IMAGE COMPRESSION USING DISCRETE COSINE TRANSFORM (DCT)

B.Tech Minor Project Report

by

TUSHAR RAJ (20104108019)

ANCHAL RANI (20104108013)

ANUJ KUMAR (20104108056)

NADEEM KAUSAR (21104108907)

Under the Guidance of

Mr. Rohit Kumar

Assistant Professor

Dept. of Electronics and Communication Engineering

Department of Electronics & Communication Engineering

BHAGALPUR COLLEGE OF ENGINEERING, BHAGALPUR

April 2024

IMAGE COMPRESSION USING DISCRETE COSINE TRANSFORM (DCT)

Project Report Submitted to
Bhagalpur College of Engineering, Bhagalpur in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology in Department of Electronics and Communication Engineering

By

TUSHAR RAJ (20104108019)
ANCHAL RANI (20104108013)
ANUJ KUMAR (20104108056)
NADEEM KAUSAR (21104108907)

Under the Guidance of
Mr. Rohit Kumar
Assistant Professor
Dept. of Electronics and Communication Engineering

Department of Electronics & Communication Engineering
BHAGALPUR COLLEGE OF ENGINEERING, BHAGALPUR
April 2024

Bhagalpur College Of Engineering,Bhagalpur

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the present project work entitled "**IMAGE COMPRESSION USING DISCRETE COSINE TRANSFORM (DCT)**" submitted by **Tushar Raj** (20104108019), **Anchal Rani** (20104108013), **Anuj Kumar** (20104108056) & **Nadeem Kausar** (21104108907) in partial fulfilment of requirement for the award of degree of Bachelor of Technology in the department of **Electronics and Communication Engineering**
to the **Bihar Engineering University, Patna,** in an authentic record of our work carried out at **Bhagalpur College Of Engineering, Bhagalpur**, under my supervision and guidance.

....

Mr. Rohit Kumar
Associate Professor
Department of Electronics &
Communication Engineering, Bhagalpur

Examined & approved on ....

Internal Examiner .... **External Examiner ....**

Bhagalpur College Of Engineering,Bhagalpur

Department of Electronics & Communication Engineering

DECLARATION

I, **Tushar Raj** (20104108019), **Anchal Rani** (20104108013), **Anuj Kumar** (20104108056) & **Nadeem Kausar** (21104108907) students of Bachelor of Technology (2020- 2024)hereby declare that the work which is being presented in the report entitled **"IMAGE COMPRESSION USING DISCRETE COSINE TRANSFORM (DCT)** in partial fulfilment of requirement for the award of the **Degree of Bachelor of Technology in Electronics and Communication Engineering**. The work has been carried out at **Bhagalpur College of Engineering, Bhagalpur** (Affiliated to **BEU Patna**) and is an authentic record of my own work carried out under the supervision and guidance of Rohit Kumar (Asst. Prof. ECE Deptt.)

Tushar Raj (20104108019)

Anchal Rani (20104108013)

Anuj Kumar (20104108056)

Nadeem Kausar (21104108907)

ACKNOWLEDGEMENT

We would like to thank **Dr. Pushpalata,** Head of Department, Dept. of Electronics and Communication Engineering, **BCE BHAGALPUR**, for her constant guidance and support as well as mentoring throughout the project work.

We express our sincere thanks and profound gratitude to **Mr. Rohit Kumar** (Assistant Professor, Dept. of Electronics & Communication Engineering) for assigning us such an interesting project, for his guidance and constant supervision amid the study and implementation of this project. I acknowledge his valuable and timely help and morale boosting without which this project would not have been possible. His insights, knowledge, experience and technical skills have always guided me in my work.

We are also thankful to whole **Electronics & Communication Engineering department** for providing us the technical support to carry out the project work, to let us utilize all the necessary facilities of the institute and guidance at each and every step during project work. I feel a deep sense of gratitude for my parents who formed a part of my vision and taught me the good things that really matter in life. We would like to thank family members for their support.

