JPEG Compression Pipeline

IMAGE AND VIDEO PROCESSING LABORATORY

Jayadithya Sreekar V 21EC39050
Samineni Rahul 21EC39021

Introduction

JPEG compression is one of the most widely used image compression methods for reducing the size of digital images while maintaining visual quality.

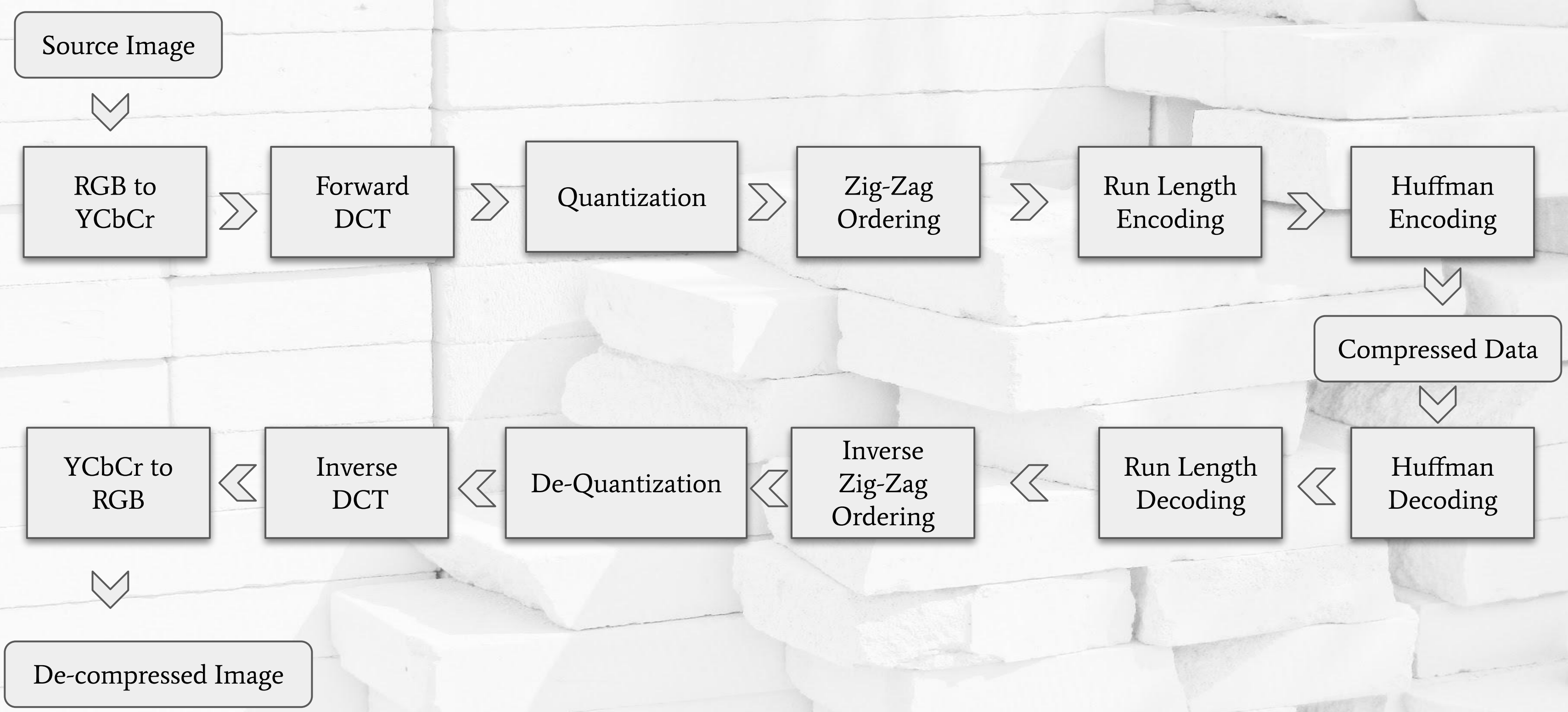
JPEG stands for Joint Photographic Experts Group, the committee responsible for developing this widely used image compression standard.

This project implements a JPEG compression pipeline incorporating techniques such as color space conversion, Discrete Cosine Transform (DCT), quantization, zigzag ordering, Run-Length Encoding (RLE), and Huffman coding.

Idea

- JPEG compression relies on the human eye's sensitivity to luminance (brightness) over chrominance (color).
- We perceive changes in brightness more clearly than subtle variations in color.
- JPEG reduces the quality of chrominance data more aggressively than luminance to achieve efficient compression.
- This approach transforms and separates brightness from color.
- The result is a significant reduction in file size while preserving perceived visual quality.

