# Image Compression with Principal Component Analysis

Andrew Roberts

## Background

Principal Component Analysis (PCA) is a statistical technique used to transform a data matrix into an orthogonal basis set. PCA computes the "principal axes" present in the data matrix; the centered data can then be orthogonally projected onto these principal axes to yield the *principal components (PCs)*. By discarding a number of the principal axes, dimensionality reduction can be performed on the data. In this paper, I utilize this technique to produce low-rank approximations of the *Olivetti Faces,* a set of face images taken at AT&T Laboratories in the mid 1990s.

## Brief overview of PCA and dimensionality reduction

PCA performs an eigendecomposition of the covariance matrix, yielding a set of eigenvectors that represent the principal axes of the data. The axis associated with the dominant eigenvalue represents the direction capturing the maximum variance in the data. In other words, if all observations existing in d-dimensional space were projected orthogonally onto a single axis, the eigenvector associated with the largest eigenvalue is the axis that maximizes the variance of these projected points. Of the remaining variance, the eigenvector associated with the next largest eigenvalue captures the most variability, and so on for the remaining eigenvectors. By taking only the k dominant eigenvectors (where $k < d$), the d-dimensional data can be projected onto a smaller k-dimensional subspace spanned by the first k principal axes. If most of the variability in the data is captured by relatively few of these principal axes, one can obtain a good low-rank approximation of the original data.

## PCA Applied to Image Compression: Three Approaches

Let $\mathbf{X}_{\mathbf{raw}}$ be a 400 x 4096 matrix storing the pixel values of 400 64 x 64 facial images. By subtracting the column-wise mean from each column I obtain the centered data matrix **X**. The eigendecomposition of the covariance matrix, $\mathbf{X}^{\mathbf{T}}\mathbf{X}/n$, gives the principal axes. I place these eigenvectors as columns in a 4096 x 4096 matrix **V** in *ascending* order according to the magnitude of their respective eigenvalues. The PCA projections ("scores") can now be easily computed by a single matrix multiplication, **XV**. In order to map the scores from PC space back to the original data space the scores matrix is then multiplied by $\mathbf{V}^{\mathbf{T}}$. I use these techniques in my image analysis by projecting onto the subspace spanned by only the k dominant eigenvectors and then projecting back onto the original data space, thus obtaining a low-rank approximation of the images. I discuss three mathematically identical but intuitively distinctive methods for achieving this below.

### 1.) Two Distinct Projections

This first approach is essentially the procedure I briefly described above. It is distinct from the other approaches outlined here not because the mathematical result is distinct, but rather because it offers a unique intuition. First, the centered data matrix is multiplied by the matrix containing the eigenvectors of $\mathbf{X}^{\mathbf{T}}\mathbf{X}/n$. The interpretation of the resulting 400 x 4096 matrix is straightforward. The $j^{th}$ column of **XV** is a 400-dimensional vector containing all 400 scalar projections of the images onto the jth principal axis. Row i of the matrix represents the ith image in PC space. Let us briefly consider projecting the first row of **XV** back onto the original data space. Let $\mathbf{z_1} = (z_1, z_2, \ldots, z_{4096})$ be a row vector signifying the first face image in PC space. It can be mapped to the original space with a simple computation: $\mathbf{x_1} = \mathbf{z_1}\,\mathbf{V}^T$. Intuitively, this computation represents a linear combination of the rows of $\mathbf{V}^T$ (i.e., the eigenvectors of the covariance matrix) scaled by the entries of $\mathbf{z_1}$. Let **Z** be the "scores" matrix discussed above (the product of **X** and **V**); then the product $\mathbf{ZV}^T$ carries out this projection for every face image in a single computation. If all 4096 eigenvectors are utilized then the data matrix will remain unchanged. However, **V** and **Z** can be narrowed to the first k dominant eigenvectors. In this application, this corresponds to selecting the last k columns of the two matrices, yielding $\mathbf{V_k}$ and $\mathbf{Z_k}$. Now computing $\mathbf{Z_k V_k}^T$ produces a rank-reduced approximation of the original data matrix. To return to the truly original data representation I also add the 4096 x 1 mean vector $\boldsymbol{\mu}$ to this product, where $\boldsymbol{\mu}_j$ represents the mean of the jth column of $\mathbf{X_{raw}}$. The computation $\mathbf{Z_k V_k}^T + \boldsymbol{\mu}$ will produce a good approximation of the original images as long as k is large enough and there exists a latent structure underlying the higher dimensional data.

