SINGULAR VALUE DECOMPOSITION IN IMAGE COMPRESSION

KHANH DINH

ABSTRACT. This project explores the use of Singular Value Decomposition for image compression, covering both theory and practical applications. It reviews key linear algebra concepts, examines low-rank approximations, and evaluates how varying the number of singular values affects image quality and compression efficiency. It also investigates the impact of removing large singular values, compares compression across color spaces, and tests robustness against noise.

1 INTRODUCTION

Digital images are fundamentally represented as matrices of pixel values. Grayscale images utilize a single matrix to capture intensity values, whereas color images require three channels—red, green, and blue—to represent their full spectrum. For example, a high-resolution 4K image (3840x2160 pixels) contains over 8 million pixels [6], which can take up a lot of memory. Therefore, in today's data-driven digital era, image compression is crucial for efficient storage and transmission of visual information. In this project, we will learn about Singular Value Decomposition (SVD), a powerful linear algebra tool for image compression.

The development of SVD has a rich history, beginning with foundational theoretical advances by mathematicians like Beltrami and Jordan (1873), Sylvester (1889), Schmidt (1907), and Weyl (1912). These milestones, detailed by Stewart in [8], provide a comprehensive view of the field's evolution. Stewart's work not only contextualizes these contributions but also bridges them to more recent computational advancements, emphasizing progress in stability and efficiency.

Date: January 17, 2025.
My sincere gratitude to Professor Margaret Robinson for her meticulous proofreading, invaluable feedback, and inspirational guidance throughout MATH251. Her introduction to this fascinating topic and MATLAB has been instrumental in my project.

In this project, we explore the mathematical foundation of SVD, its application in low-rank approximations for image compression, and its performance evaluation using Compression Ratio (CR) and Mean Squared Error (MSE). We examine the impact of retained singular values, color space transformations, and noise on compression effectiveness through four experiments. The report begins with an introduction to essential linear algebra concepts, followed by a detailed explanation of SVD mechanics and applications (Section 2). In Section 3, we conduct four experiments on rank variation, singular value removal, color spaces, and noise. Each experiment covers methodology, results, and analysis. Finally, the conclusion summarizes our findings and acknowledges the limitations of the project.

All computations and visualizations were performed using MATLAB's built-in image manipulation functions and custom functions from the `SVDCompressor` package, which was developed by the author and is available at this GitHub repository.

We begin by reviewing fundamental linear algebra concepts that provide the necessary groundwork for understanding the SVD:

Definition 1.1. An $n \times n$ matrix P is called **orthogonal** if it satisfies the condition $P^T = P^{-1}$. Two vectors are said to be **mutually orthogonal** if their dot product is zero, meaning they are perpendicular to each other in space. Orthogonal matrices have this property: their columns and rows are mutually orthogonal unit vectors. [4]

Example 1.2. Consider the 2×2 rotation matrix: $P = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, where θ is the angle of rotation. Note that $P^T P = I$, where I is the identity matrix:

$$P^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \quad P^T P = \begin{bmatrix} \cos^2\theta + \sin^2\theta & 0 \\ 0 & \cos^2\theta + \sin^2\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Thus, $P^T P = I$ and P is orthogonal.

Definition 1.3. A set of vectors is said to be **orthonormal** if the vectors are mutually orthogonal and each vector has magnitude one. [4]

Example 1.4. Consider the standard basis vectors in $\mathbb{R}^3$: $\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, $\mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$. We know that the dot product between any two distinct vectors is zero: $\mathbf{e}_1 \cdot \mathbf{e}_2 = 0$, $\mathbf{e}_2 \cdot \mathbf{e}_3 = 0$, $\mathbf{e}_1 \cdot \mathbf{e}_3 = 0$, and each vector has a magnitude of 1: $\|\mathbf{e}_1\| = 1$, $\|\mathbf{e}_2\| = 1$, $\|\mathbf{e}_3\| = 1$. Thus, the set $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ is orthonormal.

Definition 1.5. For real-valued matrices, a square matrix U is **unitary** if $UU^T = U^T U = I$.

Unitary matrices can have complex or real elements, while orthogonal matrices must have real elements. Both orthogonal and unitary matrices share the property that their columns (and rows) form an orthonormal set of vectors. Thus, the vectors are mutually orthogonal and **normalized** (which means each vector has a length of one). For real-valued matrices, the **conjugate transpose** (denoted by *) and the **transpose** are the same. Therefore, an orthogonal matrix is a unitary matrix with real-valued elements. [2] In the next section, we will learn that the terms U and V in the singular value decomposition (SVD) are both unitary and orthogonal matrices. The 2×2 rotation matrix described in 1.2 is an example of a unitary matrix that is also orthogonal.

Definition 1.6. The **trace** of a square matrix $A \in \mathbb{R}^{n \times n}$ is defined as the sum of its diagonal elements: $tr(A) = \sum_{i=1}^{n} a_{ii}$, where a_{ii} is the i-th diagonal entry of the matrix A. [9] For example, the trace of $A = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$ is $tr(A) = 1 + 5 = 6$.

Definition 1.7. The **Frobenius norm** (also called the **Hilbert-Schmidt norm**) of a matrix $A \in \mathbb{R}^{m \times n}$ is a measure of the magnitude of the entries in the matrix. It is defined as the square root of the sum of the absolute squares of all the entries in the matrix. The Frobenius norm is expressed as:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{tr(A^T A)},$$

where $\|\mathbf{A}\|_F$ represents the Frobenius norm of the matrix $\mathbf{A}$, and a_{ij} represents the entry in the i-th row and j-th column of the matrix A.

Example 1.8. Consider the matrix $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. The Frobenius norm of A is calculated as the square root of the sum of the squares of all the entries in A:

$$\|A\|_F = \sqrt{1^2 + 2^2 + 3^2 + 4^2} = \sqrt{1 + 4 + 9 + 16} = \sqrt{30}.$$

Alternatively, using the trace formula, we compute $A^T A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 10 & 14 \\ 14 & 20 \end{bmatrix}$, and $tr(A^T A) = 10 + 20 = 30$. Thus, $\|A\|_F = \sqrt{\text{trace}(A^T A)} = \sqrt{30}$.

2 THEORETICAL BACKGROUND

2.1 Theory of Singular Value Decomposition

We first examine the mathematical foundations and factorization properties behind SVD. The SVD of a matrix is defined as follows:

Definition 2.1. The **Singular Value Decomposition (SVD)** is a matrix factorization technique that can be applied to any matrix A, regardless of size or shape. [2] The SVD of a matrix $A \in \mathbb{R}^{m \times n}$ expresses A as the product of three matrices $A = U\Sigma V^T$ (see Figure 1 for illustration), where:

- $U \in \mathbb{R}^{m \times m}$ is an orthogonal matrix. The orthonormal columns u_i of U are called the *left singular vectors* of A, and they correspond to the eigenvectors of AA^T.
- $V \in \mathbb{R}^{n \times n}$ is an orthogonal matrix. The orthonormal columns v_i of V are called the *right singular vectors* of A, and they correspond to the eigenvectors of $A^T A$.
- $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix (with nonnegative real numbers on the diagonal) whose diagonal entries are the *singular values* of A, denoted by $\sigma_1, \sigma_2, \ldots, \sigma_r$. These singular values are ordered as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and are the square roots of the eigenvalues of $A^T A$. If $m > n$, Σ contains n singular values followed by $m - n$ zeros. Similarly, if $m < n$, Σ contains m singular values followed by $n - m$ zeros.

