CSE 475: Statistical Methods in AI	Date: *31/10/2019*
Lecturer: *C. V. Jawahar*	Student: Anurag Jain

Assignment 2: Face Classification/Verification

1 Dimensionality Reduction

1. (a) **What are eigen faces?**

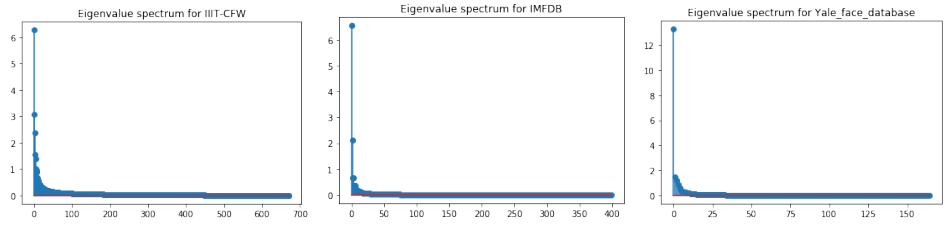
   The significant features / eigenvectors which on projecting the data span the maximum variance. Therefore they are able to capture maximum information from the dataset.

   (b) **How many eigen vec-tors/faces are required to "satisfactorily" reconstruct a person in these three datasets?**

   Ideally, since there are similarities in the data we only need few eigenvectors to reconstruct the data satisfactorily. As shown by the eigenspectrum (plot of the eigenvalues) the last eigenvalues contribute little information. % information retained in first k-eigenvectors = $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \times 100$ % To reconstruct the images satisfactorily we need 230 images for IIIT-CFW dataset, 10 for IMFDB and 21 for Yale Face Dataset.

   (d) **Which person/identity is difficult to represent com-pactly with fewer eigen vectors? Why is that?**

   The images from the cartoon dataset are difficult to represent compactly with few eigen vectors due to lack of similarity and changes in pose. Since the variations lie along many axes, it is difficult to capture the same using only a few eigenvectors. In the IIIT-CFW Dataset Manmohan Singh has a high reconstruction error since he has a turban which others do not have, from the IFMDB Database Amitabh Bachan is difficult to capture because he has a beard whereas other actors do not and in the Yale Dataset Class 7 & 13 has high reconstruction error.



2 Classification

2. (a) **Comparative Study**

   i. *IIIT-CFW* - For this dataset containing many variations, Kernel LDA produces the best features for classification. Along with this, SVM offered the best classification accuracy since one the data is linearly separable (after using Kernel LDA), SVM works quite well. The VGG Features are not very helpful in case of cartoons and perform poorly here. PCA is also not able to properly capture the variations and performs poorly. However, the best option is to chose all the features and classify them using a MLP, and let the network learn which features are important and which are not.

Method	Accuracy	Precision	Recall	F1 Score
All Features + MLP	0.988095	0.992417	0.988366	0.990194
Kernel LDA + SVM	0.964286	0.970833	0.965852	0.967000
LDA + SVM	0.910714	0.911505	0.912389	0.909830
PCA + LR	0.517857	0.514521	0.516662	0.505075
VGG Features + SVM	0.690476	0.698431	0.636413	0.646773
Kernel PCA + DT	0.369048	0.372536	0.371692	0.359440

Table 1: Results for Classification on IIIT-CFW Dataset

ii. *IMFDB* - IMFDB was extracted from Indian Movies, since characters in movies show different variations in expressions and pose, there would be higher variation in illumination, resolution, blur among the samples from the same person. Due to this variation we would need to consult a lot of features before classifying a sample, thus best option would be to choose all features and then use a SVM to classify the data. Due to the variations, PCA does not work well here.

Method	Accuracy	Precision	Recall	F1 Score
All Features + SVM	0.99	0.992188	0.992188	0.991935
LDA + LR	0.96	0.958097	0.964343	0.957463
Kernel LDA + LR	0.96	0.955941	0.964583	0.958830
VGG Features + DT	0.90	0.904167	0.898758	0.899248
PCA + SVM	0.77	0.768945	0.764704	0.764351

Table 2: Results for Classification on IMFDB Dataset

iii. *Yale Face Database* - This dataset is the simplest one with the least variations among the same person but considerable variation between different people. In this case we have many feature-classifier combos offering good performance for e.g. LDA + SVM is a simple classifier but offers good performance here.