Pipeline


Source Image
RGB to YCbCr
Forward DCT
Quantization
Zig-Zag Ordering
Run Length Encoding
Huffman Encoding
Compressed Data
Huffman Decoding
Run Length Decoding
Inverse Zig-Zag Ordering
De-Quantization
Inverse DCT
YCbCr to RGB
De-compressed Image

ALGORITHM

1. *Color Transform:*
   Convert the input image from RGB to YCbCr color space to separate luminance (Y) and chrominance (Cb, Cr) components.
2. *Discrete Cosine Transform (DCT):*
   Apply the DCT to each 8×8 block of the Y, Cb, and Cr channels to transform spatial pixel data into frequency domain coefficients.
3. *Quantization*:
   Divide each DCT coefficient by a corresponding value in a quantization matrix and round the result to reduce precision and compress the data.
4. *Zigzag Ordering:*
   Reorder the quantized 8×8 block coefficients into a one-dimensional array following a zigzag pattern to group low-frequency coefficients before high-frequency ones.
5. *Run-Length Encoding (RLE):*
   Encode the zigzag-ordered coefficients by representing consecutive zeros with a count, effectively compressing sequences of zeros.
6. *Huffman Encoding:*
   Apply Huffman coding to the run-length encoded data to assign shorter binary codes to more frequent symbol patterns, further compressing the information.

COLOR TRANSFORM

- The RGB color space, commonly used for display, is less efficient for compression. Transforming the image into a different color space, such as YCbCr, separates luminance (brightness) from chrominance (color).
- Human vision is more sensitive to changes in luminance than chrominance. This characteristic allows the JPEG algorithm to compress chrominance data more aggressively with minimal perceived loss of image quality.

JPEG uses YCbCr model
Y: Luminance (Brightness)
Cb: Chrominance (blue-difference)
Cr: Chrominance (red-difference)

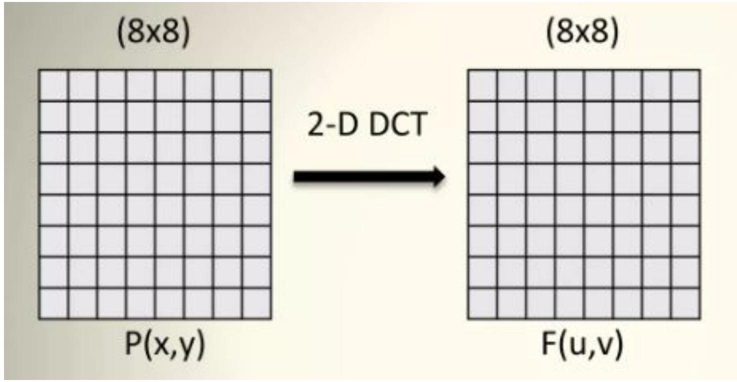
Discrete Cosine transform

- Before applying the DCT, the image is divided into non-overlapping 8x8 pixel blocks.
- The Discrete Cosine Transform (DCT) relies on the cosine function, avoiding interactions with complex numbers entirely.
- Human vision is less sensitive to high-frequency components, allowing high-frequency data to be treated as redundant.
- Converting raw image data into the frequency domain helps separate data based on frequency, making DCT the key function for this purpose.

2D-DCT:

$$F(m,n) = \frac{2}{\sqrt{MN}} C(m)C(n) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\frac{(2x+1)m\pi}{2M} \cos\frac{(2y+1)n\pi}{2N}$$

DC AND AC COMPONENTS



- The top-left value F(0,0) in the transformed matrix is called the **DC component** . It represents the average value of all 64 values in the input 8x8 block.
- All other values in the transformed matrix are called **AC components** , representing varying frequency coefficients within the block.
- For u=v=0, the cosine terms reduce to 0, making F(0,0) a function of the sum of all values in the input matrix P(x,y).
- The DC component holds the mean value of the matrix, while AC components store frequency-related data, capturing changes and details in the image block.

Quantization

- Humans are less sensitive to high-frequency details, making them less visually significant.
- Quantization leverages this by reducing or zeroing out less important values, freeing memory and reducing data.
- *Uniform Quantization*: Uses a constant quantization value q(u,v).
- *Non-Uniform Quantization:* Utilizes custom quantization tables, with two default tables defined in JPEG standards for luminance (Y) and chrominance (Cb, Cr).
- The quantization step introduces controlled information loss, which is key for data reduction.

