Indian Institute of Technology, Kharagpur

2.

DESIGN AND IMPLEMENTATION OF JPEG COMPRESSION

SREENIJA REDDY NALLAGATLA [24EC65R16]

KANCHI DEEPIKA [24EC65R13]

Vision and Intelligent Systems
Dept of E&ECE

Friday 8th November, 2024

Contents

1.INTRODUCTION

What is JPEG?
JPEG (Joint Photographic Experts Group) is a popular image compression format that reduces file size by discarding some visual details, which makes it "lossy".
What is lossy compression?
While lossy compression reduces quality to some extent, it achieves much smaller file sizes compared to lossless compression.
Human Visual Perception: JPEG compresses data based on how the human eye perceives image details, discarding less perceptible information.
Flexible Compression Levels: JPEG allows users to choose a trade-off between compression ratio and image quality.

2.JPEG Compression Overview

- Color Space Conversion
- Subsample Chrominance Channels
- Discrete Cosine Transform (DCT)
- Quantization
- Zigzag Scanning
- Run-Length Entropy Coding (Huffman Coding)
- Encoding

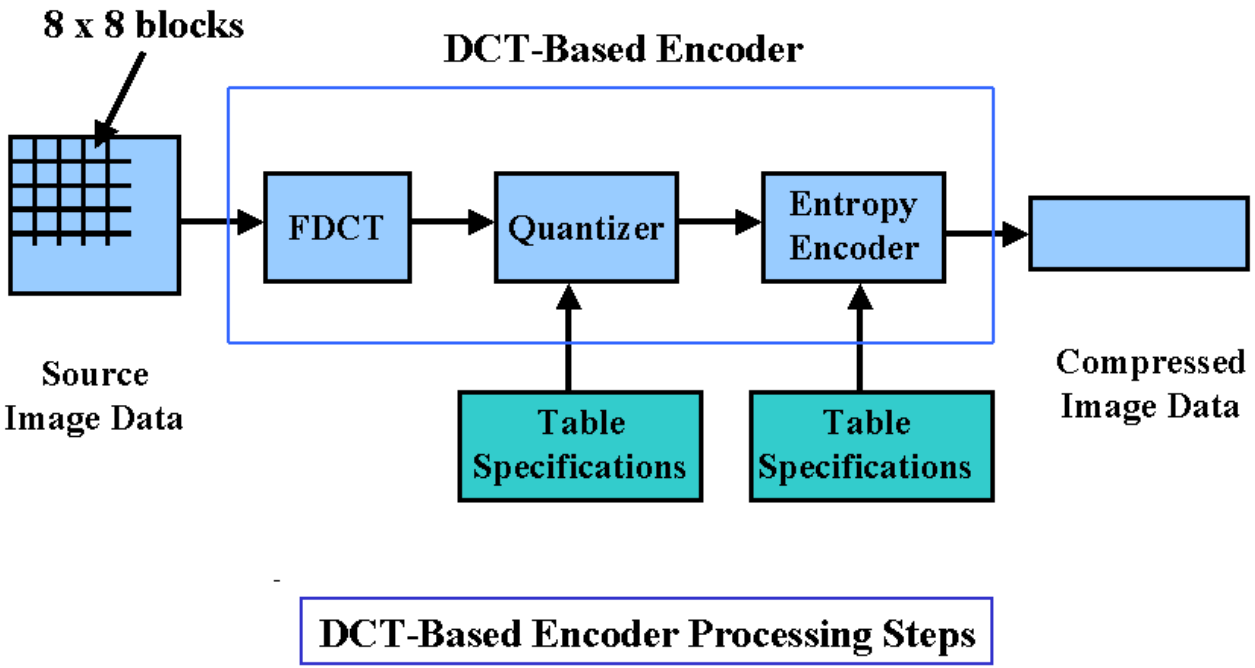

Figure: JPEG Block Digram

Color Space conversion

- Y (luminance) represents brightness, while Cb and Cr (chrominance) represent color details.

$$
\begin{aligned}
Y &= 0.299R + 0.587G + 0.114B, \\
Cb &= 128 - 0.168736R - 0.331264G + 0.5B, \\
Cr &= 128 + 0.5R - 0.418688G - 0.081312B.
\end{aligned} \tag{1}
$$

