Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Improved intensity rounding and division near lossless image compression algorithm using delta encoding

Mahmoud Al Qerom[a*]

[a]*Department of Information Communication Technology, British University of Bahrain, Bahrain*

C H R O N I C L E

Article history:
Received: July 20, 2024
Received in revised format: August 20, 2024
Accepted: September 1, 2024
Available online: September 1, 2024

Keywords:
Near-Lossless Image Compression
RIFD-DLT
Delta Encoding
Rounding the Intensity Followed by Dividing (RIFD)

A B S T R A C T

This paper presents RIFD-DLT, an advanced near-lossless image compression algorithm that combines Delta Encoding with the original rounding the intensity followed by division (RIFD) method. The RIFD method first minimizes the image intensities, which makes the next compression stages more efficient. Subsequently, Delta Encoding subtracts neighboring rows in each of the image's three-color matrices, using the proximity of pixel values in adjacent rows to further reduce the image intensity. Extensive investigations show that RIFD-DLT outperforms the state-of-the-art algorithms and benchmarks with respect to compression ratio and processing time. More specifically, RIFD-DLT compresses data from 11520 KBs to 2131 KBs with an 81.93% reduction and a 43.7% improvement over original RIFD-Huffman when compressing Kodak Image set. When comparing the RIFD-DLT with LICA algorithm, the total file size is reduced by 71.2%, representing a 10.73% improvement for RIFD-DLT. Also, RIFD-DLT shows notable speed gains over RIFD-Huffman, requiring only 34.62 seconds to compress and decompress all images from three datasets (Waterloo, Kodak and EPFL), as compared to 50.99 seconds for RIFD-Huffman. As for the image quality, the proposed algorithm achieved an average PSNR values of 58.51 dB, 51.3 dB, and 52.22 dB for the EPFL, Kodak, and Waterloo image sets, respectively, demonstrate the excellent image quality that persists after decompression, with a minimal distortion that is imperceptible to the human visual system and identically to the RIFD-Huffman PSNR. These findings show that, while preserving excellent image quality, RIFD-DLT provides an incredibly efficient and fast method of image compression.

1. Introduction

The production and use of digital images has increased at an unprecedented rate due to the rapid growth of digital technology. Because of this rise, there is an urgent need for effective image compression algorithms since the massive volumes of data generated demand for efficient means of transmission and storage. Digital images that have been compressed into smaller files can be stored and shared more easily without sacrificing quality significantly. This is especially important in domains where maintaining high accuracy and precision is required, like scientific research, medical imaging, precision engineering, and remote sensing (Hu *et al.,* 2021; Furong *et al.,* 2022; Cao *et al.,* 2020).

As a result, research on image compression has become essential to meet the competing demands of image quality and compression ratio (Strümpler *et al.,* 2022).

* Corresponding author.
E-mail address m.a.qerom@bub.bh (M. A. Qerom)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2024.9.002

1. Background and Motivation

The necessity for effective information handling systems, particularly in fields requiring high fidelity and precision, has been highlighted by the exponential growth in data generation, which has also highlighted the significance of image compression (Hu *et al*., 2021; Bai *et al*., 2024). As we move through a time characterized by enormous and constantly expanding amounts of digital imagery, efficient image compression methods are becoming more and more important (Strümpler *et al*., 2022). This is especially important in areas including medicine, precision engineering, and remote sensing, where maintaining image quality is crucial (Lungisani *et al*., 2023; Dua *et al*., 2020).

Two types of traditional image compression techniques exist: lossy and lossless. For applications like scientific and medical imaging where the original data quality must be maintained, lossless image compression is essential to ensure that every detail can be properly recreated (Dua *et al*., 2020; Ungureanu *et al*., 2024). Nevertheless, in comparison to lossy approaches, this leads to poorer compression ratios. Conversely, lossy image compression significantly reduces file size by removing less important information, which makes it perfect for online and mobile applications with constrained bandwidth and storage. Higher compression ratios are obtained, but certain visual features are permanently lost. Consequently, the development of advanced image compression techniques is motivated by the need to balance the demands for high compression ratios and high image quality in these essential fields (Dumka *et al*., 2020; Alqerom, 2020; Cavallaro *et al*., 2022).

2. The Role of Near-Lossless Image Compression

Therefore, when it's essential to preserve image quality while attaining larger compression rates, near-lossless image compression becomes necessary. While some slight data loss may occur during near-lossless compression, the image quality is guaranteed to stay almost identical to the original, unlike lossy compression, which sacrifices more data to minimize file size. This is essential in applications where high fidelity is still needed but small changes are allowed. For example, in professional photography, an image's visual is preserved only if its integrity is maintained. Near-lossless techniques provide effective transmission and storage while guaranteeing high image quality (Otair *et al*., 2022; Rahman *et al*., 2021; Sharma *et al*., 2020; Liu *et al*., 2022).

3. Existing Techniques and Their Limitations

Several image compression methods currently in use, including Joint Photographic Experts Group (JPEG), Delta Encoding, and rounding the intensity followed by division (RIFD), they demonstrated varied degrees of effectiveness in finding a balance between compression ratio and image quality. Nevertheless, these approaches frequently fail to achieve the ideal balance. Although RIFD can reduce file sizes by simplifying an image intensities value, it may also introduce some distortion that degrades the quality of the image. While it can efficiently eliminate redundancy, delta encoding, which compresses data by recording the difference between successive data points may not always achieve the appropriate compression ratios without sacrificing some detail. Although JPEG can greatly reduce file sizes, it frequently causes artifacts that deteriorate image quality and leads in the loss of fine details (Zhang et al., 2020; Abrardo *et al*., 2011).

These limitations highlight the ongoing challenge in developing compression algorithms that can provide high compression ratios while preserving high reconstructed image quality.

4. Introduction to RIFD-DLT: A Novel Approach

This paper presents RIFD-DLT, a unique near-lossless image compression algorithm that combines RIFD with Delta Encoding, to overcome the limitations of existing techniques. To enable more effective compression, the RIFD-DLT method reduces image intensities first by using the RIFD technique. After that, the values of adjacent rows in each of the image's three-color matrices are subtracted in order to apply Delta Encoding to take advantage of the proximity of pixel values in adjacent rows. This hybrid technique enhances the speed of the compression and decompression processes while maintaining a higher compression ratio with acceptable distortion that can't be detected by the human visual system.

5. Contributions and Outline of the Paper

The main contributions of this paper are as follows:

- Development of RIFD-DLT: a new image compression algorithm that combines the best features of Delta Encoding and RIFD is provided to achieve high compression ratios with nearly lossless image quality.
- Performance Evaluation: By means of comprehensive evaluations, the work demonstrates that RIFD-DLT performs better in terms of processing time and compression ratio than other algorithms, including RIFD, LICA-CS, and Huffman.
- Analysis of Image Quality: An in-depth investigation presented of the decompressed image quality, demonstrating that RIFD-DLT preserves a high degree of quality with minimal distortion that is undetectable to the human eye.

