INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

PROJECT REPORT ON

"**JPEG COMPRESSION**"

Submitted By:-

Anmol Sharma

(Roll No:-23EC65R02)

Ajay Kumar Mahto

(Roll No:-23EC65R16)

To

Department of Electrical & Electronics Communication(E&EC)

Vision and Intelligent System

MTech 1st Year (2023-2025)

ACKNOWLEDGEMENT

If words are considered as a symbol of approval and token of appreciation then let the words play the heralding role expressing my gratitude. The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We would like to express our gratitude towards **DR. Ritwik Kumar Layek and Dr. Saumik Bhattacharya** for his guidance, constant supervision, constructive suggestions and for his support in completing the project. We would like to express our gratitude towards our parents & members of **Indian Institute of Technology, Kharagpur** for their kind cooperation and encouragement which helped us in completion of this project.

We have put in efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. Our thanks and appreciations go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

Anmol Sharma (23EC65R02)

Ajay Kumar Mahto (23EC65R16)

INDEX:

1 INTRODUCTION

The rise in digital imaging devices, including scanners and digital cameras, has led to an unprecedented increase in the volume of digital images. Each image is essentially a two-dimensional matrix of pixels, representing the intensity of the image at various locations. The size of an image is defined by the number of pixels, calculated by its width multiplied by its height.

The proliferation of digital images has imposed a substantial demand for data storage due to the sheer volume of pixels each image comprises. Additionally, the transfer of these images across networks requires significant bandwidth. In response to these challenges, the development of image compression techniques has become essential to manage and optimize the storage and transmission of digital images for various applications.

Image compression serves as a means to reduce the size of digital images. It generally involves two primary approaches: lossless compression and lossy compression.

Lossless Compression: - In this type of compression, the compressed file is almost identical to the original file. For images the compression ratio goes up to 4:1.

Example: Huffman coding

Lossy Compression*:* - This compression is visually lossless. It maximizes the compression ratio while maintaining the required level of quality. In this technique there will be some degradation in the quality of the compressed file when compared with the original one. The compression ratio goes up to 80:1.

Example: DCT transform

The level of compression is defined by the term 'Compression Ratio (CR)', defined as the ratio of the number bits required to represent the data before compression to the total number of bits required to represent the data after compression.

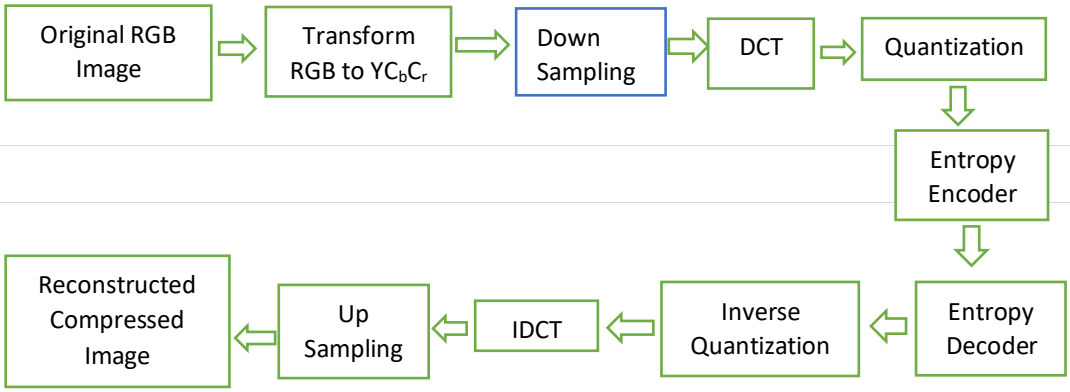
Compression Ratio = # of bits before compression/ # of bits after compression

Images usually have large amounts of redundant data in them. We exploit these redundancies to achieve compression by removing them. There are mainly four types of redundancies when talking about images.
They are: -

- **Interpixel Redundancy**: - Also known as spatial redundancy, in an image the values of the adjacent pixels are highly correlated. Hence, these pixels are redundant with each other. This refers to interpixel redundancy.
- **Spectral Redundancy**: - This refers to the correlation between different color planes or spectral bands.
- **Psychovisual Redundancy**: - Human visual system has unequal sensitivity to different visual information, for example human eye is less sensitive to very high and very low intensities than normal intensity. This refers to psychovisual redundancy.
    - **Coding Redundancy**:- This refers to the entropy of the source data. In this project, we deal with the JPEG compression technique, which is a lossy compression technique.