Thanking You

ABSTRACT

In today's technological world as our use of and reliance on computers continues to grow, so too does our need for efficient ways of storing large amounts of data and due to the bandwidth and storage limitations, images must be compressed before transmission and storage. For example, someone with a web page or online catalog that uses dozens or perhaps hundreds of images will more than likely need to use some form of image compression to store those images. This is because the amount of space required to hold unadulterated images can be prohibitively large in terms of cost. Fortunately, there are several methods of image compression available today. This fall into two general categories: lossless and lossy image compression. However, the compression will reduce the image fidelity, especially when the images are compressed at lower bit rates. The reconstructed images suffer from blocking artifacts and the image quality will be severely degraded under the circumstance of high compression ratios. In order to have a good compression ratio without losing too much of information when the image is decompressed, we use DCT. A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. The JPEG process is a widely used form of lossy image compression that centers on the Discrete Cosine Transform. DCT and Fourier transforms convert images from time-domain to frequency domain to decorrelate pixels. The DCT transformation is reversible. The DCT works by separating images into parts of differing frequencies. During a step called quantization, where part of compression actually occurs, the less important frequencies are discarded, hence the use of the term "lossy". Then, only the most important frequencies that remain are used retrieve the image in the decompression process. As a result, reconstructed images contain some distortion; but as we shall soon see, these levels of distortion can be adjusted during the compression stage. The JPEG method is used for both color and black and-white images.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background and Motivation:

Image compression plays a pivotal role in various fields such as digital photography, medical imaging, video streaming, and multimedia communication. The primary goal of image compression is to reduce the size of digital images while preserving important visual information to the greatest extent possible. This reduction in size allows for efficient storage, transmission, and processing of images, leading to benefits such as reduced storage requirements, faster transmission speeds, and improved user experience.

One of the fundamental techniques used in image compression is the Discrete Cosine Transform (DCT). The DCT is a mathematical transformation that converts spatial domain information into frequency domain information. It has been widely adopted in image compression standards such as JPEG (Joint Photographic Experts Group) due to its ability to concentrate most of the image information in a small number of coefficients, allowing for high compression ratios with minimal loss of image quality.

This project's motivation lies in optimizing image compression efficiency through DCT, addressing industry demands for data economy and fast transmission. It aims to enhance understanding of DCT-based techniques, develop practical skills, and potentially innovate in an area crucial for multimedia, telecommunications, and healthcare applications.

1.2 Objectives and Scope

The objective of image compression is to reduce the size of an image file while preserving its visual quality to an acceptable level. This reduction in file size is crucial for various applications, including:

1. Storage Efficiency: Compressed images occupy less disk space, making it easier and more cost-effective to store and manage large collections of images, such as those used in digital archives, websites, or databases.
2. Bandwidth Conservation: Smaller file sizes mean reduced data transfer requirements, making image compression essential for efficient transmission over networks with limited bandwidth, such as the internet or wireless networks. This is particularly important for mobile devices and applications where data usage is a concern.
3. Faster Transmission: Compressed images can be transmitted more quickly than their uncompressed counterparts, leading to faster loading times for web pages, improved

streaming performance for multimedia content, and better user experiences overall.

4. Resource Optimization: In resource-constrained environments, such as embedded systems or mobile devices, image compression helps optimize memory and processing resources by reducing the amount of data that needs to be stored, processed, or displayed.

1.3 Problem Statement

This project focuses on improving image compression efficiency using Discrete Cosine Transform Compression (DCT) amidst ongoing challenges in maintaining image quality. Key objectives include enhancing compression ratios while preserving image details, reducing computational demands for real-time applications, and ensuring adaptability across diverse image types. By addressing these challenges, the project aims to advance compression technology, meeting the growing need for efficient data management and high-quality image representation across various domains.

1.4 Research Questions

Digital Photography:

1. How can DCT-based compression techniques be tailored to preserve fine details and colour accuracy in high-resolution digital photographs?

