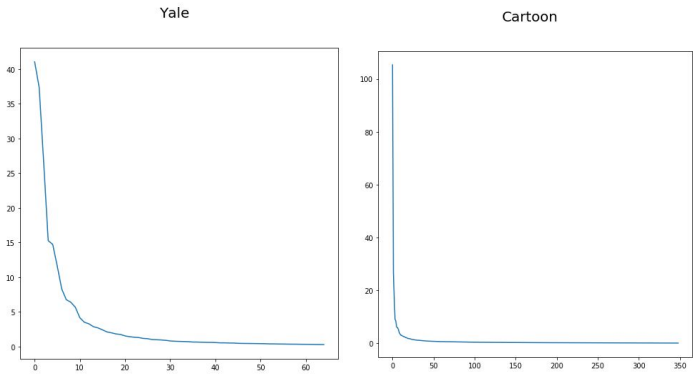
SMAI Assignment II

Deepti Mahesh - 20171212

1.1 Eigen Faces

- Analogous to eigenvectors and eigenvalues, eigenfaces are an orthogonal basis set that describe face information. Any face can be reconstructed from a suitable linear combination of the eigen faces.
- In the context of human face recognition, eigenfaces are a set of eigenvectors of the face data covariance matrix - the principal components of a given distribution of faces. Eigenfaces are an efficient representation of face images,
  and a good indicator of relevant facial information.

1.2 Eigenvalue Spectrum



The first few primary components are close to each
other - a higher importance to these components, and the primary eigenvectors are able to capture a lot of the variance of the image. Require smaller number of principal components for satisfactory reconstruction.

- **Yale < IMFDB < Cartoon**

1.3 Reconstruction

The number of eigenvectors needed to withhold **95% variance** is as displayed. It is apparent to see that Cartoon dataset needs highest amount of eigenvectors to capture required variance. This is expected, since the cartoon database has very less features which capture more variance , in contrast to real faces which have more dense features with considerable variances.

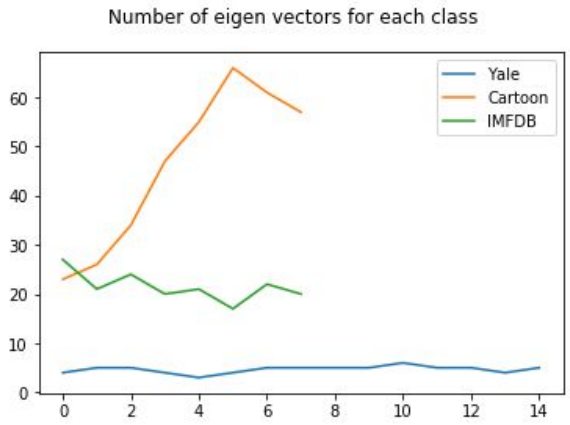
```
Yale:
Dataset shape: (165, 64, 64)
65

IMFDB:
113

Cartoon:
Dataset shape: (672, 64, 64)
349
```

To achieve a representation of about ~50% of the original image, IMFDB < Yale components. But to reach a satisfactory reconstruction, reverse trend is observed - more components are required.

1.4 Difficult Persons/Classes



For each dataset, the reconstruction error per class was computed. The difficulty level of the highest reconstruction error classes in dataset
CFW > IMFDB > Yale and this holds true for even class-wise eigenvector representation.

Based on Optimal #Eigenvectors:
Cartoon: Narendra Modi
IMFDB : Madhuri Dixit
Yale : Class 10
Minimum Eigen vectors for IMFDB: Katrina Kaif.

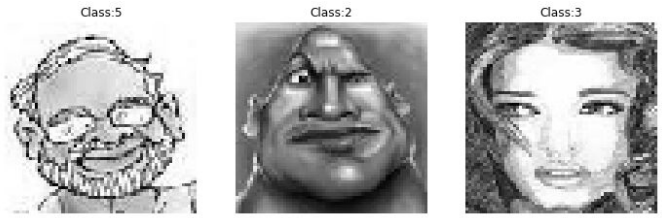
To find the most difficult ones, number of eigenvectors needed to achieve >95% variance for each class was computed.
We know that **maximizing variance implies minimizing reconstruction error** and from the results proved we can see that by taking a high number of components while performing PCA on the entire dataset, we essentially reduce the reconstruction error for Class 0.

