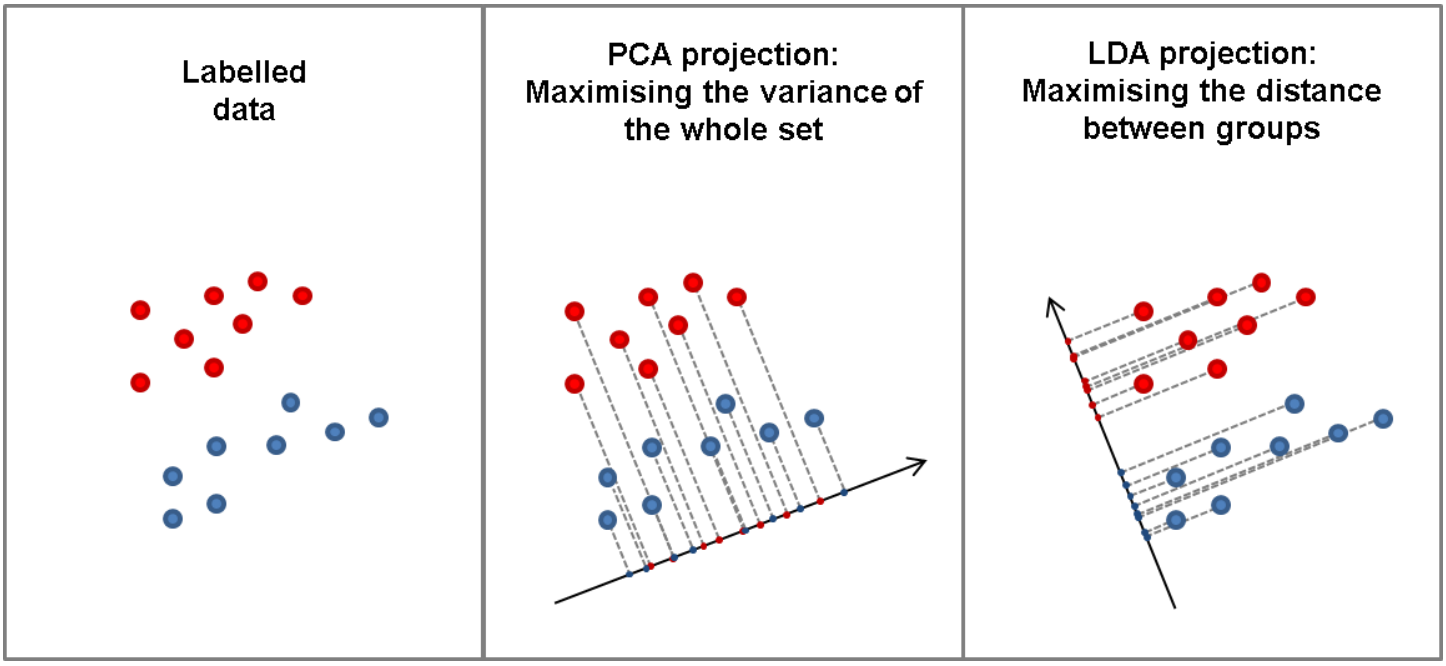
Face Recognition

Lina Hazem 5613 / Mariam Fekry 5614 / Mariam Matar 5653

Table of Contents

INTRODUCTION

Face recognition has been one of the most interesting and important research fields in the past two decades. The reasons come from the need of automatic recognitions and surveillance systems, the interest in human visual system on face recognition, and the design of human-computer interface, etc. These researches involve knowledge and researchers from disciplines such as neuroscience, psychology, computer vision, pattern recognition, image processing, and machine learning, etc. A bunch of papers have been published to overcome difference factors (such as illumination, expression, scale, pose) and achieve better recognition rate, while there is still no robust technique against uncontrolled practical cases which may involve kinds of factors simultaneously. In this report, we'll go through two approaches for face recognition LDA (Linear Discriminant Analysis) and PCA (Principle Component Analysis), using simple classifier KNN.



ALGORITHM OVERVIEW

Approach 1: Classification Using PCA

Principal components analysis (PCA) is a distance-based ordination technique used primarily to display patterns in multivariate data. It aims to display the relative positions of data points in fewer dimensions while retaining as much information as possible and explore relationships between dependent variables.