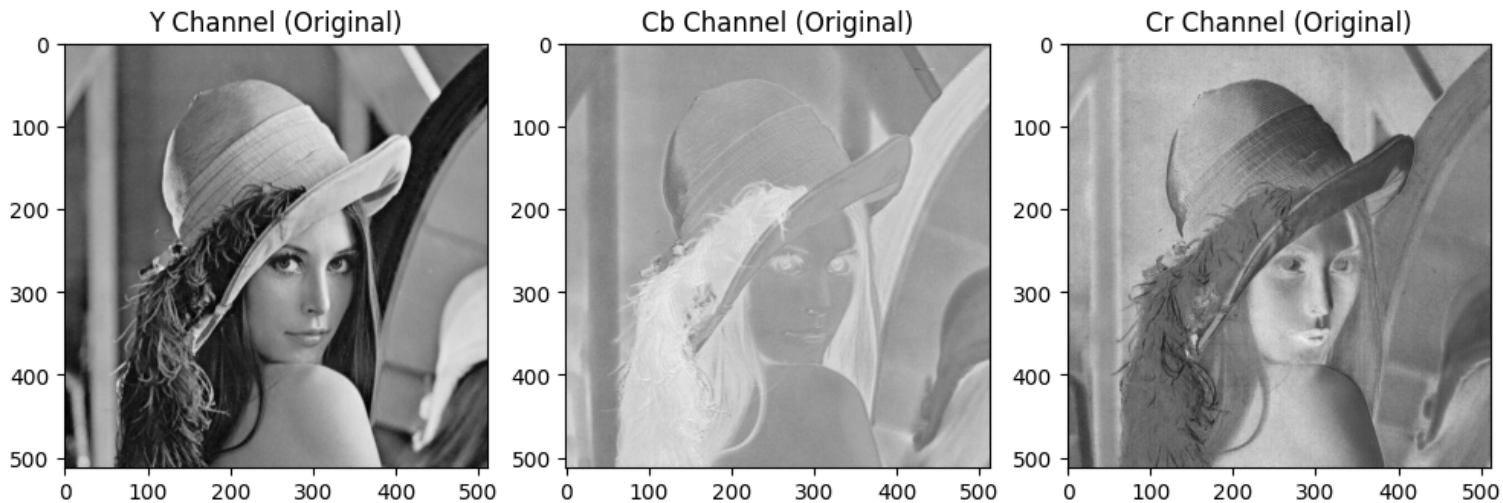

Figure: Output for color space conversion

Subsample Chrominance Channels

- The Cb and Cr channels are subsampled, cutting down the data that needs to be stored for color information.

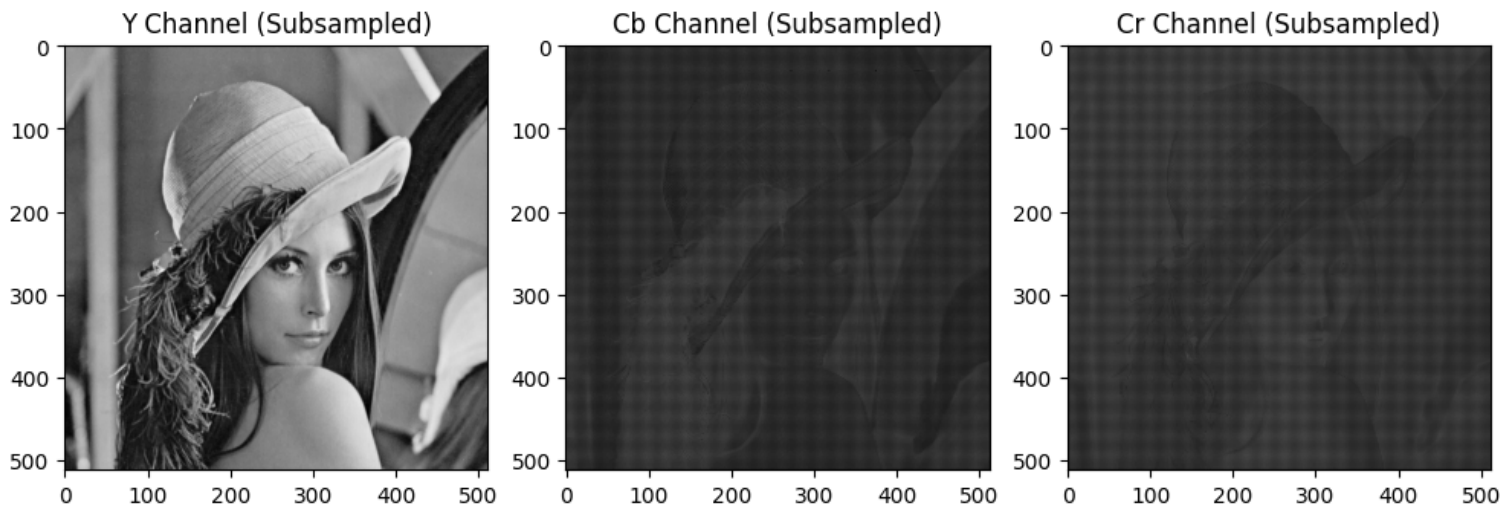

Figure: Sub sampling the Chrominance channels by 4:2:0 Format

Discrete Cosine Transform (DCT)

- The Discrete Cosine Transform (DCT) is applied to each block, converting pixel values into a matrix of frequency components.

