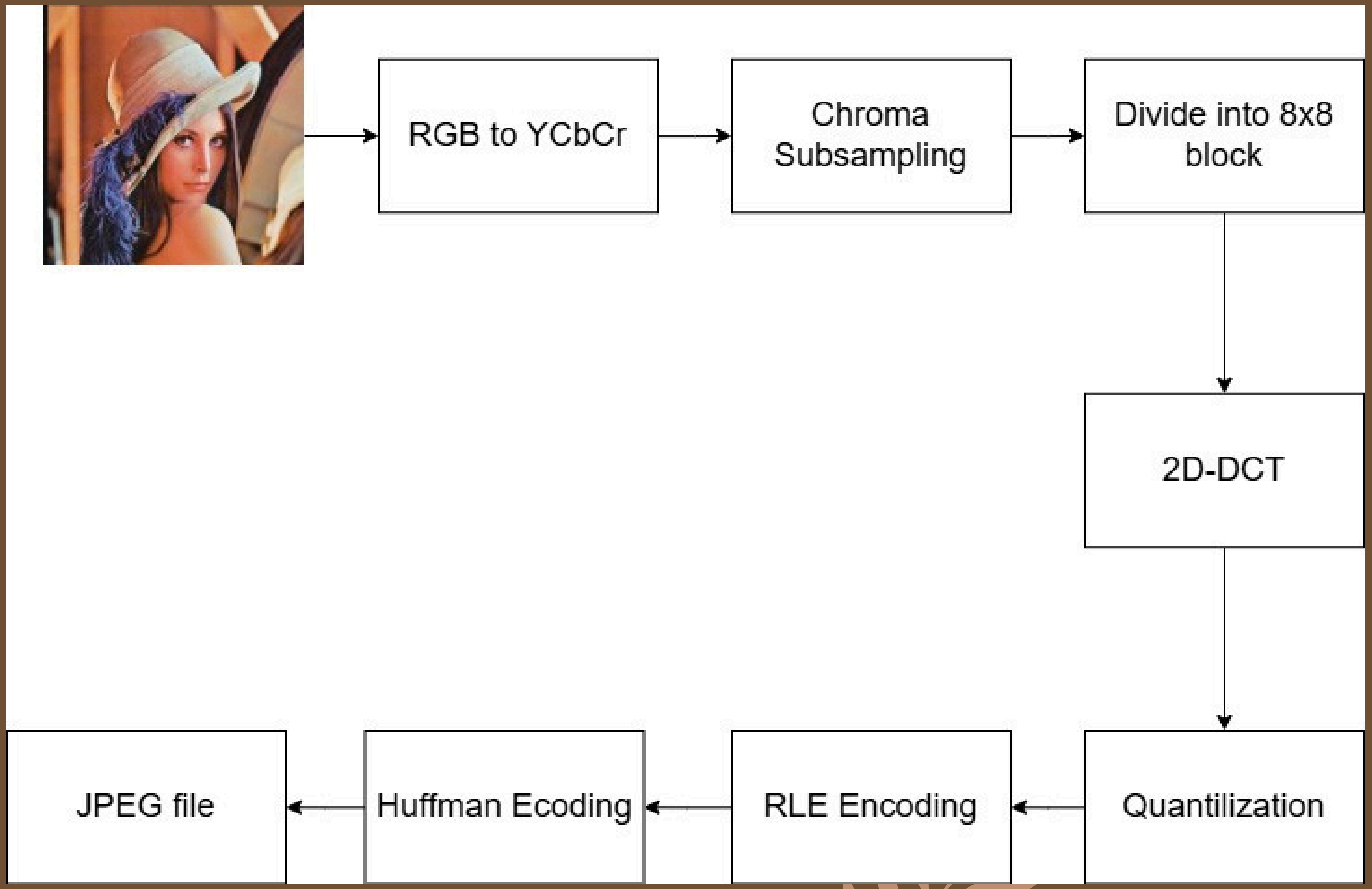
(JPEG) LOSSY IMAGE COMPRESSION WITH DCT, RLE AND HUFFMAN ENCODING

WHAT IS JPEG

- JPEG is a lossy image compression standard
- Data loss during compression. Decompressed images may have slight quality loss
- Make image files much smaller, which easier to store and send, with little loss in visual quality
- Analyzes the image pixels and removes elements that are less noticeable to the human eye, particularly high-frequency changes that are harder to perceive

HIGH-LEVEL OVERVIEW OF JPEG COMPRESSION PIPELINE



STEP 1: RGB TO YCBCR

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

- Y (Luminance)
- Cb (Blue Chrominance)
- Cr (Red Chrominance)

