Software Assignment - Image Compression Using Truncated SVD

AI25BTECH11011 - G.VARUN KUMAR

Contents

1. **Objective**

2. **Summary of Gilbert Strang's SVD Lecture**

3. **Explanation of the Implemented Algorithm**
   - 3.1. **Mathematical Background**
   - 3.2. **Pseudocode**

4. **Comparison of Algorithms**

5. **Reconstructed Images for Different Values of** k

6. **Error Analysis**

7. **Discussion of trade-offs and reflections on implementation choice**

1) **Objective :**

The main goal of this assignment is to perform image compression using the concept of Singular Value Decomposition (SVD). By keeping only a few of the largest singular values, the size of the image can be reduced while maintaining good visual quality. This project helps to understand how mathematical concepts like matrix factorization can be practically applied to real-world problems such as data storage and transmission.

This approach also demonstrates how SVD can identify and retain the most important information from an image, showing the relationship between mathematics and efficient data representation.

2) **Summary of Gilbert Strang's SVD Lecture :**

a) Singular Value Decomposition (SVD) is a method to break any real matrix A into three matrices: $A = U\Sigma V^T$, where U and V are orthogonal, and Σ holds positive singular values.
b) The columns of U and V represent the main directions in the row and column spaces of A.
c) SVD can be seen geometrically as transforming a unit sphere into an ellipsoid, with singular values representing the stretching in each direction.
d) The singular values are always arranged in decreasing order: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.
e) V contains eigenvectors of A^TA, and U contains eigenvectors of AA^T.
f) The rank of the matrix equals the number of non-zero singular values.
g) The best rank-k approximation is given by:

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T \tag{2.1}$$

h) SVD is used in applications like image compression, denoising, and dimensionality reduction.

3) **Explanation of the Implemented Algorithm :**

a) **Mathematical Background**

The Singular Value Decomposition (SVD) represents any real matrix $A \in \mathbb{R}^{m \times n}$ as:

$$A = U\Sigma V^T \tag{3.1}$$

where:

- U is an $m \times m$ orthogonal matrix.
- V is an $n \times n$ orthogonal matrix.
- Σ is a diagonal matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$.

In image compression, we use only the top k singular values to form a low-rank approximation that captures most of the image's information. i think it is better to write what our code will do instead of it

b) **Pseudocode of Truncated SVD Algorithm**

```
Input: Matrix A (m x n), number of components k
Output: Approximation A_k

for i = 1 to k do
    Initialize random vector v
    repeat
```

```
      v = normalize(A^T * (A * v))
   until convergence
   sigma = norm(A * v)
   u = (A * v) / sigma
   Store (u, sigma, v)
   A = A - sigma * u * v^T
end for

Reconstruct A_k = sum_{i=1}^{k} sigma_i * u_i * v_i^T
```

This iterative approach efficiently compresses the image while retaining essential visual information.

4) **Comparison of Algorithms:**

a) **Direct SVD (Full Decomposition):**

- Performs complete matrix decomposition to get all singular values.
- Accurate but computationally expensive for large matrices.
- Manual implementation without libraries is difficult.

b) **Iterative / Power Method:**

- Finds dominant singular values by repeated multiplication.
- Simple and efficient for large data.
- Used for image compression in this project.

c) **QR + SVD Hybrid Method:**

- Uses QR decomposition to simplify the matrix before applying SVD.
- Improves numerical stability but increases complexity.

d) **Jacobi Method:**

- Based on successive orthogonal rotations.
- Accurate but too slow for image data.

Comparison Summary:

- **Direct SVD:** Accurate but slow.
- **Power Method:** Simple and efficient.
- **QR + SVD:** Stable but complex.
- **Jacobi:** Accurate but slow.

Why Iterative Power Method Was Used:

- Balanced accuracy and speed.
- Easy to implement manually.
- Handles large images effectively.
- Easy to understand, so I used this method in my project.

5) **Reconstructed Images for Different Values of k :**

- k determines how many singular values are kept for reconstruction.
- Smaller k means higher compression but less quality.
- Larger k gives better quality but lower compression.
- Below are results for different values of k.
- As k increases, reconstructed images become clearer and closer to the original.
- Small k produces high compression but blurry results.

