Eigenface Decomposition

Preston Dunton
Colorado State University MATH 469

Introduction / Abstract

Eigenface decomposition is the application of Principal Component Analysis (PCA) on images that contain faces in them. This decomposition allows implementers to achieve data reduction in images by calculating the vectors in high dimensional space that represent the largest variation in images. These vectors are called eigenfaces, and are the principal components of a dataset of face images. The corresponding eigenvalues represent the variance in image data each eigenface explains. Applying this to data reduction, we observe a 2.701 : 1 data reduction, but also that modern data reduction techniques are more efficient. Furthermore, we observe that by keeping the first 1540 eigenvectors in $\mathbb{R}^{4096}$, we can retain 99.9

Eigenface Decomposition

1 Principal Component Analysis

The process of creating eigenfaces is a the use of a statistical method called Principal Component Analysis (PCA) on a dataset of face images. Each observation in the dataset is a different face, and each variable is the value that a given pixel takes on. For color images, each pixel has 3 values (red, green, and blue), but for greyscale images, there is only 1 value. In this project, greyscale images are used.

Principal Component Analysis is the process of taking data, and writing it in a more efficient representation. Let each image be $N \times N$ pixels. This means that we can flatten the image's pixels to be an $N^2 \times 1$ vector (a row vector here). We can think of our data as a collection of points $\mathbf{x}_i$ $(i = 1...M)$ in the N^2 dimensional space $\mathbb{R}^{N^2}$. Let X be our dataset with M observations. We can write X as an $N^2 \times M$ matrix [1].

Flattening Images

$$\mathbf{x}_i = \begin{bmatrix} x_{11} & \dots & x_{1N} \\ \dots & \dots & \dots \\ x_{N1} & \dots & x_{NN} \end{bmatrix} \longrightarrow \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ \dots \\ x_{NN} \end{bmatrix}$$

Data Matrix

$$\underset{(N^2 \times M)}{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_M \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \dots & \dots & \dots & \dots \\ x_{N^21} & x_{N^22} & \dots & x_{N^2M} \end{bmatrix}$$

PCA considers the covariance matrix of X. The covariance matrix is a multidimensional generalization of the variance statistic, and describes how data (vectors) are spread out in higher dimensional space. In a dataset with N^2 dimensions, the covariance matrix S will be of size $N^2 \times N^2$. S_{ij} represents the covariance between the i^{th} and j^{th} variables. When $i = j$, S_{ij} represents the variance of the i^{th} variable. To calculate S, we center our data X by subtracting its row-wise mean vector μ, multiply by the transpose, and divide by the corresponding degrees of freedom $(M - 1)$ [1].

Mean

$$\underset{(N^2 \times 1)}{\mu} = \frac{1}{M} \underset{(N^2 \times M)}{X} \underset{(M \times 1)}{\mathbf{1}} = \frac{1}{M} \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \dots & \dots & \dots & \dots \\ x_{N^21} & x_{N^22} & \dots & x_{N^2M} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \dots \\ \bar{x}_{N^2} \end{bmatrix}$$

Mean Centered Data

$$\underset{(N^2 \times M)}{\bar{X}} = \underset{(N^2 \times M)}{X} - \underset{(N^2 \times M)}{\begin{bmatrix} \mu & \dots & \mu \end{bmatrix}}$$

Covariance Matrix

$$\underset{(N^2 \times N^2)}{S} = \frac{1}{M - 1} \underset{(N^2 \times M)}{\bar{X}} \underset{(M \times N^2)}{\bar{X}^T}$$

Because the covariance matrix is square, we can interpret it as an operator on $\mathbb{R}^{N^2}$. In other words, $S \in \mathcal{L}(\mathbb{R}^{N^2})$. Consider the eigenvectors, $e_1, ..., e_{N^2}$, and eigenvalues, $\lambda_1, ..., \lambda_{N^2}$, of S. Because S is a covariance matrix, it is guaranteed to be positive, meaning that its eigenvalues are real and positive [2]. This corresponds to stretching space by positive scalars. These eigenvectors also form an orthonormal basis of $\mathbb{R}^{N^2}$, which means that their inner products are all 0, and there is no correlation between the eigenvectors [3]. Because there is no correlation between the eigenvectors, each eigenvector represents unique information in the space, and therefore is the optimal basis for representing our data in $\mathbb{R}^{N^2}$.

