Assignment 3: Visualization Using Principal Component Analysis

08.01.2022

CMPE 481

—

Sevde SARIKAYA || 2017400081

TABLE OF CONTENTS

Description

We are asked to implement Principal Component Analysis from scratch on various data sets.

Main Steps:

1. Calculation of the mean image, eigenvectors, and eigenvalues
2. Using the test set, reducing the dimensionality of features to two for PCA visualization
3. t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization by scikit-learn
4. Reconstruction of images using the PCA approach with different number of eigenvectors, e.g., i.e., first project to 2,12, 22, 32, ... ,784 dimensions
5. Calculation of explained variance ratio and receiving the least number of eigenvectors required to achieve a good reconstruction.

Please see the notebook to gain insights about the functions. They are explained by supplying necessary mathematical functions.

PCA on MNİST DATA

1. 10 Sample Digit Images per Digit Class:

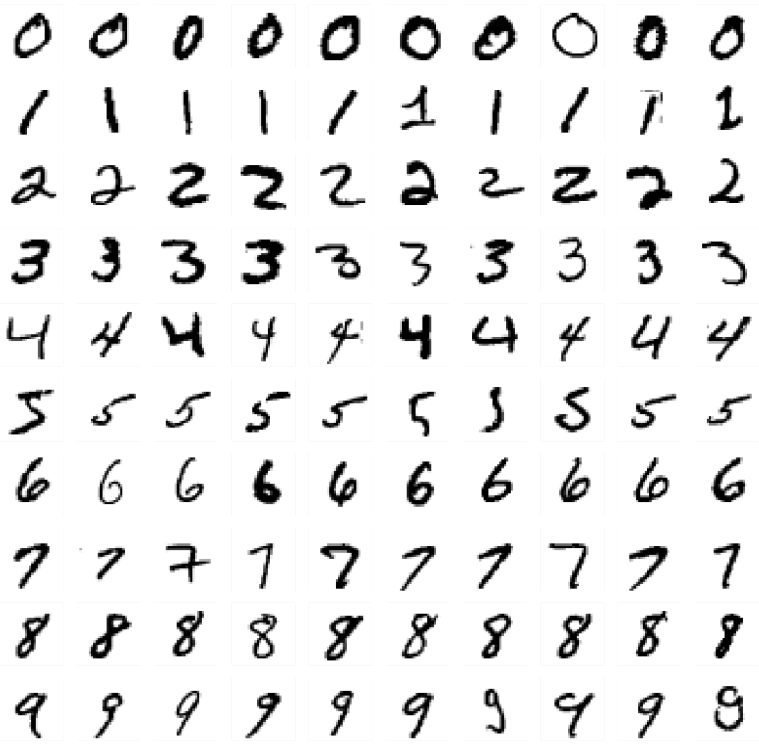


Figure 1. Sample Images

2. Generation of Eigenvectors

2.1 Mean Digit Image

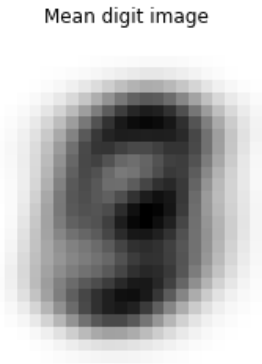


