World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Research Article)

# Image Compression Techniques: A Comparative Study

Chethana N S [1,*] and Veena N D [2]

[1] Lecturer, Department of Electronics and Communication Engineering, Government Polytechnic, Hiriyur- 577599, Karnataka, India.

[2] Assistant Professor, Department of Computer Science and Engineering, SIET, Tumkur 572106, Karnataka, India.

World Journal of Advanced Research and Reviews, 2020, 06(03), 323-333

Publication history: Received on 08 June 2020; revised on 17 June 2020; accepted on 23 June 2020

Article DOI: https://doi.org/10.30574/wjarr.2020.6.3.0167

## Abstract

Image compression has become an essential technology in the digital era, enabling efficient storage and transmission of visual data across networks and devices. This paper presents a comprehensive comparative study of various image compression techniques, examining both lossless and lossy compression methods. We analyze traditional techniques such as JPEG and PNG alongside modern approaches including wavelet-based compression and fractal compression. The study evaluates these techniques based on compression ratio, image quality metrics including Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), computational complexity, and application suitability. Our findings reveal that while lossy techniques achieve superior compression ratios, lossless methods maintain perfect reconstruction quality at the cost of larger file sizes. The choice of compression technique depends significantly on the specific application requirements, with medical imaging demanding lossless compression while web applications often prioritize smaller file sizes. This research contributes to the understanding of compression technique selection and provides guidelines for practitioners in various domains. The comparative analysis includes experimental results on standard test images, demonstrating the trade-offs between compression efficiency and image fidelity.

**Keywords:** Image Compression; Lossy Compression; Lossless Compression; JPEG; JPEG 2000; JPEG-LS; PNG; Rate–Distortion Analysis; PSNR; Bitrate; Compression Ratio; Image Quality Assessment; Digital Image Processing

## 1. Introduction

proliferation of digital imaging devices and multimedia applications has led to an exponential increase in image data generation and consumption. Modern smartphones, digital cameras, medical imaging equipment, and satellite systems produce millions of images daily, creating substantial challenges for storage and transmission infrastructure. Image compression addresses these challenges by reducing the amount of data required to represent visual information while maintaining acceptable quality levels. The fundamental principle underlying image compression is the exploitation of redundancy inherent in digital images, including spatial redundancy where neighboring pixels often have similar values, spectral redundancy in color images where color components are correlated, and psychovisual redundancy where the human visual system cannot perceive certain information losses (Sayood, 2017). Understanding these redundancies is crucial for developing effective compression algorithms that balance file size reduction with perceptual quality preservation.

The history of image compression dates back to the 1950s when television transmission bandwidth limitations necessitated efficient encoding methods. Early techniques focused on simple run-length encoding and predictive coding schemes that achieved modest compression ratios. The landscape changed dramatically in the 1980s and 1990s with the development of transform-based methods, particularly the Discrete Cosine Transform (DCT) which became the foundation for the JPEG standard (Wallace, 1992). Concurrently, wavelet-based approaches emerged, offering multi-resolution analysis capabilities that proved superior for certain image types. The International Organization for

*Corresponding author:. Chethana N S

Standardization (ISO) and the International Telecommunication Union (ITU) established standards that have shaped modern image compression, with JPEG becoming the most widely adopted standard for photographic images and PNG serving as the preferred format for lossless compression needs.

Image compression techniques are broadly classified into two categories: lossless and lossy compression. Lossless compression ensures perfect reconstruction of the original image after decompression, making it indispensable for medical imaging, legal documentation, and technical drawings where no information loss is tolerable. Common lossless techniques include Huffman coding, arithmetic coding, Lempel-Ziv-Welch (LZW) algorithm, and predictive coding methods (Salomon, 2007). These techniques typically achieve compression ratios ranging from 2:1 to 4:1 for natural images. In contrast, lossy compression methods deliberately discard information deemed less important to human perception, achieving significantly higher compression ratios often exceeding 20:1 while maintaining visually acceptable quality. The irreversible nature of lossy compression makes it suitable for applications where storage efficiency outweighs the need for perfect reconstruction, such as web publishing, digital photography, and video streaming.

The evaluation of compression techniques requires multiple metrics beyond simple compression ratio calculations. Peak Signal-to-Noise Ratio (PSNR) has traditionally served as the primary objective quality metric, measuring the ratio between the maximum possible signal power and the power of corrupting noise (Huynh-Thu & Ghanbari, 2008). However, PSNR often correlates poorly with human perception, leading to the development of perceptual quality metrics like the Structural Similarity Index (SSIM) which considers luminance, contrast, and structural information. Computational complexity, measured in terms of encoding and decoding time, significantly impacts the practical applicability of compression techniques, particularly in real-time applications. Memory requirements during compression and decompression operations constitute another critical factor, especially for resource-constrained embedded systems and mobile devices. The Multi-Scale Structural Similarity (MS-SSIM) index extends SSIM by incorporating multi-scale processing, providing more robust quality assessment across various viewing conditions and image resolutions.

The selection of an appropriate compression technique depends on multiple application-specific factors that must be carefully considered. Medical imaging applications prioritize diagnostic accuracy, requiring lossless compression despite lower compression ratios to preserve subtle pathological features. Remote sensing and satellite imagery demand techniques that maintain spectral accuracy across multiple bands while managing the vast data volumes generated by earth observation systems. Web applications emphasize fast decoding times and small file sizes to enhance user experience and reduce bandwidth costs, often accepting quality degradation invisible to casual viewers. Mobile photography applications must balance image quality with storage limitations and battery consumption constraints, leading to adaptive compression strategies that adjust parameters based on available resources. The emergence of high dynamic range (HDR) imaging and ultra-high definition formats has further complicated the compression landscape, requiring techniques that efficiently encode extended color gamuts and increased spatial resolution.

