**Homework 2: Support vector machines for face recognition**

Overview:

Face recognition is implemented using linear and polynomial support vector machines (SVM) algorithms with one vs rest methodology. Design is validated via five-fold cross validation and average accuracy is reported for each implementation, linear and polynomial (degree = 2).

Dataset:

The Database of Faces is used to perform all of the training and testing in this implementation. There are ten different images of each of 40 distinct subjects. The size of each image is 92x112 pixels, with 256 grey levels per pixel. The images are organised in 40 directories (one for each subject), which have names of the form sX, where X indicates the subject number (between 1 and 40). In each of these directories, there are ten different images of that subject, which have names of the form Y.pgm, where Y is the image number for that subject (between 1 and 10). This database is made available by AT&T Laboratories Cambridge and can be downloaded here: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

Prerequisites:

* python 2.7
* cv2
* numpy
* cvxopt
* math
* random
* itertools

Running:

python svm\_main.py

Function blocks:

* main()
  + Reads in images
  + Breaks images into 5 image groups for five-fold cross validation
  + Iterates through unique combinations of 5 image groups and calls linear and polynomial svm() to train on and test on these unique combinations
  + Reports average accuracies for linear and polynomial implementations
* svm(num\_of\_people, img\_training, classes\_training, img\_testing, classes\_testing, polynomial\_flag = False, polynomial\_degree = 2.0)
  + Calls training() and testing() function blocks
  + Returns accuracy
* training(img\_training, classes\_training, num\_of\_people, polynomial\_flag = False, polynomial\_degree = 2.0)
  + Calculates a model (w and b) for every class using quadratic programming solver
  + Returns a list of w’s and b’s each corresponding to one of the classes (40 people)
* testing(img\_testing, classes\_testing, num\_of\_people, w, b)
  + Tests each test images against every class (40 people) and reports on total of correctly and incorrectly classified images
  + Returns accuracy

Results:

Average linear accuracy:

Average quadratic (degree = 2) accuracy:

Group combinations:

All linear accuracies (1 per group combination):

All quadratic accuracies (1 per group combination):