Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) to Solve the Problem of Facial Images

Nishanta Baral

Montclair State University

April, 2022

Abstract

In this work, we apply the concept of Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) to the problem of facial images. We explore average face, eigenfaces, rank-r SVD basis approximation on test images, and eigenfaces approximation on an image of an animal and an inanimate object. Similarly, we explore the eigenface approximation on different lighting conditions, and build a machine learning algorithm that can recognize an individual face.

Contents

1. Introduction

We consider a famous set of data known as the Yale Face Database, which contains 165 grayscale images of 15 individuals. There are 11 images for each individual in which they make different facial expressions (happy, sad, surprised, etc.) or have different elements (glasses, lighting, etc.). We also consider the Extended Yale Face Database which consists of cropped images of 38 individuals who perform 9 different poses under 64 different lighting conditions. In a large dataset, PCA is an extremely handy tool as it is a dimensionality-reduction method that reduces the dimensionality of large data sets, by transforming a large set of variables into a smaller one. Similarly, any image can be stored in a a matrix form and decomposed using SVD. Then, we can reconctruct the original image using the first r singular values which is also known as rank-r approximation. We use PCA and SVD to analyse the data set provided.

2. Theoretical Background

Singular value decomposition (SVD) of an m x n matrix, A, is the factorization of M into three matrices such that $A = U\Sigma V^T$ such that U is an m x n matrix of the orthonormal eigenvectors of AA^T, V is the transpose of a n x n matrix which contains the orthonormal eigenvectors of A^TA, and Σ is an n x n diagonal matrix which contains the the singular values which are the square roots of the eigenvalues of A^TA or AA^t.

PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. In summary, PCA seems to suggest that that all we are doing is expanding the solution in an orthonormal basis which diagonalizes the underlying system.

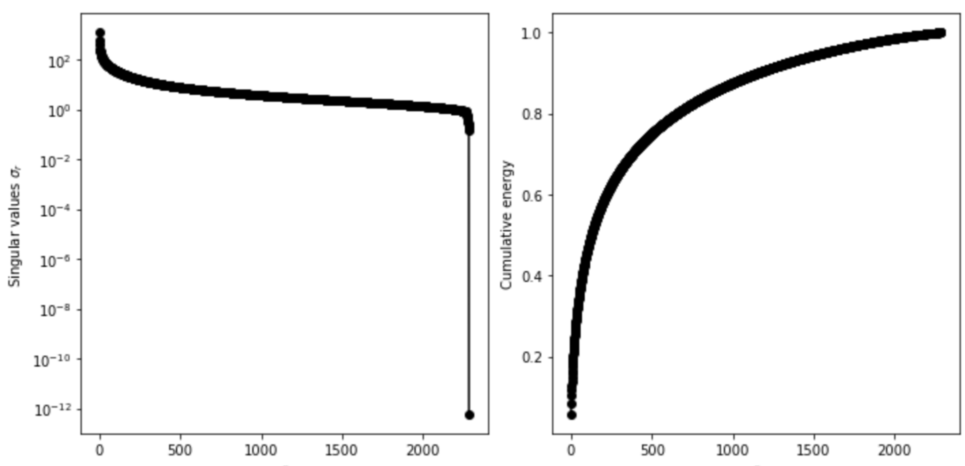

Figure 1. Figure on the left shows the log of the singular values and the number of singular values. The second figure shows the cumulative sum of the singular values/cumulative sum of the first r singular values which tells us the fraction of the energy captured by first r singular values. We also see this on the right graph, approximately 800 modes capture 80% of the image.

3. Numerical Implementation

We begin by creating two figures, first of which shows a single image for each of the 36 individuals and the latter showing all 64 images for a single individual. We then import images of 36 individuals under

64 different lighting conditions and convert them into a single dataframe. We then de-mean our data and reshape it into a column vector so that we can find the average face. This was then subtracted from each column vector. The column vectors were stacked to get matrix B. We perform SVD on the matrix B to get $U\Sigma V^T$ decomposition and take r approximation on the matrix U to get different first r eigenface. We use the following equation, $\tilde{x}_{test} = UU^T * x_{test}$, where UU^T is the covariance matrix, for our projection and explored how different ranks, r, approximate the test image. We used rank r projection on two individuals we held back (37th and 38th people from the dataset), an animal, and an inanimate object. We then added images from uncropped database to our dataframe to approximate image on a different lighting condition. We observe that starting at $r = 400$, we start to see the resemblance on our projection, and get a decent approximation on $r = 1600$. This is consistent with Cumulative energy graph from singular values which shows that about 80% of image is captured in about 900 modes.