2. What strategies can be developed to minimize compression artifacts and perceptual quality loss in DCT-compressed images intended for professional photography applications?

3. How does the choice of DCT quantization matrix and compression parameters affect the visual quality and file size of compressed digital photographs?

Video Compression:

1. How can DCT-based compression algorithms be adapted to efficiently compress video sequences while maintaining temporal coherence and visual quality?
2. What methods can be devised to exploit temporal redundancy in video data to enhance compression efficiency using DCT?
3. How does the choice of DCT block size and motion estimation technique affect the compression performance of video sequences?

1.5 Methodology

The project will involve the following steps:

1. **Image Preprocessing:**
    - Collect a diverse dataset of digital images.
    - Convert images to a suitable format and normalize pixel values.
2. **DCT Transformation in MATLAB:**
    - Implement DCT transformation algorithm.
    - Apply DCT to image blocks and generate coefficient matrix.
3. **Quantization and Encoding:**
    - Design quantization matrix and apply to DCT coefficients.
    - Encode quantized coefficients using entropy encoding techniques.
4. **Performance Evaluation:**
    - Measure compression ratio, assess image quality using metrics.
    - Validate implementation, test with various image types, and optimize parameters for efficiency and quality.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction to DCT:

Discrete Cosine Transform (DCT) is a widely used technique in signal processing and image compression. It converts spatial-domain information into frequency-domain representation, facilitating efficient data compression while retaining essential image features. DCT breaks down an image into frequency components, with lower frequencies containing most image energy and higher frequencies representing fine details. This transformation reduces redundancy in the image data, allowing for high compression ratios with minimal loss of visual quality. DCT forms the basis of many popular image compression standards, including JPEG, and is implemented in various software tools like MATLAB for applications ranging from digital photography to video streaming. Its versatility, effectiveness, and computational efficiency make DCT a cornerstone in the field of image processing and compression.

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency.

This section describes the DCT and some of its important properties.

1) The One-Dimensional DCT:

The most common DCT definition of a 1-D sequence of length N is

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{\pi(2x+1)u}{2N}\right],$$

For u = 0, 1, 2, …, N −1. Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right]$$

for x = 0, 1, 2, …, N −1.

In both equations as above, α (u) is defined as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u \neq 0. \end{cases}$$

The Discrete Cosine Transform (DCT) is one of many transforms that takes its input and transforms it into a linear combination of weighted basis functions. These basis functions are commonly the frequency. The 2-D Discrete Cosine Transform is just a one-dimensional DCT applied twice, once in the x direction, and again in the y direction. One can imagine the computational complexity of doing so for a large image. Thus, many algorithms, such as the Fast Fourier Transform (FFT), have been created to speed the computation. The DCT equation (Eq.1) computes the i, jth entry of the DCT of an image.

2) The Two-Dimensional DCT:

The Discrete Cosine Transform (DCT) is one of many transforms that takes its input and transforms it into a linear combination of weighted basis functions. These basis functions are commonly the frequency. The 2-D Discrete Cosine Transform is just a one-dimensional DCT applied twice, once in the x direction, and again in the y direction. One can imagine the computational complexity of doing so for a large image. Thus, many algorithms, such as the Fast Fourier Transform (FFT), have been created to speed the computation. The DCT equation (Eq.1) computes the i, jth entry of the DCT of an image.

