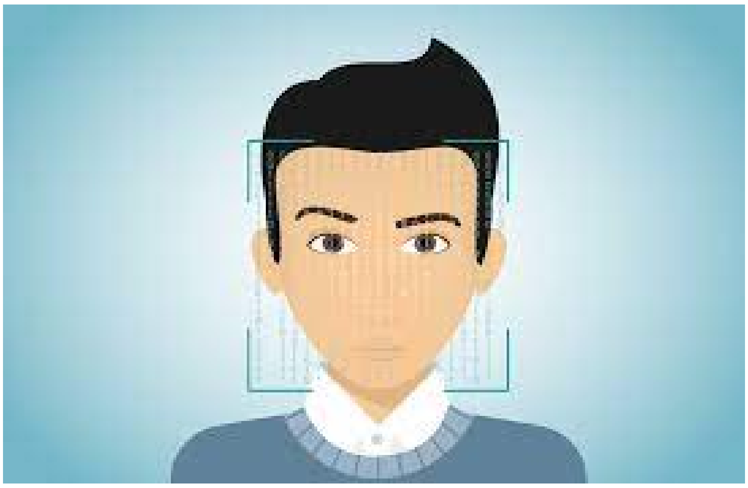


# Lab 1 Pattern Recognition

| Names | IDs |
|---|---|
| Katrin Magdy Girgis | 18011250 |
| Eman Mohamed Abdo | 18010431 |
| Youssef Hussien | 18012118 |

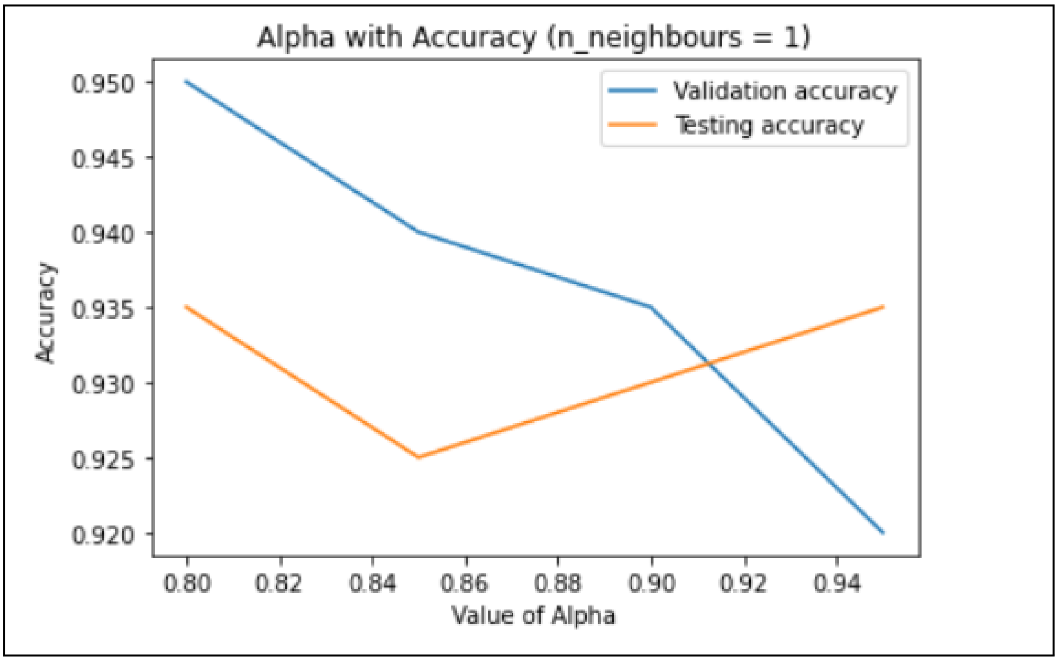
# Steps we did in this Lab :

- We mounted a google drive to the notebook.
- Then we uploaded the kaggle datasets - att-database-of-faces and natural images - to the drive.
- We wrote the important imports needed in this assignment.
- Then we generated the data matrix and Label vector from the faces data set by reading each image from the folder in the drive and converting it to a 1d array and reading the label from the folder name containing the image.
- Next we split the data matrix and the label vector into odd rows for training data and even rows for testing data
- We made a **PCA** method which decreases the dimensions of the data matrix to k dimensions. We did that by following the pseudo code of the pca algorithm. Then with the reduced data we got , this method prints the validation accuracy (we got after cross validation) and testing accuracy of the Knn model.

