Wavelet-Based Image Compression: A Comparative Analysis

Group 7
Yavuz Selim Çanakçı - 23050111005
Gürkan AĞIR - 20050111073
Muhammet Emir Zayıfoğlu - 20050111032
Abdullah Enes Güzlek - 22050161002

December 16, 2025

Abstract

This project investigates the application of Discrete Wavelet Transform (DWT) for lossy image compression, aiming to balance file size reduction with visual quality preservation. We compare three distinct wavelet families—Haar, Daubechies-2, and Symlet-4—across four diverse image categories: digitally generated checkered patterns, satellite imagery, natural photographs, and patterned textures. Using a threshold-based coefficient elimination method, we analyze the trade-off between compression ratio (sparsity) and reconstruction quality (PSNR). Our experimental results demonstrate that wavelet selection is highly content-dependent; Haar excels for sharp-edged binary images, whereas Symlet-4 provides superior performance for complex textures and natural scenes. Furthermore, we establish 30 dB as a critical perceptual quality threshold, below which image degradation becomes visually significant.

1 Introduction

1.1 Problem Statement and Motivation

Digital images constitute a vast majority of data transmitted over networks and stored in databases today. With the exponential growth of image data sources such as high-resolution satellite systems, medical imaging (MRI, CT), and digital photography, efficient compression techniques have become essential. Traditional compression methods often struggle to provide an optimal balance between minimizing file size and preserving essential visual details.

Wavelet-based compression offers significant advantages over these traditional methods, primarily due to its multi-resolution analysis capability. This allows for capturing both spatial and frequency information simultaneously. Key benefits include energy compaction, where most image information is concentrated in a few coefficients, and adaptive compression, allowing different regions of an image to be processed at varying rates.

1.2 Project Objectives

This study aims to:

- Implement and compare the performance of different wavelet families (Haar, db2, sym4).
- Quantify the trade-off between sparsity (compression ratio) and image quality (PSNR).
- Establish practical quality thresholds for real-world applications.
- Provide clear guidelines for selecting the appropriate wavelet based on specific image characteristics.

2 Methodology

2.1 Discrete Wavelet Transform (DWT)

The 2D Discrete Wavelet Transform decomposes an image into four distinct subbands at each level:

- **LL (Approximation):** Contains the low-frequency components and the main structure of the image.
- **LH (Horizontal Details):** Captures vertical edges and horizontal features.
- **HL (Vertical Details):** Captures horizontal edges and vertical features.
- **HH (Diagonal Details):** Captures diagonal edges and corner features.

