SMAI Assignment 2

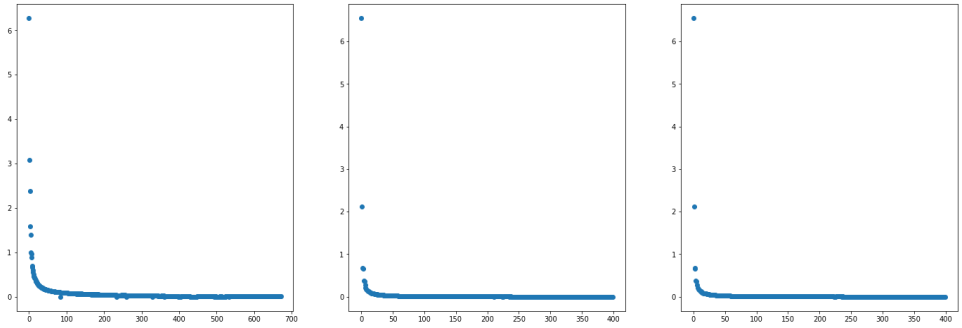
Abhigyan Ghosh 20170189

Eigen Face

Eigenfaces is the name given to a set of eigenvectors when they are used in the computer vision problem of human face recognition.

Eigen Spectrum

The eigenvalue spectrum for the 3 datasets are:



From the eigen value plot, we can see that the minimum number of significant eigenvalues are 11. We use 90% of the sum of the eigenvalues.

Dataset	Eigen Values
CFW-IIIT	207
IMFDB	66
Yale	34

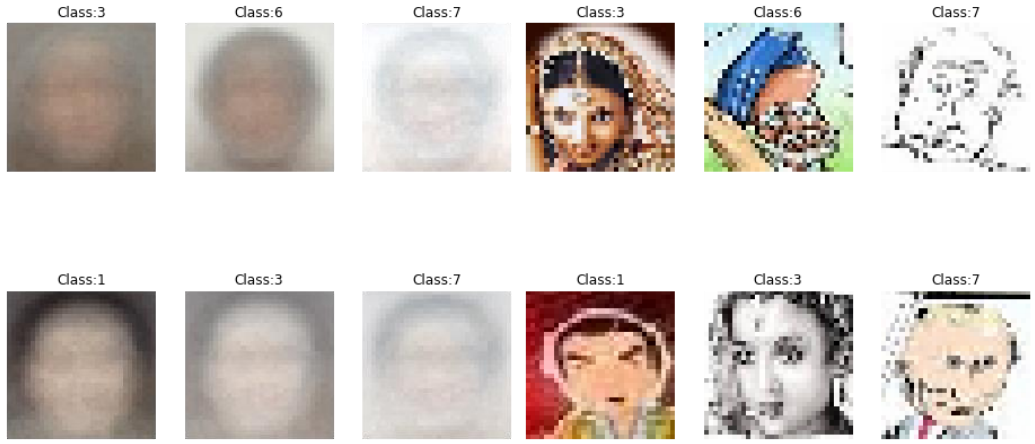
Reconstruction Loss

CFW-IIIT, IMFDB and Yale datasets are projected to 207, 66 and 34 dimensions respectively and then reconstructed back. Reconstruction loss used is MSE loss.

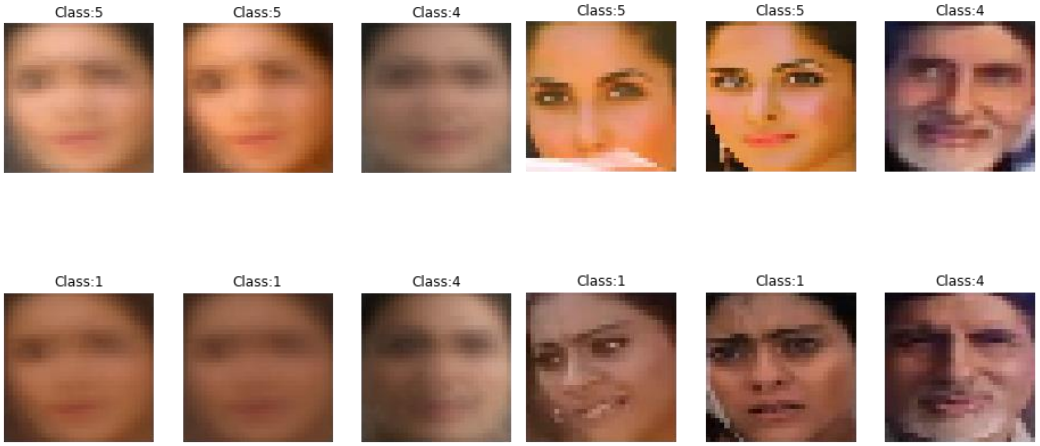
Dataset	Eigen Values
CFW-IIIT	0.23416578143524974
IMFDB	0.1233671529342786
Yale	0.18206447468765843

