IMAGE COMPRESSION USING DISCRETE COSINE TRANSFORM (DCT)

B.Tech Major 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

Bihar Engineering University, Patna

IMAGE COMPRESSION USING DISCRETE COSINE TRANSFORM (DCT)

B.Tech Major 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

August 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.

LIST OF FIGURES

Figure 3.3 File Format 18

Figure 4.2 Two-Dimensional DCT **21**

Figure 4.3 flowcharts 22

Figure 4.6 DCT Coding **26**

Figure 4.9 DCT Algorithm and Block Diagram **28**

Figure 5.1 Images - Input/Output/Implementation **31**

Figure 5.3 DCT to Rows x Column 34

Figure 6.1 Satellite Imagery **35**

Figure 6.2 Facial Recognition **36**

Figure 6.3 Digital Photography **37**

ABBREVIATIONS

DCT	Discrete Cosine Transformation
DWT	Discrete Wavelet Transformation
DFT	Discrete Fourier Transformation
FFT	Fast Fourier Transformation
FWT	Fast Wavelet Transformation
JPEG	Joint Photographic Expert Group
JPEG-2000	Joint Photographic Expert Group-2000
MPEG	Moving Pictures Experts Group
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
SNR	Signal-to-noise Ratio
ISO	International Standards Organization
LOT	Lapped Orthogonal Transforms
IEC	International Electro-Technical Commission
DPCM	Discrete pulse code modulation
FAX	Facsimile transmission
KLT	Karhunen Lòeve Transform
IDCT	Inverse Discrete Cosine Transform
BTC	Block Truncation coding

TABLE OF CONTENTS

First Page i

Certificate ii

Declaration iii

Acknowledgement iv

Abstract v

1. Introduction **01**

1.1 Background and Motivation 01

1.2 Objective and Scope 01

1.3 Problem Statement 02

1.4 Research Question 02

1.5 Methodology 03

2. Literature Review **04**

3. Analysis and Discussion **07**

3.1 Basic concept of Image 07

3.2 Principle of Image Compression 07

3.3 Need of Image Compression 08

3.4 Redundancy of Image Compression 08

3.5 File Format 09

4. Methodology

4.1 Introduction to DCT 10

4.2 Image Compression 10

4.2.1 The One-Dimensional DCT 10

4.2.2 The Two-Dimensional DCT 11

4.3 Flowchart 12

4.4 DCT Matrix 13

4.5 Quantization 14

4.6 Coding 15

4.7 Decompression 15

4.8 Comparison of Matrices 16

4.9 Algorithm & Block Diagram of DCT 17

5. Result **19**

5.1 Images – Input/Output/ Implementation.. 21

5.2 Code Explanation 22

5.3 DCT to Rows x Column 24

6. Applications... **25**

6.1 Remote Sensing and Satellite Imaging... 25

6.2 Facial Recognition Systems 26

6.3 Digital Photography 27

7. Conclusion and Scope for Future Works **29**

REFERENCES **32**

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

1. Prof. Sumit Gupta & Roopesh Kumar Kurmi, International Journal of Engineering and Advanced Technology (UEAT) ISSN: 22498958, Volume-6 Issue-6, August 2017. Department of Computer Science & Engineering, Lakshmi Narain College of Technology Excellence, Bhopal (M.P)-462021, India. Their article topic is "A Review of Lossless and Lossy Based Image Compression Techniques." In this journal I observe that the lossless and lossy image compression is based on DCT (Discrete cosine transform) and DWT (Discrete wavelet transform) in which Discrete cosine transform have the very low PSNR and very low compression ratio. And the Discrete wavelet transform have high PSNR then the DCT and also have high compression ratio then DCT.

2. Rawsam Abdaladheem Hasan, European Scientific Journal November 2014 edition vol. 10, No.33 ISSN: 1857-7881 (Print) e-ISSN 1857-7431. Computer Science Dept. Faculty of Science Mustansiriyah University Baghdad Iraq. The Topic of the Article is "Combination of Lossy and Lossless for Image Compression". In this paper, 1 observe that the image compression algorithm depends on discrete wavelet transform and entropy coding is suggested. Two images of only (bmp format) image format and resolution is fixed in (256x256) pixels are used as samples to test the algorithm. Output result of the image are in the form of PSNR which is very low and compression ratio are very low.

3. Mr. Amit G. Kadam & Prof. Neeta Pingle. International Journal of Engineering Sciences & Research Technology, 2018, 7(2), 232-240. MIT College of Engineering, Aurangabad, Maharashtra, India*2 Professor, MIT College of Engineering, Aurangabad, Maharashtra, India they write a journal of topic "Overview Of SPIHT Based Image Compression Algorithm" in International Journal Of Engineering Sciences & Research Technology Of I observed the algorithms of image compression also observe that Observe that Image compression algorithms based on Embedded Zero Wavelet (EZW) SPIHT algorithm mainly depends upon its three lists LIP, LIS and LSP. To achieve Image Compression using discrete wavelet transform. DWT is used to separate the image into a pixel. To achieve image decompression using IDWT.

4. M.Deriche et al proposed "a new wavelet-based environment friendly image compression algorithm using compressive sensing" in 2015, which is based totally on compressive sensing. Its begins with a customary multilevel 2-D Wavelet deterioration, which offers a reduced portrayal of picture pixels. The proposed wavelet-based CS recreation, with the standardized estimation grid, brings about execution increment contrasted with different

regular CS-based methods. The proposed approach gives a totally new system for using CS in the wavelet area. To assess the execution of the proposed calculation, we began by way of choosing ten dim scale test photographs of various sorts (BMP, PNG, PGM) oftentimes utilized as a part of benchmarking. Exploratory outcomes show that the proposed calculation, accomplishes better remaking quality, at a similar estimation proportion, contrasted with reducing edge CS-based picture strain strategies.

