ECE3093 Semester 1, 2013
Assignment 1, Part B (worth 5%)
Due: Friday 19th April by 5pm
Assignment box, ground floor, building 28
(Clayton student only; Malaysian students please check with Dr. Kamrani)

Task 3: Age Estimation (3 marks)

Download the zip file ages.zip which contains two directories (called Train and Test) containing images that can be used for training and testing an age estimation system.

Data: There are in total 982 face images in the "Train" directory, and 20 in the "Test" directory. Each image is named by the following pattern:

xxxAyyz.jpg

where "xxx" is the person ID, "yy" is the age of the person in the image, and "z" is an optional mark only used when both xxx and yy are same. The age of each image can therefore be extracted using the 5th and 6th characters of the filename.
The face images are already normalized (non-face regions blackened out, and resized to 69 x 45 pixels) and are ready to be directly used in the algorithm.

Your task:

1. Modify the eigenfaces.m file (from Assignment 1, Part A) to train an age estimation system on the images in the "Train" directory, and then estimate the age of the images in the "Test" directory. The required code is very similar to the provided code for eigenfaces, which you will need to fully understand before modifying for age estimation. This will involve:
    a. Reading the training images into a matrix (one image per row);
    b. Performing PCA on the training matrix;
    c. Selecting the number of principal components you will use (k);
    d. Projecting each image in the training set onto the k principal components, and storing the projection vector, one for each image, as a feature vector for the image;
    e. Modelling the relationship between the feature vector **x** and the known age of an image using a quadratic function $age = \mathbf{w}_1^T(\mathbf{x}^2) + \mathbf{w}_2^T\mathbf{x} + b$, where $\mathbf{w}_1$, $\mathbf{w}_2$, and b are unknown parameters to be learned from the training set ($\mathbf{w}_1$ and $\mathbf{w}_2$ are vectors, and b is a scalar). $\mathbf{x}^2$ is the vector containing the squares of the elements in the feature vector **x**. To determine $\mathbf{w}_1, \mathbf{w}_2$, and b, you will need some polynomial regression techniques (see hint below).
    f. Once you have a trained quadratic model for age estimation, apply the test images (read in the test images into a matrix; project test vectors onto k principal components to get feature vector **x** for each test image; don't forget to subtract off the mean of the training images)
    g. Substitute **x** into the aging function and regard the output of the function as the estimated age of the test image.

h. Record the estimated age for each test image. Then calculate the Mean Absolute Error (MAE) by

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|a_n - \hat{a}_n|$$

where a_n is the real age of the *n*-th test image, and $\hat{a}_n$ is the estimated age of the *n*-th test image. *N* the total number of test images (here *N*=20).

2. Input your own (pre-processed) face image into the system (subtract off the mean image and project to obtain the feature vector) and see what the estimated age is (Don't worry if the output is inaccurate because this is only a simplified system. For a reasonable performance, you would need a much more sophisticated system).
3. Write a brief report containing:
   a. Your modified code; (1 mark)
   b. Analysis of the best k value to choose from the training data, and the resulting MAE for the 20 test images; (1 marks)
   c. Your image (suitably pre-processed to match the training images – cropped, face-region only, 69 x 45 pixels), and the error in your predicted age; (0.5 marks)
   d. A brief discussion (one paragraph) about the limitations of this system, and what could be done to improve the accuracy and usability. (0.5 marks)

Hint for quadratic function approximation:

Suppose "**a**" is a column vector storing all the ages of the training images (the i-th element stores the age of the i-th image). **X** is the matrix storing all the feature vectors of the training images (the i-th row stores the feature vector of the i-th image). Then the unknown parameters can be calculated using the Matlab command:

```
w=[X.^2, X, ones(size(X,1),1)]\a
```

Note that "\" is not the divide operator "/". Suppose the feature vector **x** is of the dimensionality *n*, then the first *n* elements in **w** compose $\mathbf{w}_1$, the next *n* elements in **w** compose $\mathbf{w}_2$, and the last element in **w** is b.

Reference

A. Lanitis, C. J. Taylor, and T. Cootes, "Toward automatic simulation of aging effects on face images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 4, pp. 442–455, 2002.

Task 4: Image Compression (2 marks)

This task builds upon Task 2 from Assignment 1, Part A. Suppose we have an image file of $m \times n$ pixels stored in the matrix A. The matrix A can be approximated using the outer product expansion of the SVD as:

$$A \approx A_k = \sum_{i=1}^{k<r} \sigma_i u_i v_i^T$$

where r=rank(A), σ_i are the singular values of A, and the u_i and v_i are the singular vectors corresponding to each σ_i. Let us denote by $s(A_k)$ the storage requirements for the image A_k measured as the total numbers that need to be stored to produce the image. The approximated image A_k will require the storage of less information (numbers) than the original image A, so $s(A_k)<s(A)$ for $k \neq r$. The compression rate that can be achieved by image A_k is defined as $100(1 - \frac{s(A_k)}{s(A)})$. Suppose we define a high compression rate to be 95%, a medium compression rate to be 90%, and a low compression rate to be 70%. An example is provided below:

Original low compression medium compression high compression

Your task:

a) Take a high resolution image of yourself, and save it as <surname.jpg>. Using MATLAB (modifying your code from the union jack assignment task), perform singular value decomposition of this image. Plot the singular values, and determine the smallest value of k for which you expect that the approximated image would still look like the original image. Submit your code, as well as the original image, the approximated image (stating the k value), and the plot of the singular values.

b) Derive an expression for $s(A_k)$ based on the SVD structure, and determine mathematically the closest integer k needed to generate images corresponding to low, medium and high compression rates. Submit your mathematical solution for each k, and the corresponding approximated image.

Hint: note that the imread function creates 3 matrices (red, green, blue), and if you are using a colour image, you need to do SVD separately for each and then combine the resulting low rank approximations into a colour image at the end. If using gray-scale images, the red, green, blue matrices will all be identical and you can take any of them (e.g. img(:,:,1)).

END OF ASSIGNMENT