## 2.) The projection matrix

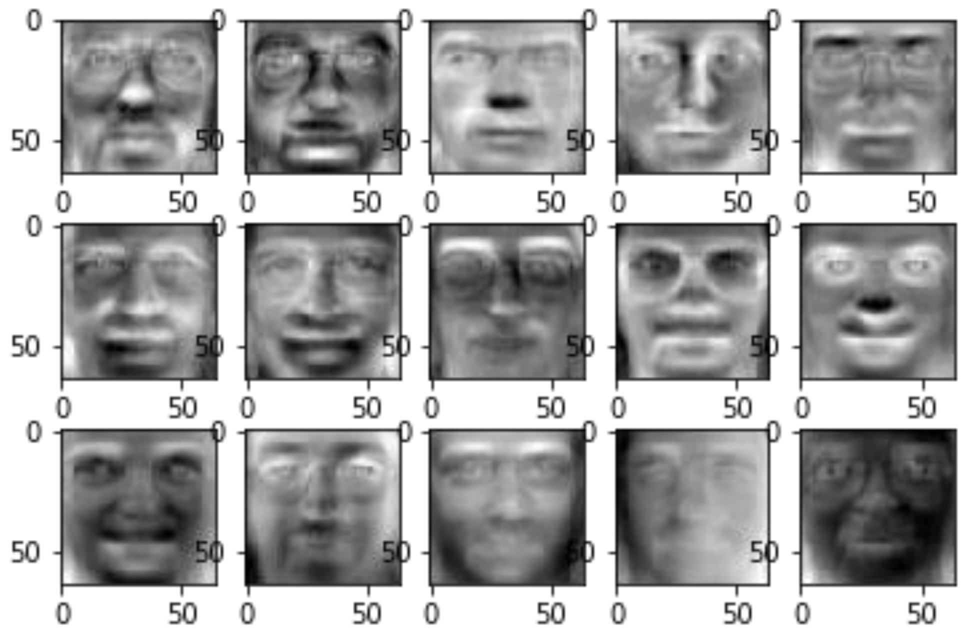
I need not spend much time discussing this second method, as it is computationally identical to method 1. However, it provides a distinct perspective from which to view the computation $\mathbf{Z_k V_k}^T + \boldsymbol{\mu}$. It can be derived that, given a basis **A** for $R^n$, a subspace **V** of $R^n$, and a vector **x** in $R^n$, then $\text{proj}_V\,\mathbf{x} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{x}$. Let us apply this fact to the image compression example. The ultimate goal is to project vectors existing in 4096-dimensional space into a k-dimensional subspace spanned by the k dominant eigenvectors of the covariance matrix. The symmetry of $\mathbf{X}^T\mathbf{X}/n$ implies that its eigenvectors form an orthonormal eigenbasis of the data space. This implies that the columns of **V** are linearly independent and thus $\mathbf{V}^T\mathbf{V}$ is invertible. I will first look at the case where k = 4096 (i.e., no eigenvectors are discarded). Therefore, the projection matrix becomes $\mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T = \mathbf{V}\mathbf{I}^{-1}\mathbf{V}^T = \mathbf{V}\mathbf{I}\mathbf{V}^T = \mathbf{V}\mathbf{V}^T = \mathbf{I}$, where I is the identity matrix. The first equality draws on the orthogonality of **V** and the fourth is a direct result of including all eigenvectors in **V**. This is the same result briefly mentioned above; if the projection matrix is applied to **X**, it leaves **X** unchanged. Now I will explore the truly interesting case, where k < 4096. The computation is exactly the same except that $\mathbf{V_k V_k}^T$ doesn't reduce to the identity. Therefore, applying the projection to **X** results in $\mathbf{X}(\mathbf{V_k V_k}^T) = \mathbf{Z_k V_k}^T$, the same exact result as discussed in the first approach.

