



Improved near-lossless technique using the Huffman coding for enhancing the quality of image compression

Mohammed Otair¹ · Laith Abualigah^{1,2} · Mohammed K. Qawaqzeh¹

Received: 27 November 2020 / Revised: 1 February 2022 / Accepted: 9 March 2022 /

Published online: 30 March 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Digital data compression aims to reduce the size of digital files in line with technological development. However, most data is distinguished by its large size, which requires a large storage capacity, and requires a long time in transmission operations via the Internet. Therefore, a new compress files method is needed to reduce the image size, maintain its quality, utilize storage spaces, and minimize time. This paper aims to improve digital image compression's compression rates by dividing the image into several blocks. Thus, a new near-lossless method using the Huffman Coding technique is proposed. Digital image compression techniques are classified as lossless and lossy. Huffman Coding is a lossless-based technique used in the proposed method to maintain image quality during compression. The proposed method consists of several steps, which are dividing the image into blocks, finding the lowest value in each block and subtracting it from the rest of the values in the same block, then subtracting one from the odd numbers, dividing all the values on two, and finally applying the Huffman Coding technique to the block. The proposed method is applied to a well-known gray and color set with different types and different dimensions. Standard evaluation measures are used (i.e., PSNR, MSE, and CR) to evaluate the proposed method's performance. When compressing images using the proposed method, the results demonstrated 0.11% enhancement when used two by two blocks. It also got high compression rates (25%).

Keywords Near-lossless technique · Huffman coding · Image quality · Image compression

✉ Laith Abualigah
Aligah.2020@gmail.com

Mohammed Otair
Otair@aau.edu.jo

Mohammed K. Qawaqzeh
Qawaqzeh@aau.edu.jo

¹ Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

² School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia

1 Introduction

Compressing digital images is considered one of the most important research fields in image processing due to the benefits yielded [1], where digital image compression works to reduce the size of images in its multiple forms [10, 12]. Because of the tremendous technological development, the captured images have become very large [41], so it has become necessary to search for new compression technologies capable of compressing these images in order to provide enough storage spaces, store the images, and to facilitate the processes of transfer and transmission via the Internet [2, 35].

There are many techniques used to compress images [26, 27]. However, there is a need to search for new ways to provide more excellent compression rates that commensurate with the type of images and the goal of using the images [46]; as the compression rate changes from one technique to another and from one image to another, depending on the goal of using the images [17]. Accordingly, there are two types of classifications for compressing digital images to preserve digital image data [11, 13, 47].

We do not lose any data from the images after decompressing through compression techniques of this type, and lossless compression is usually used to eliminate unimportant data during transmission operations [14, 43]. However, the compression rate in this type of compression is meager. Unlike lossless techniques, some data is lost in this type of compression when decompressing [19]. However, high storage space is provided compared to the lossy compression type [44]. Therefore this type of compression is often used in digital cameras to provide storage space. Due to the enormous technological developments and advancements in images [38–40]. The goal of all types of compression is to reduce the images' size to the lowest extent and not distort the images significantly while maintaining image quality in terms of accuracy and efficiency. Hence the need for more ways and methods to compress images is in high demand.

This paper addresses one of the most critical fields in the modern world, digital images are characterized as massive quantities because almost everyone has a smartphone or a digital camera, and everyone wants to share or display these images. Image quality is rising daily, led by the unprecedented race between smartphone manufacturers to provide the most unique and accurate images. High-quality cameras that would give the perfect image created the need for a compression method that would help transfer such images quickly, without the need for much storage and with the best quality possible. The proposed digital image compression method is based on dividing the image into blocks of different sizes; two-by-two, four-by-four, and eight-by-eight. Next, pass the blocks in these stages by applying mathematical operations to the values in each block and then use the Huffman Coding Technique; it has to compress it, keep the image quality as possible. Therefore, the main goal of the proposed method would yield results with a high compression rate by taking advantage of the lossy compression and maintaining a good ratio of quality using the lossless compression type, meaning benefit from the best of both types. This paper tested a new method to calculate the effect inflicted on the compression rate if a new way of dividing the image into blocks is implementing, thus deciding if the proposed method is efficient and effective. The proposed method got better-compressed images than other comparative methods when used two-by-two blocks.

The contributions of the paper are given as follows.

- Develop a new digital image near-lossless technique.
- Improve the compression rate along with the quality using a new technique.

- Validate and compare the performance of the developed techniques with previously available ones.

The rest of this paper is organized as follows. Section 2 presents the related works that have been published in the literature. Section 3 presents Preliminaries of the current basic method. Section 4 offers the proposed image compression method. Section 5 presents the experimental results and discussion—finally, the research conclusion and future work directions are given in Section 6.

2 Related works

In this section, a preview some related and similar previous works are given.

A new practical learned lossless image compression system is proposed [28], L3C, which outperforms the popular engineered codecs, PNG, WebP, and JPEG2000. The new system replicates the image supply jointly with educated auxiliary representations in RGB space. It also requires three forward passes to predict all pixel probabilities in the place of each pixel. As a result, when compare L3C with the fastest PixelCNN variant (MultiscalePixelCNN), L3C gives a two order speed up, and the auxiliary learning representation is crucial and outperforms predefined support pictures such as an RGB.

The paper presented a technique for processing compressed digital images with compression techniques with the same compression ratio by performing procedures before any lossless method is executed [5]. The method divides the image into 2×2 and reduces the minimum values in pixels for each row and column. The efficiency of the technology has been proven to be applied to images of different sizes and types. Furthermore, the decompressed image using the technique has been improved to compression measures such as mean square error (MSE), mean absolute error (MAE), and peak signal-to-noise ratio (PSNR).

[8] created an end-to-end trainable model for image compression that is based on variational auto-encoders. This model integrates a hyperprior to capture spatial dependencies in the latent representation successfully. This hyperprior relates to side information, a notion that applies to all modern image codecs, however primarily unexplored in the image compression area using artificial neural networks (ANNs); this model trains a complex prior conjointly with the underlying auto-encoder. Therefore it leads to an advanced image compression when measuring visual quality using the popular Multi-structure similarity index and yields rate-distortion performance surpassing known artificial neural networks-based methods when evaluated using a more traditional metric based on squared error.

[42] proposed a new simple algorithm that is more flexible than existing codecs for optimizing auto-encoders for lossy image compression. The end-to-end trained architecture was demonstrated to achieve performance on high-resolution images. Furthermore, the simple algorithm shows effective ways and better understanding than JPEG 2000 regarding Structural Similarity Index and Mean opinion scores for dealing with non-differentiability in training auto-encoders for lossy compression.

