AURORA HIGHER EDUCATION AND RESEARCH ACADEMY
Deemed-to-be University
(Estd. u/s. 03 of UGC Act 1956)
Uppal, Hyderabad, Telangana - 500 098.
www.aurora.edu.in
Department of Mathematics

Course Name: LINEAR ALGEBRA

Course Code: MA103

Year: F. Y. B. Tech (AIML – "C")

I YEAR III TERM

Title : Image Compression Using Rank Nullity Theorem

Faculty Name:

Ms. Gita S Parthiban

Associate Professor ,

Head of the Department Mathematics

Student Names :

A.Vignesh(241U1R2089)

OUTLINE :

- ABSTRACT
- INTRODUCTION
- LITERATURE SURVEY
- MATHEMATICAL FORMULA
- OBJECTIVES
- METHODOLOGY
- SOFTWARE REQUIREMENTS
- FUNCTIONAL REQUIREMENTS
- CODE IN SEPARATE FILE
- EXAMPLE
- RESULTS
- CONCLUSION
- REFERENCES

ABSTRACT

This project explores how the Rank-Nullity Theorem, a fundamental concept in Linear Algebra, can be applied to image compression. Digital images can be represented as matrices, and by using the theorem, we analyze the rank (important information) and nullity (redundant or compressible data) of these matrices. By reducing the rank, we can compress the image while retaining its essential features. This method helps in reducing storage space and transmission time without losing much image quality. The project shows how mathematical concepts can be effectively used in real-world applications like image processing and data compression.

INTRODUCTION

In the digital era, managing large image files can be a challenge due to limited storage. Image compression helps solve this by reducing file size while. storage space and transmission time without losing much image quality. This project focuses on a mathematical method of compression using the Rank-Nullity Theorem from linear algebra. By representing an image as a matrix, we can analyze which data is essential (rank) and which is redundant (nullity). Removing the redundant parts leads to a smaller, compressed image that still looks very similar to the original.

Through this approach, we highlight how mathematics can be applied creatively to solve real-world problems in image processing and data optimization.

LITERATURE SURVEY

1. The Rank-Nullity Theorem and Singular Value Decomposition.

Author: Nicholas J. Higham

- Explains how Rank-Nullity Theorem helps to understand the structure of matrices by linking rank (useful data) and nullity (redundant data).
- Describes Singular Value Decomposition (SVD) as a method to break down a matrix into simpler parts, showing the most important features of the data.
- Shows how combining Rank-Nullity and SVD gives deep insights into how data is transformed in applications like image processing and compression.

LITERATURE SURVEY

2. An Application of Linear Algebra to Image Compression.

Authors: Khalid Elasnaoui, Mohamed Ouhda, B. Aksasse, Mohammed Ouanan

- Explains how Singular Value Decomposition (SVD) is used to compress images by keeping only the most important parts of the image matrix.
- Shows how less important data can be removed to save storage space while keeping the image quality acceptable.
- Compares SVD-based compression with traditional methods and shows its effectiveness in maintaining high image quality with smaller file sizes.
- Applies this method to both grayscale and color images, proving it works well for various types of image data.

MATHEMATICAL FORMULA

- **Matrix Representation**

  Image = Matrix A (Each number represents a pixel's intensity)

- **Rank of a Matrix**

  Rank(A) = Important Data (Keeps essential details)

- **Nullity of a Matrix**

  Nullity(A) = Redundant Data (Can be removed for compression)

- **Rank-Nullity Theorem** $T : V \rightarrow W$, where V is a finite-dimensional vector space:

$$\text{Rank}(A) + \text{Nullity}(A) = \dim(A)(\text{Number of columns of } A)$$

In matrix terms, for a matrix A of size $m \times n$:

$$\text{Rank}(A) + \text{Nullity}(A) = n$$

MATHEMATICAL FORMULA

- **Singular Value Decomposition (SVD)**:

You apply SVD to decompose the image matrix into three components:

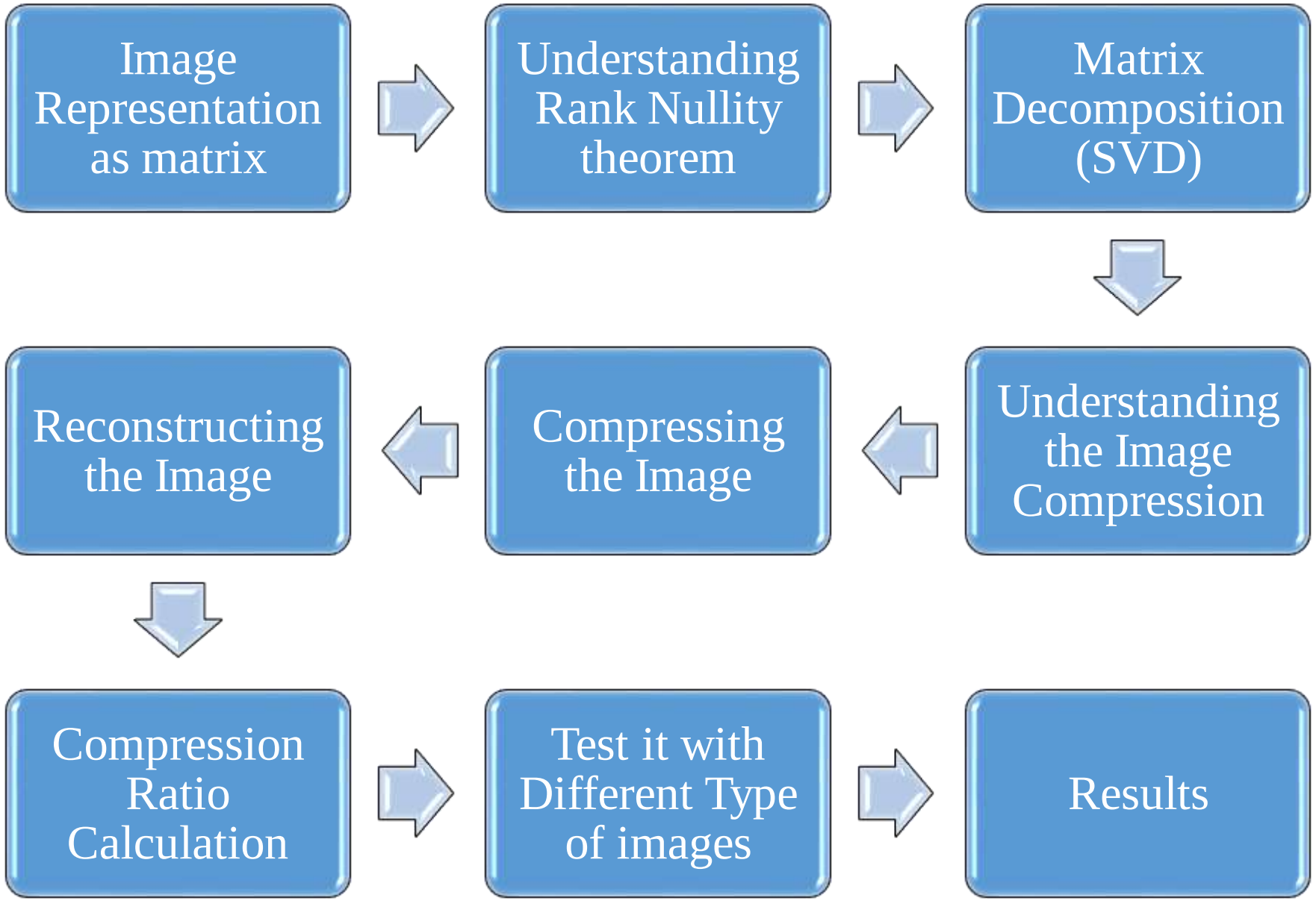
$$A = U\Sigma V^T$$

- **U**: Left singular vectors.
- **S**: Singular values, stored in a diagonal matrix **Σ**.
- $\mathbf{V^T}$: Right singular vectors (transposed).

OBJECTIVES

- **Understand Image Representation**: Learn how digital images are represented as matrices of pixel values.
- **Apply Rank-Nullity Theorem**: Use the Rank-Nullity Theorem to identify essential and redundant data in image matrices.
- **Optimize Data Storage**: Explore how reducing redundant data can compress image files without losing significant quality.
- **Enhance Compression Efficiency**: Improve the efficiency of image compression by leveraging mathematical techniques from linear algebra.
- **Improve Image File Handling**: Demonstrate how the Rank-Nullity Theorem can lead to faster storage, transfer, and processing of image files.