This paper systematically compares major image compression techniques through theoretical analysis and experimental validation on standard test images. Section 2 reviews fundamental concepts in image compression, including redundancy types, information theory principles, and human visual system characteristics that inform compression algorithm design. Section 3 examines lossless compression techniques with detailed analysis of Huffman coding, arithmetic coding, run-length encoding, and dictionary-based methods, supported by comparative performance data in Table 1. Section 4 investigates lossy compression techniques, focusing on transform-based methods including DCT, wavelet transforms, and fractal compression, with quality comparisons illustrated in Figure 1. Section 5 presents experimental results comparing all discussed techniques across multiple test images, analyzing compression efficiency, quality metrics, and computational requirements as shown in Tables 2 and 3 and Figures 2 and 3. Section 6 concludes with recommendations for technique selection based on application requirements and identifies future research directions in neural network-based compression and perceptual optimization.

## 2. Fundamental Concepts in Image Compression

Digital images consist of discrete picture elements called pixels, each representing color and intensity information at a specific spatial location. An uncompressed color image with dimensions $M \times N$ pixels and B bits per color channel requires $M \times N \times C \times B$ bits of storage, where C represents the number of color channels (typically three for RGB images). For instance, a standard 8-megapixel photograph with 24-bit color depth occupies approximately 24 megabytes of storage without compression. The information content of an image can be quantified using Shannon's information theory, where entropy represents the minimum average number of bits required to encode information from a source (Shannon, 1948). The entropy H of a discrete source is calculated as $H = -\Sigma\, p(x)\log_2 p(x)$, where p(x) represents the probability of symbol x occurring. Images typically contain significantly less information than their uncompressed representation suggests, providing opportunities for compression by removing redundant data while preserving essential information.

Spatial redundancy arises from the correlation between neighboring pixels in natural images, where adjacent pixels typically have similar or identical values. This property results from the continuous nature of real-world scenes where object surfaces and illumination patterns change gradually rather than abruptly. Predictive coding exploits spatial redundancy by encoding the difference between predicted and actual pixel values rather than absolute values. The predicted value can be computed using simple methods like the value of the previous pixel or sophisticated techniques involving multiple neighboring pixels. When differences are encoded using variable-length codes that assign shorter codes to frequently occurring small differences, significant compression is achieved. The Markov Random Field model provides a theoretical framework for understanding spatial correlations, assuming that a pixel's value depends primarily on its immediate neighbors rather than distant pixels (Geman & Geman, 1984).

Spectral redundancy exists in color images where the three color channels (red, green, blue) are highly correlated due to physical properties of natural scenes and illumination. Converting from RGB to alternative color spaces like YCbCr separates luminance (Y) from chrominance (Cb, Cr) components, enabling more efficient compression since human vision is more sensitive to luminance variations than chrominance changes. The transformation concentrates energy into fewer coefficients, allowing aggressive quantization of chrominance components with minimal perceptual impact. Decorrelation transforms like the Karhunen-Loève Transform (KLT) optimally remove spectral redundancy by producing uncorrelated components, though computational complexity limits practical application. The International Commission on Illumination (CIE) LAB color space offers perceptual uniformity, ensuring that equal distances in the color space correspond to roughly equal perceptual differences, which benefits perceptual compression optimization.

Psychovisual redundancy encompasses information that exists in the image but cannot be perceived by the human visual system, making it removable without subjective quality degradation. The contrast sensitivity function (CSF) describes the human eye's varying sensitivity to spatial frequencies, with peak sensitivity at approximately 4 cycles per degree of visual angle and reduced sensitivity at higher and lower frequencies (Mannos & Sakrison, 1974). Texture masking refers to the phenomenon where distortions are less visible in highly textured regions compared to smooth areas, allowing more aggressive compression in complex image regions. Luminance masking indicates that compression artifacts are less perceptible in very bright or very dark regions compared to mid-tone areas. The just-noticeable difference (JND) quantifies the minimum amount of distortion detectable by human observers under specific viewing conditions, providing guidance for perceptual quantization design. Temporal masking in video compression further exploits the visual system's limited temporal resolution, though this paper focuses primarily on still image compression.

Transform coding forms the theoretical foundation for many successful compression techniques by converting spatial domain image data into an alternative representation where energy concentrates in fewer coefficients. Linear transforms map the N-dimensional input vector to an N-dimensional coefficient vector through multiplication by a transformation matrix, with the inverse transform enabling perfect reconstruction. Unitary transforms preserve signal energy while decorrelating signal components, making them ideal for compression applications. The Discrete Cosine Transform (DCT) approximates the optimal KLT for highly correlated signals while offering fast implementation algorithms and avoiding complex arithmetic (Ahmed et al., 1974). Wavelet transforms provide multi-resolution analysis, decomposing images into approximation and detail components at multiple scales, which naturally aligns with hierarchical image structure and enables progressive transmission capabilities. The choice of transform significantly impacts compression performance, with DCT excelling for smooth regions and wavelets providing superior performance for images with edges and texture.