Figure 2: Mean digit image

2.2 Eigenvectors

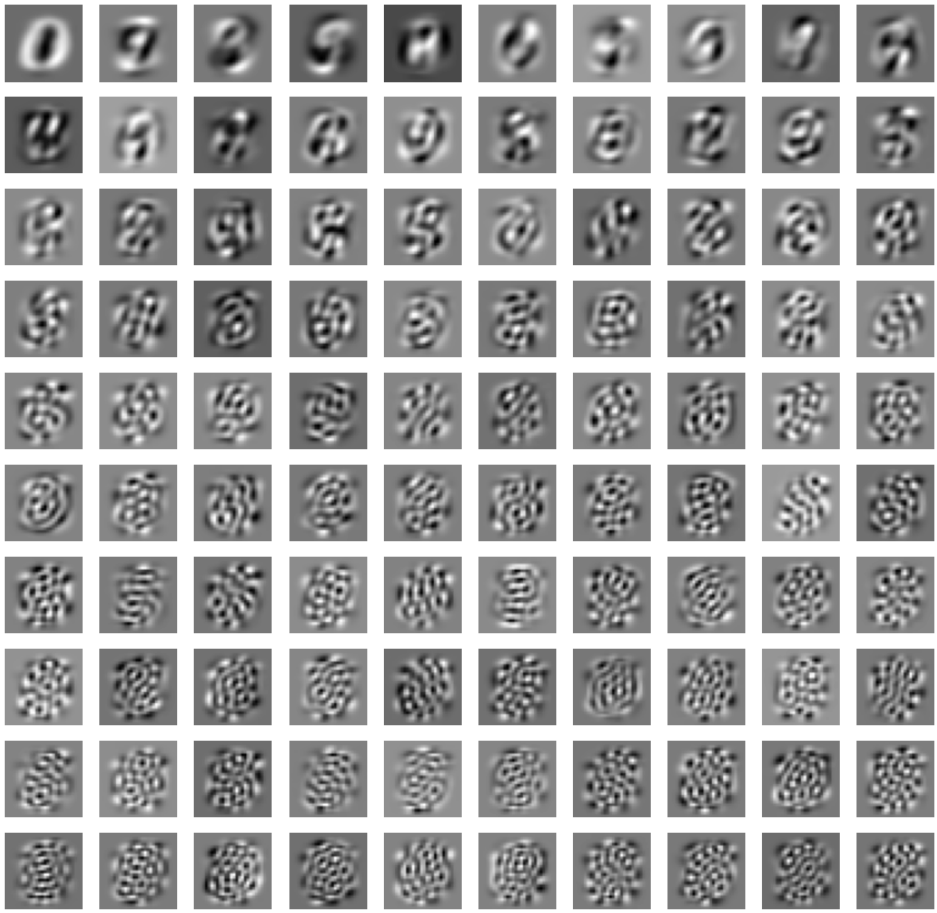


Figure 3: Largest 100 eigenvectors.

2.3 Eigenvalues

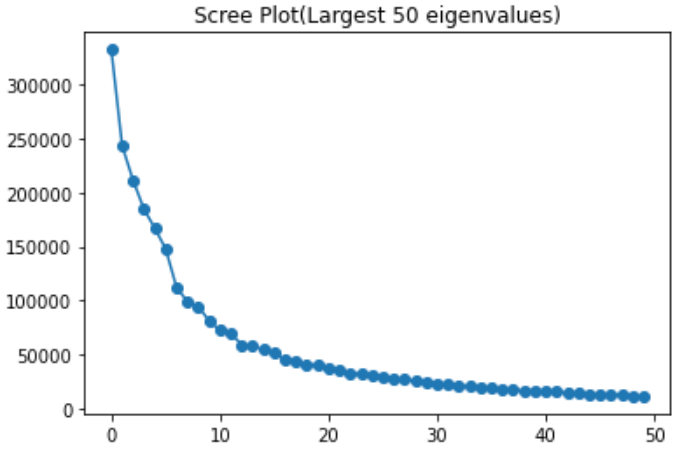


Figure 4: Scree plot (Largest 50 eigenvalues)

3. PCA Visualization

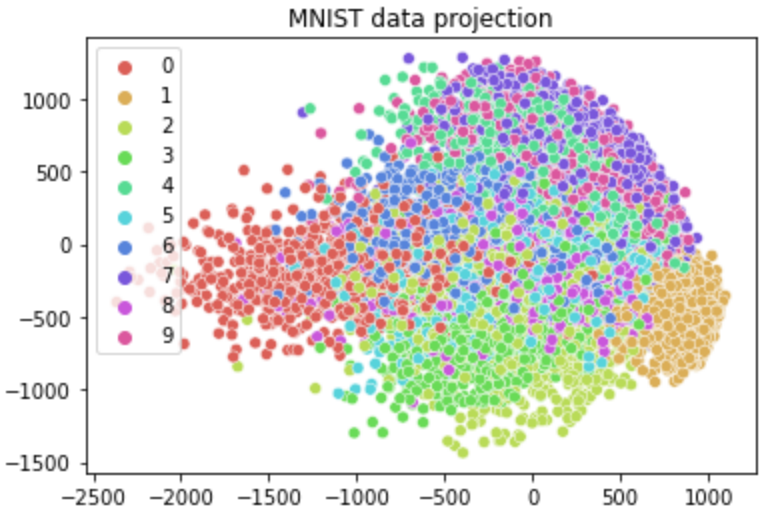


Figure 5: MNIST projection (Largest 50 eigenvalues)

The projection to two looks like clusters. The images labeled with the same digits are clustered very close to each other, which actually proves that the algorithm works fine because the images with the same labels should be similar to each other.

4. t-Distributed Stochastic Neighbor Embedding (t-SNE):

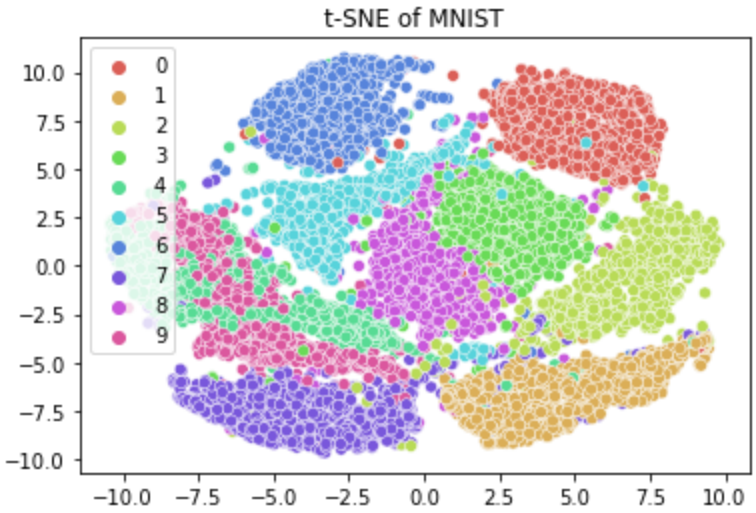


Figure 6: t-SNE of MNIST

5. Fundamentals of t-SNE:

t-Distributed Stochastic Neighbor Embedding (t-SNE) was developed by Laurens van der Maatens and Geoffrey Hinton in 2008. It is an unsupervised technique which is mostly used for visualizing high-dimensional data such as image datasets that we used. As a result of the t-SNE technique, you can get insight about how the data is distributed over the feature space.