  This method returns the validation and testing accuracy for every alpha where alpha =[0.8,0.85,0.9,0.95] where n-neighbors in Knn = 1.

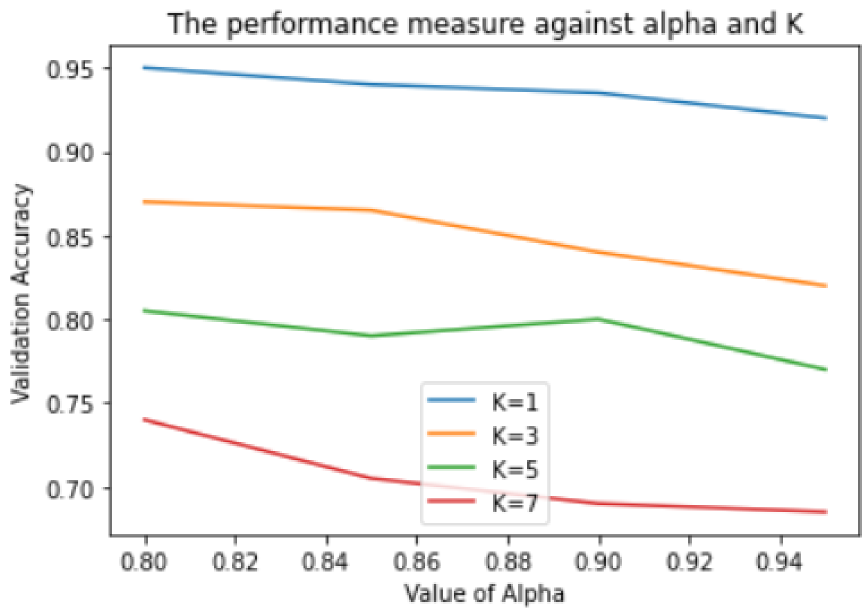
Then we plot the returned validation and testing accuracy with alpha.

```
Alpha: 0.8
No.of components37
shape of Projection matrix(10304, 37)
Validation Accuracy: 0.9499999999999998
Testing Accuracy: 0.935
Alpha: 0.85
No.of components53
shape of Projection matrix(10304, 53)
Validation Accuracy: 0.9399999999999998
Testing Accuracy: 0.925
Alpha: 0.9
No.of components77
shape of Projection matrix(10304, 77)
Validation Accuracy: 0.9349999999999999
Testing Accuracy: 0.93
Alpha: 0.95
No.of components116
shape of Projection matrix(10304, 116)
Validation Accuracy: 0.9200000000000002
Testing Accuracy: 0.935
```




**We observed from the above plot that as the alpha increases the classification accuracy decreases and that is by : increasing alpha the model becomes overfitted so the classification accuracy decreases.**

- We made a **pca_tunning method** , this was made mainly for tunning (to choose best number of neighbors and best alpha which gives the best validation accuracy) .We plot alpha , k with validation accuracy .Finally, we print best alpha, best n neighbors ,best validation accuracy  and testing accuracy after training model with the best parameters .

