Comparative Study of Lossy Image Compression Techniques

Project ID - 9
Course: Digital Image Analysis (CSL7320)
Semester: AY 2024–25, Semester – II

REPORT

Under the guidance of
Dr. Bikash Santra
Assistant Professor
School of AI and DS
Indian Institute of Technology Jodhpur

Submitted by:

Ratnesh Dubey M24CSA027
Ritul Jangir M24CSA028
Shivani Tiwari M24CSA029

April 2025

Table of Contents

1 Problem Statement

Lossy image compression is essential for applications where some loss of image quality is acceptable in exchange for higher compression ratios. This project involves a comparative study of different lossy image compression techniques, specifically DCT-based compression (simplified JPEG) and wavelet-based compression (simplified JPEG 2000), evaluating their performance in terms of compression efficiency, image quality, and computational requirements.

2 Introduction

Lossy image compression is critical in digital applications where storage space and transmission bandwidth are limited. This study implements and analyzes two fundamental approaches:

1. **DCT-based compression** (JPEG-like): Uses Discrete Cosine Transform on 8×8 pixel blocks
2. **Wavelet-based compression** (JPEG 2000-like): Uses multi-resolution wavelet transforms

The primary objectives are to:

- Implement both compression techniques
- Compare compression ratio vs. image quality trade-offs
- Analyze characteristic artifacts of each method
- Provide recommendations for specific application scenarios

3 Theoretical Concepts

3.1 DCT-based Compression

The Discrete Cosine Transform (DCT) is a frequency-domain transformation technique that expresses a sequence of data points as a sum of cosine functions oscillating at different frequencies. For image compression, the 2D DCT is applied to small blocks of pixels, typically 8×8, to convert spatial information into frequency coefficients.

For an 8×8 block of pixels, the 2D DCT is defined as:

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

Where:

- $f(x,y)$ is the pixel value at coordinates (x,y)

- $F(u, v)$ is the DCT coefficient at frequency coordinates (u, v)
- $C(u)$ and $C(v)$ are normalization factors:
  - $C(u) = \frac{1}{\sqrt{2}}$ when $u = 0$, and $C(u) = 1$ otherwise
  - $C(v) = \frac{1}{\sqrt{2}}$ when $v = 0$, and $C(v) = 1$ otherwise

The inverse DCT, used to reconstruct the image, is defined as:

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

3.1.1 DCT properties relevant to compression

1. **Energy Compaction**: For natural images, most of the signal energy is concentrated in the low-frequency coefficients (small u and v values). The DC coefficient at $(0, 0)$ represents the average value of the block.
2. **Decorrelation**: The DCT tends to decorrelate the image data, making many coefficients close to zero and thus highly compressible.
3. **Coefficient Significance**: The significance of coefficients typically follows a zigzag pattern from the top-left (low frequencies) to the bottom-right (high frequencies) of the DCT matrix.

3.1.2 Quantization in DCT compression

After DCT transformation, coefficients are quantized to reduce precision and increase compressibility:

$$F_Q(u, v) = \text{round}\left(\frac{F(u, v)}{Q(u, v)}\right) \tag{3}$$

Where $Q(u, v)$ is the value from the quantization matrix at position (u, v).
Two quantization approaches were implemented:

1. **Standard Quantization**: Uses a simple position-based weighting:

$$Q(i, j) = 1 + (i + j) \cdot \frac{100 - \text{quality}}{10} \tag{4}$$

2. **Perceptual Quantization**: Uses the standard JPEG luminance quantization matrix, which is designed based on human visual perception studies. The matrix is scaled

according to a quality factor:

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{5}$$

The quality scaling follows JPEG's standard approach:

$$\text{scale} = \begin{cases} \frac{5000}{\text{quality}} & \text{if quality} < 50 \\ 200 - \text{quality} \cdot 2 & \text{otherwise} \end{cases} \tag{6}$$

$$Q = \text{clip}(\text{round}((Q \cdot \text{scale})/100.0), 1, 255) \tag{7}$$

3.2 Wavelet-based Compression

Wavelet transforms decompose signals into components at different scales and positions, providing both frequency and spatial localization. Unlike the block-based DCT, wavelets operate on the entire image, avoiding blocking artifacts.

