Assignment-2 Report

Rollno. 20171107

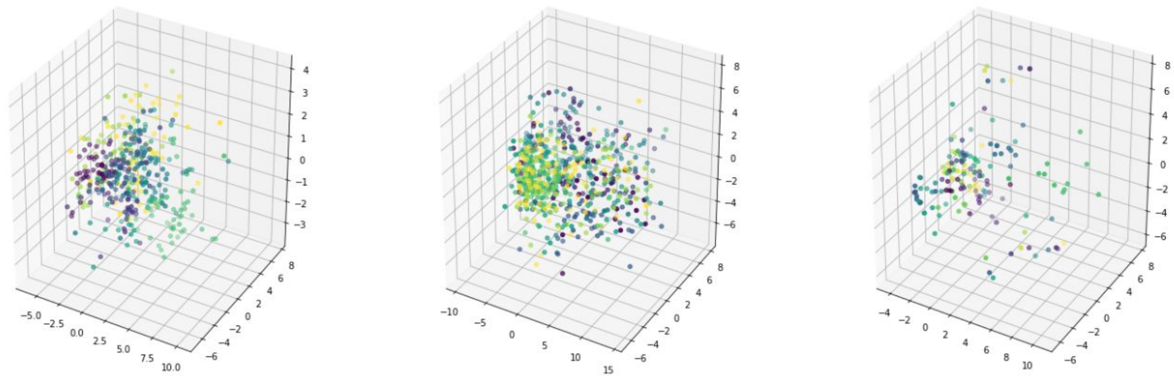
1(a). What are eigenfaces?

Ans: Eigen Faces are a set of vectors which help to visualise images in compressed forms.They are used as basis for images which help in dimensionality reduction.So, eigenfaces are just a name given to eigenvectors when we are working with images.

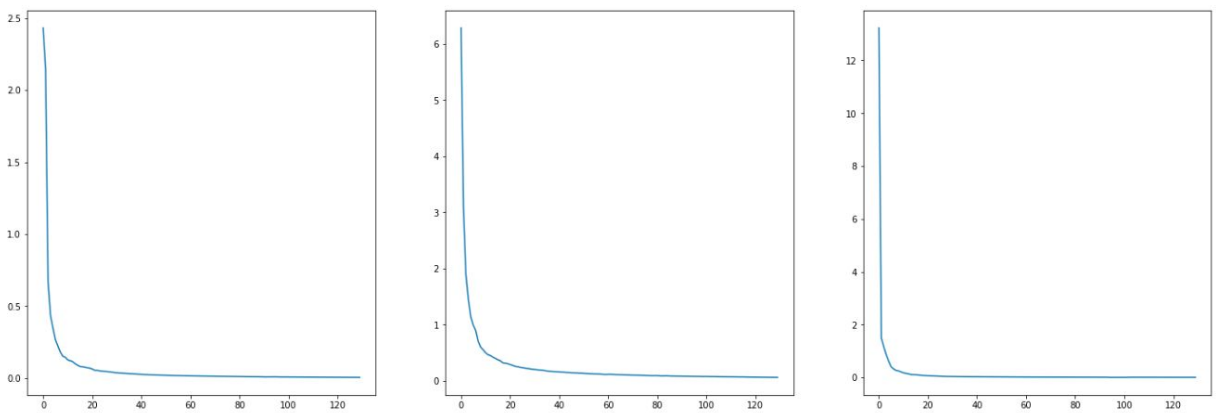
1(b). How many eigenvectors/ eigenfaces are required to “satisfactorily” reconstruct a person in these three datasets? (Don’t forget to make your argument based on eigenvalue spectrum) Show appropriate graphs, qualitative examples and make a convincing argument.

Ans: We can look at the eigenvalue spectrum to find out how many eigenvectors are significant as most of them are very small (~zero).In most cases we use the eigenvectors which preserve more than 95% of the information and we discard the rest.