- **Algorithm**

ALGORITHM 7.1. Principal Component Analysis

PCA $(\mathbf{D}, \alpha)$:

1 $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ // compute mean
2 $\mathbf{Z} = \mathbf{D} - \mathbf{1}\cdot\boldsymbol{\mu}^T$ // center the data
3 $\boldsymbol{\Sigma} = \frac{1}{n}(\mathbf{Z}^T\mathbf{Z})$ // compute covariance matrix
4 $(\lambda_1, \lambda_2, \ldots, \lambda_d) = \text{eigenvalues}(\boldsymbol{\Sigma})$ // compute eigenvalues
5 $\mathbf{U} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_d) = \text{eigenvectors}(\boldsymbol{\Sigma})$ // compute eigenvectors
6 $f(r) = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$, for all $r = 1, 2, \ldots, d$ // fraction of total variance
7 Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality
8 $\mathbf{U}_r = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_r)$ // reduced basis
9 $\mathbf{A} = \{\mathbf{a}_i \mid \mathbf{a}_i = \mathbf{U}_r^T\mathbf{x}_i, \text{for } i = 1, \ldots, n\}$ // reduced dimensionality data

Fig(1)

Approach 2: Classification Using LDA

LDA is a dimensionality reduction algorithm, similar to PCA. However, while PCA is an unsupervised algorithm that focusses on maximizing variance in a dataset, LDA is a supervised algorithm that maximizes separability between classes.

- **Algorithm**

ALGORITHM 20.1. Linear Discriminant Analysis

LINEARDISCRIMINANT $(\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n})$:

1 $\mathbf{D}_i \leftarrow \{\mathbf{x}_j \mid y_j = c_i, j = 1, \ldots, n\}, i = 1, 2$ // class-specific subsets
2 $\boldsymbol{\mu}_i \leftarrow \text{mean}(\mathbf{D}_i), i = 1, 2$ // class means
3 $\mathbf{B} \leftarrow (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$ // between-class scatter matrix
4 $\mathbf{Z}_i \leftarrow \mathbf{D}_i - \mathbf{1}_{n_i}\boldsymbol{\mu}_i^T, i = 1, 2$ // center class matrices
5 $\mathbf{S}_i \leftarrow \mathbf{Z}_i^T\mathbf{Z}_i, i = 1, 2$ // class scatter matrices
6 $\mathbf{S} \leftarrow \mathbf{S}_1 + \mathbf{S}_2$ // within-class scatter matrix
7 $\lambda_1, \mathbf{w} \leftarrow \text{eigen}(\mathbf{S}^{-1}\mathbf{B})$ // compute dominant eigenvector

Fig(2)

DISCUSSION

• Exploring the ORL Dataset and generating the Data Matrix

The **ORL** Database of Faces contains 400 images from 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). The size of each image is (92x112) pixels, with 256 grey levels per pixel.

After downloading each image is transformed into a vector of length (10304) and stacked into the data matrix of size(400x10304) and a label vector was generated where the label vector consists of integers (1:40) corresponding to the subject id.

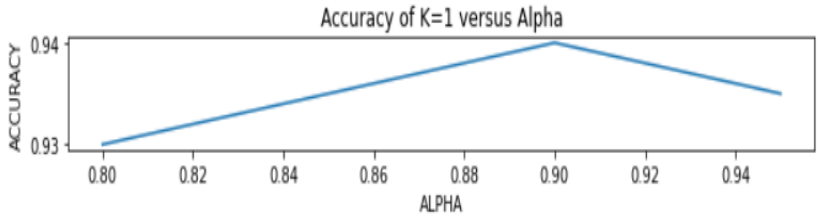
• Splitting the Data Matrix into training and testing sets

The data matrix was split with 50% ratio, where for each person there is 5 instances in the training and testing sets respectively, the splitting was made using the mod operator (%2) to split in odd-even manner where odd rows are for training set and even rows are for testing set and their corresponding labels.

• Classification using PCA

The above algorithm in fig(1) was implemented using the three main functions: Reduced_Dimension(), PCA(), and projection( ). First PCA( ) is called with parameters training and testing sets and it computed the mean of each set, centering data, and getting covariance matrix for the training set to obtain the eigen values and eigen vectors. Then, for each value of Alpha the reduced dimensions matrix is computed from the projection( ) function where it internally calls Reduced_Dimension( ) to get r (number of dominant eigen vector) corresponding to the current Alpha value, after getting the new reduced matrix obtained from calling projection( ) function, finally the simple classifier KNN where k=1 is called.