This paper aims to introduce a novel lossy technique called RFID for image compressing [30]. The technique depends on collecting the redundancy and similarity between the neighboring pixels of images by rounding the pixels' strengths trailed by the dividing process, which will decrease the range of the intensities and increase the redundancy of these intensities. The algorithm can be applied alone or followed by any lossless compression algorithm; thus, the

RFID is helpful for all-natural images of high-bit depths and colored images. Also, it shows excellent performance when the Huffman algorithm follows RIFD.

The paper presented a discussion on digital image compression techniques: Huffman, Dwt, and Fractal Coding [21]. By taking the resulting images from the three-way compression, comparing the compression ratio, PSNR compared to each other. After comparison, it shows that Fractal Coding is an excellent compression technique for comparing images in two ways. The same comparison can be made in other different methods, such as Neural Networks and Fuzzy logic.

A hybrid of DCT and fractal image compression techniques was proposed in [33], which was implemented using Matlab. The proposed hybrid coding scheme was evaluated using color images. Incomes of the Huffman encoding technique encode the given image. The result shows the effectiveness of the proposed scheme in compressing the color image. When matched to JPEG with image quality 14,12,10,5,3 respectively, they concluded that the proposed technique has successfully compressed the images with high PSNR values.

The authors have produced an approach for recognizing two-dimensional objects by involving digital image processing and geometric logic to recognize breast cancer cells from a given tissue sample [34]. The methods converted the three-dimensional RGB image into two-dimensional black and white image adaptation, color pixel classification for object-background departure, calculating object metrics using area-based filtering, and use of bounding box. Using MATLAB, the algorithm was developed and simulated. The results were 99% accurate using a set of 180 images of the three primary colors (red, green, and blue), and four basic 2D geometric shapes were used for analysis.

[7] present a multi-scale data-compression and a non-separable 2D error control algorithm based on Harten's interpolatory framework for multiresolution that gives a specific estimate of the precise error among original and decoded images. The proposed algorithm does not trust a tensor-product strategy to compress two-dimensional images. After data-compression by applying this non-separable multi-scale transformation, the user gets the exact value of the RMSE, and PSNR before the decoding process occurs. As a result, the proposed algorithm can use to obtain lossless and near-lossless image compression.

A simple and effective method is proposed to compress images in [37]. The method succeeded in the size reduction of images while keeping the quality. The method based on the Wavelet Transform was used to transform the original image. After quantization and thresholding of DWT coefficients, Run-length coding and Huffman coding schemes were used to encode the image. DWT is based on the immensely popular JPEG 2000 technique. As a result, Run-Length Encoder provides a lossless representation of data with a reduced number of bits. Also, the Huffman encoder makes compressed data ready for transmission in the form of the bit stream. Finally, many beneficial researches on images can be found in [1, 6, 16, 18, 25].

3 Preliminaries

This section contains the main concepts of the science of digital image processing and the science of digital image compression, an explanation of some measures of digital image compression, and some previous literary studies related to the paper's subject.

3.1 Digital image processing

Most people worldwide are considered visual beings; they rely on their vision to gather and process information. Some people say they will not believe until they see, and as the famous proverb says, “a picture is worth a thousand words”, therefore the goal of creating images grow dramatically, and with the massive surge of new cheap technologies to capture pictures, the need for processing such images became a necessity [29].

Digital image processing is defined as the science of applying a group of techniques and algorithms on a digital image in order to process, analyze, enhance, extract information or optimize image features such as sharpness and contrast, using a digital computer [36]. This science started back in the 60s in some research. The main goal was to enhance the image’s quality where the image used to be of bad quality. It was then manipulated to make it more transparent, such as the images collected on the moon. It was manipulated after calculating the sun’s position when that image was taken to adjust lighting and other features; such a process was costly and time-consuming. In the 70’s all of that changed due to the production of cheaper, more specialized hardware. It only became cheaper, faster, and more available because all personal smart devices nowadays have all the capabilities to modify. Process images [45], the process of Image processing generally has three steps:

- Importing the image through image acquisition tools;
- Manipulating and analyzing the acquired image;
- The result is an altered image.

An image is defined as a two-dimensional function $f(x, y)$, where x and y are the spatial (plane) coordinates. The amplitude of any pair of coordinates (x, y) is called the intensity of the image at that level; if x, y and the amplitude values of f are finite and discrete quantities, we call the image a digital image. A digital image is composed of a finite number of elements called pixels. Each of which has a particular location and value, the image accuracy and number of pixels in an image go hand in hand. When one goes up, the other goes up too; in other words, a digital image is a 2-D array of pixels. The following figure is an example of a digital image array (matrix) (Fig. 1).

There are three types of digital images [9].

1. Binary Images: Black and white images with a pixel value of 0 or 1.

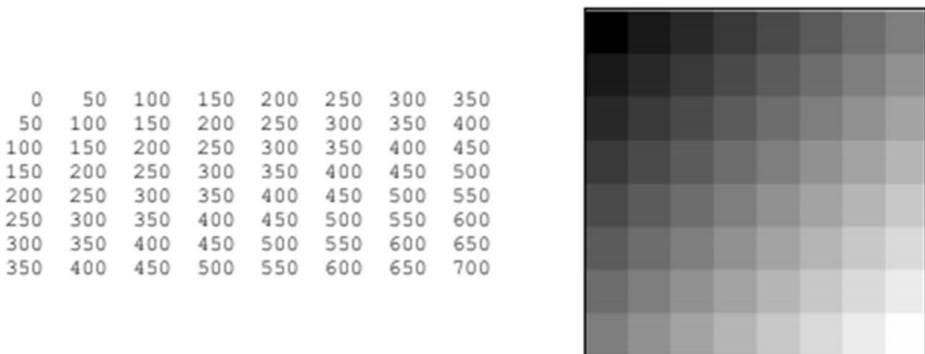


Fig. 1 2-D Image and matrix

2. Grayscale Images: Black and white color and all color gradients in between pixel value between 0 and 255; each pixel represents eight bits to express the color scale.

3. Color Images: images with colors, pixel consists of three eight-bit parts representing the intensity of the red, green, and blue base color. The source of digital images is the electromagnetic (EM) energy spectrum, which is divided into many types shown in Fig. 2 below.

3.2 Digital image compression

Images that are produced nowadays are incredibly high in numbers. With larger size and resolution, and with the advanced technologies and trends among people such as sharing images and videos instantly, the need to reduce an image's size became crucial. Images take up much storage. The bigger the image, the slower it gets transferred; therefore, scientists developed methods to reduce that size, namely image compression.