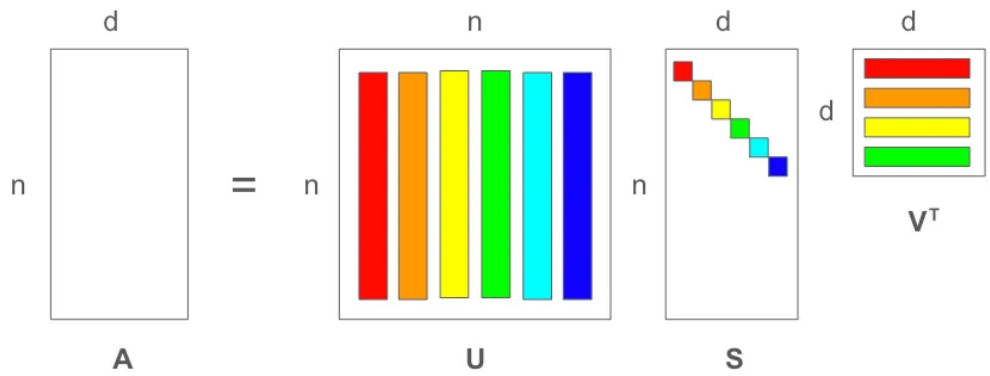


FIGURE 1. Visual representation of SVD for a matrix A. Source: [7].

Example 2.2. Consider the matrix $A = \begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix}$. We will compute the SVD of this matrix. (If interested, Ernstberger and Compton have provided a more detailed, step-by-step guide on computing the SVD of a matrix in [3].)

First, we calculate $A^T A = \begin{bmatrix} 4 & -3 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix} = \begin{bmatrix} 25 & 7 \\ 7 & 25 \end{bmatrix}$. The eigenvectors of $A^T A$ will give us the right singular vectors $\mathbf{v}_i$, and the eigenvalues will give us the squared singular values σ_i^2. Two orthogonal eigenvectors of $A^T A$ are $\mathbf{v}_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$ and $\mathbf{v}_2 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$. These eigenvectors correspond to the eigenvalues $\sigma_1^2 = 32$ and $\sigma_2^2 = 18$, giving us the singular values: $\sigma_1 = 4\sqrt{2}$, $\sigma_2 = 3\sqrt{2}$. Thus, the diagonal matrix Σ is $\Sigma = \begin{bmatrix} 4\sqrt{2} & 0 \\ 0 & 3\sqrt{2} \end{bmatrix}$. Next, we calculate $AA^T = \begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix} \begin{bmatrix} 4 & -3 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 32 & 0 \\ 0 & 18 \end{bmatrix}$. Since AA^T is diagonal, the eigenvectors are the standard basis vectors: $\mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathbf{u}_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$. Thus, the matrix U is $U = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. Finally, the SVD of A is:

$$A = U\Sigma V^T = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 4\sqrt{2} & 0 \\ 0 & 3\sqrt{2} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}.$$

We can think of A as a linear transformation that takes a vector $\mathbf{v}_1$ in its row space to a vector $\mathbf{u}_1 = A\mathbf{v}_1$ in its column space. The SVD arises from finding an orthogonal basis for the row space, which gets transformed into an orthogonal basis for the column space. Specifically, $A\mathbf{v}_i = \sigma_i \mathbf{u}_i$, where σ_i are the singular values. It is straightforward to find an orthogonal basis for the row space using methods like the Gram-Schmidt process. Then the SVD becomes powerful: it ensures that the transformation A maps orthonormal vectors in the row space to orthonormal vectors (scaled by σ_i) in the column space. The transformation A effectively zeros out components of vectors in the null spaces, which is one of the key features of the singular value decomposition.

2.2 Low-Rank Approximation

A significant application of SVD is in constructing low-rank approximations of matrices, which are especially useful in data compression, noise reduction, and feature extraction. The SVD naturally decomposes a matrix $A \in \mathbb{R}^{m\times n}$ into a sum of rank-one matrices: $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$, where $r = \text{rank}(A)$, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ are the singular values, with corresponding left singular vectors u_i and right singular vectors v_i. These singular values are arranged in descending order, making $\sigma_1 u_1 v_1^T$ the most significant component of A, followed by $\sigma_2 u_2 v_2^T$, and so on [9].

To approximate A with a matrix of rank $k < r$, we **truncate** the SVD, which means keeping only the top k singular values and their corresponding singular vectors while discarding the smaller singular values and the associated vectors. The resulting low-rank approximation is given by:

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T = U_k \Sigma_k V_k^T,$$

where $U_k \in \mathbb{R}^{m\times k}$, $\Sigma_k \in \mathbb{R}^{k\times k}$, and $V_k \in \mathbb{R}^{n\times k}$ are the truncated matrices. This approximation minimizes the Frobenius norm of the error $\|A - A_k\|_F$, as stated in the Eckart-Young theorem:

Theorem 2.3 (Eckart-Young)**.** *Let A_k be the rank-k approximation of A obtained via SVD truncation. Then A_k is the best approximation of A in the least-squares sense:*

$$A_k = \arg \min_{B,\, rank(B)=k} \|A - B\|_F,$$

where:

- *B is a candidate matrix of rank k used in the optimization problem.*
- *$\|A - B\|_F$: The Frobenius norm of the difference between A and B which measures the total squared error between A and B.*

This theorem guarantees that the truncated SVD provides the best rank-k approximation to the original matrix in a least-squares sense. Thus, the reconstructed image using the truncated SVD is the closest possible representation to the original image, given the constraint of using only k singular values and vectors. In practical applications, the choice of k depends on the desired trade-off between approximation accuracy and data reduction. As k increases, the approximation A_k becomes closer to A, but the storage requirements also increase. [9].

2.3 SVD in Image Compression

Image compression is a crucial application of SVD in digital image processing. By utilizing the properties of SVD and low-rank approximation, SVD enables efficient storage and transmission. This section explores the use of SVD for image compression and outlines the process.

A digital image can be represented as a matrix, where each entry corresponds to the intensity value of a pixel. For color images, each pixel is defined by three intensity values—red, green, and blue—each scaled between 0 and 255. In the case of grayscale images, the matrix $A \in \mathbb{R}^{m\times n}$ represents the intensity of each pixel, where A_{ij} denotes the intensity at pixel (i, j). These values range from 0 (pure black) to 255 (pure white) [5].