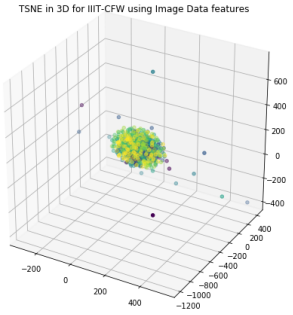
Method	Accuracy	Precision	Recall	F1 Score
LDA + SVM	1.000000	1.000000	1.000000	1.000000
PCA + LR	0.928571	0.936667	0.951111	0.934180
PCA + MLP	0.833333	0.838889	0.840000	0.795661
Kernel PCA + MLP	0.714286	0.751111	0.718889	0.681958

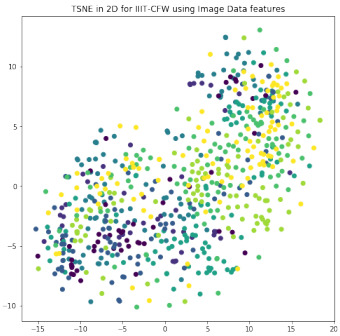
Table 3: Results for Classification on Yale Face Dataset

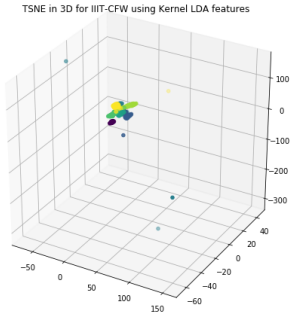
3 Visualization

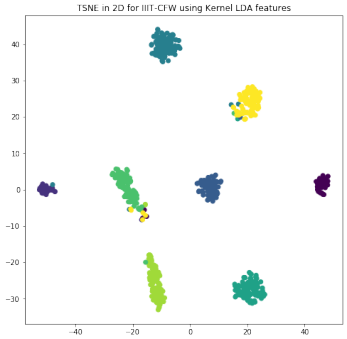
Use t-SNE based visilization of faces? Does it make sense? Do you see similar people coming together? or something else?

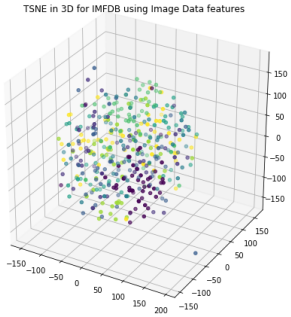
t-SNE is a clustering technique which is used for visualizing the relation between the points in a dataset.