3D-Scatter Plot

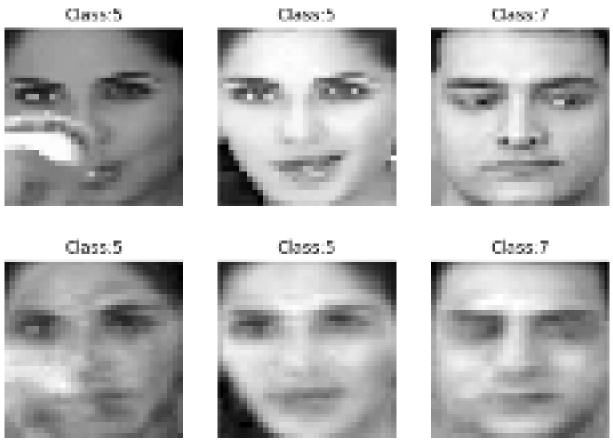


EigenValue Spectrum



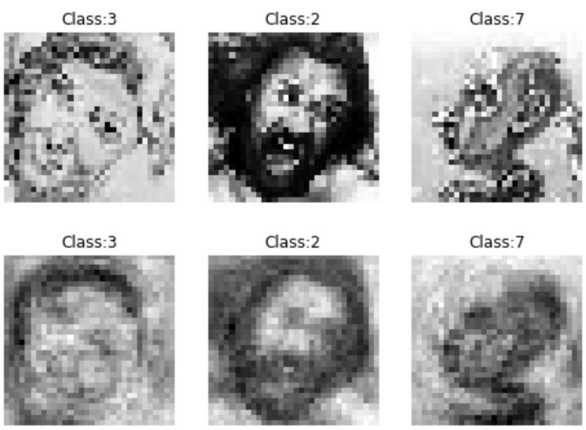
Reconstructed Images

Dataset-1:



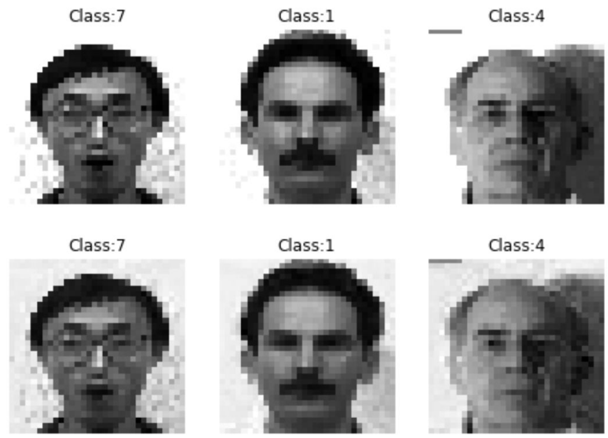
0.8557511587316354186

Dataset-2:



0.1286595160099747

Dataset-3:



0.031572327548944734

1(d). Which person/identity is difficult to represent com-pactly with fewer eigenvectors? Why is that? Explain with your empirical observations and intuitive answers

Ans: First we find out the respective images in each database with the maximum error,
Dataset-1 : Image-3
Dataset-2 : Image-5
Dataset-3 : Image-7
We can explain this by seeing that these images have a lot more variance than the respective images in the datasets.
Moreover in the dataset-2, the error is relatively higher than the other two as the number of images is high so reconstruction would need more eigenvectors, hence less accuracy and high error rate.

2.

Feature/combination of feature used, reduced dimension space, classification error, accuracy, f1-score Table

Results for dataset-1

Feature	Red dimen.	Error	Accuracy	F1-score
PCA with SVM	50	0.32499999999999996	0.675	0.675
PCA with LOGISTIC	50	0.19999999999999996	0.8	0.8000000000000002
PCA with MLP	50	0.30000000000000004	0.7	0.7

Results for dataset-2

Feature	Red dimen.	Error	Accuracy	F1-score
PCA with SVM	50	0.5777777777777777	0.4222222222222222	0.4222222222222222
PCA with LOGISTIC	50	0.5481481481481482	0.45185185185185184	0.45185185185185184
PCA with MLP	50	0.5407407407407407	0.45925925925925926	0.45925925925925926