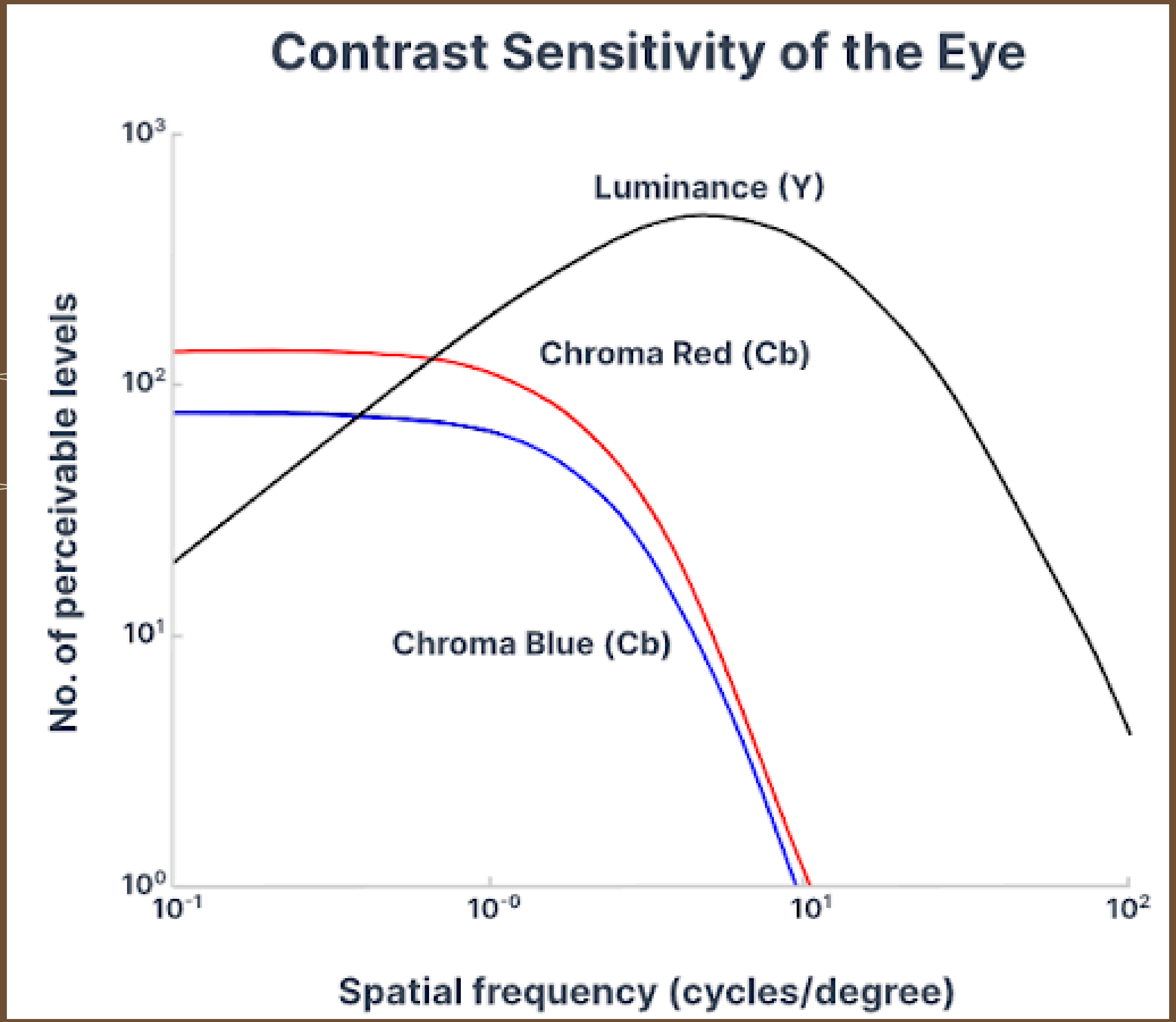
Given R = 81, G = 35, B = 0

$$Y = 0.299(81) + 0.587(35) + 0.114(0) = 147$$

$$Cb = -0.1687(81) - 0.3313(35) + 0.5(0) + 128 = 86$$

$$Cr = 0.5(81) - 0.4187(35) - 0.0813(0) + 128 = 14$$

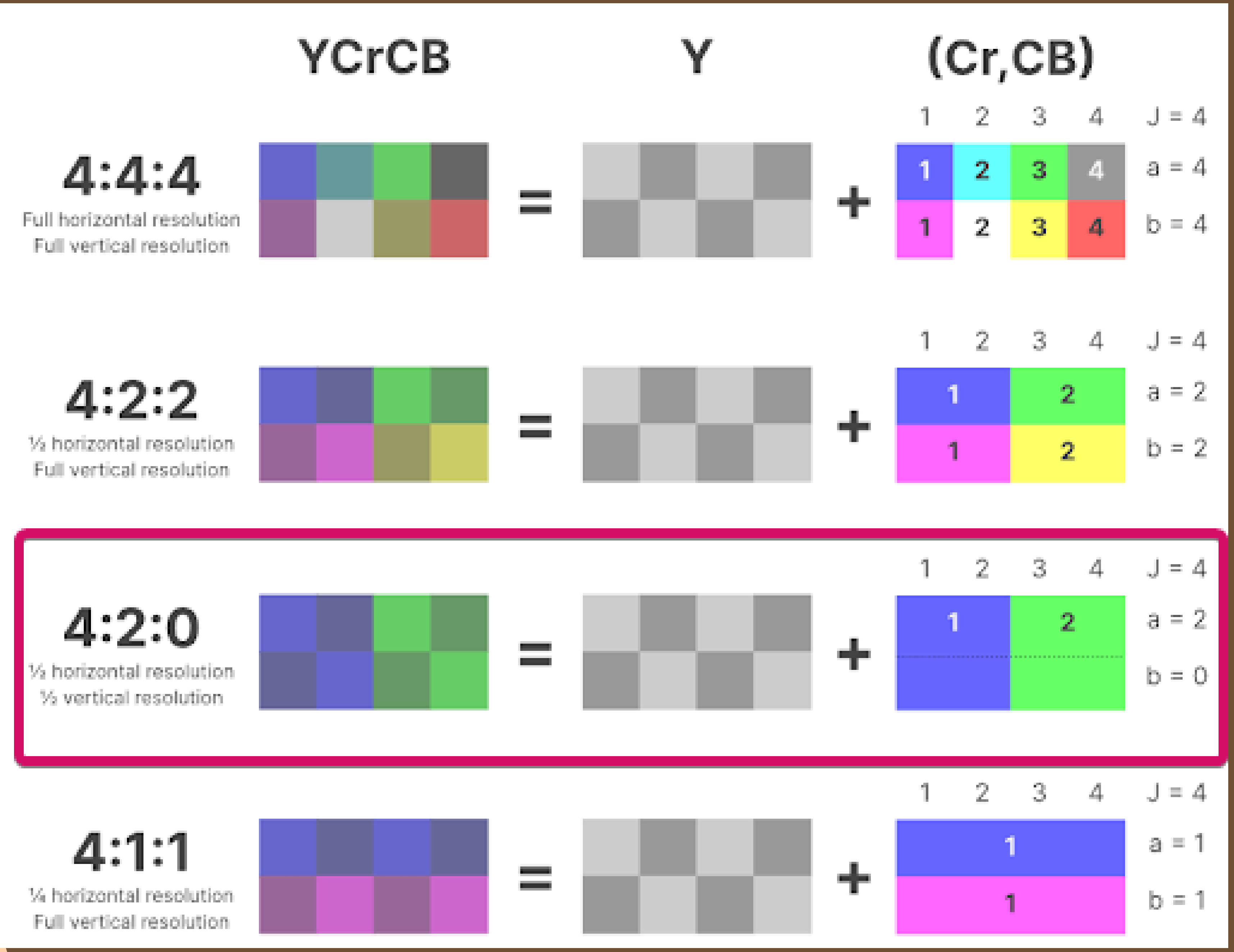
WHY CONVERT RGB COLOR SPACE TO YCBCR?