In the **Cartoon Face** dataset, Manmohan Singh has the most variations, since their images
consists of various styles, emotions and colour schemes. Thus most errors can be attributed to bias in datasets.

2.1 Classifiers

The classification algorithm follows the pipeline:
1. Load the dataset
2. Create a test_train split of the data
3. Extract a feature representation
5. Train the classifier on the train split
6. Validate the classifier and report the performance metrics

Cartoon:



Features used : PCA,KPCA,LDA,KLDA,VGG,ResNet
Classifiers : MLP, SVM, LR, Decision Tree

The train test split for all these datasets was 75% and 25%. The best classifier for each dataset were:
Yale: MLP, **IMFDB:** LR, **IIIT-CFW:** SVM
The parameters used were:

- MLPClassifier(hidden_layer_sizes=(200,50), max_iter=4000)
- LogisticRegression(random_state=0, solver='lbfgs', multi_class = 'ovr', max_iter=4000)
- SVC(kernel='linear', C=3, max_iter=4000, gamma = 0.1)
- tree.DecisionTreeClassifier()

2.2 Comparative Study

After trying a variety of combinations, these were some of the diverse results found. Resnet with Kernel LDA and LDA yielded best results each time, while VGG, Decision Trees, PCA yielded considerably lower accuracies.
However seeing as PCA is unsupervised dimensionality reduction, the results are not surprising.
100% is often achieved by ResNet features are pre trained to obtain to make the data as linearly separable as possible.

Table for Yale_face_database

	Method	Reduced Dim	Classification Error	Accuracy	F1 Score
1	Resnet + MLP	2048	0.000000	100.000000	100.000000
2	Resnet + LR	2048	2.380952	97.619048	96.444444
3	Resnet + SVM	2048	2.380952	97.619048	96.444444
4	Resnet + LDA + MLP	7	0.000000	100.000000	100.000000
5	Resnet + LDA + LR	7	0.000000	100.000000	100.000000
6	PCA + MLP	100	19.047619	80.952381	80.253968
7	PCA + SVM	100	11.904762	88.095238	87.195767
8	Resnet + rbf KernelLDA + MLP	7	0.000000	100.000000	100.000000
9	VGG + LR	4096	45.238095	54.761905	53.722944
10	VGG + SVM	4096	45.238095	54.761905	53.722944

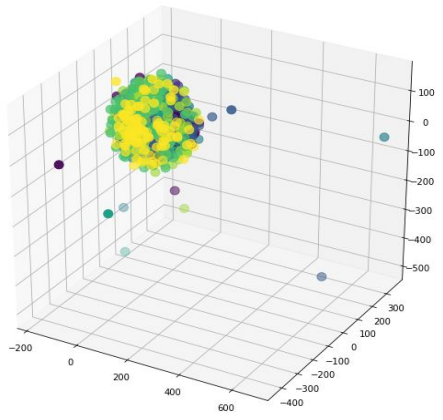
Table for IMFDB

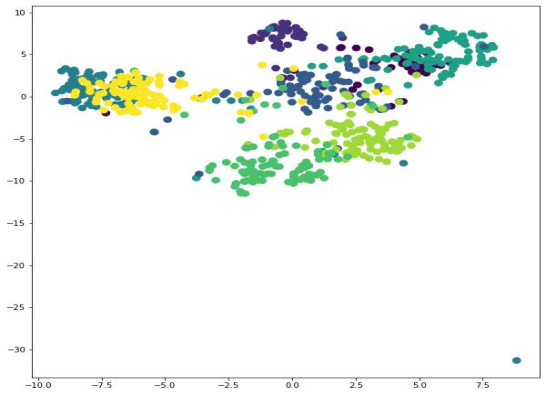
	Method	Reduced Dim	Classification Error	Accuracy	F1 Score
1	Resnet + MLP	2048	4.0	96.0	95.454343
2	Resnet + LR	2048	5.0	95.0	94.589579
3	Resnet + SVM	2048	6.0	94.0	93.964836
4	Resnet + LDA + MLP	7	0.0	100.0	100.000000
5	Resnet + LDA + LR	7	0.0	100.0	100.000000
6	PCA + MLP	100	35.0	65.0	64.819327
7	PCA + SVM	100	29.0	71.0	70.787246
8	Resnet + rbf KernelLDA + MLP	7	0.0	100.0	100.000000
9	VGG + LR	4096	8.0	92.0	92.489688
10	VGG + SVM	4096	8.0	92.0	92.489688