Ordered, Positive Eigenvalues $\lambda_1 \geq ... \geq \lambda_{N^2} \geq 0$

Orthonormal Eigenvectors $\langle e_i, e_j \rangle = \begin{cases} 0 & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases}$

We call $e_1, ..., e_{N^2}$ the principal components of X. The eigenvalues, $\lambda_1, ..., \lambda_{N^2}$ (with multiplicity), are the variances of X that each corresponding principal component explains [2]. When we transform space by the operator S, we are essentially stretching the principal components by the variances each principal component explains.

By ordering the eigenvectors in decreasing order of their eigenvalues and selecting the first k eigenvectors, we can create a basis for a subspace of $\mathbb{R}^{N^2}$ that maximally represents the variance of X [2]. In this application, if we plot the eigenvectors as images, they resemble faces. Therefore, they are also referred to as eigenfaces.

By projecting the observations of X onto the subspace spanned by our basis $e_1, ..., e_k$, we can perform dimension reduction. Dimension reduction is an important technique used in data compression and machine learning, which are discussed in later sections.

2 Eigenfaces and Reconstructed Images

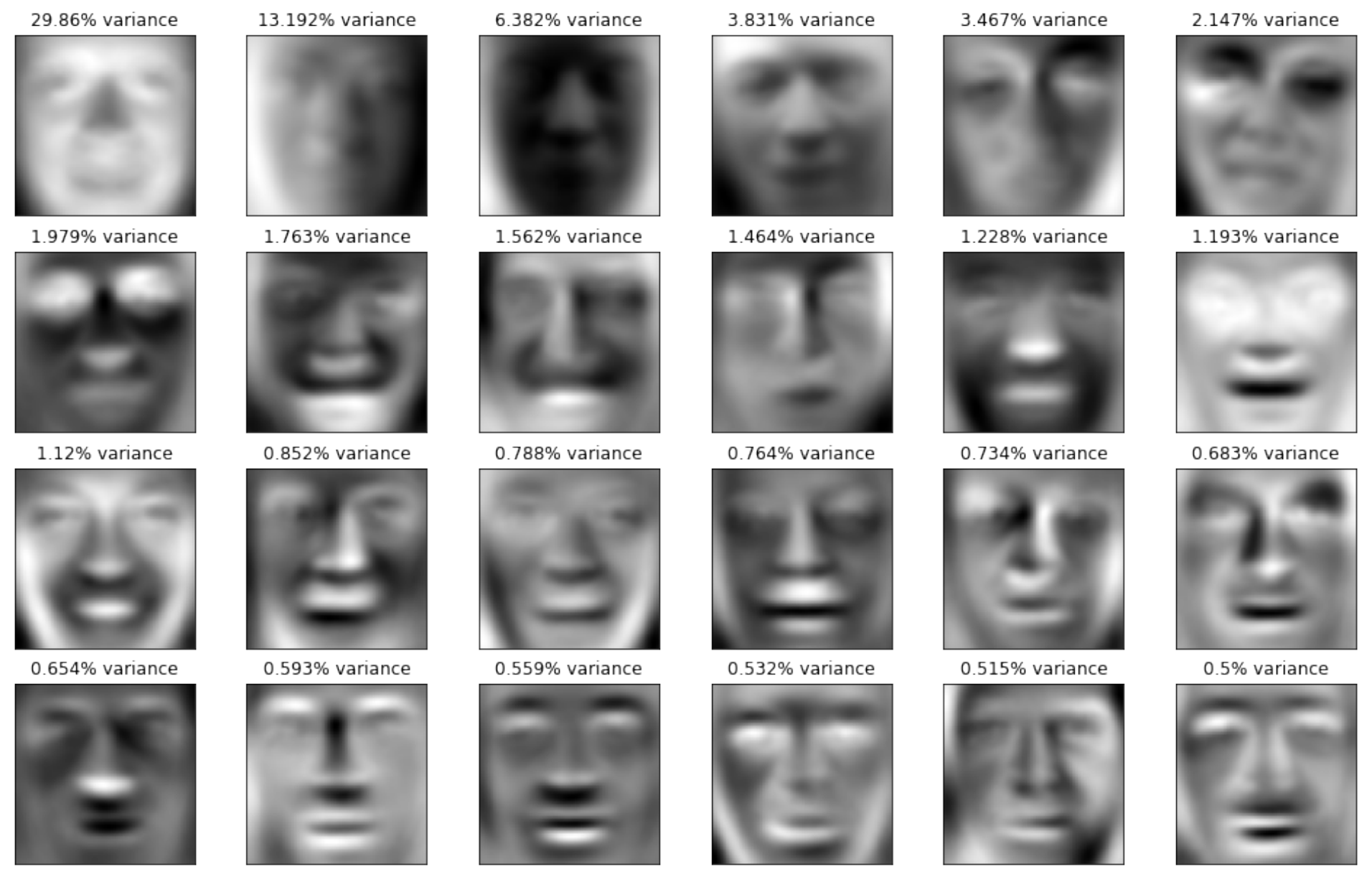

Fig. 1: The orthonormal basis of eigenvectors (eigenfaces)

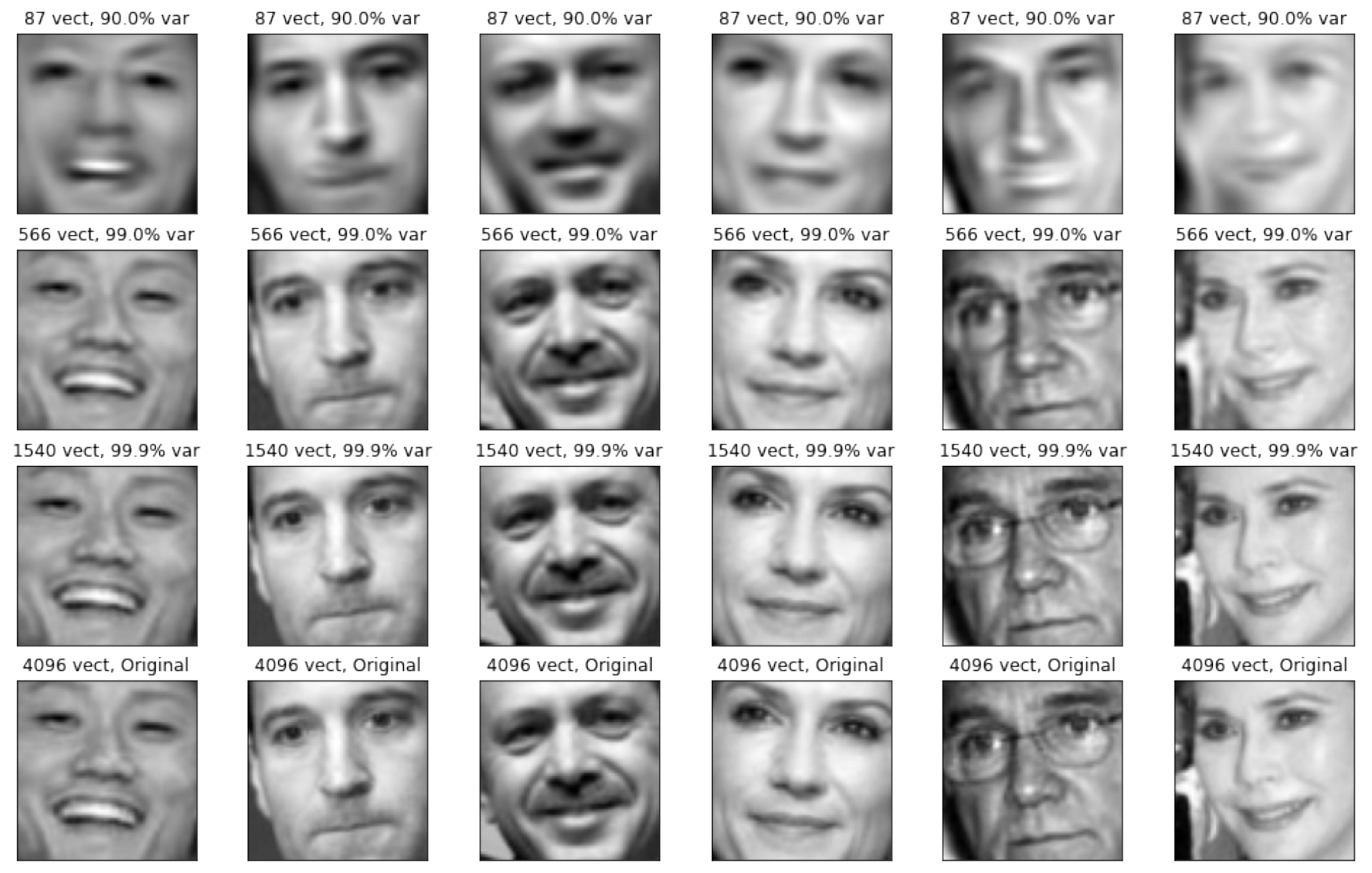

Fig. 2: Reconstructed images using eigenfaces.

