Assignment 5 – Image Compression Using SVD

Implementation :

This assignment explores image compression using Singular Value Decomposition (SVD) applied block-wise to a grayscale image. The image was divided into 8×8 non-overlapping blocks, and SVD was performed on each. Only the top-k singular values were retained during reconstruction, allowing us to reduce data while preserving image structure.

We implemented two functions for block-wise compression and image reconstruction using SVD (detailed in the Code Snippets section)

Analysis of Results:

We evaluated the effect of different values of k (1 to 8) using three metrics:

Compression Ratio

We calculated compression ratio by comparing how much data is used before and after SVD compression.
Each 8×8 image block originally has 64 values. After compression with k singular values, we store 17×k values (from U, Σ, and V^T).
So the formula becomes:
Compression Ratio = 64 / (17 × k)

As k increases, more data is retained per block, and compression becomes less efficient. The graph shows a steep drop in compression ratio between k = 1 and k = 4.

Reconstruction Error (Frobenius Norm)

To measure the difference between the original and compressed image, we used the Frobenius norm — a mathematical way to sum all the differences between corresponding pixel values.

The error dropped sharply from k = 1 to k = 5, meaning even a few singular values preserved most of the image's structure.

PSNR (Peak Signal to Noise Ratio)

PSNR is a standard image quality metric that compares the original and compressed images using the Mean Squared Error (MSE).
Higher PSNR means better quality.

PSNR increased as k increased — and spiked at k = 8, showing the reconstructed image was nearly identical to the original.

Visual Comparison (Reconstructed Images):

A grid of reconstructed images (for k = 1 to k = 8) visually demonstrates how image quality improves as more singular values are retained.

- **k = 1 to 3**: Heavily compressed, blurry, loss of fine details.
- **k = 4 to 5**: Clearer structure, good balance between compression and quality.
- **k = 6 to 8**: Very close to the original image.

Important Code Snippets :

1.SVD-Based Block Compression Function

This function takes an 8×8 block of the image and compresses it by retaining only the top-k singular values.

```
def compress_block(block, k):
    U, S, VT = np.linalg.svd(block, full_matrices=False)
    U_k = U[:, :k]
    S_k = np.diag(S[:k])
    VT_k = VT[:k, :]
    return U_k @ S_k @ VT_k
```

2. Apply Compression Across the Image

This function divides the full image into 8×8 blocks and applies `compress_block()` to each, returning the compressed version.

```
def compress_image(img, k):
    h, w = img.shape
    compressed = np.zeros_like(img)
    for i in range(0, h, 8):
        for j in range(0, w, 8):
            block = img[i:i+8, j:j+8]
            compressed[i:i+8, j:j+8] = compress_block(block, k)
    return compressed
```

3. Reconstructed Image Visualization Grid

```
fig, axes = plt.subplots(2, 4, figsize=(12, 6))
for i, ax in enumerate(axes.ravel()):
    ax.imshow(images[i], cmap='gray')
    ax.set_title(f'k = {i+1}')
    ax.axis('off')
plt.suptitle("Reconstructed Images with Varying k")
plt.tight_layout()
plt.show()
```

