CSE 475: Statistical Methods in AI	Date: November, 2019
Roll No.: 20171177	

Face Classification/Verification

1 Basic Questions

1. Eigen Faces

   - Eigenfaces is a set of eigenvectors used in face recognition. The eigenvectors are derived from the covariance matrix of features of the various examples available. The eigenfaces form a basis set for the images. This can produce a dimension reduction by allowing a subset of such images to represent the original training images. Classification can be achieved by comparing how faces are represented using these basis set.

2. Number of eigenfaces required: This is decided by choosing eigenvalues such that the sum of eigenvalues turns out to be at least 95% of the total eigenvalues.

   - Yale Dataset: 61 eigenvectors are enough to represent the dataset.

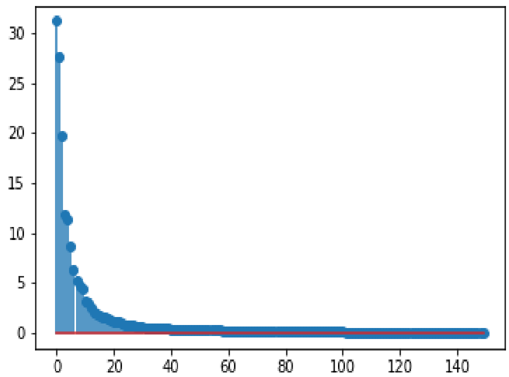

Figure 1: Yale dataset

   - CFW Dataset: 308 eigenvectors are enough to represent the dataset.

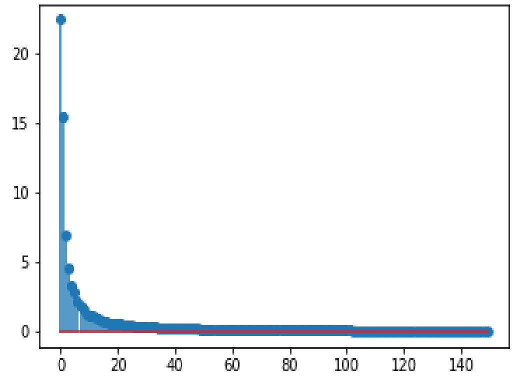

Figure 2: CFW dataset

   - IMDB Dataset: 123 eigenvectors are enough to represent the dataset.

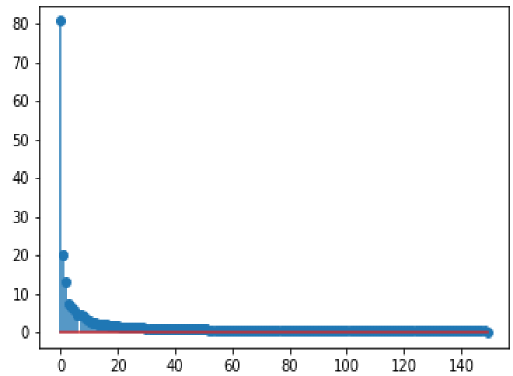

Figure 3: IMDB dataset

3. Person most difficult to represent

   - Yale Dataset: Class 14 is most difficult to represent
   - CFW Dataset: Class 1 is most difficult to represent
   - IMDB Dataset: Class 14 is most difficult to represent

