**Data Communication - 2CS202**

**Innovative Assignment**

**B. Tech. Semester IV**

Prepared By

**PARTH SHAH (23BCE217)**

**NIHAR MEHTA (23BCE197)**



**School of Engineering**

**Institute of Technology**

**Nirma University**

**Ahmedabad 382481**

**Image and Text Compression Techniques**

**1. Introduction**

Data compression is a fundamental concept in data communication and storage. It reduces the size of data to save bandwidth, storage space, and transmission time. Compression techniques can be broadly classified into:

* **Lossless Compression:** (no data loss, used for text, executable files)
* **Lossy Compression:** (some data loss, used for images, audio, video)

This report discusses two compression implementations:

1. Image Compression using Singular Value Decomposition (SVD)

2. Text Compression using Huffman Coding and Run-Length Encoding (RLE)

**2. Image Compression using Singular Value Decomposition (SVD)**

**2.1 Algorithm Used**

The image compression technique implemented here uses Singular Value Decomposition (SVD), a matrix factorization method from linear algebra.

**Key Steps:**

1. Convert the image to grayscale (if it is RGB).

2. Apply SVD to the image matrix:

A = U Σ V^T

- A: Input image matrix

- U: Left singular vectors (orthogonal)

- Σ: Diagonal matrix of singular values (sorted in descending order)

- V^T: Right singular vectors (orthogonal)

3. Truncate the SVD components by keeping only the top k singular values.

4. Reconstruct the compressed image using:

A\_compressed = U\_k Σ\_k V\_k^T

**2.2 Concepts Related to Data Communication**

* **Lossy Compression**: SVD-based compression is lossy because discarding smaller singular values removes some image details.
* **Trade-off Between Compression and Quality**: Higher k values retain more details but reduce compression.
* **Bandwidth Efficiency**: Compressed images require less storage and transmit faster over networks.

**2.3 Advantages & Limitations**

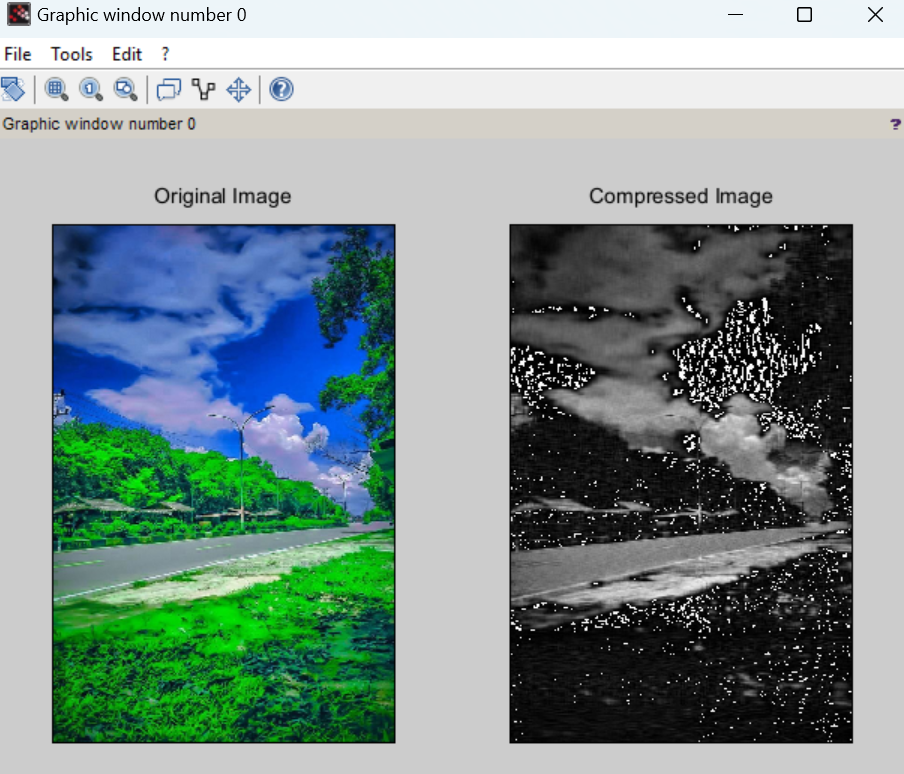
✔ Effective for images with low-rank structure (e.g., faces, smooth textures).

✔ Adjustable compression level by varying k.

✖ Not suitable for lossless compression.

✖ Computationally expensive for very large images.

