CS 5661: Topics in Data Science
Homework3, Due Date: Fri, Apr 20
Instructor: Dr. Mohammad Pourhomayoun

Face Recognition Using SVM:

Write and submit your python codes in "Jupyter Notebook" to perform the following tasks. Make sure to provide proper descriptions as MarkDown for each section of your code.

a- Download the dataset "Face" from CSNS. Check out the dataset. Open some of the jpg images. This is the Oivetti database of face images from AT&T research lab. It includes 400 faces (64x64 pixels) from 40 people (10 images per person).
You have to also download the csv file that includes the labels of the images (the label is person's ID). The goal is to build a **Face Recognition** algorithm to recognize each person using PCA dimensionality reduction and a non-linear SVM.
you can use:
mpimg.imread(file_name) to load an image, and
plt.imshow(image_name, cmap=plt.cm.gray) to show an image (This is a little different from what we had in HW2!). Add **%matplotlib inline** at top of your code to make sure that the images will be shown inside the Jupyter explorer page.

b- Build the feature matrix and label vector: Each image is considered as a data sample with pixels as features. Thus, to build the feature table you have to convert each 64x64 image into a row of the feature matrix with 4096 columns.

c- Normalize each column of your feature matrix (**This is required!**).

d- Use sklearn functions to split the **Normalized** dataset into testing and training sets with the following parameters: **test_size=0.25, random_state=5**.

e- The dimensionality of the data samples is 4096. Use PCA to reduce the dimensionality from 4096 to 50 (i.e. only 50 principal components!). You should "**fit**" your PCA on your training set only, and then use this fitted model to "**transform**" both training and testing sets (When you finish this step, the number of columns in your testing and training sets should be 50). We will cover the details of PCA in next session of class. For now, you can use this format:

```
from sklearn.decomposition import PCA
k = 50 # (k is the number of components (new features) after dimensionality reduction)
my_pca = PCA(n_components = k)
# X_Train is feature matrix of training set before dimensionality reduction,
# X_Train_New is feature matrix of training set after dimensionality reduction:
X_Train_new = my_pca. fit_transform(X_Train)
X_Test_new = my_pca. transform(X_Test)
```

f- Design and Train a non-linear SVM classifier to recognize the face based on the training dataset that you built in part (d). Use **SVC(C=1, kernel='rbf', gamma=0.0005, random_state=1)**. Then, Test your SVM on testing set (from part(d)), and calculate and report the accuracy. Also, calculate and report the Confusion Matrix.

g- Now, use **GridSearchCV** to find the best value for parameter **C** in your SVM. Search in this list: [0.1, 1, 10, 100, 1e3, 5e3, 1e4, 5e4, 1e5]. Remember that we want to use cross-validation method (GridSearchCV) to find the best C. Thus, you can again merge X_train_new and X_test_new (after dimensionality reduction), and also merge y_train and y_test, and then use GridSearchCV with 10-fold cross validation to find C.