Rabib Husain
Dr. Alex Dillhoff
CSE 3380 Honors Contract

Image Compression Using Singular Value Decomposition (SVD)

Image compression is a crucial aspect in various fields such as telecommunications, medical imaging, and multimedia applications. Efficient compression techniques are essential to reduce storage space, transmission bandwidth, and processing time while preserving the essential information in an image. One powerful method for image compression is Singular Value Decomposition (SVD). SVD is a mathematical technique that breaks down a matrix into three constituent matrices, providing a compact representation of the original data. This essay explores the principles and applications of image compression using SVD.

Singular Value Decomposition (SVD):

The singular value decomposition (SVD) is a fundamental matrix factorization in computational science, forming the basis for various data methods. In the context of this book, the SVD is utilized for tasks such as obtaining low-rank approximations, calculating pseudo-inverses for non-square matrices, and solving systems of equations (Ax = b). Additionally, the SVD plays a crucial role in principal component analysis (PCA), breaking down high-dimensional data into its most statistically descriptive factors.

This technique has broad applications in science and engineering, providing a numerically stable decomposition that guarantees existence. In contrast to the fast Fourier transform (FFT), which is commonly found at the beginning of many engineering texts but operates in idealized settings, the SVD is a more generic and data-driven approach. The book focuses on the SVD as it tailors a basis to specific data, unlike the FFT, which offers a generic basis.

Many real-world complex systems generate data naturally organized in large matrices or arrays. For instance, time-series data from experiments or simulations can be structured in a matrix, with each column representing measurements at a specific time. Similarly, the pixel values in a grayscale image can be stored in a matrix, and these images can be reshaped into column vectors to represent frames in a movie. Remarkably, such data tends to be low-rank, indicating a few dominant patterns that explain the high-dimensional data. The SVD emerges as a numerically robust and efficient method for extracting these dominant patterns from diverse datasets.

The passage introduces the singular value decomposition (SVD) and emphasizes its significance as a foundational technique in data processing. The SVD is showcased through various examples to develop an intuitive understanding of its application. It lays the groundwork for subsequent

techniques explored in the book, such as classification methods, dynamic mode decomposition (DMD), and proper orthogonal decomposition (POD).
The challenge of high dimensionality in processing data from complex systems, encompassing audio, image, or video data, is highlighted. Many natural systems exhibit dominant patterns, characterized by a low-dimensional attractor or manifold. For instance, images, despite being elements of a high-dimensional vector space, often possess compressible information in a lower-dimensional subspace. Similarly, complex fluid systems, like Earth's atmosphere or turbulent wakes, demonstrate low-dimensional structures within high-dimensional state-spaces.

The SVD is introduced as a systematic approach to deriving low-dimensional approximations from high-dimensional data by identifying dominant patterns. This data-driven technique requires no expert knowledge or intuition, offering a numerically stable and hierarchical representation of data based on dominant correlations. Importantly, the SVD is guaranteed to exist for any matrix, unlike the eigendecomposition.

Beyond dimensionality reduction, the SVD finds applications in computing the pseudo-inverse of non-square matrices, solving underdetermined or overdetermined matrix equations ($Ax = b$), and denoising datasets. Additionally, the SVD plays a crucial role in characterizing the input and output geometry of linear maps between vector spaces. These diverse applications provide an intuition for matrices and high-dimensional data, setting the stage for further exploration in the chapter.

The Singular Value Decomposition (SVD) has a rich and extensive history, evolving from early theoretical foundations to contemporary advancements in computational stability and efficiency. A historical review by Stewart [502] provides valuable context and details, highlighting key contributors and milestones.

The early theoretical work of Beltrami and Jordan (1873), Sylvester (1889), Schmidt (1907), and Weyl (1912) is discussed in the historical review. These pioneers laid the groundwork for the understanding and development of the SVD. The review also delves into more recent contributions, particularly the influential computational work of Golub and collaborators [212, 211]. Golub's contributions have significantly shaped the practical aspects of implementing the SVD.