## 3.) The Singular Value Decomposition (SVD)

The above two approaches focus on the eigendecomposition of the covariance matrix. I will briefly show here how the SVD can be utilized to obtain the same result. First, let us view the covariance matrix from a slightly different perspective. $\mathbf{X}^T\mathbf{X}/n$ is symmetric and can therefore be diagonalized as $\mathbf{X}^T\mathbf{X}/n = \mathbf{VDV}^T$, where **V** is the matrix of eigenvectors and **D** is the diagonal matrix of corresponding eigenvalues arranged in decreasing order. Just as before, one can compute **XV** to obtain the PC scores matrix. Now, I will perform the SVD of the data matrix to produce $\mathbf{X} = \mathbf{USV^T}$, where $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ are 400 x 400, 400 x 4096, and 4096 x 4096 matrices, respectively. For the purposes of performing PCA, it is significant that $\mathbf{V}$ contains the right singular vectors of $\mathbf{X}$ while $\mathbf{S}$ is a diagonal matrix containing the associated singular values (again, arranged in decreasing order). To make the final connection to PCA I will re-compute the covariance matrix utilizing the factors from the SVD: $\mathbf{X^TX}/n = \mathbf{(USV^T)^T(USV^T)}/n = \mathbf{(VS^TU^T)(USV^T)}/n = \mathbf{(VSU^T)(USV^T)}/n = \mathbf{VSISV^T}/n = \mathbf{V(S^2}/n\mathbf{)V^T}$. This derivation follows from the facts that $\mathbf{S^T=S}$ by symmetry and $\mathbf{U^TU=I}$ by orthogonality. Writing the covariance matrix in this form makes the parallels between PCA and SVD clear. The principal axes are contained in the right singular vector matrix, $\mathbf{V}$, while the singular values of $\mathbf{X}$ are simply the square roots of the eigenvalues of $\mathbf{X^TX}/n$. Therefore, the PCs are given by $\mathbf{XV} = \mathbf{USV^TV} = \mathbf{US}$ ($\mathbf{V}$ is orthonormal). Rank reduction can be performed by taking the first k columns of $\mathbf{U}$, the k x k upper-left part of $\mathbf{S}$ and the first k rows of $\mathbf{V^T}$: $\mathbf{X_k} = \mathbf{U_kS_kV_k^T}$.