5. Priya Bajpai et al has presented Algorithm for Image Compression in Image Processing on 2017. In this paper, alternate in the WDR calculation is been proposed and change depends on desire calculation. The WDR is the productive procedure in which the entire photograph is partitioned into little network and lattice which has unique homes are expelled from the picture. The proposed method has been actualized in MATLAB by taking the informational index of 10 photograph which are in the grey scale. Hence this enhanced calculation is better because its pressure proportion is higher.

6. Mander et al proposed an algorithm of "Multiplied image compression decompression technique the usage of block truncation and Wavelets" on 8 Aug, 2017. The paper centers round pressure method referred to as rectangular truncation coding as it helps in diminishing the span of the picture with the goal that it consumes much less room in memory and simple to transmit and moreover gives the quantized reproduced picture in a quick timeframe. Along these lines, BTC is utilized to pack grey scale pictures. After pressure Discrete Wavelet Change with spine addition is connected to remake the pictures. The proposed philosophy is tried upon a few prestigious benchmark photographs and their PSNR esteems have been recorded which brings about preferable quality pictures over in current strategies as the distinction acquired is 43% of PSNR esteems when computed as a normal. The proposed technique can be additionally utilized as a phase of packing and sending pictures among sensors dispatched to get the data gathered through pictures.

7. Veerpal Kaur, Gurwinder Kaur: They proposed a new method of compression which consists of the combination of "Discrete Wavelet, Discrete Cosine and Huffman" compression schemes. They show that DWT and DCT is very proper to cope up compression ratio but as they are lossy techniques so their exceptional measurement which they concluded with the help of PSNR is reducing due to so, further to enhance CR. they are the use of Huffman compression method because of its lossless compression nature and it will grant them good PSNR and high CR value. This concludes that after making use of lossy techniques it's better to use lossless too to beautify compression at same PSNR.

8. Ridhi Jindal in 2017 named "Digital image compression technology" for the rapid transmission and continuous preparing of computerized photo data on the web. Albeit still image pressure is a strategy produced for pretty a while, in addition to there are a few methodologies which lessen the pressure rate, and quicken calculation time, there are nonetheless a ton to go to enhance the adequacy of pressure. This paper is an examination of different stress methods can be connected to a range of kinds of pictures. By breaking down the focal points and detriments of a number of systems, we can pick the right approach that can be utilized for picture pressure.

9. In the year 2017 [30], Ahmed and George proposed a low-cost lossy compression for a color image in a study. The data of the RGB image is transformed to YUV color space, and then U and V bands are down-sampled by the propagation step. Each color sub-band is decomposed separately using the biorthogonal wavelet transform. The Low-Low (LL) sub-band is then encoded using the (DCT). Scalar Quantization is used to code the remaining wavelet sub-bands. The quadtree coding method was also used to code the results of DCT and quantization procedures. Finally, adaptive shift coding is employed as a high-order entropy encoder to remove any statistical redundancy and boost compression efficiency. The system was put to the test on a series of standard color photographs, and the compression results revealed that it was capable of reducing the size while keeping fidelity levels above the acceptable level, with compression ratios of around 1:30 for Color Barbara and 1:40 for Color Lena.

10. Ariatmanto and Ernawan [35] in the year 2020 proposed a new scaling factor for selected Discrete Cosine Transform (DCT) coefficients in image watermarking, where these factors employ particular guidelines to reduce distortion. Image blocks with the lowest pixel variances are chosen as embedding places. The best image quality is used to determine the ideal scaling factors for specified DCT coefficients on the middle frequencies. the scaling factors are used to accomplish the embedding procedure, the results indicate that the proposed method achieves higher Normalized Cross-Correlation (NC) values of watermark recovery against various attacks than existing schemes, also this scheme maintains watermarked images with a PSNR value of 45 dB in quality.

CHAPTER 3: INTRODUCTION

3.1 Basic Concepts of Image:

Image formed with the combination of dots which is called pixels. Pixel can also be defined as tiniest element which is capable of being addressed. Image size contains number of pixels which consists of width and height. Pixels of image have certain color for example: 8 bits for each pixel, which is sufficient to speak to each shade of dim that a human eye can recognize. When colored pictures are chosen for analysis, then it became more difficult to deal with. It is calculated by the depth of image which consists of number of bits per pixel. A bit plane of n-bits can have 2n colors. But human eye is capable to differentiate 224 colors, albeit some claim that the quantity of hues the eye can recognize is substantially higher. The most widely recognized shading profundities are 8, 16, and 24.two prominent approaches to distinguish shading data in a picture

• To begin with path is to speak to every pixel's shading is by giving a requested three times of number, that is the mix of red, green, and blue. This is known as colored (RGB) image.

3.2 Need of Image Compression

Uncompressed images occupy lot of space and difficult to store and transmit or download. So we opted an image compression technique. Uncompressed image takes lot of space of your memory so we compress the image data to store and transmission.

Compressed documents require considerably much less storage capability than uncompressed files that mean an enormous minimize in prices for storage. A compressed file additionally requires much less time for switch whilst ingesting much less community bandwidth.

Photograph compression is to reduce irrelevance and redundancy of the image information to be able to store or transmit statistics in an efficient form. It is concerned with minimizing the wide variety of bits required to represent an image. Image compression may be lossy or lossless. Image compression is minimizing the measurement in bytes of a graphics file without degrading the pleasant of the image to an unacceptable level. The reduction in file measurement allows more images to be stored in a given amount of disk or reminiscence space.