Image compression techniques can be broadly classified into two categories: lossless compression and lossy compression. SVD-based image compression is a lossy compression technique. The goal of image compression is to reduce the amount of data required to represent an image with little to no perceptible loss in quality, which is achieved by eliminating redundancies in the data. These redundancies can be broadly categorized into three types: (1) *Coding redundancy*, which occurs when the encoding of image data is not optimal; (2) *Interpixel redundancy*, which arises from correlations between neighboring pixels; (3) *Psychovisual redundancy*, which refers to image details that are not perceptible to the human visual system. [2]

When an image is transformed using SVD, the data takes the form of a decomposition where the first singular value contains most of the image's information. To compress the image, we truncate the SVD to retain only the largest r singular values, resulting in a low-rank approximation:

$$A \approx A_r = U_r \Sigma_r V_r^T,$$

where U_r, Σ_r, and V_r contain only the first r columns of U, the first r singular values, and the first r rows of V^T, respectively. The storage required for the low-rank approximation A_r is reduced to $k(m+n+1)$, where k is the rank of the approximation. Eventually, we can summarize the image compression process using SVD in the following steps:

(1) Convert the input image into a matrix A.

(2) Transform image data from integer to double data type for computation.

(3) Decompose A using SVD into matrices U, Σ, and V^T, where $A = U\Sigma V^T$.

(4) Choose the rank k, which determines how many singular values to retain based on the desired balance between compression ratio and image quality.

(5) Truncate the SVD by retaining only the top k singular values and their corresponding singular vectors: $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$.

(6) Reconstruct the compressed image using U_k, Σ_k, and V_k^T.

(7) Convert the compressed matrix back to integer values for storage or display.

Through this process, SVD makes image compression efficient by keeping only the essential data, reducing storage needs while maintaining good visual quality.

2.4 Image Compression Measures

To evaluate the performance of SVD-based image compression, we will use two metrics and visually inspect the images for psychovisual quality.

- Compression Ratio (CR): CR measures the efficiency of the image compression process. It is defined as the ratio of the original image size to the size of the compressed image. The CR can be computed using the formula: $\mathrm{CR} = \frac{m \cdot n}{k \cdot (m+n+1)}$, where m and n are the dimensions of the original image, and k is the number of singular values retained in the compression. A higher CR indicates better compression efficiency.
- Mean Squared Error (MSE): MSE quantifies the difference between the original and the compressed image, and therefore it measures the quality of the compressed image. It is calculated as the average squared difference between the corresponding pixel values of the original and compressed images: $\mathrm{MSE} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij} - A'_{ij})^2$, where A_{ij}

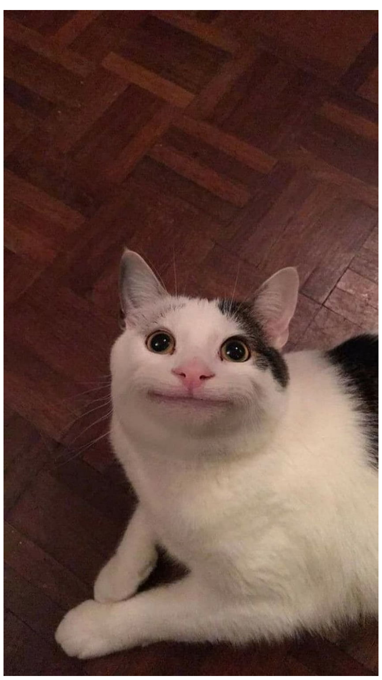
represents the pixel value in the original image and A'_{ij} is the corresponding pixel value in the compressed image. A lower MSE indicates better image quality.

While a higher CR generally indicates more efficient storage, it may come at the cost of a higher MSE. Therefore, it is important to find an appropriate balance between CR and MSE. In the next section, we will experiment with different values of k to see if we can find the optimal trade-off between compression efficiency and image quality in the image compression process.

3 EXPERIMENTS AND RESULTS

3.1 Initial Observations

Before getting into the core experiments of this project, we will perform an initial observation and analysis of the input image, `cat.jpg`. This preliminary step aims to understand the color composition and intensity distribution within the image and preparing us for subsequent experiments. We have the original image as shown in Figure 2a and some of its attributes in Figure 2b.



(A) cat.jpg

Name	Size	Bytes	Class	Attributes
cat	1510x850x3	3850500	uint8	

(B) Name, Size, Bytes, and Class of the original image.

FIGURE 2. The image cat.jpg features Harold, known online as Ollie or "the Polite Cat". For the purposes of this project, we will refer to him as "cat".

The original image was decomposed into red, green, and blue (R, G, B) channels. This separation allows for a more detailed examination of the intensity distribution of each color within the image. We visualize the grayscale representation of each channel in Figure 3.

- The red channel appears brightest overall, especially in the lighter areas of the cat's fur. The wooden floor and the cat's face, while still visible, do not appear as bright as the fur.
- The green channel's intensity is more balanced than the red, with the cat's fur appearing bright but less intense and the floor moderately gray, reflecting subtler green intensity.
- The blue channel is the darkest, with the floor appearing significantly darker. We say the blue pixels contribute the least to the image.

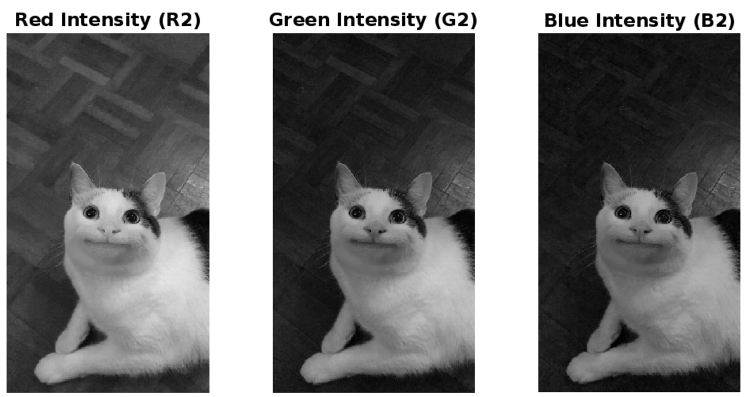


FIGURE 3. Grayscale intensity of individual color channels (R2, G2, B2).

In summary, the red channel dominates the brighter regions of the image, such as the cat's fur, followed by a more moderate green presence, while the blue channel has a noticeably lower intensity across the entire image. We can confirm this color distribution by looking at Figure 4.

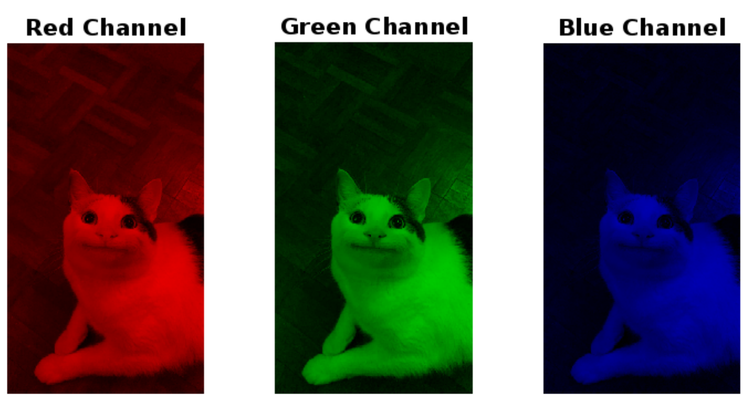


FIGURE 4. Decomposition of RGB Color Channels.

