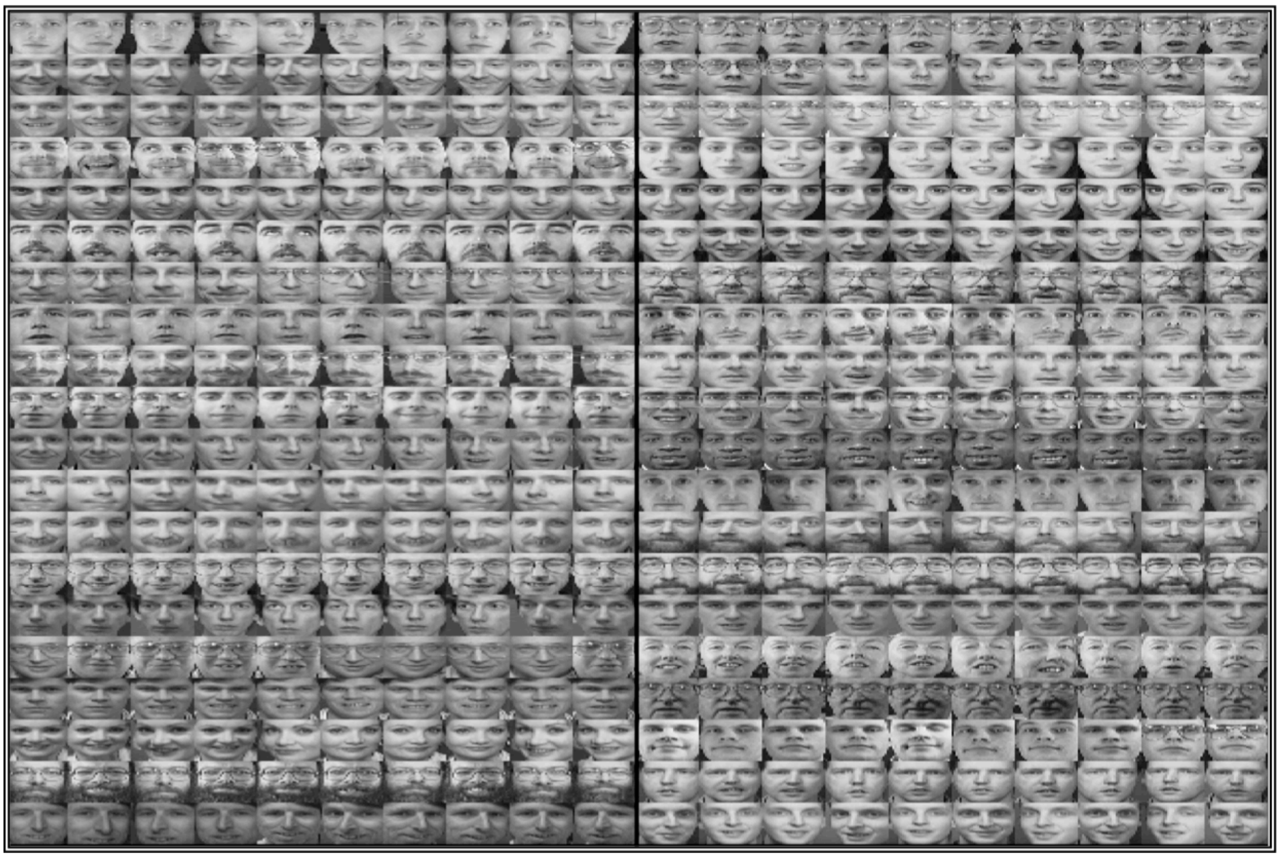
COMP24111 – Exercise 2: *Face Recognition*

Deadline: See website.

Extension policy: No extensions will be possible.



You need to do 3 things:

1. Download a zip file.
2. Read the help text and experiment with the utility functions provided.
   Then, and ONLY then ...
3. Do the lab exercises.

Download a zip file:
https://personalpages.manchester.ac.uk/staff/tingting.mu/Site/Teaching_files/comp24111code_ex2.zip

In this zip file you will find two types of files as listed in the table.

4 files with extension ".m": **`ShowFace.m`** **`ShowResult.m`** **`PartitionData.m`** **`knearest.m`**	These are the utility functions that you will use in this mini project. Use the help command to find out how they work, or look at the code itself – be sure to know what arguments they take, and what they return, so you **`don't overwhelm the TAs`** with trivial information that you could see by just typing, for instance **`help ShowFace`**
1 file with extension ".mat": **`ORLfacedata.mat`**	This is a data file, containing the data you will use for the project. If you have unzipped this, start Matlab. Then, use the Matlab command line (not the Unix command line) to navigate to the appropriate directory where these files are. Now type: **`load ORLfacedata`** This will have loaded two variables into memory: **`data`** and **`labels`**. These are samples of some face data, including 10 different images of each of 40 distinct subjects. The **`data`** matrix of 400 rows and 1024 columns contains the actual images, and the **`labels`** variable containing integers states which subject the face belongs to.

Preparation

As a starter, try the following:

```
load ORLfacedata;
X = data(1:20,:);
ShowFace(X,5);
```

This has pulled out the first 20 rows, and all columns from the face data matrix. There are totally 10 examples of each subject. Try others for yourself.