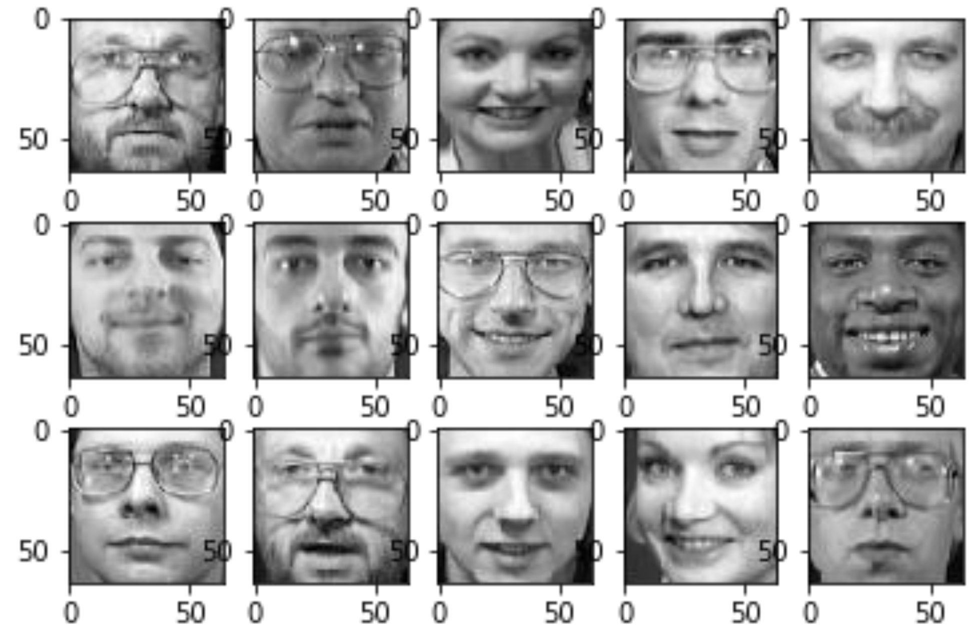
## Reshaping eigenvectors as images

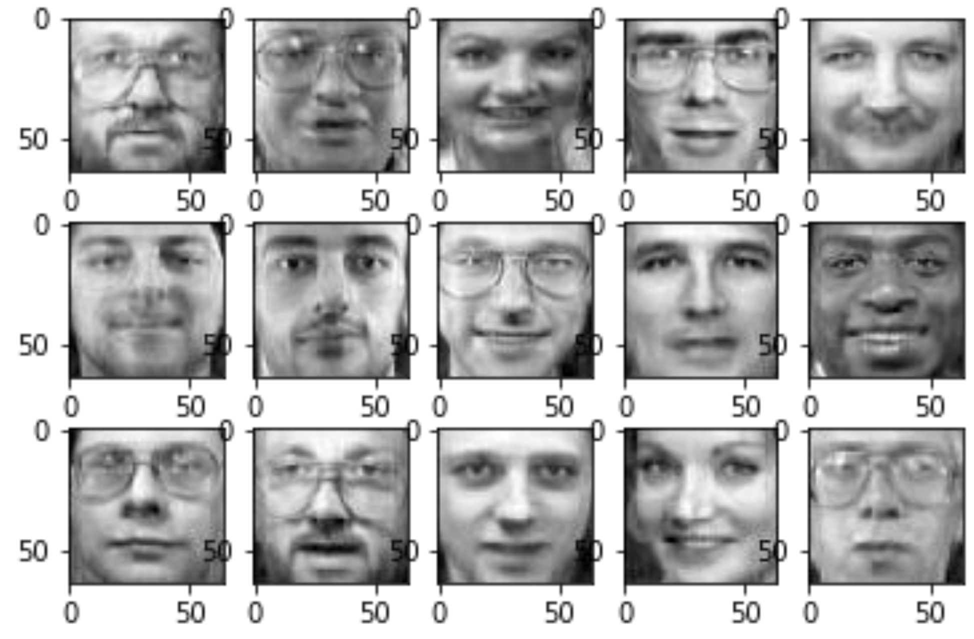
The set of eigenvectors of the covariance matrix (or, alternatively, the set of right singular vectors of the data matrix) form a 4096 x 4096 square matrix containing the principal axes as columns. These columns can easily be reshaped into 64 x 64 images and visualized. I take the first 15 dominant eigenvectors and visualize them as images below. The bottom right image is the dominant eigenvector while the top left is the eigenvector with the 15th largest eigenvalue. For a rank 15 approximation, all of the images can be thought of as linear combinations of these 15 faces.