JPEG is the most important current standard for image compression**.** ***International Organization for Standardization (ISO)***created this standard called the ***Joint Photographic Experts Group (JPEG)***.JPEG takes advantage of the redundancies in the images to achieve higher compression rates. It has many modes and, in this project, we deal with the sequential mode, which is the default mode. Sequential Mode: Each Gray level image or color image component is encoded in a single left to right and top to bottom scan.

2 JPEG COMPRESSION PROCEDURE



2.1 Original image in RGB color space:-

Image are made up of pixels and the color of each pixel in the original image can be represented by 3- dimensional vector (R,G,B). The color of each pixel can be specified using intensities of red, green and blue. The intensity of each color varies from 0 to 255. Hence each color component can be represented as an integer. In a typical natural image, there is a significant amount of correlation between these components i.e. take a pixel and the pixel around this pixel will be similar. it is consequence of the fast that surface are smooth .Our aim is to find redundancies in order to reduce the amount of data required to represent the image.

2.1 TRANSFORM RGB TO YCBCR :-

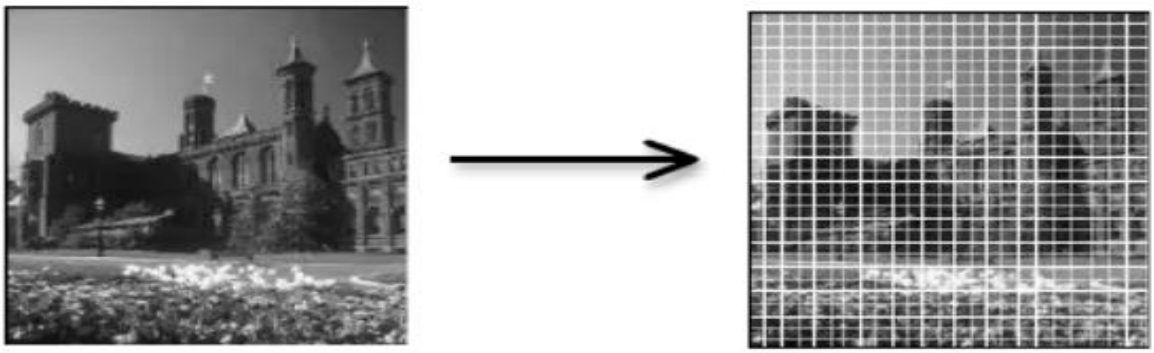
We represent color images by three components; they are red (R), green (G) and blue (B) components. This format is called RGB format. In this we represent each pixel with 3 bytes, one for each component. JPEG converts RGB color space to YCbCr color space because YCbCr tends to compress more tightly than RGB and is much easier than compressing the RGB image. The human eye is more sensitive to brightness than color. Therefore, the algorithm can exploit this by making significant changes to the picture's color information without affecting the picture's perceived quality. To do this, JPEG must first change the picture's color space from RGB to YCbCr.

Y-Cb-Cr consists of 3 different channels: Y is the luminance channel containing brightness information, and Cb and Cr are chrominance blue and chrominance red channels containing color information. In this color space, JPEG can make changes to the chrominance channel to change the image's color information without affecting the brightness information that is located in the luminance channel.

After conversion, each pixel has three new values that represent the luminance, the chrominance blue, and the chrominance red information.

2.2 DOWN SAMPLING: -

Human eye is more perceptible to luminance compared to chrominance. Therefore image can be down sampled by assuming the chrominance values to be constant on 2x2 block in our 8x8 block therefore reducing few values. Each block is encoded 'almost' independently hence we will assume for now that each 8x8 block is encoded independently. Down sampling reduces the data but also reduces the quality of the image. Most software use down sampling of two i.e. assume 2x2 block is constant (4*less color), however this can be increased.



2.3 DCT (DISCRETE COSINE TRANSFORM):-

There are many types of DCT but for JPEG, DCT is used most commonly. The main idea of DCT is to represent data of 8x8 pixel blocks as the sum of cosine functions. Each of the 8x8 pixel blocks are separately encoded with its own discrete cosine transform. Each of the 8x8 blocks can be exactly replicated, hence we have 64 cosine waves. This is true for all three of our components Y, Cb and Cr. From here on, we'll talk about luminance (Y) but Cb and Cr are similar.