Table for IIIT-CFW

	Method	Reduced Dim	Classification Error	Accuracy	F1 Score
1	Resnet + MLP	2048	2.380952	97.619048	97.396010
2	Resnet + LR	2048	2.380952	97.619048	97.395087
3	Resnet + SVM	2048	2.976190	97.023810	96.605312
4	Resnet + LDA + MLP	7	0.000000	100.000000	100.000000
5	Resnet + LDA + LR	7	0.000000	100.000000	100.000000
6	PCA + MLP	100	55.357143	44.642857	45.585992
7	PCA + SVM	100	60.119048	39.880952	39.177926
8	Resnet + rbf KernelLDA + MLP	7	0.000000	100.000000	100.000000
9	VGG + LR	4096	27.976190	72.023810	67.598383
10	VGG + SVM	4096	27.976190	72.023810	67.598383

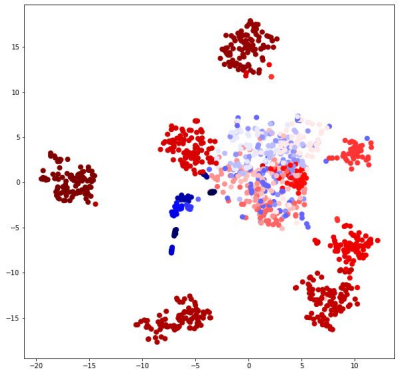
3 TSNE





- Takes a set of points in a high-dimensional space and find a faithful representation of those points in a lower-dimensional space, typically the 2D plane.
- Tunable parameter, “perplexity,” which deals with how to balance attention between local and global aspects of your data. Typical values are between 5 and 50.
- t-SNE naturally expands dense clusters, and contracts sparse ones, evening out cluster sizes. Therefore two differently sized clusters look the same.

Since **perplexity is a global parameter**, there may not be one perplexity value that will capture distances across all clusters. Therefore, distances between well-separated clusters in a t-SNE plot may mean nothing. Thus, usually unclustered except in the case of combined dataset.



Here, **Blue** is the Yale dataset, the **lighter colors** are the IMFDB dataset and **Red** implies Cartoon dataset. Thus you can see the datasets cluster together and it is easy to differentiate between them.

4 KNN

Parameters: KNeighborsClassifier (n_neighbors = 5)

For k = 7

Predicted / Actual	0	1	2	3	4	5	6	7
0	10	1	2	1	1	0	1	1
1	0	4	1	1	4	0	2	2
2	0	0	6	4	3	0	0	1
3	0	0	0	1	0	1	0	1
4	0	0	0	1	4	0	0	0
5	0	0	0	0	0	1	0	1
6	0	1	0	0	0	0	1	0
7	0	0	0	1	0	0	1	1

```
------------------------------
Accuracy = 46.67%
Precision = 47.11
Recall = 45.26
F1 Score = 41.07
```

Taking 5 as default value, k = 7, 11, 17 are tried out, giving different results. K > 17 was found to overfit and thus drastically decrease in accuracy for IMFDB dataset.

After trying a variety of combinations, these were some of the diverse results found.