Digital Image Compression compresses and reduces the size of images with the aid of use of various algorithms and standards. The lossless compression technique, as the identify indicates, produces no loss in the quality of image. This technique is used in locations where the quality and accuracy of photograph is extremely important and can't be compromised on. Some of the examples are technical drawings, medical images etc. Lossy compression approach is one which produces a minor loss of quality to the output image. This minor loss is almost invisible and difficult to identify. This technique finds use where minor alteration or loss of first-rate causes no problem like in photographs. There are exclusive methods and algorithms used in lossless and lossy compression.

3.3 File Format

File formats are used to represent an image, out of many formats, these two popular formats are mainly used. Which are as given in fig1.3

• GIF (Graphics Interchange Format) The most widely recognized picture design on web is 1 to 8-bit shading or greyscale pictures.

• TIFF (Tagged Image File Format) The standard picture organizes found in most paint, imaging, and work area distributing programs.

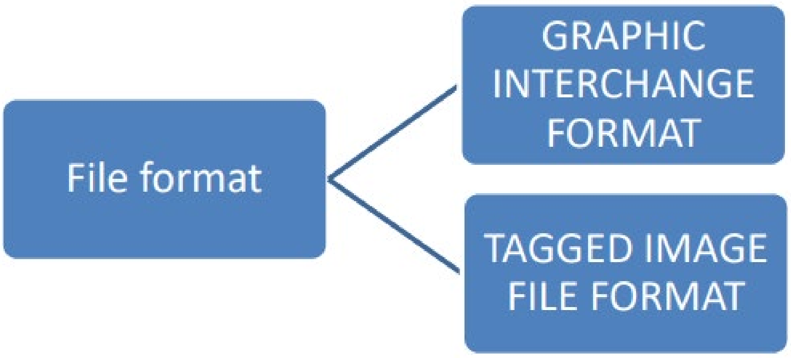


Fig 3.3: Types of file format

3.4 Principle of Image Compression:

Image compression reduces the size of image files by eliminating redundant or less important information. The core principle involves transforming the image data into a more compact format, often through techniques like Discrete Cosine Transform (DCT) or wavelet transforms. These methods exploit the fact that human vision is less sensitive to certain details, allowing for the reduction of image data without significant loss in perceived quality. Image compression is essential for efficient storage and transmission, especially in digital media, where bandwidth and storage space are limited.

3.5 Redundancy in Image Compression:

Redundancy in image compression refers to the presence of repetitive or predictable information within an image that can be reduced or removed without affecting the overall visual quality. There are three types of redundancy: spatial, spectral, and temporal. Spatial redundancy occurs when neighboring pixels have similar values, spectral redundancy is due to correlations between different color channels, and temporal redundancy arises in sequences of images (like videos) where consecutive frames are similar. Compression algorithms leverage these redundancies to decrease file size, making image storage and transmission more efficient.

CHAPTER 4: METHODOLOGY

4.1 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.

4.2 Types of DCT

4.2.1 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

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

4.2.2 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.

