# Image Compression using Singular Value Decomposition (SVD)

Aroshish

August 6, 2024

## 1 Introduction

Singular Value Decomposition (SVD) is a mathematical technique used to decompose a matrix into three constituent matrices: $U$, $\Sigma$, and $V^T$. This decomposition is particularly useful in various applications, including image compression, where it allows for efficient data reduction while preserving essential information. This report details the principles of SVD, the methodology used for image compression, the results of the experiments, and the performance of an interactive tool developed using Streamlit.

## 2 Understanding SVD

SVD decomposes an $m \times n$ matrix $A$ into three matrices:

$$A = U\Sigma V^T$$

where:

- $U$ is an $m \times m$ orthogonal matrix whose columns are the left singular vectors of $A$.
- $\Sigma$ is an $m \times n$ diagonal matrix with singular values (non-negative) on the diagonal, arranged in descending order.
- $V^T$ is an $n \times n$ orthogonal matrix whose columns are the right singular vectors of $A$.

The singular values in $\Sigma$ represent the importance of each corresponding singular vector. We can approximate $A$ with reduced complexity by retaining only the top $k$ singular values of the vectors.

# 3 Image Compression

## 3.1 Image Representation

An image is represented as a matrix of pixel values. For RGB images, three matrices represent the Red, Green, and Blue channels.

## 3.2 SVD Decomposition and Reconstruction

The RGB channels are considered separately and decomposed using SVD. The top k singular values are then retained and the channels are then reconstructed. Finally, they are merged together into a single matrix.

# 4 Experiments with Image Compression

## 4.1 Experimental Setup

- **Original Image**: High-quality RGB images were used.
- **Values of $k$**: Different values of $k$ (50, 100, 200, 500 and 1000) were tested.
- **Tools**: Python, NumPy for SVD computation, PIL for image handling, and Streamlit for the interactive tool.

## 4.2 Results

- **Low $k$ Values (e.g. 50)**: The compressed images show significant loss of detail. Important features and fine textures are often blurred or lost, resulting in loss of image quality.
- **Moderate $k$ Values (100, 200)**: As k increases, more details are preserved. The images appear clearer with fewer artifacts, but some loss of fine detail are still present.
- **High $k$ Values (500, 1000)**: The images closely resemble the original with minimal perceptible loss of quality. The reconstruction retains most of the details and textures, showing a clear trade-off between compression and fidelity.

# 5 Trade-offs

- **Compression Ratio**:Lower k values provide higher compression ratios but at the cost of image quality. Higher k values reduce compression but retain more detail.

- **Image Quality**: The visual quality of the image improves as k increases, with a more accurate representation of the original image. However, the file size also increases with k.

# 6 Performance of the Interactive Tool

An interactive tool was developed using Streamlit to allow users to upload images, choose compression levels, and view the results in real-time.

## 6.1 Features

- **User Interface**: Simple and user-friendly interface for uploading images and selecting the compression level.
- **Real-Time Compression**: Immediate feedback on the compressed image as the user adjusts the $k$ value.
- **Side-by-Side Comparison**: Original and compressed images are displayed side by side for easy comparison.
- **Download Option**: Users can download the compressed image directly from the interface.

## 6.2 Evaluation

- **Usability**: The tool is intuitive and easy to use, with a clear layout and responsive controls.
- **Performance**: Real-time compression and display work efficiently for various image sizes. The use of in-memory buffers and Streamlit's `st.download_button` provides a seamless experience.
- **Flexibility**: Users can experiment with different compression levels and instantly see the effects on image quality and file size.

# 7 Conclusion

SVD is a powerful technique for image compression, offering a balance between compression ratio and image quality. The interactive tool developed using Streamlit provides an effective way for users to explore the effects of SVD-based compression. The user can adjust the compression level depending on how much image quality they are willing to give up. This hackathon demonstrated the usability of SVD as a tool in image processing.