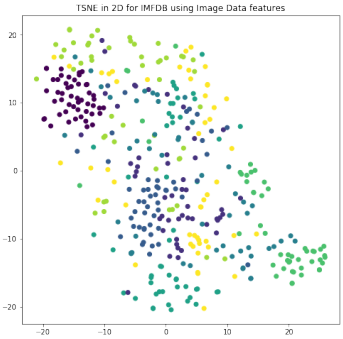

TSNE in 3D for IIIT-CFW using Image Data features

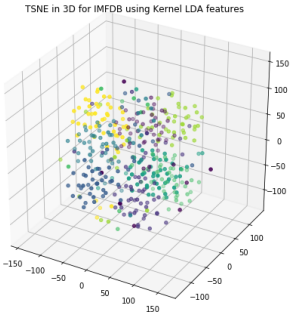

TSNE in 2D for IIIT-CFW using Image Data features

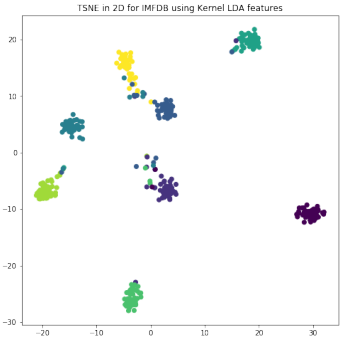

TSNE in 3D for IIIT-CFW using Kernel LDA features

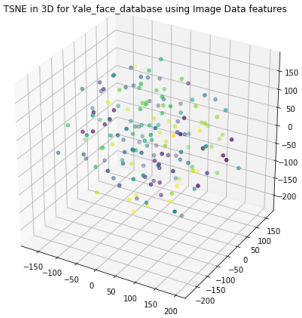

TSNE in 2D for IIIT-CFW using Kernel LDA features

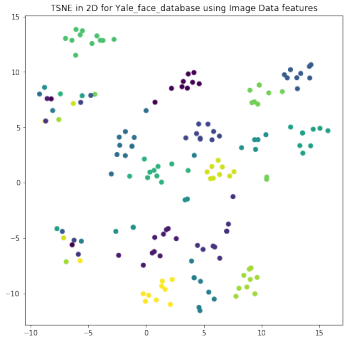

TSNE in 3D for IMFDB using Image Data features

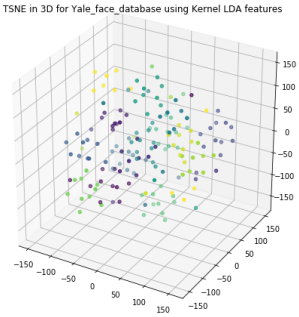

TSNE in 2D for IMFDB using Image Data features

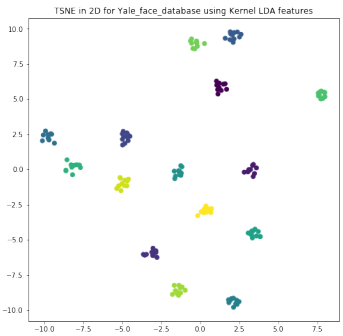

TSNE in 3D for IMFDB using Kernel LDA features

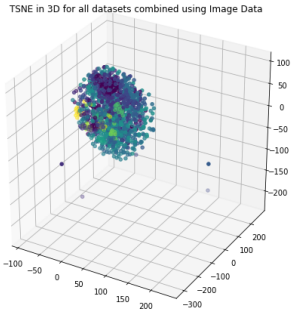

TSNE in 2D for IMFDB using Kernel LDA features

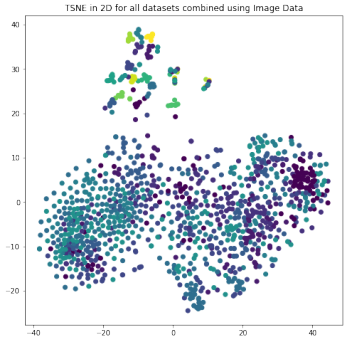

TSNE in 3D for Yale_face_database using Image Data features


TSNE in 2D for Yale_face_database using Image Data features


TSNE in 3D for Yale_face_database using Kernel LDA features


TSNE in 2D for Yale_face_database using Kernel LDA features


TSNE in 3D for all datasets combined using Image Data


TSNE in 2D for all datasets combined using Image Data

One can observe faces from the same class (identifiable by colour) to be clustering together.

4 Face Verification

4. (a) **How do we formulate the problem using KNN?**

   Given a face sample we use k-Nearest Neighbour Algorithm to find out which class the given 'face' belongs to and check if the predicted class matches with the identity class. Output yes if it is a match and no otherwise

   (b) **How do we analyze the performance? suggest the metrics (like accuracy) that is appropriate for this task.**

   Since we want to use it for verification, the most important metric to optimize would be Precision since we don't want the system to yield false positives. (i.e. a sample being able to pass verification when it shouldn't have)

   (c) **Show empirical results with all the representations**

Method	K	Precision	Accuracy	Error
All Features	1	0.988095	0.992424	0.011905
Resnet Features	1	0.970238	0.967671	0.029762
Kernel LDA	3	0.964286	0.970833	0.035714
PCA	5	0.482143	0.507018	0.517857

Table 4: Results for Face Verification on IIIT-CFW Dataset

Method	K	Precision	Accuracy	Error
Resnet Features	1	0.95	0.950000	0.05
LDA	1	0.96	0.957516	0.04
PCA	5	0.63	0.690728	0.37
VGG Features	5	0.92	0.916203	0.08

Table 5: Results for Face Verification on IMFDB Dataset

Method	K	Precision	Accuracy	Error
LDA	1	1.000	1.000	0.000
Resnet Features	1	0.976190	0.983333	0.023810
All Features	1	0.952381	0.961111	0.047619
PCA	5	0.761905	0.855556	0.238095

Table 6: Results for Face Verification on Yale Face Dataset

5 Extension / Application

1. **Briefly explain the problem. Why the problem is not trivial.**

   From the IMFDB dataset and given labels, we try to predict emotions from the faces. This is not a trivial task since the dataset has been extracted from movies where the script requires the action to convey a variety of emotions and sometimes even mix the emotions, it is not easy to label the data let alone predicting the emotion! Another issue is the variety in pose and different expressions of different actors for the same emotion (e.g. Shahrukh Khan's Happy face vs Akshay Kumar's Happy face are quite different)

2. **Why a solution to this may be of some use. Suggest good applications.**

   Once we identify emotions from as particular face, we could take a still from the movie and identify emotions of characters. Then we could take stills from a scene and identify emotions, we could identify the range of emotions in a movie. This identification of emotions can be quite useful to:

   (a) Use as features for movie recommender systems like Netflix since many people like particular range of emotion e.g. Comedy, Romance, Drama etc.

