SMAI Assignment 2 Report | Date: *03.11.2019*

Zubair Abid: *20171076* . .

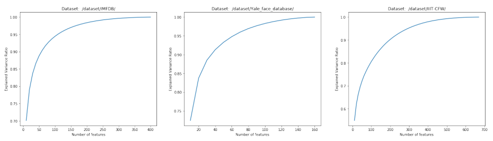
# 1 Questions

## 1.1 Basic

1. **What are eigen faces?**
   Eigenfaces is the name given to a set of eigenvectors when they are used in the computer vision problem of human face recognition. It is essentially an eigendecomposition of faces to lower dimensions, using multiple methods such as basic PCA, LDA, Fisherface, etc.

2. **How many eigenvectors/faces are required to "satisfactorily" reconstruct a person in these three datasets?**
   We do not need too many features to reconstruct a person, per se. However, to be absolutely certain, it would be correct to claim that a representation that has about 95% of the features of the original image should be satisfactory to reconstruct a person in this three databases.

   Going with this logic, we attempt to calculate the number of features needed by a basic PCA eigenface representation to get this 95%. IMFDB: 110 features; Yale face database: 70 features; IIIT CFW: 300 features

3. **Which person/identity is difficult to represent com-pactly with fewer eigen vectors? Why is that? Explain with your empirical observations and intuitive answers**
   Within each class, the hardest to represent compactly (by small margins) have been:

   (a) IMDB: 2

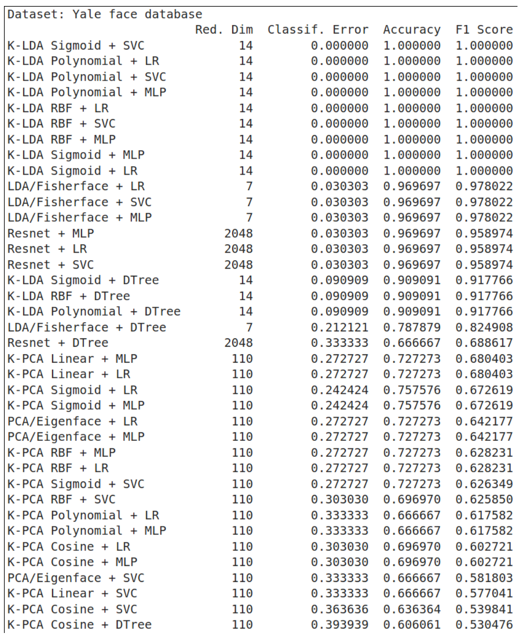
   (b) Yale: 0

   (c) IIIT-CFW: 3

   In general, the hardest to represent compactly is the IIIT-CFW. It may have to do with a large number of feature vectors, but even the curve (plotted above) is indicative of such. To achieve similiar representation feature ratios, Yale needs around 70 features, IMFDB needs 110, and CFW needs 300.

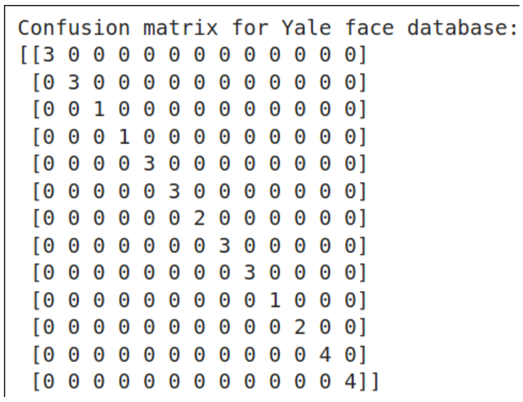
## 1.2 Classification

1. **Use any classifier(MLP, Logistic regression, SVM, Decision Trees) and find the classification accuracy. Which method works well? Do a comparitive study.**
   After running tests with all classifiers, and multiple dimensionality reduction tools, the best methods were evaluated. Here I present the results for the Yale dataset.



```
Dataset: Yale face database
                            Red. Dim  Classif. Error  Accuracy  F1 Score
K-LDA Sigmoid + SVC               14        0.000000  1.000000  1.000000
K-LDA Polynomial + LR             14        0.000000  1.000000  1.000000
K-LDA Polynomial + SVC            14        0.000000  1.000000  1.000000
K-LDA Polynomial + MLP            14        0.000000  1.000000  1.000000
K-LDA RBF + LR                    14        0.000000  1.000000  1.000000
K-LDA RBF + SVC                   14        0.000000  1.000000  1.000000
K-LDA RBF + MLP                   14        0.000000  1.000000  1.000000
K-LDA Sigmoid + MLP               14        0.000000  1.000000  1.000000
K-LDA Sigmoid + LR                14        0.000000  1.000000  1.000000
LDA/Fisherface + LR                7        0.030303  0.969697  0.978022
LDA/Fisherface + SVC               7        0.030303  0.969697  0.978022
LDA/Fisherface + MLP               7        0.030303  0.969697  0.978022
Resnet + MLP                    2048        0.030303  0.969697  0.958974
Resnet + LR                     2048        0.030303  0.969697  0.958974
Resnet + SVC                    2048        0.030303  0.969697  0.958974
K-LDA Sigmoid + DTree             14        0.090909  0.909091  0.917766
K-LDA RBF + DTree                 14        0.090909  0.909091  0.917766
K-LDA Polynomial + DTree          14        0.090909  0.909091  0.917766
LDA/Fisherface + DTree             7        0.212121  0.787879  0.824908
Resnet + DTree                  2048        0.333333  0.666667  0.688617
K-PCA Linear + MLP               110        0.272727  0.727273  0.680403
K-PCA Linear + LR                110        0.272727  0.727273  0.680403
K-PCA Sigmoid + LR               110        0.242424  0.757576  0.672619
K-PCA Sigmoid + MLP              110        0.242424  0.757576  0.672619
PCA/Eigenface + LR               110        0.272727  0.727273  0.642177
PCA/Eigenface + MLP              110        0.272727  0.727273  0.642177
K-PCA RBF + MLP                  110        0.272727  0.727273  0.628231
K-PCA RBF + LR                   110        0.272727  0.727273  0.628231
K-PCA Sigmoid + SVC              110        0.272727  0.727273  0.626349
K-PCA RBF + SVC                  110        0.303030  0.696970  0.625850
K-PCA Polynomial + LR            110        0.333333  0.666667  0.617582
K-PCA Polynomial + MLP           110        0.333333  0.666667  0.617582
K-PCA Cosine + LR                110        0.303030  0.696970  0.602721
K-PCA Cosine + MLP               110        0.303030  0.696970  0.602721
PCA/Eigenface + SVC              110        0.333333  0.666667  0.581803
K-PCA Linear + SVC               110        0.333333  0.666667  0.577041
K-PCA Cosine + SVC               110        0.363636  0.636364  0.539841
K-PCA Cosine + DTree             110        0.393939  0.606061  0.530476
```