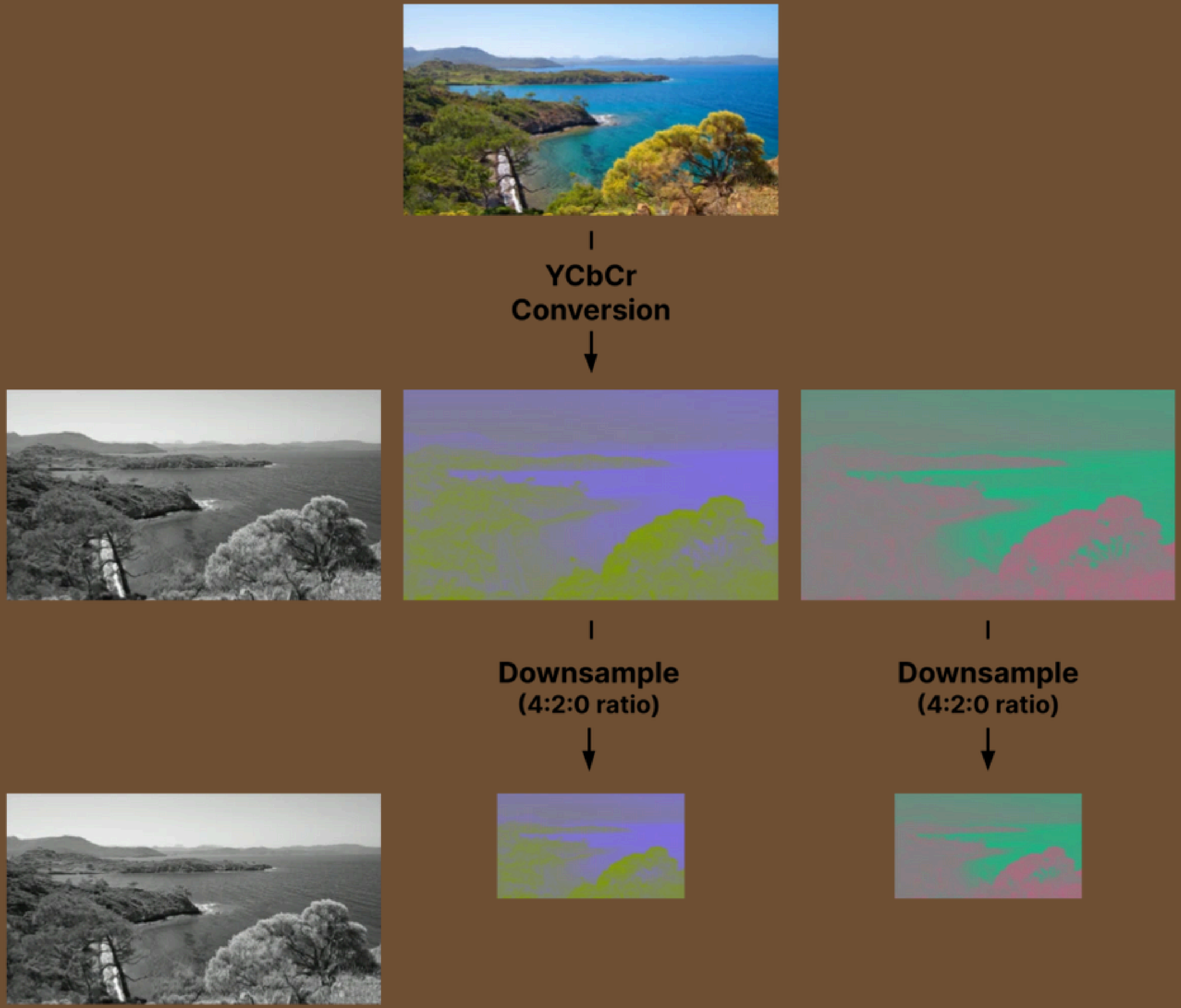


Luminance Chrominance Original

Human eye less sensitive to Chrominance

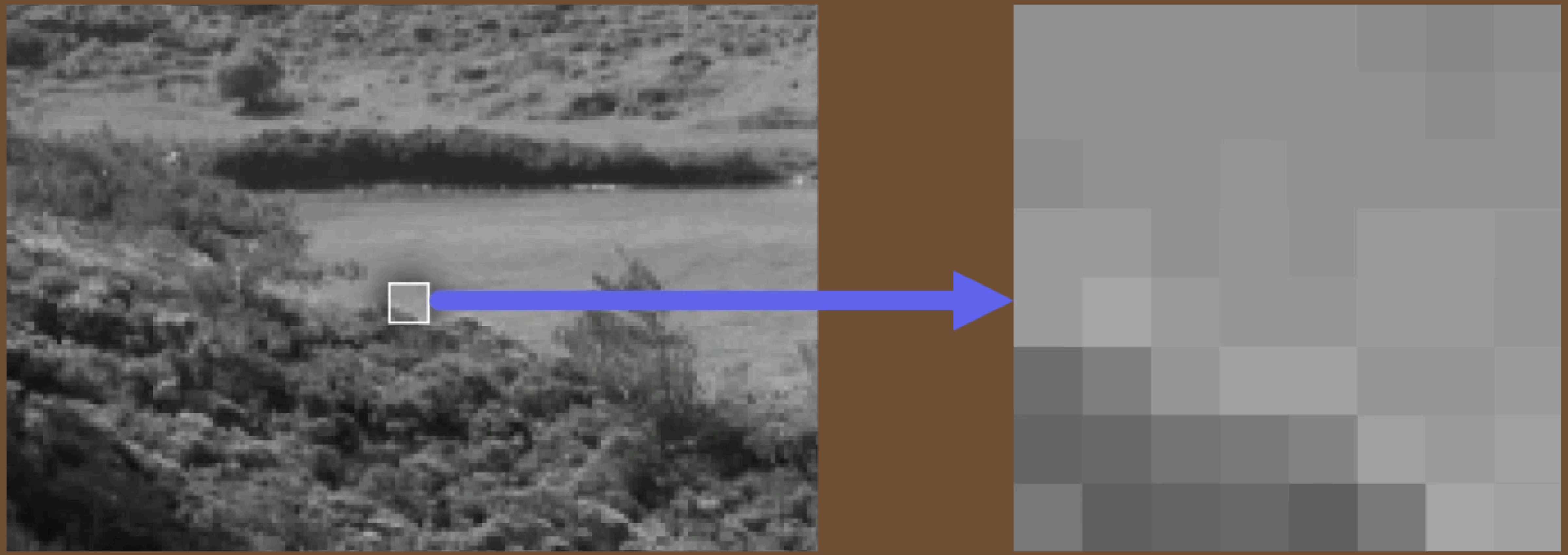
STEP 2: CHROMA SUBSAMPLING





Reduce 50% of size

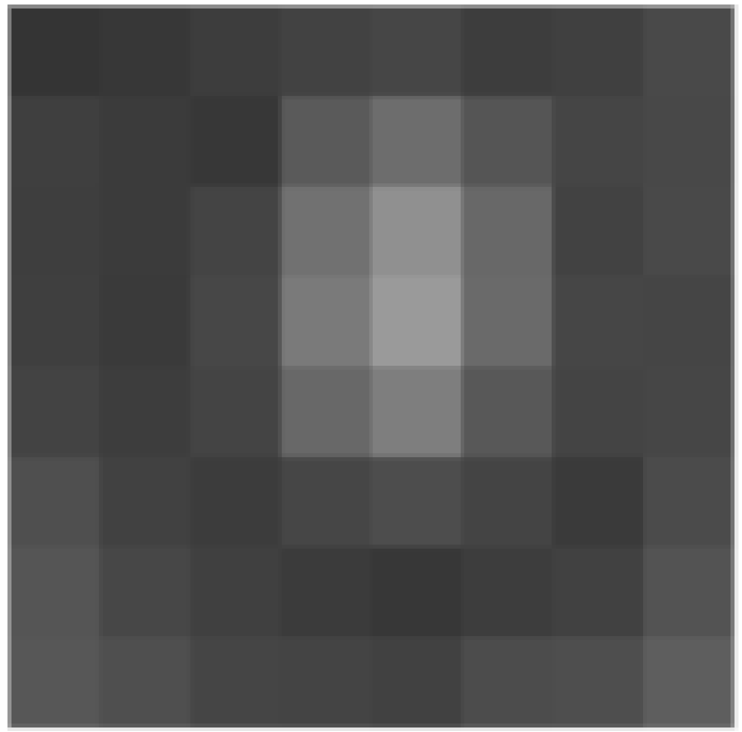
STEP 3: DIVIDE INTO 8X8 BLOCK



- The entire image is divided into non-overlapping 8x8 pixel blocks.
- Each block contains 64 pixels with luminance values ranging from 0 to 255

STEP BEFORE DISCRETE COSINE TRANSFORM (DCT)

Step 1: Convert Image to Y, Cb and Cr values range from 0 to 255.



$$\rightarrow \begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 87 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Step 2: Each pixel value is shifted by subtracting 128, transforming the range from [0,255] to -[128, 127]

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 87 & 65 & 76 & 78 & 94 \end{bmatrix} \xrightarrow{\text{Subtract 128}} \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

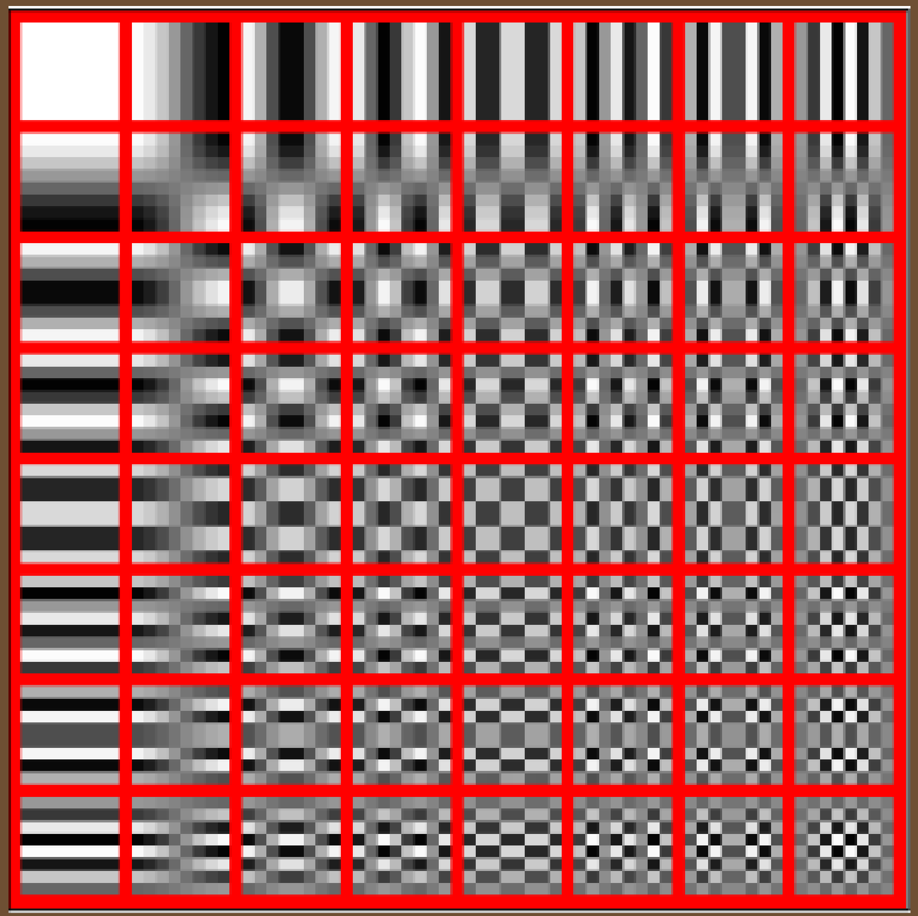
STEP 4: DISCRETE COSINE TRANSFORM (DCT)

Step 3: Apply DCT formula to compute frequency coefficients.

$$G_{u,v} = \frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y} \cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

where

- u is the horizontal spatial frequency, for the integers $0 \le u < 8$.
- v is the vertical spatial frequency, for the integers $0 \le v < 8$.
- $\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } u = 0 \\ 1, & \text{otherwise} \end{cases}$ is a normalizing scale factor to make the transformation orthonormal
- $g_{x,y}$ is the pixel value at coordinates (x, y)
- $G_{u,v}$ is the DCT coefficient at coordinates (u, v).



$$g = \begin{array}{c} x \\ \longrightarrow \\ \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} \end{array} \Bigg\downarrow y.$$

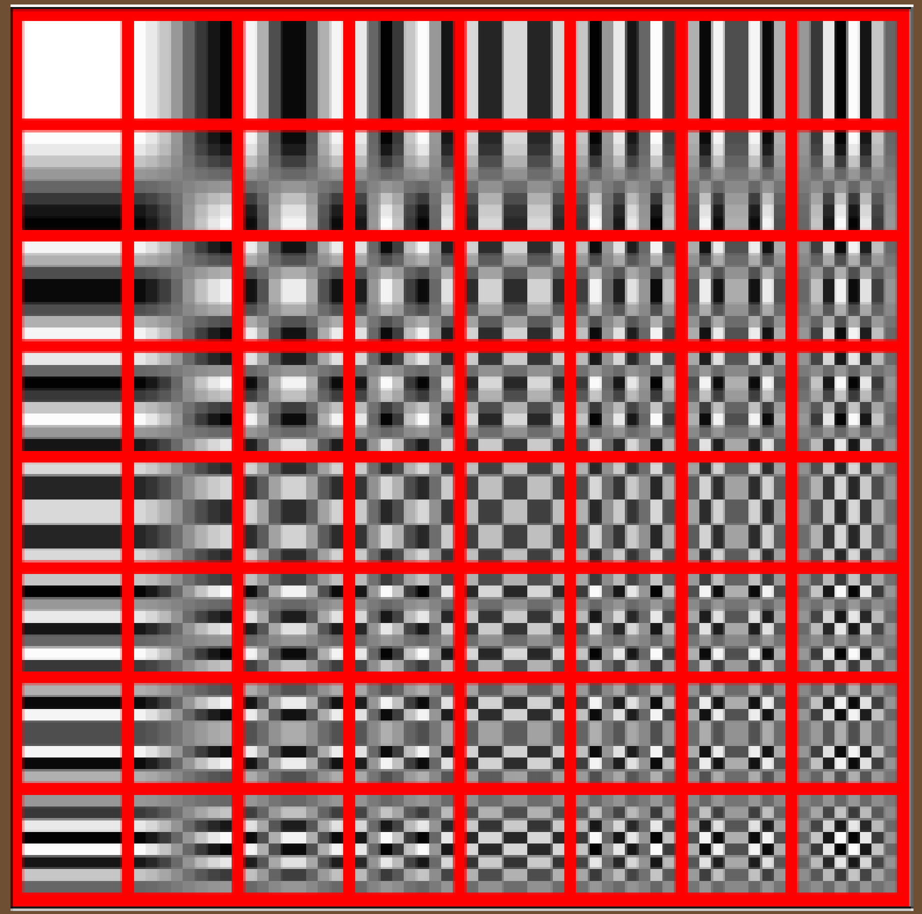
DCT Coefficient Matrix Result:

$$G = \begin{array}{c} u \\ \longrightarrow \\ \begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.12 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.87 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix} \end{array} \Bigg\downarrow v.$$

WHAT IS DCT ?