## Rank 100 Approximation

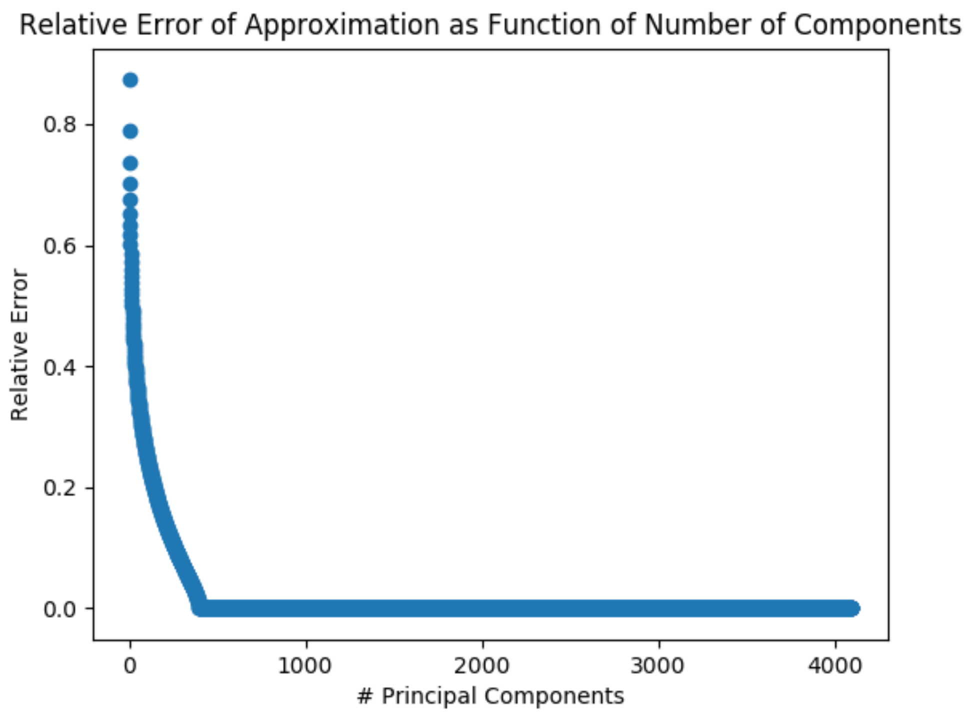
As the focus of this paper is on dimensionality reduction, I will now approximate these images by projecting them onto a 100-dimensional subspace spanned by the 100 dominant eigenvectors of the covariance matrix. To do so using the first approach discussed above, I compute $\mathbf{Z} = \mathbf{XV}$ to produce the 400 x 4096 PC matrix. I then take the last 100 columns (corresponding to the first 100 PCs) of $\mathbf{Z}$ and the first 100 rows (corresponding to the first 100 principal directions) of $\mathbf{V}^\mathbf{T}$ and compute $\mathbf{X}_{\mathbf{100}} = \mathbf{Z}_{\mathbf{100}}\mathbf{V}_{\mathbf{100}}{}^{\mathbf{T}}$. The first 15 images shown below are a random subset from $\mathbf{X}_{\mathbf{raw}}$ (i.e., the original faces), shown for comparison. The next set of images is the same 15 faces approximated in the 100-dimensional subspace determined by PCA. To the human eye, it seems that the dominant 100 principal directions captures most of the variation in the images, although they have clearly lost a small amount of information. See *Appendix* for a plot of the first 100 eigenvalues.