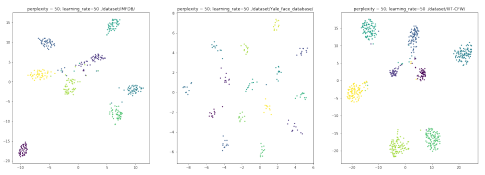
The confusion matrix of the best performing method is:



```
Confusion matrix for Yale face database:
[[3 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 3 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 1 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 3 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 3 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 2 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 3 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 3 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 2 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 4 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 4]]
```

## 1.3 t-SNE

1. **Similiar to 1(b) use t-SNE based visilization of faces? Does it make-sense? Do you see similar people coming together?or something else? Can you do visualization datasetwise and combined?**
   Due to the parameter tuning needed, I ran a script to calculate t-SNE across datasets to get a range of results.
   First, I ran just t-SNE on the data provided. As we can see, there is not much conglomeration, similar people do not come together despite multiple tests.
   As an experiment, I then ran t-SNE on data that was reduced by LDA to 2 dimensions. Here, around perplexity = 30/50 onwards we see conglomeration, especially in the IIIT-CFW dataset. While full separation across all data is not achieved yet, the closest we can get by eyeball estimates are at perplexity = 50 and learning rate = 50.
   Instead of LDA, I then decided to try using Resnet with the same parameters, and the results are promising. I have presented this as the solution.

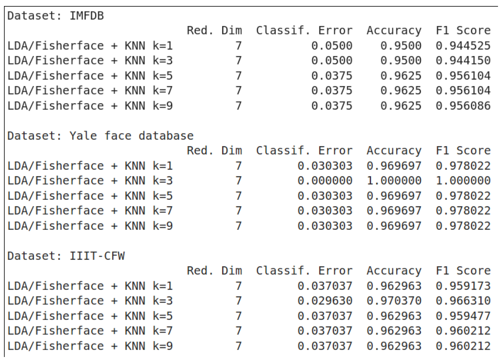


## 1.4 face is used for verification

1. **How do we formulate the problem using KNN?**
   We model the problem in much the same way as in Question 2: use KNN for multiclass classification instead of binary. The key parameter here is to decide the value of k, which can be done by simple experimentation and observing performance.

2. **How do we analyze the performance ? suggest the metrics (like accuracy) that is appropriate for this task.**
   Analyzing the performance is a matter of simply splitting the dataset into train and test, training on the train set and then testing on test. We would look at accuracy: correct classification of the test data as our primary metric, but again as in Question 2 F1-Score is useful, again.

3. **Show empirical results with all the representations**



```
Dataset: IMFDB
                          Red. Dim  Classif. Error  Accuracy  F1 Score
LDA/Fisherface + KNN k=1         7          0.0500    0.9500  0.944525
LDA/Fisherface + KNN k=3         7          0.0500    0.9500  0.944150
LDA/Fisherface + KNN k=5         7          0.0375    0.9625  0.956104
LDA/Fisherface + KNN k=7         7          0.0375    0.9625  0.956104
LDA/Fisherface + KNN k=9         7          0.0375    0.9625  0.956086

Dataset: Yale face database
                          Red. Dim  Classif. Error  Accuracy  F1 Score
LDA/Fisherface + KNN k=1         7        0.030303  0.969697  0.978022
LDA/Fisherface + KNN k=3         7        0.000000  1.000000  1.000000
LDA/Fisherface + KNN k=5         7        0.030303  0.969697  0.978022
LDA/Fisherface + KNN k=7         7        0.030303  0.969697  0.978022
LDA/Fisherface + KNN k=9         7        0.030303  0.969697  0.978022

Dataset: IIIT-CFW
                          Red. Dim  Classif. Error  Accuracy  F1 Score
LDA/Fisherface + KNN k=1         7        0.037037  0.962963  0.959173
LDA/Fisherface + KNN k=3         7        0.029630  0.970370  0.966310
LDA/Fisherface + KNN k=5         7        0.037037  0.962963  0.959477
LDA/Fisherface + KNN k=7         7        0.037037  0.962963  0.960212
LDA/Fisherface + KNN k=9         7        0.037037  0.962963  0.960212
```