3.2 Experiment 1: Image Compression with Varying Rank

This experiment examines the effect of varying the number of retained singular values k on both the visual quality of a reconstructed image and the corresponding compression metrics. Our goal is to illustrate the trade-off between achieving high CRs and maintaining acceptable image fidelity. We will demonstrate how different values of k influence the reconstructed image quality and the resulting compression performance in terms of CR, MSE, and storage requirements.

3.2.1 Methodology

We applied SVD-based compression to `cat.jpg`, reconstructing it using a set of singular values $k \in \{2, 3, \ldots, 10, 15, 20, 25, 30\}$. For each k, we generated a reconstructed image and evaluated CR, MSE, and the approximate storage space after compression. The complete code for this experiment is accessible here.

3.2.2 Results and Analysis

Figures 5 and 6 show the reconstructed images for different k values. Figure 7 summarizes the CR, MSE as k varies. To view the complete metric table, including a column for the compressed storage size, you can execute the code for Experiment 1 linked above.

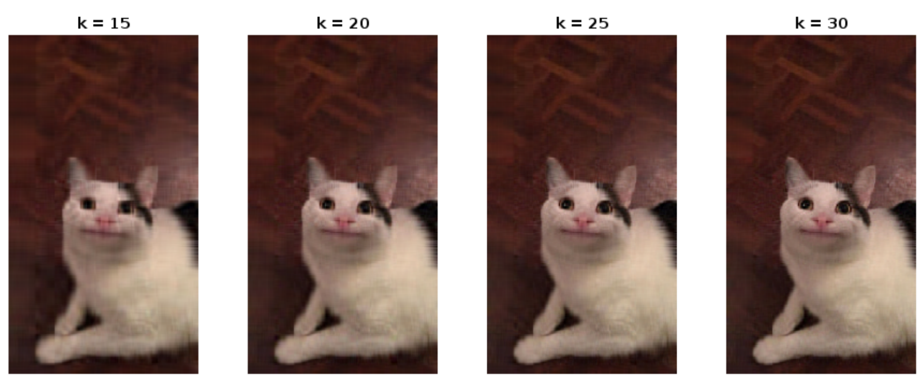


Figure 5. Reconstructed images for $k = 15, 20, 25$, and 30.

For very small $k = 2$, the image is heavily blurred, and features are faintly visible due to the high CR but poor image quality resulting from a large MSE. By $k = 5$, basic features of the image become recognizable, and as k increases to 8, the cat is clearly identifiable. At $k = 15$, finer details, such as the cat's smile, become visible, with significant improvements in image quality as the MSE decreases. Acceptable image quality is achieved around $k = 25$, though this improvement comes at the cost of a lower CR and increased storage requirements. By $k = 30$, the reconstructed image is nearly identical to the original, with a CR of 18.121 and an MSE smaller than 0.0004.

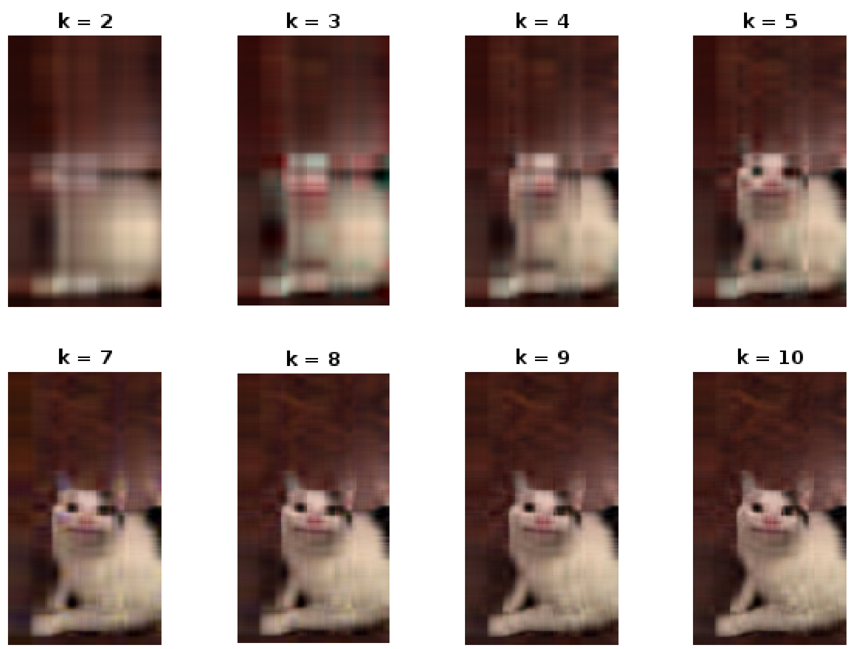


FIGURE 6. Reconstructed images for ranks $k = 2$ to $k = 10$.

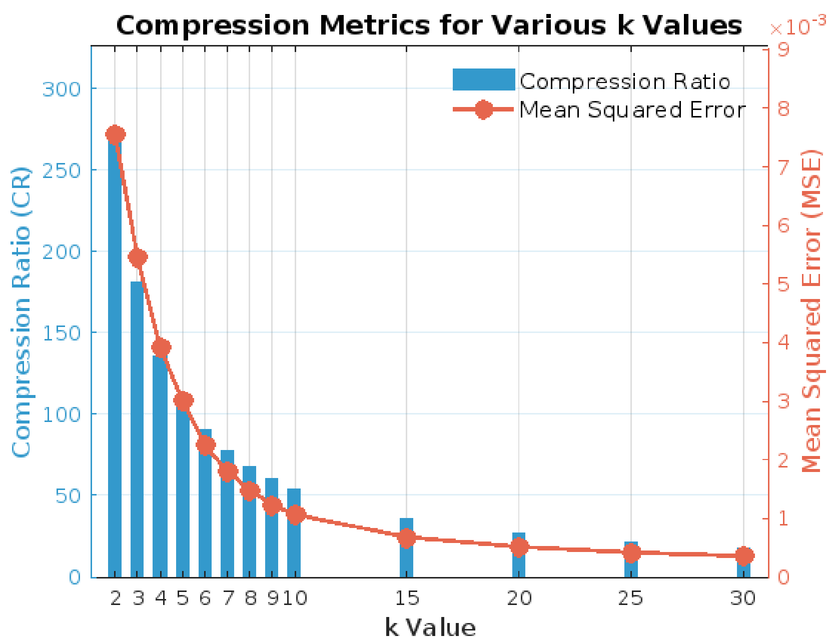


FIGURE 7. Compression metrics for various k values.

Experiment 1 demonstrates that varying the number of retained singular values provides a simple way to balance compression quality and storage. Smaller k values yield higher CRs and reduce file size but sacrifice sharpness and clarity. In contrast, larger k values improve image details and reduce MSE at the expense of storage and lower CRs.

3.3 Experiment 2: Image Compression by Removing Larger Singular Values

Inspired by the work of [3], this experiment investigates the effect of removing larger singular values on image reconstruction. While the method in [3] focuses on grayscale images, we extend it to full-color images by separately applying the SVD to the red, green, and blue channels and subsequently reconstructing the image. We inherently understand that this approach would not effectively preserve image quality, but curious about the extreme appearance of a full-color image without the largest singular values and how it differs from a grayscale reconstruction. Our objective is to analyze the significance of the largest singular values and their role in preserving image quality. We assess the visual impact and examine how the MSE behaves as these values are removed.