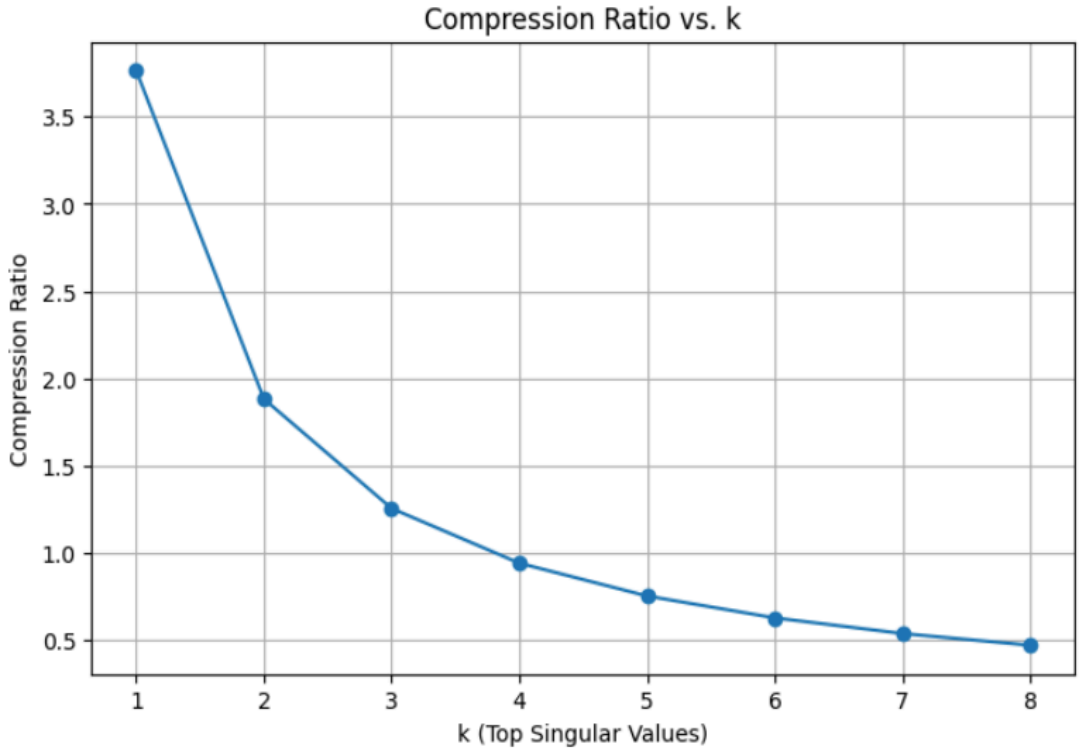
Outputs: Original picture

Grid Reconstructed image samples for k = 1 to 8 (top-left to bottom-right)

Plots :

1. Compression Ratio vs. k – Summary

As the number of singular values k increases, the compression ratio decreases.
This is expected because retaining more information per block requires storing more values, reducing the space savings.
The plot shows a steep drop between k = 1 and k = 4, after which the curve flattens. This indicates that compression efficiency declines rapidly at first, then levels out.