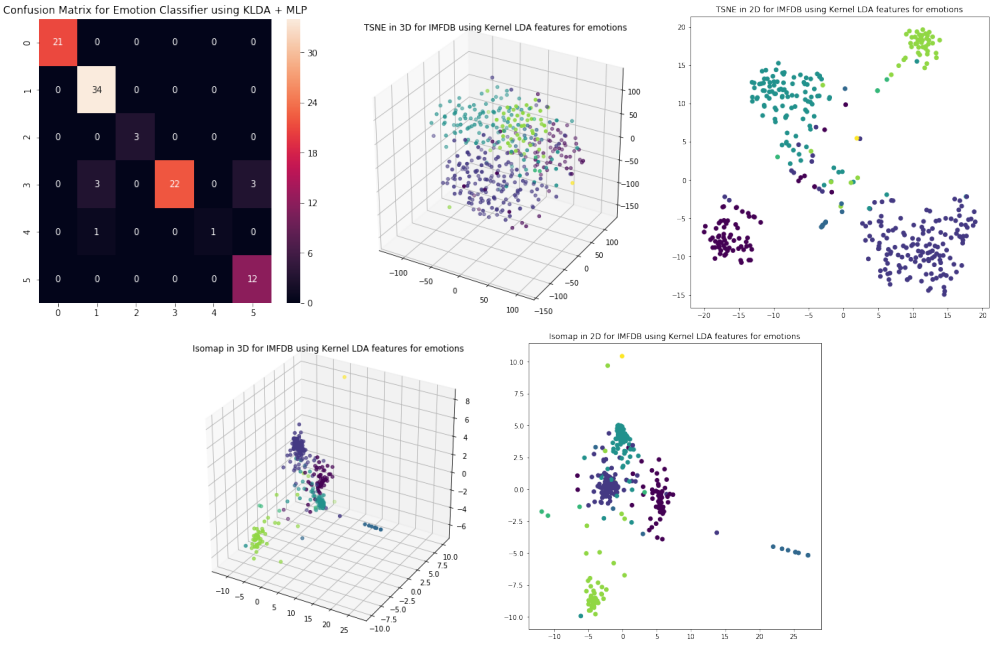
   (b) Identify important scenes in a movie to summarise the movie / produce trailers.

   (c) Suggest background score for visuals using the emotions.

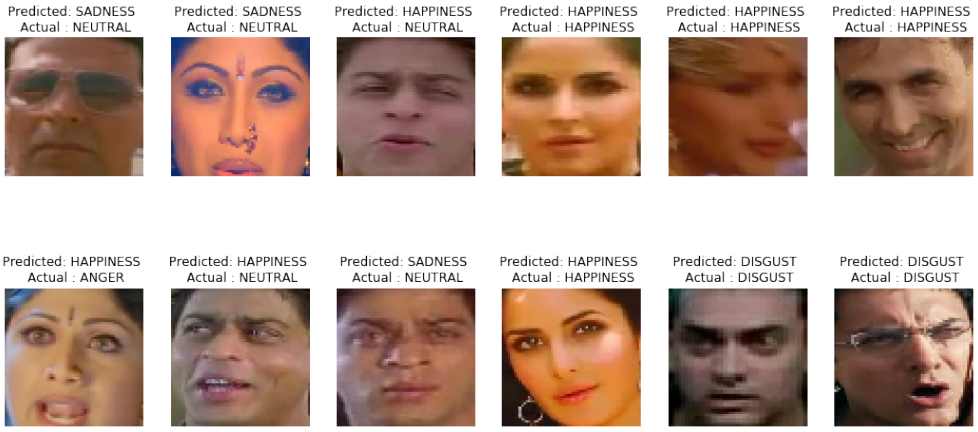
3. **Explain your experimental pipeline, splits, evaluation metrics, quantitative results, qualitative results.**

   First we use LDA to reduce the data to 6 dimensions, then we use MLP with 2 hidden layers with 50 neurons to classify the data. Doing so, we obtain the following results:

   - *Accuracy* : 93 %
   - *Recall* : 88 %
   - *F1 Score* : 89 %
   - *Mean Accuracy from 8-fold Validation* : 91.25 %



4. **Qualitative Results**



Basic expressions have been captured well by the model, e.g. teary eyes in case of sadness or smile in case of happiness. In some cases the expression may deceive the actual conveyed emotion (recall that these are trained actors!)

Misclassifications In general Shah Rukh Khan's eyes are more shiny than others which the model interprets as tears, misclassifying emotion as sadness. In some cases when the actor wears sunglasses, his eyes are not visible which confuses the model. Open mouth may also be interpreted as a smile by the model misclassifying anger as happiness.