3.3.1 Methodology

We reconstructed the image by removing the k largest singular values and retaining the smaller ones. It is important to note that, in this experiment, k represents the number of first singular values removed, not the rank retained. Reconstructed images were generated for $k \in \{1, 2, 3, 4, 5, 6, 8, 10, 15, 20, 30, 50\}$. To evaluate the impact of singular value removal, we quantified the compression error using MSE between the original image and the reconstructed image. The full code for this experiment is available here.

3.3.2 Results and Analysis

Figure 8 shows the reconstructed images as progressively larger singular values are removed. At $k = 1$, the image barely retains the basic structure of the cat, with the lower part of the face, body, and background appearing mostly black. As k increases to 2, 3, and 4, the image loses structural information and appears even darker. By $k = 3$, the cat's features become unrecognizable. At $k = 10$, most of the visual information is lost, leaving only dark, blurry remnants. Beyond $k = 20$, the image is almost entirely black.

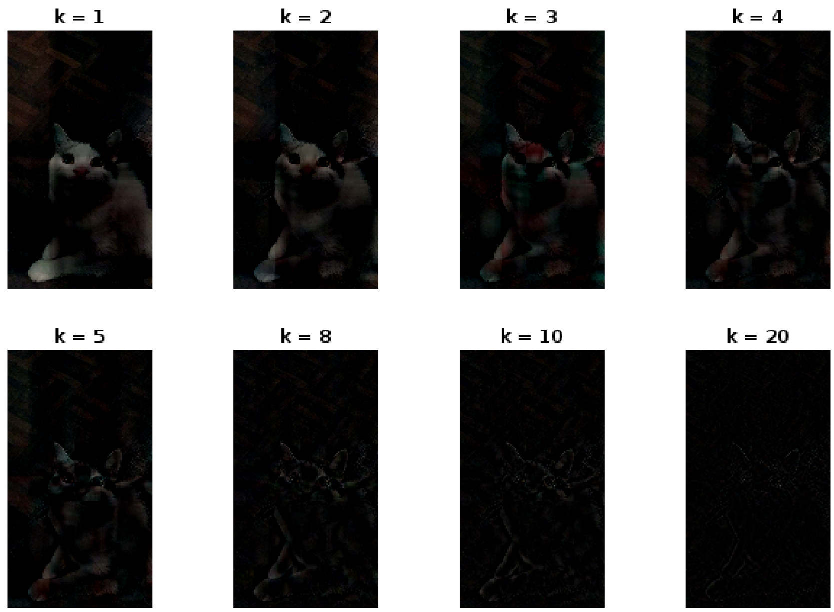


FIGURE 8. Reconstructed images after removing largest singular values k.

Figure 9 illustrates the relationship between the number of singular values removed and the MSE. The graph demonstrates a rapid increase in MSE as k increases from 1 to 10, highlighting the significant contribution of the largest singular values to the image. After approximately $k = 10$, the rate of increase in MSE begins to plateau, suggesting that the smallest singular values contribute very little to the overall reconstruction.

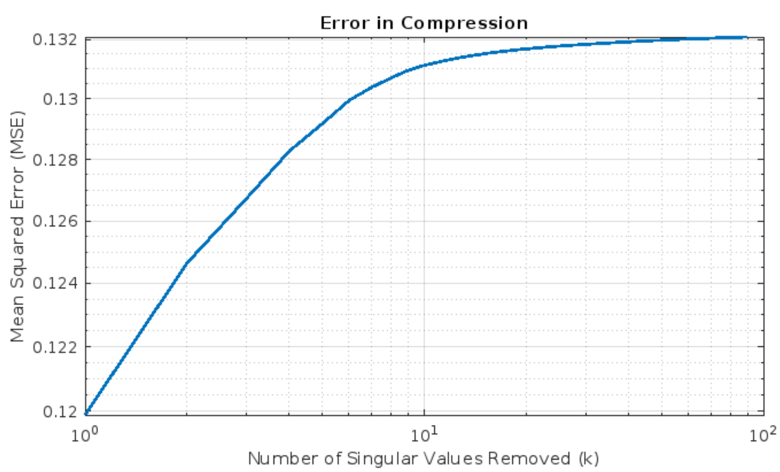


FIGURE 9. Relationship between (k) and MSE.

To further illustrate the dominance of the largest singular values, Figure 10 compares the original image and two cases:

- Only the largest singular value: The reconstructed image retains dominant patterns but lacks fine details, such as edges and textures.
- All but the largest singular value: The image retains some edge structures but loses most color intensity.

FIGURE 10. Comparison of reconstructions.

Overall, when comparing these findings with the grayscale-focused approach by Compton et al. [3], our results confirm that the largest singular values play a crucial role in maintaining image quality. Conversely, smaller singular values have minimal impact on image reconstruction. Applying SVD separately to the red, green, and blue channels of full-color images reveals a similar reliance on their largest singular values. Removing these major singular values in color images results in more noticeable degradation than in grayscale images.

3.4 Experiment 3: Exploring A Different Color Space

This experiment investigates the impact of performing SVD-based compression in different color spaces. While earlier tests focused on compressing images directly in the RGB domain, here we examine whether converting an image into the YCbCr color space before applying SVD leads to more efficient compression at similar perceptual quality levels.

Our main goal is to understand how decomposing an image into luminance (Y) and chrominance (Cb, Cr) components affects the performance of SVD-based compression. The experiment also compares the results with those obtained by compressing directly in the RGB space.

3.4.1 Methodology

We begin by loading the original image and converting it from RGB to YCbCr. Next, we compute the SVD for each channel (Y, Cb, and Cr) and truncate the singular values based on the predefined rank (k) parameters.

- Case 1 (kY = kC = 5): To compare to RGB compression with the same k.
- Case 2 (kY = 3, kC = 7): To test the sensitivity of image quality to reductions in luminance.
- Case 3 (kY = 7, kC = 3): Given human vision's higher sensitivity to brightness contrasts, we expect this scenario to maintain essential structural details of the image.
- Case 4 (kY = 5, kC = 3): This case combines equal luminance rank with lower chrominance rank to test the trade-off between CR and perceptual quality.

After reconstruction, we convert the approximated YCbCr image back to RGB for visualization and compute the CR and MSE for each configuration. For benchmarking, we also compare these results to those obtained when compressing solely in the RGB domain. The complete code for this experiment can be accessed here.

3.4.2 Results and Analysis

Figure 11 displays the final reconstructed images under different conditions. The RGB (k=5) and Case 1 images exhibit a perceptually similar quality.