METHODOLOGY



SOFTWARE REQUIREMENTS

1.Libraries:

- **NumPy**: For matrix operations and Singular Value Decomposition (SVD).
- **Matplotlib**: To display and compare images.
- **Pillow (PIL)**: For loading, converting, and saving images.

2. Visual Studio Code (VS Code)

Use VS Code as your integrated development environment (IDE). Make sure you have the following extensions:

- **Python extension**: Provides support for running Python code and debugging.

3. Operating System

The project will work on Windows, Linux, or macOS, as long as Python and the necessary libraries are installed.

FUNCTIONAL REQUIREMENTS

I. **Image Input**

Accept an image file (e.g., PNG, JPEG) for compression.

II. **SVD Decomposition**

Perform Singular Value Decomposition (SVD) on the image matrix.

III. **Rank Reduction**

Compress the image by keeping only the most significant singular values.

IV. **Image Reconstruction**

Reconstruct the image using the reduced matrices.

V. **Compression Ratio** Calculate and display the compression ratio.

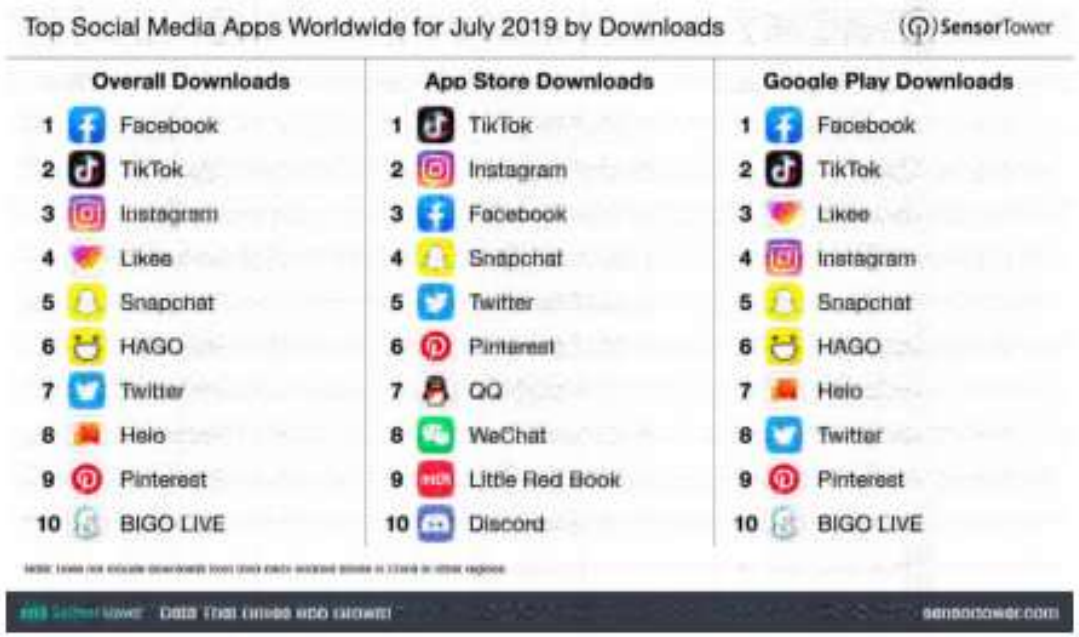
VI. **Quality Evaluation** Compare the original and compressed images visually

CODE IN SEPARATE FILE

RESULTS

Original Image

Compressed Image



(i) Here we can see the original and compressed images of the social media applications list