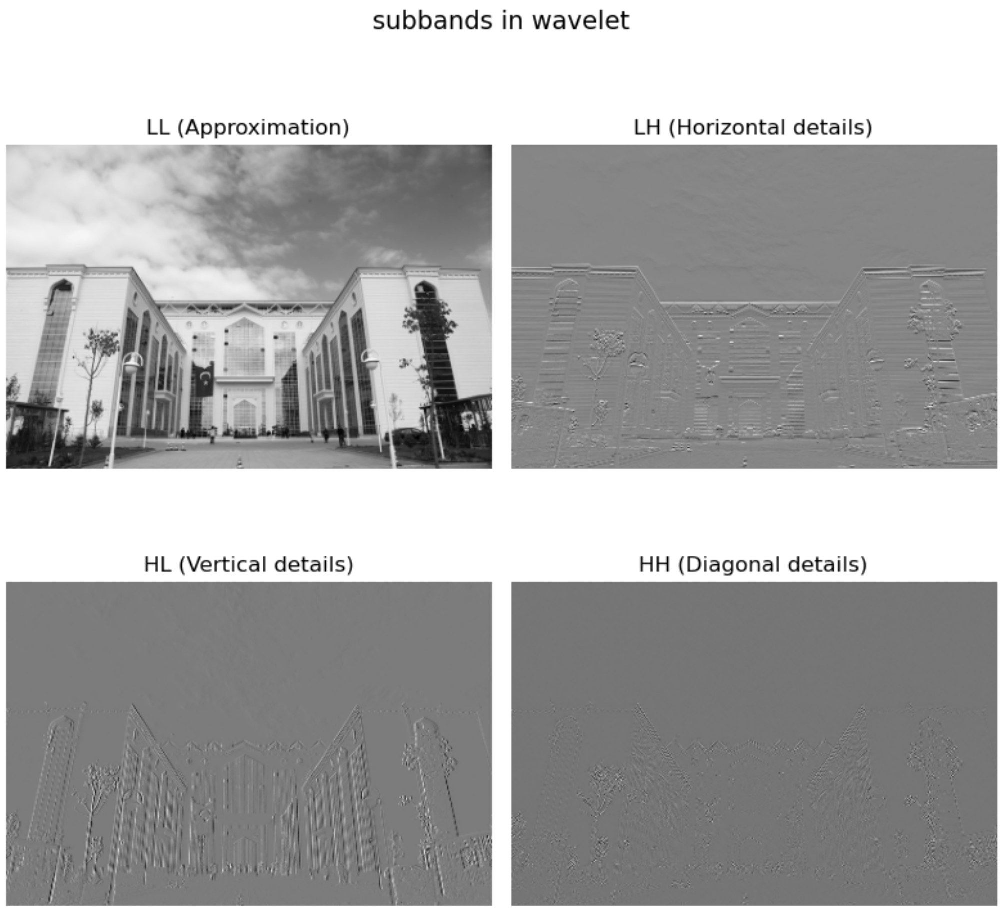


Figure 1: Visual representation of LL, LH, HL, and HH subbands generated from the test image. Note how the LL band retains the structural "ghost" of the original image.

2.2 Wavelet Families Evaluated

We selected three wavelet families to test different mathematical properties:

1. **Haar Wavelet:** The simplest wavelet with discontinuous, rectangular basis functions. It is computationally efficient and ideal for signals with sharp transitions.

2. **Daubechies-2 (db2):** Offers a smoother basis than Haar with 2 vanishing moments, making it better suited for gradual transitions in natural images.

3. **Symlet-4 (sym4):** A modified version of Daubechies with better symmetry and 4 vanishing moments, providing balanced performance for complex textures.

2.3 Compression Algorithm Metrics

Compression is achieved by applying a hard threshold to the detail coefficients. Coefficients with absolute values smaller than a threshold T are set to zero, effectively increasing sparsity. The thresholding logic is defined as:

$$\tilde{c}_{ij} = \begin{cases} c_{ij} & \text{if } |c_{ij}| \geq T \\ 0 & \text{if } |c_{ij}| < T \end{cases} \tag{1}$$

We evaluated threshold values ranging from 0.05 to 0.35. The quality of the reconstructed image is measured using the Peak Signal-to-Noise Ratio (PSNR), where values above 30 dB are generally considered acceptable:

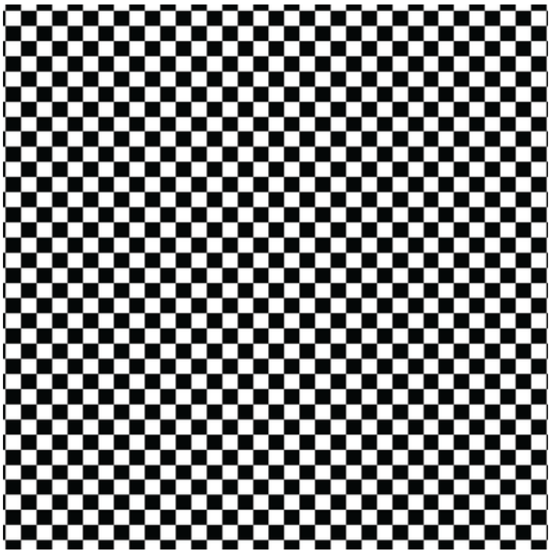
$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \text{ dB} \tag{2}$$

3 Data and Implementation

3.1 Dataset Description

To ensure a robust evaluation, we selected four image types representing different frequency characteristics:

- **Digital Checkered:** Binary, high-contrast, sharp edges.
- **Satellite Image:** Complex textures, multiple frequency components.
- **Natural Photo:** Smooth gradients, low-frequency dominance.
- **Patterned Texture:** Regular, repeating structures.