- Optimal k balances compression and image quality.

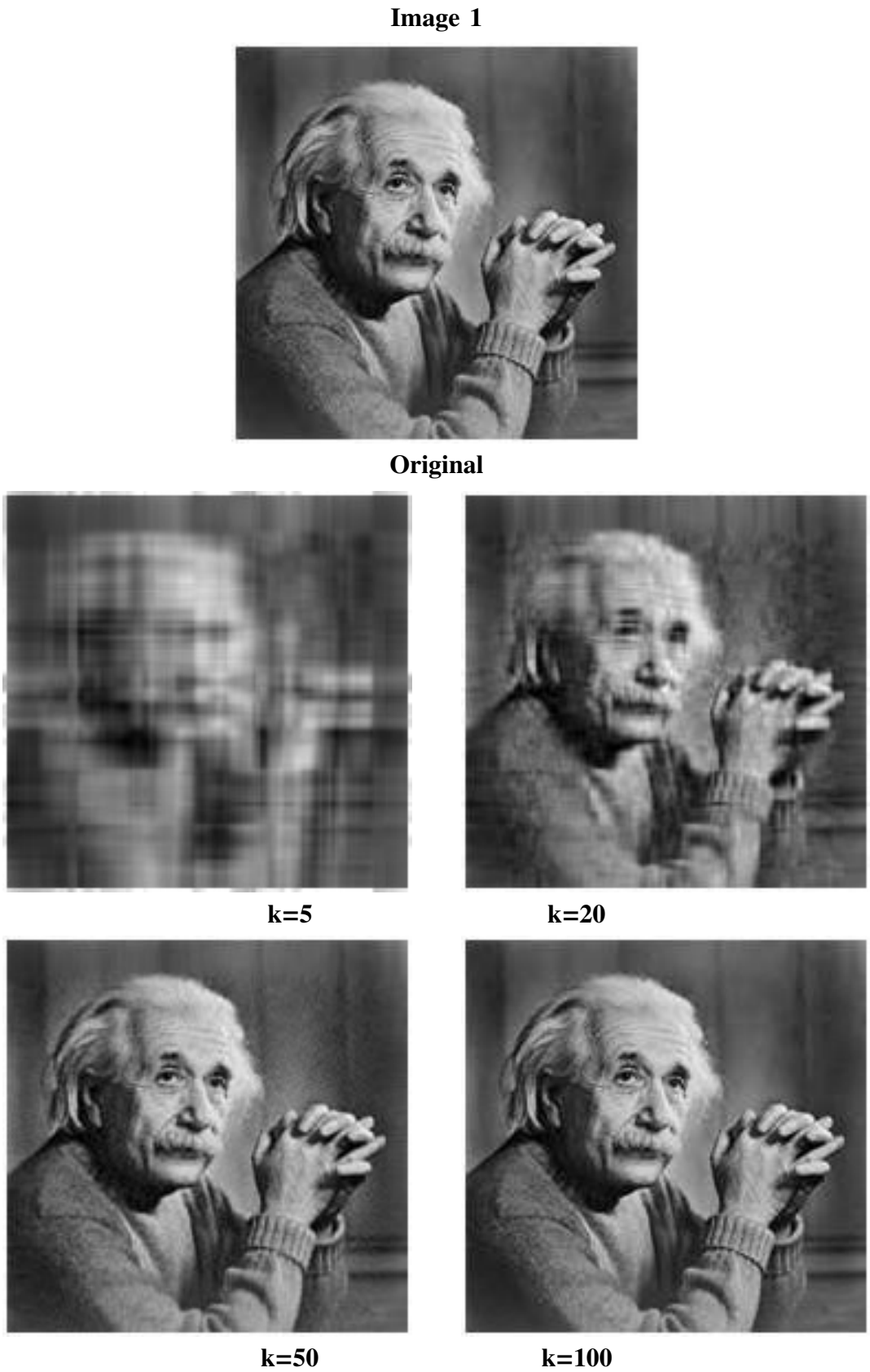

Fig. 5.1: Original image and its reconstructed versions for different k values (Image 1).