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

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

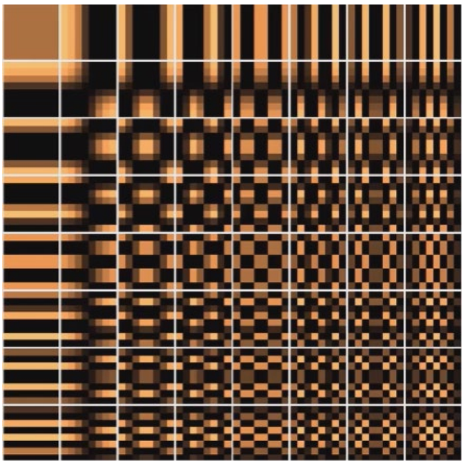


Figure **4.2** Two-Dimensional DCT

4.3 Flowchart of proposed Method

a) Image compression

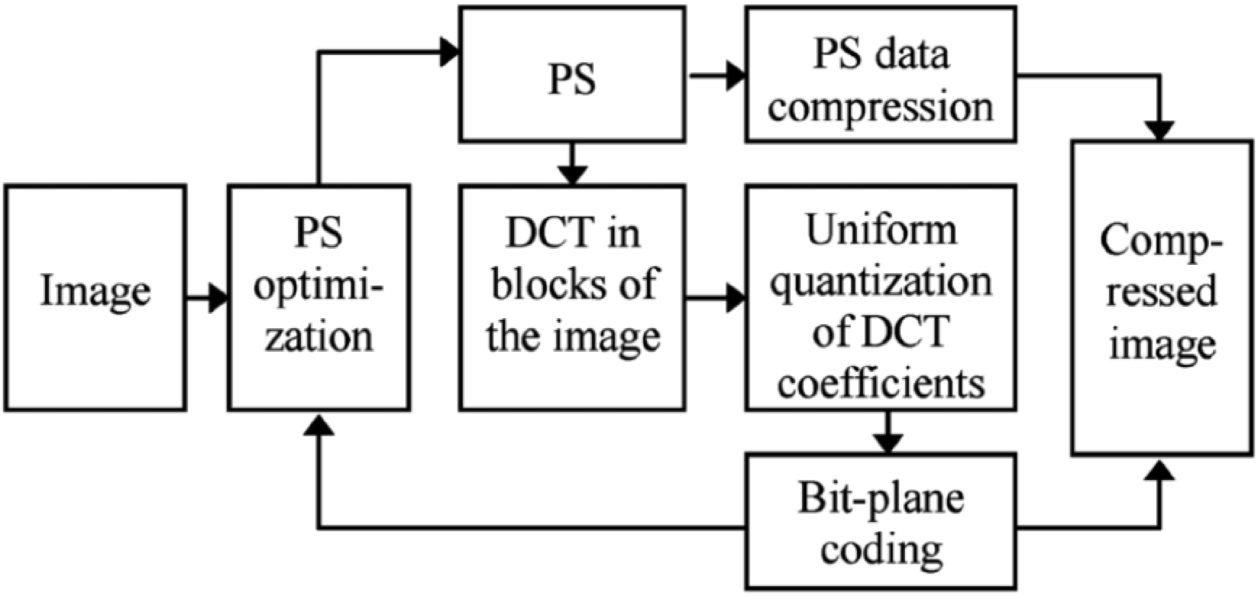


Figure 4.3 flowchart of Image compression

This flowchart illustrates the process of image compression using the Discrete Cosine Transform (DCT). The process begins with the input image, which undergoes PS optimization to enhance the image data before compression. The optimized image is then divided into blocks, and the DCT is applied to each block to convert the image from the spatial domain to the frequency domain, where most of the image's energy is concentrated in lower frequencies.

Following the DCT, uniform quantization is performed on the DCT coefficients, reducing the precision of less significant components to decrease the data size. The quantized coefficients are then encoded using bit-plane coding, which compresses the binary representations of the image. An additional PS data compression step is applied to further reduce redundancy and achieve a higher compression ratio. The process concludes with the output of the compressed image, which is significantly smaller in size compared to the original, with minimal loss of quality.

b) Image decompression.

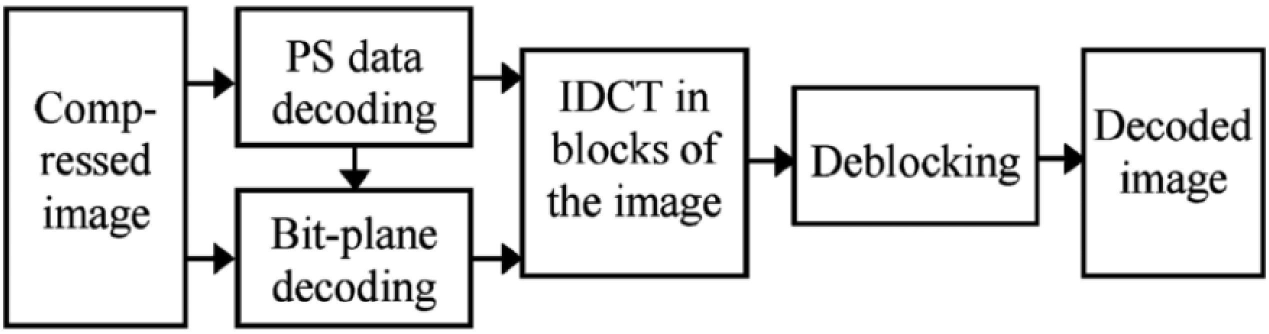


Figure 4.3 flowchart of Image compression

This flowchart illustrates the process of decoding a compressed image that was initially compressed using techniques like DCT (Discrete Cosine Transform). The process begins with the compressed image as the input.

The first step involves PS data decoding, where the previously compressed data is decompressed. Simultaneously, bit-plane decoding is performed to decode the binary representations of the quantized DCT coefficients.

Once the data is decoded, the image undergoes IDCT (Inverse Discrete Cosine Transform) in blocks, which transforms the frequency domain data back into the spatial domain, reconstructing the image blocks from the DCT coefficients. The resulting image may still contain some artifacts, particularly blocking artifacts due to the block-based nature of DCT. Therefore, a deblocking step is applied to smooth out these block boundaries and improve the visual quality of the image. Finally, the process yields the decoded image, which closely resembles the original image before compression.