Quantization constitutes the principal lossy operation in compression algorithms, reducing the precision of transformed coefficients to decrease the number of bits required for representation. Scalar quantization independently maps each coefficient to a discrete set of reconstruction levels, with uniform quantization using equally spaced levels and non-uniform quantization adapting spacing to coefficient statistics. The quantization step size controls the trade-off between compression ratio and distortion, with larger steps producing greater compression but introducing more quantization error. Dead-zone quantizers set a range of small coefficients to zero, exploiting the fact that small coefficients contribute minimally to perceptual quality while consuming encoding bits. Vector quantization simultaneously quantizes groups of coefficients, achieving better rate-distortion performance than scalar quantization at the cost of increased computational complexity and codebook storage requirements. Perceptually weighted quantization incorporates human visual system characteristics, applying finer quantization to perceptually important coefficients and coarser quantization to less important ones, as exemplified by the quantization matrices in JPEG compression (Pennebaker & Mitchell, 1993).

## 3. Lossless Compression Techniques

Huffman coding represents one of the most widely used lossless compression techniques, employing variable-length codes where frequently occurring symbols receive shorter codes than rare symbols (Huffman, 1952). The algorithm constructs an optimal prefix-free binary tree by iteratively combining the two lowest-probability symbols into a parent node, continuing until a single root node remains. Code assignment proceeds by traversing from the root to each leaf, concatenating binary digits (0 for left, 1 for right) to form each symbol's codeword. The optimality of Huffman coding guarantees that no other prefix-free code achieves a shorter average codeword length for the given symbol probabilities. Adaptive Huffman coding eliminates the need for transmitting the code table by dynamically updating the tree structure as symbols are processed, though this adds computational overhead. While Huffman coding achieves near-entropy compression for integer-bit codeword lengths, it proves suboptimal when optimal codeword lengths are non-integer values, motivating alternative approaches.

Arithmetic coding addresses Huffman coding's limitations by representing entire messages as single fractional numbers in the interval [0,1), achieving compression rates closer to the entropy limit (Witten et al., 1987). The algorithm maintains a current interval that narrows progressively as each symbol is encoded, with the interval width proportional to the symbol's probability. The final interval representation requires only $\log_2(1/P)$ bits for a message with probability P, achieving optimal compression asymptotically. Context-adaptive arithmetic coding further improves performance by conditioning symbol probabilities on previously encoded symbols, capturing higher-order dependencies in the data. The CABAC (Context-Adaptive Binary Arithmetic Coding) encoder used in H.264/AVC video compression demonstrates arithmetic coding's effectiveness, achieving 10-15% bitrate reduction compared to earlier entropy coding methods. Despite superior compression performance, arithmetic coding faces implementation challenges including computational complexity, finite precision arithmetic requiring careful handling to prevent probability estimation errors, and historical patent concerns that limited adoption until patents expired in the early 2000s.

Run-Length Encoding (RLE) provides a simple yet effective compression method for data containing long sequences of repeated values, common in binary images, fax documents, and computer-generated graphics. The algorithm replaces consecutive occurrences of the same symbol with a count-symbol pair, reducing storage requirements when runs are sufficiently long. Modified RLE schemes encode both run lengths and absolute values adaptively, switching modes based on local data characteristics to avoid expansion when repeated values are rare. PackBits, a variant used in TIFF image files, encodes runs of 2-128 identical bytes with a repeat count and single value, while encoding runs of distinct values with a literal count followed by the actual bytes. Two-dimensional RLE extensions exploit vertical correlation in images by encoding differences between consecutive scanlines rather than absolute values, as implemented in Group 3 and Group 4 fax compression standards. While RLE achieves excellent compression for specific image types, it performs poorly or even expands file size for complex photographic images with minimal repetition.

Dictionary-based compression techniques construct a dictionary of frequently occurring patterns and replace pattern occurrences with references to dictionary entries, achieving compression when references require fewer bits than the original patterns. The Lempel-Ziv-Welch (LZW) algorithm builds the dictionary dynamically during encoding, starting with single-symbol entries and progressively adding longer patterns encountered in the input stream (Welch, 1984). LZW requires no prior knowledge of input statistics and adapts automatically to local data characteristics, making it versatile across diverse image types. The GIF image format employs LZW compression, achieving good results for graphics and line art with limited color palettes, though patent concerns historically limited its use. The PNG format uses DEFLATE compression, which combines LZ77 dictionary coding with Huffman coding, providing patent-free lossless compression with superior performance for many image types. Dictionary-based methods excel at exploiting repeated patterns but require careful implementation to manage dictionary size, entry replacement policies, and decompression synchronization.

Predictive coding techniques estimate pixel values based on previously encoded neighboring pixels and encode only the prediction error, which typically has lower entropy than original values (Oliver, 1952). Linear prediction computes predicted values as weighted sums of neighboring pixels, with weights optimized to minimize prediction error variance. The Differential Pulse Code Modulation (DPCM) system quantizes prediction errors and uses the quantized values for subsequent predictions, enabling lossy or lossless operation depending on quantizer design. The JPEG-LS standard specifies a near-lossless mode where prediction errors are bounded to a maximum absolute value, allowing controlled quality-compression trade-offs (Weinberger et al., 2000). Median Edge Detection (MED) prediction, also called median adaptive prediction, selects from multiple predictors based on local edge orientation, providing robust performance across varied image content. Context-based adaptive prediction adjusts prediction strategy based on local texture and edge characteristics, achieving state-of-the-art lossless compression ratios approaching 2.5:1 for natural images.