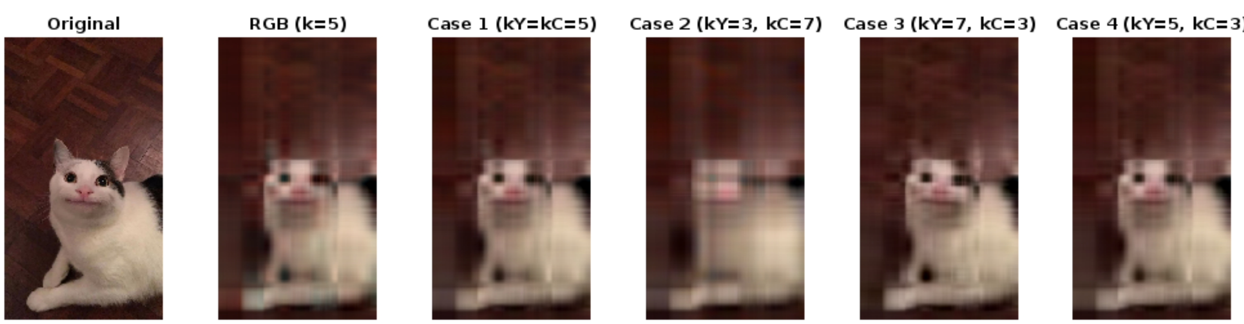


FIGURE 11. Final reconstructed images for various conditions.

As shown in Table 12, YCbCr compression achieves the same CR as RGB (108.73) with a slightly higher MSE (a difference of 0.1588×10^{-3}). This minor increase in MSE can be attributed to additional steps involved in the YCbCr process, such as converting the image between color spaces, which might introduce subtle data loss. However, this difference is negligible and becomes even less noticeable as k increases. For instance, at $k = 20$, the difference in MSE is almost 0 (1.1358×10^{-6}). This trend indicates that compressing directly in the RGB or YCbCr color space with equal ranks for RGB, luminance and chrominance channels yields comparable results.

Comparison Table: CR and MSE for YCbCr and RGB

k	CR_YCbCr	CR_RGB	MSE_YCbCr	MSE_RGB	MSE_YCbCr - MSE_RGB
5	108.73	108.73	0.0031818	0.003023	0.00015881
10	54.363	54.363	0.0010732	0.0010707	2.574e-06
20	27.181	27.181	0.00051762	0.00051649	1.1358e-06

Figure 12. CR and MSE for YCbCr and RGB (case 1) at various ranks k.

We generated chart 13 to easily compare the CR and MSE across the four YCbCr cases. To view the detailed metric table underlying this chart, execute the code provided for Experiment 3 linked above. To present both metrics on a consistent scale where ”higher is better” in this visualization, MSE was inverted (1/MSE) since lower MSE indicates better reconstruction quality.

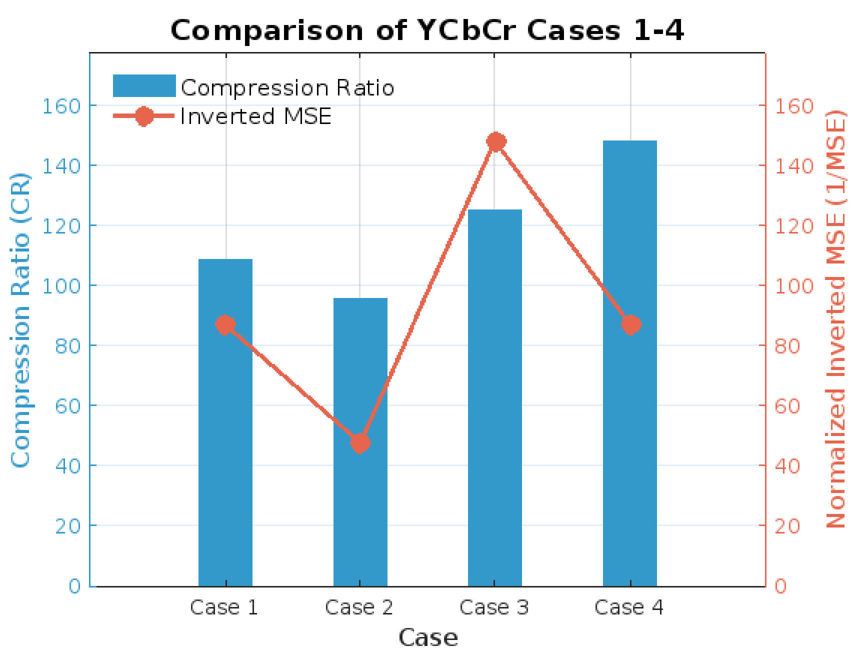


Figure 13. Comparison of result metrics using YCbCr for cases 1-4.

From the graph, table, and images above, we can derive the following insights:

- Despite having the highest k_C, Case 2 performs worst in both CR and MSE. Its final compressed image lacks a lot of detail, making it almost impossible to recognize the cat.

- Case 1 vs Case 3: Case 3 has slightly better perceptual quality than Case 1. In terms of metrics, Case 3 achieves a higher CR by about 15% and a lower MSE despite having lower k_C thanks to the higher k_Y. Therefore, we can say that luminance (Y) plays a more crucial role in perceptual quality than chrominance (Cb and Cr).
- Case 1 vs Case 4: Both cases exhibit comparable perceptual quality. However, in terms of metrics, Case 4 achieves a much higher CR, showing a 36.3% improvement over Case 1, with a negligible increase in MSE. This result suggests that reducing chrominance ranks significantly improves CR with little impact on perceptual quality.
- Case 3 vs Case 4: Case 3 achieves the best MSE and a very good CR (125.45), while Case 4 achieves the highest CR among all (148.26) with a slightly higher MSE.

Final Choice: If the primary goal is to maximize data compression (highest CR with a slight trade-off in quality), Case 4 is the most suitable option. If image quality is the top priority, Case 3 offers the best balance of metrics and perceptual fidelity among the four. Both cases are performed in the YCbCr color space and outperform compression conducted directly in the RGB domain.

Experiment 3 demonstrates that SVD-based compression in the YCbCr color space is more flexible and efficient than RGB, especially when varying channel ranks. YCbCr achieves similar image quality with only slightly higher, negligible MSE. The results indicate that luminance (Y) significantly influences perceptual quality more than chrominance (Cb, Cr). By allocating higher ranks to luminance and reducing chrominance ranks, we attain lower MSE and better CRs. We can maximize CR by significantly lowering chrominance ranks with little quality loss. Emphasizing chrominance offers little perceptual benefit and reduces efficiency.

3.5 Experiment 4: Impact of Noise on SVD Compression

Noise is unwanted information that can reduce image quality and may appear during various stages of image handling. It can result from image capture devices, environmental conditions, sensors, electronic interferences, and errors in analog-to-digital conversion and bit transmission. [1]

In the context of image compression, noise can significantly affect the quality of the reconstructed image, especially when employing techniques like SVD. This experiment aims to investigate the sensitivity of SVD-based image compression to varying levels of Gaussian noise. Specifically, we seek to understand how different noise intensities impact both the visual quality of the reconstructed images and the efficiency of compression. We intuitively hypothesize that higher noise levels will degrade reconstruction quality, requiring higher ranks (k) in the SVD to achieve acceptable image fidelity, thus reducing the CR.

3.5.1 Methodology

The code for this experiment can be accessed here. The following steps were conducted:

(1) Gaussian noise was introduced into the original image at three standard deviation levels: Low Noise with $\sigma = 0.05$, Medium Noise with $\sigma = 0.10$, and High Noise with $\sigma = 0.20$.
(2) SVD was applied to both the original and the noisy images. The images were then reconstructed using different ranks (k).
(3) A fixed rank approach was first employed, using $k = 15$, to observe the baseline reconstruction quality across different noise levels.
(4) Subsequently, the rank k was varied for each noise level to determine the optimal balance between minimizing error and maintaining a high CR.
(5) We computed and analyzed CR and MSE for the necessary cases.