DCT Coefficient Matrix Result:

$$
G = \begin{array}{c}
\overset{u}{\longrightarrow} \\
\begin{bmatrix}
-415.38 & -30.19 & -61.20 & 27.24 & 56.12 & -20.10 & -2.39 & 0.46 \\
4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\
-46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\
-48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\
12.12 & -6.55 & -13.20 & -3.95 & -1.87 & 1.75 & -2.79 & 3.14 \\
-7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\
-1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\
-0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68
\end{bmatrix}
\end{array}
\Big\downarrow v.
$$



- We take image data and we try to represent it as the sum of multiple cosine waves
- In JPEG compression, a set of 64 cosine cover a range of frequencies, with higher horizontal frequencies increasing as you move from left to right and higher vertical frequencies increasing as you move from top to bottom
- The purpose of calculating DCT (Discrete Cosine Transform) coefficients is to figure out how much each "frequency" contributes to making up an image

STEP 5: QUANTIZATION

$$B_{j,k} = \text{round}\left(\frac{G_{j,k}}{Q_{j,k}}\right) \text{ for } j = 0,1,2,\ldots,7; k = 0,1,2,\ldots,7$$

where G is the unquantized DCT coefficients; Q is the quantization matrix above; and B is the quantized DCT coefficients.

DCT Coeeficient

-415.38	-30.19	-61.20	27.24	56.12	-20.10	-2.39	0.46
4.47	-21.86	-60.76	10.25	13.15	-7.09	-8.54	4.88
-46.83	7.37	77.13	-24.56	-28.91	9.93	5.42	-5.65
-48.53	12.07	34.10	-14.76	-10.24	6.3	1.83	1.95
12.12	-6.55	-13.20	-3.95	-1.87	1.75	-2.79	3.14
-7.73	2.91	2.38	-5.94	-2.38	0.94	4.30	1.85
-1.03	.18	0.42	-2.42	-0.88	-3.02	4.12	-0.66
-0.17	.14	-1.07	-1.07	-1.17	-0.10	0.50	1.68

Divide

Luminance quantization tables with 50% 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

Quantized DCT Coeeficient

-26	-3	-6	2	2	-1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-3	1	2	-1	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

STEP 6: ZIG ZAG

Order the Quantized DCT Coefficient in Zigzag Order:

−26	−3	−6	2	2	−1	0	0
0	−2	−4	1	1	0	0	0
−3	1	5	−1	−1	0	0	0
−3	1	2	−1	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

−26 −3 0 −3 −2 −6 2 −4 1 −3 1 1 5 1 2 −1 1 −1 2 0 0 0 0 0 −1 −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

Order the coefficients horizontally:

−26	−3	−6	2	2	−1	0	0
0	−2	−4	1	1	0	0	0
−3	1	5	−1	−1	0	0	0
−3	1	2	−1	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

-26 -3 -6 2 2 -1 0 0 0 0 0 1 1 -4 -2 0 -3 1 5 -1 -1 0 0 0 0 0 0 0 -1 2 1 -3 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

RLE ENCODING

- Group consecutive repeated values with pairs indicating the value and its frequency.
- Represented as (Pixel Value, Frequency)

Input Array

[−26 −3 0 −3 −2 −6 2 −4 1 −3 1 1 5 1 2 −1 1 −1 2 0 0 0 0 0 −1 −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]

Run-length encoded result :

[(-26, 1), (-3, 1), (0, 1), (-3, 1), (-2, 1), (-6, 1), (2, 1), (-4, 1), (1, 1), (-3, 1), (1, 2), (5, 1),
(1, 1), (2, 1), (-1, 1), (1, 1), (-1, 1), (2, 1), (0, 5), (-1, 2), (0, 38)]

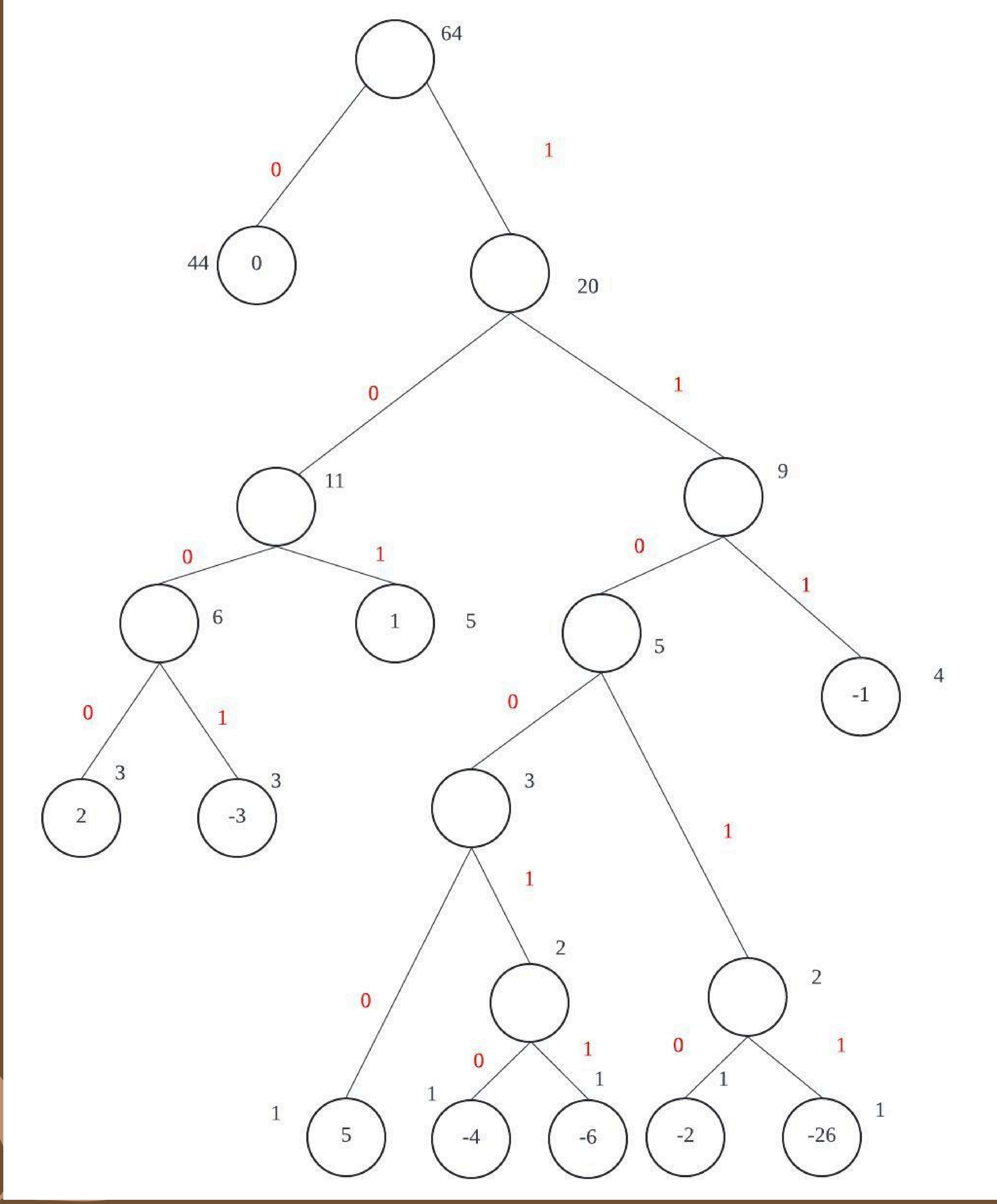
HUFFMAN ENCODING

STEP 1:
BUILD A FREQUENCY TABLE

Quantized DCT Coefficient	Frequency
-26	1
-6	1
-4	1
-2	1
5	1
-3	3
2	3
-1	4
1	5
0	44

HUFFMAN ENCODING

STEP 2:
CONSTRUCT HUFFMAN TREE



HUFFMAN ENCODING

STEP 3:
BUILDING HUFFMAN TABLE

Quantized DCT Coefficient	Binary Code
-26	11011
-6	110011
-4	110010
-3	1001
-2	11010
-1	111
0	0
1	101
2	1000
5	11000

HUFFMAN ENCODING

STEP 4: GENERATE HUFFMAN CODE

RLE ENCODED CODE:
[(-26, 1), (-3, 1), (0, 1), (-3, 1), (-2, 1), (-6, 1), (2, 1), (-4, 1), (1, 1), (-3, 1), (1, 2), (5, 1), (1, 1), (2, 1), (-1, 1), (1, 1), (-1, 1), (2, 1), (0, 5), (-1, 2), (0, 38)]