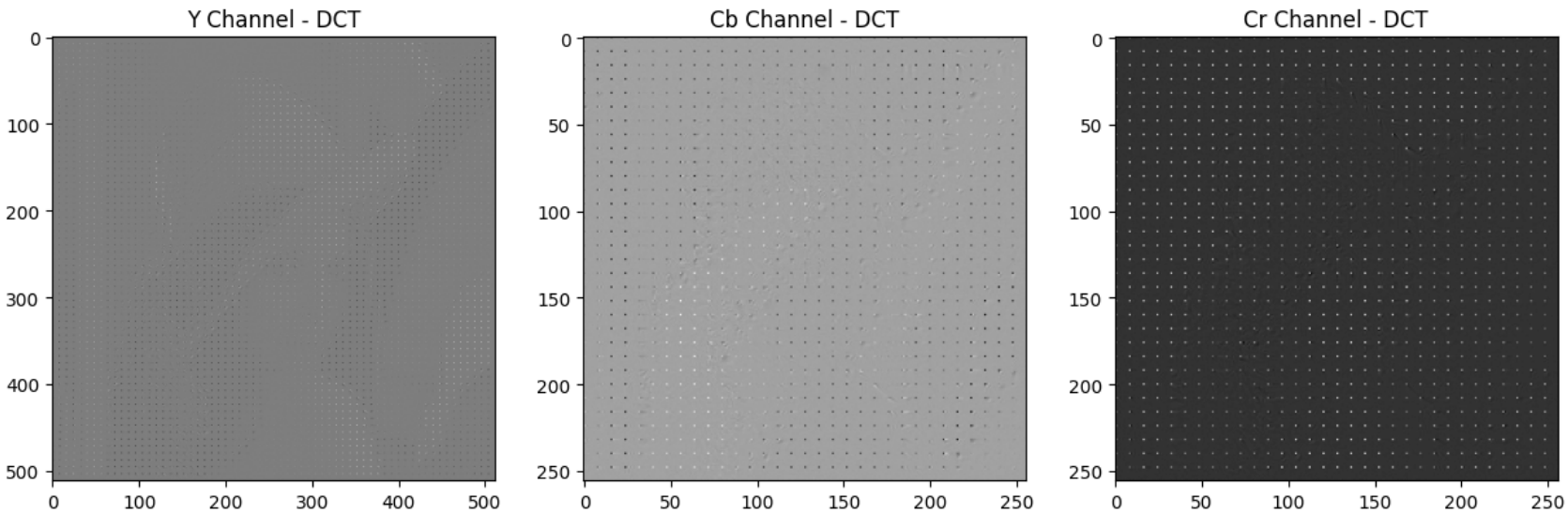

Figure: output for DCT

Quantization

- JPEG divides each frequency component in the DCT matrix by a predefined quantization table and rounds the result to the nearest integer.