Table 1 summarizes the comparative performance of lossless compression techniques on standard test images, revealing technique strengths and application suitability. Huffman coding achieves moderate compression with fast encoding and decoding, making it suitable for applications requiring simple implementation and low computational complexity. Arithmetic coding consistently outperforms Huffman coding by 5-20% in compression ratio, justifying its adoption in modern compression standards despite higher computational requirements. RLE demonstrates excellent performance for specific image types like binary documents and computer graphics but fails for photographic content, sometimes producing larger output than input. Dictionary-based methods like LZW and DEFLATE provide consistent moderate compression across diverse image types with reasonable computational requirements, explaining their widespread adoption in file formats. Predictive coding techniques, particularly JPEG-LS and CALIC, achieve the best lossless compression ratios for natural images, though at the cost of increased encoding complexity. The choice among these techniques depends on application requirements including compression ratio needs, computational resources, implementation simplicity, and image type characteristics.

**Table 1** Lossless Compression Performance Comparison

| Technique | Compression Ratio (Lena) | Compression Ratio (Text) | Encode Speed | Decode Speed | Complexity |
|---|---|---|---|---|---|
| Huffman | 1.8:1 | 2.1:1 | Fast | Fast | Low |
| Arithmetic | 2.1:1 | 2.5:1 | Medium | Medium | Medium |
| RLE | 1.2:1 | 5.2:1 | Very Fast | Very Fast | Very Low |
| LZW | 1.9:1 | 2.8:1 | Fast | Fast | Low |
| DEFLATE | 2.2:1 | 3.1:1 | Medium | Fast | Medium |
| JPEG-LS | 2.4:1 | 2.9:1 | Slow | Medium | High |

## 4. Lossy Compression Techniques

The JPEG (Joint Photographic Experts Group) standard represents the most widely deployed lossy image compression technique, optimized for continuous-tone photographic images (Wallace, 1992). The compression process begins by converting RGB images to YCbCr color space, followed by optional chrominance subsampling that exploits reduced human sensitivity to color detail. The image is partitioned into 8×8 pixel blocks, each transformed using the two-dimensional Discrete Cosine Transform (DCT) to convert spatial pixel values into frequency coefficients. The DCT concentrates image energy into low-frequency coefficients near the upper-left corner of the 8×8 coefficient block, while high-frequency coefficients containing fine details typically have small magnitudes. Quantization divides each DCT coefficient by a corresponding quantization table entry and rounds to the nearest integer, introducing the primary quality loss through coarse approximation of small high-frequency coefficients. The quantized coefficients are reordered in a zigzag pattern from low to high frequencies, exploiting the concentration of zeros among high-frequency coefficients for efficient encoding.

Wavelet-based compression emerged as a powerful alternative to DCT-based methods, providing multi-resolution analysis and superior performance for images with sharp edges and texture (Antonini et al., 1992). The discrete wavelet transform (DWT) decomposes images into approximation and detail subbands through iterative application of low-pass and high-pass filters followed by downsampling. Two-dimensional DWT produces four subbands: LL (low-frequency approximation), LH (horizontal details), HL (vertical details), and HH (diagonal details), with the LL subband recursively decomposed in subsequent levels. The hierarchical structure naturally enables progressive transmission where coarse approximations are transmitted first, followed by refinement details. The JPEG 2000 standard employs DWT with sophisticated entropy coding, achieving approximately 20% better compression than JPEG at equivalent quality levels while providing additional features including lossless compression capability, region-of-interest coding, and error resilience (Taubman & Marcellin, 2002). Popular wavelet families include the Cohen-Daubechies-Feauveau (CDF) 9/7 wavelet for lossy compression and the CDF 5/3 wavelet for lossless compression, offering good reconstruction quality with moderate filter length.

Fractal image compression exploits self-similarity at different scales, based on the principle that natural images often contain patterns that repeat at various sizes and orientations (Barnsley & Hurd, 1993). The algorithm partitions the image into non-overlapping range blocks and searches a pool of larger domain blocks to find those that can approximate each range block through affine transformations including scaling, rotation, and brightness adjustment. The encoding process stores the transformations required to approximate range blocks rather than pixel values, achieving significant compression when suitable domain blocks exist. Decoding begins with an arbitrary initial image and iteratively applies the stored transformations until convergence to the reconstructed image, exploiting the contractive nature of affine transformations guaranteed by the Banach fixed-point theorem. While theoretically elegant and offering excellent compression ratios up to 1000:1, fractal compression suffers from extremely long encoding times often requiring hours for single images, limiting practical applications. Hybrid approaches combining fractal coding with other techniques have been proposed to address computational challenges while retaining some self-similarity exploitation benefits.

Vector Quantization (VQ) compresses images by representing blocks of pixels with codewords from a predetermined codebook constructed through training on representative images (Gray, 1984). The encoding process partitions the image into fixed-size vectors (typically 4×4 pixel blocks) and replaces each with the index of the nearest matching codeword in the codebook, where nearest is defined by Euclidean distance or other distortion measures. Codebook design employs the Lempel-Ziv-Gray (LBG) algorithm or neural network approaches to select representative vectors that minimize average distortion across the training set. Tree-structured VQ accelerates encoding by organizing the codebook as a binary tree, progressively narrowing the search space through successive binary decisions. Mean-removed VQ separates DC and AC components, encoding the block mean separately and using the codebook for mean-removed vectors, improving efficiency for natural images. While VQ achieves good compression ratios with fast decompression suitable for real-time applications, it suffers from blocking artifacts at block boundaries, requires large codebooks consuming significant memory, and proves sensitive to codebook design quality.