4. Computational Results

Figure 2. Figure in the left shows an image grid of 36 individuals under the same lighting conditions, and the figure on the right shows an image grid of a single individual under different lighting conditions.

Figure 3. After each image of 36 individuals under 64 different lighting conditions was reshaped into a column vector, we de-meaned our data and computed our average face.

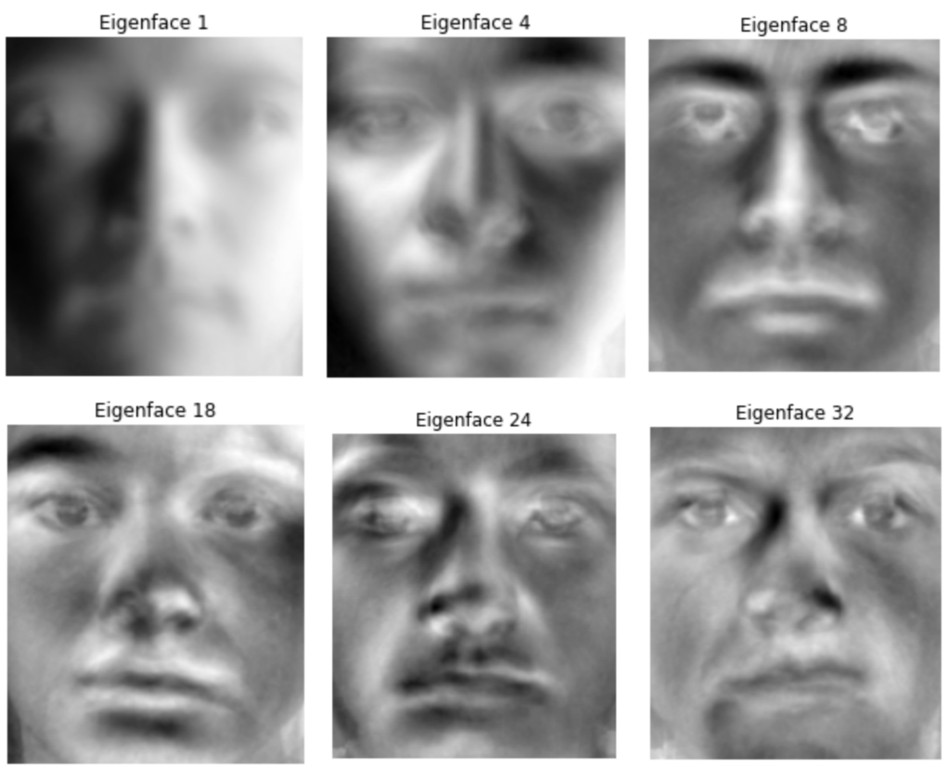

Figure 4. Different rank r eigenfaces.