3.2.1 Wavelet transform fundamentals

The discrete wavelet transform (DWT) decomposes a signal through a series of filtering and downsampling operations. For a 1D signal, the process can be described as:

$$a_j[n] = \sum_k h[k - 2n] a_{j-1}[k] \tag{8}$$

$$d_j[n] = \sum_k g[k - 2n] a_{j-1}[k] \tag{9}$$

Where:

- $a_j[n]$ are approximation coefficients at level j
- $d_j[n]$ are detail coefficients at level j
- $h[n]$ and $g[n]$ are the low-pass and high-pass filter coefficients

For 2D images, the wavelet transform applies these filters along rows and columns, resulting in four subbands at each decomposition level:

1. **LL** (Low-Low): Approximation coefficients - lower resolution version of the image
2. **LH** (Low-High): Horizontal detail coefficients - detect horizontal edges
3. **HL** (High-Low): Vertical detail coefficients - detect vertical edges
4. **HH** (High-High): Diagonal detail coefficients - detect diagonal features

3.2.2 Wavelet types implemented

1. **Haar Wavelet**:
   - Simplest wavelet with compact support
   - Filter coefficients:
     - Low-pass: $h = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$
     - High-pass: $g = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$
   - Basis function is a step function, making it effective for detecting sharp transitions
   - Limited frequency localization due to discontinuities
2. **Daubechies-4 (DB4) Wavelet**:
   - Four vanishing moments (can represent polynomials up to degree 3 exactly)
   - Filter coefficients:
     - Low-pass: $h = [0.4830, 0.8365, 0.2241, -0.1294]$
     - High-pass: $g = [-0.1294, -0.2241, 0.8365, -0.4830]$
   - Smoother basis functions than Haar
   - Better frequency localization and preservation of smooth regions

3.2.3 Coefficient thresholding for wavelet compression

Compression is achieved by thresholding wavelet coefficients, setting small values to zero:

$$\hat{d}_j[n] = \begin{cases} d_j[n], & \text{if } |d_j[n]| \geq T \\ 0, & \text{if } |d_j[n]| < T \end{cases} \tag{10}$$

Where T is the threshold value.

The implementation uses adaptive thresholding based on coefficient percentiles:

1. Calculate the magnitude of all detail coefficients
2. Set threshold at the (100-P)th percentile, where P is the threshold percentage parameter
3. Apply hard thresholding to detail coefficients
4. Preserve approximation coefficients (LL band) to maintain overall image structure

This adaptive approach ensures consistent quality across different image types, as the threshold adjusts to the distribution of coefficient values in each image.

3.3 Quality Assessment

3.3.1 Objective quality metrics

1. **Peak Signal-to-Noise Ratio (PSNR):** Measures the ratio between the maximum possible power of a signal and the power of corrupting noise:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right) \tag{11}$$

   Where:

   - MAX_I is the maximum possible pixel value (255 for 8-bit images)
   - MSE (Mean Squared Error) is calculated as:

$$\text{MSE} = \frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{12}$$

   - I is the original image, K is the compressed image
   - m, n are the image dimensions

2. **Structural Similarity Index (SSIM):** Measures perceptual similarity by considering structural information:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{13}$$

   Where:

   - μ_x and μ_y are the average pixel values
   - σ_x^2 and σ_y^2 are the variances
   - σ_{xy} is the covariance
   - c_1 and c_2 are constants to stabilize division with weak denominators

3.3.2 Subjective quality assessment

A 5-point quality scale based on PSNR and SSIM thresholds:

4 Methodology

4.1 Implementation Pipeline

4.1.1 DCT-based Compression

1. Divide image into 8×8 blocks

Table 1: Subjective Quality Scale

Score	Category	PSNR	SSIM	Description
5	Excellent	> 40 dB	> 0.97	No visible compression artifacts
4	Good	> 35 dB	> 0.94	Compression artifacts not noticeable in normal viewing
3	Fair	> 30 dB	> 0.90	Minor artifacts visible upon close inspection
2	Poor	> 25 dB	> 0.80	Visible artifacts affecting viewing experience
1	Bad	≤ 25 dB	≤ 0.80	Severe artifacts significantly degrading image quality

2. Apply DCT to each block (shift values by -128 first)
3. Quantize coefficients using either:
   - Standard quantization: Based on frequency position
   - Perceptual quantization: Using JPEG luminance matrix
4. Count non-zero coefficients for compression ratio
5. Reconstruct image using inverse DCT