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \tag{1}$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases} \tag{2}$$

2.2 DCT Matrix

To get the matrix form of Equation (1), we will use the following equation, For an 8x8 block it results in this matrix:

$$T_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{cases}$$

For an 8x8 block it results in this matrix:

```
K =

    0.3536    0.3536    0.3536    0.3536    0.3536    0.3536    0.3536    0.3536
    0.4904    0.4157    0.2778    0.0975   -0.0975   -0.2778   -0.4157   -0.4904
    0.4619    0.1913   -0.1913   -0.4619   -0.4619   -0.1913    0.1913    0.4619
    0.4157   -0.0975   -0.4904   -0.2778    0.2778    0.4904    0.0975   -0.4157
    0.3536   -0.3536   -0.3536    0.3536    0.3536   -0.3536   -0.3536    0.3536
    0.2778   -0.4904    0.0975    0.4157   -0.4157   -0.0975    0.4904   -0.2778
    0.1913   -0.4619    0.4619   -0.1913   -0.1913    0.4619   -0.4619    0.1913
    0.0975   -0.2778    0.4157   -0.4904    0.4904   -0.4157    0.2778   -0.0975
```

2.2.1 Quantization

It is made up of three layers. These layers are :-

Our 8x8 block of DCT coefficients is now ready for compression by quantization. A remarkable and highly useful feature of the JPEG process is that in this step, varying levels of image compression and quality are obtainable through selection of specific quantization matrices. This enables the user to decide on quality levels ranging from 1 to 100, where 1 gives the poorest image quality and highest compression, while 100 gives the best quality and lowest compression. As a result, the quality/compression ratio can be tailored to suit different needs.

Subjective experiments involving the human visual system have resulted in the JPEG standard quantization matrix. With a quality level of 50, this matrix renders both high compression and excellent decompressed image quality

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

If, however, another level of quality and compression is desired, scalar multiples of the JPEG standard quantization matrix may be used. For a quality level greater than 50 (less compression, higher image quality), the standard quantization matrix is multiplied by (100-quality level)/50.

For a quality level less than 50 (more compression, lower image quality), the standard quantization matrix is multiplied by 50/quality level. The scaled quantization matrix is then rounded and clipped to have positive integer values ranging from 1 to 255. For example, the following quantization matrices yield quality levels of 10 and 90.

$$Q_{10} = \begin{bmatrix} 80 & 60 & 50 & 80 & 120 & 200 & 255 & 255 \\ 55 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

$$Q_{90} = \begin{bmatrix} 3 & 2 & 2 & 3 & 5 & 8 & 10 & 12 \\ 2 & 2 & 3 & 4 & 5 & 12 & 12 & 11 \\ 3 & 3 & 3 & 5 & 8 & 11 & 14 & 11 \\ 3 & 3 & 4 & 6 & 10 & 17 & 16 & 12 \\ 4 & 4 & 7 & 11 & 14 & 22 & 21 & 15 \\ 5 & 7 & 11 & 13 & 16 & 12 & 23 & 18 \\ 10 & 13 & 16 & 17 & 21 & 24 & 24 & 21 \\ 14 & 18 & 19 & 20 & 22 & 20 & 20 & 20 \end{bmatrix}$$

Quantization is achieved by dividing each element in the transformed image matrix D by corresponding element in the quantization matrix, and then rounding to the nearest integer value. For the following step, quantization matrix Q50 is used.

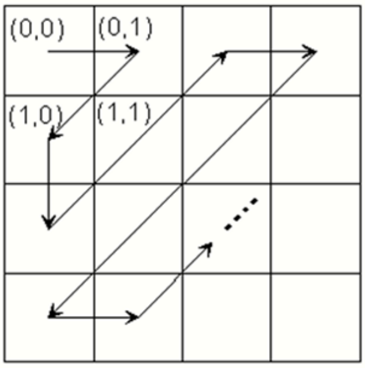
$$C_{i,j} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right)$$

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The coefficients situated near the upper-left corner correspond to the lower frequencies œ to which the human eye is most sensitive œ of the image block. In addition, the zeros represent the less important, higher frequencies that have been discarded, giving rise to the lossy part of compression. As mentioned earlier, only the remaining nonzero coefficients will be used to reconstruct the image. It is also interesting to note the effect of different quantization matrices; use of Q10 would give C significantly more zeros, while Q90 would result in very few zeros.