4.4 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
```

4.5 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.

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.

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

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

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

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

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.

4.6 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.

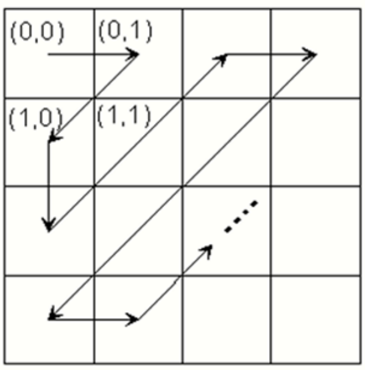


Figure 4.6 DCT Coding

4.7 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.

4.8 COMPARISON OF MATRICES:

Matrices are crucial in representing images and applying various transformations that lead to compression. The comparison of matrices in this context often involves evaluating different techniques like the Discrete Cosine Transform (DCT), Singular Value Decomposition (SVD), and Wavelet Transform, which utilize matrices differently. The DCT transforms an image into a sum of cosine functions at different frequencies. The image is first divided into blocks (commonly 8x8), and each block is represented by a DCT matrix.

DCT works by concentrating the image's energy into a few coefficients (typically the upper-left corner of the matrix), allowing the rest to be discarded or quantized with minimal loss in quality.

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

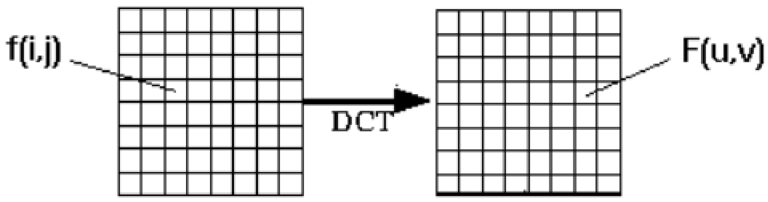
$$\begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198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

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

The wavelet transform applies a series of filters to the image matrix to produce different frequency sub-bands. The image is transformed into a matrix of coefficients representing these sub-bands. The transform provides both spatial and frequency information. By discarding the less significant coefficients, the image can be compressed.

4.9 DCT Algorithm and Block Diagram:

'The discrete cosine transforms (DCT)' is a method which can convert signals into frequency components. It is commonly used in compressing the images sizes.



Less significant coefficients are produced in DCT, which inevitably generates better larger compression. DCT usually works on 1-dimendional data. Though image data is presented in 2-dimension blocks, therefore summing term is introduced to the DCT. After which DCT becomes 2-dimensional equation. It implies that the DCT which is 1- dimensional is applied two times, once in 'x direction' and further in 'y direction'. Which leads to an efficient 2-dimensional Discrete Cosine Transform
Few simple functions can be developed to cipher the DCT and to compressing images. MATLAB systematically implements DCT and modifications can be done to the code, for it to work in a better fashion. MATLAB has an IMAQ block which analyses and investigates the result of 'Image Compression using DCT'. Resulting picture and error picture is also shown.

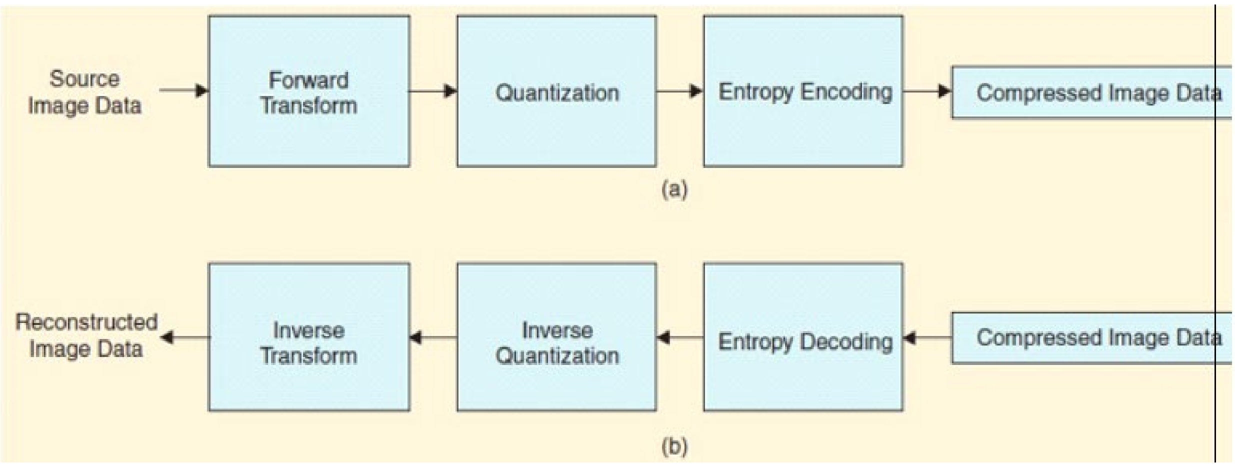


Fig 4.9 : Block diagram for DCT

A 2-dimensional DCT is studied everywhere. In the process, the original old image is altered to 8 x 8 blocks form and then Forward Transformations (DWT, DFT) are applied to filter out the image and take out the required information. After this is the Quantization which is done by Quantizer , followed by Entropy Encoding. Where redundancy is found and removed. Here, there is deconstruction of image data. Afterwards the process of reconstructing compressed image data starts with same components. Then inverse 'Discrete Cosine Transform' is being applied to blocks to give reconstructed picture. This technique is used primarily in reducing space which is occupied by unimportant bits and costs. ' The image compression techniques' can be split into two broad sections: lossy and lossless. In lossy image compression the original digital image is usually a bit distorted as quality is not maintained and there is data loss. In lossless image compression, the original image quality is maintained and image size is still reduced. Quantization is a lossy process and reduces information colour bits. 'Entropy coding' is a mode which illustrates the 'quantized coefficients' as concisely as possible.

CHAPTER 5: MATLAB CODE & RESULT

```
clc;
I1 = imread(tiger.jpg');
imshow(I1);
Size12 = dir(tiger.jpg');
Size1 = (Size12.bytes)/1024;
    %%For first channel
      %%
      I = I1(:,:,1);
      I = im2double(I);
      T = dctmtx(8);
      B = blkproc(I,[8 8],'P1*x*P2',T,T');