4.1.2 Wavelet-based Compression

1. Apply wavelet decomposition to 3 levels
2. Calculate threshold based on coefficient percentile
3. Apply thresholding to detail coefficients
4. Preserve approximation coefficients
5. Reconstruct image using inverse wavelet transform

4.2 Experimental Setup

- Test images selected from UCID dataset
- Quality parameters varied systematically:
  - DCT: Quality factors 80, 60, 40, 20, 10, 5
  - Wavelet: Threshold percentages 50, 30, 20, 10, 5, 2

- Metrics calculated for each compression setting
- Artifact visualization with zoomed regions

5 Results and Observations

5.1 Single Image Analysis

Analysis was performed on image "10.tif" (512×384 pixels) with both compression techniques.

Table 2: Compression Performance Comparison at Similar Ratios

Method	**Setting**	**Ratio**	**PSNR (dB)**	**SSIM**	**Time (ms)**
DCT Standard	Q=60	3.30	32.34	0.9340	235.03
DCT Perceptual	Q=60	3.85	29.98	0.9165	497.03
Haar Wavelet	T=30%	3.22	36.19	0.9470	48.49
DB4 Wavelet	T=30%	3.20	36.47	0.9514	90.19
DCT Perceptual	Q=20	8.09	25.64	0.8242	290.61
Haar Wavelet	T=10%	8.75	27.81	0.7860	55.91
DB4 Wavelet	T=10%	8.59	28.20	0.8076	70.66
DCT Perceptual	Q=5	24.57	22.04	0.6304	397.60
Haar Wavelet	T=2%	28.31	22.27	0.5944	56.75
DB4 Wavelet	T=2%	26.38	22.79	0.6276	53.18

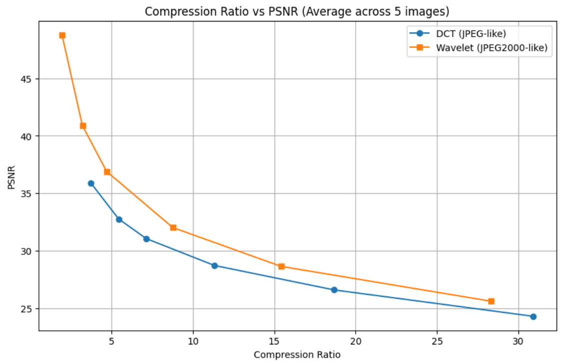
5.2 Multi-Image Analysis

Results averaged across 5 UCID dataset images:

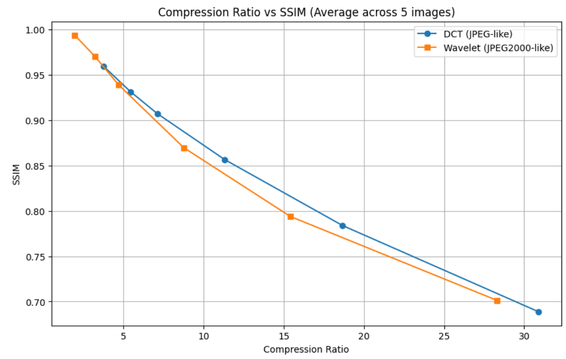
Table 3: Average Performance Across Dataset (Selected Parameters)

Method	**Ratio**	**PSNR (dB)**	**SSIM**	**Execution Time**
DCT (Q=60)	5.09	31.99	0.9242	Slower (200-400ms)
Wavelet (T=30%)	3.21	39.14	0.9628	Faster (40-90ms)