The following figure shows the relation between the different values of Alpha and the accuracy

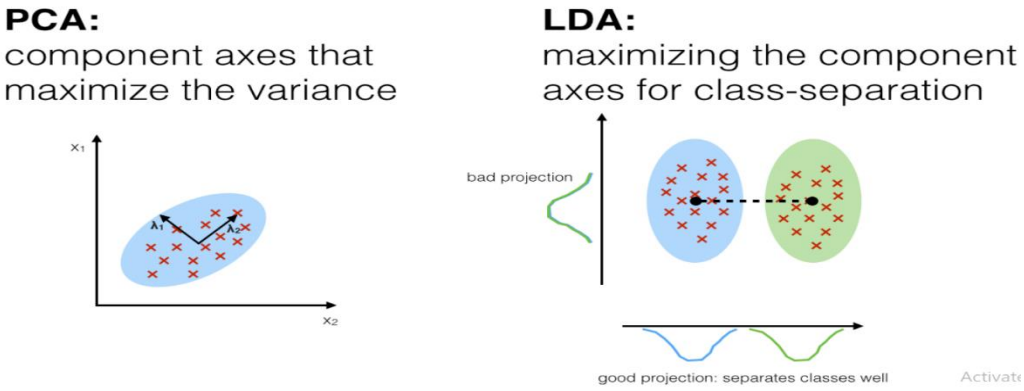


Accuracy Vs Alpha

It is then obvious from the shown figure, that after a certain alpha value there is no clear or big change in accuracy and it actually decreases when alpha is set to 0.95 as it is over-fits. So, we can deduce the best alpha value here is 0.9, which achieves very good accuracy with no or little redundant dimensions in the subspace.

- ## Classification using LDA

The above algorithm in fig(2) was implemented as follows; First start by dividing the training set into 40 equal classes using the mod operator (%5) and the array slicing technique where every class consists of 5 instances then the mean for every class, and the overall mean are computed. Using these outputs the between class scatter and the within class scatter matrices are computed. The within class scatter matrix was used to get its inverse and multiplied by the between class scatter matrix to get the eigen values and eigen vectors. The first 39 dominant eigen vectors were taken and multiplied by the centered data to get the reduced dimension array. Using a simple classifier the accuracy was calculated and found to be (0.95). The result of PCA was found to be (0.93) for k = 1 and Alpha = 0.8 while the result for LDA was (0.95) for k = 1, by comparing these two results it was deduced that PCA performs better in case where number of samples per class is less. Whereas LDA works better with large dataset having multiple classes; class separability is an important factor while reducing dimensionality.



PCA Vs LDA

- ## K Nearest Neighbor and Classifier Tunning

Using KNN class in sklearn the model is trained with several k values (1,3,5,7), model.fit ( ) was given the training set and its labels then the predict( ) function is called with parameter testing set and testing labels and the predictions are compared to calculate the accuracy. This function is called in both algorithms with different values of k in PCA and LDA.

```
For Alpha =  0.8        With  1  Neighbours     The Accuracy Will be =  0.93
For Alpha =  0.8        With  3  Neighbours     The Accuracy Will be =  0.85
For Alpha =  0.8        With  5  Neighbours     The Accuracy Will be =  0.82
For Alpha =  0.8        With  7  Neighbours     The Accuracy Will be =  0.78
For Alpha =  0.85       With  1  Neighbours     The Accuracy Will be =  0.935
For Alpha =  0.85       With  3  Neighbours     The Accuracy Will be =  0.855
For Alpha =  0.85       With  5  Neighbours     The Accuracy Will be =  0.835
For Alpha =  0.85       With  7  Neighbours     The Accuracy Will be =  0.77
For Alpha =  0.9        With  1  Neighbours     The Accuracy Will be =  0.94
For Alpha =  0.9        With  3  Neighbours     The Accuracy Will be =  0.845
For Alpha =  0.9        With  5  Neighbours     The Accuracy Will be =  0.815
For Alpha =  0.9        With  7  Neighbours     The Accuracy Will be =  0.75
For Alpha =  0.95       With  1  Neighbours     The Accuracy Will be =  0.935
For Alpha =  0.95       With  3  Neighbours     The Accuracy Will be =  0.85
For Alpha =  0.95       With  5  Neighbours     The Accuracy Will be =  0.81
For Alpha =  0.95       With  7  Neighbours     The Accuracy Will be =  0.73
```