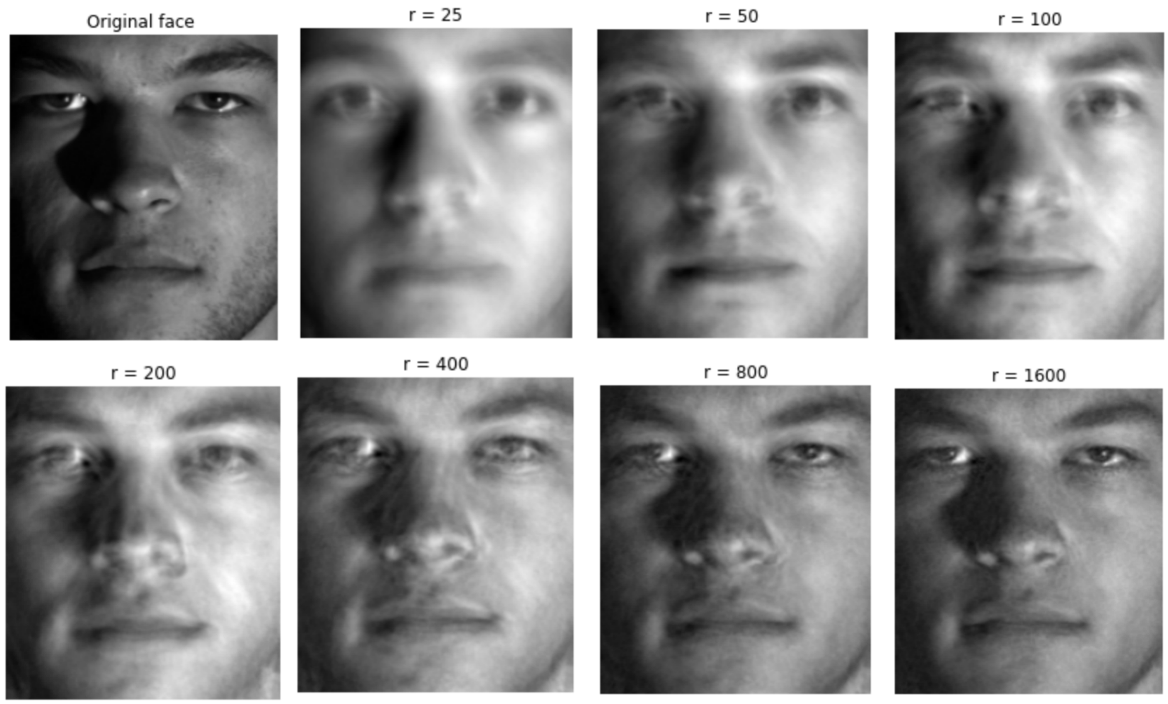

Figure 5. Eigenface approximation for 37th individual. Original image is provided for reference in the top left corner. r=800 and r=1600 provide the best approximations.

