You may also wish to repeat this experiment by pre-adding your own faces into the training dataset. Of course, make sure your test face image is a different one - that is not included in the training set.

4. Some Notes:

   - Before doing anything, you must make sure that all face images must be geometrically aligned. Specifically, you need to take into account of the differences in position of the face in each image. A simple way is, before you do any training or testing, you should manually crop the face region out, define a standard window size, resize the face image, make sure the face region are all aligned - e.g. eyes, noses, mouths are roughly at the same positions in an image - save the results into disk, so you don't have to do the above pre-processing more than once.

   - In doing eigen-decomposition, always remember to subtract the mean face and, when reconstructing images based on the first *k*-principal components, add the mean face back in at the end.

- You can Matlab's ***eigs***() or ***svds***() functions to implement PCA. Other than this, you should not use Matlab's inbuilt eigenface function if there is one.
- If you encounter some difficulty in solving the eigenvalues of the covariance matrix $\boldsymbol{AA^T}$, think about whether or not you can use a faster way to compute them using $\boldsymbol{A^TA}$. Read lecture notes if you do not know what I mean by this faster method.
- If necessary, refer to the Turk and Pentland's original paper on the Eigenfaces in Wattle

Report Requirements:

Apart from including most of your representative results obtained in above experiments in your Lab report, you also need to answer the following questions in your Report:

Q1. For your own face image, how much of the energy is captured by the first k principal components (in percentage %)? Plot a curve of the energy percentages as a function of k, for k = 2, 3, 4, 5, 6 , 7, 8, 9,10. The energy is defined as the sum of the squared coefficients projected onto the eigenface space.

Q2. How many principal components are necessary to obtain a visually recognisable face reconstruction? That is, if you take an image and project it onto the space spanned by the first *k* principal components, how big does *k* have to be before the images look recognisable? You may try this for k=2, 4, 6, 8, 10, … and display the reconstruction results.

============= End of CLAB ===========