Assignment-1 Report

K. Anjali Sreeja
Healthcare and Artificial Intelligence
International Institute of Information Technology, Hyderabad
anjali.kadiyala1999@gmail.com

Abstract

This is the first assignment given in order to understand classification problems with respect to face recognition and verification. Through this assignment, I delved deeper into various concepts and got a better understanding. Link to report: https://www.overleaf.com/read/nqvcrvhbkjmz*. Link to GitHub repository:* https://github.com/Anjali-Kadiyala/HAI_Assignment_1.git.

1. Dataset

Three datasets were used in this project namely Indian Movie Face Database (IMFDB), IIIT Cartoon Face Dataset (IIIT-CFW), and Yale Face Database.

1.1. Indian Movie Face Database

This dataset consists of 50 face images of 8 actors each.

1.2. IIIT Cartoon Face Dataset

This dataset consists of 100 face images of 8 subjects each.

1.3. Yale Face Database

This dataset consists of 11 faces images of 15 subjects each.

2. Question 1

2.1. What are eigen faces?

Eigenfaces are nothing but eigenvectors. When dealing with the "face" problem space, eigenvectors are referred to as eigenfaces. Eigenfaces are basically a result of the concept that not all parts of the face are useful or important for face recognition problems. Just like how humans look at the places with maximum variation to recognise a person, PCA takes in all the images as a whole, keeps only the important parts (with maximum variation) and discards the rest. These important parts are the eigenvectors, i.e., eigenfaces.

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

For IMFDB dataset, the "elbow" in the scree plot (Figure 1) is at 3, and can be seen that after 10-11 principle components, the percentage of variance becomes too little and can be neglected. Therefore, 10-11 eigenvectors/eigenfaces are required to satisfactorily reconstruct a person.

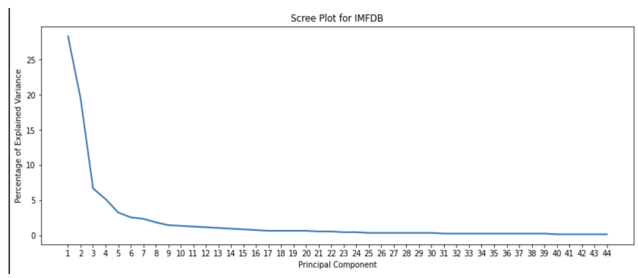

Figure 1. Scree Plot for IMFDB dataset

For IIIT-CFW dataset, the "elbow" in the scree plot is at 6 principle components, after 6, the percentage of variance becomes too little and can be neglected. Therefore, 6 eigenvectors/eigenfaces are required to satisfactorily reconstruct a person.

For this dataset, the "elbow" in the scree plot is at 11, and can be seen that after 24-25 principle components, the percentage of variance becomes too little and can be neglected. Therefore, 24-25 eigenvectors/eigenfaces are required to satisfactorily reconstruct a person.

2.3. Reconstruct the image back for each case

The reconstructed image for IMFDB dataset is shown in Figure 2.

2.4. Which person/identity is difficult to represent compactly with fewer eigen vectors?

In my opinion, the IIIT-CFW dataset is the most difficult to reconstruct with the least number of eigenfaces. I believe it is because the original dataset doesn't have high

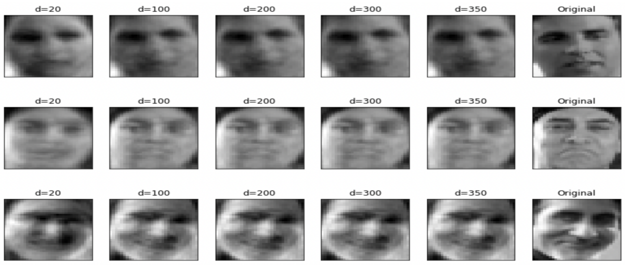

Figure 2. Reconstructed images from IMFDB dataset

quality images. In the IMFDB dataset, the most difficult persons to be reconstructed would be with the lease number of eigenvectors would be classes 1,3, and 5 as can be seen in the t-SNE plot as well. These classes are Kajol, Shilpa Shetty, and Katrina Kaif respectively. I think these would need more eigenvectors as their images are quite similar to one another.

3. Question 2

3.1. Use any classifier and find the classification accuracy

This can be viewed in the github repository.

3.2. Which method works well? Do a comparitive study.

For different dataset, different model-feature combinations worked best. For the IMFDB dataset, best feature-model combination was ResNet50 and logistic regression. For the IIIT-CFW dataset, the best feature combination is LDA and logistic regression. For the Yale face database, best feature-model combination is LDA and multiLayer Perceptron.

4. Question 3: t-SNE

4.1. Does the t-SNE visualization make sense?

With PCA, on all the three datasets, it was difficult to notice similar faces coming together. However, with LDA, on all the datasets, it could be noticed that similar faces came together. In Figure 3, t-SNE with LDA on Yale dataset can be seen. VGG and ResNet50 performed decently too.

5. Question 4: Face Verification

5.1. How do we formulate the problem using KNN

KNN basically classifies a new data point based on how its neighbors' are classified. It does so based on the similarity measure of the earlier stored data points. Pipeline for this problem: Load the dataset (in our case, it is already loaded) Split the dataset into training and testing data. Using KNeighborsClassifier from sklearn, build the model by

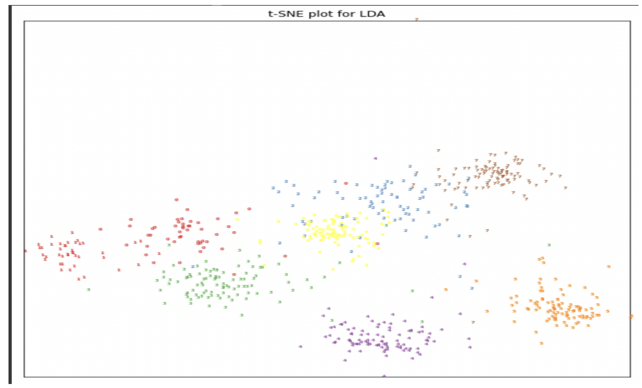

Figure 3. t-SNE visualization of Yale faces database with LDA

fitting the data on the training data. Make a prediction on test data. Calculate the accuracy. There are various metrices that are available to measure the distance like Manhattan distance, Euclidean distance, Chebyshev distance and Minkowski distance. Therefore, I tried to build a model to test out all these. Results are in Figure 4.

	Metrics	Accuracy
0	manhattan	0.833333
1	euclidean	0.785714
2	chebyshev	0.809524
3	minkowski	0.785714

Figure 4. KNN with various metrics on Yale face database

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

For this problem, I believe precision would be the right metrics. Precision is the number of correct predictions made and when it comes to face verification, this is highly important.

5.3. Show empirical results with all the representation

Can be viewed in the github repository.