Then, try the following:

```
load ORLfacedata;
X = data([1:10,31:40, 51:60],:);
Y = labels([1:10,31:40, 51:60]);
[Xtr, Xte, Ytr, Yte] = PartitionData(X, Y, 3);
figure(1); ShowFace(Xtr,3);
figure(2); ShowFace(Xte,7);
```

This has pulled out totally 30 examples for the subjects 1, 3 and 5 (10 for each subject). The function **`PartitionData`** divides these examples into a training set containing 3 examples for each subject and a testing set containing 7 examples for each subject. The function **`ShowFace`** is used to observe these examples.

Part 1: k-Nearest Neighbour

a. Build a k-NN classifier to recognise face images for subjects "1" and "30".

Perform the experiments following the instructions.	• Extract all the face images taken for the subjects "1" and "30". Get familiar with the provided file **`PartitionData.m`** for data partitioning. Prepare a dataset by randomly partitioning the extracted images into a training set containing 3 examples per subject and a testing set containing the remaining examples. Repeat this process and prepare 50 such datasets. • Get familiar with the provided file **`knearest.m`** that implements a k-NN classifier. Try the k-NN classifier on the 50 prepared datasets for varying values of k. Compute the training and testing accuracies. • Display the averaged training accuracy and the standard deviation as error bars using the command **`errorbar`** (i.e. k on the x-axis, and accuracy on the y-axis). Provide the error bar plot for the averaged testing accuracy and the standard deviations in a different graph. • Observe the classification results of the k-NN classifier on different datasets using the **`ShowResult.m`** function.
Answer the questions.	Q1. What is the training accuracy obtained when k=1? Explain it. Do testing and training accuracies differ and why? Q2. Do you get the same behaviour over different datasets? If not, why not? Q3. Is it a good idea to set your k value as an even number and why? Do the accuracies differ over different values of k? What happens when k gets bigger? Q4. Between subjects "1" and "30", which one is more difficult to classify? Explain it.

b. Build a k-NN classifier to recognise face images for all the 40 subjects.

Perform the experiments following the instructions.	• Prepare a dataset by randomly partitioning all the images into a training set containing 5 examples per subject and a testing set containing the remaining examples. Repeat this process and prepare 50 such datasets. • Pick one dataset. Implement the leave-one-out cross-validation to decide the value of the hyperparameter k for the k-NN classifier. Compute the testing accuracy for this selected k value. • Try the k-NN classifier with its k value selected as in the previous step on the 50 datasets. Compute the averaged testing accuracy and the standard deviation. • The simple accuracy measure we adopted does not tell us which subjects we commonly misclassify. This information obviously might be useful to know. Try to gather more information on your classification results, e.g., the averaged precision for each subject.
Answer the questions.	Q1. Is it a good idea to randomly select 200 training samples from the whole image set without specifying the number of selected samples for each subject, and why? Q2. Analyse the classification accuracies for individual subjects, and pick some subjects that is considered the most difficult to recognise by your k-NN classifier.

Part 2: Linear Least Squares

a. Build a linear binary classifier to classify face images for subjects "1" and "30".

Perform the experiments following the instructions.	• Implement a linear binary classifier based on the least squares approach using the normal equations. Try your classifier on the same 50 datasets that you prepared in Part1.a. • Generate a graph to display the testing accuracies. • Observe the classification results of your classifier on different datasets using the **`ShowResult.m`** function.
Answer the questions.	Q1. What training accuracies do you obtain with the linear classifier? Explain the reason. Q2. Between subjects "1" and "30", which one is more difficult for the linear classifier to recognise? Explain it.

b. Build a linear multi-class classifier to recognise face images for all the 40 subjects.

Perform the experiments following the instructions.	• Implement a linear multi-class classifier based on the least squares approach using the normal equations. Try your classifier on the same 50 datasets that you prepared in Part1.b. • Display and compare the testing accuracies of your k-NN classifier and your linear classifier in the same graph. • Examine the classification results of your linear classifier for each subject.
Answer the questions.	Q1. Compare your linear classifier with the k-NN classifier in terms of classification accuracy, time and space complexity. Q2. Analyse the classification results, and pick some subjects that is considered the most difficult to recognise by your linear classifier. Do k-NN and linear classifiers find the same subjects difficult?

Remember all graphs should have axis labels and a title. If you don't know what Matlab commands to use, try searching the internet of Matlab's help facility.

DELIVERABLES

By the deadline, you should submit two files, using the **`submit`** command as you did last year:

`ex2code.zip - all your ".m" files, zipped up`

`ex2report.pdf - your report, as detailed below`

REMEMBER to write your NAME and STUDENT ID on it!

The report should be **no longer than 2 pages** and converted to a PDF file (also print it to show your TA). You should give a full description of any graphs that you think represent your achievements.

There is no specific format – marks will be allocated roughly on the basis of:

- rigorous experimentation,
- knowledge displayed when talking to the TA,
- self-learning ability,
- how informative and well presented your graphs are,
- problem solving skill,
- language and ease of reading.

The lab is marked out of 15:

Part 1: k-Nearest Neighbour	**Total: 9 marks**
Part 1.a	**5 marks**
Part 1.b	**4 marks**
Part 2: Linear Least Squares	**Total: 6 marks**
Part 2.a	**3 marks**
Part 2.b	**3 marks**

Marking will be based on both the report and the code submitted. In other words, a new version of the report inconsistent with the submitted one is NOT accepted. The lab ex. will NOT be marked if only the code is submitted.