4. Dataset most difficult to represent compactly:
   Error for Yale dataset: 5.45%

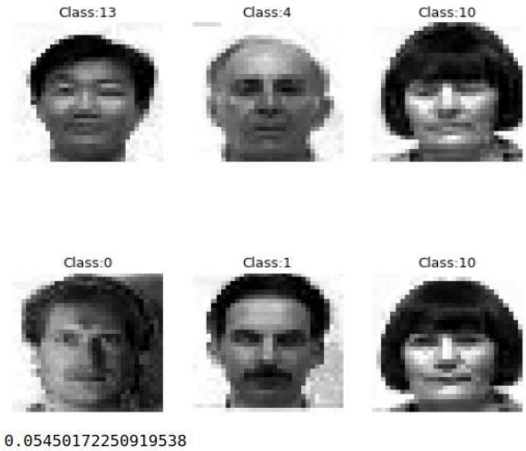

Figure 4: Reconstruction error for Yale dataset

Error for IMDB dataset: 3.87%

Figure 5: Reconstruction error for IMDB dataset

Error for CFW dataset: 6.84%

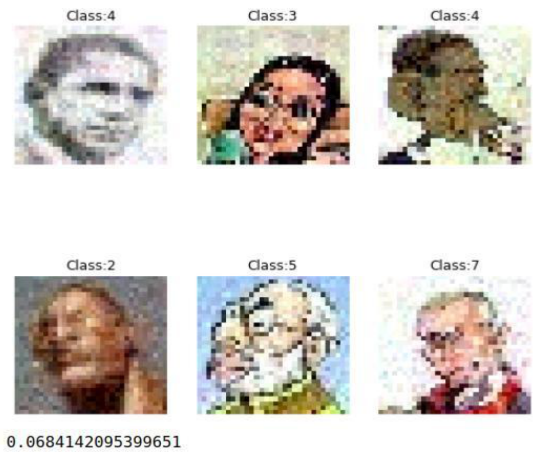

Figure 6: Reconstruction error for CFW dataset

We can see from the error values for reconstruction error that CFW dataset is the most difficult to represent using fewer eigenvectors.

2 Classification Accuracy

1. It can be seen that ResNet provides the best results among all the representations used.

2. Yale Dataset:
   Logistic Regression with LDA and Kernel LDA give the best results with 100% accuracy.

Feat./Modl.	Red. Dim.	Error	Accu.
LDA/MLP	10	0.02	0.98
LDA/Logistic Regression	10	0.0	1.0
Kernel LDA/Logistic Regression	10	0.0	1.0
ResNet/MLP	2048	0.02	0.98
ResNet/Logistic Regression	2048	0.02	0.98
ResNet/SVM	2048	0.02	0.98

Table 1: Analysis of Yale face dataset

3. IMDB Dataset:
   Logistic Regression and SVM along with ResNet feature representation works best in this dataset with 93% accuracy.

Feat./Modl.	Red. Dim.	Error	Accu.
VGG/MLP	4096	0.13	0.87
VGG/Logistic Regression	4096	0.12	0.88
VGG/SVM	4096	0.13	0.87
ResNet/MLP	2048	0.06	0.94
ResNet/Logistic Regression	2048	0.07	0.93
ResNet/SVM	2048	0.07	0.93
ResNet/D-Tree	2048	0.13	0.87

Table 2: Analysis of IMDB face dataset

4. CFW Dataset:
   ResNet feature representation work best with any of the classification models.

Feat./Modl.	Red. Dim.	Error	Accu.
ResNet/MLP	2048	0.04	0.96
ResNet/Logistic Regression	2048	0.04	0.96
ResNet/SVM	2048	0.04	0.96
ResNet/D-Tree	2048	0.08	0.92

Table 3: Analysis of CFW face dataset

3 t-SNE Visualization of faces

t-SNE is a technique to visualize high dimentional datasets. Below are some of the t-SNE plots for the 3 datasets and their combination. Refer to the notebook for all 36 plots.