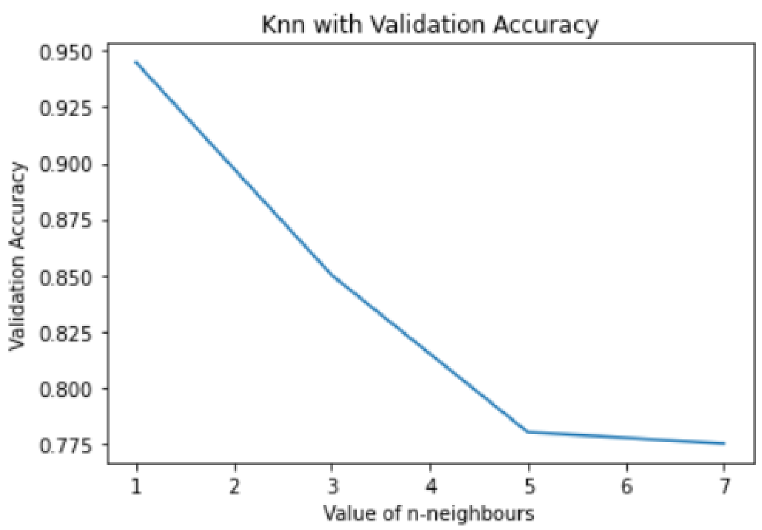



```
The Best Validation Accuracy: 0.9499999999999998
The Best Alpha: 0.8
The Best n-neighbours: 1
Test accuracy with best alpha and n-neighbours : 0.935
```

- We made an **LDA** method that also reduces dimensions but with an LDA algorithm . it aims to find the separator which decreases the variance within class and separate means of classes from each other. Here we print validation accuracy and testing accuracy of the knn model (with n-neighbors = 1) after reducing dimensions using the LDA algorithm.

```
Validation Accuracy in LDA: 0.945
Testing Accuracy in LDA: 0.925
```

- We made the **LDA_tunning method** which aims to get the best n neighbors in the knn model after reducing dimensions using the LDA algorithm . Here we plot validation accuracy with the number of neighbors , print best validation accuracy and testing accuracy after training the knn model with best k and reduced data.




```
The best Validation accuracy = 0.945
The best n-neighbours = 1
Test accuracy with the best n-neighbours : 0.925
```

**Compare LDA and PCA**

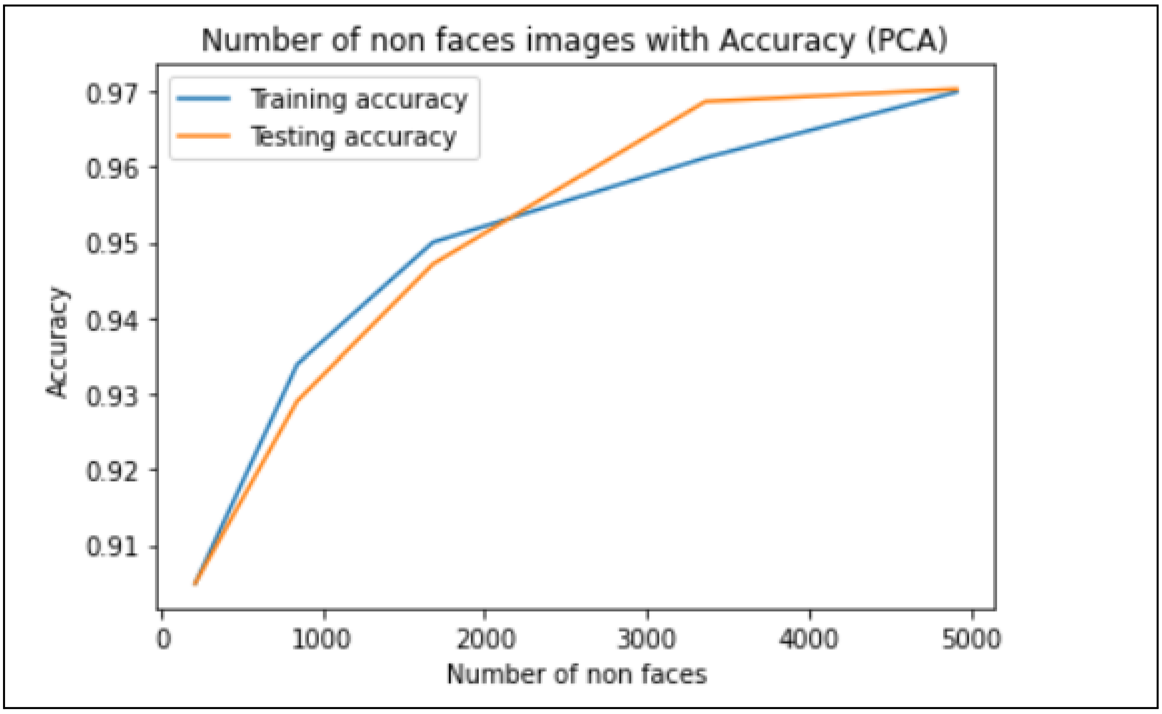
From the plots and the data above we observed that the pca algorithm gives better accuracy than lda.as pca performs better when the number of samples in each class is less.