Data Reduction Analysis

By choosing a number of eigenvectors k such that $k < N^2$, we can reduce the number of scalars we need to store for each image. This method is not lossless, but because the eigenvectors are sorted in order of the variance they explain, we can select the eigenvectors that will retain the most variance.

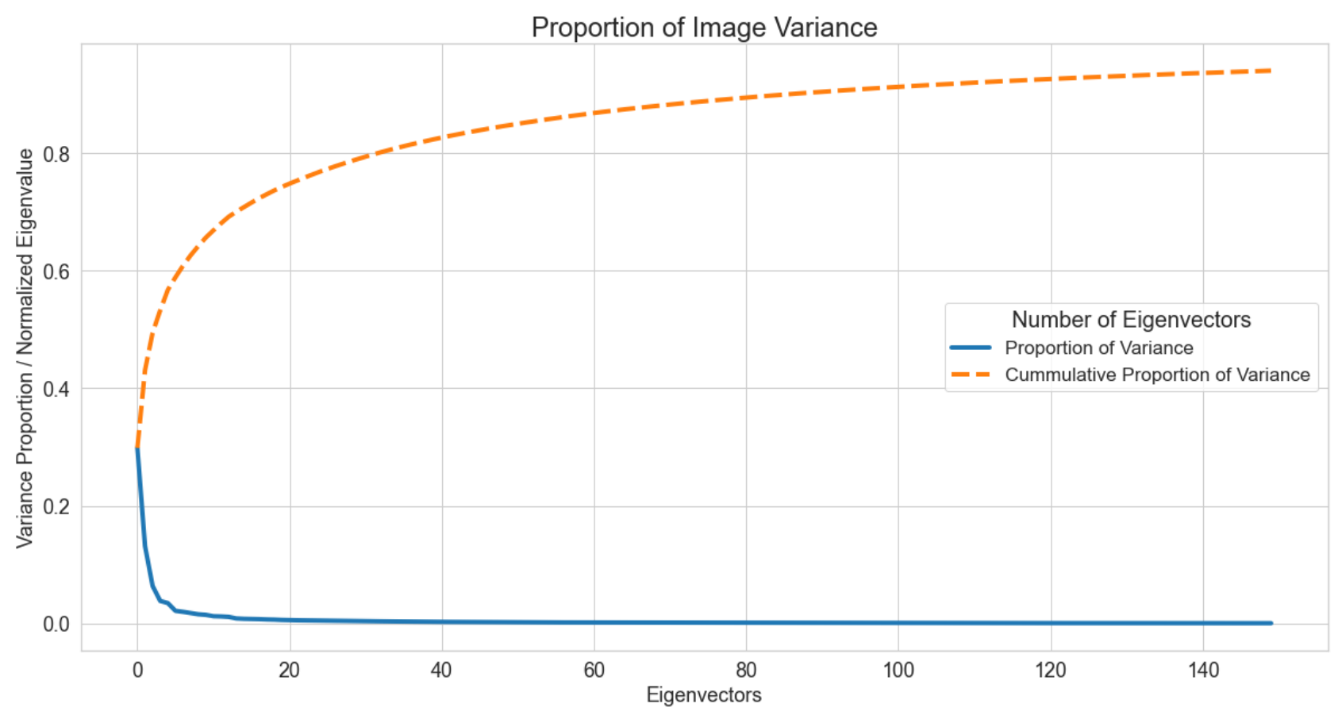

Fig. 3: Eigenvectors are ordered according to their eigenvalues (the variance they explain).