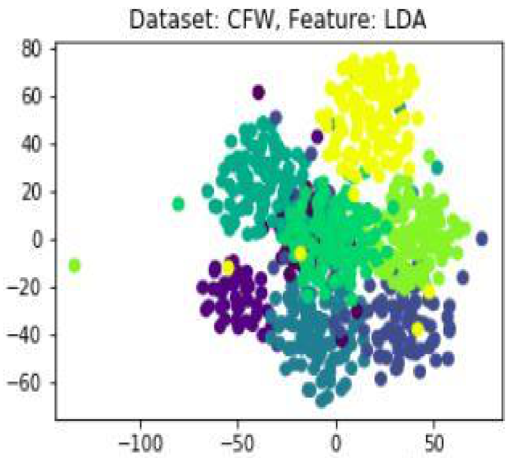

Figure 7: t-SNE plot

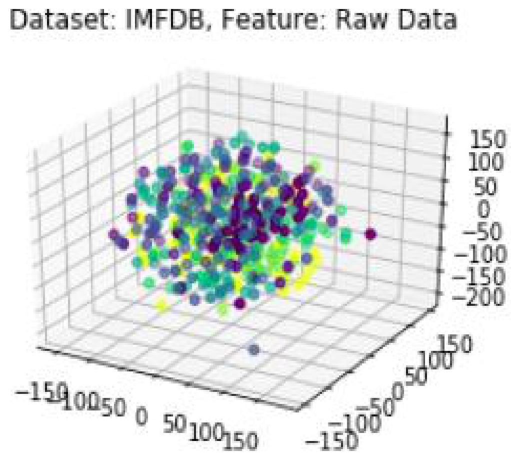

Figure 10: t-SNE plot

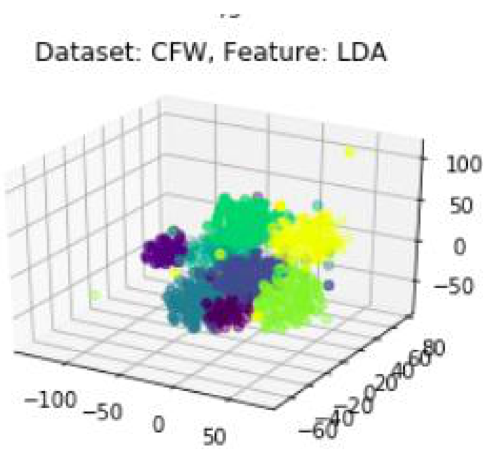

Figure 8: t-SNE plot

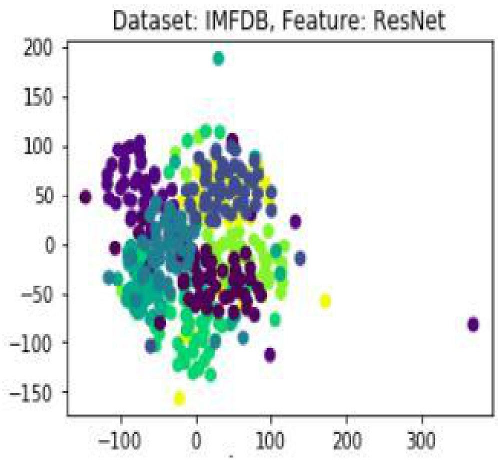

Figure 11: t-SNE plot

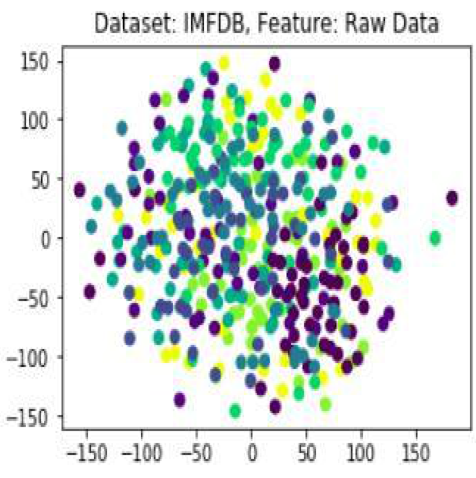

Figure 9: t-SNE plot

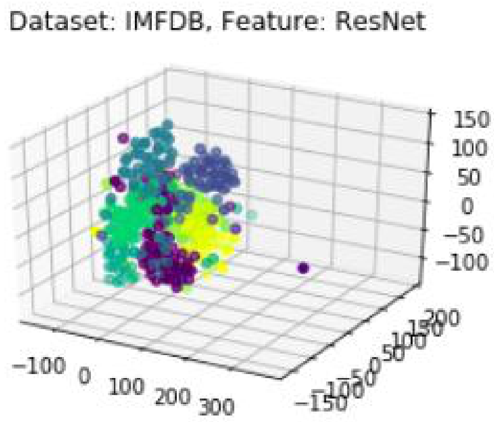

Figure 12: t-SNE plot

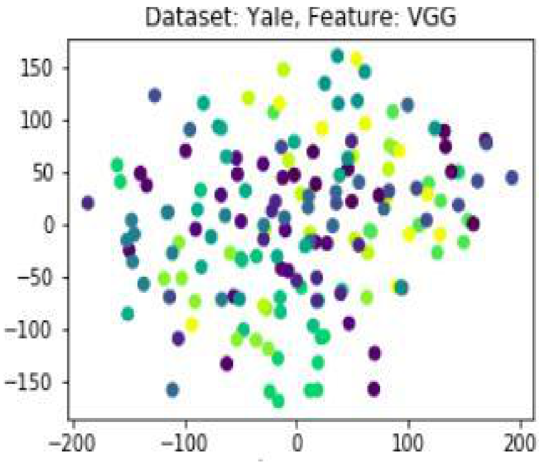

Figure 13: t-SNE plot

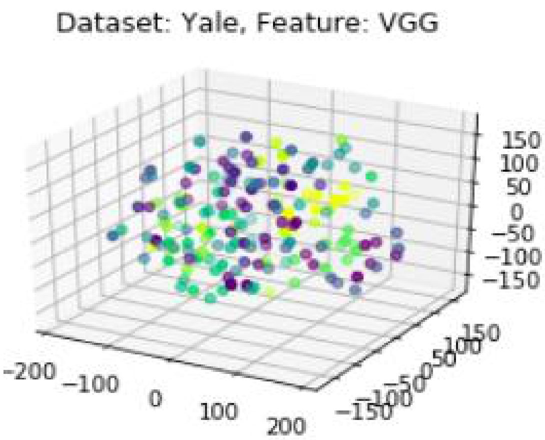

Figure 14: t-SNE plot

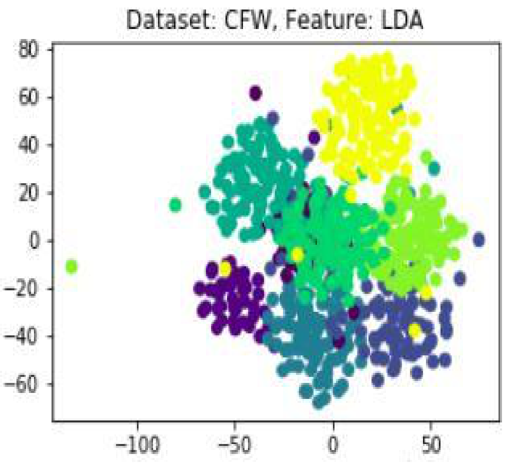

Figure 15: t-SNE plot

4 Verification Problem

1. Formulation of problem using KNN

    - If we consider our system to be a black box:

      Input: An image (or an image feature set) X_i and a class label Y_i
      Output: Boolean True if our classifier in the $FaceVerification()$ class classifies the image X_i as Y_i, and False otherwise.

    - Here, the problem is implemented using the $KNeighborsClassifier$ classifier from $sklearn$ library, and the value of k is chosen to be 5.

    - The classifier is run for all of the 6 feature representations used, and accuracy is tabulated in the *jupyter notebook*.

    - If the class ID predicted is the same as the input class ID, result is True, otherwise it is False. Once the result is determined for all the samples, it is compared with the expected labels, and performance metrics are computed.