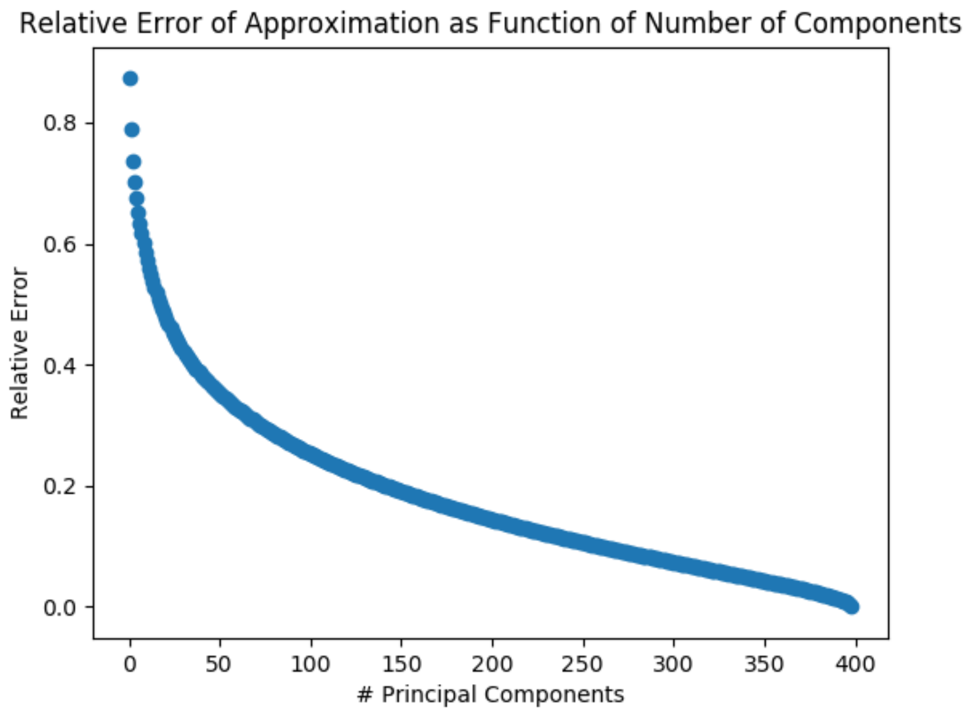




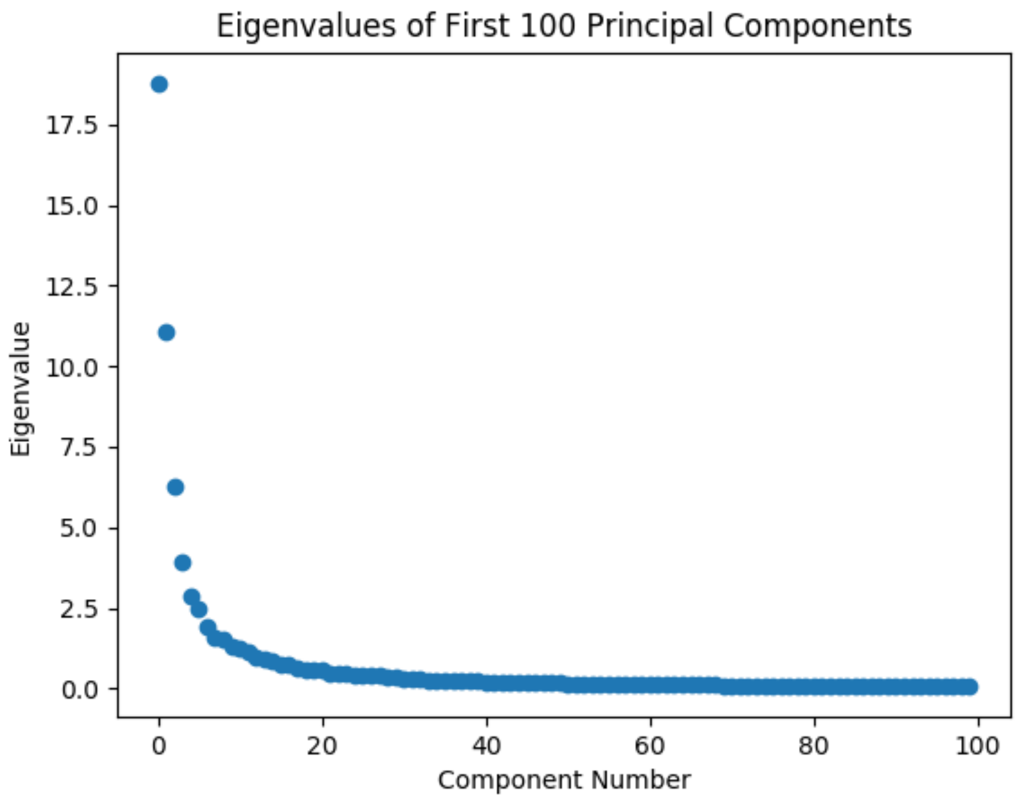
## Relative Error of Approximation

The above section reduces the dimensionality of the data matrix using a rank-100 approximation. I do the same for all values of k in the range [1, 4096] and compute the relative error between **X** and $\mathbf{X_k}$ for each k. The Frobenius Norm is used for the error calculation. The first plot below shows the entire range of k, while the second cuts out all PCs beyond the "elbow" (i.e., where the change in relative error is essentially zero).





## Appendix: Eigenvalue (“Scree”) Plot


Eigenvalues of First 100 Principal Components
Eigenvalue
Component Number