Digital Image Compression has multiple applications, which vary from image compression for personal use to compressing more essential images such as medical images. Digital Image Compression aims to save a lot of memory space and is therefore extensively used for compression of photos, technical drawings, medical imaging, artworks, maps, and others. Images condensed in size by Digital Image Compression can be quickly sent, uploaded, or downloaded in less time, making the sharing of images a lot easier and faster. When choosing a compression algorithm, the following must be considered [15]:

1. Efficiency: one must use an algorithm that best suits the type of image that would be compressed.
2. Lossless: an algorithm would be chosen based on how much is ok to lose within the image's quality; for example, a medical image must be returned without any data loss; therefore, a lossless method would be used.
3. Compression rate: Higher compression rate means higher efficiency.
4. Complexity/time: an algorithm would be chosen based on the application where the image would be used or sent to some acquire less computational complexity, and faster processing others do not.

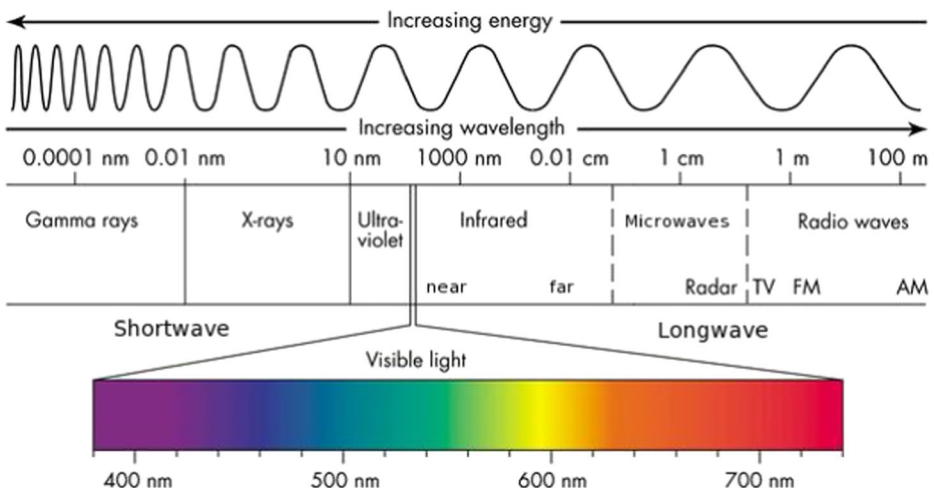


Fig. 2 Electromagnetic energy spectrum

There are multiple benefits gained from image compression [3], such as image transmitting would be of less cost because the cost is related to the time spent on transmitting the image, computing power use would be reduced therefore saved because the less size an image is, the less power for transmission is needed, transmission errors would be less because fewer bits are being transferred, and encoding and compressing of the image would help have a certain level of transmission.

Compression techniques are given as the following general outline:

1. Identifying all similar colored pixels by the same color name, code, and the number of pixels; by doing this, one pixel can match hundreds or thousands of pixels within the image.
2. Create and represent the image using mathematical wavelets.
3. Image would be split into multiple parts, each distinguishable using a fractal. Figure 3 represents the general steps used in image compression [31].

3.2.1 Lossless image compression

As the name tells, lossless image compression tends not to lose any quality within the image, therefore this type of compression is applied to essential types of images such as medical images, which acquire high quality and extreme accuracy. Image formats that use lossless image compression are RAW, BMP, GIF, and PNG. The following techniques are included in lossless compression [28, 32]:

- **Huffman encoding**

An algorithm based on the frequency of occurrence of a symbol in the file is compressed and on statistical coding, meaning that the probability of a symbol has a direct bearing on the length of its representation [22]. The more likely the occurrence of a symbol is, the shorter will the bit-size representation be. Usually, specific characters are used more than others in any given file. Huffman compression is a variable-length coding system that allocates smaller codes for the most frequently used characters and more extensive codes for the least frequently used characters to reduce files' size being compressed and transferred.

- **Run length encoding**

Run-length encoding (RLE) is considered a simple type of data compression in which runs of data (that is, series in which the same data value appears in many successive data elements) are deposited as a single data value and count instead of as the original run [4]. It is the most useful type to be used on data that has many of such runs; it is considered not very beneficial on files that do not have many runs as because it might result in a file with big size, it is considered

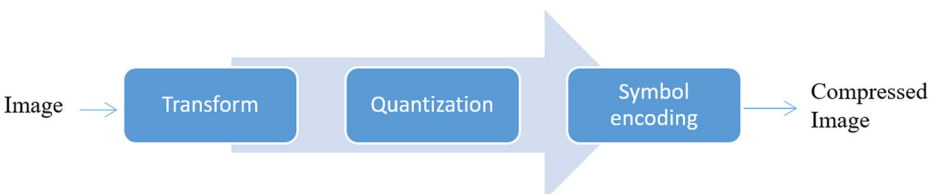


Fig. 3 Image compression steps

appropriate to palette-based images, which means it does not work well on continuous-tone images like as photographs.

- **LZW coding**

It is generally used for lossless text compression and was invented in [20]. It is considered very easy to use and broadly applied for UNIX file compression; this method encodes a series of characters with a unique code using a table-based lookup algorithm. The first 256 8-bit code, 0–255, is entered in a table as an initial entry because an image contains 0–255 distinct pixels. The following codes come from 256 to 4095, inserted into the bottom of the table. This algorithm works better with text compression and performs well when presented with too redundant data files, such as tabulated numbers and computer sources; on the other hand, it does not particularly suit other types of compression.

3.2.2 Lossy image compression

Lossy image compression is where some of the data from your image will be removed, which will minimize the image's size. This slight loss is almost invisible and hard to recognize, but all at the expense of the image's quality. In this technique, the compression rate is higher than the one in lossless compression, meaning the size of the image would be smaller, but this process can not be reversed; therefore, it is always recommended to have a backup of the original image, image formats that use lossy image compression are GIF, and JPEG [23, 24].

4 Procedures and methodology

Huffman technique is the chosen technique in this paper since it is lossless. It maintains the image quality and accuracy. Huffman is applied after several steps in this paper. In this section, the mechanism of work on this technique will be explained in detail. In the next section, many different types of images will be applied to the proposed technology that we called Near-Lossless Image Compression Technique. Then apply the same images to the Huffman technique and compare the results with each other. Explain and discuss the Compression Rate (CR) ratios and compare the quality criteria for images according to the following criteria: Peak signal-to-noise ratio (PSNR), and mean squared error (MSE).

4.1 Set of benchmark images

There are a bunch of special images for checking new algorithms, it can be obtained from the signal and image processing institute of the University of Southern California ("USC-SIPI image Database", 2018), research results are generally compared in a quantitative or visual way, and there are many special measuring tools for measuring research results in digital photo science. There are many types of digital images, digital images were named after this fact that they can be represented digitally, in this paper, two types of digital images were chosen:

- **The gray scale image**

Is the image with the gradation between black and white? These images' matrix values are between (0–255) for black and white, where the value (0) represents black. The value (255) represents white; the remaining numbers are gray hues between dark gray and light gray.

