SMAI ASSIGNMENT 2

Ishita Vohra

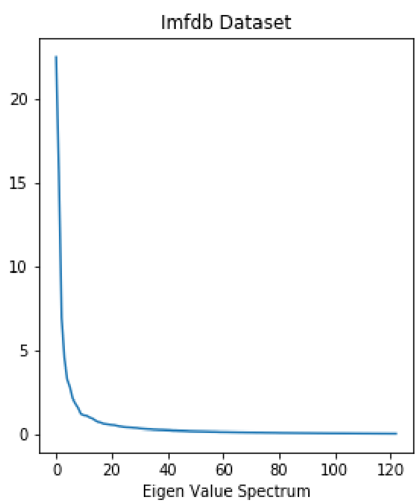
20171054

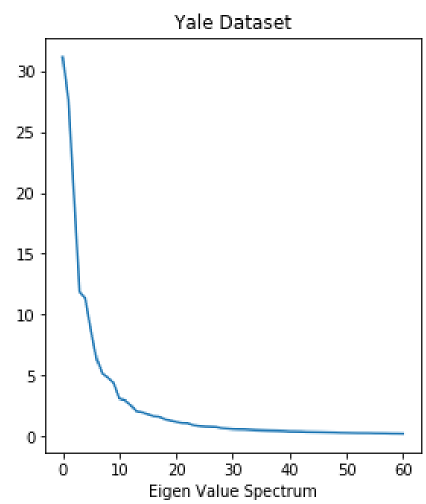
1. I. Eigen Faces

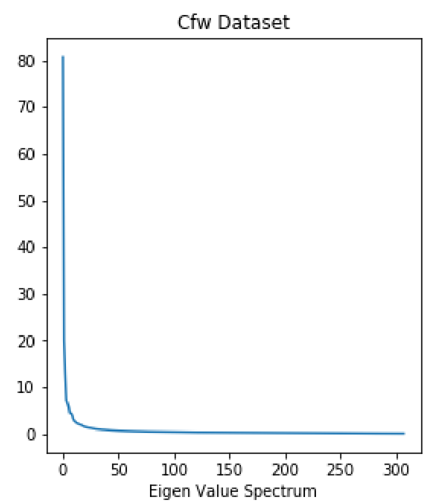
1.1. What are eigen faces?

Eigen faces are orthogonal basis set from which most of the faces can be reconstructed. They are constructed by training on set of real faces. Each of the face highlights a certain type of feature. They are used in facial recognition and characterization applications.

1.2. How many eigen vectors/faces are required to “satisfactorily” reconstruct a person in these three datasets ?







Dataset	No of Eigen Vectors
Imfdb	123
Yale	61
IIIT-Cfw	308

(Sum of k eigenvalues)/(Total number of eigenvalues) should be greater than 0.95. At this k,the eigen values become nearly zero as compared to maximum eigen value, hence the latter ones can be ignored,since they don’t contribute much to our feature space. When we reconstructed image using these eigen vectors, the reconstucted images were almost similar to the actual image.

1.3. Which person/identity is difficult to represent com-pactly with fewer eigen vectors? Why is that? Explain with your empirical observations and intuitive answers

The class for which reconstruction error is maximum is most difficult to identify. CFW can't be respresented using few eigen vectors as it has the maximum reconstruction error out of all the three datasets. Intutively we can say that it has maximum variation in between classes. The eigen faces which can represent one of the class are not sufficent to represent some other class of this dataset

Dataset	Most diff Class to identify	Reconstruction Error
Imfdb	2	0.038
Yale	0	0.054
IIIT-Cfw	1	0.068

2. II. Classifier

Multiple models were tested and only the best ones are reported.VGG+Resnet(concatenation of features) gave the best results for all the datasets. MLP and LR both gave similar results as classifiers. LDA+KPCA(concatenation of models) also gave good results for the YALE dataset. It gave around 85-90percent for IMFDB.