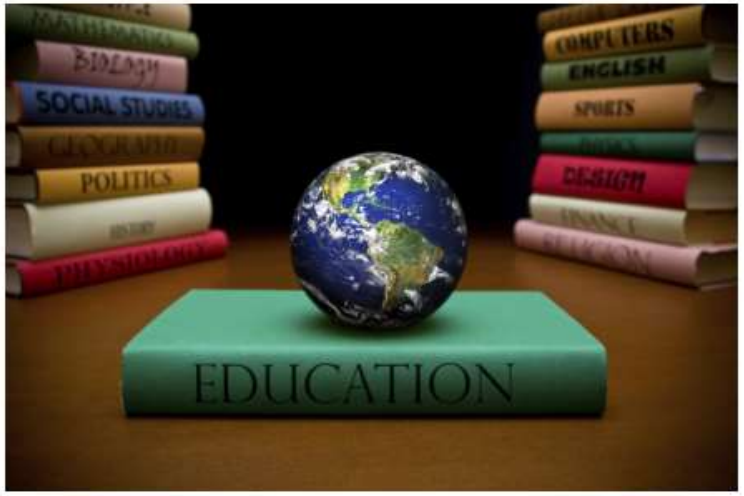
RESULTS

Original Image

Compressed Image

(ii) This the image compressed of text image

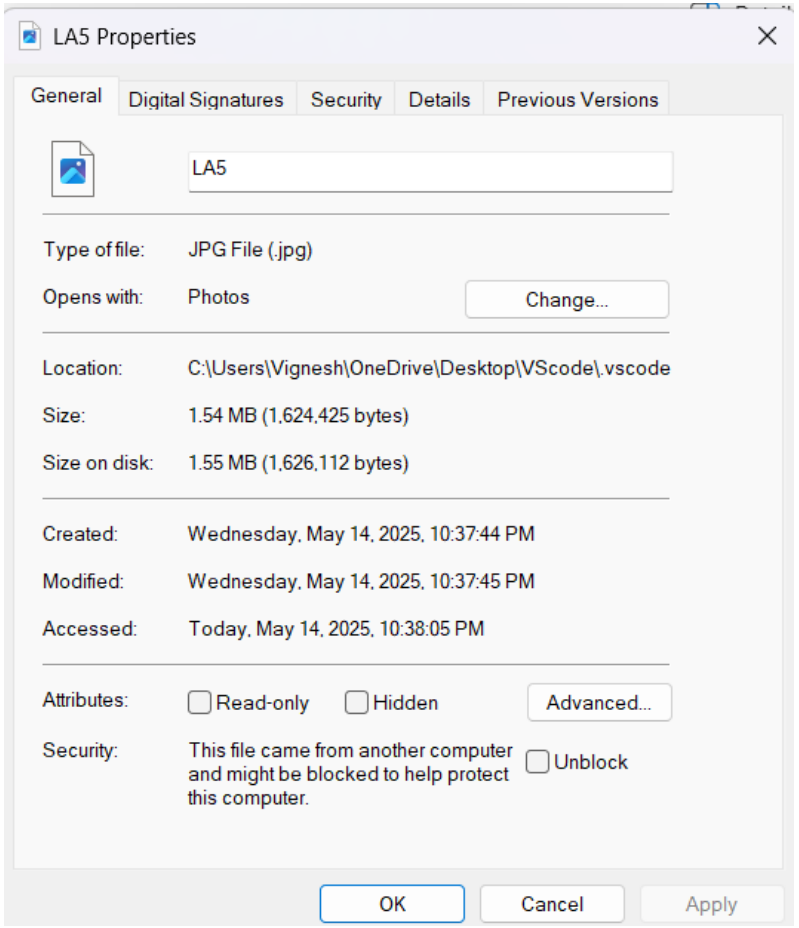
Original Image

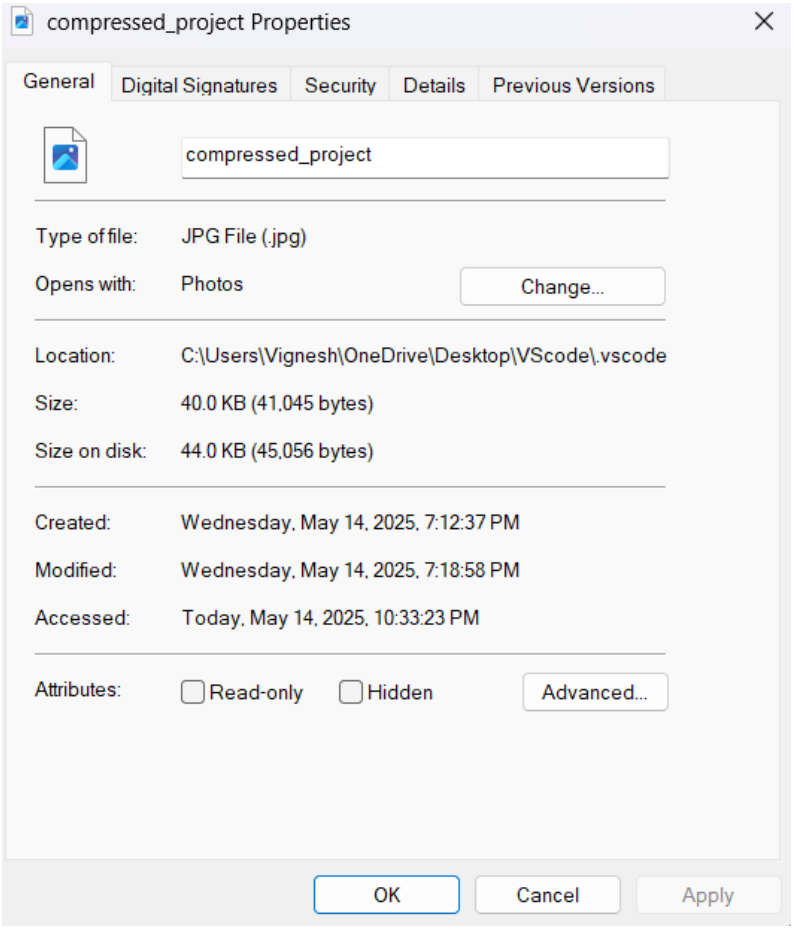


Compressed Image

(iii) In this image we can see both text and a globe

RESULTS





(iv) Here we can the difference between the size of original and the compressed images of the Fig-(iii)

EXAMPLE

Step 1: Start with a simple image matrix

Say you have a grayscale image stored as a matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

This is a 3×3 image, and each number represents the intensity (brightness) of a pixel.

Step 2: Check for Rank

We want to determine the rank of matrix A, which tells us how many independent rows or columns it has.

Let's check if Row 3 can be written as a linear combination of Row 1 and Row 2:

$$\text{Row 3} = 2 \times \text{Row 2} - \text{Row 1}$$

So, the image does not contain 3 independent rows it only needs 2 rows (rank = 2) to represent the same information.

EXAMPLE

Step 3: Use Rank-Nullity Theorem

Let's say you consider the matrix A as a transformation from $R^3 \rightarrow R^3$.

Then:

$$\text{Rank(A)} + \text{Nullity(A)} = 3$$

$$\Rightarrow \text{Rank} = 2 \Rightarrow \text{Nullity} = 1$$

This means one direction in the input space is "wasted" that's where compression becomes possible.

Step 4: Compress Using Row Dependency

Since:

$$\text{Row3} = -1 \cdot \text{Row 1} + 2 \cdot \text{Row2}$$

EXAMPLE

$$A' = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

And the reconstruction rule for Row 3.

- **Original** storage: 9 values
- **Compressed** storage:
- 6 values for the first 2 rows
- 2 coefficients: −1 and 2
- **Total** = 8 values