Here are some of the reconstructed faces randomly sampled from each dataset:

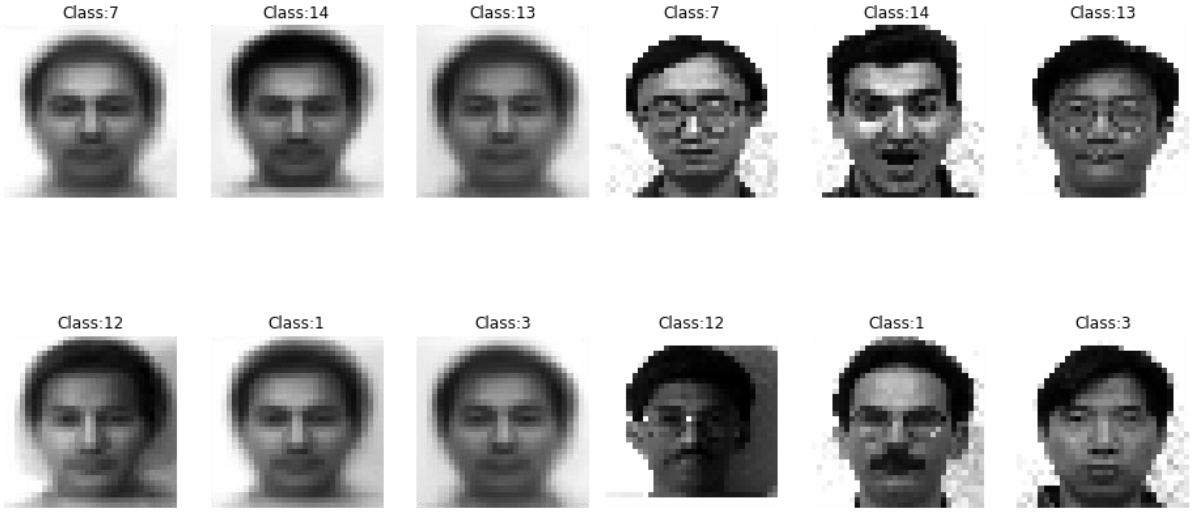
IIIT-CFW



IMFDB



Yale Database



In the following datasets, the reconstruction losses class-wise are:

IIIT-CFW Dataset	
Class Label	Reconstruction Loss
0	0.222971
1	0.238942
2	0.198245
3	0.224615
4	0.223832
5	0.238271
6	0.247141
7	0.228823

The most difficult class to reconstruct is class 6 which is for Manmohan SIngh

IMFDB	
Class Label	Reconstruction Loss
0	0.085976
1	0.113470
2	0.107106
3	0.126598
4	0.138194
5	0.133583
6	0.116460
7	0.133032

The most difficult class to reconstruct is class 4 which is for Amitabh Bachan

Yale Database	
Class Label	Reconstruction Loss
0	0.192097
1	0.171668
2	0.182702
3	0.178547
4	0.169317
5	0.177254
6	0.179727
7	0.184212
8	0.187478
9	0.183212
10	0.193199
11	0.163692
12	0.192094
13	0.164460
14	0.192149

The most difficult class to reconstruct is class 10

MLP Classifier

MLP classifier is trained with one hidden layer having size 100 for each dataset. Each dataset is projected to different feature space and various parameters are calculated. Dataset is split into 20% test set and 80% training data. Below are the results of MLP classification for each dataset.