2.1. Imfdb Dataset

Model	Accuracy	Reduced Dim	f1-score	Training Error
VGG+RESNET+MLP	97.50	6144	0.981071	0
RESNET+SVM	95.00	2048	0.951161	0
RESNET+LR	93.75	2048	0.936227	0
VGG+MLP	91.25	4096	0.907153	0
KPCA+SVM	88.75	150	0.880902	0
LDA+KPCA+MLP	87.50	7	0.861182	0
KPCA+MLP	85.00	150	0.849325	0
LDA+KPCA+SVM	83.75	7	0.820142	0
LDA+PCA+LR	82.50	7	0.808082	0

2.2. Yale Dataset

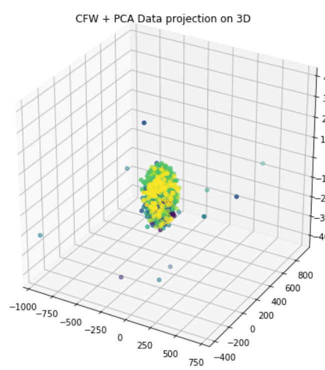
Model	Accuracy	Reduced Dim	f1-score	Training Error
LDA+KPCA+LR	100.000000	14	1.000000	0
KLDA+KPCA+MLP	100.000000	14	1.000000	0
RESNET+LR	100.000000	2048	1.000000	0
PCA+LR	96.969697	75	0.975510	0
RESNET+SVM	96.969697	2048	0.952381	0
KPCA+MLP	93.939394	75	0.894444	0
PCA+SVM	90.909091	75	0.880272	0
KLDA+PCA+MLP	87.878788	14	0.812698	0
KPCA+SVM	84.848485	75	0.760000	0

2.3. CFW Dataset

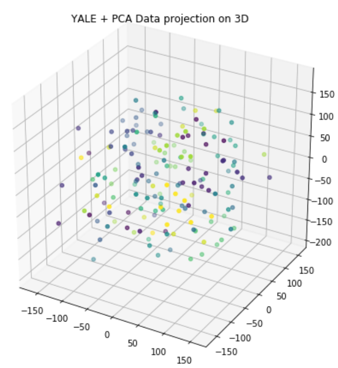
Model	Accuracy	Reduced Dim	f1-score	Training Error
VGG+RESNET+MLP	98.518519	6144	0.988035	0
RESNET+LR	97.777778	2048	0.979284	0
VGG+LR	72.592593	4096	0.687424	0
VGG+SVM	67.407407	4096	0.632285	0
PCA+MLP	60.740741	200	0.609836	0
LDA+PCA+MLP	53.333333	7	0.536714	0
LDA+KPCA+SVM	51.851852	7	0.509253	0
KPCA+MLP	51.111111	200	0.505837	0
KLDA+KPCA+MLP	48.148148	7	0.475322	0

3. III. TSNE

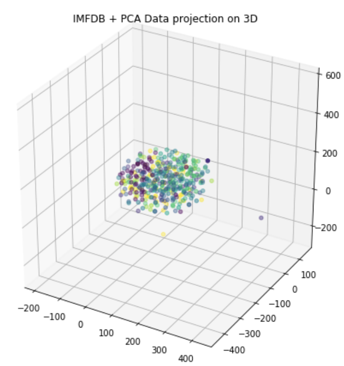
TSNE is technique of dimensionality reduction which is used for the visualisation of high dimensional datasets. It minimizes the divergence between two distributions and is a probablistic method. Before applying TSNE, apply PCA, or any other dimensionality reduction techniques. We will then be able to distinguish between classes. Each class can be seen as separate clusters(eg: Cfw,Pca+Lda). Although in some cases we can't differentiate between classes (eg: Cfw Pca).