2.2.2 Coding:-

The quantized matrix C is now ready for the final step of compression. Before storage, all coefficients of C are converted by an encoder to a stream of binary data (01101011...). In-depth coverage of the coding process is beyond the scope of this article. However, we can point out one key aspect that the reader is sure to appreciate. After quantization, it is quite common for most of the coefficients to equal zero. JPEG takes advantage of this by encoding quantized coefficients in the zig-zag sequence shown in Figure as under. The advantage lies in the consolidation of relatively large runs of zeros, which compress very well. The sequence in Figure 1(4x4) continues for the entire 8x8 block.



2.2.3 Decompression:-

Reconstruction of our image begins by decoding the bit stream representing the Quantized matrix C. Each element of C is then multiplied by the corresponding element of the quantization matrix originally used

$$R_{i,j} = Q_{i,j} \times C_{i,j}$$

$$R = \begin{bmatrix} 160 & 44 & 20 & 80 & 24 & 0 & 0 & 0 \\ 36 & 108 & 14 & 38 & 26 & 0 & 0 & 0 \\ -98 & -65 & 16 & -48 & -40 & 0 & 0 & 0 \\ -42 & -85 & 0 & -29 & 0 & 0 & 0 & 0 \\ -36 & 22 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The IDCT is next applied to matrix R, which is rounded to the nearest integer. Finally, 128 is added to each element of that result, giving us the decompressed JPEG version N of our original 8x8 image block M.

2.2.4 COMPARISON OF MATRICES:

Let us now see how the JPEG version of our original pixel block compares, Original and decompressed matrices --

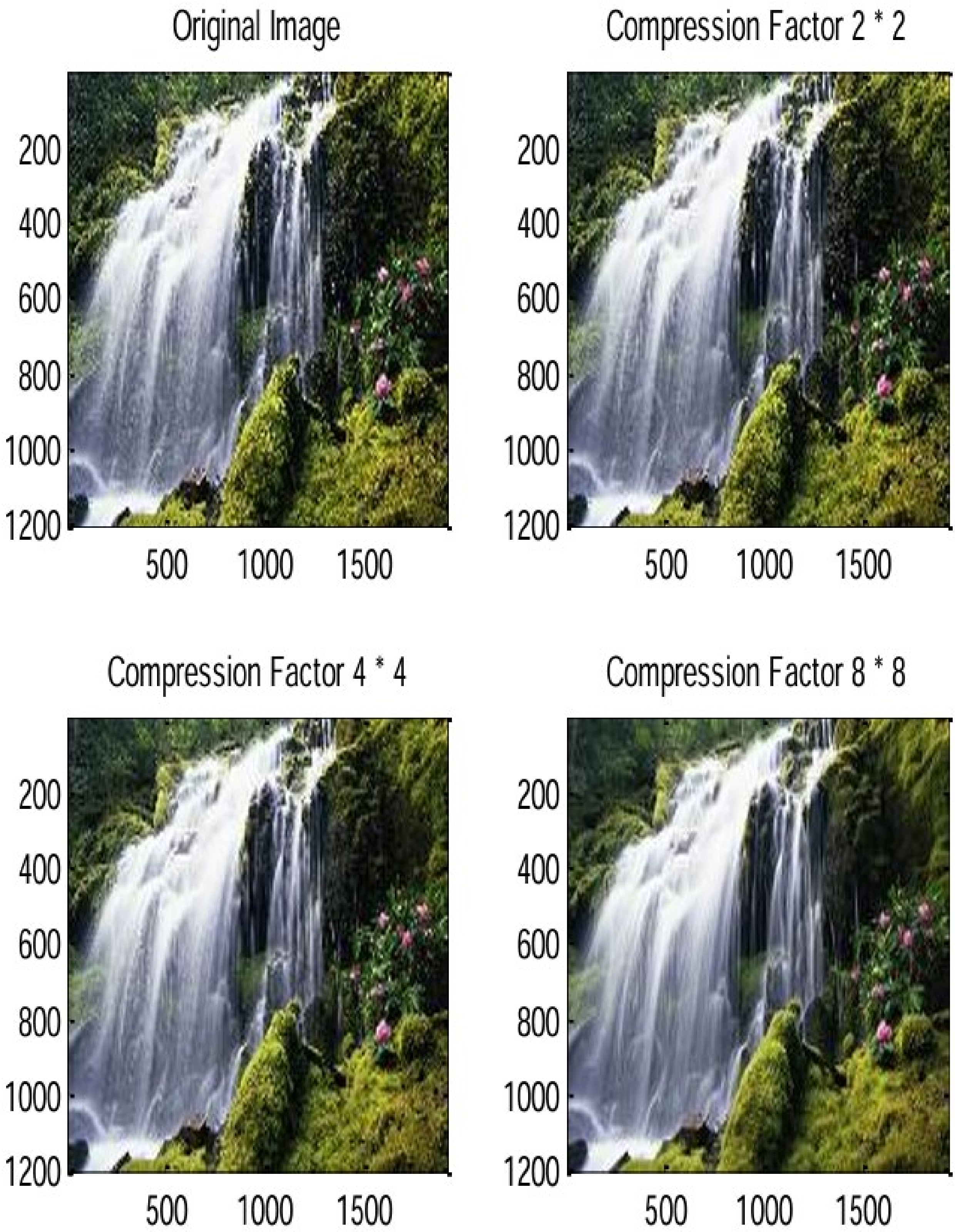
$$\begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