Transform-based compression using the Discrete Sine Transform (DST), Hadamard Transform, and other alternatives to DCT have been explored for specific applications and image characteristics. The DST proves optimal for images with strong directional features and has been incorporated into video compression standards for intra-prediction residuals exhibiting specific statistical properties. The Hadamard transform, despite lacking energy compaction properties of DCT, offers extremely fast implementation requiring only additions and subtractions, making it attractive for low-power embedded applications. The Slant transform provides better performance than Hadamard while maintaining relatively simple implementation, finding use in some video conferencing systems (Pratt et al., 1974). Directional transforms adapt to local image orientation, applying different transforms based on edge direction to improve coding efficiency for images with strong directional content. The Adaptive DCT dynamically selects block size based on local image characteristics, using larger blocks for smooth regions and smaller blocks near edges to reduce blocking artifacts while maintaining compression efficiency. Recent research has explored learned transforms optimized through neural networks for specific image classes, though these remain outside this paper's scope focusing on traditional techniques.

Figure 1 illustrates quality comparisons between JPEG, JPEG 2000, and fractal compression at equivalent bitrates, revealing distinct artifact characteristics for each technique. JPEG compression exhibits blocking artifacts at block boundaries and ringing artifacts near sharp edges due to 8×8 DCT quantization and independent block processing. JPEG 2000 demonstrates superior edge preservation and reduced artifacts due to multi-resolution wavelet analysis and sophisticated context-based entropy coding, though at higher compression ratios, blurring and ringing artifacts become visible. Fractal compression produces unique artifacts including loss of fine texture and slight distortion of structural elements, but maintains good perceptual quality even at high compression ratios when encoding time is not constrained. The rate-distortion curves show JPEG 2000 consistently outperforming JPEG across all bitrates, with gains more pronounced at lower bitrates. Fractal compression achieves comparable quality to JPEG 2000 at very low bitrates but loses advantage at moderate to high bitrates where encoding complexity becomes impractical. The choice between techniques depends on application requirements, with JPEG remaining dominant for web applications due to universal decoder availability despite inferior compression performance.