6) **Error Analysis**

In image compression using Singular Value Decomposition (SVD), the **reconstruction error** represents the difference between the original image matrix A and the reconstructed matrix $\hat{A}_k$, which is obtained using only the top k singular values. This error helps to measure how much information or detail is lost

due to compression.

The reconstruction error is calculated using the *Frobenius norm*, which measures the overall difference between two matrices. It is mathematically expressed as:

$$E_k = \|A - \hat{A}_k\|_F \tag{6.1}$$

$$= \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij} - \hat{A}_{ij})^2} \tag{6.2}$$

where:

- A is the original image matrix of size $m \times n$,
- $\hat{A}_k$ is the reconstructed image matrix using the first k singular values,
- E_k is the reconstruction error or deviation between A and $\hat{A}_k$.

a) The reconstruction error E_k indicates how well the image is approximated for a given k value.

b) A smaller E_k means the reconstructed image is closer to the original, hence better quality.

c) When k is small, only a few dominant singular values are retained; the image loses details and appears blurred, leading to high error.

d) As k increases, more singular values are included in reconstruction, improving visual quality and reducing E_k.

e) The rate of error reduction depends on the image content, smoother images show faster convergence with fewer singular values.

TABLE 6: Reconstruction Error for Different Images using Truncated SVD

Original Image	**k Value**	**Reconstruction Error (E_k)**
Einstein	5	4712.9979
Einstein	20	2126.4778
Einstein	50	880.4435
Einstein	100	164.8112
Globe	5	20703.9655
Globe	20	10633.9403
Globe	50	6185.1881
Globe	100	3672.6511
Greyscale	5	11154.4725
Greyscale	20	3815.2310
Greyscale	50	1178.3995
Greyscale	100	549.9458

7) **Discussion of Trade-offs and Reflections on Implementation Choice**

Trade-offs in SVD-based Compression

The SVD-based image compression technique presents several important trade-offs between compression efficiency, reconstruction accuracy, and computational complexity. These trade-offs are consistent with the discussions in the reference material and are summarized below:

a) **Compression Ratio vs. Image Quality:**

- When the number of singular values k is small, the image can be stored using fewer parameters (U_k, Σ_k, V_k), leading to a higher compression ratio.
- However, the reconstructed image appears blurred or loses fine details because most of the smaller singular values (which capture texture and edge information) are discarded.
- As k increases, more singular components are included, resulting in better reconstruction but larger

storage size.

b) **Error vs. Computational Time:**

- The reconstruction error decreases with increasing k, as more information from the original matrix is retained.
- However, this comes at the cost of higher computational time since the power iteration and deflation process must be repeated for each additional singular value.

c) **Optimal Selection of k:**

- Choosing an appropriate value of k is critical.
- A smaller k offers efficient compression but noticeable visual degradation, while a larger k maintains quality but increases both computation and storage.
- Practically, a moderate value such as $k = 50$ or $k = 100$ often provides a good trade-off for most grayscale images.

Mathematically, the relationship between reconstruction error and k can be represented as:

$$E_5 > E_{20} > E_{50} > E_{100} \tag{7.1}$$

where E_k denotes the Frobenius norm of the difference between the original image matrix A and its rank-k reconstruction A_k, computed as:

$$E_k = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij} - A_{k,ij})^2} \tag{7.2}$$

This clearly shows that as k increases, the approximation becomes more accurate and the error monotonically decreases.

Reflections on Implementation Choices

a) **Use of Power Iteration Method:**

- The power iteration algorithm was chosen to compute the dominant singular value and corresponding singular vectors.
- It iteratively updates u and v vectors until convergence, providing an efficient approach for the first few singular values.

b) **Deflation Strategy:**

- After each singular triplet (σ_i, u_i, v_i) is obtained, the matrix is deflated using:

$$A \leftarrow A - \sigma_i u_i v_i^T \tag{7.3}$$

- This ensures subsequent iterations extract orthogonal singular vectors, gradually decomposing the matrix into its major components.