The structure of the paper is as follows: In Section 2, relevant research and current image compressing methods are reviewed. Section 3 describes the elements of the RIFD-DLT algorithm. In Section 4, the experimental setup and results are presented along with a comparison of RIFD-DLT with other cutting-edge techniques. Section 5 brings the paper to a close and explores possible directions for future research in this field.

RIFD-DLT is an effective solution for applications that require high-quality images along with high compression ratio, and it offers a significant advancement in image compression by meeting the growing need for efficient near-lossless image compression.

2. Relevant research and current image compressing methods

The ongoing growing in image generation coupled with the need for effective storage and transmission solutions have encouraged research into the development of efficient lossless and Near-Lossless image compression algorithms (Rahman *et al.,* 2021; Patidar *et al*., 2020; Xue *et al.,* 2023).

An overview of recent developments is given in this section, with particular attention dedicated to the integration of advanced methods with more traditional approaches and recent advances.

Traditional lossless image compression algorithms have laid the groundwork for more advanced algorithms by addressing basic redundancies in image data. Among these methods, Huffman Coding, Run-Length Encoding (RLE), Lempel-Ziv-Welch (LZW) and Bi-Level Burrows-Wheeler Compression Algorithm (BBWCA) compression are notable. Each of these techniques has made a substantial contribution to the advancement of image compression technology (Baidoo*,* 2023; Klen, 2023; Jabeen *et al*., 2022).

Huffman coding is a basic lossless compression method that gives characters variable-length codes according to how frequently they appear. Shorter codes are given to characters that appear more frequently, whereas longer codes are given to characters that appear less frequently. Although it may be less effective for data with less frequent distributions, this approach works well for images with more repeated intensities. Huffman coding frequently requires integration with other methods to enhance image compression performance. This constraint leaves a gap since Huffman coding is unable to completely utilize the spatial properties of images on its own (Karim *et al*., 2021; Ahmadu *et al*., 2021; Rahman *et al*., 2019).

RLE compresses Images by detecting and storing sequences of repeated values. RLE is easy to use and efficient for data with long runs of repeated values, but it is poor at handling high variability or noisy data (Shnain *et al*., 2020).

LZW is a dictionary-based compression dynamically creates a dictionary of patterns found in the input data. LZW has been effectively utilized in formats such as Graphics Interchange Forma (GIF) and Tagged Image File Format (TIFF) due to its efficiency in compressing repeating sequences. However, for non-repetitive or highly complex data, its performance may deteriorate (Hossain *et al*., 2023; Amoah *et al*., 2023).

The Burrows-Wheeler Transform (BWT) is creatively combined with a two-level compression strategy designed specifically for bi-level (binary) images in the BBWCA (Ali *et al*., 2010). Two BWT applications are used in BBWCA's process: first, the image is segmented into blocks, and then each block is transformed to improve locality and suitability for conventional entropy coders such as Huffman or arithmetic coding. A second BWT transformation then clusters similar values more successfully by capturing higher-order redundancy patterns. By extensively leveraging local and global redundancies, BBWCA can achieve large data size reduction while maintaining image quality, outperforming competing algorithms such as RLE and Huffman coding, as well as classic BWT-based approaches. This is made possible by the dual transformation strategy. The improved compression ratios of BBWCA are validated by experimental results, which show that it is effective in obtaining high compression efficiency for bi-level images (Uthayakumar *et al*., 2020; Shalayiding *et al*., 2020). The BBWCA's very high time complexity for the compression and decompression processes is one of its most noticeable drawbacks. The algorithm's block-based processing, which requires significant CPU resources to handle each block separately, is the source of this inefficiency. As a result, BBWCA may perform slower in terms of processing time than algorithms that do not rely on block-based approaches.

Joint Photographic Experts Group (JPEG) lossless compression version combines entropy and predictive coding, which improves upon fundamental methods like Huffman coding. With this approach, image data performs better, and the compression results are acceptable. But there are several drawbacks to JPEG's lossless mode, especially when it comes to utilizing spatial redundancies to their maximum. The JPEG lossless mode's performance may not be ideal for complex images. Although it works well in many circumstances, it might not always attain the maximum compression ratios. JPEG's near-lossless compression version uses quantization and other techniques to reduce data size while maintaining acceptable image quality, providing a balance between high compression ratios and minimum perceptual loss. Compared to traditional lossless techniques, this mode performs better, especially in situations when a small degradation is acceptable in exchange for much smaller file sizes. But just like its lossless equivalent, JPEG's near-lossless mode might have trouble resolving spatial redundancy in highly detailed or complex images, which could limit its ability to achieve the ideal balance between image quality and compression ratio (Xiao *et al*., 2020; Otair *et al*., 2022).

The JPEG-LS technique for lossless image compression combines Golomb-Rice coding with predictive coding to produce effective compression at a low computational cost. It is excellent at preserving high image quality and providing many image forms with efficient compression. JPEG-LS, however, does not always offer the best compression ratios, especially for images with complex patterns. JPEG-LS can be modified for near-lossless applications, allowing for minor image fidelity compromises that result in better compression ratios and generally acceptable quality. Even with these advancements, its effectiveness is still dependent on the complexity of the image, indicating a gap where more sophisticated methods could provide better outcomes (Descampe *et al*., 2021; Cao *et al*., 2021).

JPEG 2000, Advanced features for near-lossless and lossless image compression are available with JPEG 2000. Wavelet transforms and arithmetic coding are two techniques used by JPEG 2000 in its lossless mode to achieve high compression efficiency while maintaining the original image data. Although this approach guarantees perfect data recovery, its usage in real-time or resource-constrained contexts may be restricted due to its high computational cost. The near-lossless mode, on the other hand, uses comparable wavelet transformations but allows controlled quantization, resulting in a small amount of data loss in order to obtain larger compression ratios. Because it finds a balance between compression efficiency and image quality, it requires less computing power and is therefore more appropriate for applications where a small amount of quality loss is acceptable in exchange for improved compression (Otair *et al*., 2022; Koc *et al*., 2020).

Delta Encoding is a lossless method that compresses data by recording differences between values rather than the actual values. This technique uses similarities between neighboring pixel values to significantly decrease the size of the data. But when applied alone, Delta Encoding performs poorly, especially in complex images where pixel differences are less regular. Therefore, to get the best results, Delta Encoding frequently needs to be integrated with other techniques (Zhou *et al*., 2022; Tan *et al*., 2020).

Efficient Lossless Image Compression Algorithm via Column Subtraction Model (LICA-CS) is used to address data redundancy. This technique can deal with different image types and effectively reduce repetition. Nevertheless, LICA-CS compression ratio is poor when dealing with low resolution images such as raster map images. This variability point to a weakness in the overall efficiency of LICA-CS (Alqerom *et al*., 2024);

Graphics Interchange Format (GIF) is being replaced by the widely used Portable Network Graphics (PNG) format, which uses lossless image compression. To improve compression performance, PNG uses adaptive filtering in conjunction with Deflate compression, which combines Huffman and LZ77 coding. By predicting pixel values based on nearby pixels, the filtering procedure maximizes the efficiency of Deflate. PNG may be less effective for images that are extremely detailed or complicated, even while it works well for images with huge uniform areas or simple patterns. Furthermore, in situations when resources are limited or real-time processing is required, the computational costs of PNG's compression techniques may affect performance. Despite these drawbacks, PNG is still a well-liked option for lossless compression because of its cross-platform compatibility and capacity to maintain picture quality. Although PNG compression is basically lossless, it can be adjusted for nearly lossless compression by changing certain parameters (Thakur, 2022; Prasantha *et al*., 2020).

Rounding the Intensity Followed by Dividing (RIFD) near lossless image compression algorithm begins by rounding pixel values. Next, a division phase is applied to further compress the data. By prepping image data for more efficient encoding, this method increases compression ratios. Rounding and division may result in some quality loss. RIFD is frequently used in conjunction with other methods, such as enhanced entropy coding, to address this problem. With this integration, performance is improved and improved image quality is maintained across a variety of image formats (Thakur & Shehadeh, 2016).

An enhanced merging of the RIFD and Delta Encoding methods is represented by the RIFD-DLT algorithm. This hybrid strategy improves overall compression efficiency while addressing the shortcomings of each separate method:

- RIFD: By dividing and rounding pixel intensities, this technique compresses image data and gets images ready for more effective compression by lowering pixel variability.
- Delta Encoding: This technique, which comes after RIFD, significantly reduces the size of the image by encoding the variations in pixel values between neighboring rows. This simplifies the data and cuts down on redundancy for better compression.

RIFD-Delta outperforms traditional and other advanced algorithms in terms of performance and compression ratios. Its efficiency in achieving superior outcomes in reducing data size while providing high image quality is illustrated by its higher compression ratios on benchmark image datasets from Kodak, Waterloo, and EPFL. For applications requiring high-quality image reconstruction, RIFD-DLT provides a reliable solution by efficiently decreasing data size with undetectable distortion by the human visual system.

Experiments comparing RIFD-DLT to well-known algorithms like JPEG-LS, JPEG 2000, and JPEG-XR show significant gains in compression effectiveness. The RIFD-DLT framework's combination of RIFD and Delta Encoding provides a sensible trade-off between computational complexity and compression efficiency. Because of this combination, RIFD-DLT performs more effectively and practically than previous approaches, which makes it an excellent option for real-world applications where we need to optimize the processing needs and the compression performance.

3. The proposed RIFD-DLT algorithm

We first go over the general idea behind the RIFD-DLT algorithm in this section, along with the reversible encoding and decoding stages.

3.1 Overview of the proposed Algorithm

A near lossless image compression algorithm adopting Rounding the Intensity Followed by Dividing RIFD and Delta Encoding RIFD-DLT is being proposed for compressing colored and grayscale images. The RIFD-DLT algorithm is a highly adaptable and efficient technique for compressing images with minimal distortion that is imperceptible to the human visual system. It can be applied to various applications with great effectiveness. The algorithm is universally compatible with most image formats and has the capability to process images of varying resolutions, ranging from low-quality to high- quality images. RIFD-DLT can function autonomously as a compression algorithm or as a pre-processing phase for any other lossless compression methods.

3.2 RIFD-DLT Encoding & Decoding

Fig. 1 depicts the RIFD-DLT image compression algorithm, which has two primary phases: RIFD and Delta Encoding. The objective of this strategy is to achieve near-lossless compression by rounding and dividing image intensity values, ensuring high visual fidelity with minimal distortion and decreasing the correlation between different parts of the image, effectively. By integrating delta encoding with RIFD, the algorithm improves the RIFD compression efficiency while maintaining image quality, making it suitable for both colored and grayscale images. The objective of the RIFD-DLT algorithm is to provide a flexible and adaptable compression technique that preserves image details, ensuring that any distortion remains imperceptible to the human visual system.

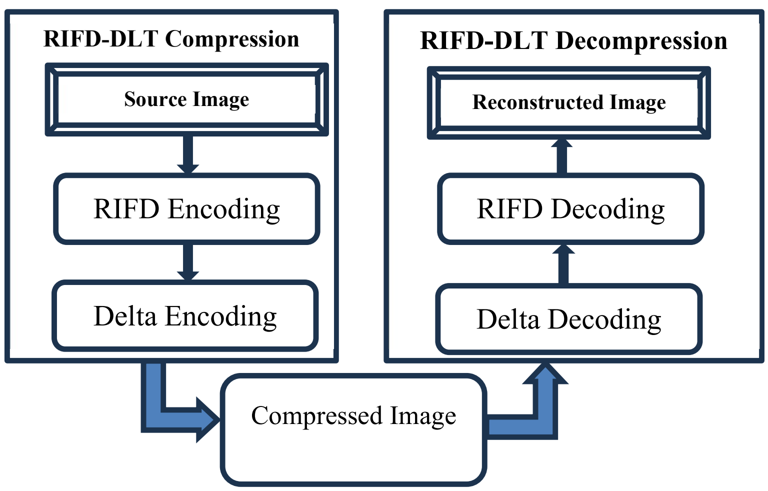

Fig. 1. RIFD-DLT image compression scheme

Phase 1- Rounding the Intensity Followed by Dividing RIF*D:* An essential part of the proposed near-lossless image compression methodology is the RIFD technique. By taking advantage of features of the human visual system, this method seeks to reduce the quantity of image data while preserving excellent visual quality. Rounding the intensity values and dividing the rounded values are the two main processes in RIFD.

The first process is rounding the image pixel intensity values to the closest predetermined levels. By decreasing the granularity of intensity values, this rounding technique aids in lowering the entropy of the image data. The alterations are undetectable to the human eye since the rounding levels are carefully selected to guarantee that the resulting distortion stays below the threshold of human visual perception.

The RIFD second process is to divide each of the intensities by a fixed factor. This division narrows the intensity value range, which further compresses the data. Selecting the appropriate division factor is essential as it must strike a balance between the trade-offs of compression effectiveness and image quality. While a smaller division factor keeps more detail but accomplishes less compression, a larger factor results in better compression but may produce more obvious distortions. After completing the two RIFD phases the image is ready to compress using any compression algorithm such as Huffman Coding.

In digital imaging, an RGB image is composed of three-color channels: Red, Green, and Blue. Each pixel in an RGB image is represented by a triplet of intensity values corresponding to these three primary colors. These values typically range from 0 to 255

in an 8-bit image, where 0 represents the minimum intensity of color and 255 represents the maximum intensity. The combination of different intensity levels of the red, green, and blue channels creates the full spectrum of colors perceived in the image.
To apply the RIFD equation to an RGB image, we process each of the three-color channels separately using the same RIFD transformation. The process for each Red, Green, and Blue involves rounding the intensity values to the nearest predefined level and then dividing the rounded values by a constant factor. Eq. (1) expresses the transformation on each channel as:

$$Dc(i,j) = \frac{Round(Ic(i,j))xk}{d} \qquad \forall 1 \leq i \leq m \ \& \ 1 \leq j \leq n \tag{1}$$

Let m and n be the matrix row size and column size respectively. DC(i,j) is the final transformed intensity value of the channel Red, Green, or Blue at position (i,j). IC(i,j) is the original intensity value of the channel Red, Green, or Blue at position (i,j). k is the rounding factor. d is the division factor.

Rounding and dividing together result in a substantial reduction in the quantity of data needed to represent the image, all the while maintaining a high level of visual quality. Due to its ability to reduce noticeable compression artifacts, this approach works especially well for images that have regions of uniform intensity and smooth gradients. The proposed technique, which uses the RIFD method as preprocessing phase, produces a high compression efficiency with minimal loss of image quality, which makes it appropriate for a variety of uses such as multimedia storage, remote sensing, and some medical imaging (Rossinelli *et al.*, 2020). Further encoding using lossless compression algorithms such as Huffman Coding can be done as a post processing phase for RIFD to achieve a higher compression ratio. In this paper the author adopted the delta encoding to improve the total compression performance, but it all starts with the RIFD approach.

Phase 2- Delta Encoding: Delta Encoding is an essential phase in the proposed near-lossless image compression algorithm. This technique focuses on encoding the differences between consecutive pixel values rather than the pixel values themselves. By exploiting the spatial redundancy present in images, delta encoding significantly reduces the amount of data required for image representation.

Delta starts by implementing the initial phase of delta encoding on each of the transformed matrices Dc(i,j) to calculate differences. The algorithm iterates over each pixel in the transformed matrices and computes the delta by subtracting the current pixel value from the value of the pixel immediately next to it. For a given pixel Dc(i,j) in a matrix, the delta value DLT(i,j) is computed as defined in Eqs.2 for each of the three matrices separately. The resultant delta is placed in the identical position as the original pixel value in the new matrix D(i,j) as specified in Eqs.3. This successfully captures the difference between consecutive pixel values, commencing with the first row. It is crucial to acknowledge that the final value does not have a subsequent value and should be directly saved without any subtraction. This calculation is performed row-wise, and similar operations are done column-wise to further enhance the compression efficiency.

$$DLT(i,j) = Dc(i,j) - Dc(i-1,j) \qquad \forall 1 \leq i \leq m \ \& \ 1 \leq j \leq n \tag{2}$$

$$D(i,1) = \mathrm{Dc}(i,1) \qquad \forall 1 \leq i \leq m \ \& \ 1 \leq j \leq n-1 \tag{3}$$

Delta encoding produces a more compressed representation by concentrating on the differences rather than the absolute values. It enhances the compression ratios due to the decreased entropy and variability of the delta values. Delta encoding helps to retain high visual fidelity by preserving the important image information by mainly utilizing the redundancy within the image. By significantly reducing the visual data without causing noticeable distortion, delta encoding improves the RIFD technique.

A highly effective and versatile near-lossless image compression system is produced when RIFD and delta encoding are combined. This approach can be used for a variety of purposes, such as multimedia storage and remote sensing. The combination between RIFD and Delta encoding achieved a compromise between compression effectiveness and image quality, guaranteeing that the compressed images are both less in size and high quality.

4. Numerical results

Extensive experiments were carried out to evaluate the effectiveness of the proposed RIFD-DLT algorithm. This section describes the methodologies utilized as baselines for comparison, the datasets that were used, and the conclusions drawn from the datasets.

4.1 About the Datasets

The evaluation utilized the following datasets of 8-bit RGB images:

Kodak: This benchmark dataset, which comes from the Kodak corporation, is utilized in studies on image processing and compression. It has twenty-four images that span a range of subjects and scenes (KaggLE, 2024).

Waterloo: This well-known dataset, made available by the University of Waterloo's Fractal Coding and Analysis Group, is widely utilized in studies to assess different imaging methods since it offers a thorough examination for various compression situations. It comes in three sets: the Color Set, the Greyscale Set 1, and the Greyscale Set 2, each set contains twelve small, twelve medium, and eight large images respectively (Gregory, 2009).

EPFL: This well-known collection of images, from the École Polytechnique Fédérale de Lausanne, is frequently used in computer vision and image processing research. It includes ten color, high-resolution images that could be used for assessing compression performance on detailed visuals (EPFL, 2009).

4.2 Experiments setup

A computer system with an Intel(R) Core (TM) i7-10510U CPU (1.80–2.30 GHz), 8 GB of RAM, and 1 GB of virtual memory was used for the research. The MATLAB programming environment was used to implement the algorithms.

4.2 Experimental Results

Compression ratios compared with standard algorithms: Initially the compression ratios of the compressed images expressed in Bits Per Pixel (BPP) and examined using Eqs.4 to determine these ratios (Alqerom *et al*., 2024).

$$BPP = \frac{8 \times s}{n} \tag{4}$$

The original image has n pixels, and the compressed image's size in bytes, including its three independently compressed components, is represented by s. The compression ratio is calculated by dividing the size of the compressed image by the total number of pixels. Therefore, a lower compression ratio signifies more efficient compression. The saving percentage is calculated using Eq. (5) (Alshami *et al*., 2018).

$$Saving\% = \frac{Original\ Image\ Size\ -\ Compressed\ Image\ Size}{Original\ Image\ Size} \times 100 \tag{5}$$

Simple statistical error metrics are essential in full-reference objective quality assessment, including:

Mean Squared Error (MSE): As one of the simplest and most widely used methods for full-reference image quality assessment, MSE is defined by Eq.6. In this equation, r and j represent pixel coordinates, x1 denotes the pixels of the compressed image, and x2 represents the pixels of the source image (Otair *et al*., 2023). Better detail preservation of the original image is shown by a smaller MSE between the original and reconstructed images.

$$\mathrm{MSE} = \frac{\sum_{r}^{j}[x1(\mathrm{r,j}) - x2(\mathrm{r,j})]^2}{\mathrm{r} \times \mathrm{j}} \tag{6}$$

Peak Signal to Noise Ratio (PSNR): Commonly used to quantify image quality, PSNR is defined by Eq. (7) and is particularly useful within optimization frameworks (Otair *et al*., 2023). These metrics are crucial for measuring and analyzing distortion in image compression processes. Better quality is indicated by higher PSNR values.

$$\mathrm{PSNR} = \frac{10 \times \log_{10}(\mathrm{Intensity})^2}{\mathrm{MSE}} \tag{7}$$

Various tests were conducted to evaluate the performance of the proposed algorithm, aiming to demonstrate its effectiveness and improvements in terms of compression ratio, image quality, and running time. The following subsections discuss these aspects in detail.

5. A comparison of compression size between RIFD-DLT and standard approaches

Several lossless and near lossless standard techniques are used to assess and contrast the performance of the proposed method, including RCT with JPEG2000, JPEG-LS, JPEG XR, Huffman algorithms from (Alqerom *et al*., 2024), and RIFD-Huffman. Eq. (4) was employed to compute the compression ratios (BPP) that resulted from applying these techniques to three colored datasets (Waterloo, Kodak and EPFL). Table 1 presents a summary of the findings, with the best values indicated in bold. The popular and well-established standard methods are outperformed by the proposed algorithm. The best results are obtained with RIFD-DLT compression (method #F), which is followed by RIFD-Huffman (method #E) and RCT-JPEG-LS (method #B).

Table 1
Average compression ratios (BPP) for the proposed RFID-DLT compared with standard algorithms

Standard Methods			Datasets			Average
No	Type	Method	Waterloo	Kodak	EPFL	
A	Lossless	RCT-JPEG2000	11.21	9.51	10.84	10.52
B	Lossless	RCT-JPEG LS	8.96	9.57	10.47	9.67
C	Lossless	RCT-JPEG XR	13.32	10.92	11.76	12.00
D	Lossless	RCT-Huffman	16.55	15.06	16.9	16.17
E	Near Lossless	RIFD-Huffman	9	8.72	9.35	9.02
F	Near Lossless	RIFD-DLT	4.1	4.34	4.19	4.21

When Delta encoding is used after RIFD, the RIFD-DLT approach produces the best results, with an average compression ratio of 4.21. This combination demonstrates how RIFD and Delta encoding work well together to produce more efficient compression results. RIFD reduces the data's range and noise to prepare the data for Delta encoding, which effectively compresses the data by encoding the differences. This combination improves the compression ratio by utilizing the advantages of both approaches. In particular, with an average improvement in compression ratio of 4.81 and 5.46 ahead than (method #E) and in (method #B) respectively, RIFD-DLT surpasses both the original RIFD-Huffman strategy and the best-performing standard approach, JPEG-LS. Fig. 2 provides a visual representation of the data in Table 1, which facilitates understanding. It shows a comparison between standard methods and the average compression ratios (BPP) of the Proposed RIFD-DLT algorithm.

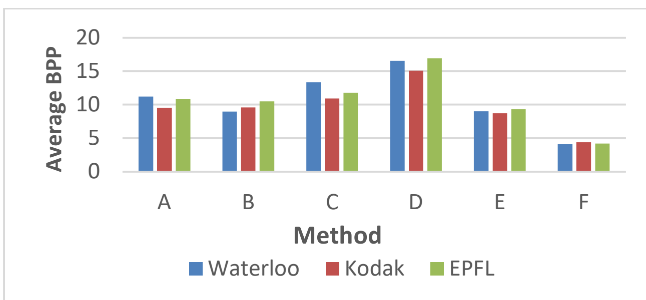

Fig. 2. Average compression ratios (BPP) for the proposed RFID-DLT algorithm compared with standard algorithms

Running time compared with RIFD-HUFFMAN schemes: The same settings as in Section (ii) were used for the evaluation. The three colored datasets (Waterloo, Kodak and EPFL) were subjected to testing of both proposed methods and the RIFD-Huffman method. Table 2 lists the total running time for compression and decompression in seconds for each algorithm. The RIFD-Huffman algorithm takes 50.99 seconds to compress and decompress all the images from the three datasets, compared to 34.62 seconds for the RIFD-DLT technique. Consequently, the RIFD-DLT outperforms the RIFD-Huffman by 16.37 seconds and is considered as one of the fastest algorithms.

Table 2
Total compression and decompression time (In Secs) for the propsoed method compared with RFID-HUFFMAN method using the three image datasets

Image	RIFD-Huffman		RIFD-DLT	
	Compression Time (s)	Decompression Time (s)	Compression Time (s)	Decompression Time (s)
Waterloo	11.6	9.72	1.57	2.23
Kodak	3.72	4.63	3.61	4.71
EPFL	11.6	9.72	10.2	12.3
Total	50.99		34.62	

Image quality compared with RFID-HUFFMAN schemes: The proposed method's image quality performance was assessed and compared with RIFD-Huffman. Three color datasets (Waterloo, Kodak, and EPFL) were used to evaluate the image quality obtained from applying these approaches using MSE Eq.6 and PSNR Eq. (7). The results as illustrated in Table 3 show that the proposed near-lossless technique produced excellent image quality following decompression. The RIFD phase is the reason for the slight distortion that has been noticed. Interestingly, both the proposed RIFD-DLT algorithm and the RIFD-Huffman algorithm show the same degrees of distortion that are unnoticeable to the human visual system with an average PSNR of 52.22 dB, 51.3dB and 58.51dB for the Waterloo, Kodak and EPFL image sets respectively. Because of this, they are very efficient at compressing images while maintaining a high compression ratio and good image quality.

Table 3
Average MSE and PSNR for the proposed method compared with RIFD-HUFFMAN method

	RIFD-Huffman		RIFD-DLT	
Image	MSE	PSNR	MSE	PSNR
Waterloo	2.78	52.22 dB	2.78	52.22 dB
Kodak	2.93	51.30 dB	2.93	51.30 dB
EPFL	2.88	58.51 dB	2.88	58.51 dB

Fig. 3 provides a visual representation of the data in Table 3, which facilitates understanding. It shows a comparison between RIFD-Huffman algorithm and the Proposed RIFD-DLT algorithm in terms of MSE.

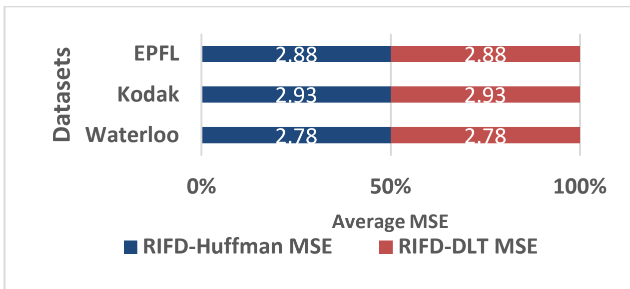

Fig. 3. Average MSW for the proposed RFID-DLT algorithm compared with RIFD-HUFFMAN

Comparison with benchmark schemes: The proposed RIFD-DLT algorithm was assessed using benchmark software and several popular image compression algorithms (Alqerom *et al*., 2024); (Khan *et al*., 2017); (Shalayiding *et al*., 2020); (Waleed *et al*., 2020). in addition to the findings that were previously published. We evaluated compression efficiency using the Kodak dataset. Table 4 provides a summary of each technique's compressed file sizes in kilobytes (KB). The RIFD-DLT method outperforms all other benchmark software and approaches stated, exhibiting notable gains in compression efficiency. In particular, RIFD-DLT (method #S) achieves a remarkable 81.5% size reduction by compressing 11,520 KB of data down to 2,131 KB. The file size was reduced to 3,039 KB with the YCoCg-CSC (method #V) approach, demonstrating competitive results as well. This letter is a 73.15% reduction. LICA-CS (method #M) and LZ4X (method #N), on the other hand, exhibit noteworthy performance, achieving average reductions of 71.2% and 62.6% respectively.

The data from Table 4 are displayed in Fig. 4 as a chart that displays the size of compressed images in (KB) for the proposed RIFD-DLT method in addition to various compression techniques.

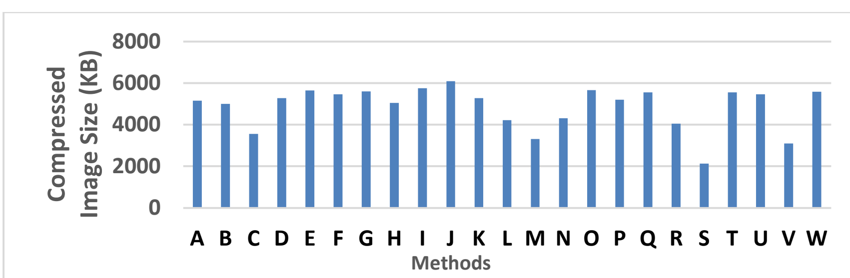

Fig. 4. Total compressed image size in (KB) for the proposed RIFD-DLT algorithm along with several benchmark software and compression algorithms

Table 4
Compressed image sizes (KB) for the proposed RIFD-DLT along with several benchmark software and compression algorithms

		KODAK TEST IMAGE										
No	**Scheme**	**Kodim02**	**Kodim08**	**Kodim13**	**Kodim21**	**Kodim22**	**Kodim23**	**Kodim07**	**Kodim20**	**Kodim05**	**Kodim11**	**Total**
A	ADVANCE COMP	608	422	495	493	477	666	460	468	614	451	5154
B	ALLUME	386	576	487	503	699	407	788	370	410	372	4998
C	BBWCA	381	463	352	480	308	346	350	332	244	297	3553
D	BCM	457	785	628	490	532	737	375	381	507	390	5282
E	BULK ZIP	450	710	372	441	753	613	469	751	482	603	5644
F	CAESIUM	475	422	496	518	797	500	706	407	736	411	5468
G	C-MIX	453	510	457	497	625	520	761	666	657	456	5602
H	COMPRESSOR.IO	413	765	425	597	380	415	392	391	603	668	5049
I	CRUSH	409	692	556	536	544	795	646	612	583	376	5749
J	FILE MINIMIZER	560	758	471	521	740	681	691	596	573	495	6086
K	FILE OPTIMIZER	409	432	721	665	559	502	401	447	618	518	5272
L	HEVC (x265)	403	493	343	487	397	538	326	418	462	353	4220
M	LICA-CS	320	395	396	311	355	297	282	270	376	348	3310
N	LZ4X	370	420	412	373	402	537	465	456	444	429	4308
O	MRP	497	760	550	511	513	791	534	474	628	409	5667
P	NANOZIP	475	556	472	519	446	543	490	431	551	715	5198
Q	PAQ8PXD_V4	450	490	598	607	489	655	733	596	372	569	5559
R	RIFD-Huffman	273	488	451	388	418	438	399	383	447	369	4054
S	RIFD-DLT	**186**	**248**	**279**	**218**	**199**	**171**	**194**	**184**	**240**	**212**	**2131**
T	UPACK 0.25	661	710	379	675	503	529	371	572	587	568	5555
U	WINRK 3.1.2	598	515	398	783	674	457	374	611	456	593	5459
V	YCoCg-CSC	296	380	380	287	334	265	254	259	355	283	3093
W	ZCM 0.92	495	631	772	542	408	714	565	597	416	450	5590

Comparison of compression ratios for grayscale images: To assess the obtained compression ratios, a comparison of twelve chosen images from Waterloo grayscale image sets 1 and 2 was conducted. The final results, displayed in Table 5, demonstrate how well each algorithm performs in terms of compression ratio. On average, the proposed approach outperformed all other algorithms, achieving a 43.7% better compression ratio over the RIFD-Huffman.

Table 5
Compression ratios (BPP) for the proposed RIFD-DLT algorithm compared with state-of-the-art schemes using 12 selected Waterloo images

Image	**Huffman**	**Arithmetic**	**J2K**	**JLS**	**7-Zip**	**IWT-HF**	**LICA-CS**	**RIFD-Huffman**	**RIFD-DLT**
bird.TIF	6.8016	6.7747	3.1390	3.4712	4.2333	2.8630	2.45	2.66	1.17
bridge.TIF	7.6937	7.6689	5.9086	5.7904	6.3164	4.9043	4.00	3.07	1.83
circles.TIF	1.8484	1.7811	1.2627	0.1526	0.1136	1.3375	1.11	1.30	1.05
crosses.TIF	1.0001	0.1879	1.4285	0.3855	0.1786	2.0281	1.24	1.00	1.13
slope.TIF	7.5411	7.5177	1.0643	1.5713	1.6929	1.6449	1.77	3.37	1.11
squares.TIF	1.3517	1.0776	0.2505	0.0771	0.0500	0.6989	1.05	1.04	1.02
boat.TIF	7.1468	7.1238	4.1005	4.2498	5.2881	3.5184	3.13	2.46	1.35
goldhill2.TIF	7.4970	7.4779	4.6544	4.7116	5.5985	3.8035	3.29	2.93	1.40
lena2.TIF	7.4683	7.4456	4.0166	4.2437	5.5190	3.2545	3.12	2.96	1.27
library.TIF	5.8704	5.8490	5.8350	5.1011	4.2544	5.1664	3.72	2.61	1.85
mandrill.TIF	7.3804	7.3580	6.0232	6.0365	6.3869	5.0441	4.24	2.86	2.00
peppers.TIF	7.5951	7.5716	4.4042	4.4887	5.5480	3.5105	3.16	3.04	1.32
Average	**5.7662**	**5.6528**	**3.5073**	**3.3566**	**3.7650**	**3.1478**	**2.69**	**2.44**	**1.37**

Using a selection of Waterloo images from Table 5, Fig. 5 shows the average Compression Ratios (BPP) for the proposed RIFD-DLT algorithm in comparison to State-of-the-Art algorithms.

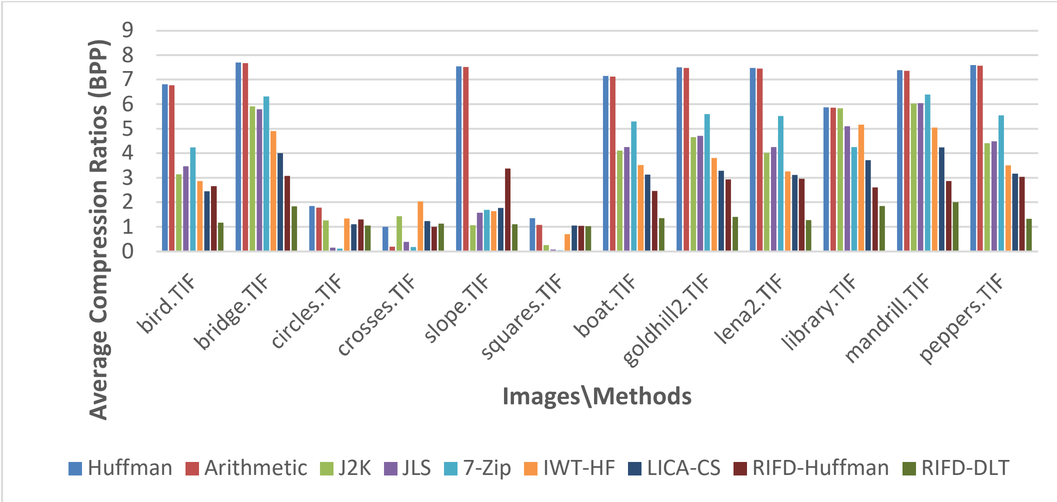

Fig. 5. Average compression ratios (BPP) for the proposed RIFD-DLT algorithm with state-of-the-art schemes using selected Waterloo images

6. Conclusion

In order to present a near-lossless image compression algorithm known as RIFD-DLT, the author of this study introduced a hybrid technique combining Original Rounding the Intensity Followed by Dividing (RIFD) with Delta Encoding. The proposed approach uses RIFD initially to lower image intensities, which prepares the image for further compression phases. Then, by effectively encoding the updated data and taking advantage of the closeness of pixel values in nearby rows, Delta Encoding is used to further reduce image intensity. Using this procedure, the adjacent rows in each of the three-color matrices of the image are subtracted. Multiple evaluations showed that RIFD-DLT performs far better than state-of-the-art algorithms and benchmarks, especially when it comes to compression ratio and running time.

Improvement for RIFD-DLT in terms of compression ratio: As demonstrated in Table 4, the proposed approach achieves an 81.93% reduction and a 43.7% improvement in the compression ratio over the RIFD-Huffman by compressing the data from 11520 KBs to 2131 KBs. The RIFD-DLT approach performs more effectively than any benchmark software/algorithm combination shown in Table 4. With 3310KBs, the overall file size was reduced by 71.2% in comparison to the LICA, which displayed competitive compression ratios.

Improvement for RIFD-DLT in terms of running time: Table 2 illustrates that, although the RIFD-DLT method takes 34.62 seconds to compress and decompress all the images from the three datasets, the RIFD-Huffman approach takes 50.99 seconds. As a result, the RIFD-DLT is considered as one of the fastest algorithms, surpassing the RIFD-Huffman by 16.37 seconds.

Improvement for RIFD-DLT in terms of Image Quality: Table 3 summarizes the results, which demonstrate that the proposed near-lossless technique generated outstanding image quality after decompression. The RIFD phase is responsible for the minor distortion that was noticed. The RIFD-Huffman algorithm and the proposed RIFD-DLT algorithm, on the other hand, show the same distortion level that is undetectable to the human visual system, with average PSNR values of 58.51 dB, 51.3 dB, and 52.22 dB for the EPFL, Kodak, and Waterloo image sets, respectively. They can therefore compress images very effectively while preserving a high image quality.

More work could be done on improving the RIFD procedures. Enhancing overall image quality could be accomplished by optimizing the pre-processing stages and investigating adaptive algorithms that dynamically modify compression parameters based on image content.

Acknowledgement

I would like to thank the British University of Bahrain for their initial funding, which was very helpful in making this research possible.

References

Abrardo, A., Barni, M., Bertoli, A., Grimoldi, R., Magli, E., & Vitulli, R. (2011). Low-complexity approaches for lossless and near-lossless hyperspectral image compression. *Satellite Data Compression*, 47-65.

Ahmadu, A. S., Garba, E. J., & Manga, I. (2021). Lossless Image Compression Schemes: A Review. *Journal of Scientific Research and Reports, 27*(6), 14-22.

Al Qerom, M., Otair, M., Meziane, F., AbdulRahman, S., & Alzubi, M. (2024). LICA-CS: Efficient Lossless Image Compression Algorithm via Column Subtraction Model. *Journal of Robotics and Control (JRC), 5*(5), 1311-1321.

Ali, M. A., Khan, A., Javed, M. Y., & Khanum, A. (2010, October). Lossless image compression using kernel based Global Structure Transform (GST). *In 2010 6th International Conference on Emerging Technologies (ICET) (pp. 170-174). IEEE*.

Alqerom, M. (2020). An intelligent system for the classification and selection of novel and efficient lossless image compression algorithms. University of Salford (United Kingdom).

Alshami, A. L., & Otair, M. (2018). Enhancing quality of lossy compressed images using minimum decreasing technique. *International Journal of Advanced Computer Science and Applications, 9*(3).

Amoah, G. A., & Wahab, E. R. A. (2023). Comparative analysis of multimedia compression algorithms. *Asian Journal of Mathematics and Computer Research, 30*(2), 26-37.

Bai, Y., Liu, X., Wang, K., Ji, X., Wu, X., & Gao, W. (2024). Deep lossy plus residual coding for lossless and near-lossless image compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

Baidoo, P. K. (2023). Comparative analysis of the compression of text data using HUFFMAN, arithmetic, run-length, and LEMPEL ZIV WELCH coding algorithms. *Journal of Advances in Mathematics and Computer Science, 38*(9), 144-156.

Cao, S., Wu, C. Y., & Krähenbühl, P. (2020). Lossless image compression through super-resolution. arXiv preprint arXiv:2004.02872.

Cao, Z., Zhang, T., Liu, M., & Luo, H. (2021, January). Wavelet-supervision convolutional neural network for restoration of JPEG-LS near lossless compression image. *In 2021 IEEE Asia Conference on Information Engineering (ACIE)* (pp. 32-36). IEEE.

Cavallaro, G., Heras, D. B., Wu, Z., Maskey, M., Lopez, S., Gawron, P., ... & Datcu, M. (2022). High-Performance and Disruptive Computing in Remote Sensing: HDCRS—A new Working Group of the GRSS Earth Science Informatics Technical Committee [Technical Committees]. *IEEE geoscience and remote sensing magazine, 10*(2), 329-345.

Descampe, A., Richter, T., Ebrahimi, T., Foessel, S., Keinert, J., Bruylants, T., ... & Rouvroy, G. (2021). JPEG XS—A new standard for visually lossless low-latency lightweight image coding. *Proceedings of the IEEE, 109*(9), 1559-1577.

Dua, Y., Kumar, V., & Singh, R. S. (2020). Comprehensive review of hyperspectral image compression algorithms. *Optical Engineering, 59*(9), 090902-090902.

Dumka, A., Ashok, A., Verma, P., & Verma, P. (2020). Advanced digital image processing and its applications in big data. CRC Press.

EPFL. (2023). *http://documents.epfl.ch/groups/g/gr/gr-eb-unit/www/IQA/Original.zip*

Furong, W., Yixing, L., & Long, H. (2022, September). An Image SVD Compression Algorithm for UAV Based on Region of Interest. *In International Conference on Autonomous Unmanned Systems* (pp. 3270-3277). Singapore: Springer Nature Singapore.

Gregory, M. (2009). The Waterloo Fractal Coding and Analysis Group. *https://links.uwaterloo.ca/Repository.html*

Hossain, M. B., & Rahman, M. N. J. (2023, March). An Empirical Analysis on Lossless Compression Techniques. *In International Conference on Computer and Communication Engineering* (pp. 158-170). Cham: Springer Nature Switzerland.

Hu, Y., Yang, W., Ma, Z., & Liu, J. (2021). Learning end-to-end lossy image compression: A benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(8), 4194-4211.

Jabeen, S., Iqbal, K., Maqbool, M. S., & Basit, A. (2022). Predictive Coding Based Lossless Image Compression Model. *Machines and Algorithms, 1*(3).

KaggLE. (2024) Kodak Lossless True Color Image Suite, *https://www.kaggle.com/datasets/sherylmehta/kodak-dataset*.

Karim, A. Z., Miah, M. S., Al Mahmud, M. A., & Rahman, M. T. (2021, September). Image Compression using Huffman Coding Scheme with Partial/Piecewise Color Selection. *In 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)* (pp. 1-6). IEEE.

Khan, A., Khan, A., Khan, M., & Uzair, M. (2017). Lossless image compression: application of Bi-level Burrows Wheeler Compression Algorithm (BBWCA) to 2-D data. *Multimedia Tools and Applications, 76,* 12391-12416.

Klen, D. (2023). Comparison of Traditional Compression Methods of Different Datasets for Cloud Storage Systems (Doctoral dissertation, University of Rijeka. Faculty of Engineering).

Koc, B., Arnavut, Z., Voronin, S., & Koçak, H. (2020, July). Near-lossless Image Compression with Parity Reduction. *In 2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)* (pp. 225-230). IEEE.

Liu, X., An, P., Chen, Y., & Huang, X. (2022). An improved lossless image compression algorithm based on Huffman coding. *Multimedia Tools and Applications, 81*(4), 4781-4795.

Lungisani, B. A., Lebekwe, C. K., Zungeru, A. M., & Yahya, A. (2022). Image compression techniques in wireless sensor networks: A survey and comparison. *IEEE Access, 10*, 82511-82530.

Otair, M. A., & Shehadeh, F. (2016). Research article lossy image compression by Rounding the Intensity Followed by Dividing (RIFD). Research Journal of Applied Sciences, *Engineering and Technology, 12*(6), 680-5.

Otair, M., Abualigah, L., & Qawaqzeh, M. K. (2022). Improved near-lossless technique using the Huffman coding for enhancing the quality of image compression. *Multimedia Tools and Applications, 81*(20), 28509-28529.

Otair, M., Hasan, O. A., & Abualigah, L. (2023). The effect of using minimum decreasing technique on enhancing the quality of lossy compressed images. *Multimedia Tools and Applications, 82*(3), 4107-4138.

Patidar, G., Kumar, S., & Kumar, D. (2020, February). A review on medical image data compression techniques. *In 2nd International Conference on Data, Engineering and Applications (IDEA)* (pp. 1-6). IEEE.

Prasantha, D. H. (2020). Novel Approach for Image Compression using Modified SVD. *International Journal of Creative Research Thoughts (IJCRT), 8*(8), 2234-2243.

Rahman, M. A., & Hamada, M. (2019, October). A semi-lossless image compression procedure using a lossless mode of JPEG. In 2019 IEEE 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC) (pp. 143-148). IEEE.

Rahman, M. A., & Hamada, M. (2021). PCBMS: a model to select an optimal lossless image compression technique. *IEEE Access, 9*, 167426-167433.

Rahman, M. A., Hamada, M., & Shin, J. (2021). The impact of state-of-the-art techniques for lossless still image compression. *Electronics, 10*(3), 360.

Rossinelli, D., Fourestey, G., Schmidt, F., Busse, B., & Kurtcuoglu, V. (2020). High-throughput lossy-to-lossless 3D image compression. *IEEE transactions on Medical Imaging, 40*(2), 607-620.

Shalayiding, A., Arnavut, Z., Koc, B., & Kocak, H. (2020, November). Burrows-Wheeler Transformation for Medical Image Compression. *In 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference* (IEMCON) (pp. 0723-0727). IEEE.

Sharma, U., Sood, M., & Puthooran, E. (2020). A block adaptive near-lossless compression algorithm for medical image sequences and diagnostic quality assessment. *Journal of digital imaging, 33*(2), 516-530.

Shnain, H. W., Abdullah, M. N., & Jeiad, H. A. (2020). Lossless JPEG image compression based on FPGA. *Solid State Technology, 63*(3), 3595-3603.

Strümpler, Y., Postels, J., Yang, R., Gool, L. V., & Tombari, F. (2022, October). Implicit neural representations for image compression. *In European Conference on Computer Vision* (pp. 74-91). Cham: Springer Nature Switzerland.

Tan, H., Zhang, Z., Zou, X., Liao, Q., & Xia, W. (2020, September). Exploring the potential of fast delta encoding: Marching to a higher compression ratio. *In 2020 IEEE International Conference on Cluster Computing (CLUSTER)* (pp. 198-208). IEEE.

Thakur, A. J. (2022). Digital Imaging and Formats. *In Tapping the Power of PowerPoint for Medical Posters and Presentations* (pp. 21-27). Singapore: Springer Nature Singapore.

Ungureanu, V. I., Negirla, P., & Korodi, A. (2024). Image-Compression Techniques: Classical and “Region-of-Interest-Based” Approaches Presented in Recent Papers. *Sensors, 24*(3), 791.

Uthayakumar, J., Elhoseny, M., & Shankar, K. (2020). Highly reliable and low-complexity image compression scheme using neighborhood correlation sequence algorithm in WSN. *IEEE Transactions on Reliability, 69*(4), 1398-1423.

Waleed, M., Um, T. W., Khan, A., & Khan, A. (2020). On the Utilization of Reversible Colour Transforms for Lossless 2-D Data Compression. *Applied Sciences, 10*(3), 937.

Xiao, W., Wan, N., Hong, A., & Chen, X. (2020, November). A fast JPEG image compression algorithm based on DCT. *In 2020 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 106-110). IEEE.

Xue, X., Marappan, R., Raju, S. K., Raghavan, R., Rajan, R., Khalaf, O. I., & Abdulsahib, G. M. (2023). Modelling and analysis of hybrid transformation for lossless big medical image compression. *Bioengineering, 10*(3), 333.

Zhang, H., Cricri, F., Tavakoli, H. R., Zou, N., Aksu, E., & Hannuksela, M. M. (2020). Lossless image compression using a multiscale progressive statistical model. *In Proceedings of the Asian Conference on Computer Vision*.

Zhou, X., Qi, C. R., Zhou, Y., & Anguelov, D. (2022). Riddle: Lidar data compression with range image deep delta encoding. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 17212-17221).