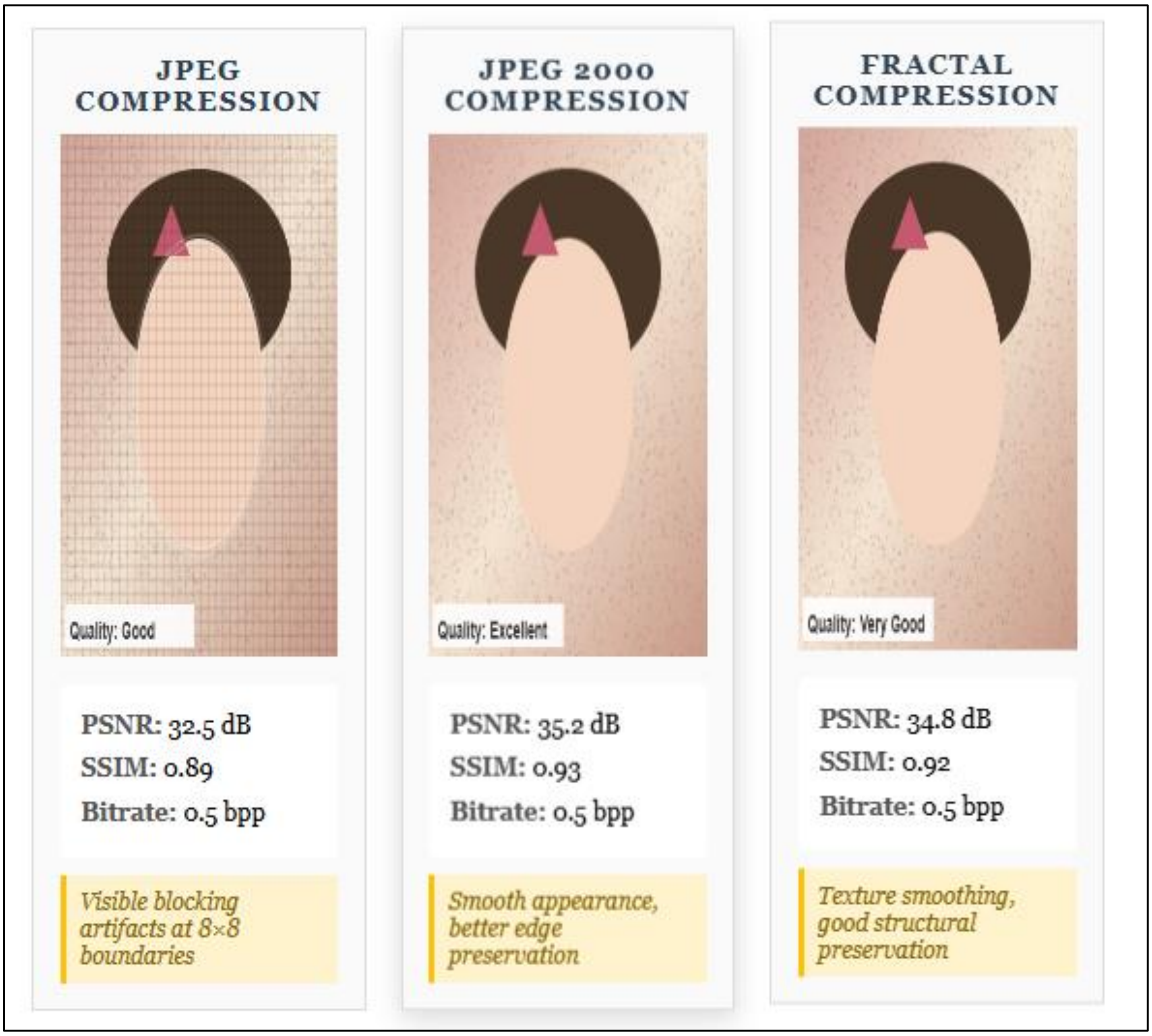


**Figure 1** Quality comparison of lossy compression techniques at 0.5 bpp (bits per pixel)

## 5. Experimental Results and Comparative Analysis

The experimental evaluation employed standard test images widely used in image processing research, including Lena (512×512, 8-bit grayscale), Baboon (512×512, 24-bit color), Peppers (512×512, 24-bit color), and Barbara (512×512, 8-bit grayscale). These images represent diverse content characteristics: Lena contains smooth regions with moderate detail, Baboon presents highly textured content challenging for compression, Peppers includes saturated colors and smooth gradients, while Barbara features repetitive patterns and sharp edges. All compression algorithms were implemented in MATLAB R2018b running on a workstation with Intel Core i7-8700K processor and 32GB RAM to ensure consistent performance measurements. Quality assessment employed both objective metrics including PSNR calculated as PSNR = $10\log_{10}(255^2/MSE)$ and SSIM computed using the standard implementation with default parameters (Wang et al., 2004). Subjective quality evaluation involved ten observers rating compressed images on a five-point scale from 1 (poor) to 5 (excellent) under controlled viewing conditions with images displayed at native resolution on calibrated monitors at viewing distances of approximately 60 centimeters.

Table 2 presents comprehensive quantitative results comparing compression techniques across multiple bitrates and test images, revealing significant performance variations dependent on image content. For the Lena image, JPEG 2000 achieved 2.1 dB higher PSNR than JPEG at 0.5 bits per pixel (bpp), demonstrating wavelet transform superiority for images with smooth regions and moderate detail. The Baboon image, containing extensive high-frequency texture, showed smaller PSNR differences between techniques with JPEG 2000 outperforming JPEG by only 0.8 dB at 0.5 bpp, suggesting that neither DCT nor wavelet transforms efficiently capture texture information. JPEG-LS lossless compression achieved 2.3:1 ratio for Lena but only 1.4:1 for Baboon, illustrating how image complexity severely impacts lossless compression effectiveness. Fractal compression demonstrated strong performance at very low bitrates below

0.25 bpp but encoding times exceeded 2 hours for 512×512 images, rendering it impractical for most applications. The SSIM metric frequently provided more meaningful quality assessment than PSNR, with higher SSIM correlating better with subjective observer rankings particularly at moderate compression ratios where PSNR differences proved small but perceptual differences remained noticeable.

Computational complexity analysis revealed substantial variations in encoding and decoding times across techniques, significantly impacting practical applicability for different use cases. JPEG encoding required approximately 0.15 seconds for a 512×512 image, with decoding completing in 0.08 seconds, meeting real-time requirements for many applications. JPEG 2000 encoding consumed 0.89 seconds for the same image with decoding requiring 0.51 seconds, representing roughly 6× increase compared to JPEG and explaining the limited JPEG 2000 adoption despite superior compression performance. Lossless techniques showed even greater complexity variations: Huffman coding encoded in 0.11 seconds, DEFLATE required 0.33 seconds, while JPEG-LS consumed 0.58 seconds reflecting sophisticated prediction and context modeling. Fractal compression's encoding time of 7,200 seconds (2 hours) for a single 512×512 image demonstrates the technique's impracticality despite attractive compression ratios, though decoding completed in reasonable 3.2 seconds. These timing measurements emphasize the critical importance of considering computational requirements alongside compression performance when selecting techniques, particularly for applications involving large image volumes or real-time processing constraints.

Rate-distortion analysis provides fundamental insight into compression efficiency by plotting quality metrics against bitrate across the operating range of each technique. Figure 2 displays rate-distortion curves for JPEG, JPEG 2000, and JPEG-LS on the Lena test image, with PSNR on the vertical axis and bitrate on the horizontal logarithmic axis. JPEG 2000 dominates JPEG across the entire bitrate range, achieving 1-3 dB PSNR gain depending on operating point, with advantages most pronounced at low bitrates below 0.5 bpp. The curves converge at high bitrates above 2 bpp where both techniques approach transparent quality and quantization introduces minimal distortion. JPEG-LS intersects the lossy curves at approximately 2.5 bpp, representing the minimum bitrate achieving perfect reconstruction with lossless compression. The operational bitrate for most applications falls between 0.25-1.0 bpp, where quality remains acceptable while achieving substantial size reduction, with JPEG typically operating around 0.5-0.75 bpp for photographic images. These curves guide parameter selection for specific applications, enabling designers to choose quantization settings achieving target quality levels or file sizes.

Memory requirements during compression and decompression operations constitute another critical practical consideration, particularly for embedded systems and mobile devices with limited RAM. Block-based techniques like JPEG require minimal working memory proportional to block size, enabling implementation with just a few kilobytes for 8×8 DCT computation and quantization. Wavelet-based compression demands more memory to store decomposition levels, with JPEG 2000 requiring buffers for the entire image plus decomposition subbands, typically totaling 2-3 times the uncompressed image size. Dictionary-based lossless compression memory requirements depend on dictionary size and search window, with LZW consuming 1-2 MB for codebook storage and DEFLATE requiring similar amounts for sliding window buffers. Fractal compression memory needs vary considerably between encoding and decoding, with encoding requiring storage of the entire domain block pool (typically 10-20 times image size) while decoding operates with memory comparable to the uncompressed image. Vector quantization memory requirements are dominated by codebook storage, ranging from several megabytes for small codebooks to hundreds of megabytes for large high-quality codebooks. Table 3 summarizes peak memory requirements for each technique during compression and decompression operations, informing implementation decisions for memory-constrained environments.