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

C(i,j)= $\frac{1}{\sqrt{2}}$ if i,j=0

= 1 otherwise

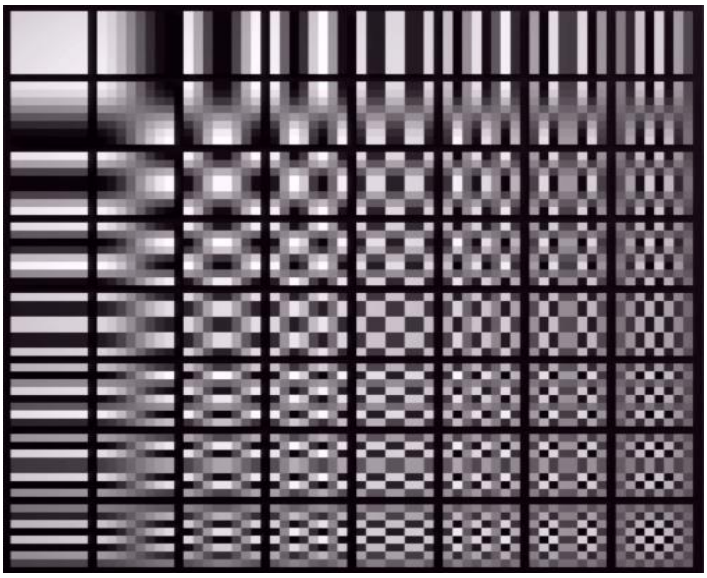
p(x,y) is the x,yth element of the image represented by the matrix p.

N is the size of the block that the DCT is done on.

The equation calculates one entry (i,jth) of the transformed image form the pixel values of the original image matrix.

First, each pixel value for each channel is subtracted by 128 to make the value range from -128 to +127.

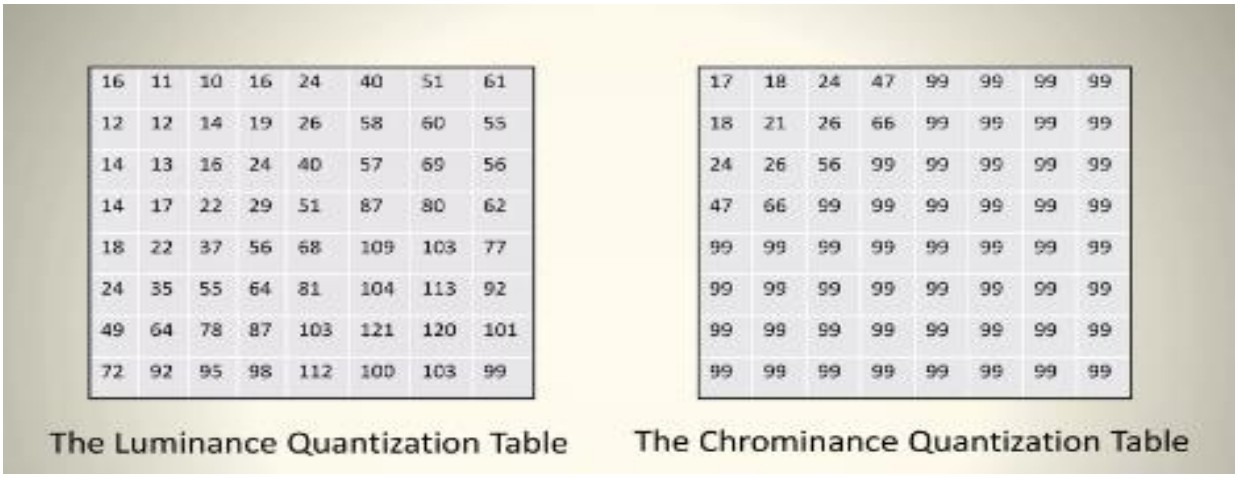
Using **DCT**, for each channel, each block of 64 pixels can be reconstructed by multiplying a constant set of base images by their corresponding weight values and then summing them up together. Here is what the base images look like:



2.4 QUANTIZATION:-

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. The 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 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.



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

The Luminance Quantization Table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

The Chrominance Quantization Table

Quantization steps is given by:-

$$F(i,j)=round(D(i,j)/Q(I,j))$$

D(i,j) represent the DCT coefficient, Q(i,j) is quantization matrix , and F(i,j) represent the quantized DCT coefficient. The user can choose the range of compression depending on his application. More compression means there are large entries in Q. There is a trade off between quality and quantization. Hence, quantization represents the DCT coefficient with no greater precision than required to achieve the desired level of quality.

2.5 ENTROPY ENCODING:-

The values that are inside each matrix, run **RLE** (Run Length Encoding) since we have a lot of zeroes, and then run the **Huffman Coding** algorithm before storing the data. RLE and Huffman Coding are lossless compression algorithms that reduce the space needed to store information without removing any data.