(a) Digital Checkered (Binary)

(b) Satellite Image (Complex)

(c) Natural Photo (Smooth)

(d) Patterned Texture

Figure 2: The diverse dataset samples used for compression experiments.

3.2 Preprocessing

All images are converted to grayscale using standard luminance weights and normalized to a [0, 1] float range to ensure consistent thresholding behavior.

```
def prep_img(path):
    img = mpimg.imread(path)
    # Convert RGB to grayscale using luminance weights
    if img.ndim == 3:
        gray = np.dot(img[...,:3], [0.2989, 0.5870, 0.1140])
    else:
        gray = img
    # Convert to float32 and normalize to [0, 1]
    gray = gray.astype(np.float32)
    if gray.max() > 1.5:
        gray = gray / 255.0
```

```
    return gray
```

Listing 1: Image Preprocessing Function

3.3 Compression Pipeline Implementation

The core compression logic involves wavelet decomposition, threshold application across all subbands, and inverse reconstruction.

```
def compress_and_evaluate_images(path, wavelet_name, threshold):
    # Step 1: Preprocess
    gray = prep_img(path)
    # Step 2: Apply wavelet transform
    coeffs2 = pywt.dwt2(gray, wavelet_name)
    coeff_A, (coeff_H, coeff_V, coeff_D) = coeffs2
    # Step 3: Apply threshold to all subbands
    coeff_A_thr, zeros_A = apply_threshold(coeff_A, threshold)
    coeff_H_thr, zeros_H = apply_threshold(coeff_H, threshold)
    coeff_V_thr, zeros_V = apply_threshold(coeff_V, threshold)
    coeff_D_thr, zeros_D = apply_threshold(coeff_D, threshold)
    # Step 4: Reconstruct image
    coeffs2_thr = (coeff_A_thr, (coeff_H_thr, coeff_V_thr, coeff_D_thr))
    gray_c, reconstructed_c = reconstruct(coeffs2_thr, gray,
   wavelet_name)
    # Step 5: Evaluate quality
    mse, psnr = calculate_mse_psnr(gray_c, reconstructed_c)
    return psnr, sparsity
```

Listing 2: Compression and Evaluation Function

4 Results and Discussion

4.1 Trade-off Analysis by Image Type

Digital Checkered Image

This image represents an extreme case of high-frequency transitions. The **Haar wavelet** achieves the highest performance, maintaining PSNR $>$ 30 dB even at 60% sparsity. This is because Haar's rectangular basis functions align perfectly with the sharp, pixel-perfect edges of the checkered pattern. Conversely, smoother wavelets like sym4 introduce ringing artifacts (Gibbs phenomenon) around edges, reducing performance.