For now, it sounds very similar to PCA technique. However, as we know PCA aims to preserve the maximum amount of variance and it is a linear technique. However, this can result in poor visualization of non-linear/manifold data such as 3D objects like cylinders.

How does t-SNE work?

Step 1: Find the pairwise similarities

The similarities are proportional to probabilities p_ij which is computed by:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

set p_i|i = 0

- σ_i is the variance of the Gaussian that is centered on datapoint x_i
- x_i is where a Gaussian centered on

The neighbors of x_i are chosen by the proportion of this probability.

To get final similarities, symmetrize probabilities by getting the average:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

Step 2: Based on the pairwise similarities in the high dimensional space, map the data to a low dimensional space.

Since t-SNE is used for visualization, we map the data to 2 or 3 dimensional space.

The similarities q_ij between two points in the map y_i and y_j in the low dimensional space should be measured by using a very similar approach below:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

Heavy-tailed Student t-Distribution with one degree of freedom is used to calculate the similarities in the low dimensional space. It helps us to place dissimilar points far away from each other in the map.

Step 3: Use gradient descent based on Kullback-Leibler divergence (also called relative entropy) to minimize the difference between p_ij (similarity in high dimensional space) and q_ij (similarity in low dimensional space)

$$\sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

Parameters:

n_components: Dimension of the new space (Mostly 2 or 3 used)

Perplexity: Related to the number of nearest neighbors used in calculation of probabilities.

n_iter: maximum number of iterations

t-SNE	PCA
Non-deterministic: It doesn't give the same results in each run	Deterministic
Handles non-linear data	Handles only linear data
Since it doesn't learn a function from the data, new points can't be used or projected to the low dimensional space.	PCA has eigenvectors. New points can be projected to the map by using these vectors.
It can handle the outliers.	It is affected by the outliers since it maximizes the variance.