Classifier Tunning for PCA

```
For  1 Neighbours       The Accuracy Will be =  0.95
For  3 Neighbours       The Accuracy Will be =  0.88
For  5 Neighbours       The Accuracy Will be =  0.865
For  7 Neighbours       The Accuracy Will be =  0.85
```

Classifier Tunning for LDA

- ## Faces Vs Non-Faces

  - A new dataset was used for the non-faces part (one type of animals) with a size of (4000), each photo is turned into a gray scale and is resized to (92x112) then flattened into a vector of size (10304), a new data matrix of dimensions (4200x10304) was generated that contains (200) rows of faces and (4000) non-faces.

  - A new function for training and testing sets splitting was developed where it takes a variable number of non-faces as a parameter and then it splits the data matrix taking a fixed number of faces from the first (200) rows ex: (100) faces per set and a variable number of non-faces (100,400,1000,2000,4000) where it divides the number of non-faces equally between the two sets.

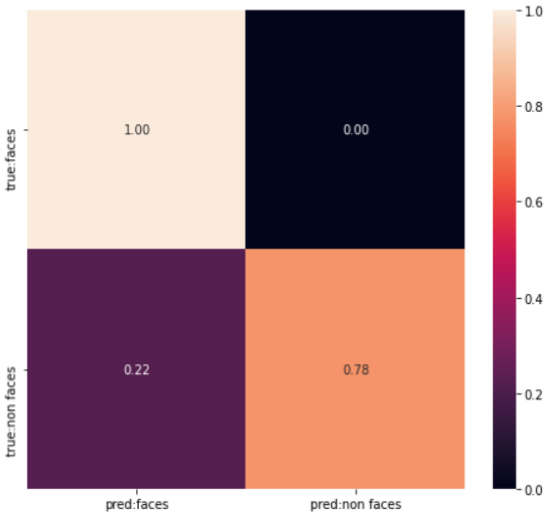
  - Unlike previously now it's a two class classification problem between Faces and Non-Faces, this change will not affect the algorithm for PCA but in case of LDA its no longer a multiclass LDA so one eigenvector is needed where the dominant eigenvector can be calculated using the following formula:

$$w = S^{-1}(\mu_1 - \mu_2)$$

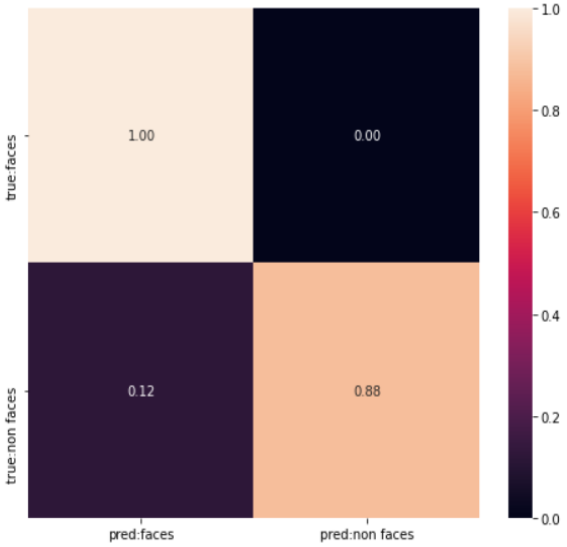
  - PCA and LDA were applied on the new training and testing sets for different distributions (slight imbalance and severe imbalance), the following results in both cases were obtained for the KNN classifier where k = 1 and Alpha = 0.9 (in case of PCA) as shown below we can see that due to the imbalance the model is not behaving as expected as the minority class is harder to predict because there are few examples of this class, by definition.