3.5.2 Results and Analysis

Figure 14 displays the original and noisy images at low, medium, and high noise levels. As expected, increasing the noise level progressively degrades the visual quality of the images.

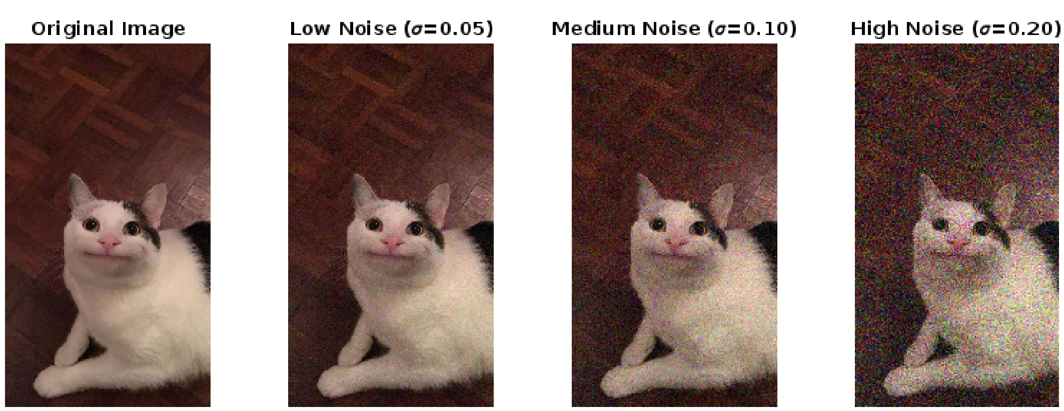


FIGURE 14. Original image and noisy images at low, medium, and high noise levels.

Applying SVD with a fixed rank $k = 15$, as shown in Figure 15, we see that the reconstruction quality slightly degrades as the noise level increases.

Figure 15. Reconstructed images using a fixed rank $k = 15$.

Figures 16 and 17 depict the MSE as a function of k for varying noise levels. For the noise-free case, the MSE steadily decreases as k increases. For all noise levels in noisy cases, the MSE initially decreases with increasing k, reaching a minimum (the "elbow"), before gradually increasing.

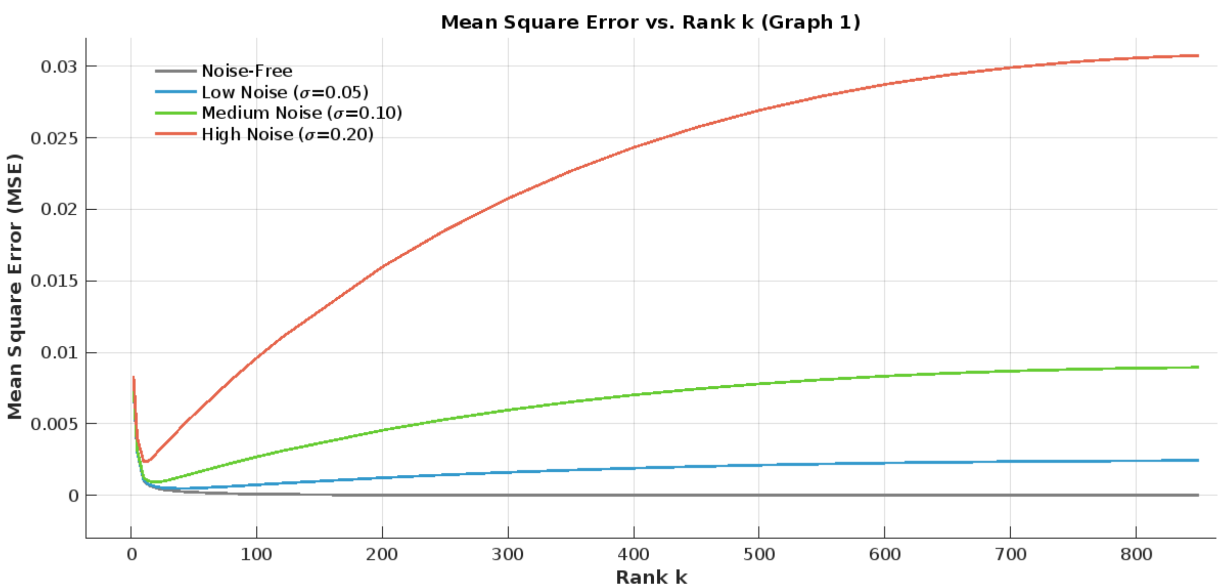


Figure 16. MSE for noise levels and k-values up to 850.

The MSE behavior can be explained by the inherent properties of the SVD.The MSE behavior can be explained by the inherent properties of the SVD. As k increases, lower-rank components (which generally capture image details) are retained while higher-rank components (which often correspond to noise) are discarded. Initially, this filtering process helps suppress noise, leading to a decrease in MSE. However, as k increases further, the decomposition begins to include more noise components, which reduces reconstruction quality. The “elbow” in the MSE curve marks the point where this balance between noise suppression and detail retention is optimal. As noise intensity increases, higher ranks bring in more noise, which moves the elbow to lower k values.

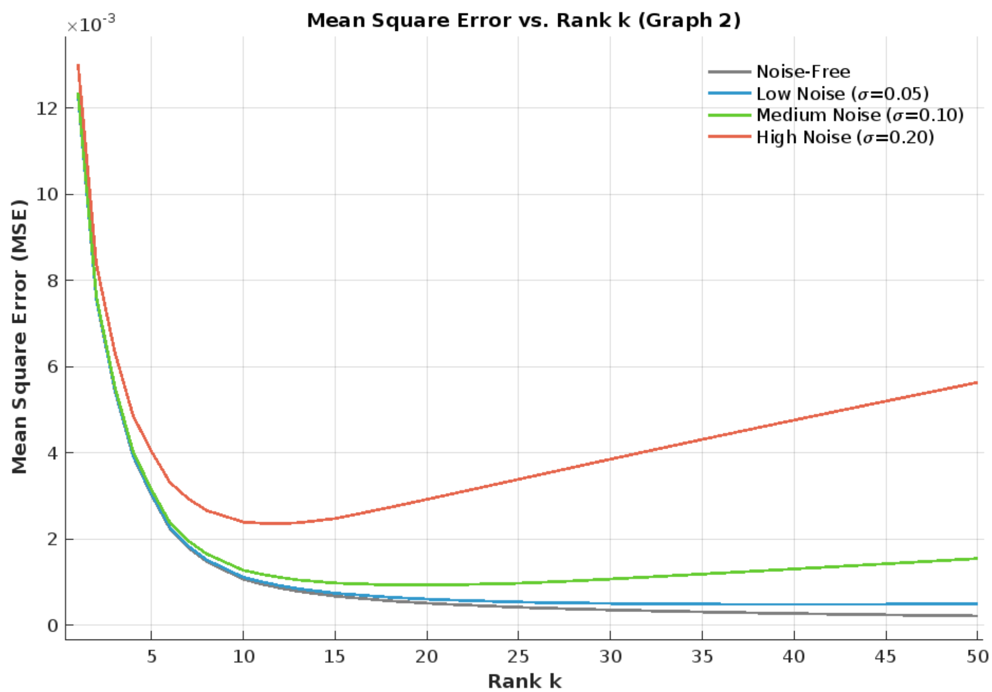


Figure 17. MSE for noise levels and k-values up to 50.