6. Reconstruction of Images Using PCA with Different Number of Eigenvectors :

6.1 MNIST DATA

6.1.1 Reconstruction with Different Dimensions

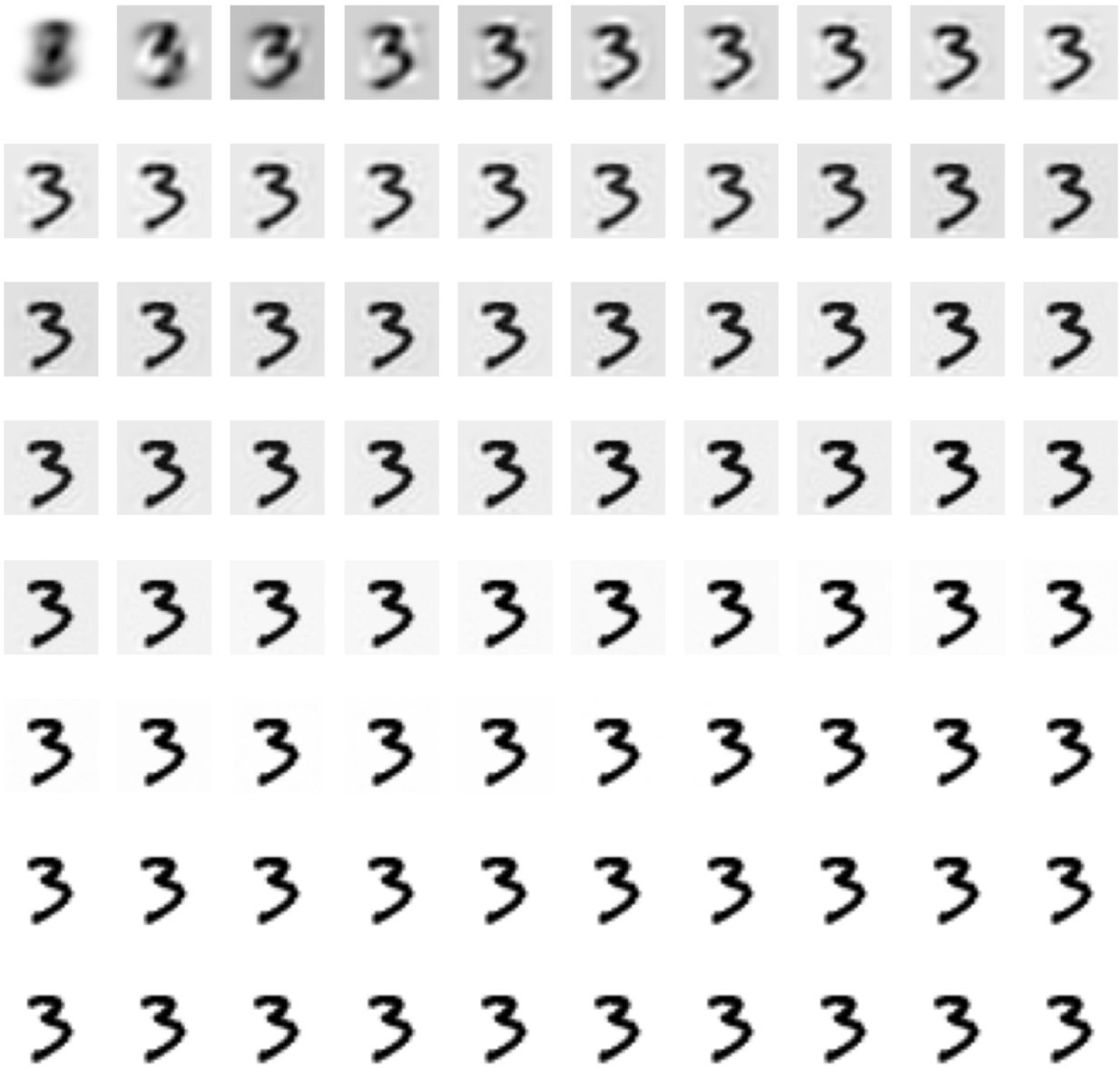


Figure 7: Reconstruction of MNIST

6.1.2 Explained Variance Ratio

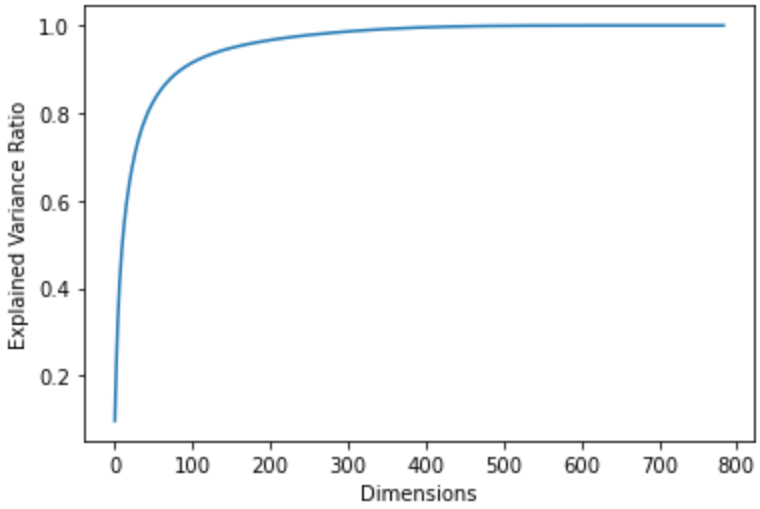


Figure 8: MNIST - Explained Variance Ratio

6.1.3 Reconstruction of Image by Using Least Number of Eigenvectors

Figure 9: Reconstruction of 3

It is possible to obtain a good reconstruction with a dimension of 144. The corresponding explained variance ratio is 94,6% The image above is reconstructed by this value.

6.2 FASHION DATA

6.2.1 Reconstruction with Different Dimensions

Figure 10: Fashion Data Reconstruction

6.2.2 Explained Variance Ratio

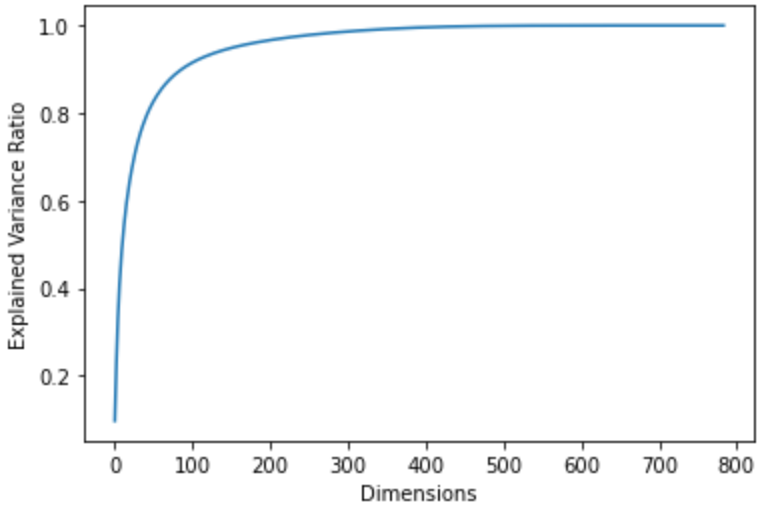


Figure 11: Fashion Variance Ratio

6.2.3 Reconstruction of Image by Using Least Number of Eigenvectors

Figure 12: Fashion Top Reconstruction

It is possible to obtain a good reconstruction with a dimension of 144. The corresponding explained variance ratio is 94,6% The image above is reconstructed by this value.

PCA on HUMAN FACES

1. 10 Sample Digit Images per 10 Human Face Class:

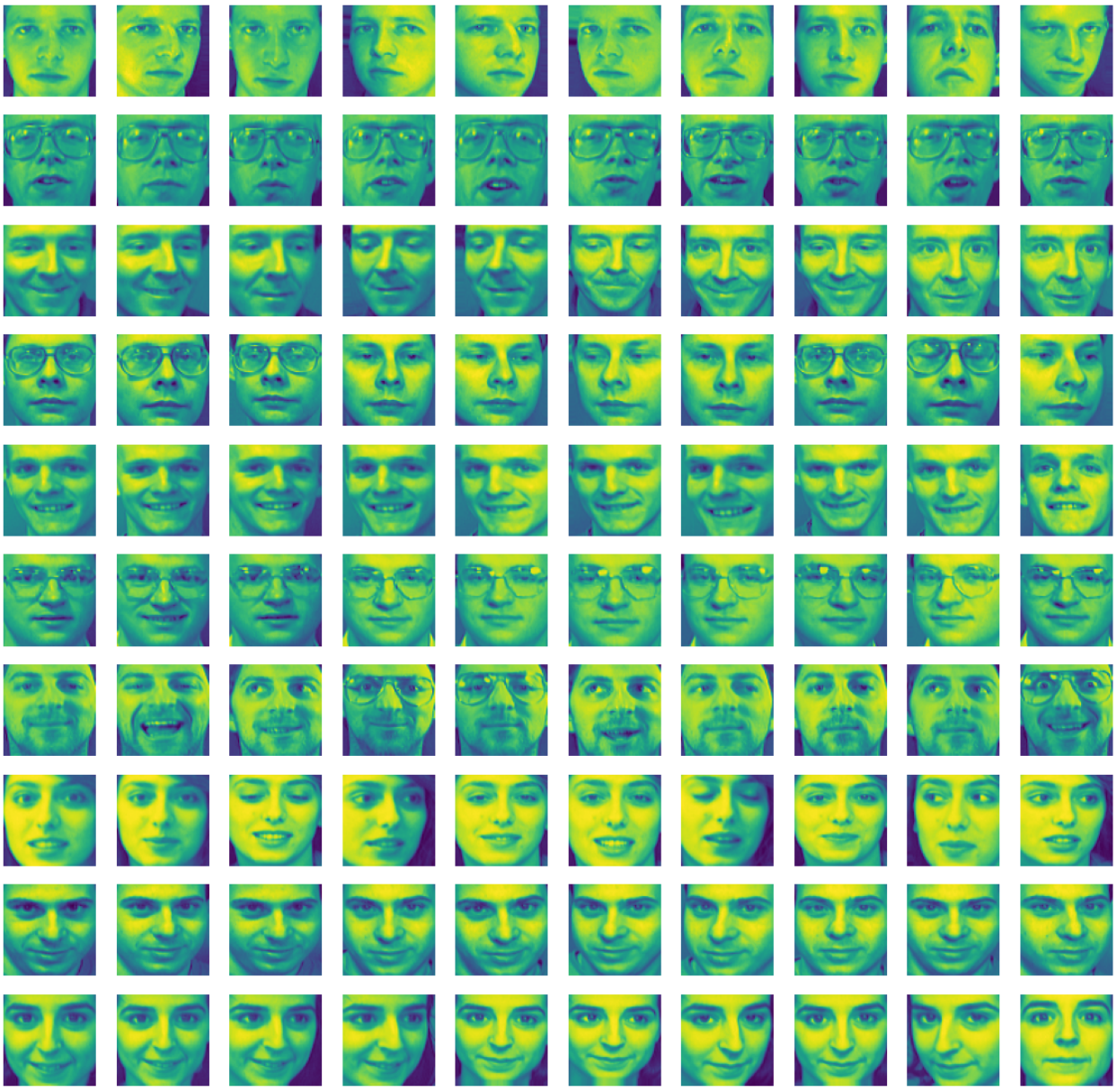


Figure 13: Human Face - Plot

2. Generation of Eigenvectors:

2.1 Mean Human Face Image:

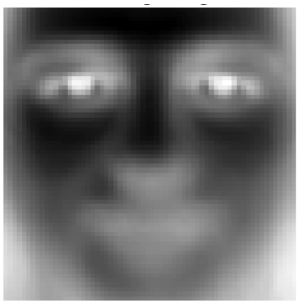


Figure 14: Human Face - Mean

2.2 Eigenvectors:

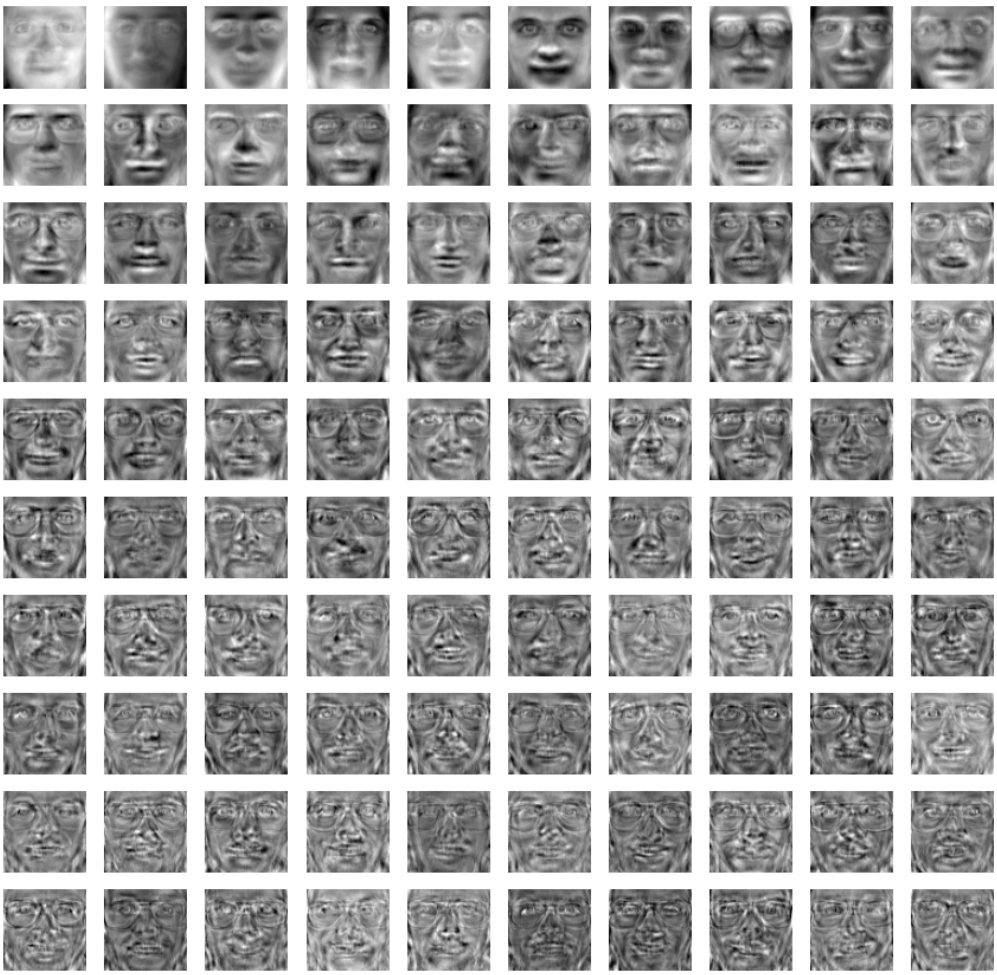


Figure 15: Human Face - Top 100 Eigenvectors