PCA Results:

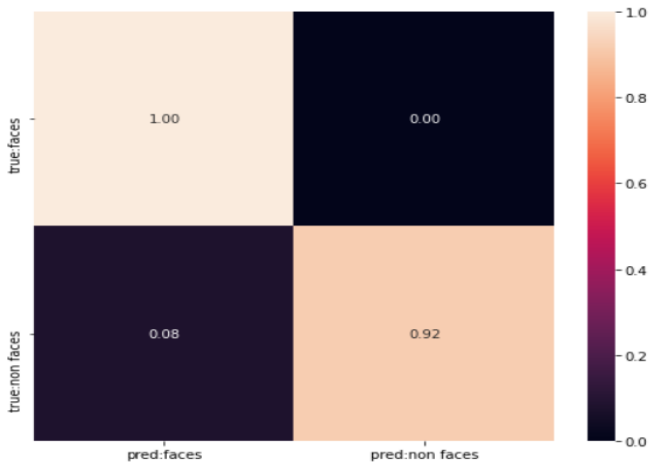
```
For the following range of non-faces in the training set  50 :
For Alpha =  0.8         With  1  Neighbours     The Accuracy Will be =  0.9266666666666666
 NORMALIZED CONFUSION MATRIX
```



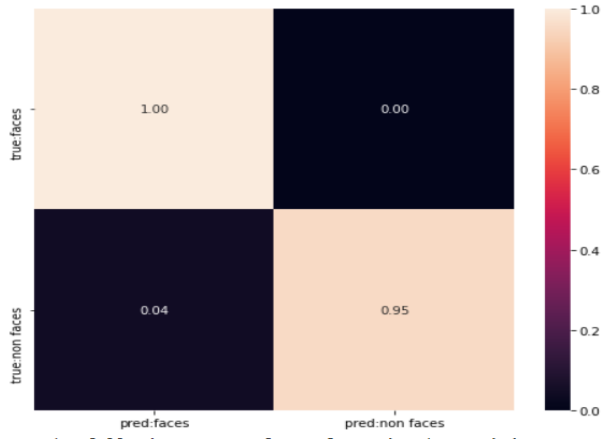

```
For the following range of non-faces in the training set  200 :
For Alpha =  0.8         With  1  Neighbours     The Accuracy Will be =  0.92
 NORMALIZED CONFUSION MATRIX
```



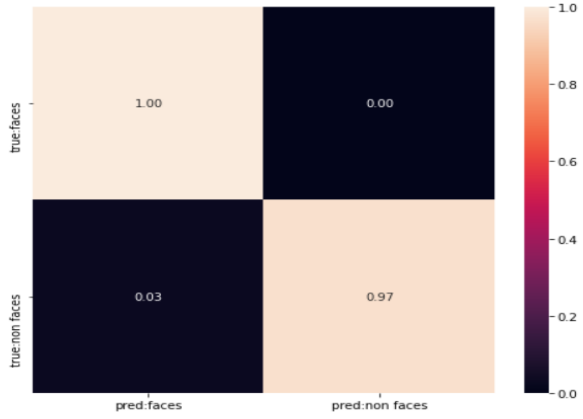
```
For the following range of non-faces in the training set  500 :
For Alpha =  0.8         With  1  Neighbours     The Accuracy Will be =  0.9316666666666666
 NORMALIZED CONFUSION MATRIX
```

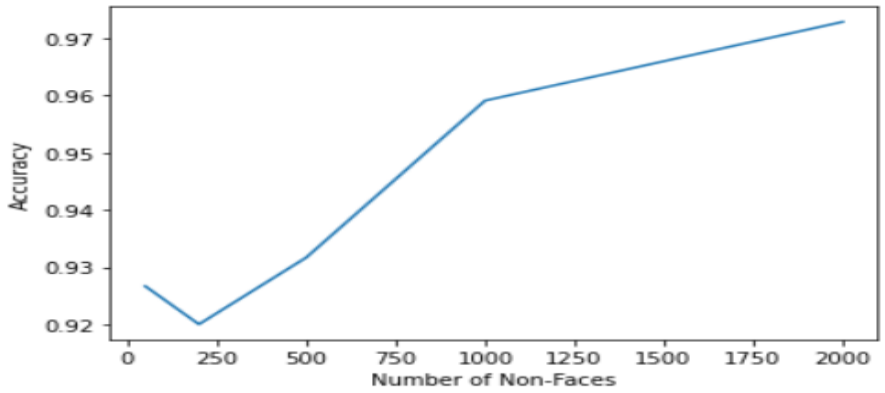



```
For the following range of non-faces in the training set  1000 :
For Alpha =  0.8         With  1  Neighbours     The Accuracy Will be =  0.9590909090909091
 NORMALIZED CONFUSION MATRIX
```