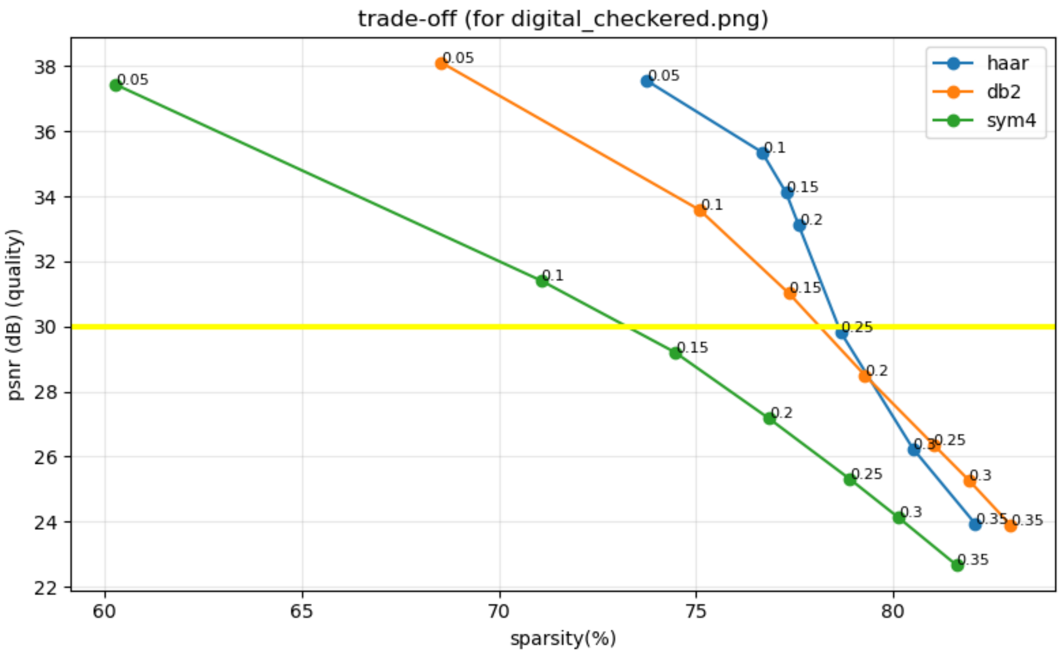


Figure 3: Sparsity vs. PSNR trade-off for Digital Checkered Image.

Satellite Image

Satellite imagery contains complex textures and details at multiple scales. Here, **Symlet-4** consistently outperforms Haar. The Haar wavelet degrades quality rapidly at moderate compression levels because it cannot effectively represent the smooth, continuous features of the terrain. Symlet-4 maintains acceptable quality (>30 dB) up to 55% sparsity.

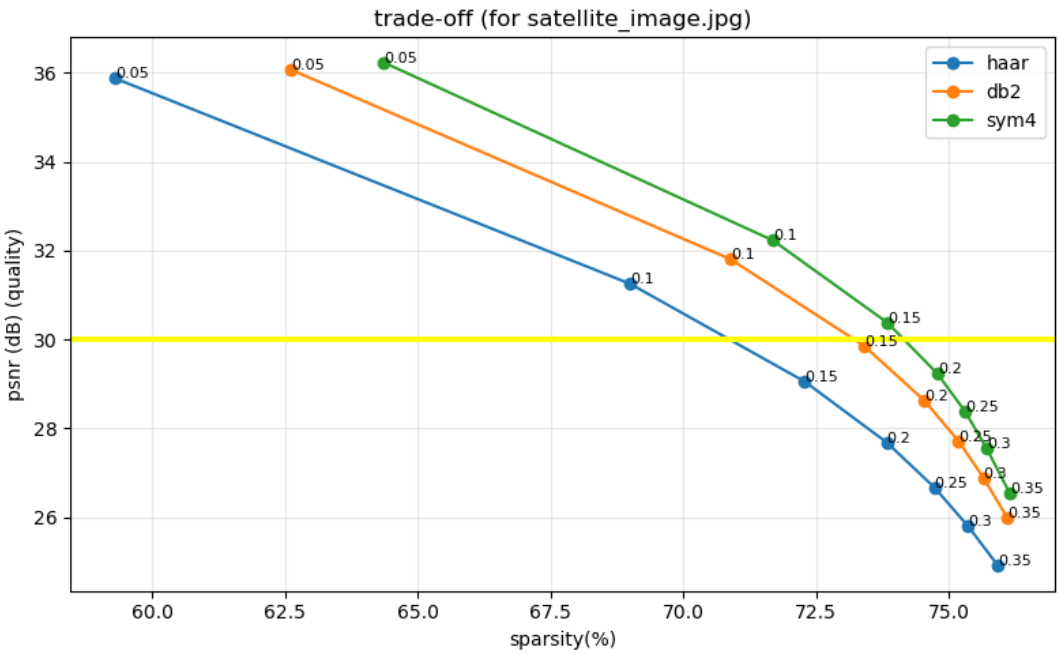


Figure 4: Sparsity vs. PSNR trade-off for Satellite Image.