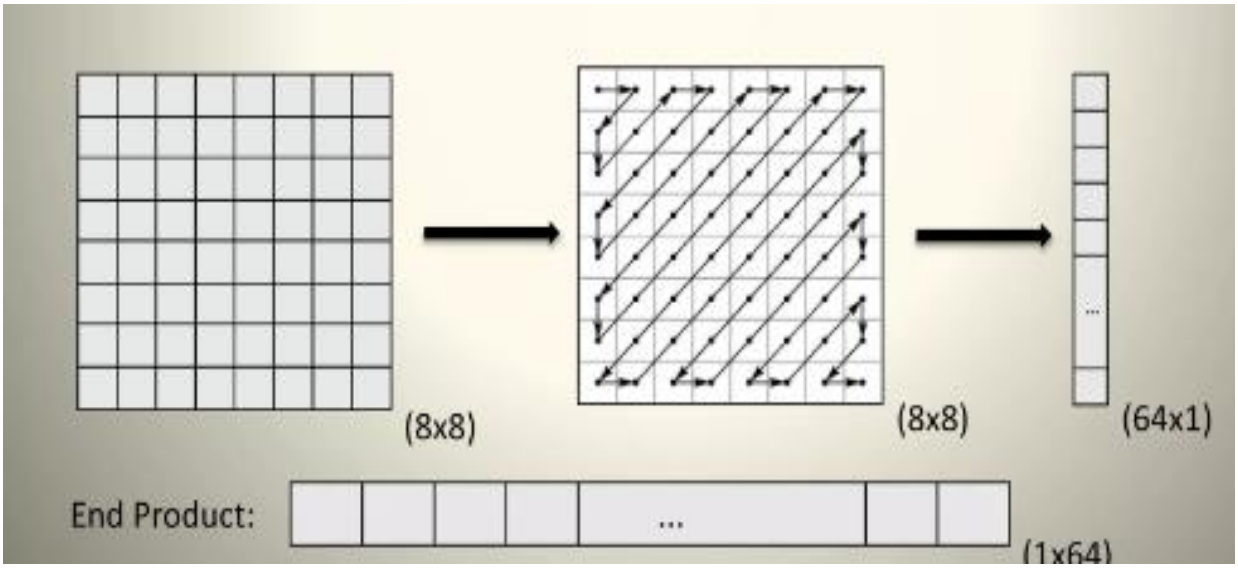
- VLC, VLI Encoding:
  - Each sample is represented by two symbols: (SIZE, VALUE)
  - Huffman Code for SIZE, VLI for VALUE
  - Example coding -9 and +9
    - Both -9 and +9 belong to category 4
    - Samples in category 4 represented by 4-bit VLI
    - Assume Huffman code for category 4 is '010'
    - 4-bit VLI for +9 is '1001; VLI for -9 is '0110'
    - Code for -9 is '010 0110'

➢ Zigzig Sequence Example:

- -6 -6 6 -5 -2 0 -1 0 0 0 0 0 0 -1 0 0 -1 1 0 0 0 …….0
- Assume Previous DC coefficient = -4
- The intermediate symbols are:

(-2), (0,3)(-6), (0,3)(6), (0,3)(-5), (1,2)(2), (1,1)(-1), (5,1)(-1), (2,1)(-1), (0,1)(1), (0,0)

As we can see, the result of quantization has many 0s toward the bottom left corner. To keep the 0 redundancy, the JPEG algorithm does a zig-zag scanning pattern that keeps all zeroes together



After zig-zag scanning, we get a list of integer values, and after running RLE on the list, we get each number and how many times they're repeated.

➢ Huffman Coding:

- DC and AC components finally need to be represented by a smaller number of bits
- Why Huffman Coding?
- Significant levels of compression can be obtained by replacing long strings of binary digits by a string of much shorter code words.
- • How?
- The length of each code word is a function of its relative frequency of occurrence. Normally a table of code words is used with the set of code words pre- computed using the Huffman Coding Algorithm.
- In Huffman Coding, each DPCM-coded DC coefficient is represented by a pair of symbols: (Size, Amplitude)

- where Size indicates number of bits needed to represent the coefficient and Amplitude contains actual bits.

➢ Huffman coding: Dc Components:

- DC components are coded as (Size, Value).The look –up table for generating code word for value is as given below :-

SIZE	Value	Code
0	0	---
1	-1,1	0,1
2	-3, -2, 2,3	00,01,10,11
3	-7,..., -4, 4,..., 7	000,..., 011, 100,...111
4	-15,..., -8, 8,..., 15	0000,..., 0111, 1000,..., 1111
.		.
.		.
11	-2047,..., -1024, 1024,... 2047	...