- **The color image**

The color image is composed of three matrices; each matrix contains specific hues, the colors in the three arrays are RGB (Red, Green, and Blue), the combination of three colors gives us the rest of the colors that we see on digital screens. Color digital images are common to use on computer monitors.

4.2 Technologies used to compress digital images

Digital image compression is divided into two types: lossy and lossless. They differ in terms of compression ratio and amount of deformation, each of which has positives; one of the positives of lossy is compression of images with significant compression ratios, but it causes data loss, as for lossless, it maintains data and quality, but with few pressure ratios.

4.2.1 Lossless part

The Huffman technique proposed to work on its compressed images relatively little. It does not affect the accuracy and quality of the image; it was chosen in this proposal to experiment with increasing the compression rate with minimal effect on image resolution and quality. By dividing the image, which helps increase the amount of compress, the proposed technique will also be explained in detail and experiment with a variety of images on it. Color images grayscale images to increase the compression rate while maintaining the image quality and accuracy as possible (Fig. 4).

4.2.2 Enhancement of Huffman by splitting the image into blocks

The main idea of this proposal is to take several steps before applying the Huffman technique to increase the amount of compression and maintain image quality, as shown by the model in the following Fig. 5 and in Algorithm 1. Figure 5, an illustration of the image components below, where the value of the pixel shows the color gamut of the image.

Algorithm 1: Near-Lossless Image Compression Technique steps	
1	Read image file.
2	Split the image into blocks $8 * 8$.
3	Find the minimum value in each block (Store it in a new array) and subtract it from all elements in the block.
4	Subtract one from all odd element in the blocks.
5	Divide all elements from Block by two.
6	Apply Huffman

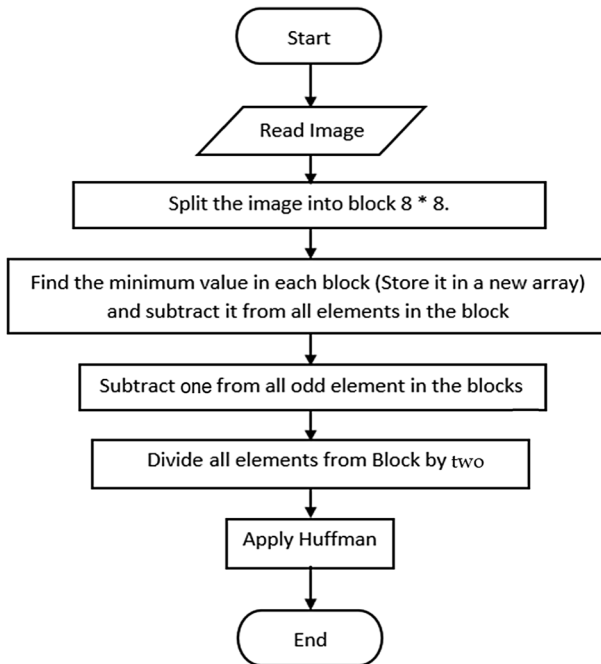


Fig. 4 Near-Lossless Image Compression Technique

4.2.3 Near-lossless image compression technique implementing

Figure 6 (A) Displays the original image data. Figure 6(B) After the process of finding the lowest value in the block and subtracting it from all the values in (A). Figure 6(C) Data after subtracting one from the odd values in Block (B). Figure 6(D) Data (C) after the process of dividing all the numbers by two. Figure 6(E) Huffman application outputs on (D)



Fig. 5 Crop out portrait of Lina

188	231	188	230	188	230	188	230
230	188	231	194	231	188	231	188
195	230	188	230	188	230	188	230
230	188	231	188	189	230	188	189
189	188	230	188	230	194	188	230
230	188	189	230	188	230	189	188
230	188	230	188	188	230	188	230
188	231	188	230	189	188	230	188

A

0	43	0	42	0	42	0	42
42	0	43	6	43	0	43	0
7	42	0	42	0	42	0	42
42	0	43	0	1	42	0	1
1	0	42	0	42	6	0	42
42	0	1	42	0	42	1	0
42	0	42	0	0	42	0	42
0	43	0	42	1	0	42	0

B

0	42	0	42	0	42	0	42
42	0	42	6	42	0	42	0
6	42	0	42	0	42	0	42
42	0	42	0	0	42	0	0
0	0	42	0	42	6	0	42
42	0	0	42	0	42	0	0
42	0	42	0	0	42	0	42
0	42	0	42	0	0	42	0

C

0	21	0	21	0	21	0	21
21	0	21	3	21	0	21	0
3	21	0	21	0	21	0	21
21	0	21	0	0	21	0	0
0	0	21	0	21	3	0	21
21	0	0	21	0	21	0	0
21	0	21	0	0	21	0	21
0	21	0	21	0	0	21	0

D

0	1	0	1	0	1	0	1
1	0	1	2	1	0	1	0
2	1	0	1	0	1	0	1
1	0	1	0	0	1	0	0
0	0	1	0	1	2	0	1
1	0	0	1	0	1	0	0
1	0	1	0	0	1	0	1
0	1	0	1	0	0	1	0

E

Fig. 6 Implementation steps for Near-Lossless Image Compression Technique implementing

After performing the previous studied steps, the Huffman technique was chosen to make the compression rate higher.

4.2.4 Algorithm: Near-lossless image compression technique steps

- 1 Read image file.
- 2 Reverse Huffman compression.
- 3 Split the image into block 8 * 8.
- 4 Multiplying all values in the block by the number (2).
- 5 Add the value stored in dictionary file to all block values.

5 Experimental results and discussion

This section illustrates the proposed Near-Lossless Image Compression Technique results and contribution to the knowledge by conducting a comparison with Huffman coding algorithm results in terms of image quality and size by using two image sets (Colored and Grayscale images). Matlab programming language was used to develop the proposed algorithm as described in Fig. 5. Then the new software had been used to test the images results.

Matlab is software that uses the fourth-generation programming language and works on numerical analysis, includes strengthening and running algorithms. It is possible to work on arrays and set up user interfaces. It is one of the most prominent programs used in the digital image range. There are many versions of Matlab software. We used version (MATLAB 9.4.0) because this version has features that are not available in previous versions, for example, graphical controls.

5.1 Experimental settings

A set of standardized images was applied, and the performance of the proposed technique was measured based on the results. Standard images such as: (baboon, arctic hare, boat, cameraman, Lena) are all (8, 16, and 24) bits for color and grayscale images. Standard images are obtained from the available databases: (Image Database, 2018).

Choosing the optimal block size that achieves the best compression rates and less image distortion is necessary to enhance the algorithm performance. The researcher tested the algorithm using (2×2 , 4×4 , and 8×8 blocks) separately for the colored and grayscale image. The three-block size results were analyzed to find the suitable block size that achieves the best performance.