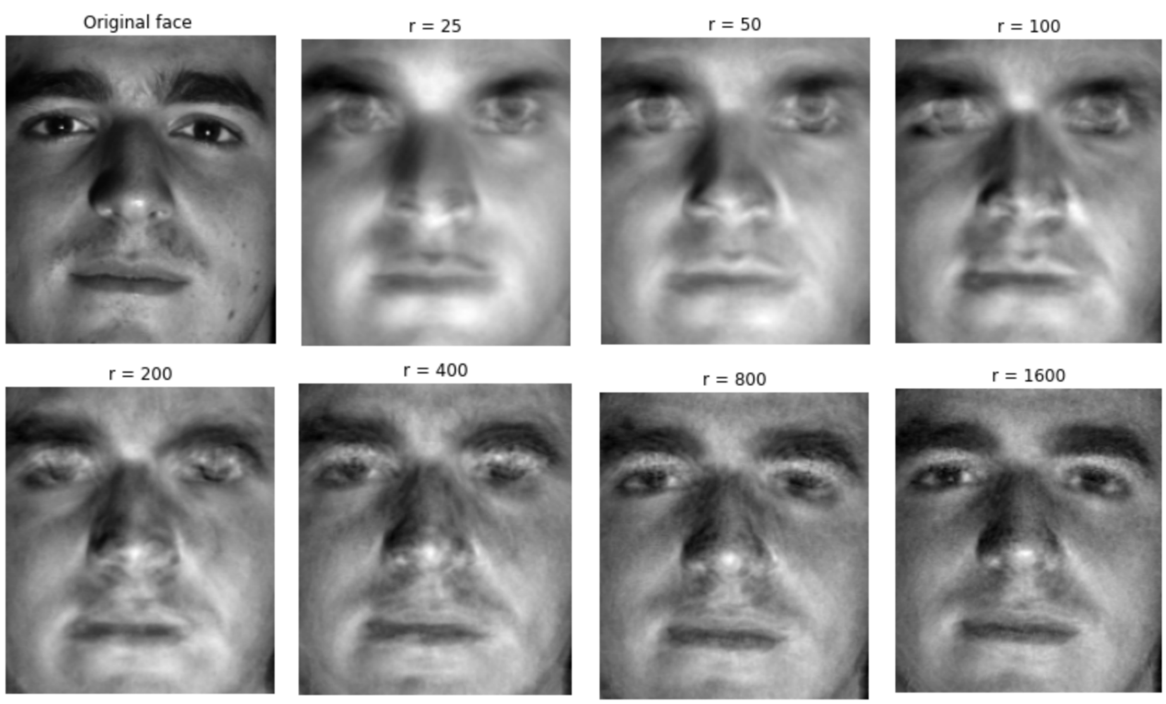

Figure 6. Eigenface approximation for 38th individual. Original image is provided for reference in the top left corner. r=800 and r=1600 provide the best approximations.

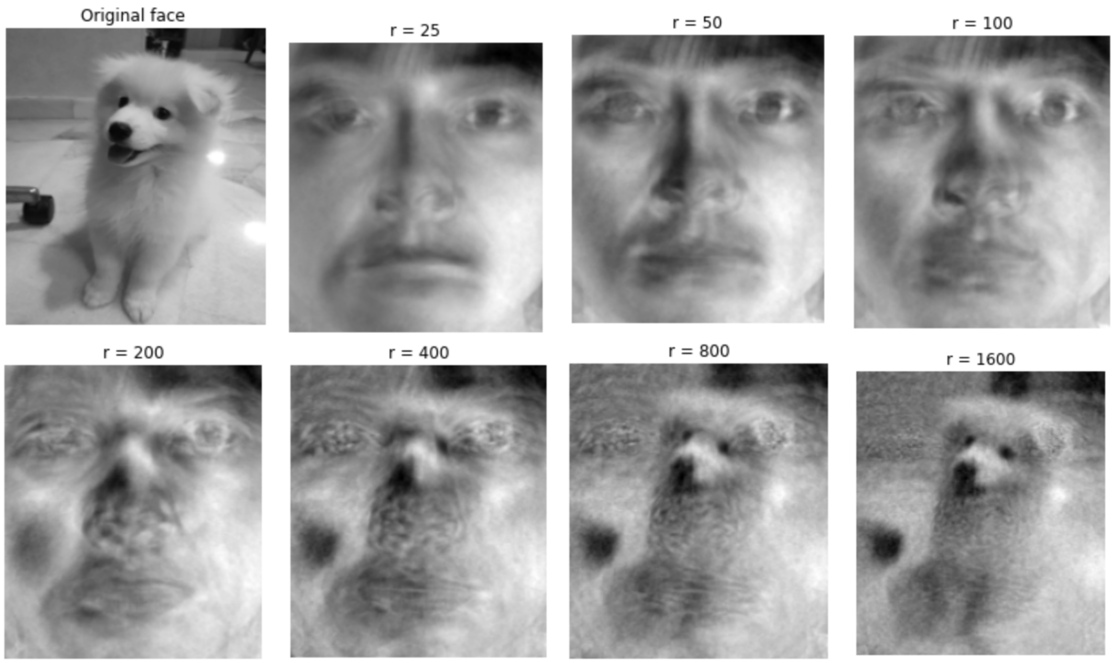

Figure 7. Eigenface approximation of a dog. As the rank r increases, the eigenface approximation becomes clearer.$r = 1600$ projections of PCA components can reconstruct a good eigenface approximation.

