



## Block-Wise Context-Adaptive Re-Indexing for Lossless Compression of Grayscale Medical Images

Hind J. Serteep<sup>1\*</sup>, Sally Ali Abdulateef<sup>1</sup>, Lamyaa Mohammed Kadhim<sup>2</sup>

<sup>1</sup> Department of Computer Science, College of Education, AL- Mustansiriyah University, Baghdad 10052, Iraq

<sup>2</sup> College of Dentistry, AL- Mustansiriyah University, Baghdad 10013, Iraq

Corresponding Author Email: [hind\\_2018@uomustansiriyah.edu.iq](mailto:hind_2018@uomustansiriyah.edu.iq)

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310523>

### ABSTRACT

**Received:** 11 March 2026

**Revised:** 30 April 2026

**Accepted:** 10 May 2026

**Available online:** 31 May 2026

#### Keywords:

*compression, grayscale images, context-adaptive coding, block-wise re-indexing, local pixel statistics, compression ratio, Digital Imaging and Communication in Medicine*

Lossless compression is essential for medical images because diagnostic information must be preserved after decompression, yet many conventional entropy coders rely mainly on global symbol statistics and may underuse local redundancy. This study proposes a block-wise context-adaptive re-indexing method for grayscale medical images. Each image is divided into non-overlapping  $B \times B$  blocks. For each block, local pixel frequencies are computed, pixel intensities are ranked by occurrence, and frequently occurring values are represented with shorter local indices. The number of bits required for each block is determined from the number of distinct local values, while block headers and mapping information are entropy-coded to support exact reconstruction. Experiments were performed on representative medical images and additional Digital Imaging and Communication in Medicine (DICOM) slices using  $4 \times 4$  and  $8 \times 8$  blocks. On the first test set,  $4 \times 4$  blocks achieved compression ratios from 2.07 to 3.74, whereas  $8 \times 8$  blocks produced ratios from 1.40 to 2.76; both settings exceeded standard Huffman coding, which ranged from 1.08 to 1.77. The DICOM evaluation confirmed lossless recovery, with Mean Squared Error (MSE) = 0 and Structural Similarity Index (SSIM) = 1 for all tested images. The results show that smaller blocks had better exploit localized intensity repetition, although they introduce higher header overhead. The proposed method offers a simple, reversible, and low-complexity alternative for grayscale medical image storage, with future work needed on larger clinical archives and diverse imaging modalities.

## 1. INTRODUCTION

These days, there are many different ways to deliver information. One of these is a digital image, which it speeds up information exchange through global networks and servers, where storage and transmission bandwidth consumption are significant and continuously rising [1]. To deal with such a Restrictions in storage and transmission bandwidth, computer researchers continue to study and create compression algorithms with the aim of storing as many images as feasible in a given storage area by lowering image file size [2]. The schemes of image compression lay in three categories; the first one is the lossy scheme, which comes with the advantage of a higher compression ratio, but once decompressed, the algorithm is unable to provide a perfect reconstruction of the original image with some data lost as long as tolerance is maintained [3]. Typically, lossy schemes compress natural images, multimedia data and applications. Like internet telephony and streaming media [4]. The lossless scheme is the second scheme that has a lower compression ratio than the lossy scheme, and it is principally used to compress medical images, where the original image must be preserved after decompression, and any loss of information could result in an incorrect diagnosis [5]. Using the biomedical image as an

example, we can determine the file size of the medical image obtained from microarray and microscopy.

The third one is the Hybrid scheme. It used to make a balance between compression efficiency and reconstruction quality by combining both lossy and lossless techniques; usually a lossless entropy stage is implemented after a lossy stage to further compress the converted data and reduce data redundancy. The majority of contemporary image compression standards uses this hybrid approach [6].

These images are used to analyze gene expression and sequence. The microarray experiment produces images with high resolution exceeding 13000-14000 pixels, with each image having a size of more than 200 MB, taking into account that the single pixel in these images has a pair of 16 bits, comparing these kinds of images with the ordinary images, which have pixels that are 8-bit in size and can be represented as numerical values ranging from 0 to 255 [7]. The image view is derived from the contributions of three matrices of pixels, which are the three basic colors that make up the image. Normally the file size of these images is from two to five MB [8].