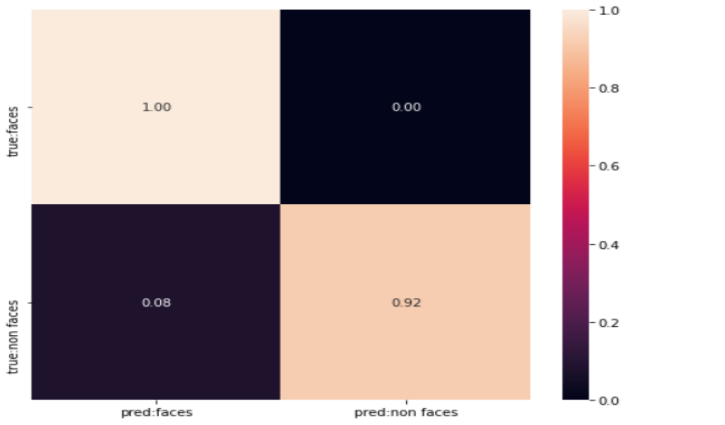

```
For the following range of non-faces in the training set  2000 :
For Alpha =  0.8         With  1  Neighbours     The Accuracy Will be =  0.9728571428571429
 NORMALIZED CONFUSION MATRIX
```



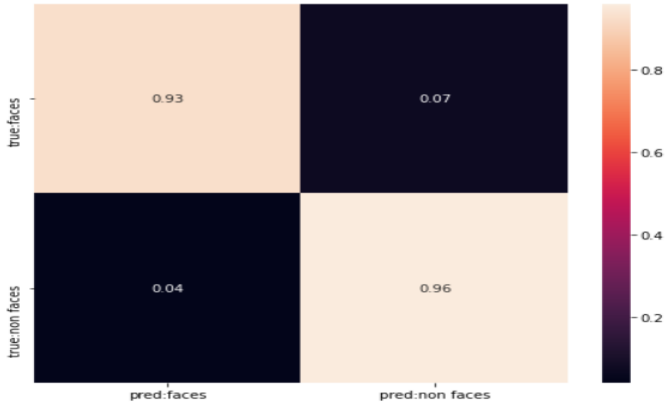


LDA Results:

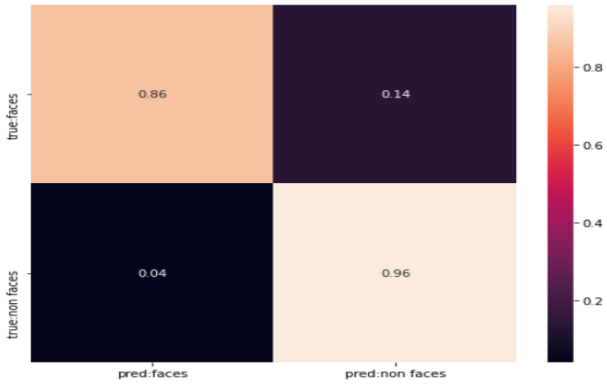
```
For the following range of non-faces in the training set  50 :
For  1 Neighbours        The Accuracy Will be =  0.9733333333333334
 NORMALIZED CONFUSION MATRIX
```



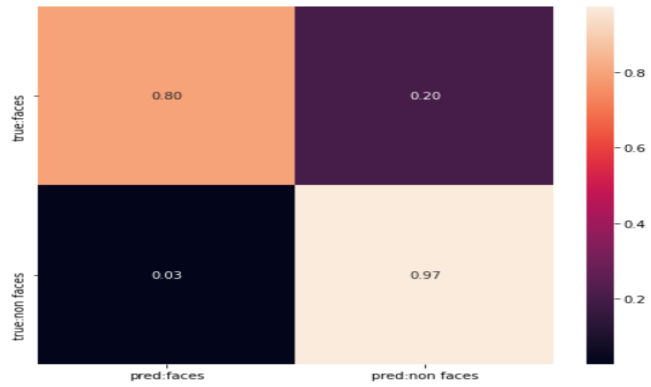
```
For the following range of non-faces in the training set  200 :
For  1 Neighbours         The Accuracy Will be =  0.95
 NORMALIZED CONFUSION MATRIX
```