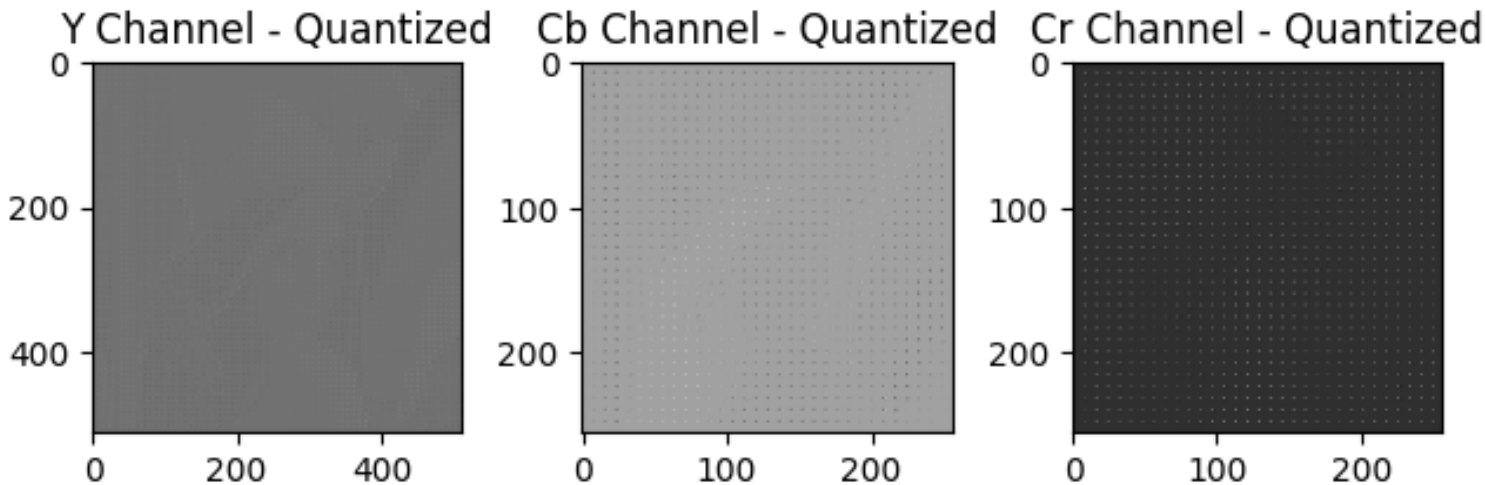

Figure: output for Quantization

Zigzag Scanning and Run-Length Encoding

- The quantized matrix is read in a zig-zag pattern to group zeroes together.
- The zig-zag-ordered data is encoded by representing consecutive zeroes and repeating values in a more compact form.

```
RLE Encoded Y Channel: [[((0, 4), -11), ((0, 2), 2), ((0, 1),
RLE Encoded Cb Channel: [[((0, 5), -19), (0, 0)], [((0, 5), -1
RLE Encoded Cr Channel: [[((0, 5), 24), ((0, 2), -2), ((0, 1),
total_rel_data: [[((0, 4), -11), ((0, 2), 2), ((0, 1), 1), ((
DC Components (Y, Cb, Cr): [-11, -11, -13, -14, -14, -13, -12,
AC Components (Y, Cb, Cr): [(0, 2), (0, 1), (2, 1), (0, 1), (
```

Figure: Enter Caption

Huffman coding

- **DC Huffman Table:** Encodes the difference between DC coefficients of adjacent 8x8 blocks by categorizing the difference and assigning a unique Huffman code based on bit length.
- **AC Huffman Table:** Encodes run-length and size of non-zero AC coefficients in each 8x8 block. Each (RUNLENGTH, SIZE) pair is assigned a Huffman code, with shorter codes for frequent patterns.

```
Huffman Codes for DC Components:
6: 00000
-8: 000010
-32: 000011000
-34: 000011001
-23: 00001101
25: 00001110
```

```
Huffman Codes for AC Components:
(0, 2): 00
(0, 3): 010
(1, 2): 01100
(0, 5): 011010
(1, 3): 0110110
(2, 2): 0110111
(2, 1): 0111
```

Huffman Encoding and Compression Process

- **Data Encoding:** The image is processed using zigzag and RLE, followed by creating DC and AC Huffman codebooks to encode coefficients into compressed data.
- **Compression Evaluation:** The compression ratio is computed by comparing the sizes of the original and compressed binary data, showing Huffman encoding efficiency.

```
Encoded Binary Data:
1000010100001010110001111011010110110101011000110011101
Sample of Encoded Binary Data (First 100 bits):
1000010100001010110001111011010110110101011000110011101

--- Compression Results ---
Original Data Size: 574784 bits
Compressed Data Size: 47714 bits
Compression Ratio: 12.046443391876599
```

3.Results and Conclusion

- **Original Data:** The image is represented by 16-bit DC and AC coefficients.
- **Compression Process:** RLE reduces redundancy, followed by Huffman coding for further compression.
- **Encoded Data:** Binary data is compressed using variable-length Huffman codes.
- **Compression Ratio:** The ratio compares original and compressed data sizes, indicating compression effectiveness.
- **Efficiency:** Significant data size reduction, demonstrating RLE and Huffman coding efficiency.

4.Future Scope

- **Improved Compression Techniques:** Exploring advanced compression algorithms like Arithmetic Coding and Adaptive Huffman Coding for better efficiency.
- **Real-time Processing:** Implementing real-time compression for video streaming and live media applications.
- **Lossless Compression:** Developing methods to achieve lossless compression while maintaining efficient storage and processing speeds.
- **Hybrid Compression Methods:** Combining different techniques like wavelet transforms and Huffman coding to enhance compression for specific image types.

Thank You!