Reconstructing Pictures using PCA and calculation of percentage of data loss

Anil Katwal
School of Natural science and Mathematics
University of Southern Mississippi , Hattiesburg, Mississippi
Anil.Katwal@usm.edu

November 14, 2023

Abstract

This report explores the reconstruction of the well-known Lena.jpg image using the Principal Component Analysis (PCA) algorithm. We vary the number of principal components to understand how it affects the quality of reconstructed images and measure the percentage of data loss in each reconstructed photo. The study sheds light on the balance between reducing complexity and maintaining image details in PCA. Results show that increasing the number of principal components improves the accuracy of reconstructed images, revealing insights that can guide decisions for practical applications of PCA in image processing.

1 Introduction

Principal Component Analysis (PCA) has a rich history rooted in the early 20th century, with key developments by influential statisticians. Karl Pearson's work in 1901 laid the groundwork by introducing the concept of correlation, which set the stage for understanding relationships between variables. In 1933, Harold Hotelling expanded on Pearson's ideas and introduced the concept of PCA, demonstrating that it's possible to transform original variables into uncorrelated principal components that capture significant information. Parallelly, Eckart and Young, in 1936, contributed to the development of Singular Value Decomposition (SVD), a mathematical technique closely related to PCA. The work of John W. Tukey in 1965 on exploratory data analysis and visualization techniques also influenced PCA's evolution. Golub and Kahan's contributions in 1965 to the numerical computation of SVD were pivotal. Over time, Gideon Schwarz in 1978 introduced a method for determining the number of principal components. Today, PCA is a fundamental technique applied in statistics, signal processing, image analysis, and machine learning for dimensionality reduction, feature extraction, and data visualization in various scientific and industrial fields.

Principal Component Analysis (PCA) is a widely used statistical method with diverse applications in data analysis and pattern recognition. One of its primary uses is in dimensionality reduction, where it transforms a high-dimensional dataset into a lower-dimensional space while retaining as much variance as possible. This reduction in dimensionality brings several advantages, such as enhanced computational efficiency and improved model performance.

PCA finds its utility in feature extraction, particularly when dealing with datasets characterized by highly correlated variables. By creating uncorrelated variables, known as principal components, PCA simplifies the interpretation of data, making it easier to identify and focus on the most significant features. Furthermore, the method is instrumental in data visualization, aiding in the representation of complex datasets in a more manageable form. This visualization facilitates a deeper understanding of the underlying patterns and relationships within the data.

An additional benefit of PCA is its ability to filter out noise and irrelevant information, leading to a more refined dataset with an improved signal-to-noise ratio. In applications like image processing and speech recognition, where noise reduction is crucial, PCA plays a vital role in enhancing the quality of the analysis.

However, PCA is not without its limitations. One notable disadvantage is the potential loss of interpretability, as the principal components derived through PCA are linear combinations of the original variables. This lack of direct correspondence with the initial features can complicate the interpretation of results. Furthermore, PCA assumes linearity in the relationships between variables, and its sensitivity to outliers may impact the robustness of the analysis.

Despite these limitations, PCA remains a valuable tool in various fields, offering a balance between dimensionality reduction, feature extraction, and noise reduction. Its application requires careful consideration of its assumptions and potential drawbacks, ensuring that it aligns with the specific characteristics and goals of the dataset under examination.

2 Problem

PCA can be used for data compression. We can apply the PCA to compress the given picture (lena.jpg), then reconstruct the picture using various numbers of principal components. Show the percentage of data loss in each reconstructed photo. Note: Do not use MATLAB 'pca' command.

Figure 1: Original Lena image.

3 Data Collection and Preprocessing

Step 1: Image Loading

The initial step involves loading the image data into a format suitable for processing. We use the well-known "lena.jpg" figure for our study.

Step 2: Image Representation

Images are represented as matrices, with each element corresponding to the pixel value. For grayscale images, the matrix is 2D, and for color images, it is 3D, representing height, width, and color channels.

Step 3: Grayscale Conversion

To make our analysis simpler, especially for PCA, we're using a grayscale image. Grayscale images have just one channel, which makes things easier. Luckily, we already have a grayscale image, so we don't need to go through the process of converting a color image.