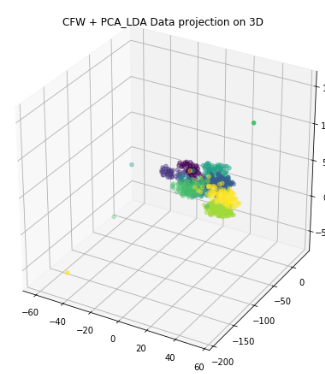
(a) Cfw Pca



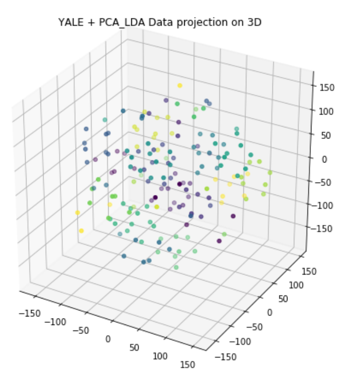
(b) Yale Pca



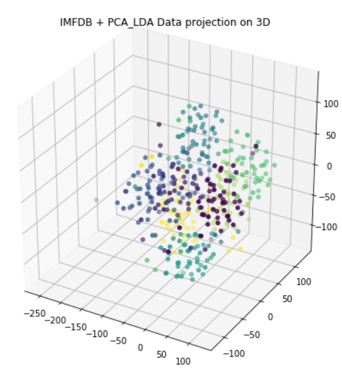
(c) Imfdb Pca



(d) Cfw Pca+Lda



(e) Yale Pca+Lda



(f) Imfdb Pca+Lda

Figura 1: TSNE plots for all datasets

4. Face Verification

4.1. KNN Formulation

- Divide your data into training and testing samples
- Reduce dimensions of your data by applying any feature transformation like PCA,LDA,KLDA,VGG etc.
- Apply KNN algorithm with k being 3,5,9 etc
- Calculate accuracy and test your KNN's perfomance

4.2. Perfomance Analysis

Perfomance can be analyzed using accuracy,computing the confusion matrix or calculating recall. In this method accuracy and precision have been used as a performance metric. We also varied our k and calculated accuracies for various features.

4.3. Empirical Results

The features used were VGG+RESNET(concatenation of features),VGG,RESNET,LDA+PCA,KLDA+PCA,KLDA+ of models). k was varied and values which it could take were 3,5,9. For some features varying k didn't have any major affect on accuracy. It remained the same. But for some features the accuracies changed by significant quantity

4.3.1. Imfdb Dataset

Model	Accuracy	Reduced Dim	Percision	Training Error
VGG+RESNET,5	97.50	6144	0.977183	0.99062
VGG+RESNET,3	96.25	6144	0.961558	0.98750
RESNET,3	95.00	2048	0.954545	0.97187
RESNET,5	93.75	2048	0.938636	0.97187
RESNET,9	93.75	2048	0.938636	0.97187
VGG,9	92.50	4096	0.921729	0.906250
VGG,3	91.25	4096	0.908780	0.934375
VGG,5	88.75	4096	0.879132	0.91562
LDA+KPCA,3	83.75	7	0.837714	0.99062
KLDA+KPCA,3	82.50	7	0.823478	0.99062
LDA+PCA,9	82.50	7	0.838217	0.99375

4.3.2. Yale Dataset

Model	Accuracy	Reduced Dim	Percision	Training Error
LDA+KPCA,3	100.000000	14	1.000000	1.000000
RESNET,5	100.000000	2048	1.000000	0.992424
LDA+KPCA,9	100.000000	14	1.000000	1.000000
RESNET,9	100.000000	2048	1.000000	0.992424
LDA+KPCA,5	96.969697	14	0.964286	1.000000
KLDA+PCA,3	87.878788	14	0.866667	1.000000
KLDA+PCA,5	87.878788	14	0.883333	1.000000
KLDA+PCA,9	87.878788	14	0.883333	1.000000
PCA,3	75.757576	61	0.754762	0.909091
PCA,5	75.757576	61	0.730952	0.871212
PCA,9	69.696970	61	0.671429	0.863636

4.3.3. Cfw Dataset

Model	Accuracy	Reduced Dim	Percision	Training Error
RESNET,3	97.777778	2048	0.975815	0.977654
RESNET,5	97.037037	2048	0.973280	0.981378
RESNET,9	97.037037	2048	0.974739	0.977654
VGG,9	69.629630	4096	0.674027	0.748603
VGG,5	68.148148	4096	0.673355	0.757914
VGG,3	67.407407	4096	0.663551	0.774674
KLDA+KPCA,3	49.629630	7	0.487237	0.979516
KLDA+KPCA,9	49.629630	7	0.501565	0.964618
KLDA+KPCA,5	48.888889	7	0.481471	0.964618
KPCA,9	34.074074	308	0.503436	0.387337
KPCA,3	31.851852	308	0.441911	0.461825
KPCA,5	28.888889	308	0.292517	0.439479

5. Extension/Application: Gender Prediction

5.1. Problem Description

In this problem we are given two datasets Imfdb and Cfw and we need to identify people based on their gender. Solving the above problem is not trivial since lot of data is needed to correctly determine the gender of actors and politicians.