IIIT-CFW

	Recall	Precision	Accuracy	F1 Score
Eigen Face	0.452111	0.43265	0.444444	0.43259
Kernel Face	0.440129	0.43284	0.444444	0.425573
Fischer Face	0.955629	0.950347	0.955556	0.952436
Kernel Fischer Face	0.969518	0.96598	0.962963	0.966779
VGG Face	0.626119	0.651735	0.674074	0.629921
Resnet Face	0.98382	0.975815	0.977778	0.979284

IMFDB

	Recall	Precision	Accuracy	F1 Score
Eigen Face	0.7692	0.747167	0.75	0.752786
Kernel Face	0.652416	0.60209	0.625	0.599229
Fischer Face	0.975	0.984375	0.9875	0.977778
Kernel Fischer Face	0.975	0.984375	0.9875	0.977778
VGG Face	0.911298	0.922917	0.9125	0.914296
Resnet Face	0.948664	0.945933	0.95	0.945486

Yale Database

	Recall	Precision	Accuracy	F1 Score
Eigen Face	0.77381	0.732143	0.787879	0.733844
Kernel Face	0.555556	0.522222	0.575758	0.504444
Fischer Face	1	1	1	1
Kernel Fischer Face	1	1	1	1
VGG Face	0.483333	0.551111	0.575758	0.482222
Resnet Face	0.964286	0.964286	0.969697	0.952381

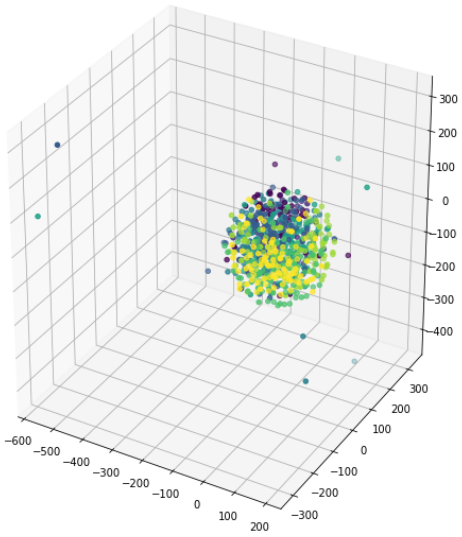
The best classification results and the corresponding confusion matrices for each are:

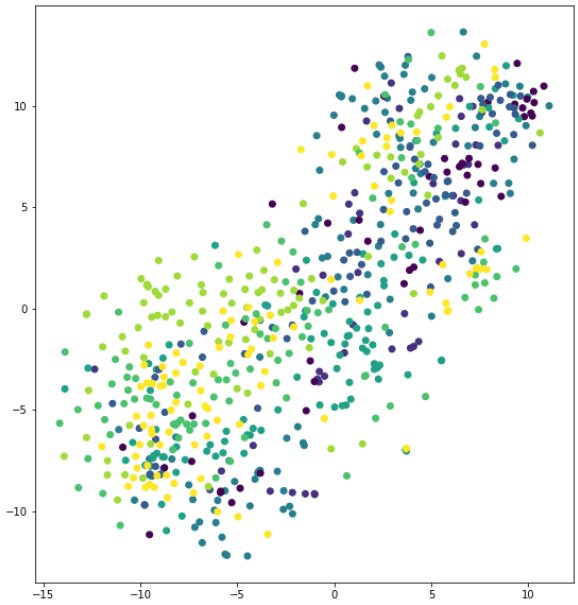
IIIT-CFW: Resnet Face	IMFDB: Kernel Fischer Face	Yale Face Database: Kernel Fischer Face
[[9 0 0 0 0 0 0 0] [0 9 0 0 0 0 0 1] [0 0 19 0 0 0 0 0] [0 0 0 29 0 0 0 0] [0 0 0 0 18 0 0 0] [0 0 0 0 0 15 0 0] [0 0 0 0 0 0 11 0] [0 0 0 1 1 0 0 22]]	[[10 0 0 0 0 0 0 0] [0 7 1 0 0 0 0 0] [0 0 4 0 0 0 0 0] [0 0 0 13 0 0 0 0] [0 0 0 0 11 0 0 0] [0 0 0 0 0 8 0 0] [0 0 0 0 0 0 13 0] [0 0 0 0 0 0 0 13]]	[[2 0 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 2 0 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 1 0 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 0 4 0 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 3 0 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 0 5 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 0 1 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 3]]