Results for dataset-3

Feature	Red dimen.	Error	Accuracy	F1-score
PCA with SVM	50	0.030303030303030276	0.9696969696969697	0.9696969696969697
PCA with LOGISTIC	50	0.12121212121212122	0.8787878787878788	0.8787878787878788
PCA with MLP	50	0.2727272727272727	0.7272727272727273	0.7272727272727273

Confusion Matrices

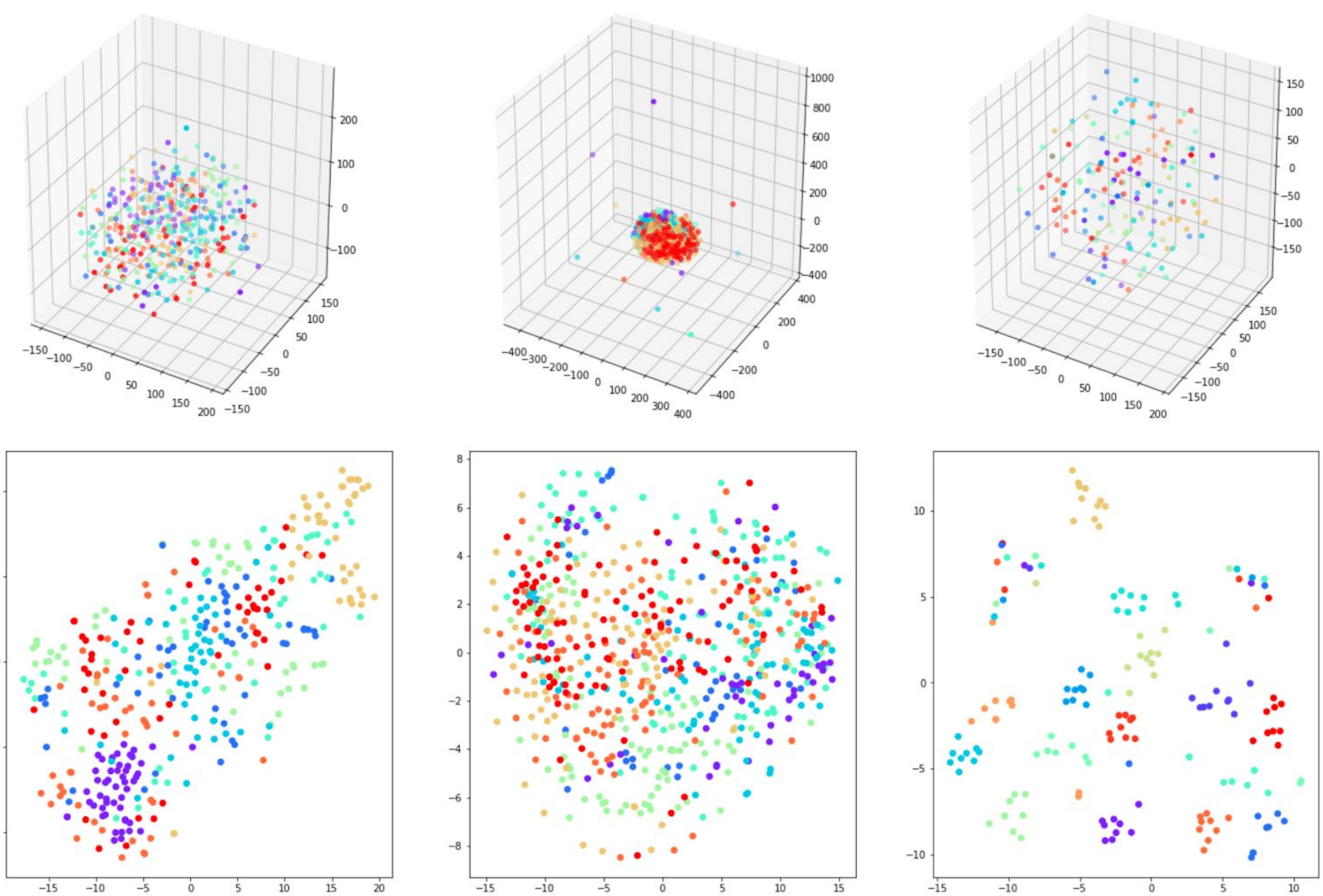
```
Confusion Matrix for dataset-1

[[13  1  0  0  0  0  0  0]
 [ 0 10  0  0  0  0  0  0]
 [ 1  0  6  0  0  0  0  0]
 [ 0  0  0  8  0  0  0  0]
 [ 0  0  0  0 10  0  0  0]
 [ 0  0  0  0  0 10  0  0]
 [ 0  0  0  0  0  0 12  0]
 [ 0  0  1  0  0  0  0  8]]


Confusion Matrix for dataset-2

[[13  0  0  0  0  0  0  0]
 [ 0  8  0  0  0  0  0  0]
 [ 0  0 14  0  0  0  0  0]
 [ 0  1  0 15  0  1  0  0]
 [ 0  0  0  0 21  0  0  0]
 [ 0  0  0  0  0 24  0  0]
 [ 0  0  0  0  0  0 18  0]
 [ 0  0  0  0  0  1  0 19]]


Confusion Matrix for dataset-3

[[2 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 2 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 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 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 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 2 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 5]]
```