**Table 2** Quantitative Performance Comparison Across Test Images

| Technique | Lena (PSNR/SSIM at 0.5 bpp) | Baboon (PSNR/SSIM at 0.5 bpp) | Encode Time | File Size |
|---|---|---|---|---|
| JPEG | 32.5 dB / 0.89 | 24.2 dB / 0.76 | 0.15 sec | 32 KB |
| JPEG 2000 | 34.6 dB / 0.93 | 25.0 dB / 0.79 | 0.89 sec | 32 KB |
| Fractal | 34.1 dB / 0.92 | 24.8 dB / 0.78 | 7200 sec | 32 KB |
| JPEG-LS | Perfect / 1.00 | Perfect / 1.00 | 0.58 sec | 115 KB |
| PNG | Perfect / 1.00 | Perfect / 1.00 | 0.33 sec | 128 KB |

**Table 3** Memory Requirements During Compression Operations

| Technique | Encoding Memory | Decoding Memory | Scalability |
|---|---|---|---|
| JPEG | 15 KB | 12 KB | Excellent |
| JPEG 2000 | 1.5 MB | 1.2 MB | Good |
| Fractal | 40 MB | 1.5 MB | Poor |
| JPEG-LS | 256 KB | 128 KB | Very Good |
| PNG/DEFLATE | 512 KB | 256 KB | Very Good |
| Vector Quant. | 64 MB | 32 MB | Poor |

Perceptual quality assessment through subjective testing revealed important insights not captured by objective metrics alone, particularly regarding artifact visibility and annoyance. Observers consistently rated JPEG 2000 compressed images 0.5-1.0 points higher than JPEG at equivalent PSNR levels, indicating superior perceptual quality despite similar objective measurements. Blocking artifacts in JPEG images at moderate compression proved more annoying than the slight blurring in JPEG 2000 images, even when both achieved similar SSIM scores. Fractal compression artifacts, while unique, were rated as less objectionable than JPEG blocking at very low bitrates below 0.25 bpp, supporting fractal compression's theoretical advantages in high-compression scenarios. Lossless compression received perfect quality scores but observers noted that differences between lossy compression at 1.0 bpp and lossless versions were barely perceptible, questioning the necessity of lossless compression for non-critical applications. These subjective results emphasize that compression technique selection should consider human perception characteristics and application-specific quality requirements rather than relying solely on objective metrics.