TSNE

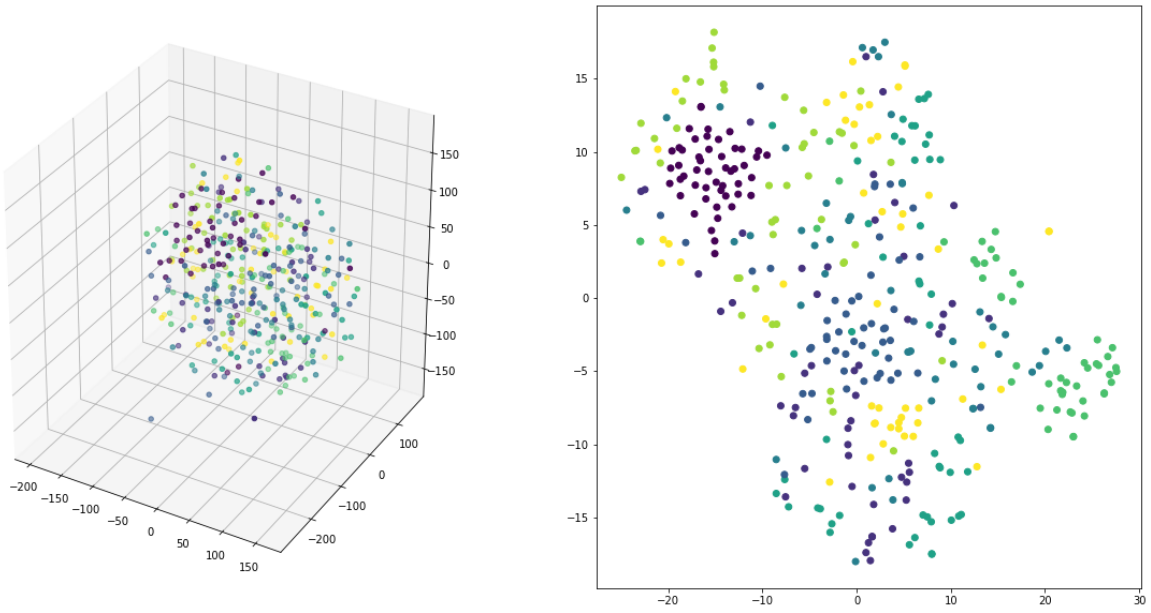
IMFDB, CFW-IIIT, Yale and combined data are projected using t-SNE features. Below are the 3d and 2d projection of data