For those interested in delving deeper into the subject, there are numerous excellent chapters on the SVD in modern texts [524, 17, 316]. These chapters provide contemporary perspectives on the SVD, covering both theoretical aspects and practical applications. They contribute to the ongoing dialogue surrounding the SVD's historical evolution and its enduring significance in various fields of study.

The historical journey of the SVD reflects a convergence of theoretical insights and computational innovations, illustrating its enduring importance in the realm of numerical analysis and scientific computing.

SVD is a matrix factorization method that represents a matrix A as the product of three matrices
$A=U\Sigma VT$
where:

- U is an orthogonal matrix representing the left singular vectors
- Σ is a diagonal matrix with singular values on the diagonal,
- VT is the transpose of an orthogonal matrix representing the right singular vectors

Image Compression using SVD:

The key idea behind using SVD for image compression is to retain only the most significant singular values and their corresponding singular vectors. Since the singular values are sorted in descending order, discarding the less significant ones results in a compressed representation of the original image. The reduced number of singular values directly correlates with a reduction in storage requirements.

In the context of image compression, a grayscale image can be conceptualized as a real-valued matrix $\in \times X \in \mathrm{R} n \times m$ where n and m represent the number of pixels in the vertical and horizontal directions, respectively. The choice of representation basis, whether it be pixel-space, Fourier frequency domain, or SVD transform coordinates, plays a crucial role in determining the efficiency of approximations for images.

The fundamental idea is that, based on the chosen representation basis, images can be approximated in a highly compact manner. This insight underscores the broader theme in the book, where the recognition of underlying patterns in large datasets enables the construction of efficient low-rank representations. This principle is particularly evident in the context of natural images, where inherent compressibility allows for concise approximations depending on the selected basis of representation.
The compression process involves three main steps:

1. Decomposition:
    - Given an image matrix, decompose it using SVD into the three constituent matrices U, Σ, and VT.
    - The singular values in Σ are arranged in descending order.
2. Thresholding:
    - Retain only the first k singular values in Σ, where k is a user-defined parameter.
    - Set the remaining singular values to zero.

3. Reconstruction:
    - Reconstruct the compressed image using the modified Σ and the retained singular vectors from *U* and *VT*.
    - The reconstructed image is an approximation of the original image with reduced information.

Applications of SVD in Image Compression:

Data Compression:
SVD's capability to discard less important singular values enables significant data reduction, making it well-suited for applications with limited storage capacity. This is particularly valuable in scenarios where large datasets need to be efficiently stored and managed.

Transmission Emission:
In telecommunications, bandwidth is often a limiting factor. Compressing images using SVD reduces the amount of data that needs to be transmitted, leading to faster and more efficient communication. This is crucial in applications such as video streaming and image transfer over networks.

Noise Reduction:
SVD-based compression serves as an effective tool for denoising images. By eliminating less significant information, the compression process focuses on preserving the essential features of an image, reducing the impact of noise and enhancing overall image quality.

Facial recognition and Biometrics:
The application of SVD to compress facial images while preserving essential facial features is particularly valuable in facial recognition and biometric systems. This compression technique ensures that the distinctive features necessary for accurate recognition are retained, even in the presence of reduced data.

In conclusion, the principles and applications of image compression using Singular Value Decomposition provide a versatile and powerful approach to managing and transmitting visual data efficiently. The interplay between decomposition, thresholding, and reconstruction allows for tailored compression strategies, finding applications in diverse fields where storage, bandwidth, and data quality are critical considerations. As technology continues to advance, the role of SVD in image compression remains pivotal in addressing the evolving demands of various industries.