2.3 Eigenvalues:

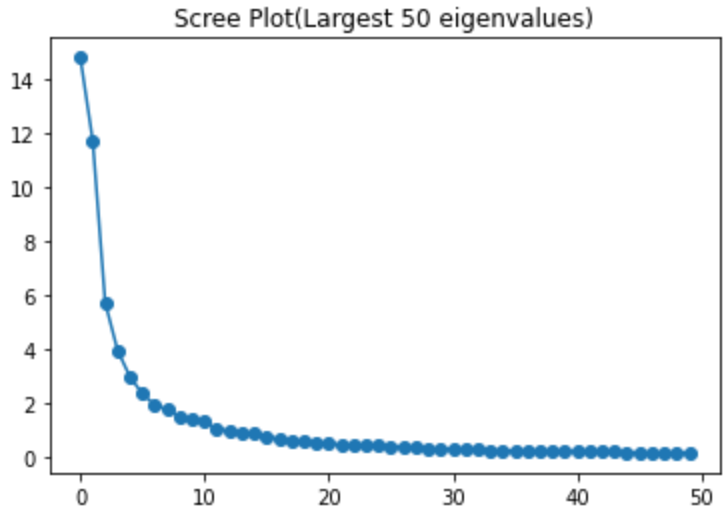


Figure 16: Human Face - Largest 50 Eigenvalues

3. PCA Visualization:

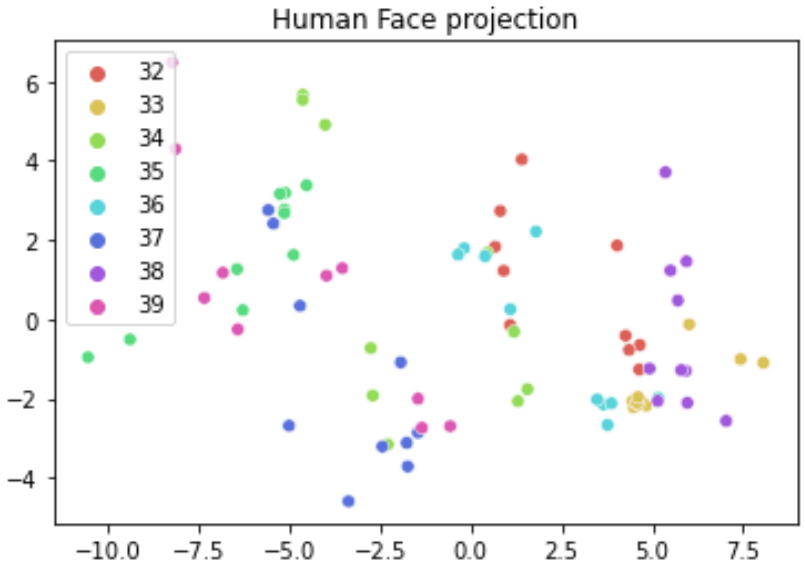


Figure 17: Human Face - Projection to two

4. t-Distributed Stochastic Neighbor Embedding (t-SNE):