      mask = [1   1   1   1   0   0   0   0
         1   1   1   0   0   0   0   0
         1   1   0   0   0   0   0   0
         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   0
         0   0   0   0   0   0   0   0];
      B2 = blkproc(B,[8 8],'P1.*x',mask);
      I2 = blkproc(B2,[8 8],'P1*x*P2',T',T);
      %%For second channel
      %%
      I = I1(:,:,2);
      I = im2double(I);
      T = dctmtx(8);
      B = blkproc(I,[8 8],'P1*x*P2',T,T');
```

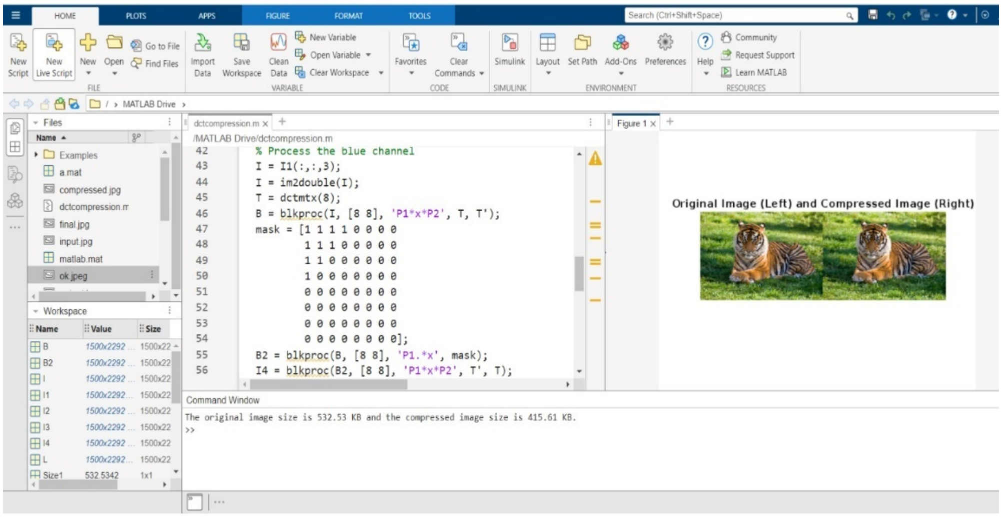
```
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   0   0   0   0   0   0
        1   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I3 = blkproc(B2,[8 8],'P1*x*P2',T',T);
%% 3 rd channel %%
I = I1(:,:,3);
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T');
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   0   0   0   0   0   0
        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   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I4 = blkproc(B2,[8 8],'P1*x*P2',T',T);
%%
L =cat(3,I2,I3,I4);
 imshow(L);

imwrite(L,'final.jpg');