Compression Ratio (Row Method)

$$\text{Compression Ratio} = \frac{Compressed\ Size}{Original\ Size} = \frac{8}{9} \approx 0.89$$

This means we store only 89% of the original data → ~11% space saved

EXAMPLE

Step 5: Use SVD for Compression

Now apply Singular Value Decomposition:

$$A=U\Sigma VTA$$

Let's assume the SVD of matrix A gives:

$$\Sigma = \text{diag}(16.8, 1.07, 0.0)$$

This means:

- The third singular value is zero ⇒ rank = 2
- So we can approximate A by using only the top 2 singular values

Compression Ratio (SVD Method)

Original matrix A has 9 values. Compressed representation has 14.

$$\text{Compression Ratio} = \frac{9}{14} \approx 0.64$$

This means we are storing only 64% of the data → ~36% compression

CONCLUSION

This project successfully demonstrates how the Rank-Nullity Theorem and Singular Value Decomposition (SVD) can be applied to compress digital images efficiently. By reducing the rank of the image matrix and keeping only the most important data, we were able to significantly reduce file size while maintaining acceptable image quality. The method is mathematically sound, practical, and provides a good balance between compression and visual clarity. This approach can be extended for larger datasets and used in real-world image storage and transmission applications.

REFERENCES

- https://en.wikipedia.org/wiki/Rank%E2%80%93nullity_theorem
- https://www.geeksforgeeks.org/image-compression-using-svd/
- https://pillow.readthedocs.io/en/stable/
- https://www.mathworks.com/help/matlab/ref/svd.html
- https://www.tutorialspoint.com/image-compression-in-python
- https://matplotlib.org/stable/users/pyplot_tutorial.html
- https://en.wikipedia.org/wiki/Singular_value_decomposition

THANK YOU