HUFFMAN CODE:
11011 , 1001 , 0 , 1001 , 11010 , 110011 , 1000 , 110010 , 101 , 1001 , 101 , 101 , 11000 , 101 , 1000 , 111 , 101 , 111 , 1000 , 0 , 0 , 0 , 0 , 0 , 111 , 111 , 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

CONCATENATED HUFFMAN CODE:
11011100101001110101100111000110010101100110110111000101100011110111110000000000
11111100000000000000000000000000000000000000000

CALCULATING THE COMPRESSION RATE

Zig Zag Quantized DCT Coefficient:
[-26 -3 0 -3 -2 -6 2 -4 1 -3 1 1 5 1 2 -1 1 -1 2 0 0 0 0 0 -1 -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]
64 number x 8 bits = 512 bits

RLE Encoded Code:
[(-26, 1), (-3, 1), (0, 1), (-3, 1), (-2, 1), (-6, 1), (2, 1), (-4, 1), (1, 1), (-3, 1), (1, 2), (5, 1), (1, 1), (2, 1), (-1, 1), (1, 1), (-1, 1), (2, 1), (0, 5), (-1, 2), (0, 38)]
21 pair x (8 bits (first num in pair) + [6|7|8] bits (second num in pair))
= 21 pair x 14 bits
= 294 bits (Reduce 43% from Original Size)

Huffman Code:
11011 , 1001 , 0 , 1001 , 11010 , 110011 , 1000 , 110010 , 101 , 1001 , 101 , 101 , 11000 , 101 , 1000 , 111 , 101 , 111 , 1000 , 0 , 0 , 0 , 0 , 0 , 111 , 111 , 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
122 bits

CALCULATING THE COMPRESSION RATE

Quantized DCT Coefficient	Binary Code
0	0
1	101
-1	111
2	1000
-3	1001
5	11000
-2	11010
-26	11011
-4	110010
-6	110011

Per Coefficient: 8 bits(for the coefficient) + 6-bit code = 14 bits.
Entire Table (10 entries): 10 entries × 14 bits = 140 bits

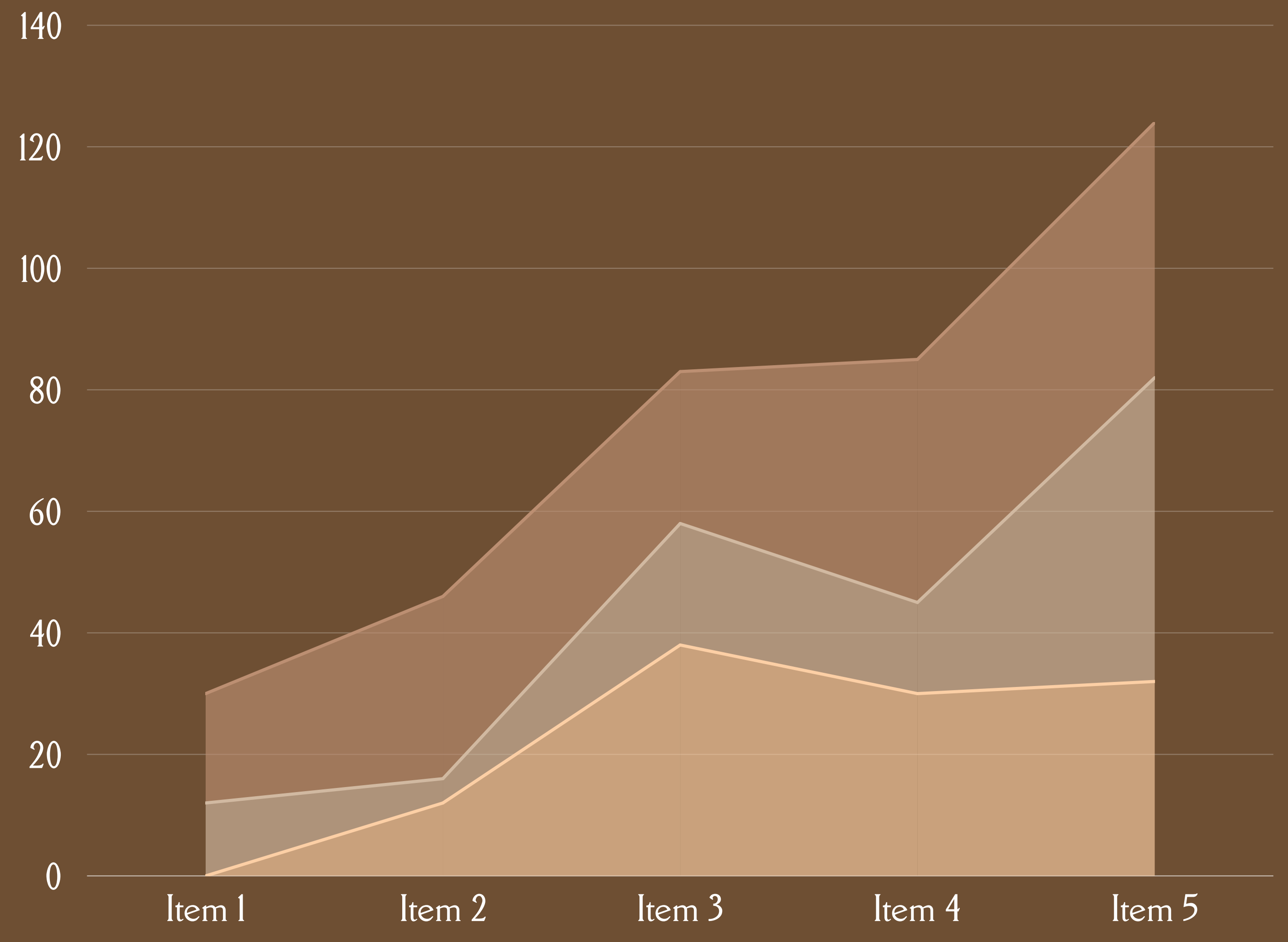
Huffman Code + Frequency Table:
122 + 140 bits = 262 bits (Reduce 49% From the Original Size)

COMPRESSED BIN FORMAT

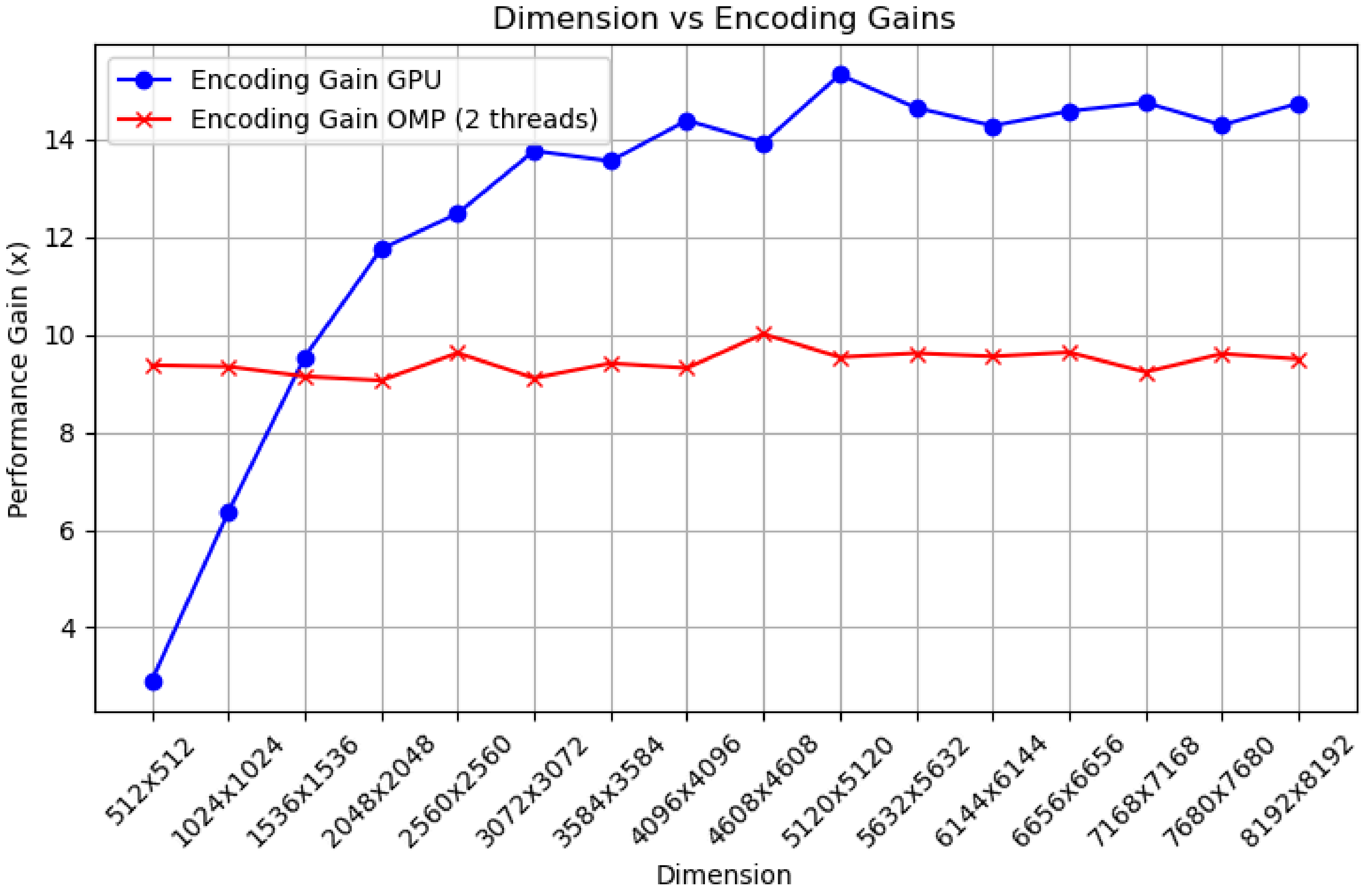
THE COMPRESSED RESULT IS STORED IN A CUSTOM BINARY FORMAT CREATED BY ME. BELOW IS THE STRUCTURE OF THE FILE HEADER FOR THE COMPRESSED BINARY FORMAT:

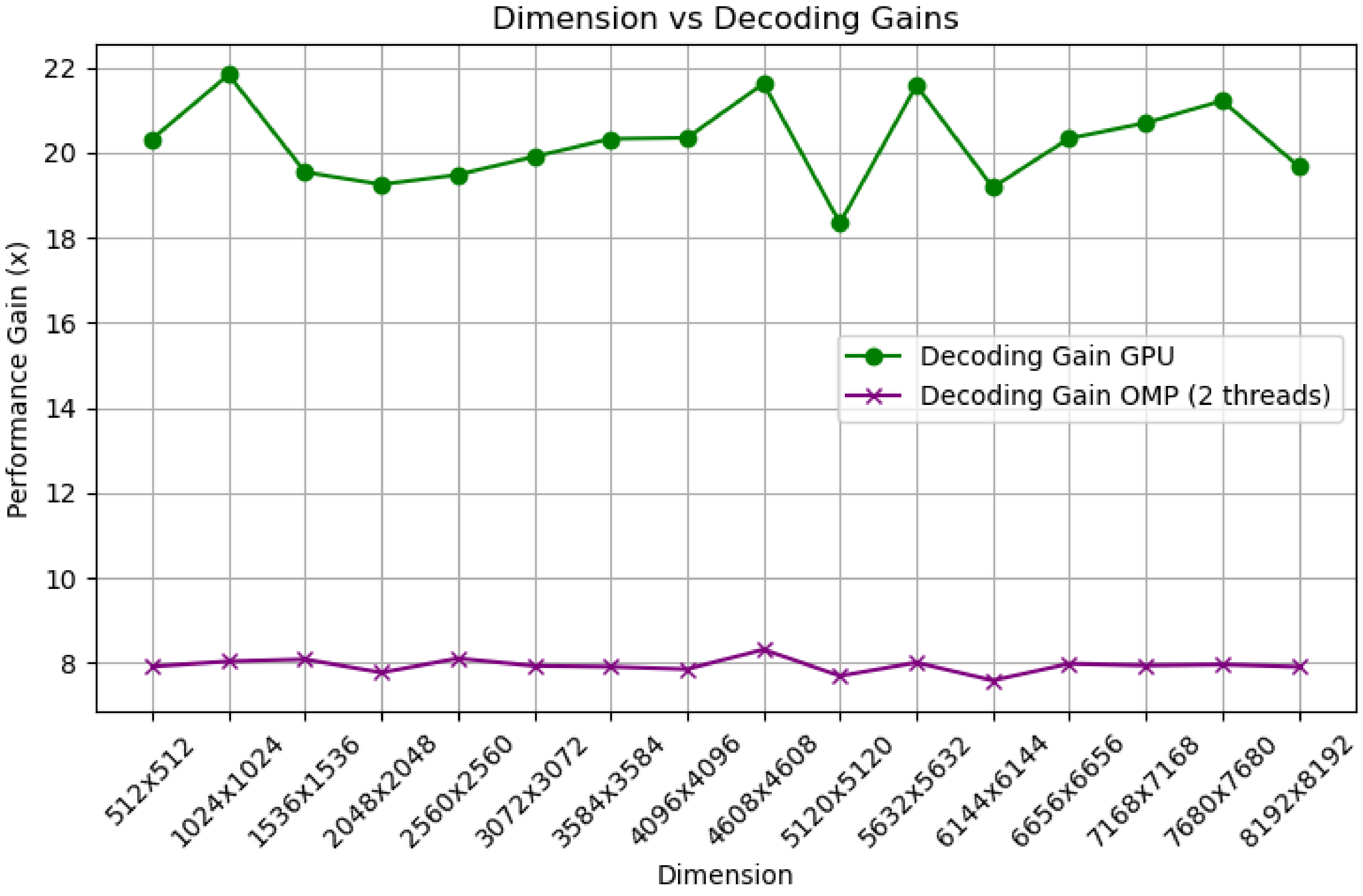
<table>
<tr><td colspan="6">Dimension</td><td colspan="3">Size(Length) of Data</td><td colspan="3">Compress Data (Encoded Huffman String)</td><td colspan="3">Size(Length) of Frequency Dictionary</td><td colspan="6">Frequency Dictionary</td></tr>
<tr><td colspan="2">Y</td><td colspan="2">Cb</td><td colspan="2">Cr</td><td rowspan="2">Y</td><td rowspan="2">Cb</td><td rowspan="2">Cr</td><td rowspan="2">Y</td><td rowspan="2">Cb</td><td rowspan="2">Cr</td><td rowspan="2">Y</td><td rowspan="2">Cb</td><td rowspan="2">Cr</td><td colspan="2">Y</td><td colspan="2">Cb</td><td colspan="2">Cr</td></tr>
<tr><td>row</td><td>col</td><td>row</td><td>col</td><td>row</td><td>col</td><td>Coefficient</td><td>frequency</td><td>Coefficient</td><td>frequency</td><td>Coefficient</td><td>frequency</td></tr>
</table>