5.2 Results and discussion]

We used two standard metrics; Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). Since the proposed algorithm is a near-lossless technique that indicates the decompressed image should have a tiny distortion, this distortion should not be seen by the human visual system. As described before, the Huffman algorithm is a lossless technique, which indicates that the MSE should be zero, and the PSNR should be infinite.

We compared the proposed algorithm and the Huffman algorithm in image quality using the (MSE and PSNR). To better understand the results, we represent the MSE results by bar-charts, where each column represents the MSE value for each image. The best image quality is achieved by using the 2×2 block size. The Huffman algorithm's MSE values are zero for all the images since it is a lossless compression technique Table 1.

The MSE value for colored and grayscale images is small, as given in Table 2, which indicates that the proposed algorithm is a near-lossless technique, and the de-compressed images should have a low rate of distortion. Nevertheless, this distortion is not recognizable to be the human visual system. The best image quality is achieved by using the 2×2 block size.

Table 3 lists the PSNR results for the colored images by using different block size and compared against Huffman algorithm PSNR results. It is clear that the proposed method got better results compared with other methods. To better understand the results, we represent the PSNR results by bar-charts, where each column represents the PSNR value for each image. The best image quality is achieved by using the 2×2 block size. The Huffman algorithm's

Table 1 Lists the MSE results for the colored images by using different block size and compared against Huffman algorithm MSE results

Image name	Colored Image MSE			
	Proposed 2×2	Proposed 4×4	Proposed 8×8	Huffman
Image 1	0.38	0.47	0.49	0
Image 2	0.34	0.45	0.48	0
Image 3	0.37	0.46	0.48	0
Image 4	0.37	0.47	0.49	0
Image 5	0.36	0.45	0.46	0
Image 6	0.36	0.45	0.47	0
Image 7	0.37	0.47	0.49	0
Image 8	0.34	0.45	0.47	0
Image 9	0.22	0.32	0.37	0
Image 10	0.23	0.39	0.47	0
Image 11	0.28	0.38	0.40	0
Average	0.32	0.43	0.45	0

PSNR values are infinite for all the images since the PSNR is calculated by dividing the distorted value by MSE (MSE is zero in this case).

Table 4 represents the PSNR results for the grayscale images. The best image quality is achieved by using a 2×2 block size. The PSNR values for colored and grayscale images indicate that the proposed algorithm is the near-lossless technique. The de-compressed images should have a small distortion rate; nevertheless, this distortion is not recognizable by the human visual system. Moreover, the proposed algorithm got a promising performance compared to other well-known algorithms.

In this part, a comparison is conducted between the proposed algorithm and the Huffman algorithm in terms of image size, as shown in Table 5. The Compression Rate (CR) is measured to find the algorithm percentage of saving bits. The CR can be measured by dividing the output compressed image size by the input image size. The best CR is when the Cr results are close to zero; therefore, the compressed images should have a smaller size than the original.

Table 2 Lists the MSE results for the grayscale images by using different block size and compared against Huffman algorithm MSE results

Image name	Grayscale Image MSE			
	Proposed 2×2	Proposed 4×4	Proposed 8×8	Huffman
Image 1	0.22	0.36	0.43	0
Image 2	0.37	0.47	0.49	0
Image 3	0.24	0.35	0.40	0
Image 4	0.37	0.47	0.49	0
Image 5	0.38	0.47	0.49	0
Image 6	0.38	0.49	0.38	0
Image 7	0.38	0.47	0.49	0
Image 8	0.26	0.32	0.35	0
Image 9	0.38	0.47	0.49	0
Image 10	0.38	0.47	0.49	0
Image 11	0.30	0.39	0.42	0
Average	0.33	0.43	0.44	0

Table 3 Lists the PSNR results for the colored images for the proposed technique and Huffman algorithm

Image name	Colored Image PSNR			
	Proposed 2×2	Proposed 4×4	Proposed 8×8	Huffman
Image 1	51	51	50	Inf
Image 2	52	50	50	Inf
Image 3	52	51	50	Inf
Image 4	51	51	50	Inf
Image 5	52	51	51	Inf
Image 6	52	51	50	Inf
Image 7	51	50	50	Inf
Image 8	52	51	50	Inf
Image 9	53	52	51	Inf
Image 10	53	52	50	Inf
Image 11	52	51	51	Inf
Average	51	51	50	Inf

It can be seen from the above data, the (8 × 8) Block achieved the best compression rates among all the block sizes and saved 28% more than the Huffman algorithm.

It can be seen from the above Table 6, the (4 × 4) Block achieved the best compression rates among all the block size with 3% CR less than the (8 × 8) blocks and saved 19% more than Huffman algorithm. So, it reflects the ability of the proposed method in giving better results. Table 6 shows that the proposed algorithm enhances the Huffman storage-saving by 16% when using (8 × 8) blocks. Finally, it is clear in Table 7 that the proposed method when use two-by-two blocks is the best method, which is according to the Friedman ranking test. In Fig. 7, the average CR results are given for various techniques.

Figures 8, 9, 10, 11, 12, 13, 14, 15 shows several examples of original images and the decompressed images, for colored images and grayscale images using various compression technique blocks. These confirmed the performance of the proposed algorithm in comparison and decompressed images

Table 4 Displays PSNR results for grayscale images in the proposed technique and Huffman

Image name	Grayscale Image PSNR			
	Proposed 2×2	Proposed 4×4	Proposed 8×8	Huffman
Image 1	53	51	50	0
Image 2	51	50	50	0
Image 3	53	51	51	0
Image 4	51	50	50	0
Image 5	51	50	50	0
Image 6	51	50	51	0
Image 7	51	50	50	0
Image 8	53	52	51	0
Image 9	51	50	50	0
Image 10	51	50	50	0
Image 11	52	51	51	0
Average	51.6	50.5	50.2	0

Table 5 Lists the CR results for the colored images for the proposed technique by using different block size and Huffman algorithm

Image name	Colored Image CR			
	Proposed 2×2	Proposed 4×4	Proposed 8×8	Huffman
Image 1	0.94	0.48	0.43	0.61
Image 2	1.17	0.71	0.65	0.71
Image 3	0.88	0.6	0.54	0.9
Image 4	0.89	0.62	0.56	0.89
Image 5	0.81	0.55	0.51	0.78
Image 6	1.09	0.58	0.53	0.84
Image 7	1.01	0.52	0.46	0.8
Image 8	0.87	0.52	0.49	0.78
Image 9	0.89	0.4	0.36	0.44
Image 10	0.66	0.5	0.44	0.86
Image 11	1.15	0.46	0.34	0.57
Average	0.94	0.54	0.48	0.74

6 Conclusions and future work

The paper represents a new technique for compressing color and gray digital images to demonstrate that the Huffman compression rate increases when the image is divided into blocks of different sizes. They are applied to images of different types and sizes with different bit depths.

