**PATTERN RECOGNITION**

*Spring 2018*

Face Recognition

*Presented by:*

* *Seif ElDin Samer 3165*
* *Mohamed Samy Aly 3311*
* *Nada Hazem Fahmy 3331*

In the 1st part of the assignment, we read the images from the folders and then we split the data into 2 sets; 1 for training and the other one for testing.We generated the label vector as well

for x in range(1,41,1):

number = str(x)

filenames = glob.glob('E:\\8th term\\Pattern Recognition\\att\_faces\\orl\_faces\\s'+ number +'/\*.pgm')

filenames.sort()

img = [Image.open(fn).convert('L') for fn in filenames]

images = np.asarray([np.array(im).flatten() for im in img])

#label= np.full((10, 1), x)

#person= np.append(images, label, axis =1)

if (x==1):

data = np.array(images)

train\_label =np.full((5, 1), x)

else:

data = np.append(data,images,axis = 0)

train\_label= np.append(train\_label , np.full((5, 1), x) , axis=0)

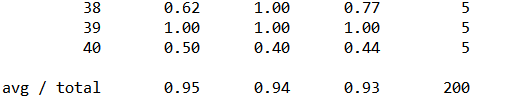
#spliting data into testing and training

#train = data[0:][::2] #even

#test = data[1:][::2] # odd

Then we followed the pseudocode in the implementation of the PCA, we calculated the eigen vectors and values only once and we saved them to a file due to the long computing time it takes the processor to compute them.

Then, we used the built in functions of the KNN in the sklearn library to calculate the accuracy and precision of our algorithm, varying each time the value of alpha. The output of the KNN function was the tabular comparison between the accuracy, precision, number of occurrences and the weighted average between the accuracy and precision.

Below the table of alpha= 0.8 and k=1 with accuracy= 0.94

Below the table of alpha= 0.85 and k=1 with accuracy= 0.94

Below the table of alpha= 0.9 and k=1 with accuracy= 0.94



Below the table of alpha= 0.95 and k=1 with accuracy= 0.94



The results are very weird but it was found that the accuracy varies when the number of neighbours in the KNN varies, but theoretically the variation of alpha varies the accuracy of the results.

For the LDA part, we followed the pseudocode and then we followed the same steps as in the previous part, we found for k=1 the accuracy= 1!

There is a huge difference between the PCA and LDA in the accuracy and it seems like the LDA is much more accurate than the PCA for the given dataset.

For the last part, the built in function of KNN tie breaks on the element with the higher index.

Bellow is the table of accuracy against different K values for PCA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K | 1 | 3 | 5 | 7 |
| Accuracy | 0.93 | 0.85 | 0.78 | 0.7 |

Bellow is the table of accuracy against different K values for LDA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K | 1 | 3 | 5 | 7 |
| Accuracy | 0.93 | 0.82 | 0.76 | 0.7 |