Image 2

Original

k=5

k=20

k=50

k=100

Fig. 5.2: Original image and its reconstructed versions for different k values (Image 2).

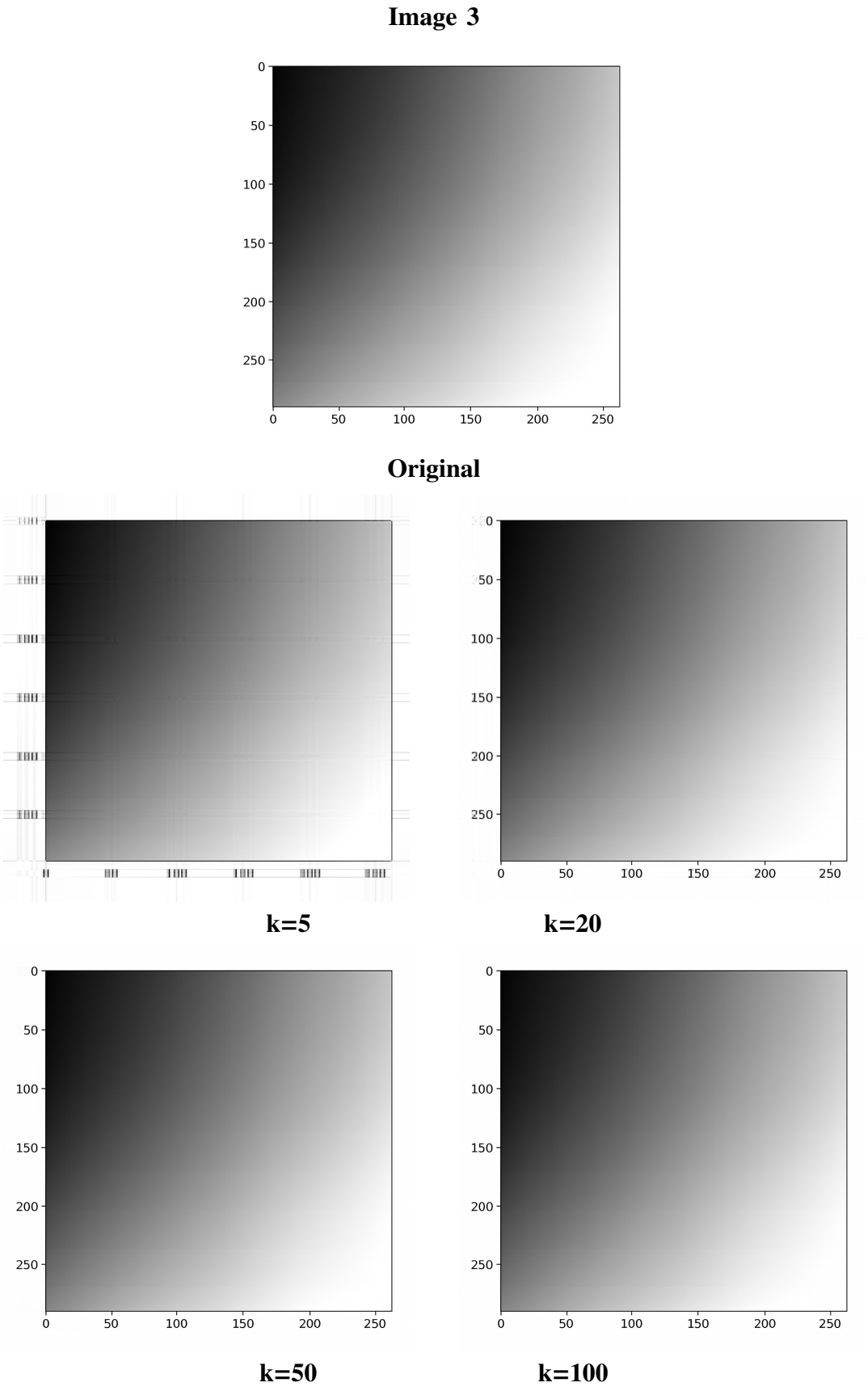

Fig. 5.3: Original image and its reconstructed versions for different k values (Image 3).