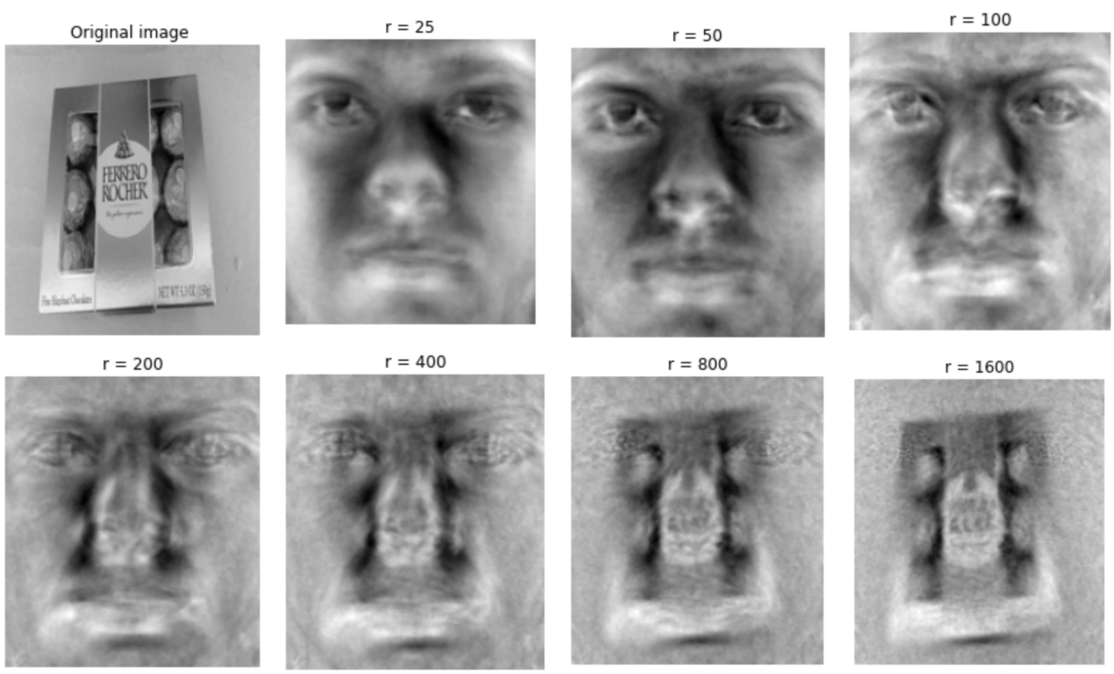

Figure 8. Eigenface approximation of abox of chocolates. As the rank r increases, the eigenface approximation becomes clearer.$r = 1600$ projections of PCA components can reconstruct a good eigenface approximation.