The location of the MSE elbow and the trends in the curves are influenced by the noise intensity:

- High Noise: The MSE decreases steeply until $k \approx 12$, after which it begins to increase.
- Medium Noise: The MSE decreases until $k \approx 20$ before increasing. The elbow shifts to a slightly higher rank compared to high noise.
- Low Noise: The MSE decreases until $k \approx 40$, after which a slow upward trend begins.

Key Observations

(1) The MSE decrease rate (before the elbow) is faster than the increase rate (after the elbow) for all noise levels.
(2) Lower noise levels exhibit a slower increase in MSE after the elbow.
(3) As k increases significantly, all MSE curves start to flatten.
(4) For the same k value, lower noise levels consistently achieve lower MSE.

Even when MSE values are similar for different noise levels at higher k, there are notable differences in the perceived visual quality of the reconstructed images. For instance, at $\sigma = 0.20$, the reconstruction at $k = 5$ is blurry and lacks the cat's structure because it fails to capture enough components of the original image. Conversely, at $k = 32$, the reconstruction reveals a clearer structure of the cat but also introduces more noise, resulting in a grainy texture with color errors.

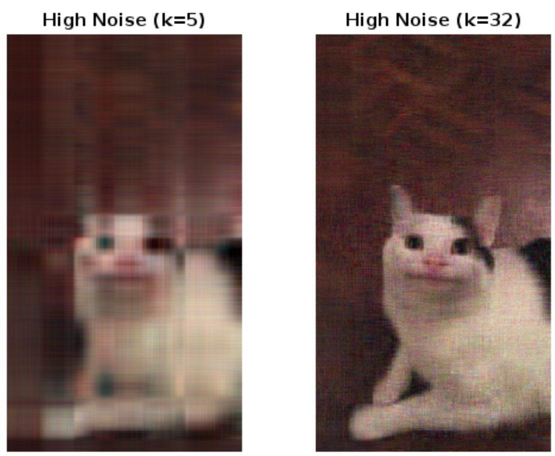


FIGURE 18. Comparison of high-noise reconstructions at $k = 5$ and $k = 32$.

Table 19 shows that the CR remains identical across all noise levels for the same k. This behavior shows that the CR depends solely on the number of singular values retained and the size of the matrix, both of which are independent of the noise level.

Compression Ratios for Selected k Values:

	Original	Low_Noise	Medium_Noise	High_Noise
k=1	543.63	543.63	543.63	543.63
k=10	54.363	54.363	54.363	54.363
k=50	10.873	10.873	10.873	10.873
k=200	2.7181	2.7181	2.7181	2.7181
k=400	1.3591	1.3591	1.3591	1.3591
k=850	0.63956	0.63956	0.63956	0.63956

FIGURE 19. CRs of reconstructed images at 4 noise levels for different k values.

Experiment 4 demonstrates the significant impact of noise on SVD-based image compression. The location of the MSE elbow provides a useful indicator for selecting an optimal rank k that balances noise suppression and image detail preservation. Higher noise levels shift the elbow to lower k values, indicating that fewer singular values are needed to suppress noise effectively. The

CR remains unaffected by noise levels. While MSE can measure the overall error, the perceived visual quality of the image may still vary depending on the rank k used. Therefore, noise levels should be considered when selecting the rank k for image compression using SVD to achieve both efficient compression and high reconstruction quality.

4 CONCLUSION

This project explored the use of Singular Value Decomposition for image compression, analyzing its effectiveness and trade-offs in various scenarios. Throughout the experiments, we observed how SVD-based techniques can significantly reduce image file sizes while preserving important visual features, though with some loss of detail depending on the number of singular values retained.

In Experiment 1, we demonstrated the direct relationship between the number of retained singular values and the compression performance. A smaller number of singular values led to higher CRs but sacrificed visual clarity, especially in finer image details and vice versa.

Experiment 2, which focused on removing the largest singular values, highlighted the critical role of these values in maintaining the integrity of the image. Removing even a few of the largest singular values resulted in substantial loss of structure.

In Experiment 3, we shifted focus to exploring compression in different color spaces. Compressing images in the YCbCr color space, as opposed to the RGB space, allowed for more efficient compression with comparable perceptual quality. This experiment showed that luminance (Y) information is more crucial for maintaining image quality than chrominance (Cb, Cr), and reducing chrominance ranks significantly improved CRs with minimal visual degradation.

The fourth experiment evaluated the sensitivity of SVD-based compression to noise, revealing that higher noise levels required more singular values to maintain acceptable image quality. The introduction of Gaussian noise led to a shift in the optimal rank k for image reconstruction, and the MSE curves provided valuable insights into balancing noise suppression and detail retention.

Throughout the project, I have achieved a deeper understanding of linear algebra concepts, especially matrix decompositions. If I further investigate this topic, I would explore the geometrical meaning of SVD to develop more intuitive compression strategies and apply SVD to image denoising and feature extraction; Investigate SVD-based compression in other color spaces like HSV or Lab; Examine how image alignment affects SVD performance, particularly when using economy-size decomposition, and implement alignment techniques to improve compression outcomes.

In conclusion, this individual project was an extremely rewarding experience for me. The topics explored, drawn from a wide range of research papers, books, and websites I have studied by myself over the past three weeks, were concepts I had encountered in theory but never had the opportunity to experiment with firsthand. Engaging with these ideas in a practical context has deepened my understanding and fueled my enthusiasm for further exploration in the field.

References

[1] Ajay Kumar Boyat and Brijendra Kumar Joshi. A review paper: noise models in digital image processing. *arXiv preprint arXiv:1505.03489*, 2015.

[2] Steven L. Brunton and J. Nathan Kutz. *Singular Value Decomposition (SVD)*, page 3–46. Cambridge University Press, 2019.

[3] Elizabeth A Compton and Stacey L Ernstberger. Singular value decomposition: Applications to image processing. *Citations Journal of Undergraduate Research*, 17, 2020.

[4] William Ford. Chapter 6 - orthogonal vectors and matrices. In William Ford, editor, *Numerical Linear Algebra with Applications*, pages 103–118. Academic Press, Boston, 2015.

[5] Connor Kuhn. Svd and dct image compression. 2016.

[6] Jean-Baptiste Lorent. Extending 8k over a single, cost-effective wire at the speed of light with tico lightweight compression. 2018.

[7] Yuki Shizuya. The mathematical and geographic understanding of singular value decomposition(svd), Feb 2024.

[8] Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.

[9] Gilbert Strang. *Introduction to linear algebra*. SIAM, 2022.

Email address: `dinh224k@mtholyoke.edu`

Department of Mathematics and Statistics, Mount Holyoke College, South Hadley, MA 01075