Step 4: Standardization

PCA is sensitive to data scale. To ensure uniformity, we standardize pixel values by subtracting the mean and dividing by the standard deviation of each channel.

Step 5: Flattening the Image

For PCA, we flatten the 2D matrix (for grayscale) or 3D matrix (for color) into a 1D vector. Each row becomes a data point in our dataset.

Step 6: Data Matrix Formation

We form a data matrix where each column represents a different data point (flattened image), and each row represents a feature (pixel).

Step 7: Data Centering

To center the data around zero, we subtract the mean of each feature (pixel) from the corresponding column in our data matrix. After completing these preprocessing steps, our data is ready for Principal Component Analysis (PCA). Subsequent sections will delve into the details of the PCA algorithm and present the results of our analysis.

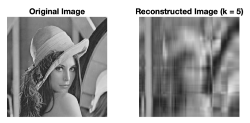
3.1 PCA Algorithm

```
Input: Data matrix $X$ $(n \times m)$, number of principal components $k$
Output: Reduced data matrix $Y$ $(n \times k)$
Standardize the Data:;
for $j = 1$ to $m$ do
    Calculate mean $(\mu_j)$ and standard deviation $(\sigma_j)$ of column $j$;
    for $i = 1$ to $n$ do
        Standardize column $j$: $x_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$;
    end
end
Compute Covariance Matrix:;
Compute the covariance matrix $C = \frac{1}{n-1} \cdot X^T \cdot X$;
Calculate Eigenvalues and Eigenvectors:;
$(\text{eigenvalues}, \text{eigenvectors}) = \text{Eigendecomposition}(C)$;
Sort Eigenvalues and Corresponding Eigenvectors:;
Sort eigenvalues in descending order and arrange corresponding eigenvectors accordingly;
Select Principal Components:;
Select the top $k$ eigenvectors to form the matrix $V_k$;
Create the Projection Matrix:;
Projection matrix $P$ = first $k$ columns of $V_k$;
Project the Data:;
Project the standardized data onto the $k$-dimensional subspace: $Y = X \cdot P$;
```

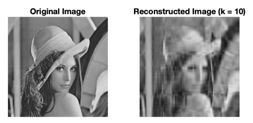
Algorithm 1: Principal Component Analysis (PCA)

4 Results

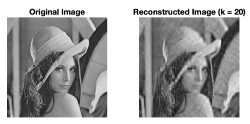
We utilized the PCA algorithm to reconstruct the image, varying the number of principal components ($K = 5, 10, 20, 30, 50, 70, 100, 200$). The reconstruction process involved capturing the essential features of the image using a reduced set of principal components, thereby reducing the dimensionality of the data. As we increased the value of K, we observed a trade-off between image clarity and computational efficiency. With smaller K, the reconstructed images retained only the most prominent features, resulting in a more compressed representation. Conversely, higher K values led to more detailed reconstructions, closely resembling the original image. This experimentation provided valuable insights into the impact of principal component selection on the trade-off between image fidelity and computational complexity in the context of PCA-based image reconstruction.



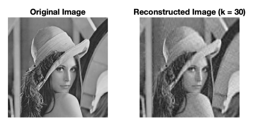
(a) $K = 5$



(b) $K = 10$



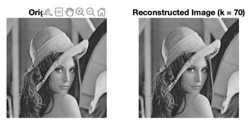
(c) $K = 20$



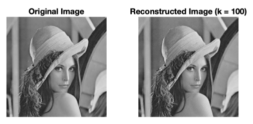
(d) $K = 30$

Figure 2: Lena image using PCA, demonstrating the reconstruction for principal components $K = 5, 10, 20, 30$, When the K is small , reconstructed images retained only the most prominent features, resulting in a more compressed representation.

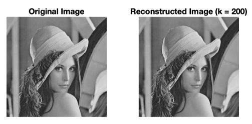
(a) $K = 50$



(b) $K = 70$



(c) $K = 100$



(d) $K = 200$

Figure 3: Lena image using PCA, demonstrating the reconstruction for principal components $K = 50, 70, 100, 200$. Higher K values led to more detailed reconstructions, closely resembling the original image.