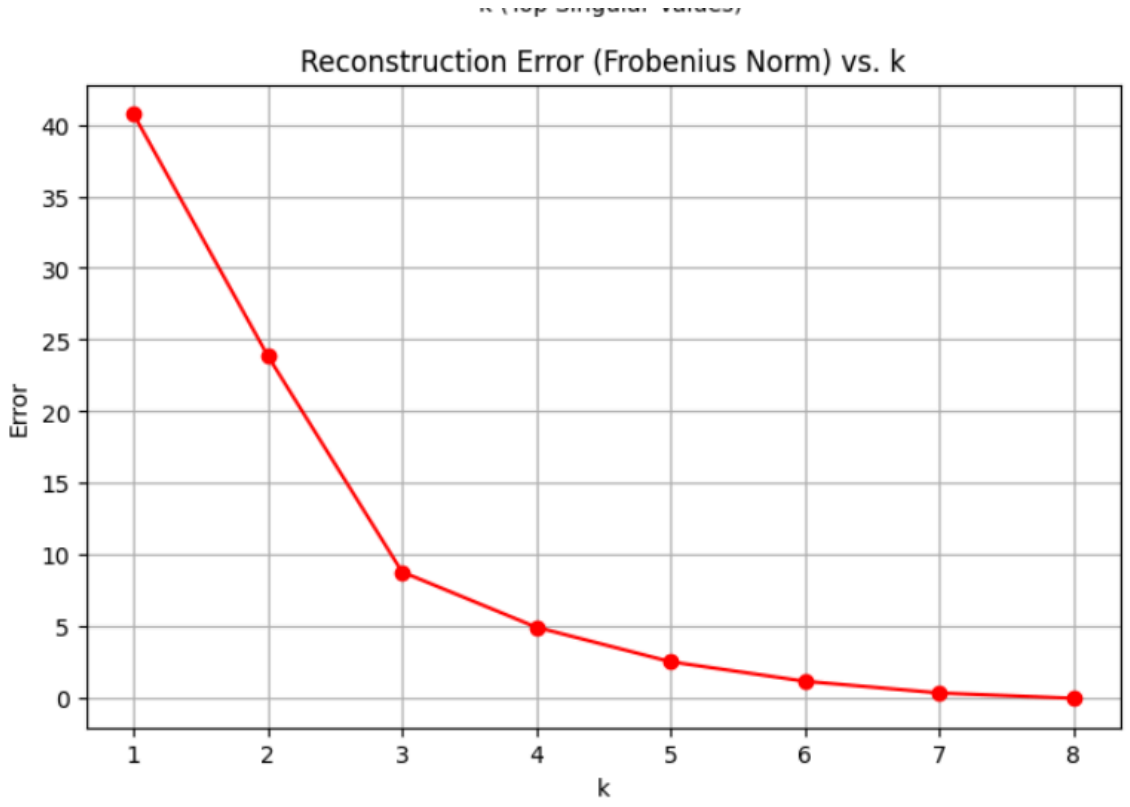


2. Reconstruction Error (Frobenius Norm) vs. k – Summary

This plot shows how much error exists between the original image and its compressed reconstruction.

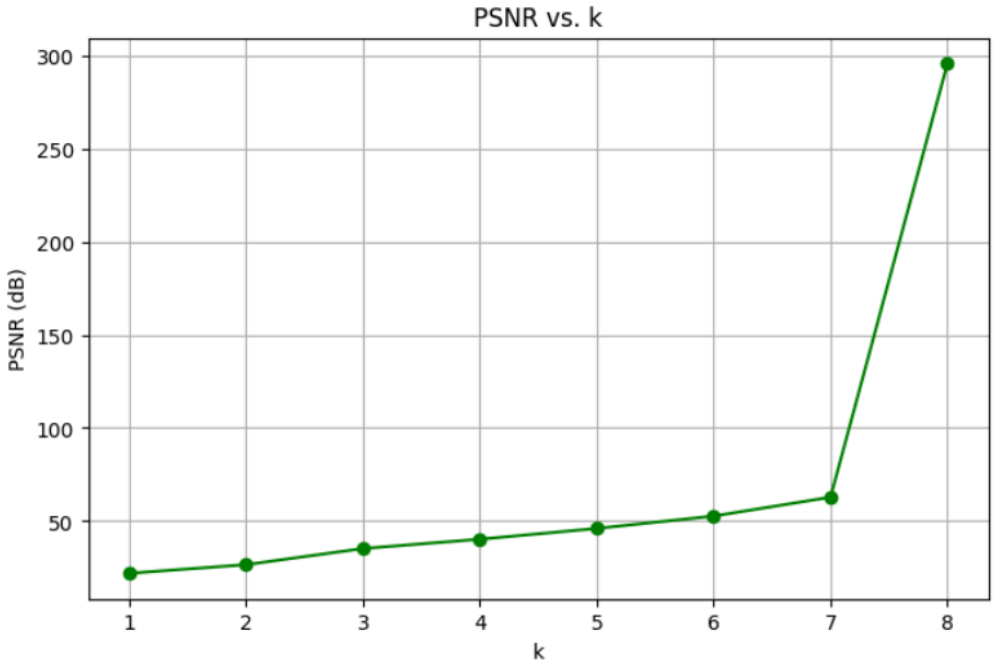
As k increases, the error drops significantly — especially from k = 1 to k = 5.

This suggests that even a small number of singular values can capture most of the image structure, and improvements beyond that are more gradual.



3. PSNR vs. k – Summary

The Peak Signal-to-Noise Ratio (PSNR) increases with higher k, indicating better reconstruction quality.
Low PSNR at k = 1 reflects poor quality, but the values improve rapidly with each additional singular value.
The plot peaks at k = 8, where the image is nearly identical to the original.



Conclusion:

This assignment provided me with both practical and conceptual insights into how Singular Value Decomposition (SVD) can be applied to image compression at a fine-grained level. By implementing block-wise SVD on non-overlapping 8×8 regions of the image, I observed how retaining only a few dominant singular values significantly reduces data storage while preserving visual quality.

The process highlighted the trade-off between compression ratio and image fidelity. Even with a small number of singular values like, k = 4 or 5, the reconstructed images maintained most of the visual structure, demonstrating that much of the image's information is encoded in just a few principal components. This became especially clear through the decreasing reconstruction error and increasing PSNR metrics as k increased.

Beyond the technical implementation, this project deepened my understanding of how linear algebra — particularly matrix factorization — directly supports real-world applications like image processing. It also clarified how compression metrics reflect perceptual quality, providing a full analytical loop from code to result.

Overall, I now have a much clearer appreciation for how SVD underpins many efficient compression techniques, and how it can be implemented in a structured and interpretable way across localized image regions.