5.2. Problem Applications

- It can be used when we want to perform analysis on the basis of specific gender. It can be used to determine number of male customers vs number of female customers visiting a particular showroom and accordingly sales can be determined/modified.
- Various schemes can be launched after doing analyses on data. If number of one gender(say M) is more than other(say F), various initiatives can be taken to increase the participation of F.

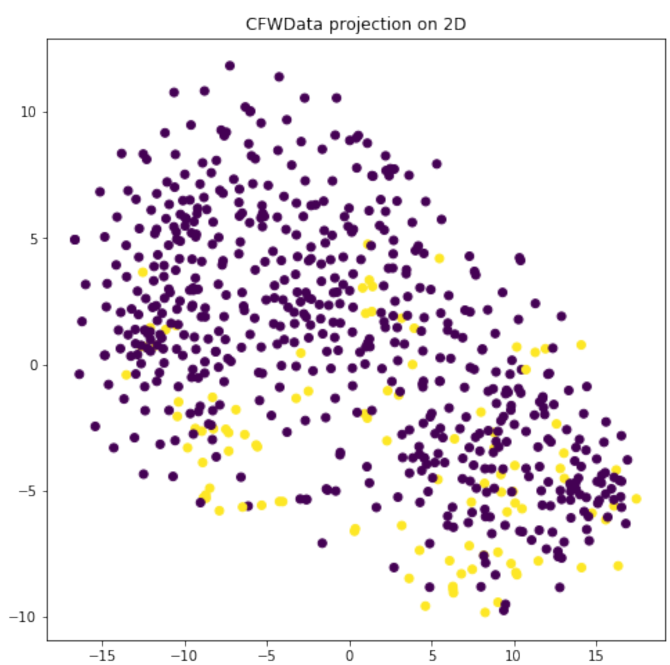
5.3. Problem Solution

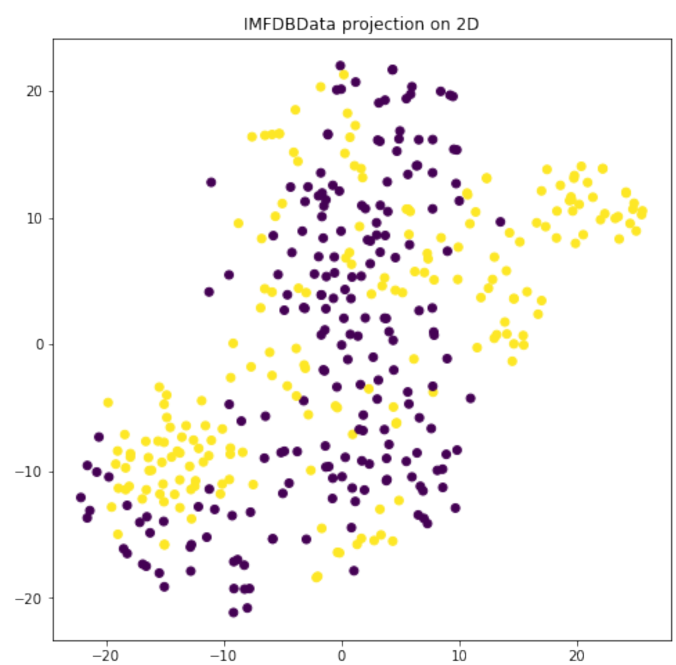
5.3.1. Pipeline,Evaluation Metrics

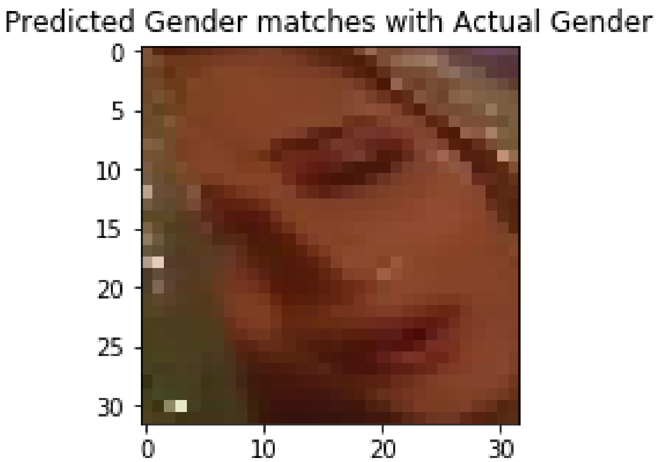
- Firstly we create a set of labels. Male represent Class 0 and Female represents Class 1. We create labels for our dataset on the basis of the above criteria.
- Split the data into testing set and validation set. Apply feature extraction techniques on the dataset. Features considered here were PCA,KPCA,PCA+LDA,PCA+KLA,KPCA+LDA,KPCA+KLDA(concatenation of models),VGG,RESNET,VGG+RESNET(concatenation of features)
- Classifiers used were MLP,SVM,LR.
  1. MLP's architecture = Hidden sizes(1000,1000)
  2. LR's architecture = Solver = lfbgs
  3. SVM's architecture = C = 300, kernel = Linear
  4. Train your features on your models and calculate accuracies for your models. The model with the best accuracy will be chosen. PCA+MLP showed the best accuracy. It showed 91 % for Imfdb whereas 88 % for Cfw.
  5. K-cross validation was applied so as to avoid overfitting. We see that the accuracy and K-cross validation are very similar so there is no leak of data. (K assumed to be 10)