$$QDCT[m][n] = Round\left(\frac{DCT\_COEFF[m][n]}{Q[m][n]}\right)$$

where Q is the Quantization Matrix

Quantization Matrices

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

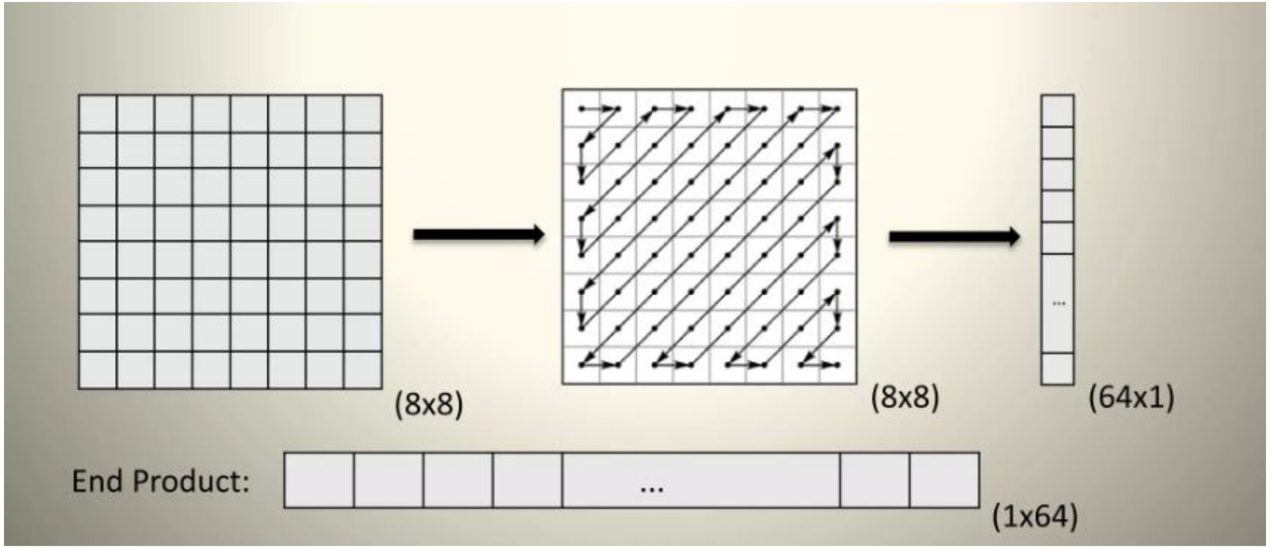
17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

The Chrominance Quantization Table

- The values of Q(u,v) typically increase towards the lower right corner, which is intended to introduce greater loss at higher spatial frequencies.

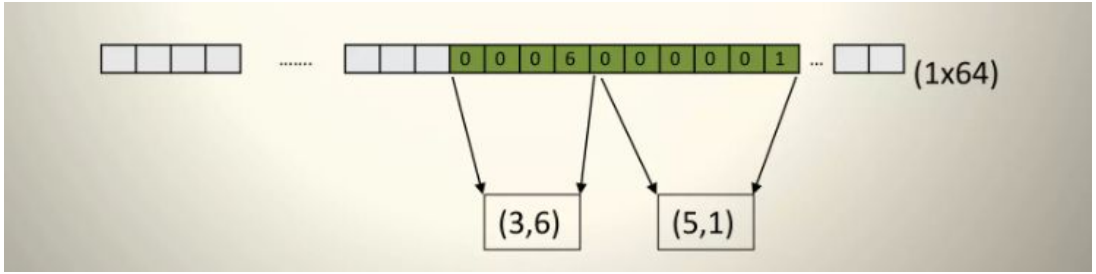
ZIG-ZAG ORDERING

- Maps 8x8 matrix to 1x64 vector
- To group low frequency coefficients at the top and high frequency coefficients at the bottom
- In Order to exploit the presence of large number of zeros in the quantized matrix ,a zigzag matrix is used



Run length encoding

- RLE compresses sequences of identical values by representing them with a count and a single value.
- After **zigzag scanning** , DCT coefficients often contain long sequences of zeros due to quantization.
- RLE efficiently compresses these sequences by replacing them with pairs (run-length, value), reducing data storage requirements.
- (0,0) is sent as End of the block sentinel value and the zeros after that are discarded. Only the number of zeros taken note of.



Huffman Encoding

- DC component are encoded using differential coding
- AC components need to be represented using fewer bits for efficient compression.
- It offers significant data compression by replacing long binary strings with shorter code words based on frequency.
- The length of each code word is determined by the relative frequency of occurrence—frequently occurring values receive shorter codes.
- A table of pre-computed code words is used for encoding, generated through the Huffman Coding Algorithm.

RESULTS

Input Image: Cameraman.bmp
Compression Ratio : 9.74

Input Image : lena.bmp
Compression Ratio:34.4

Input Image : Garden.png
Compression Ratio :18.7


THANK YOU!