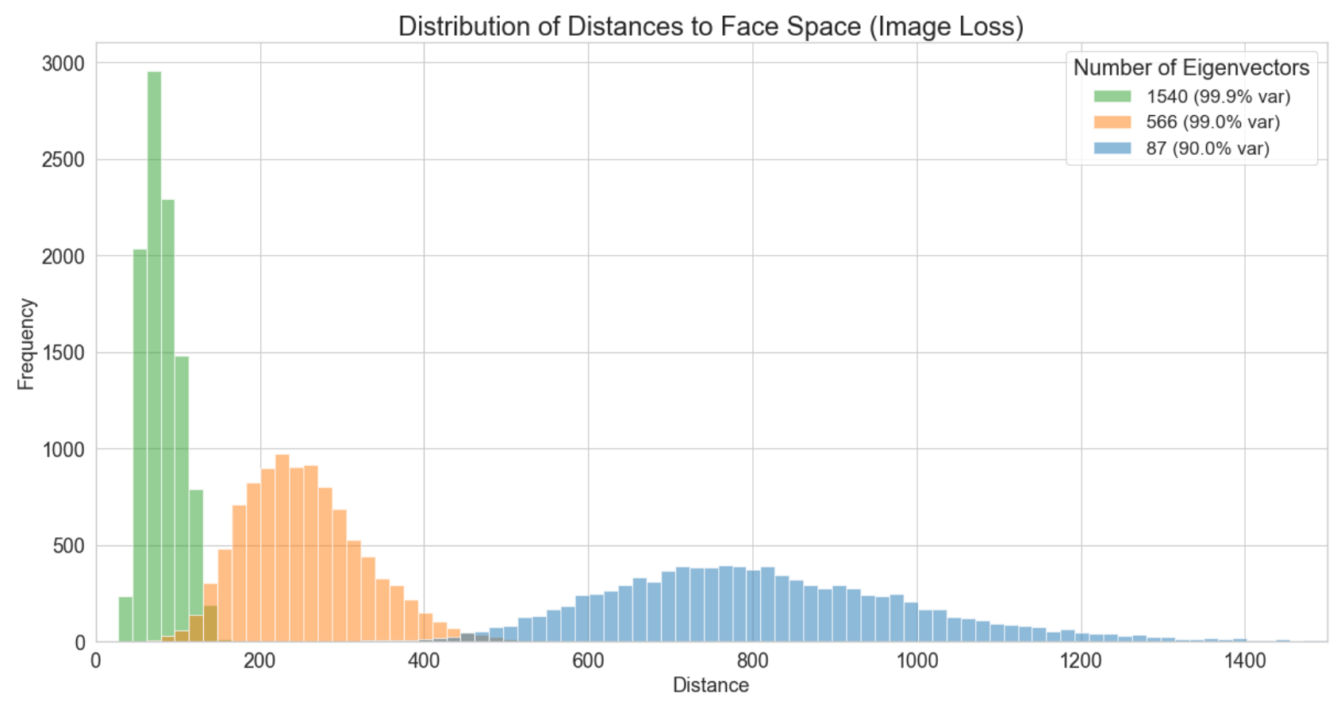

Fig. 4: Using more eigenvectors reduces the image loss (distance to face space).

Extensions

This method could be extended in many different ways. For example, the first choice is to use this method on higher resolution images. This would work because of our efficient computation of the eigenvectors of the covariance matrix. This method could also be used on other objects that contain predictable features, such as animals, flowers, and possibly cars.

Another extension is to bring this problem into the field of multi-linear algebra. By using tensors and other objects from that field, its possible to perform this decomposition for color images (a basis for each color channel), as well as perform this method without having to flatten images.

Acknowledgements

I'd like to thank Mark Shoemaker for being a great teacher of Linear Algebra II (MATH 469) during an online semester. He is very knowledgeable and has done a great job of interacting with his students.

References

[1] Turk, M.A., and A.P. Pentland. "Face Recognition Using Eigenfaces." Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991, doi:10.1109/cvpr.1991.139758.

[2] Zhou, Tianjian. "Lecture 9 Principal Component Analysis." STAT 460 - Applied Multivariate Analysis. STAT 460 - Applied Multivariate Analysis, 9 Mar. 2021, Fort Collins, CO, Colorado State University.

[3] Li, Casey Juanxi. "PCA, Visualized for Human Beings." Medium, Medium, 2 Aug. 2020, medium.com/@sometimescasey/pca-visualized-for-human-beings-47d19a122734.