IIIT-CFW

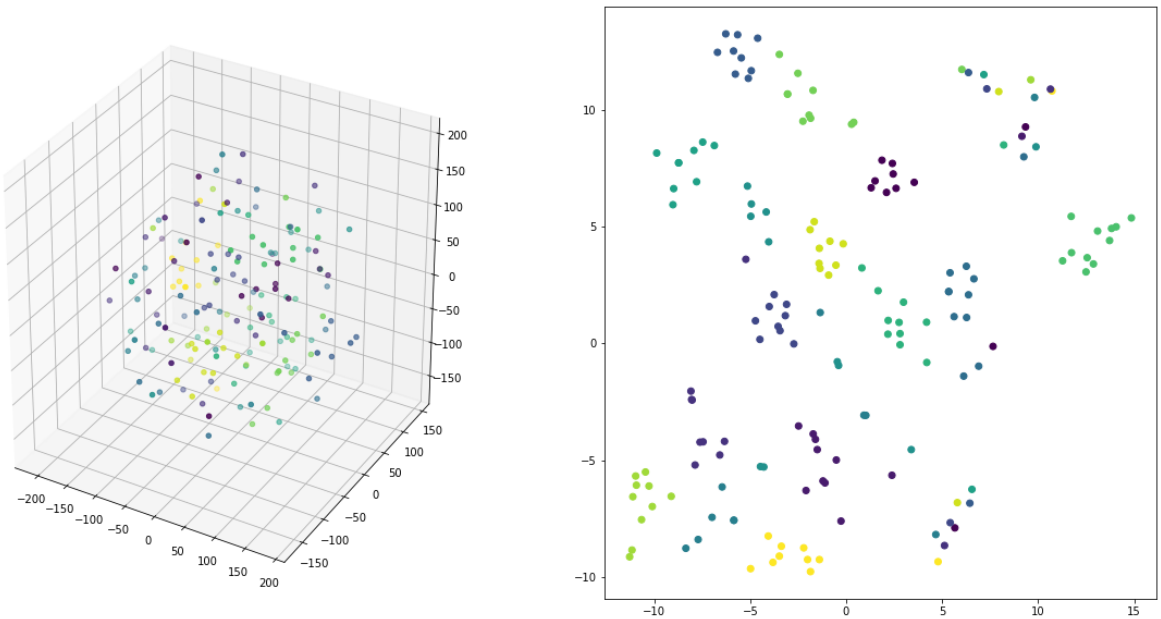




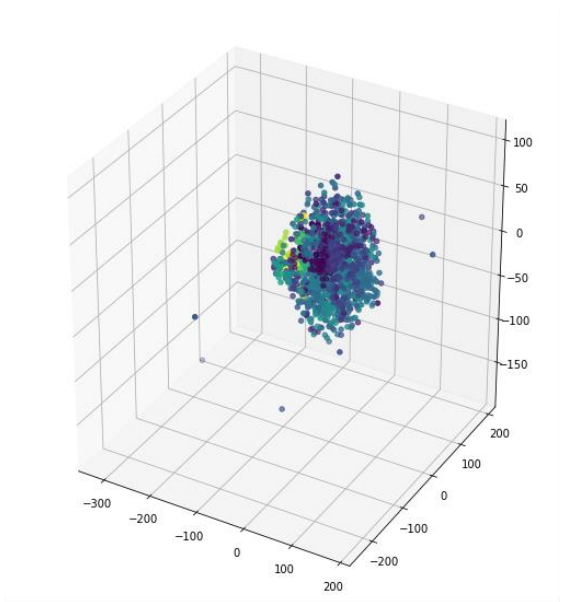
IMFDB

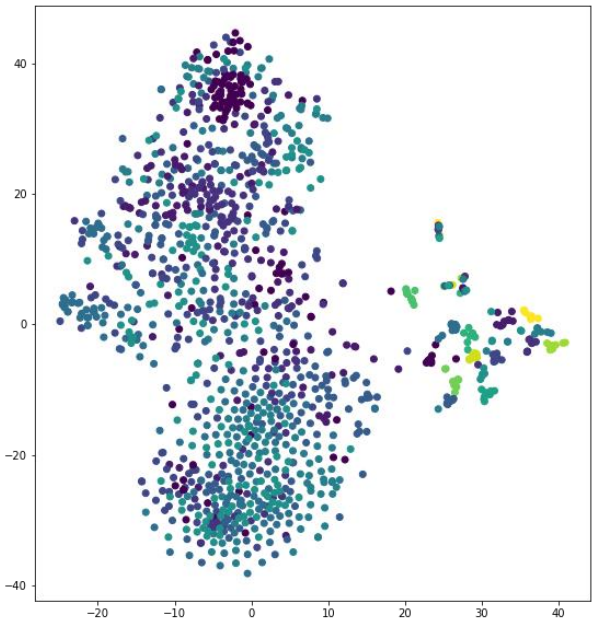


Yale Database



Combined Dataset





Face Verification

KNN algorithm is used to classify by finding the K nearest matches in training data and then using the label of closest matches to predict. So, we project the data into lower dimensions using the above methods and then apply KNN Classification to predict the label. The following are the metrics for 5 nearest neighbours.