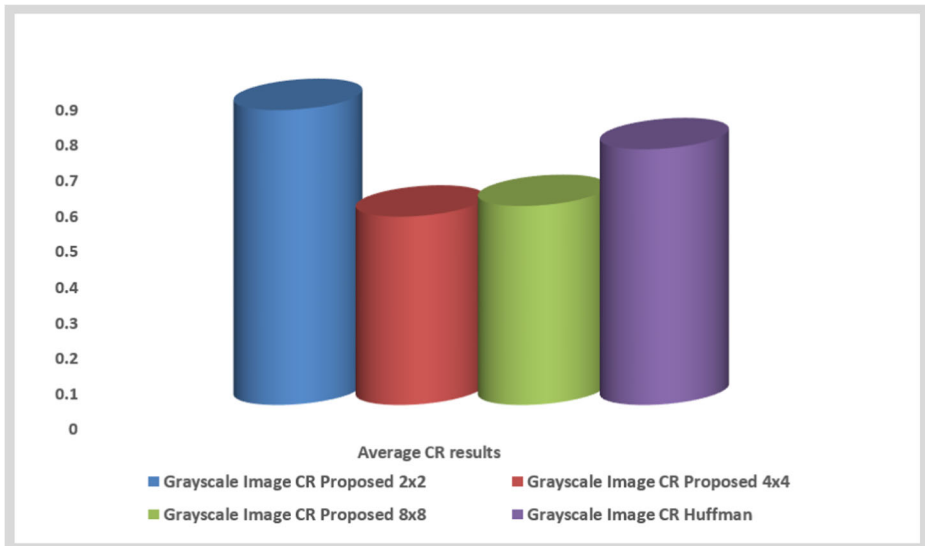
Huffman's lossless technique differs from the proposed technique in that the proposed technique is close to being lost. It is a new compression technology that performs some simple steps on the blocks before they are pressed into Huffman. Near-Lossless Image Compression Technique technology focuses on increasing the compression rate for Huffman lossless technology. After doing several experiments by compressing a set of gray and colored images onto Huffman, the same sets of images were pressed Near-Lossless Image Compression Technique. The results were compared with each other. The results showed an improvement in compression rates, a very slight loss of image resolution, and minimal deformation while

Table 6 Lists the CR results for the grayscale images for the proposed technique by using different block size and Huffman algorithm

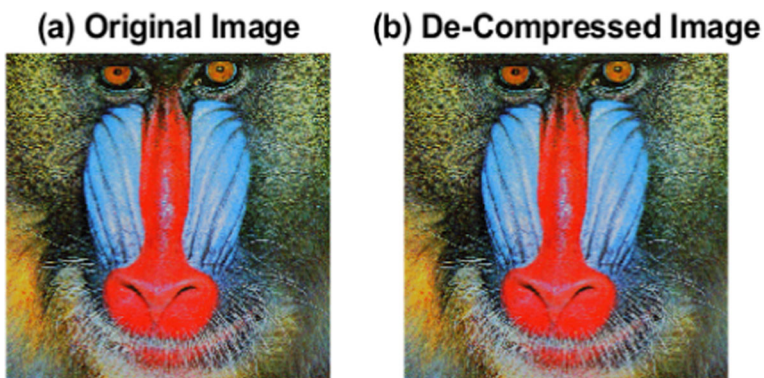
Image name	Grayscale Image CR			
	Proposed 2×2	Proposed 4×4	Proposed 8×8	Huffman
Image 1	0.77	0.49	0.50	0.46
Image 2	0.78	0.48	0.46	0.76
Image 3	0.8	0.5	0.49	0.58
Image 4	0.88	0.63	0.67	0.82
Image 5	0.82	0.55	0.54	0.88
Image 6	0.74	0.41	0.74	0.61
Image 7	0.79	0.44	0.52	0.72
Image 8	1.07	0.68	0.61	0.73
Image 9	0.8	0.53	0.52	0.85
Image 10	0.77	0.47	0.46	0.88
Image 11	0.94	0.7	0.70	0.65
Average	0.83	0.53	0.56	0.72

Table 7 The Friedman ranking test

Method	Proposed 2×2	Proposed 4×4	Proposed 8×8
MSE results for the colored images	1	2	3
MSE results for the greyscale images	1	2	3
PSNR results for the colored images	1	1	3
PSNR results for the greyscale images	1	2	3
CR results for the colored images	1	2	3
CR results for the greyscale images	1	3	2
Mean ranking	1	2	2.83
Final Ranking	1	2	3

**Fig. 7** The average CR results for various techniques

compressing (close) technology. For example: when comparing CR values between Block (2 * 2) of Near-Lossless Image Compression Technique, the Huffman Lossless Technology, (0.72–

**Fig. 8** Baboon colored image compressed on Huffman

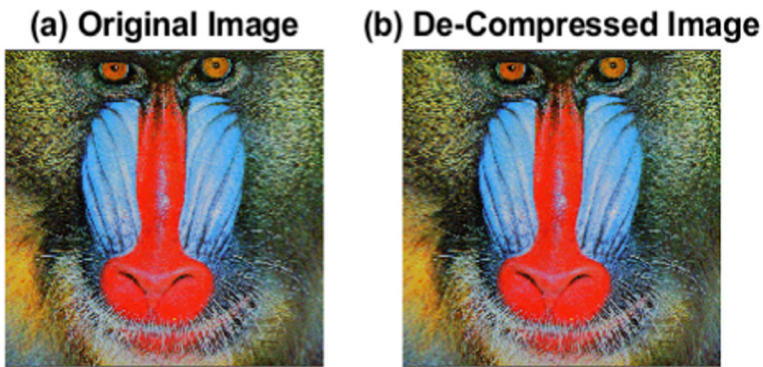


Fig. 9 Baboon color image compressed on Near-Lossless Image Compression Technique Block (2 * 2)

0.83) = -11% in gray, (0.74–0.94) = -20% when color pictures, this means that Block (2 * 2) cannot be pressed in Near-Lossless Image Compression Technique.