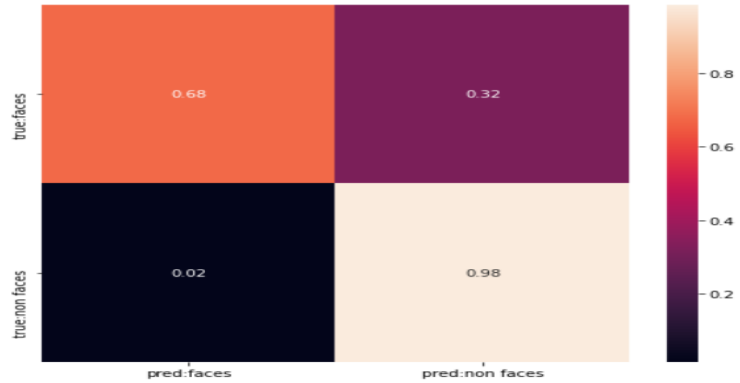

```
For the following range of non-faces in the training set  500 :
For  1 Neighbours         The Accuracy Will be =  0.9433333333333334
 NORMALIZED CONFUSION MATRIX
```

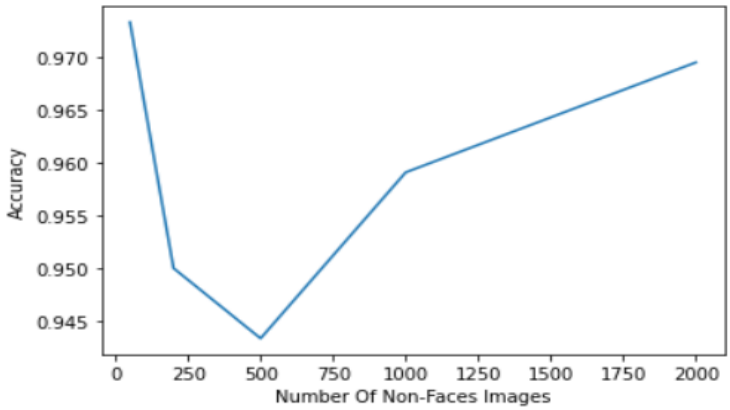



```
For the following range of non-faces in the training set  1000 :
For  1 Neighbours         The Accuracy Will be =  0.9590909090909091
 NORMALIZED CONFUSION MATRIX
```



```
For the following range of non-faces in the training set  2000 :
For  1 Neighbours          The Accuracy Will be =  0.9695238095238096
 NORMALIZED CONFUSION MATRIX
```





- ## Splitting training and testing sets 70:30 (Bonus)

  Using sklearn the above steps are repeated but with new training and testing sets that are split with ratios (70:30) where training set has 280 samples and the testing set has 120 samples, then PCA and LDA are reapplied to the new sets and KNN classifier is used with K = 1 and Alpha = 0.9 getting accuracies 0.9333 and 0.8833 respectively.

CONCLUSION

- Dimensionality reduction method was implemented using PCA and LDA to do image reconstruction and face recognition using ORL dataset that has different properties also it was applied on non-face dataset that consists of one type of animals.

- Dimensionality reduction makes our machine learning model learn faster and more efficient in terms of computational cost. Moreover, this method is to prevent our model suffers from overfitting problem due to its feature selection properties.

- Although one might think that LDA should always outperform PCA (since it deals directly with class discrimination), empirical evidence suggests otherwise. PCA might outperform LDA when the number of samples per class is small or when the training data non-uniformly sample (skewed data) the underlying distribution. In many practical domains, and especially in the domain of face recognition, one never knows in advance the underlying distributions for the different classes. So, one could argue that in practice it would be dicult to ascertain whether or not the available training data is adequate for the job.

REFERENCES

- https://cse.buffalo.edu/~jcorso/t/555pdf/pca-versus-lda.pdf
- https://machinelearningmastery.com/what-is-imbalanced-classification/?fbclid=IwAR1zLsJi2FyCF0JZPeBRnWdRRnFf5Mnobod0xgXR27ZsgD0PdvgMrk8HxqI
- https://en.wikipedia.org/wiki/Facial_recognition_system
- https://www.linkedin.com/pulse/dimensionality-reduction-pca-vs-lda-face-recognition-m-farhan-tandia/?fbclid=IwAR2vnkqF7cLbcwC1q6wj-fnOc_qCcDGuf5Efd_K2aQtvT_eEsO6aPw6IcDw