Python Implementation:

Original Image:

```
from matplotlib.image import imread
import matplotlib.pyplot as plt
import numpy as np
import os
plt.rcParams['figure.figsize'] = [16, 8]


A = imread(os.path.join('C:\\Users\\rabib\\OneDrive\\Desktop\\Python Practice\
#change path as needed

X = np.mean(A, -1);

img = plt.imshow(X)
img.set_cmap('gray')
plt.axis('off')
plt.show()
```

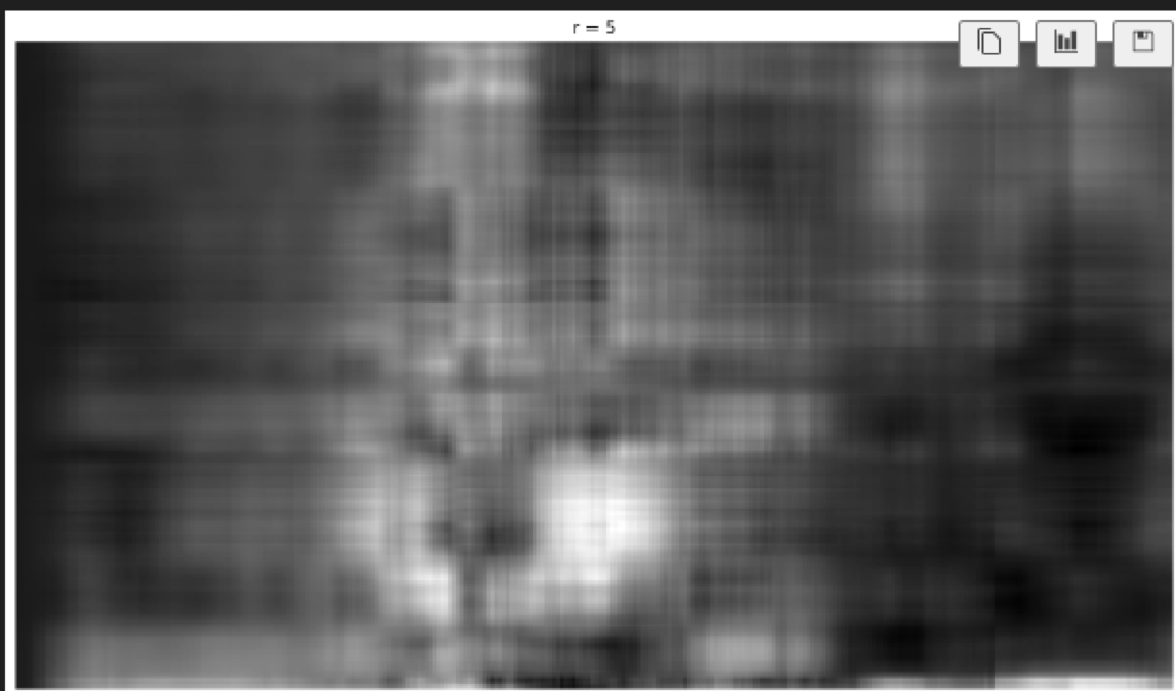
✓ 0.1s Python

Compressed Image:

```
U, S, VT = np.linalg.svd(X,full_matrices=False)
S = np.diag(S)

j = 0
for r in (5, 20, 100):
    Xapprox = U[:,:r] @ S[0:r,:r] @ VT[:r,:]
    plt.figure(j+1)
    j += 1
    img = plt.imshow(Xapprox)
    img.set_cmap('gray')
    plt.axis('off')
    plt.title('r = ' + str(r))
    plt.show()
```

✓ 0.6s Python




r = 20
r = 100

References

Golub, G. H., & Reinsch, C. (1970). Singular value decomposition and least squares solutions. Numerische Mathematik, 14(5), 403-420.

Strang, G. (1993). Introduction to Linear Algebra. Wellesley-Cambridge Press.

Jain, A. K. (1989). Fundamentals of Digital Image Processing. Prentice-Hall, Inc.

Rao, K. R., & Yip, P. (1990). Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press.

Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd ed.). Pearson.

Lu, Y., Do, M. N., & Vetterli, M. (2010). Image compression with optimized truncated singular value decomposition. IEEE Transactions on Image Processing, 19(7), 2017-2026.

Chen, S. S., Donoho, D. L., & Saunders, M. A. (1999). Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 20(1), 33-61.