- After that we wrote the method **read_non_faces_from_folder** and **read_non_faces** to read non faces images from a folder in google drive that we uploaded the natural images dataset in , we generated the non-faces data matrix by reading every non face image and converting it  to 1D array.
- We generated a new label vector where the label of the faces images = 1 and the label of the non faces images =0 .
- We generated the new data matrix by splitting the non faces data matrix into training and testing,then we combined the non faces train with faces train to get the total train data matrix and the non faces test with the faces test to get the total test data matrix then we shuffled them.
- We made **pca_non_faces** method on the data matrix (which contains faces and non faces) , Here it is the same as **pca_tunning** method we reduce dimensions using PCA algorithm then get best alpha and best number of neighbors for knn model which gives best validation accuracy . After we got the best model we train it and print best validation

accuracy , best alpha , best number of neighbors and finally the testing accuracy .

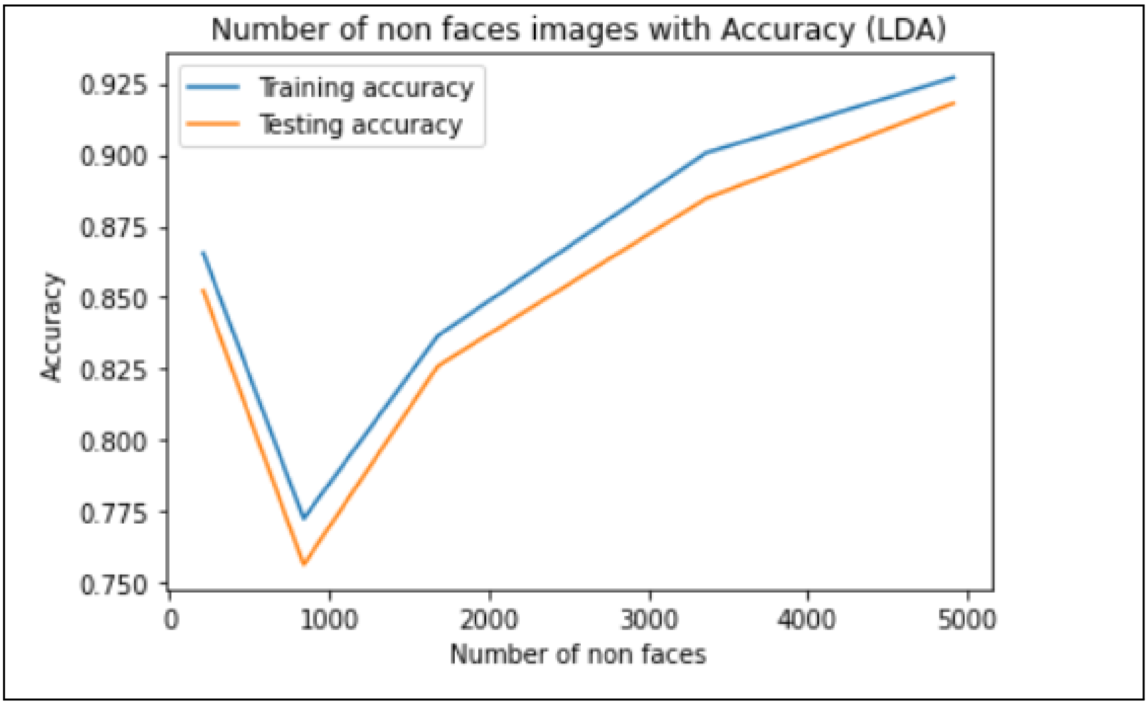
- Then for different number of non faces we ran the PCA algorithm for the data matrix which contains faces , non faces . We compared accuracies We got with the number of non faces and plotted the results (validation accuracy with number of non faces , testing accuracy with number of non faces) .

```
The number of non face images : 210
The Best Trainning Accuracy: 0.9049180327868853
The Best Alpha: 0.8
The Best n-neighbours: 1
Test accuracy with best alpha and n-neighbours : 0.9049180327868852
The number of non face images : 840
The Best Trainning Accuracy: 0.9338709677419355
The Best Alpha: 0.8
The Best n-neighbours: 1
Test accuracy with best alpha and n-neighbours : 0.9290322580645162
The number of non face images : 1680
The Best Trainning Accuracy: 0.95
The Best Alpha: 0.8
The Best n-neighbours: 1
Test accuracy with best alpha and n-neighbours : 0.9471153846153846
The number of non face images : 3360
The Best Trainning Accuracy: 0.9611702127659575
The Best Alpha: 0.8
The Best n-neighbours: 1
Test accuracy with best alpha and n-neighbours : 0.9686170212765958
The number of non face images : 4914
The Best Trainning Accuracy: 0.9698851648896252
The Best Alpha: 0.8
The Best n-neighbours: 1
Test accuracy with best alpha and n-neighbours : 0.9702672186676703
```