Table for Yale_face_database

	Method	Reduced Dim	Classification Error	Accuracy	Precision
1	Resnet + KNN	2048	0.000000	100.000000	100.000000
2	KNN, k = 11	2048	28.571429	71.428571	73.214286
3	KNN, k = 43	2048	78.571429	21.428571	32.511905
4	Resnet + sigmoid KernelLDA + KNN	7	0.000000	100.000000	100.000000
5	Resnet + LDA + KNN	7	0.000000	100.000000	100.000000
6	PCA + KNN	100	23.809524	76.190476	80.555556
7	Resnet + rbf KernelLDA + KNN	7	0.000000	100.000000	100.000000
8	VGG + KNN, k = 5	4096	47.619048	52.380952	59.230769
9	VGG + KNN, k = 11	4096	66.666667	33.333333	44.583333
10	poly KernelLDA + KNN	7	0.000000	100.000000	100.000000

Table for IMFDB

	Method	Reduced Dim	Classification Error	Accuracy	Precision
1	Resnet + KNN	2048	4.0	96.0	95.833333
2	KNN, k = 11	2048	61.0	39.0	53.879270
3	KNN, k = 43	2048	68.0	32.0	29.336363
4	Resnet + sigmoid KernelLDA + KNN	7	0.0	100.0	100.000000
5	Resnet + LDA + KNN	7	0.0	100.0	100.000000
6	PCA + KNN	100	57.0	43.0	52.492709
7	Resnet + rbf KernelLDA + KNN	7	0.0	100.0	100.000000
8	VGG + KNN, k = 5	4096	9.0	91.0	90.623283
9	VGG + KNN, k = 11	4096	5.0	95.0	95.757212
10	poly KernelLDA + KNN	7	0.0	100.0	100.000000

Table for IIIT_Cartoon

	Method	Reduced Dim	Classification Error	Accuracy	Precision
1	Resnet + KNN	2048	2.380952	97.619048	97.865172
2	KNN, k = 11	2048	65.476190	34.523810	44.234165
3	KNN, k = 43	2048	62.500000	37.500000	53.055907
4	Resnet + sigmoid KernelLDA + KNN	7	0.000000	100.000000	100.000000
5	Resnet + LDA + KNN	7	0.000000	100.000000	100.000000
6	PCA + KNN	100	61.904762	38.095238	41.739258
7	Resnet + rbf KernelLDA + KNN	7	0.000000	100.000000	100.000000
8	VGG + KNN, k = 5	4096	33.928571	66.071429	64.281290
9	VGG + KNN, k = 11	4096	33.333333	66.666667	65.117696
10	poly KernelLDA + KNN	7	0.000000	100.000000	100.000000

Yet again, ResNet gives 100% accuracy.

97.61904761904762

Predicted	0	1
Actual		
0	77	1
1	3	87

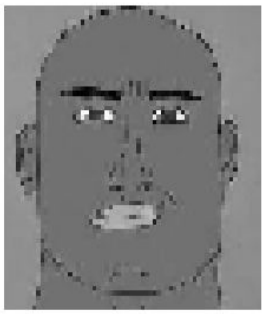
5 Extension

Politicians vs Film stars in IIIT-CFW

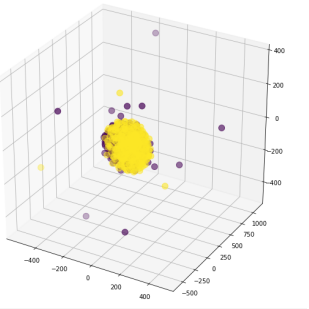
0 is taken as Film Star and 1 as Politician

After taking PCA, the scatter plot shows significant clustering even in 2 dimensions. Thus achieving hundred percent accuracy is quite easy. With shuffle in test and train data, Resnet gave 100% accuracy almost 80% of the time. However, it is apparent that there are significant outliers in the dataset which mess with the models, and thus get classified wrongly. Oftentimes, Amir Khan is classified as Vladimir and vice versa. Dwayne Johnson as Barack Obama and vice versa.

0 wrongly classified as 1



0 correctly classified as 0



TSNE did not (as expected) divide the clusters as viewed by us, but gave increasingly better results as iterations increased.

After PCA: