Quantitative Analysis of Diverse Image Compression Algorithms Across Multiple Image Datasets: A Comprehensive Examination Using Quality Metrics

Aakashneel Basak[1], Siliveri Sriharshini[2], Usha Sundari Ryali HSV[1], and Murali Krishnan S[1]

[1] Microwave Data Processing Group, Data Processing Area,
National Remote Sensing Centre, Indian Space Research Organisation,
Hyderabad, India
aakashneel_basak@nrsc.gov.in
[2] Department of Electronics & Communication Engineering, National Institute of Technology, Andhra Pradesh, India

Abstract. Image compression plays a pivotal role in various fields where efficient storage and transmission of visual data are essential. This paper presents a comparative analysis of both lossless and lossy image compression techniques, emphasizing their significance and applicability in modern digital environments. The study evaluates prominent lossless techniques such as Run-Length Encoding (RLE), LZ77, Huffman coding, among others, highlighting their strengths, weaknesses, and performance metrics. Additionally, it delves into lossy compression methods, focusing particularly on the Discrete Cosine Transform (DCT) algorithm. Through comprehensive experimentation and evaluation, this research aims to provide insights into the trade-offs between compression efficiency and image quality, enabling informed decision-making for selecting the most suitable compression technique based on specific requirements and constraints. Furthermore, the comparative analysis sheds light on the advancements, challenges, and future prospects in the realm of image compression technology.

Keywords: Image Compression · Encoding · PSNR.

1 Introduction

Image compression is a critical component of modern digital systems, facilitating efficient storage, transmission, and manipulation of visual data. In this context, a myriad of compression techniques exists, ranging from lossless methods, such as Run-Length Encoding (RLE), LZ77, and Huffman coding, to lossy approaches like the Discrete Cosine Transform (DCT) algorithm. Understanding the strengths, weaknesses, and performance characteristics of these techniques is paramount for optimizing resource utilization while maintaining acceptable image quality.

This paper undertakes a comparative analysis of various lossless and lossy image compression techniques, aiming to elucidate their efficacy and suitability for different applications. This study seeks to provide insights into the trade-offs inherent in selecting an appropriate compression method by evaluating compression efficiency, computational complexity, and image fidelity. Furthermore, it explores the evolving landscape of image compression technology, considering advancements, challenges, and future prospects in the field.

Through this comparative analysis, stakeholders in digital imaging, ranging from multimedia developers to data scientists, can make informed decisions regarding the selection and implementation of image compression techniques that best align with their requirements and constraints. The subsequent sections delve into the specific methodologies, evaluations, and findings of this comparative study, offering a comprehensive understanding of the state-of-the-art in image compression.

2 Different Image Formats

Image formats are standardized methods for encoding and storing digital images, defining how data is structured within image files. The variety of formats serves several purposes: facilitating compression to reduce file size while preserving quality, ensuring compatibility across different devices and platforms, offering specialized features like transparency and animation, accommodating various color depths and quality levels, and optimizing efficiency for specific tasks such as web performance or archival storage.

2.1 Lossless Image Formats

Lossless image formats, such as TIFF, PNG, and BMP, are vital for preserving the highest possible image fidelity without any loss of quality during compression. They are indispensable in professional photography, printing, and archival purposes, where maintaining image details is paramount. Lossless compression ensures that all original image data is retained, making these formats suitable for applications requiring maximum image quality, such as high-resolution prints and digital artwork. Although lossless formats result in larger file sizes compared to lossy compression, they guarantee pixel-perfect accuracy and are preferred in situations where image integrity is critical, such as medical imaging and scientific analysis.

RAW Image format: RAW files offer unparalleled image quality and editing flexibility, capturing all sensor data for maximum detail and dynamic range. Despite larger file sizes and slower workflow, they allow non-destructive editing and revisiting the original data. However, RAW files require compatible software due to proprietary formats, and shooting in RAW may have a learning curve for beginners. Primarily favored by professionals and serious hobbyists, RAW is ideal for detailed or challenging scenes, offering creative freedom and control over extensive editing. Its applications include professional photography, capturing high dynamic range scenes, and situations requiring significant post-processing adjustments.

Portable Network Graphics (PNG): The PNG file format offers versatile support for 8-bit, 24-bit, and 48-bit true color with or without an alpha channel, prioritizing lossless compression over JPEG's lossy approach. PNG typically achieves 10 to 30 percent more compression than GIF, boasting smaller file sizes and a broader color range. Its advantages include lossless compression, crucial for sharp graphics and text, and transparency through alpha channels, ideal for logos and web overlays. With support for up to 16 million colors and an open standard, PNG is widely compatible with image editing software. However, PNG files tend to be larger than JPEGs, impacting website loading times, and lack animation support like GIF.

Tagged Image File Format (TIFF): TIFF files, renowned for their high-quality image storage, offer lossless compression, enabling preservation of all image data with high bit depths for detailed color representation. Commonly used in professional printing, archival preservation, and scientific imaging, TIFF files excel in scenarios prioritizing image quality and long-term stability. While they may lack the editing flexibility of RAW files, their complexity and storage requirements are balanced by their suitability for high-resolution prints, digital archives, and professional photography.

2.2 Lossy Image Formats

Lossy image formats, such as JPEG and WebP, are crucial for balancing image quality with storage and bandwidth efficiency. They achieve higher compression rates by discarding some image data, resulting in smaller file sizes. This is essential for optimizing web performance, reducing loading times, and conserving storage space, especially in digital photography, web graphics, and online sharing. Lossy compression allows for faster upload and download times, enhancing user experience and improving website SEO rankings.

Joint Photography Experts Group (JPEG): JPEG files, commonly used for everyday image storage and sharing, support up to 24-bit color but employ lossy compression, sacrificing some image quality. Despite potential quality loss, JPEGs are widely recognized and compatible with various browsers, software, and apps, with small file sizes enabling quick transfer and online viewing. Post-processing is simplified with preset white balance and saturation settings. However, heavy compression may degrade image sharpness and introduce artifacts like posterization and aliasing.

High Efficiency Image Format (HEIF): HEIF is a modern image file format developed by MPEG, offering superior compression while maintaining high image quality. It uses HEVC compression, resulting in smaller file sizes compared to JPEG while preserving or enhancing image detail and color accuracy, especially in HDR imaging. HEIF supports multiple images, transparency, and rich metadata, but its adoption is hindered by compatibility issues and higher processing requirements. Despite these drawbacks, HEIF finds applications in consumer electronics like smartphones, digital cameras, and online services, as well as in professional photography and imaging. With growing support from software applications and operating systems.

Web Picture (WEBP): WebP, developed by Google, is a modern image format offering superior compression for web images without compromising quality. Supporting both lossy and lossless compression, WebP boasts significantly smaller file sizes compared to JPEG and PNG, enhancing web performance and reducing bandwidth usage. Despite its advantages, compatibility issues with older browsers and image editing software exist, requiring conversion overhead for existing image libraries. However, its versatility makes it ideal for web development, e-commerce, social media, digital advertising, and content management systems.

3 Different Compression techniques

Image compression optimizes image files for storage and transmission, offering advantages like reduced file sizes, improved performance, and cost savings. However, it comes with drawbacks such as potential loss of quality, compatibility issues, processing overhead, and limited editing capabilities.

3.1 Lossless compression techniques

Lossless compression reduces image file size without sacrificing any image data, making it suitable for applications requiring preservation of original quality. Commonly used in professional photography, graphic design, and archiving, it maintains maximum image fidelity. However, it results in larger file sizes and slower upload/download times compared to lossy compression, making it less efficient for web use.

Run length encoding Algorithm Run-Length Encoding (RLE) efficiently compresses data by replacing consecutive repeated symbols with a count of the repetition. It's effective for data with long sequences of repeated symbols but may not achieve significant compression for random or irregular data. RLE is simple, easy to implement, and suitable for real-time applications, but its effectiveness depends on the nature of the input data. Despite its limitations, RLE remains valuable in various domains such as image and video compression, text and document compression, and data transmission over limited bandwidth channels. Its straightforward encoding and decoding processes make it a practical choice for scenarios where simplicity and efficiency are paramount. However, for datasets lacking repetitive patterns, other compression algorithms may be more suitable.

Huffman Coding Algorithm Huffman coding is a widely used compression algorithm that assigns variable-length codes to input symbols based on their frequencies. It achieves compression by representing frequently occurring symbols with shorter codes and less frequent symbols with longer codes, thus reducing the average length of the encoded message. Huffman coding is particularly effective for data with skewed frequency distributions, where certain symbols occur much more frequently than others. Its efficiency lies in its ability to generate an optimal prefix code, minimizing the total number of bits required to represent the input data. However, constructing the Huffman tree and encoding the data require additional computational overhead. Despite this, Huffman coding is widely employed in various applications, including data compression in file formats like ZIP and JPEG, as well as in network protocols and data transmission schemes. Its versatility, combined with its ability to achieve significant compression ratios, makes it a fundamental component of many compression systems.

Lempel Ziv 77 It's a compression algorithm developed by Abraham Lempel and Jacob Ziv in 1977. LZ77, a lossless data compression algorithm, is widely utilized in various applications due to its simplicity and effectiveness. Operating by replacing repeated occurrences of data with references to previous occurrences, LZ77 efficiently reduces redundancy in the input data. This algorithm employs a sliding window approach, where a buffer of previously encountered data is maintained and searched for matches with the current data. By representing repeated data with shorter references, LZ77 achieves compression while preserving the original data. However, LZ77 may not be as effective for highly random or irregular data patterns, where repeated sequences are scarce. Despite this limitation, LZ77 is extensively used in file compression formats like DEFLATE (used in ZIP files) and protocols like HTTP, where efficient data transfer is crucial. Its simplicity and versatility make it a valuable tool in various compression applications, offering a balance between compression efficiency and computational complexity.

Lempel-ziv-Welch "LZW," standing for "Lempel-Ziv-Welch," is a widely-used lossless data compression algorithm renowned for its efficiency and simplicity. Its approach involves replacing repetitive sequences within the data with shorter codes, dynamically building a dictionary of codes during the compression process. This method not only achieves good compression ratios, particularly with data containing repetitive patterns, but also proves relatively straightforward to implement and efficient in terms of memory usage. LZW finds extensive applications in various file formats such as GIF, TIFF, and PDF, as well as in compression utilities like Unix compress (".Z") and the Deflate algorithm used in ZIP files. However, its performance may suffer with highly random or pre-compressed data, and its efficiency can degrade with larger dictionaries or longer input sequences. Nevertheless, LZW strikes a commendable balance between compression effectiveness and computational simplicity, making it a preferred choice for scenarios where lossless compression is necessary.

3.2 Lossy compression techniques

Lossy compression reduces image file size by permanently discarding some image data, making it suitable for scenarios where size reduction is prioritized over preserving all original details. It finds applications in web and social media uploads, digital photography, and streaming platforms. Advantages include smaller file sizes, faster upload/download times, and wide compatibility with devices and platforms. However, lossy compression can lead to quality loss, visible compression artifacts, and irreversible data loss. Editing capabilities are also limited due to the loss of original data, making it important to strike a balance between compression level and desired image quality.

Discrete Cosine Transform The Discrete Cosine Transform (DCT) is a widely used technique in image and signal compression, notably employed in popular formats like JPEG and MP3. It transforms spatial-domain data into frequency-domain coefficients, facilitating efficient compression by concentrating most of the signal energy into a smaller number of coefficients. By retaining the most significant coefficients and discarding the least significant ones, DCT achieves compression with minimal loss of perceptual quality. However, higher compression ratios may lead to visible artifacts, especially in areas with fine details or high-frequency components. Despite this, DCT remains a fundamental tool in multimedia compression due to its effectiveness in balancing compression efficiency and perceived quality. It finds extensive application in digital media compression, including images, audio, and video, enabling efficient storage and transmission of multimedia content while maintaining acceptable perceptual quality.

4 Image quality metrics

In image compression, image quality metrics are objective measures used to assess the fidelity of compressed images compared to their original counterparts. These metrics gauge factors such as loss of detail, introduced artifacts, and color accuracy, enabling quantification of the trade-off between compression ratio and image quality.

4.1 Compression Ratio

Compression ratio refers to the degree of data reduction achieved through compression, quantifying the ratio between the original uncompressed data size and the size of the compressed data. A higher compression ratio implies more efficient compression, resulting in smaller file sizes and reduced storage requirements, facilitating faster data transmission and lower bandwidth usage. It enables efficient storage and transfer of data, particularly in digital media, communication networks, and data archiving. However, achieving higher compression ratios often involves trade-offs, as more aggressive compression can lead to loss of data and reduced quality, particularly in lossy compression methods.

4.2 Peak Signal to noise Ratio

Peak Signal-to-Noise Ratio (PSNR) is a widely used metric in image compression to quantify the quality of compressed images compared to the original uncompressed ones. It measures the ratio of the maximum possible power of a signal to the power of the noise that affects the fidelity of its representation. Higher PSNR values indicate better image quality and less distortion after compression. However, PSNR has limitations, such as being sensitive to perceptual differences not captured by signal power alone and its inability to account for human perception accurately. PSNR values typically range from 0 dB (indicating no similarity) to infinity dB (perfect similarity), with higher values indicating better image quality in image compression scenarios.

4.3 Structural Similarity Index Measure

Structural Similarity Index Measure(SSIM) is a metric used in image processing to measure the similarity between two images. It evaluates the structural information, luminance, and contrast similarity, providing a more comprehensive assessment of image quality compared to traditional metrics like PSNR. It is also robust to common distortions introduced by compression. However, SSIM requires more computational resources compared to simpler metrics like PSNR. SSIM values range from -1 to 1, where 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates perfect dissimilarity. In image compression, higher SSIM values suggest better preservation of image details and perceptual quality.

4.4 Encoding time

Encoding time refers to the duration required to compress data, such as images, into a compressed format using a specific compression algorithm. Lower encoding times are desirable as they result in faster compression, facilitating real-time or near-real-time processing. Pros of shorter encoding times include improved efficiency in applications where speed is critical, such as video streaming and live broadcasting. However, reducing encoding time may require sacrificing compression efficiency or quality, leading to larger file sizes or lower compression ratios. Encoding time can vary widely depending on factors like the complexity of the compression algorithm, the size and complexity of the input data, and the computational resources available.

4.5 Decoding time

Decoding time refers to the duration required to decompress compressed data back into its original format, such as images, using a specific decompression algorithm. Lower decoding times are favorable as they result in faster data retrieval and display. Pros of shorter decoding times include reduced latency in accessing and viewing compressed content, enhancing user experience, especially in real-time applications like video streaming and gaming. However, decreasing decoding time might involve compromising decompression efficiency or quality, potentially leading to slower processing or lower image fidelity. Decoding time can vary depending on factors like compression algorithm complexity, hardware capabilities, and the size and complexity of the compressed data. Comparatively, decoding time is often shorter than encoding time as decompression typically requires fewer computational resources.

5 Literature Review

Image compression is a key component in the field of digital image processing and is essential for reducing the amount of data required to represent an image. There are two main types of image compression: lossless and lossy compression. In lossless compression, the original data can be perfectly reconstructed from the compressed data, while in lossy compression, some data is lost during the compression process, resulting in a reduction in image quality.

Several studies have compared the performance of lossy and lossless image compression techniques. For example, Smith et al. conducted a comprehensive analysis of various lossy and lossless compression algorithms and their impact on image quality. Their study found that while lossy compression techniques can achieve higher compression ratios, they often result in a noticeable loss of image quality, particularly in areas with high levels of detail.

In contrast, Liu and Wang explored the effectiveness of lossless compression methods for medical images, emphasizing the importance of maintaining image integrity and diagnostic accuracy. Their research concluded that lossless compression techniques, such as predictive coding and run-length encoding, are well-suited for medical imaging applications where preserving every detail is crucial.

Overall, the literature indicates that the choice between lossy and lossless image compression depends on the specific requirements of the application. While lossy compression may offer higher compression ratios, it may not be suitable for applications that demand high-fidelity image reproduction, such as medical imaging or archival purposes. Conversely, lossless compression methods prioritize image integrity and are favored in scenarios where preserving the original image data is paramount.

6 Methodology

Here You can find about the generalised methodologies of different lossy and lossless compression techniques.

6.1 Lossy Image compression

The generalized methodology for lossy image compression:

1. Image Representation: Represent the image as a matrix of pixels. Each pixel can consist of intensity values (for grayscale images) or color values (for color images), typically in RGB or YCbCr color spaces.

2. Transform Coding: Apply transformations to the image data to exploit redundancies and remove irrelevant information. Common transformations include Discrete Cosine Transform (DCT), Wavelet Transform, or Fourier Transform.

3. Quantization: Reduce the precision of transformed coefficients by dividing them by quantization step sizes. Higher quantization leads to greater compression but also more loss of information.

4. Entropy Coding: Apply entropy coding techniques to further compress the quantized data. Common methods include Huffman coding, Arithmetic coding etc.

5. Compression Ratio Control: Adjust compression parameters to control the trade-off between compression ratio and image quality. This step often involves setting quantization parameters and choosing appropriate compression algorithms.

6. Decompression: Reverse the compression process to reconstruct the compressed image. Perform entropy decoding, dequantization, and inverse transformations to recover the image data.

7. Quality Assessment: Evaluate the quality of compressed images using objective metrics (such as PSNR, SSIM) and subjective evaluations (human perception). Fine-tune compression parameters based on quality assessment results.

This methodology outlines the general steps involved in lossy image compression. Specific compression techniques and algorithms may vary based on the requirements, constraints, and target applications.

6.2 Lossless compression

The Generalise approach for lossless compression

1. Image Representation: Represent the image as a matrix of pixels. Each pixel can consist of intensity values (for grayscale images) or color values (for color images), typically in RGB or YCbCr color spaces.

2. Predictive Coding: Utilize predictive coding techniques to exploit spatial or temporal redundancies in the image data. Predict pixel values based on neighboring pixels or previous pixels in the image sequence.

3. Dictionary Coding: Build dictionaries or codebooks to represent frequently occurring patterns or symbols in the image. Replace repeating patterns with shorter codewords or symbols.

4. Entropy Coding: Apply entropy coding techniques to further compress the encoded data. Common methods include Huffman coding, Arithmetic coding, etc.

5. Decompression: Reverse the compression process to reconstruct the original image. Perform entropy decoding, dictionary decoding, and reverse predictive coding to recover the image data.

6. Quality Assessment: Assess the fidelity of the decompressed image compared to the original image. Use objective metrics (such as PSNR) and subjective evaluations to measure the quality of lossless compression.

This methodology outlines the general steps involved in lossless image compression. Specific compression techniques and algorithms may vary based on the requirements, constraints, and target applications.

7 Results and Discussions

Table 1 gives a summary of all heading levels.
Displayed equations are centered and set on a separate line.

$$x + y = z \tag{1}$$

Table 1: Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	**Lecture Notes**	14 point, bold
1st-level heading	**1 Introduction**	12 point, bold
2nd-level heading	**2.1 Printing Area**	10 point, bold
3rd-level heading	**Run-in Heading in Bold.** Text follows	10 point, bold
4th-level heading	*Lowest Level Heading.* Text follows	10 point, italic

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 1).

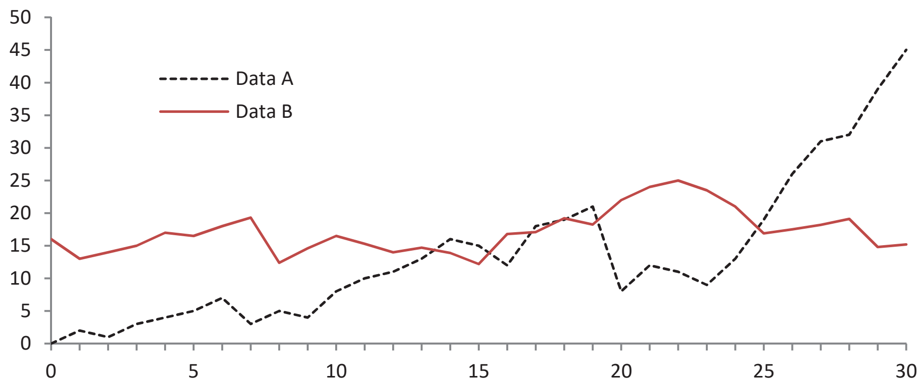

Fig. 1: A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], and a homepage [5]. Multiple citations are grouped [1–3], [1, 3–5].

Acknowledgments. A bold run-in heading in small font size at the end of the paper is used for general acknowledgments, for example: This study was funded by X (grant number Y).

Disclosure of Interests. It is now necessary to declare any competing interests or to specifically state that the authors have no competing interests. Please place the statement with a bold run-in heading in small font size beneath the (optional) acknowledgments[3], for example: The authors have no competing interests to declare that are relevant to the content of this article. Or: Author A has received research grants from Company W. Author B has received a speaker honorarium from Company X and owns stock in Company Y. Author C is a member of committee Z.

References

1. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, http://www.springer.com/lncs, last accessed 2023/10/25

[3] If EquinOCS, our proceedings submission system, is used, then the disclaimer can be provided directly in the system.