size23= dir('final.jpg');
Size2 = (size23.bytes)/1024;
fprintf('The original is %f and compressed is %f',Size1,Size2);
```

5.1 Images

Input Image

Output Image

Implementation



5.2 Code Explanation

This MATLAB code performs image compression using the Discrete Cosine Transform (DCT) on a color image. Here’s a step-by-step explanation of the process:

1. Reading the Image: The code starts by reading an image file named tiger.jpg' using the imread function and displays it with imshow. The size of this original image file is computed and converted from bytes to kilobytes for later comparison.

2. Processing Each Colour Channel: The image is in RGB format, so the code processes each of the three color channels (red, green, and blue) separately:

-Red Channel:

- The red channel of the image is extracted and converted to a double precision format using im2double.
- The 8x8 DCT matrix (dctmtx(8)) is applied to the image in blocks using the blkproc function. This function performs a block-based operation where each 8x8 block of the image is transformed using DCT.
- A mask is applied to zero out high-frequency DCT coefficients to compress the image, which essentially removes less important information.
- The inverse DCT is then applied to the masked DCT coefficients to reconstruct the compressed image of the red channel.

- Green Channel:
  - The same process is repeated for the green channel, following the same steps as with the red channel.

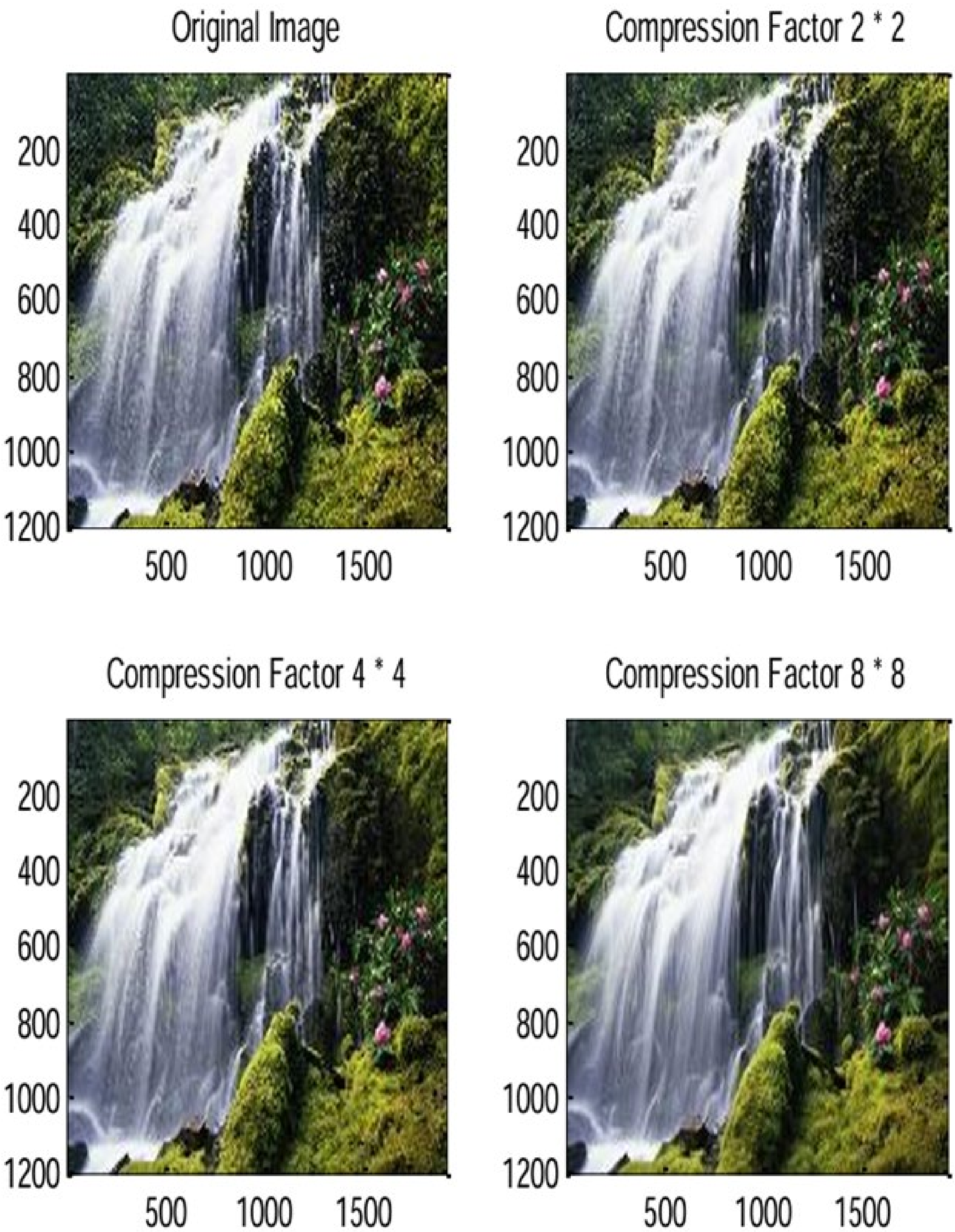
- Blue Channel:
  - The blue channel is processed similarly to the red and green channels.

3. Reconstructing the Image: After processing all three channels, the modified channels are combined back into a single image using the cat function. This combined image is then displayed and saved as 'final.jpg'.

4. Size Comparison: The size of the compressed image ('final.jpg') is computed and converted from bytes to kilobytes. The code then prints out the sizes of the original and compressed images, allowing for a comparison of the compression effectiveness.

In summary, the code compresses a color image by applying DCT to each channel, masking high-frequency coefficients to reduce file size, and then reconstructing and saving the compressed image.

5.3 DCT to rows*column:



CHAPTER 6: APPLICATIONS

6.1 Remote Sensing and Satellite Imaging

Remote sensing and satellite imaging are essential tools for monitoring and analyzing the Earth's surface, atmosphere, and oceans. These systems generate vast amounts of high-resolution image data, which must be efficiently transmitted, stored, and processed. Discrete Cosine Transform (DCT) is a key technology that enables the compression of this data, making it manageable for transmission and storage while preserving critical information.

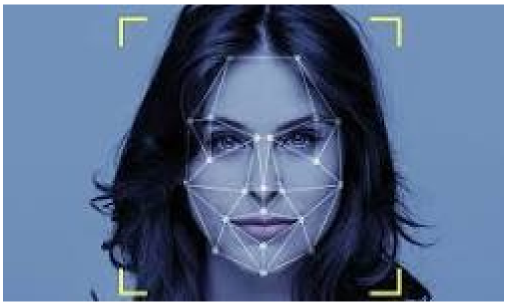
In remote sensing, satellites equipped with various sensors capture detailed images of the Earth's surface. These images are used for a wide range of applications, including environmental monitoring, urban planning, disaster management, and military reconnaissance. However, the sheer volume of data captured by these sensors presents a significant challenge. Without effective compression, transmitting this data from the satellite to ground stations would require enormous bandwidth, and storing it would demand extensive storage capacity.

DCT addresses these challenges by compressing the image data before transmission. The process begins with the division of the captured image into smaller blocks, typically 8x8 pixels. DCT is then applied to each block, transforming the spatial image data into the frequency domain. In this domain, most of the image's important information is concentrated in a few low-frequency components, while the high-frequency components, which often represent less critical details, are reduced or discarded.

By focusing on the low-frequency components, DCT achieves significant compression of the image data. This compression reduces the amount of data that needs to be transmitted, making it possible to send high-resolution images over limited bandwidth. For satellite imaging, where transmission distances are vast and communication channels are often bandwidth-constrained, this is a critical advantage.

Once the compressed data is received by ground stations, it can be decompressed and analyzed. Despite the compression, DCT ensures that the key features of the image, such as edges, textures, and patterns, are preserved, allowing for accurate analysis and interpretation. This is particularly important in applications like environmental monitoring, where changes in land cover, vegetation, and water bodies need to be detected with high precision.

6.2 Facial Recognition Systems



Facial recognition systems are increasingly prevalent in security, authentication, and identification applications. These systems rely on efficiently processing and storing vast amounts of facial image data, often under stringent constraints on storage space and computational resources. Discrete Cosine Transform (DCT) plays a crucial role in enabling the effective functioning of these systems by compressing facial images and extracting essential features.

In facial recognition systems, the first step typically involves capturing an image of a person's face. This image contains a large amount of data, much of which may be redundant or irrelevant for the purpose of identifying the individual. To reduce this data and focus on the most critical information, the system applies DCT to the image. DCT transforms the image from the spatial domain to the frequency domain, where it is easier to identify and compress significant features, such as edges, textures, and patterns that are important for distinguishing one face from another.

DCT's ability to concentrate most of the image's energy into a few low-frequency components is particularly useful for facial recognition. By focusing on these components, the system can effectively reduce the dimensionality of the data, retaining the most important information while discarding less critical details. This compression not only reduces the storage requirements but also speeds up the processing of facial images, allowing for faster recognition and matching.

Moreover, DCT facilitates the storage and retrieval of large facial image datasets. In security and surveillance systems, where thousands or even millions of facial images need to be compared in real-time, the use of DCT ensures that these comparisons can be performed efficiently. The compressed data can be stored in databases with minimal storage overhead, making it possible to maintain extensive records of individuals without overwhelming the system's storage capacity.