Natural Photograph

For standard natural photographs, which are dominated by low-frequency information and smooth gradients, the choice of wavelet has a minimal impact. All three wavelets exhibit nearly identical performance curves. However, **Symlet-4** shows a slight advantage at higher compression ratios due to its higher vanishing moments, which better approximate smooth polynomial functions.

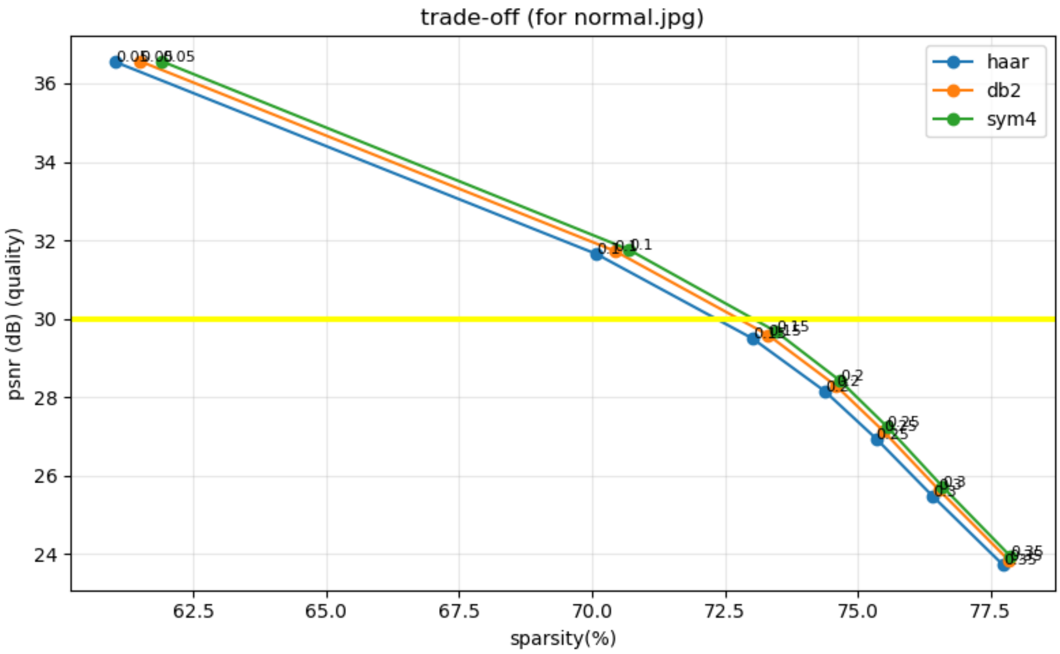


Figure 5: Sparsity vs. PSNR trade-off for Natural Photograph.

Patterned Image

The patterned texture benefits significantly from the symmetry of the wavelet basis. **Symlet-4** outperforms other wavelets, demonstrating that regular, repeating structures are best compressed using symmetric basis functions. The performance gap between Symlet-4 and Haar widens significantly at higher sparsity levels (>50%).