When comparing CR values between Block (4 * 4) of Near-Lossless Image Compression Technique, the Huffman Lossless Technology, (0.72–0.53) = 18% for grayscale images, (0.74–0.54) = 20% when color pictures, this means the compression results using a block (4 * 4) from the Near-Lossless Image Compression Technique are a bit good. When comparing CR values between Block (8 * 8) of Near-Lossless Image Compression Technique, the Huffman Lossless Technology, (0.72–0.56) = 16% when gray pictures, (0.74–0.48) = 26%, this means

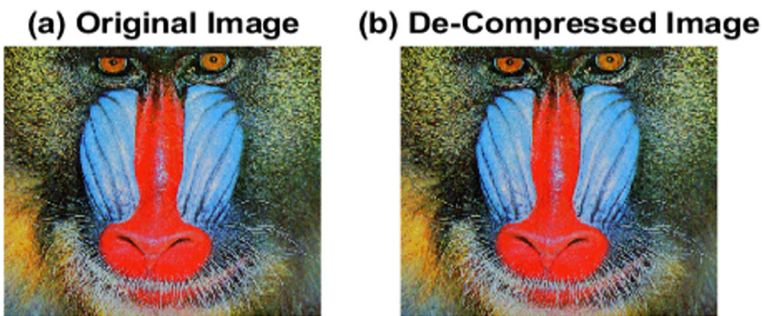


Fig. 10 Baboon color image compressed on Near-Lossless Image Compression Technique Block (4 * 4)

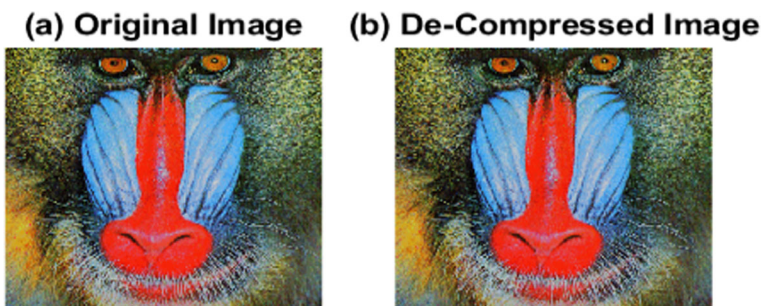


Fig. 11 Baboon color image compressed on Near-Lossless Image Compression Technique Block (8 * 8)

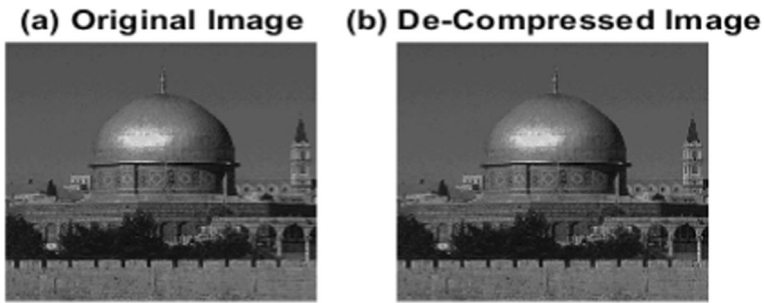


Fig. 12 Mosque grayscale image compressed on Huffman

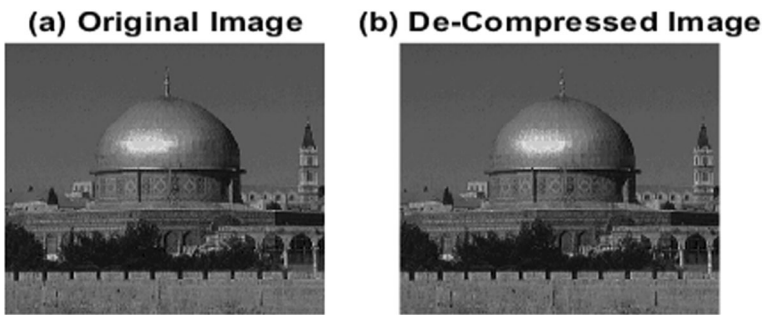


Fig. 13 Mosque grayscale compressed on Near-Lossless Image Compression Technique Block (2 * 2)

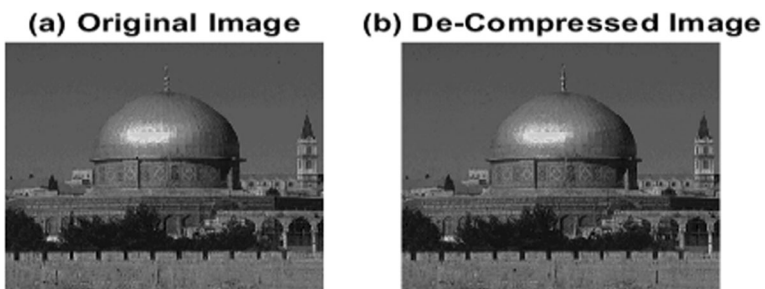


Fig. 14 Mosque grayscale compressed on Near-Lossless Image Compression Technique Block (4*4)

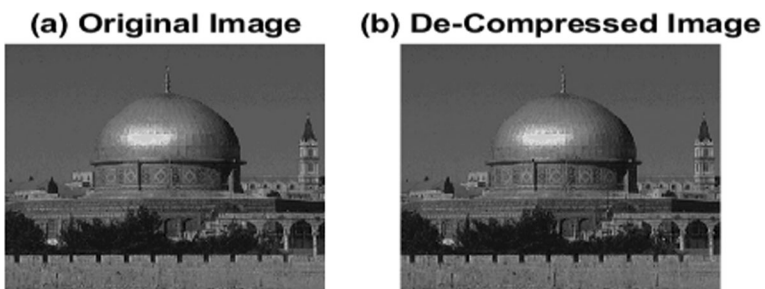


Fig. 15 Mosque grayscale compressed on Near-Lossless Image Compression Technique Block (8*8)

that the compression using a block ($8 * 8$) from Near-Lossless Image Compression Technique gives perfect results, especially on color pictures.

After performing this experiment, it is suggested to develop a digital compression science. Work to improve the stress rate on this technique by trying the same steps and applying other lossless technology, suggesting logical steps similar to this technique, and trying it on, Huffman. A new optimization algorithm can be adapted to solve these problems in the future.

Declarations

Conflict of interest There is no conflict of interest.