IIT-CFW

	Recall	Precision	Accuracy	F1 Score
Eigen Face	0.435675	0.412554	0.392593	0.394082
Kernel Face	0.435060	0.421099	0.414815	0.410049
Fischer Face	0.955629	0.953480	0.955556	0.953621
Kernel Fischer Face	0.955629	0.953480	0.955556	0.953621
VGG Face	0.647482	0.673355	0.681481	0.648618
Resnet Face	0.972456	0.973280	0.970370	0.972103

IMFDB

	Recall	Precision	Accuracy	F1 Score
Eigen Face	0.625187	0.605884	0.5875	0.570322
Kernel Face	0.625187	0.605884	0.5875	0.570322
Fischer Face	0.959375	0.959375	0.9750	0.958333
Kernel Fischer Face	0.959375	0.959375	0.9750	0.958333
VGG Face	0.873836	0.879132	0.8875	0.874352
Resnet Face	0.939048	0.938636	0.9375	0.935293

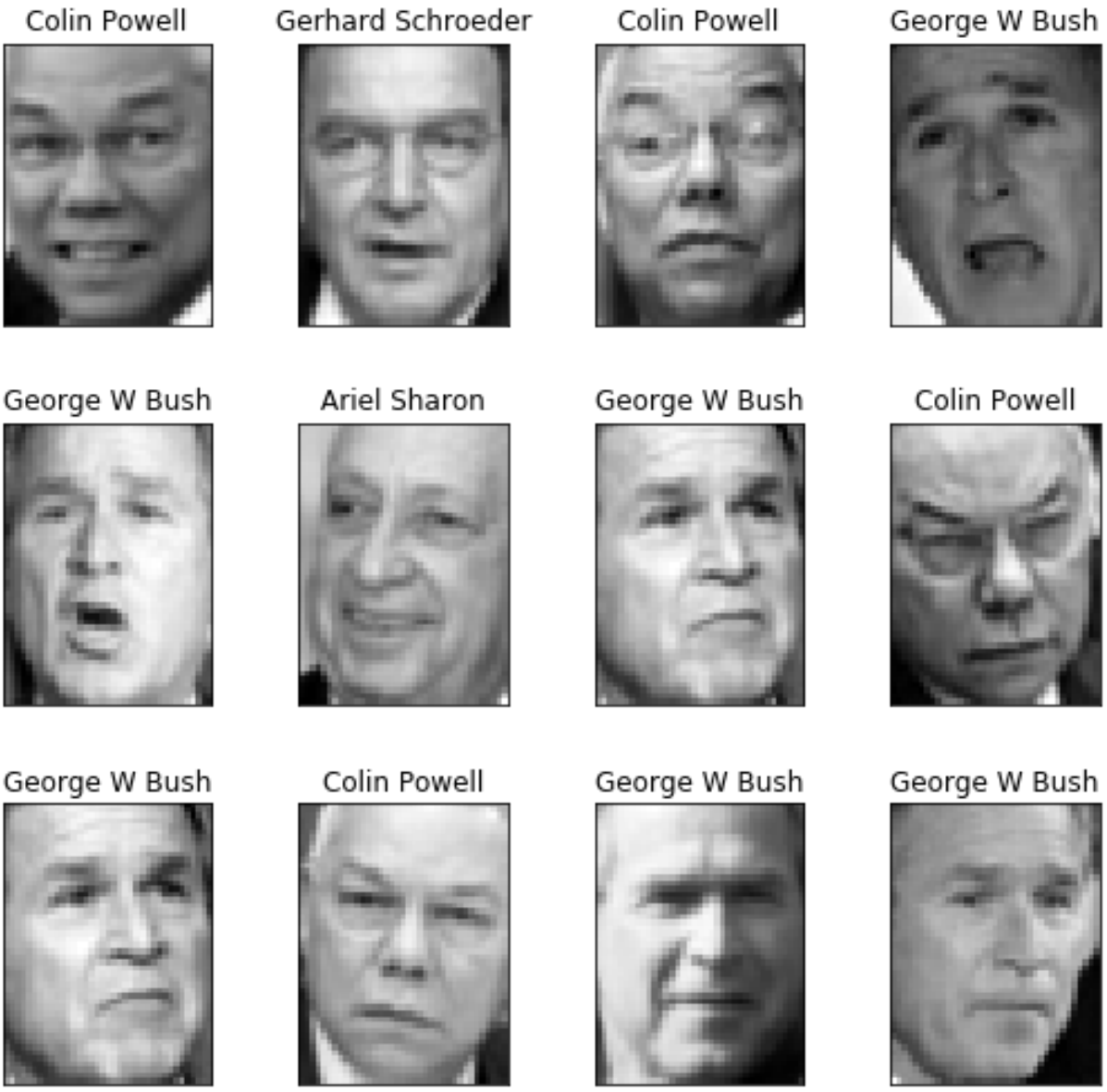
Yale Database

	Recall	Precision	Accuracy	F1 Score
Eigen Face	0.714286	0.690476	0.696970	0.661224
Kernel Face	0.678571	0.678571	0.666667	0.639796
Fischer Face	1.000000	1.000000	1.000000	1.000000
Kernel Fischer Face	1.000000	1.000000	1.000000	1.000000
VGG Face	0.494444	0.413333	0.515152	0.429654
Resnet Face	1.000000	1.000000	1.000000	1.000000