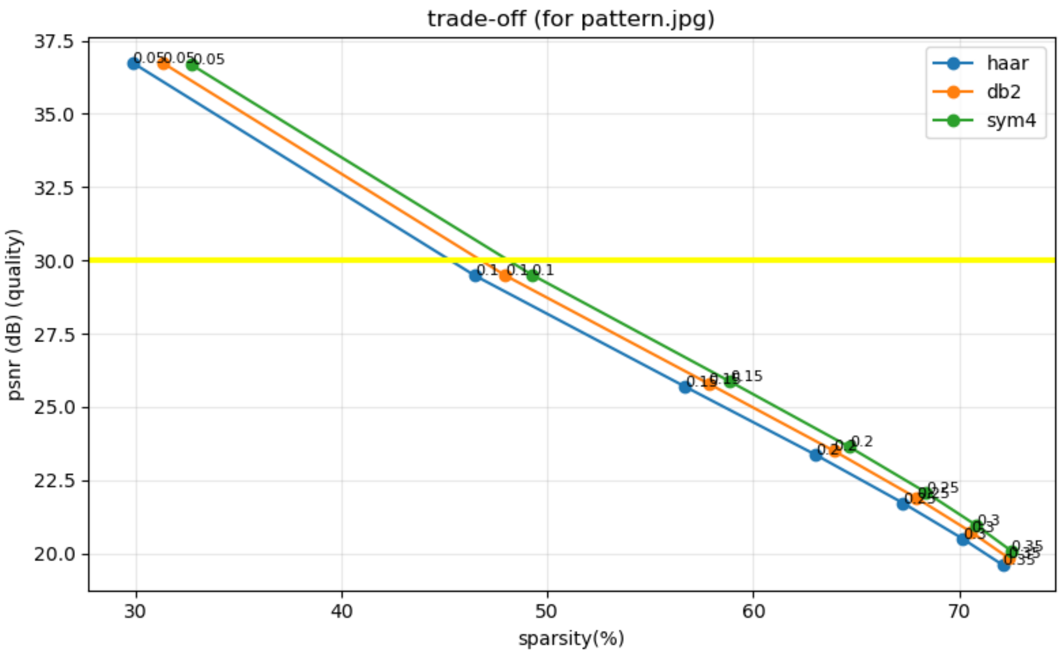


Figure 6: Sparsity vs. PSNR trade-off for Patterned Texture.

4.2 The 30 dB Quality Threshold

Visual analysis of the reconstructed images confirms that 30 dB represents a critical perceptual threshold.

- **Below 30 dB:** Images display visible degradation, including blockiness, blurring of fine details, and loss of texture.
- **Above 30 dB:** Images generally retain high visual fidelity with minimal perceptible artifacts, making them suitable for most commercial applications.

(a) Low Quality (PSNR $\approx$ 15 dB) (b) High Quality (PSNR $\approx$ 32 dB)

Figure 7: Visual comparison of reconstruction quality. Note the significant loss of detail and "blacking out" artifacts in the low-quality image compared to the high-fidelity reconstruction.

4.3 Practical Guidelines for Wavelet Selection

Based on our findings, we propose the following selection strategy:

- **Use Haar:** When the image consists of binary data, text, or sharp artificial edges, or when computational speed is the primary constraint.

- **Use Symlet-4:** As a general-purpose compressor for natural scenes, satellite imagery, and mixed content where maximum quality is desired.
- **Use Daubechies-2:** For images with smooth transitions where a balance between complexity and performance is needed.

5 Conclusion and Future Work

This project successfully implemented and evaluated wavelet-based image compression. We demonstrated that sparsity levels of 40-60% are achievable while maintaining the 30 dB quality standard. The key takeaway is that wavelet selection must be image-dependent; there is no "one-size-fits-all" solution.

5.1 Limitations

- The current implementation uses single-level decomposition. Multi-level DWT could yield higher compression ratios.
- The study focuses on grayscale images; color channels (RGB) would require separate processing.
- A global threshold was applied; adaptive thresholding based on local variance could improve results.

5.2 Future Improvements

Future work will focus on implementing multi-level decomposition to exploit redundancy at coarser scales, adding support for color images, and exploring adaptive thresholding techniques. Additionally, integrating perceptual metrics like SSIM (Structural Similarity Index) alongside PSNR would provide a more comprehensive quality assessment.

References

1. PyWavelets Documentation: https://pywavelets.readthedocs.io/
2. Daubechies, I. (1992). "Ten Lectures on Wavelets." SIAM.
3. Mallat, S. (1989). "A Theory for Multiresolution Signal Decomposition."
4. Said, A., & Pearlman, W. A. (1996). SPIHT Algorithm.
5. Wang, Z., et al. (2004). Image Quality Assessment: SSIM.
6. JPEG2000 Standard - ISO/IEC 15444-1.
7. NumPy & Matplotlib Documentation.