References

1. Abuowaida SFA et al. (2021) A novel instance segmentation algorithm based on improved deep learning algorithm for multi-object images. *Jordanian Journal of Computers and Information Technology (JJCIT)*, 7(01)
2. Aceves SM, Espinosa-Loza F, Ledesma-Orozco E, Ross TO, Weisberg AH, Brunner TC, Kircher O (2010) High-density automotive hydrogen storage with cryogenic capable pressure vessels. *Int J Hydrog Energy* 35(3):1219–1226
3. Agarwal R, Salimath C, Alam K (2019) Multiple image compression in medical imaging techniques using wavelets for speedy transmission and optimal storage. *Biomedical and Pharmacology Journal* 12(1):183–198
4. Aldemir E, Tohumoglu G, Selver MA (2019) Binary medical image compression using the volumetric run-length approach. *The Imaging Science Journal* 67(3):123–135
5. Alkhalayleh MA, Otair A (2015) A new lossless method of image compression by decomposing the tree of Huffman technique. *Int J Imaging Robot* 15(2):79–96
6. Al-Khasawneh MA et al (2021) An improved chaotic image encryption algorithm using Hadoop-based MapReduce framework for massive remote sensed images in parallel IoT applications. *Clust Comput* 25:1–15
7. Arándiga F, Mulet P, Renau V (2013) Lossless and near-lossless image compression based on multiresolution analysis. *J Comput Appl Math* 242:70–81
8. Ballé J et al. (2018) Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*
9. Chen Y, Xiao X, Zhou Y (2019) Low-rank quaternion approximation for color image processing. *IEEE Trans Image Process* 29:1426–1439
10. Cosman PC, Gray RM, Olshen RA (1994) Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. *Proc IEEE* 82(6):919–932
11. Dey N et al. (2020) Firefly algorithm and its variants in digital image processing: A comprehensive review, in *Applications of Firefly Algorithm and Its Variants*. Springer. p. 1–28
12. Dhou K (2020) A new chain coding mechanism for compression stimulated by a virtual environment of a predator–prey ecosystem. *Futur Gener Comput Syst* 102:650–669
13. Dhou K, Cruzen C (2019) An innovative chain coding technique for compression based on the concept of biological reproduction: an agent-based modeling approach. *IEEE Internet Things J* 6(6):9308–9315
14. Dhou K, Cruzen C (2021) A highly efficient chain code for compression using an agent-based modeling simulation of territories in biological beavers. *Futur Gener Comput Syst* 118:1–13
15. Diaz N, Hinojosa C, Arguello H (2019) Adaptive grayscale compressive spectral imaging using optimal blue noise coding patterns. *Opt Laser Technol* 117:147–157
16. Ewees AA, Abualigah L, Youstri D, Sahlol AT, al-qaness MAA, Alshathri S, Elaziz MA (2021) Modified artificial ecosystem-based optimization for multilevel thresholding image segmentation. *Mathematics* 9(19): 2363
17. Gong L, Qiu K, Deng C, Zhou N (2019) An image compression and encryption algorithm based on chaotic system and compressive sensing. *Opt Laser Technol* 115:257–267
18. Houssein EH, Hussain K, Abualigah L, Elaziz MA, Alomoush W, Dhiman G, Djenouri Y, Cuevas E (2021) An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. *Knowl-Based Syst* 229:107348

19. Huang L et al. (2020) OctSqueeze: Octree-Structured Entropy Model for LiDAR Compression. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition
20. Ibrahim M, Gbolagade K (2019) A Chinese Remainder Theorem based enhancements of Lempel-Ziv-Welch and Huffman coding image compression. *Asian Journal of Research in Computer Science*: 1–9
21. Jasmi RP, Perumal B, Rajasekaran MP (2015) Comparison of image compression techniques using huffman coding, DWT and fractal algorithm. In 2015 international conference on computer communication and informatics (ICCCI). IEEE
22. Kasban H, Hashima S (2019) Adaptive radiographic image compression technique using hierarchical vector quantization and Huffman encoding. *J Ambient Intell Humaniz Comput* 10(7):2855–2867
23. Kumar R, Jung K-H (2019) A systematic survey on block truncation coding based data hiding techniques. *Multimed Tools Appl* 78(22):32239–32259
24. Lee C-F et al. (2020) An improved lossless information hiding in SMVQ compressed images. in Proceedings of the 2020 The 6th International Conference on Frontiers of Educational Technologies
25. Lin S, Jia H, Abualigah L, Altalhi M (2021) Enhanced slime Mould algorithm for multilevel thresholding image segmentation using entropy measures. *Entropy* 23(12):1700
26. Liu Z et al. (2019) Machine vision guided 3d medical image compression for efficient transmission and accurate segmentation in the clouds. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
27. Ma S et al. (2019) Image and video compression with neural networks: a review. *IEEE Transactions on Circuits and Systems for Video Technology*
28. Mentzer F et al. (2019) Practical full resolution learned lossless image compression. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
29. Morales CN, Claire G, Álvarez J, Nanni A (2020) Evaluation of fiber content in GFRP bars using digital image processing. *Compos Part B* 200:108307
30. Otair M, Shehadeh F (2016) Lossy image compression by rounding the intensity followed by dividing (RIFD). *Res J Appl Sci Eng Technol* 12(6):680–685
31. Poolakkachalil TK, Chandran S (2019) Summative stereoscopic image compression using arithmetic coding. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)* 7(3):564–576
32. Rahman M, Hamada M (2019) Lossless image compression techniques: a state-of-the-art survey. *Symmetry* 11(10):1274
33. Rawat C, Meher S (2013) A hybrid image compression scheme using DCT and fractal image compression. *Int Arab J Inf Technol* 10(6):553–562
34. Rege S et al (2013) 2D geometric shape and color recognition using digital image processing. *International journal of advanced research in electrical, electronics and instrumentation engineering* 2(6):2479–2487
35. Santos L, Gómez A, Sarmiento R (2019) Implementation of CCSDS standards for lossless multispectral and hyperspectral satellite image compression. *IEEE Trans Aerosp Electron Syst* 56(2):1120–1138
36. Seeram E (2019) Digital image processing concepts, in *Digital Radiography*. p. 21–39.
37. Setia V, Kumar V (2012) Coding of DWT coefficients using run-length coding and Huffman coding for the purpose of color image compression. *International Journal of Computer and Communication Engineering* 6: 201–204
38. Shehab M, Daoud MS, AlMimi HM, Abualigah LM, Khader AT (2019) Hybridising cuckoo search algorithm for extracting the ODF maxima in spherical harmonic representation. *International Journal of Bio-Inspired Computation* 14(3):190–199
39. Simpson AL et al. (2019) A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*.
40. Sumari P, Syed SJ, Abualigah L (2021) A novel Deep learning pipeline architecture based on CNN to detect Covid-19 in chest X-ray images. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12(6):2001–2011
41. Talukder KH, Harada K (2010) Haar wavelet based approach for image compression and quality assessment of compressed image. *arXiv preprint arXiv:1010.4084*
42. Theis L et al. (2017) Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*
43. Touil DE, Terki N (2020) Optimized color space for image compression based on DCT and Bat algorithm. *Multimed Tools Appl* 80:1–21
44. Underwood R et al. (2020) FRaZ: A Generic High-Fidelity Fixed-Ratio Lossy Compression Framework for Scientific Floating-point Data. *arXiv preprint arXiv:2001.06139*
45. Wang A, Zhang W, Wei X (2019) A review on weed detection using ground-based machine vision and image processing techniques. *Comput Electron Agric* 158:226–240
46. Witten IH et al. (1999) *Managing gigabytes: compressing and indexing documents and images*: Morgan Kaufmann.

47. Yousri D, Abd Elaziz M, Abualigah L, Oliva D, al-qaness MAA, Ewees AA (2021) COVID-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions. *Appl Soft Comput* 101:107052

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.