5 Discussion

The following table summarizes the percentage of data loss for different numbers of principal components (k):

Number of Principal Components (k)	Percentage of Data Loss
200	0.00%
100	0.36%
70	1.19%
50	2.56%
30	5.83%
20	9.51%
10	19.28%
5	34.58%

The graph in Figure 4 illustrates the relationship between the number of principal components (k) and the corresponding percentage of data loss in our Principal Component Analysis (PCA) based image reconstruction method. As shown in the adjacent table, the x-axis represents the number of principal components, ranging from 5 to 200, while the y-axis represents the percentage of data loss during the reconstruction process.

The general trend observed in the graph indicates a decrease in data loss as the number of principal components increases. This suggests that higher values of k lead to more accurate image reconstructions with less information loss. The graph provides valuable insights into the trade-off between computational efficiency gained by reducing k and the resulting loss of image fidelity. Higher values of k produce reconstructed images that more closely resemble the original, but at the cost of increased computational requirements.

In conclusion, our experiment highlights the impact of principal component selection on the quality of PCA-based image reconstructions. Researchers and practitioners can refer to this analysis to make informed decisions based on their specific requirements, considering the balance between computational efficiency and image fidelity in PCA applications.

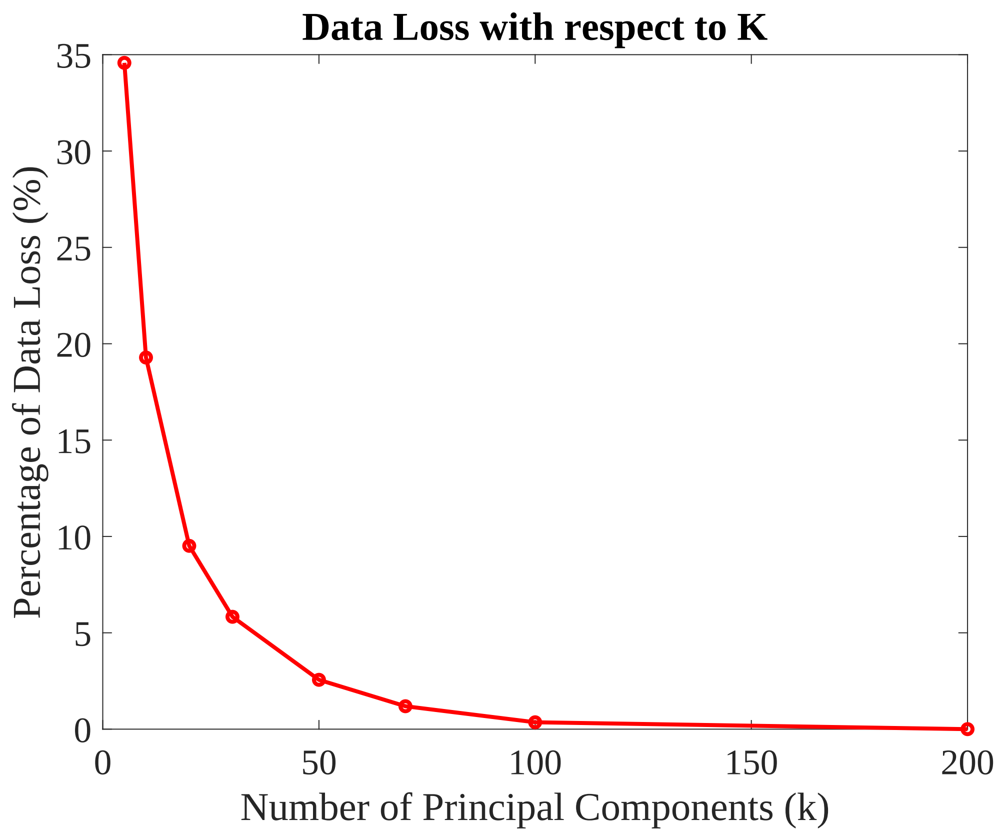


Figure 4: Data Loss vs. Number of Principal Components