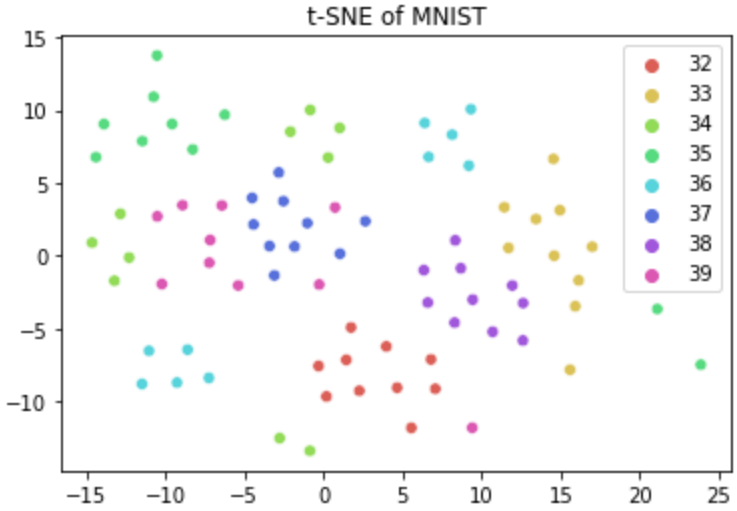
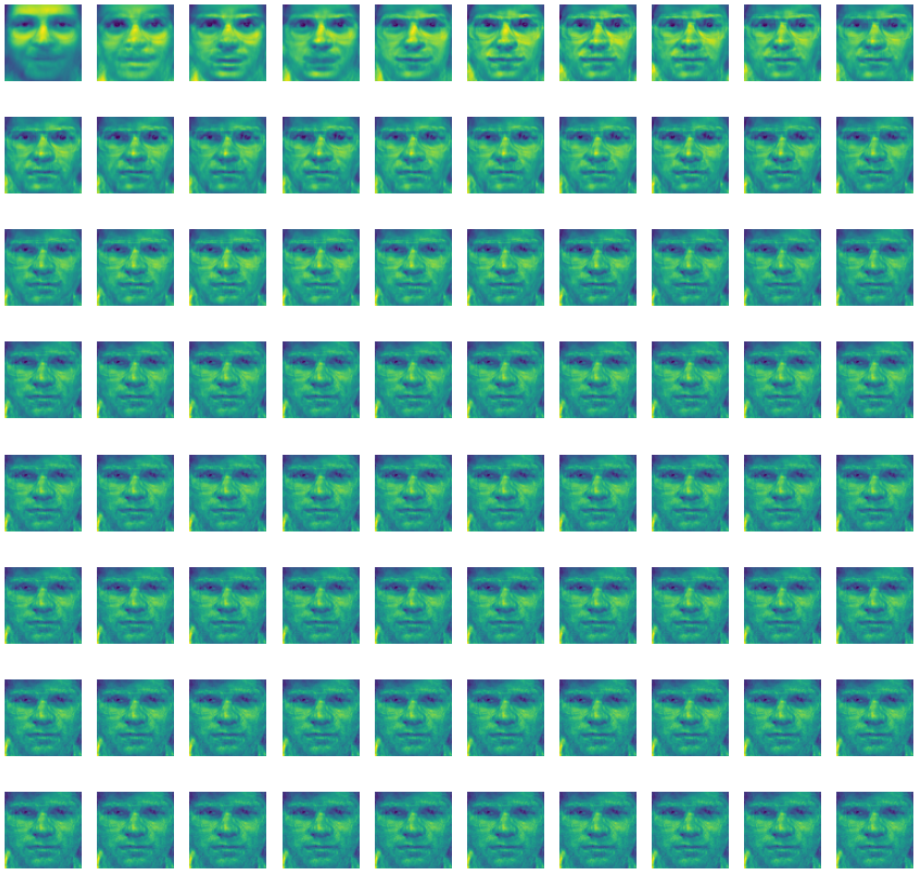