2. Analysis of performance

    - It can be seen that for all 3 datasets, ResNet performs the best. The Yale dataset is well categorized for every feature representation, telling that the images are very well spaced in the feature space. For IMDB dataset, LDA and other superior methods perform well. On the contrary, CFW requires VGG, and only ResNet performs exceptionally well, implying that the images are very close together in the feature space.

Fet.	Red. Dim.	Error	Accur.
PCA	3072	0.17	0.83
Kernel PCA	10	0.29	0.71
LDA	10	0.29	0.71
Kernel LDA	10	0.05	0.95
VGG	10	0.05	0.95
ResNet	4096	0.57	0.43

Table 4: Verification on Yale face dataset

Fet.	Red. Dim.	Error	Accur.
PCA	3072	0.49	0.51
Kernel PCA	10	0.51	0.49
LDA	10	0.51	0.49
Kernel LDA	7	0.22	0.78
VGG	7	0.22	0.78
ResNet	4096	0.14	0.86

Table 5: Verification on IMDB face dataset

Fet.	Red. Dim.	Error	Accur.
PCA	3072	0.71	0.29
Kernel PCA	10	0.61	0.39
LDA	10	0.62	0.38
Kernel LDA	7	0.62	0.38
VGG	7	0.62	0.38
ResNet	4096	0.26	0.74

Table 6: Verification on CFW dataset

5 Extension/Application

1. Explanation of problem.
   The problem chosen is that of distinguishing between cartoon and real images from a combined dataset of all the 3 given datasets. Given an image in the kernel LDA feature representation, tell whether the image is that of a cartoon or a real person.