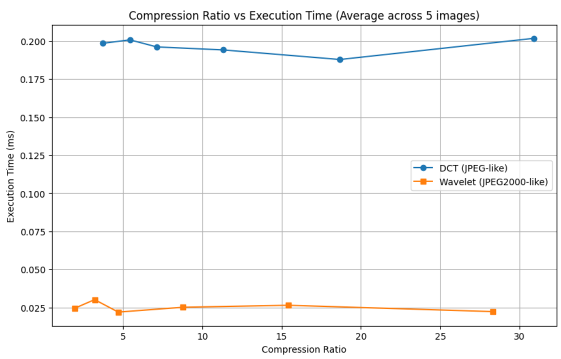
5.3 Comparison Plots



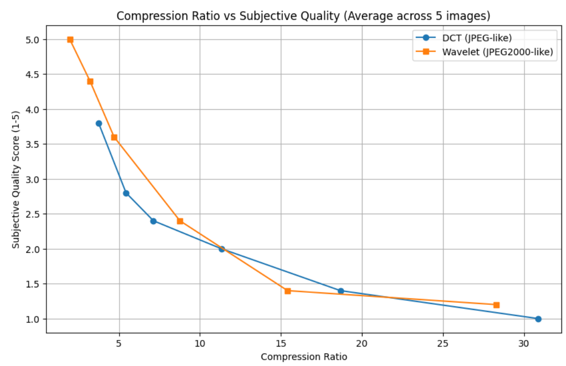
(a) Compression ratio vs PSNR



(b) Compression ratio vs SSIM



(c) Compression ratio vs Execution Time



(d) Compression ratio vs Subjective Quality

Figure 1: Average results Comparison across Dataset

5.4 Key Observations

- **Quality vs. Compression:** At low compression ratios (<4:1), wavelets delivered higher quality. At high compression ratios (>10:1), DCT with perceptual quantization performed better in terms of structural preservation.
- **Computational Efficiency:** Wavelet-based compression was 4-6× faster than DCT methods (48ms vs 235ms for similar compression ratios).
- **Artifact Characteristics:**
  - DCT: Blocking artifacts (grid patterns), more pronounced ringing at edges
  - Wavelet: Blurring of details, less visible blocking, better preservation of overall structure
- **Content Dependency:** Wavelets performed better on natural images with smooth regions; DCT better preserved text and sharp edges at moderate compression levels.

5.5 Optimal Application Scenarios

- **DCT Compression:** Preferred for moderate compression (5:1-15:1), fast viewing applications, and compatibility with existing JPEG infrastructure.
- **Wavelet Compression:** Ideal for high compression ratios (>15:1), applications where blocking artifacts must be avoided, and when computational speed at decompression is more important than at compression.

6 Challenges Faced and Solutions Implemented

6.1 Challenge 1: Block Boundary Artifacts in DCT

- **Problem:** Initial DCT implementation showed visible seams at block boundaries
- **Cause:** Rounding errors during quantization created inconsistencies between blocks
- **Solution:** Implemented precise rounding in quantization and consistent handling of block edges

6.2 Challenge 2: Wavelet Threshold Selection

- **Problem:** Fixed threshold values resulted in inconsistent quality across different images
- **Cause:** Different images have different coefficient distributions
- **Solution:** Implemented adaptive percentile-based thresholding, analyzing each image's coefficient distribution

6.3 Challenge 3: Edge Effects in Wavelet Transform

- **Problem:** Wavelet reconstruction produced artifacts at image edges
- **Cause:** The wavelet transform can extend beyond original image dimensions
- **Solution:** Implemented proper cropping of reconstructed images to original dimensions

6.4 Challenge 4: Compression Ratio Calculation

- **Problem:** Initial compression ratios didn't reflect actual file size reduction
- **Cause:** Implementation didn't account for potential entropy coding
- **Solution:** Developed a coefficient-based ratio calculation that counted non-zero elements

6.5 Challenge 5: Memory Efficiency for Large Images

- **Problem:** Processing high-resolution images caused memory issues
- **Cause:** Temporary arrays created during transform steps consumed excessive memory
- **Solution:** Implemented more memory-efficient processing by operating on image blocks when possible

7 Conclusion

This comparative study demonstrated the distinct advantages and limitations of DCT and wavelet-based compression techniques:

1. **DCT-based compression excels at:**
   - Moderate compression ratios (5:1 to 15:1)
   - Faster execution time
   - Better performance for text and sharp-edged content
   - Compatibility with existing JPEG infrastructure
2. **Wavelet-based compression excels at:**
   - High compression ratios (>15:1)
   - Avoiding blocking artifacts
   - Preserving smooth transitions and large-scale features
   - Potential for progressive decoding

The study provided practical insights for selecting the appropriate compression technique based on specific application requirements. The implementation challenges encountered and solutions developed offer valuable lessons for real-world image compression applications. Future work could explore hybrid approaches that combine the strengths of both techniques, as well as incorporate entropy coding for more complete codec implementations.

8 Links

- Colab Link can be access from here: Colab Link.
- The Project is Live at : Streamlit Link.