PERFORMANCE METRICS

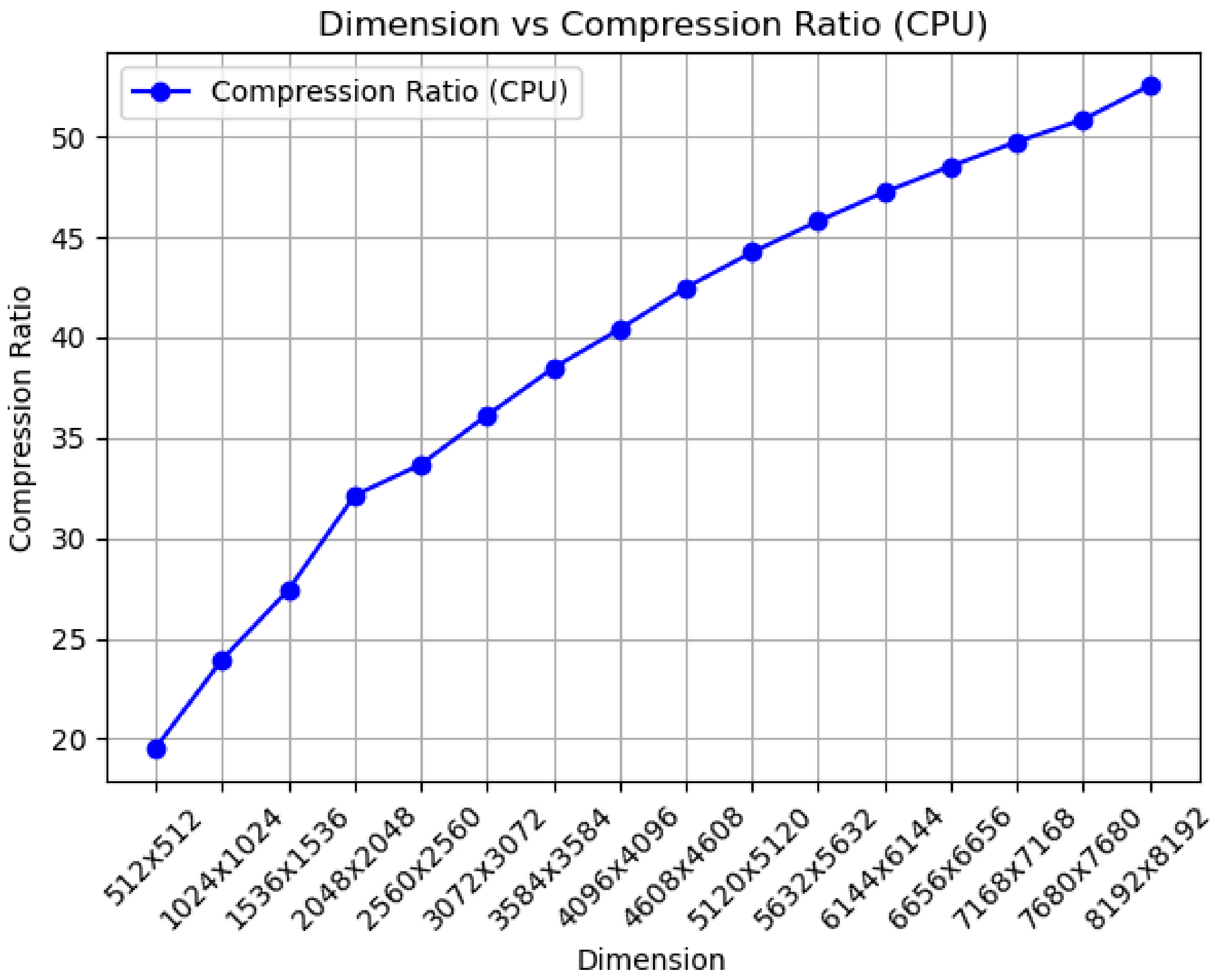


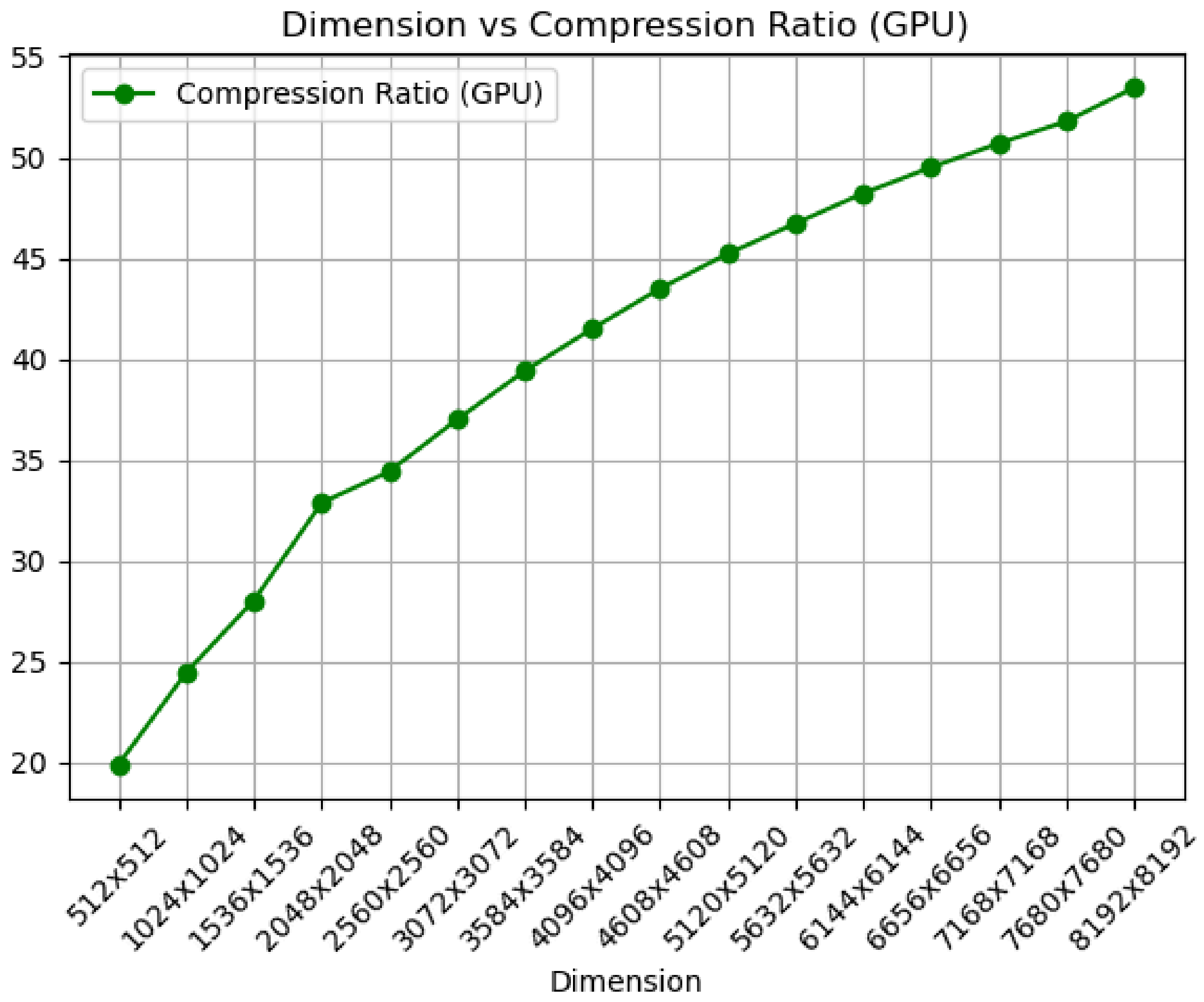
ENCODING AND DECODING GAIN

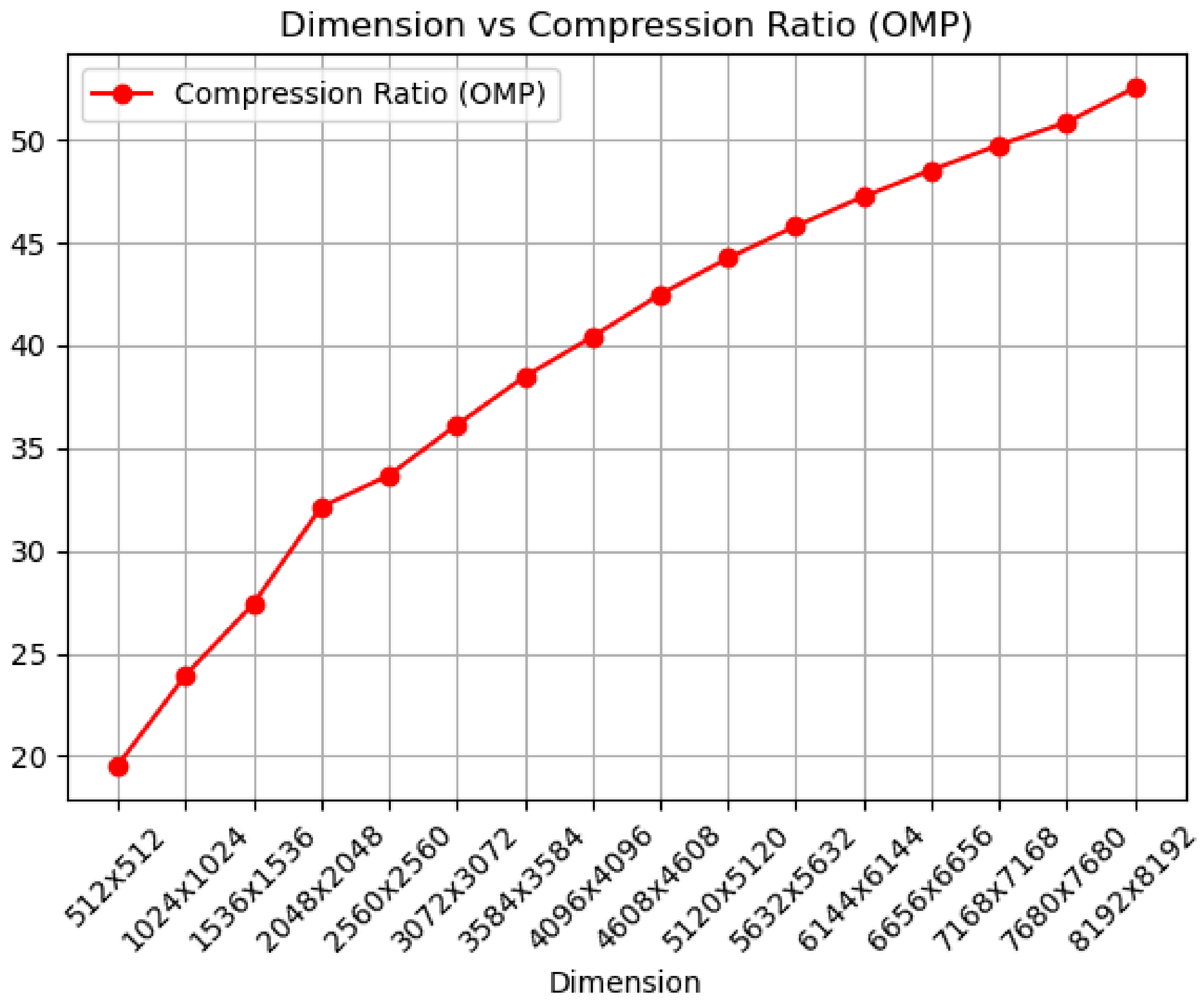




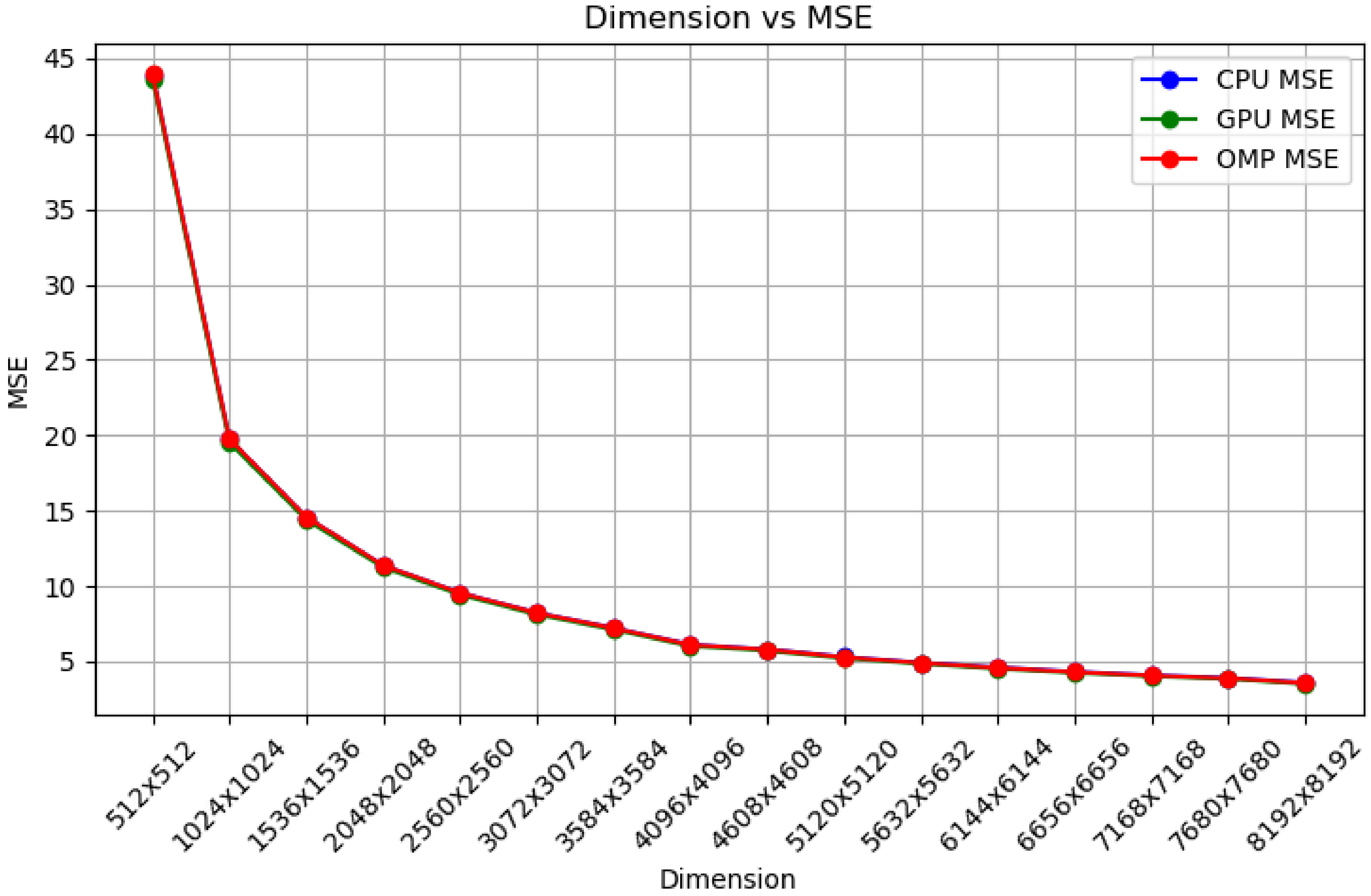
COMPRESSION RATE

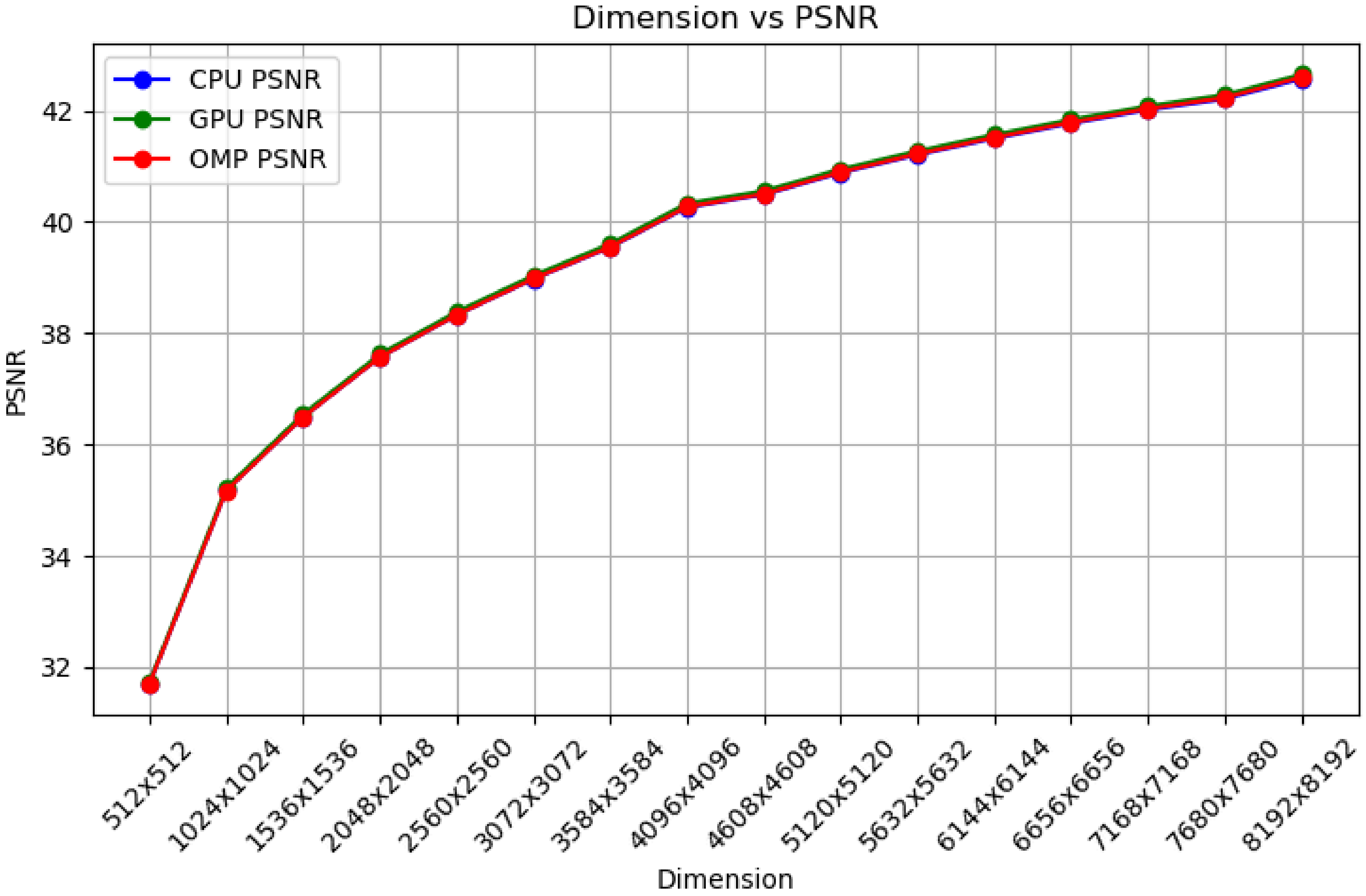






MSE AND PSNR


Dimension vs MSE
CPU MSE
GPU MSE
OMP MSE
MSE
45
40
35
30
25
20
15
10
5
512x512
1024x1024
1536x1536
2048x2048
2560x2560
3072x3072
3584x3584
4096x4096
4608x4608
5120x5120
5632x5632
6144x6144
6656x6656
7168x7168
7680x7680
8192x8192
Dimension


Dimension vs PSNR
CPU PSNR
GPU PSNR
OMP PSNR
PSNR
42
40
38
36
34
32
512x512
1024x1024
1536x1536
2048x2048
2560x2560
3072x3072
3584x3584
4096x4096
4608x4608
5120x5120
5632x5632
6144x6144
6656x6656
7168x7168
7680x7680
8192x8192
Dimension

Mean Squared Error (MSE)

- **Definition**: The MSE is the average squared difference between the pixel values of the original image and the compressed (reconstructed) image.
- **Formula**:

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j) - K(i,j))^2$$

  where:

  - $I(i,j)$ is the pixel value at position (i,j) in the original image.
  - $K(i,j)$ is the pixel value at the same position in the compressed image.
  - M and N are the dimensions of the image.