Figure 18: Human Face - t-SNE

5. Reconstruction of Images Using PCA with Different Number of Eigenvectors :

5.1. Reconstruction with Different Dimensions



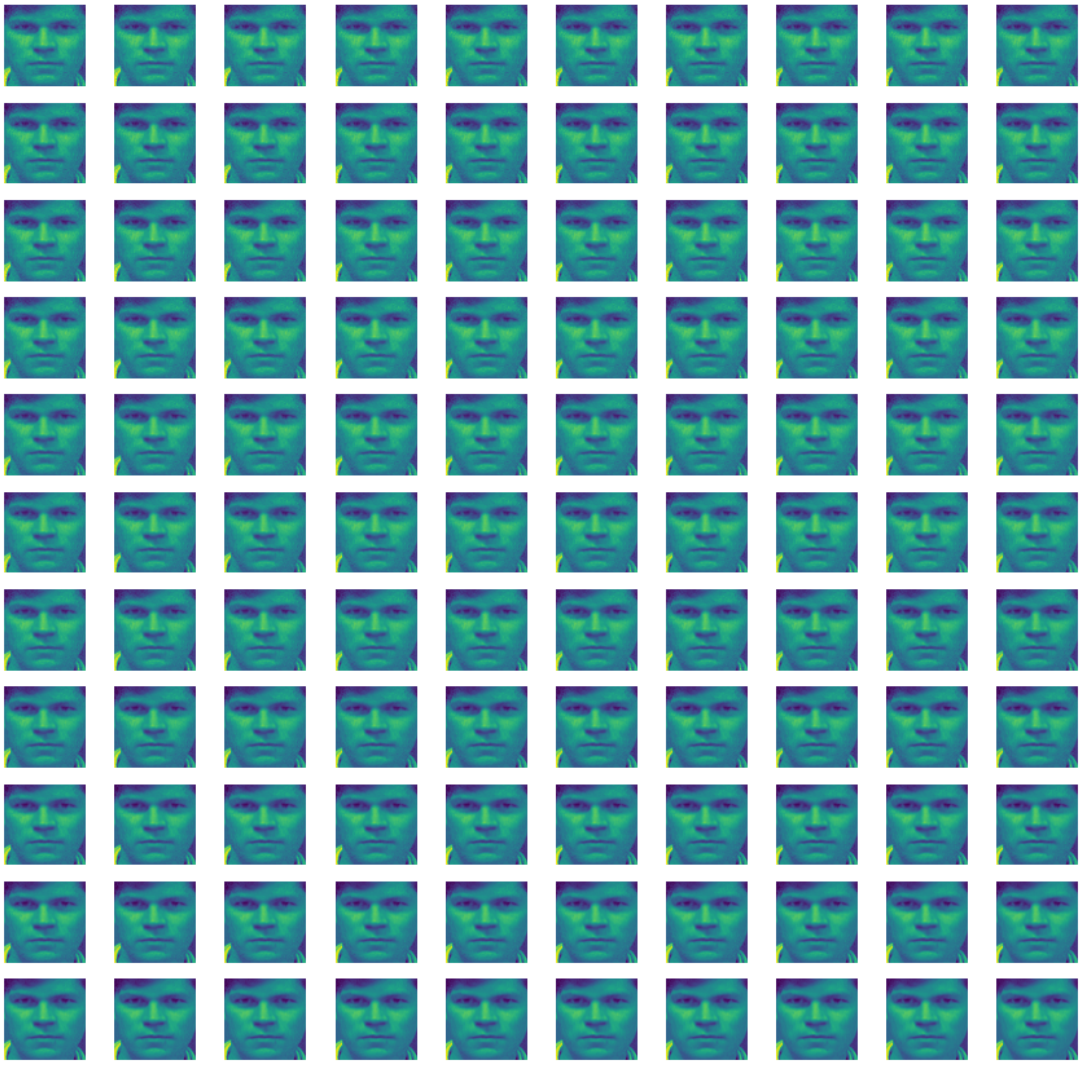


Figure 19: Human Face - Reconstruction

Since the image is 64X64, I split the reconstruction into 2 parts.

First part is from the 2 dimensions to 784 dimensions.

Second part is from the 3000 dimensions to 4096 dimensions.

5.2 Explained Variance Ratio

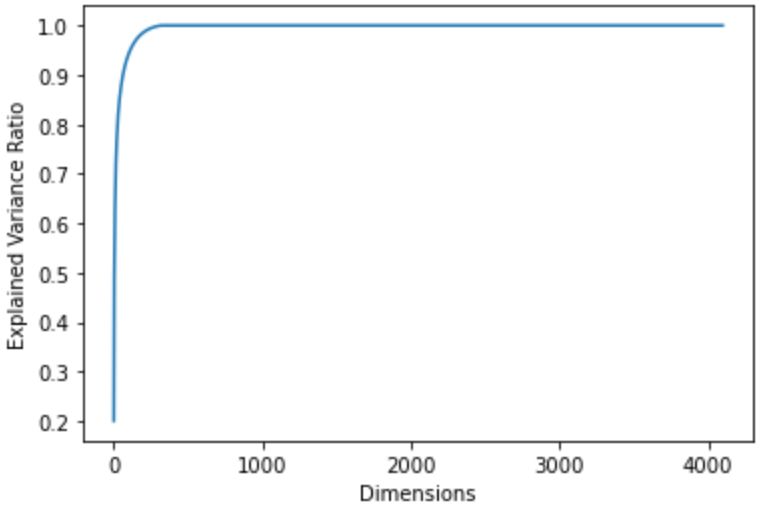


Figure 10: Human Face - Explained Variance Ratio

5.3 Reconstruction of Image by Using Least Number of Eigenvectors

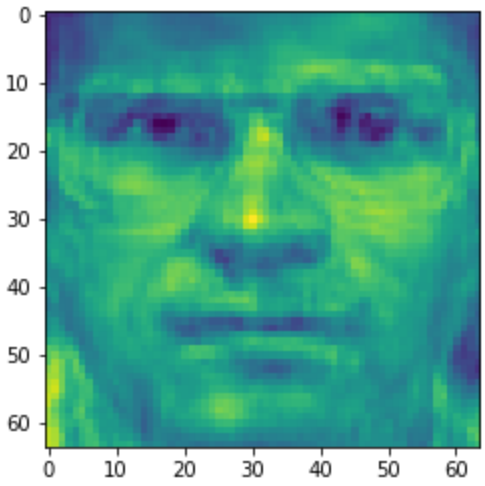


Figure 10: Human Face Reconstruction with the elbow value

It is possible to obtain a nice reconstruction with a dimension of 133. The corresponding explained variance ratio is 96,35%.

The image above is reconstructed by this value.

Resources

- PCA vs t-SNE: which one should you use for visualizationhttps://medium.com › pca-vs-...
- Are there cases where PCA is more suitable than t-SNE?https://stats.stackexchange.com › ...
- t-distributed stochastic neighbor embedding - Wikipediahttps://en.wikipedia.org › wiki
- sklearn.manifold.TSNE — scikit-learn 1.0.2 documentationhttp://scikit-learn.org › modules
- An Introduction to t-SNE with Python Example - Towards Data ...https://towardsdatascience.com › ...
- T-distributed Stochastic Neighbor Embedding(t-SNE)https://towardsdatascience.com › ...