6.3 Digital Photography

DCT is the cornerstone of efficient image storage and transmission in digital photography.

- Compression: By transforming image data from spatial to frequency domain, DCT reveals that most image information is concentrated in low-frequency components. This allows for significant data reduction by quantizing and discarding less important high-frequency coefficients. This compression is crucial for storing and sharing high-resolution images efficiently.
- Faster Processing: Compressed images require less storage space, which translates to faster loading times and smoother display on digital devices. This is especially beneficial for mobile photography and online platforms.

- Efficient Transmission: Smaller file sizes mean quicker upload and download times, making image sharing across networks more efficient. This is crucial for platforms like social media and image-sharing websites.

- Storage Optimization: DCT-based compression helps maximize storage capacity on cameras, smartphones, and computers, allowing photographers to capture and store more images.

In essence, DCT empowers photographers to capture, store, and share high-quality images without compromising on speed or storage efficiency.

CHAPTER 7: 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.

FUTURE SCOPE

The future scope of image compression is vast and evolving, driven by advancements in technology, increasing data demands, and the need for efficient storage and transmission solutions.

Here are several key areas where the future of image compression is heading:

1. **Enhanced Compression Algorithms**

Deep Learning-Based Approaches: The integration of deep learning and artificial intelligence (AI) is transforming image compression. Techniques like convolutional neural networks (CNNs) and generative adversarial networks (GANs) are being used to develop new compression methods that can outperform traditional algorithms like JPEG and JPEG2000 in terms of compression efficiency and image quality.

Learned Compression: Algorithms that learn optimal compression strategies from data are emerging. These learned approaches adapt to the characteristics of the image content, leading to potentially higher compression ratios and better quality preservation.

2. **Improved Compression Standards**

Next-Generation Standards: The development of new compression standards, such as High Efficiency Image Coding (HEIC), which is based on High Efficiency Video Coding (HEVC), is expected to provide significant improvements in compression efficiency compared to existing standards. Future standards may continue to build on these advancements, incorporating more sophisticated techniques and optimizations.

Versatile Video Coding (VVC): As the successor to HEVC, VVC is being designed to improve compression efficiency further, which could influence future image compression standards.

3. **Real-Time and Adaptive Compression**

Real-Time Compression: With the growing demand for real-time image processing in applications like streaming, augmented reality (AR), and virtual reality (VR), there is a focus on developing compression algorithms that can handle high-resolution images and videos with minimal latency.

Adaptive Compression: Adaptive algorithms that adjust compression settings based on network conditions, content type, or user preferences are becoming increasingly important. These algorithms can optimize quality and compression ratio dynamically.

4. **Compression for Emerging Technologies**

4K and 8K Resolution: As display technologies and cameras evolve towards higher resolutions, such as 4K and 8K, compression algorithms need to handle larger data volumes efficiently while maintaining high quality.

High Dynamic Range (HDR): HDR imaging requires advanced compression techniques to manage the increased range of brightness and color information, ensuring that the high dynamic range is preserved during compression.

5. **Integration with Other Technologies**

Edge Computing: The integration of image compression with edge computing is expected to enhance processing efficiency by performing compression tasks closer to where the data is generated, reducing the need for high-bandwidth data transmission.

Blockchain and Security: As data security becomes more critical, integrating image compression with blockchain technology could provide secure and tamper-proof ways of handling compressed image data.

6. **Computational Efficiency and Hardware**

Hardware Acceleration: Advances in hardware, such as specialized processors and GPUs, are enabling faster and more efficient image compression. Future developments may focus on optimizing algorithms for emerging hardware architectures to improve performance.

Low-Power Devices: As IoT devices and mobile devices proliferate, there will be a continued emphasis on developing low-power compression algorithms that can operate efficiently on devices with limited computational resources.

7. **Environmental and Ethical Considerations**

Sustainable Compression: With increasing awareness of environmental impact, there is a growing focus on developing compression algorithms that are energy-efficient and reduce the carbon footprint associated with data storage and transmission.

Ethical Considerations: As compression algorithms become more sophisticated, there will be considerations around ethical issues such as privacy, data integrity, and the potential misuse of advanced compression technologies.

REFERENCES

- [1] Digital Image Processing by “Gonzalez and Woods”.
- [2] IEEE Journals and Research papers.
- [3] Mathworks Matlab Central.
- [4] Wikipedia.org
- [5] Ahmed, N., Natarajan, T., & Rao, K. R. (1974). "Discrete Cosine Transform." IEEE Transactions on Computers, C-23(1), 90-93
- [6] Chen, W.-H., Smith, C. H., & Fralick, S. C. (1977). "A Fast Computational Algorithm for the Discrete Cosine Transform." IEEE Transactions on Communications, COM-25(9), 1004-1009.
- [7] Wallace, G. K. (1991). "The JPEG Still Picture Compression Standard." Communications of the ACM, 34(4), 30-44.
- [8] Rao, K. R., & Yip, P. (1990). Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press.
- [9] Zhu, Q., & Rao, K. R. (1991). "Adaptively Scanned DCT for the Efficient Compression of Images." IEEE Transactions on Communications, 39(9), 1285-1288.
- [10] He, L., & Sun, Z. (1997). "Deblocking Filter for JPEG Compressed Images." Proceedings of the IEEE International Conference on Image Processing (ICIP), 451-454.
- [11] International Organization for Standardization (ISO) (1992). "Information technology—Digital compression and coding of continuous-tone still images: Requirements and guidelines." ISO/IEC 10918-1:1992 (JPEG).