$$\begin{bmatrix} 149 & 134 & 119 & 116 & 121 & 126 & 127 & 128 \\ 204 & 168 & 140 & 144 & 155 & 150 & 135 & 125 \\ 253 & 195 & 155 & 166 & 183 & 165 & 131 & 111 \\ 245 & 185 & 148 & 166 & 184 & 160 & 124 & 107 \\ 188 & 149 & 132 & 155 & 172 & 159 & 141 & 136 \\ 132 & 123 & 125 & 143 & 160 & 166 & 168 & 171 \\ 109 & 119 & 126 & 128 & 139 & 158 & 168 & 166 \\ 111 & 127 & 127 & 114 & 118 & 141 & 147 & 135 \end{bmatrix}$$

2.3 Conclusion

If we look at the above two matrices, this is a remarkable result, considering that nearly 70% of the DCT coefficients were discarded prior to image block decompression/reconstruction. Given that similar results will occur with the rest of the blocks that constitute the entire image, it should be no surprise that the JPEG image will be scarcely distinguishable from the original. Remember, there are 256 possible shades of gray in a black-and-white picture, and a difference of, say, 10, is barely noticeable to the human eye. DCT takes advantage of redundancies in the data by grouping pixels with similar frequencies together. And moreover, if we observe as the resolution of the image is very high, even after sufficient compression and decompression there is very less change in the original and decompressed image. Thus, we can also conclude that at the same compression ratio the difference between original and decompressed image goes on decreasing as there is increase in image resolution.

DCT to rows*column:



2.4 Previous Research

Author	Compression Technique	Compression Method	Description
Clairaut, Lagrange, and Gauss	Lossless	DFT	Transforms spatial information into frequency domain, enabling analysis and manipulation based on frequency components
G. Sadashivappa, and K. V. S. AnandaBabu	Lossy	Sub-band coding	Explore diverse wavelet functions, analyse properties, implement SPIHT compression in MATLAB.
Somasundaram and Vimala	lossy	block truncation coding	EBTC reduces bit rate leveraging inter-pixel redundancy in compression.
Aditya Kumar and Pardeep Singh	lossy	block truncation coding	The improved algorithm deals with the grayscale image to reduce the

.

REFERENCES

- [1] Digital Image Processing by “Gonzalez and Woods”.
- [2] IEEE Journals and Research papers.
- [3] Mathworks Matlab Central.
- [4] Wikipedia.org