Dataset	K-cross Validation	Accuracy
Imfdb	85.4	91.67
Cfw	87.5	88.28

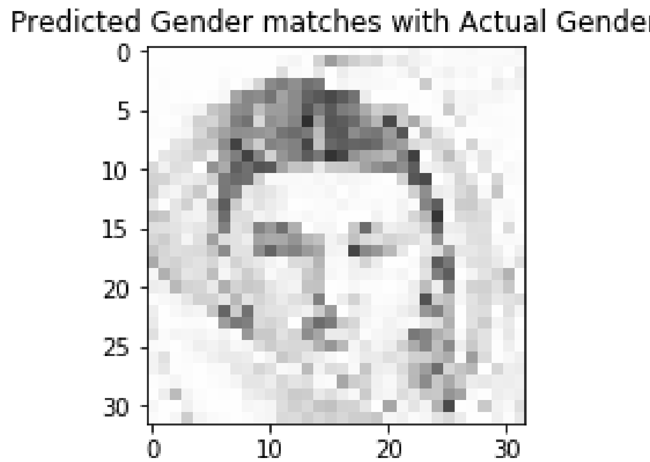
5.3.2. Quantitative(TSNE 2D Plot) and Qualitative Results



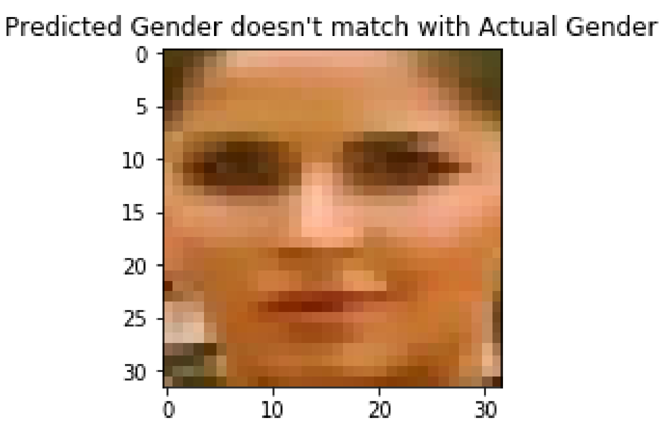




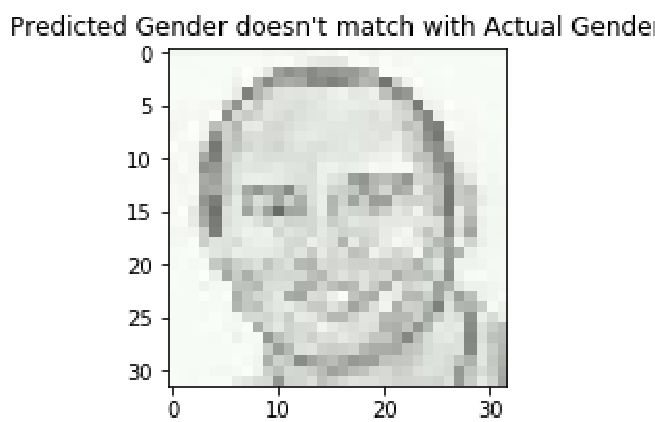
(a) Predicted:Female, Actual:Female



(b) Predicted:Female , Actual:Female



(c) Predicted:Male , Actual:Female



(d) Predicted:Female, Actual:Male

Figura 2: Qualitative Results