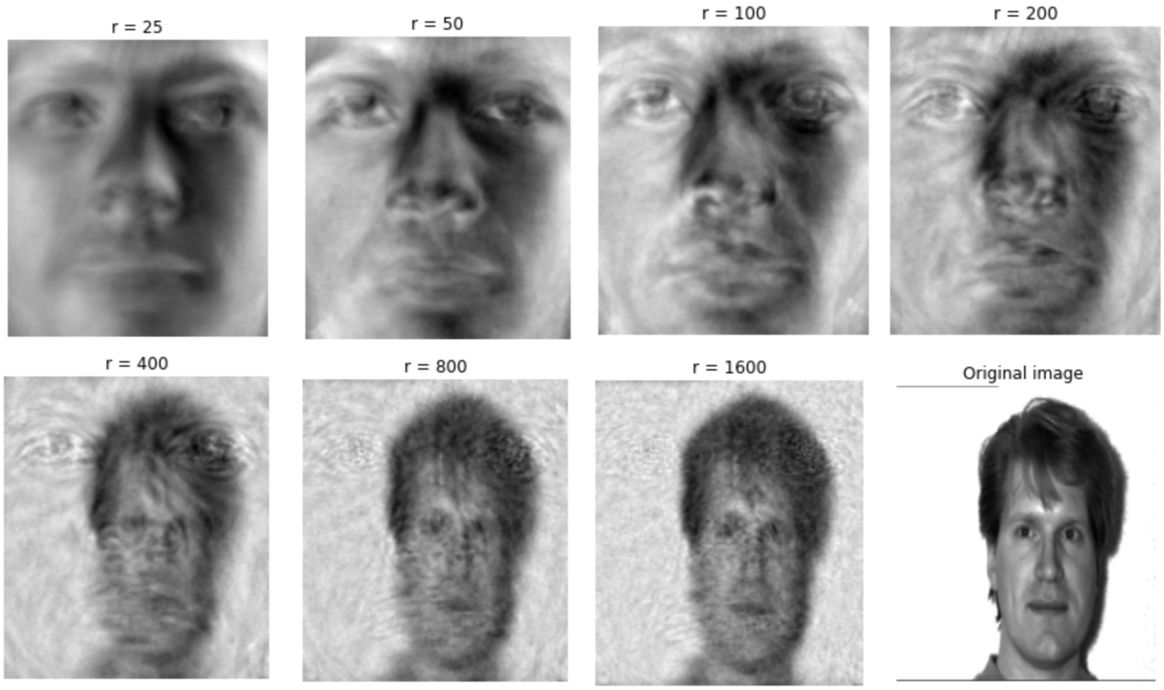

Figure 9. Eigenface approximation of one of the pose images. Eigenface approximation is still poor for the low rank approximation. Outline of a pose image can already be seen at $r = 200$, whereas before one could not discern what we were getting until $r = 800$ or $r = 1600$. Including the pose images led to better approximation, however it is important to note that some features of the individual are still indistinguishable.

5. Machine Learning

We used the concept of PCA to develop a machine learning algorithm that can recognize an individual image from the data set. To develop the algorithm, first of all we loaded the image data set of 25 individuals and converted them into *pandas* dataframe in python. We then have to separate our training set and testing set for which we used the blackbox function in python: *scikit.learn*. This function splits the input space into training data and test data; features and labels. We then performed PCA and used 150 components from the *scikit.learn*. The next step is to transform the training data into the PCA. Then we create a classifier class of Support Vector Classifier (SVC) and fit the created training data on the classifier. A bit of trial and error is needed at this point to create the best classifier fit. Now we pass test dataset to the classifier, and perform our prediction. Now we print our prediction report.

Figure 10. Figure shows some of the eigenfaces of our training dataset. The principal components captures the direction of maximum variance in the data. The white regions in the picture have the direction of maximum variance in our figures.

	precision	recall	f1-score	support
YaleB01	1.00	0.92	0.96	13
YaleB02	1.00	1.00	1.00	18
YaleB03	0.85	0.94	0.89	18
YaleB04	1.00	1.00	1.00	17
YaleB05	0.94	1.00	0.97	17
YaleB06	1.00	1.00	1.00	12
YaleB07	0.95	0.95	0.95	20
YaleB08	0.93	1.00	0.97	14
YaleB09	1.00	0.93	0.96	14
YaleB10	0.92	1.00	0.96	11
YaleB11	1.00	1.00	1.00	12
YaleB12	1.00	0.92	0.96	12
YaleB13	1.00	1.00	1.00	17
YaleB15	1.00	1.00	1.00	18
YaleB16	1.00	1.00	1.00	15
YaleB17	1.00	0.94	0.97	17
YaleB18	0.92	1.00	0.96	11
YaleB19	0.95	1.00	0.98	20
YaleB20	1.00	0.95	0.97	20
YaleB21	1.00	0.95	0.97	20
YaleB22	1.00	1.00	1.00	18
YaleB23	1.00	1.00	1.00	15
YaleB24	1.00	1.00	1.00	15
YaleB25	1.00	0.94	0.97	16
accuracy			0.98	380
macro avg	0.98	0.98	0.98	380
weighted avg	0.98	0.98	0.98	380

Figure 11. This figure shows the prediction of all the 25 individuals in the dataset. Precision column shows the accuracy in our prediction, whereas recall shows the number of times our model was correct in our prediction. There is a trade-off between precision and recall, and f-score is an indicator how well the model worked on both precision and recall.

6. Summary and Conclusion

In this project we used singular value decomposition (SVD) applications on the Yale Face Database. We used Principal Component Analysis (PCA), a which reduces the dimensionality of this data set whilst maintaining most of the information. Similarly, we explore average face, eigenfaces, rank-r SVD basis approximation, and implemented machine learning algorithm to predict individuals in a dataset.

I would like to thank my collaborators Amrit Parmar, Ariel Bonneau, and Asja Alic. I lastly want to note that all code and data files are available upon request.