**2.4 Output**



**3. Text Compression using Huffman Coding and Run-Length Encoding (RLE)**

**3.1 Algorithms Used**

**(A) Huffman Coding (Lossless)**

- **Principle**: Variable-length encoding where frequent characters get shorter codes.

- **Steps:**

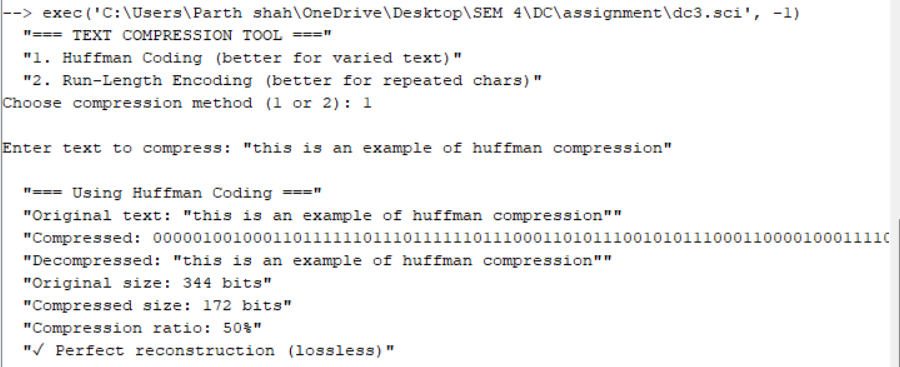
1. Build a frequency table of characters.

2. Construct a Huffman Tree using a priority queue (min-heap).

3. Generate binary codes by traversing the tree (left=0, right=1).

4. Replace characters with Huffman codes for compression.

5. Decompress by traversing the tree using the bitstream.

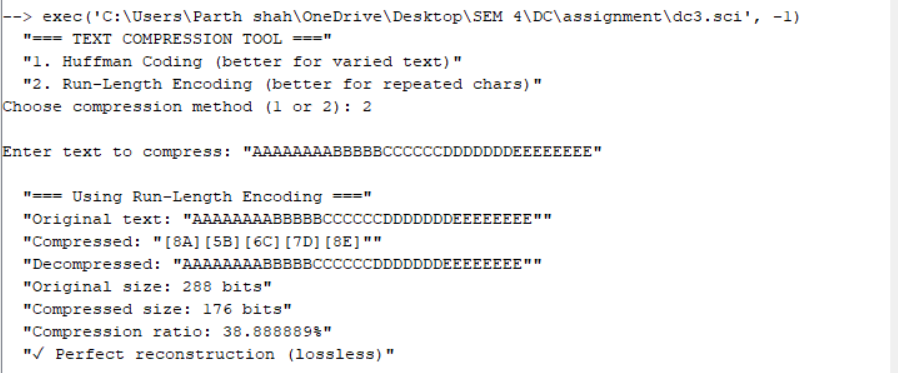


**(B) Run-Length Encoding (RLE) (Lossless)**

- **Principle**: Replace consecutive repeated characters with a count and symbol.

- **Example**: AAAABBBCC → [4A][3B][2C]

- Works best for data with long repeated sequences (e.g., binary images, simple text).



**3.2 Concepts Related to Data Communication**

* **Entropy Encoding (Huffman):** Minimizes average code length based on symbol frequency.
* **Redundancy Removal (RLE):** Exploits repeated patterns for compression.
* **Lossless Nature:** Critical for text, where data integrity is essential.
* **Compression Ratio:** Depends on input data characteristics.

**3.3 Advantages & Limitations**

**(A) Huffman Coding (Lossless)**

✔ Optimal for non-uniform symbol distributions.

✔ Guarantees lossless reconstruction.

✖ Requires frequency analysis before encoding.

✖ Inefficient for small or uniform data.

**(B) Run-Length Encoding (RLE) (Lossless)**

✔ Simple and fast.

✔ Very efficient for repeated data (e.g., AAAAA…)

✖ Ineffective for non-repetitive data (e.g., random text)

**4. Conclusion**

* SVD-based image compression is useful for reducing image sizes with adjustable quality loss.
* Huffman Coding is a powerful method for text compression, especially when symbol frequencies vary.
* Run-Length Encoding is best suited for data with repeated sequences.
* Choice of compression method depends on data type and requirements (lossy vs. lossless, speed vs. ratio).

These techniques are widely used in data communication systems (e.g., JPEG uses DCT, ZIP uses Huffman/RLE) to optimize storage and transmission efficiency.