Extension / Application

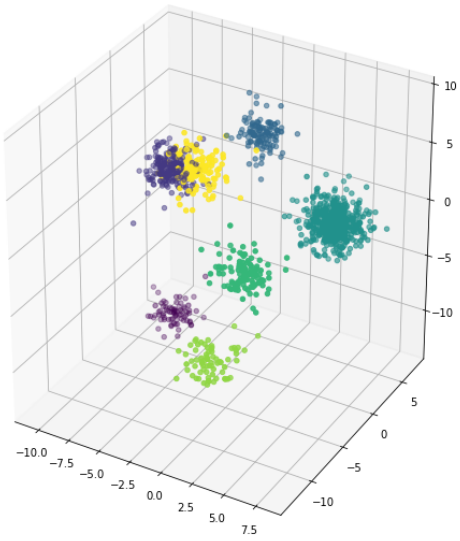
The dataset used in this example is a preprocessed excerpt of the “Labeled Faces in the Wild”, aka LFW. The dataset contains 1288 samples with 1850 features of 7 people: Ariel Sharon, Colin Powell, Donald Rumsfeld, George W Bush, Gerhard Schroeder, Hugo Chavez and Tony Blair. The dataset can not only be applied to classify these faces, but the faces of other people and as a good system of facial recognition.

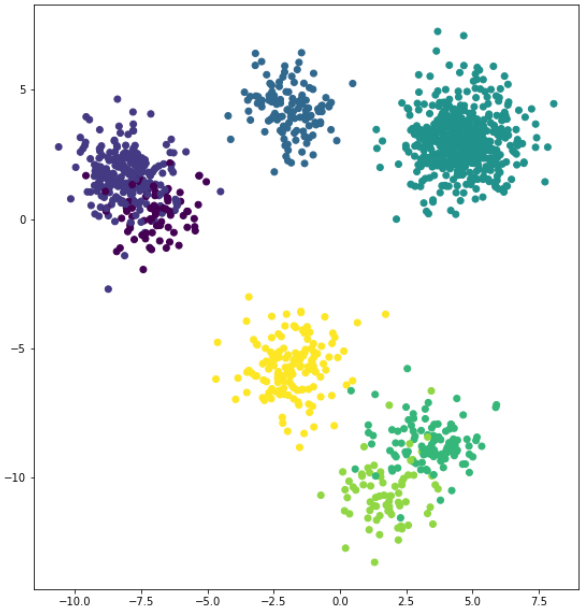
We will do a simple identification by classifying the images into their respective classes.



Qualitative Analysis

The following is the plot after projecting the dataset into 3d and 2d using simple LDA





As we can see, that the data is well clustered on applying LDA. Thus, the best approach here would be to use the KNN classifier after projecting the data into 3d using LDA. The following were the metrics in the K-Fold Validation Test of the classifier:

Metric	Score
Accuracy	1.0
Precision	1.0
Recall	1.0
F1 Score	1.0