3. About t-SNE:

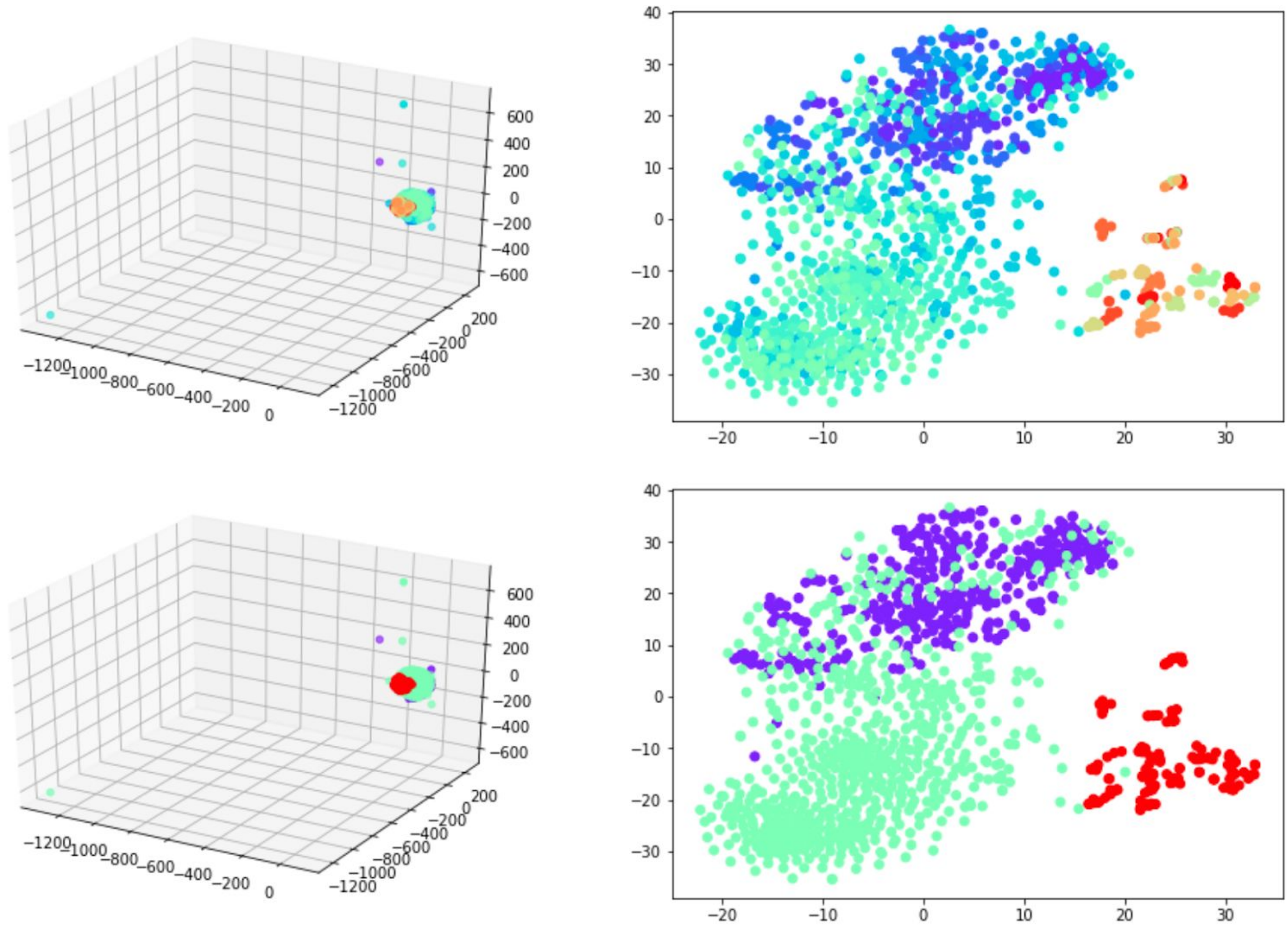
t-Stochastic neighbour embedding (t-SNE) is a dimensionality reduction technique that minimises the divergence between the two systems. It does dimensionality reduction by identifying clusters based on similarity of data points with multiple features.This brings all the points of the same cluster close and makes it easy to apply the model of multi-class classification.