Table 1: Huffman Table for DC components *Value* field

➢ Huffman coding: Dc Components:

- The look-up table for generating code word for size is as given below:-

SIZE	Code Length	Code
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Table 2: Huffman Table for DC components *Size* field

➢ Huffman DC Coding – Example:

- If the DC component is 48, and the previous DC component is 52. Then the difference between the 2 components is (48-52) = -4.
- Therefore it is Huffman coded as: 100011
- 011: The codes for representing -4 (Table 1.)
- 100: The size of the value code word is 3. From Table 2, code corresponding to 3 is 100.

➢ Huffman Coding: AC Components:

- AC components are coded in two parts:
- **Part1: (RunLength/SIZE)**
- RunLength: The length of the consecutive zero values (0.15]
- **SIZE**: The number of bits needed to code the next nonzero

AC component's value. [0-A]

- (0,0) is the End Of Block (EOB) for the 8x8 block.
- Part1 is Huffman coded (see Table 3)
- **Part2: (Value)**
- Value: Is the actual value of the AC component.(Table 1)

Run/ SIZE	Code Length	Code
0/0	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011

Run/ SIZE	Code Length	Code
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
15/A	More	Such rows

Finally, JPEG runs Huffman coding on the result, and the values that have more frequency (like the zeroes) are represented with fewer bits than less frequent values. No data is lost during this phase.

2.6 ENTROPY DECODING:-

Whenever we open a .jpg file using an image viewer, the file needs to be decoded before it can be displayed. To decode a *.jpg* image, the image viewer does all the above steps in reverse order:

1. Run Huffman Decoding on the file data
2. Deconstruct RLE
3. Put the list of numbers into an 8×8 matrix
4. Multiply the integer values by their corresponding quantization table
5. Multiply the weight values by their corresponding base images, then sum them up to get the pixel values
6. Upscale the chrominance channels
7. Change the color space from YCbCr to RGB
8. Display the image

2.7 INVERSE QUANTIZATION:-

The function is given below:-

$$\mathbf{D(i,j)=F(i,j)*Q(i,j)}$$

2.8 Inverse DCT:-

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

where C(i) ,C(j) = $\frac{1}{\sqrt{2}}$ $for\ i,j = 0$

= 1 otherwise

2.9 UP SAMPLING:-

Up sampling (or interpolation) involves increasing the resolution or dimensions of an image. This process is employed to enhance the size or quality of an image by adding more pixels to increase its resolution.

When working with images in the YCbCr color space, after down sampling the chrominance components (Cb and Cr) while retaining the luminance (Y) component in processes like JPEG compression, up sampling is performed to restore the lost chrominance information to the original resolution.

3 RESULTS:-

Original Image Size: 193 kb

Compressed Image Size: 9Kb

Original Image

Compressed Image

Original Image Size: 128 Kb

Compressed Image Size: 24 Kb

Original Image

Compressed Image

Original Image Size: 66 kb

Compressed Image Size: 7 Kb

Original Image

Compressed Image

Original Image Size: 289 Kb

Compressed Image Size:11 Kb

Original Image

Compressed Image

4 CONCLUSION:-

The JPEG compression process is a block-based compression algorithm that reduces the size of an image by removing information that is not easily visible to the human eye while keeping the information that the human eye is good at perceiving . The process involves color space conversion, down sampling, division into 8x8 pixel blocks, Discrete Cosine Transform (DCT), quantization, and Huffman coding . The quality/compression ratio can be adjusted by selecting specific quantization matrices The JPEG standard quantization matrix is based on subjective experiments involving the human visual system . With a quality level of 50, this matrix renders both high compression and excellent decompressed image quality JPEG compression process is a lossy compression algorithm that results in significantly smaller file sizes with little to no perceptible impact on picture quality and resolution . The amount of image quality loss due to data reduction can be controlled by setting or choosing presets .

5 REFERENCES: -

1.http://en.wikipedia.org/wiki/Jpeg
2. https://ieeexplore.ieee.org/document/125072
3. https://www.math.cuhk.edu.hk/~lmlui/dct.pdf
4. https://hal.science/hal-03858141/file/qtable.pdf
5.Chapter 9: Image Compression Standards from *Fundamentals of Multimedia*
By Ze-Nian Li & Mark S. Drew.
6.*The JPEG Still Picture Compression Standard*
by Gregory K. Wallace.4.
7.Introduction to Data Compression
by Khalid Sayoo