- We made the **LDA_non_faces** method , it is the same as lda_tunning except that the projection matrix contains only 1 eigen vector (the dominant eigen vector ) instead of 39 because in the new data matrix (faces and non faces) we only have two classes so we only need one linear discriminant (separator) to separate the two classes.
- Then for different number of non faces we ran the LDA algorithm for the data matrix which contains faces , non faces . We compared accuracies We got with the number of non faces and plotted the results (validation accuracy with number of non faces , testing accuracy with number of non faces) .

```
The number of non face images : 210
The best Validation accuracy = 0.8655737704918032
The best n-neighbours = 3
Test accuracy with the best n-neighbours : 0.8524590163934426
The number of non face images : 840
The best Validation accuracy = 0.7725806451612904
The best n-neighbours = 3
Test accuracy with the best n-neighbours : 0.7564516129032258
The number of non face images : 1680
The best Validation accuracy = 0.8365384615384615
The best n-neighbours = 7
Test accuracy with the best n-neighbours : 0.8259615384615384
The number of non face images : 3360
The best Validation accuracy = 0.9005319148936171
The best n-neighbours = 7
Test accuracy with the best n-neighbours : 0.8845744680851064
The number of non face images : 4914
The best Validation accuracy = 0.9269848349687779
The best n-neighbours = 7
Test accuracy with the best n-neighbours : 0.9179525780955965
```



**Observation:**

As the number of non face images increases , the accuracy increases in both LDA and PCA. That is because as more data is added the broader the purview to the problem creating increased accuracy. This will help us to know the trend of points and get a better curve which fits the points so when having new point we want to classify -> error in classification decreases .

- Finally , We showed Failure and success cases at number of non faces : 420 one time with PCA algorithm and another time with LDA algorithm .

**PCA**

```
The Best Trainning Accuracy: 0.9146341463414634
The Best Alpha: 0.8
The Best n-neighbours: 1
Test accuracy with best alpha and n-neighbours : 0.9048780487804878
Validation accuracy : 0.9146341463414634
Testing accuracy : 0.9048780487804878
```

Success

Success

Failure

Success

Success

Success

## LDA

```
The best Validation accuracy = 0.7268292682926832
The best n-neighbours = 7
Test accuracy with the best n-neighbours : 0.7292682926829268
Validation accuracy : 0.7268292682926832
Testing accuracy : 0.7292682926829268
```

Success

Success

Success

Failure

Success

Failure

# Colab Link :

# [Colab link]