Now to visualize the data, we first plotted the class distribution of each dataset in 2D and 3D, then a plot of all the combined datasets.

t-SNE:



Combined Datasets:



4.a) Formulate the problem using KNN:

We will first divide the dataset into training data and testing data then we will use the training data to calculate the k neighbours for test points. We can test the model then by sending a random image with any label and checking if it is correct or not.

b) The performance can be calculated by analysing the accuracy, but it may occur that in some cases one of the classes is massive in comparison to the other classes and the data in it is getting classified correctly, then in such a case we will need to use metrics such as recall and precision, which can be calculated as:

Recall = TP/(TP + FN)

Precision = TP/(TP + FP)

c) The accuracy obtained in the various models increases as the value of k in the KNN function increases.

Dataset-1

Feature	Red dimen.	Error	Accuracy	Precision
PCA	20	0.10833333333333334	0.8916666666666667	0.6111111111111112
KPCA	20	0.11666666666666667	0.8833333333333333	0.3333333333333333
LDA	2	0.075	0.925	0.7857142857142857
KLDA	2	0.10833333333333334	0.8916666666666667	0.3
VGG	-	0.025	0.975	0.8888888888888888
RESNET	-	0.008333333333333333	0.9916666666666667	1.0

Dataset-2

Feature	Red dimen.	Error	Accuracy	Precision
PCA	20	0.15346534653465346	0.8465346534653465	0.34615384615384615
KPCA	20	0.16336633663366337	0.8366336633663366	0.41935483870967744
LDA	2	0.034653465346534656	0.9653465346534653	0.8888888888888888
KLDA	2	0.024752475247524754	0.9752475247524752	0.92
VGG	-	0.07425742574257425	0.9257425742574258	0.8076923076923077
RESNET	-	0.009900990099009901	0.9900990099009901	0.9545454545454546

Dataset-3

Feature	Red dimen.	Error	Accuracy	Precision
PCA	20	0.04	0.96	0.8333333333333334
KPCA	20	0.06	0.94	0.8
LDA	2	0.02	0.98	1.0
KLDA	2	0.08	0.92	0.3333333333333333
VGG	-	0.1	0.9	0.16666666666666666
RESNET	-	0.0	1.0	1.0

Extension / Application

Problem- Take an image and classify it as cartoon or real
Datasets- IIITCFW and Yale-Face Dataset
Pipeline-

1. Firstly load both datasets and label the datasets with a class label with 1 for cartoon and 0 to real images respectively.
2. Then we shuffle and then split the dataset into 80-20 ratio.
3. Then find out the accuracy, precision and error by applying KNN to the testing data.
4. Then finally we just have to display the results which can be done by plotting the graphs.

Performance Criteria: Accuracy and precision would be the determining criteria for the algorithm as stated earlier.

Results: We get quite high accuracy and precision due to the fact that our data was already grouped together.

```
Precision:  0.9594594594594594
Accuracy:  0.9583333333333334
Error:  0.041666666666666664
```