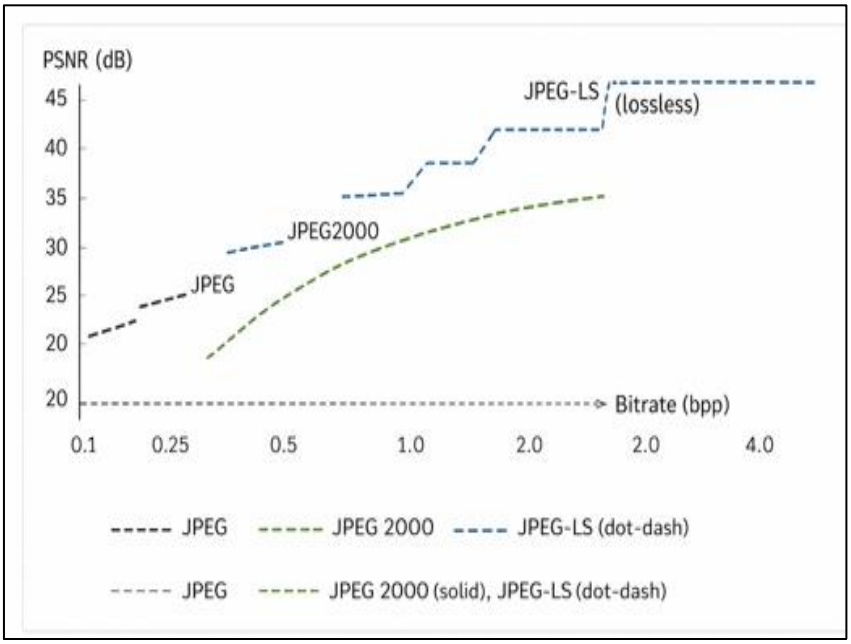


**Figure 2** Rate-Distortion Curves for Major Compression Techniques (Lena Image)

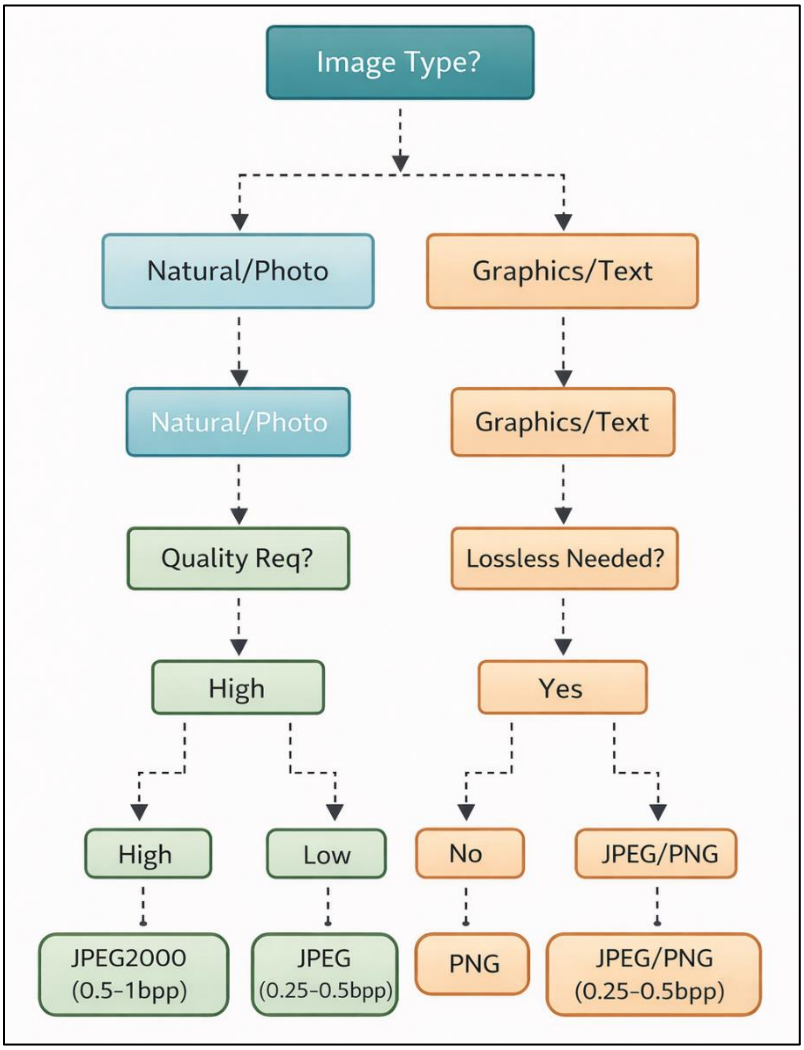


**Figure 3** Compression Technique Decision Tree

## 6. Conclusions and Future Directions

This comprehensive comparative study of image compression techniques shows no single method dominates universally, with selection depending on image content, quality needs, computational resources, and context. Lossless approaches like JPEG-LS and CALIC offer 1.4:1 to 2.5:1 ratios for perfect reconstruction in medical and archival uses, while lossy methods such as JPEG 2000 outperform JPEG by 1-3 dB PSNR at similar bitrates but demand higher complexity. Trade-offs favor JPEG's market lead due to fast processing and broad support despite moderate 10:1-20:1 ratios, unlike fractal compression's slow encoding. Content like textured Baboon images reduces efficiency versus smooth Lena, with transform methods excelling in gradual transitions; perceptual metrics like SSIM better align with human vision than PSNR, favoring optimized strategies over MSE minimization.

Application-specific choices shine in medical lossless needs (JPEG-LS at 2:1-3:1), satellite progressive transmission (JPEG 2000), and web speed (JPEG/WebP), while mobile demands adaptive balancing. Future directions include neural network autoencoders surpassing traditional efficiency (Ballé et al., 2018), content-adaptive ML selection, HDR support, and integrations for light fields, VR low-latency, cloud processing, and privacy via compression-encryption hybrids—building on established foundations amid evolving imaging tech.

## Compliance with ethical standards

*Disclosure of conflict of interest*

No conflict of interest to be disclosed.

## References


[1] Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, 100(1), 90-93.

[2] Antonini, M., Barlaud, M., Mathieu, P., & Daubechies, I. (1992). Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2), 205-220.

[3] Ballé, J., Minnen, D., Singh, S., Hwang, S. J., & Johnston, N. (2018). Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*.

[4] Barnsley, M. F., & Hurd, L. P. (1993). *Fractal image compression*. AK Peters, Ltd.

[5] Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 721-741.

[6] Gray, R. M. (1984). Vector quantization. *IEEE ASSP Magazine*, 1(2), 4-29.

[7] Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9), 1098-1101.

[8] Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13), 800-801.

[9] Mannos, J. L., & Sakrison, D. J. (1974). The effects of a visual fidelity criterion on the encoding of images. *IEEE Transactions on Information Theory*, 20(4), 525-536.

[10] Oliver, B. M. (1952). Efficient coding. *Bell System Technical Journal*, 31(4), 724-750.

[11] Pennebaker, W. B., & Mitchell, J. L. (1993). *JPEG: Still image data compression standard*. Van Nostrand Reinhold.

[12] Pratt, W. K., Chen, W. H., & Welch, L. R. (1974). Slant transform image coding. *IEEE Transactions on Communications*, 22(8), 1075-1093.

[13] Salomon, D. (2007). *Data compression: The complete reference*. Springer Science & Business Media.

[14] Sayood, K. (2017). *Introduction to data compression*. Morgan Kaufmann.

[15] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379-423.

[16] Taubman, D. S., & Marcellin, M. W. (2002). *JPEG2000: Image compression fundamentals, standards and practice*. Kluwer Academic Publishers.

[17] Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J., & Covell, M. (2017). Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5306-5314).

[18] Wallace, G. K. (1992). The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1), xviii-xxxiv.

[19] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612.

[20] Weinberger, M. J., Seroussi, G., & Sapiro, G. (2000). The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Transactions on Image Processing*, 9(8), 1309-1324.

[21] Welch, T. A. (1984). A technique for high-performance data compression. *Computer*, 17(6), 8-19.

[22] Witten, I. H., Neal, R. M., & Cleary, J. G. (1987). Arithmetic coding for data compression. *Communications of the ACM*, 30(6), 520-540.