- **Range**: MSE values range from 0 to infinity. A lower MSE indicates less distortion and higher similarity to the original image.

Peak Signal-to-Noise Ratio (PSNR)

- **Definition**: PSNR is a logarithmic measure that relates the maximum possible pixel value in an image to the degradation introduced by compression (quantified by MSE).
- **Formula**:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right)$$

  where:

  - MAX is the maximum possible pixel value of the image (e.g., 255 for 8-bit images).
  - MSE is the Mean Squared Error.
- **Units**: PSNR is measured in decibels (dB).
- **Range**: Higher PSNR values indicate better quality. Typically:
  - > 30 dB: High quality.
  - $20\text{-}30$ dB: Moderate quality.
  - < 20 dB: Poor quality.

Why Are These Important in JPEG Compression?

JPEG compression introduces artifacts due to quantization and lossy encoding. Evaluating the MSE and PSNR provides an objective way to assess:

1. **Compression Efficiency**: Lower MSE and higher PSNR mean better reconstruction quality at a given compression ratio.
2. **Tuning Compression Parameters**: These metrics can guide adjustments in quality factors during JPEG encoding to balance quality and file size.

RESULTS

Original Image

Decompressed Image (CPU)

Decompressed Image (GPU)

Decompressed Image (OMP)

DEMONSTRATION

THANK YOU