2. Why is the solution useful?
   The solution will be useful if we want to distinguish between the two kinds of images. It can be used to detect if a movie clip (from its screenshots) is animated or not. Another situation may be that we want to automate testing of an animator of human faces for game design and some other purpose, and we want to check if a classifier working well on real vs. cartoon picture is able to tell if the output of the animator is cartoon or real.

3. Analysis (experimental pipeline, splits, evaluation metrics, quantitative results, qualitative results)

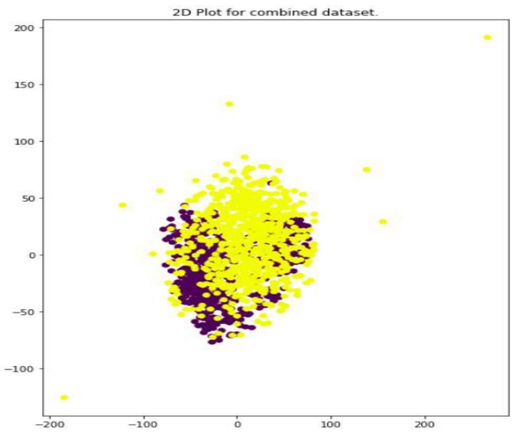

Figure 16: t-SNE plot for combined dataset

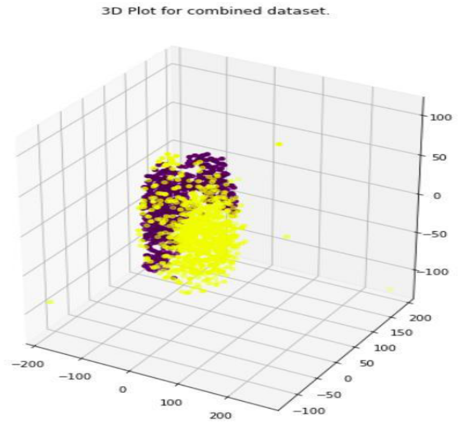

Figure 17: t-SNE 3D plot for combined dataset

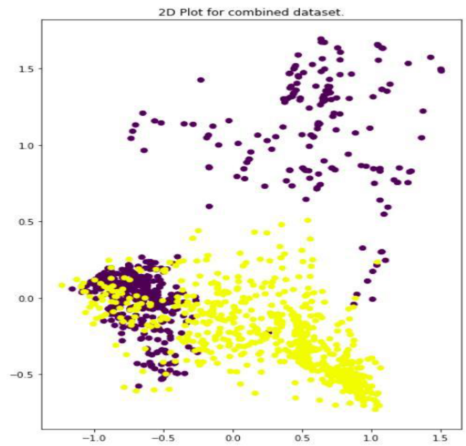

Figure 18: Isomap plot for combined dataset

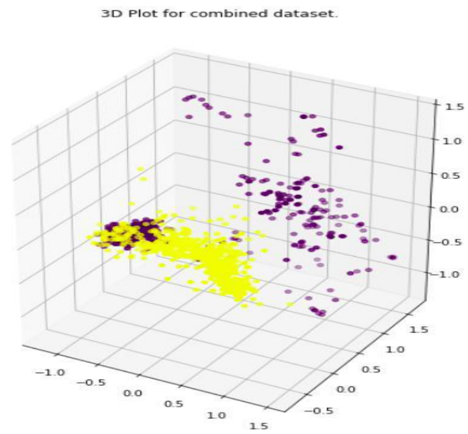

Figure 19: Isomap 3D plot for combined dataset