One of the things criticized in classical lossless method (e.g., Joint Photographic Experts Group (JPEG)-LS, Huffman, and Run-Length-Encoding (RLE)) is their inability to capture

highly localized intensity changes, and they do not capture non-uniform pixel distribution, but they efficiently reduce global statistical redundancies inherent in grayscale medical images. While the modern medical imaging approaches concentrate on reducing data as much as possible without losing bit-level diagnostic integrity, to address these shortcomings, the proposed algorithm decentralizes a block instead of a single global histogram. Frequently occurring pixels are represented with minimized local indices. These steps allow the algorithm to avoid complex transform-domain architectures and focus extensively on local entropy. Offering a high-ratio, low-complexity substitute designed for certain medical image architectures [9].

## 2. IMAGE COMPRESSION APPROACHES

Many image compression algorithms have been developed by computer scientists depending on fundamental properties of images such as spatial correlation and statistical redundancy. In case of algorithms that rely on spatial correlation, they focus on neighboring pixels whose values are often similar or related like gradual color changes in images of Walls or skins [10].

On the other hand, when pixel value or patterns are repeated across over an Image this feature well known as statistical redundancy, which is made a Solid base for a compression Algorithm, as Like sky Image which it has many pixels have same shade of blue [11].

### 2.1 Spatial correlation image compression approaches

Here are a few image compression approaches that utilized spatial correlation.

#### 2.1.1 JPEG

It is a classic image compression technique; it is a standard found that was found by the Joint Photographic Experts Group, which classified as lossy compression technique, which removes unnecessary visual information to reduce image file size. The first step of the compression process is to transform the image color space from Red, Green, Blue (RGB) to YCbCr in order to split the brightness channel (Y) from the color channels (Cb, Cr) which enables the process to reduce color data (Chroma subsampling) that is less perceptible to the human eye. After that, the process involves Discrete Cosine Transform that change spatial domain into frequency domain components [12]. Discrete Cosine Transform (DCT) takes action by splitting the image into non-overlap blocks each of which is 8 by 8 pixels in size, the transformation focuses the signal energy into few significant coefficients to make it easier to discard less important data during compression [13], and the next important step is quantization. This is where the less important high-frequency coefficient is split up by the quantization matrix and bounded off to zero. This step is where most of the "loss" comes from and allows for a significant reduction in data. Finally, a lossless entropy method used to compress the remaining coefficients such as Huffman and arithmetic coding to make the image file size even smaller [14, 15].

#### 2.1.2 Wavelet-based compression

This technique is considered a sophisticated image compression technique where the image is broken down in to

different frequency components by employing a wavelet transform. It is well implemented by the JPEG 2000 standard.

The first step starts by processing the entire image at once in order to break down the image into groups of wavelet coefficients representing the high-frequency coefficient (fine lines, edges, and textures) and the low-frequency coefficient (the overall color and shape).

Secondly, the quantization step is where the actual compression happens at this lossy step, where the quantization sets the high-frequency coefficients to zero or their precision significantly decreased, while the Low-frequency coefficients are maintained to keep the image's essential features [16]. The final step is to make the image file even smaller by using a lossless compression method such as arithmetic coding to encode the quantized coefficient (where many of them are zero now) [17].

### 2.2 Statistical redundancy image compression approaches

Here are several methods for image compression that make use of statistical redundancy.

#### 2.2.1 Huffman coding

David Huffman created this compression algorithm. In 1952, LossLess compression algorithm was characterized by efficiency and simplicity. It uses for compress images and multimedia widely, where it gives the more frequent symbols shorter binary codes and the longer codes are given to less frequent symbols [18]. Many modern studies continue to improve the Huffman algorithm by integrating this algorithm with different techniques like adaptive coding and hybrid models. Singh et al proposed an enhanced adaptive Huffman coding algorithm to improve compression efficiency in real-time applications [19].

#### 2.2.2 Run length encoding

RLE it is considered a simple lossless compression technique that store a sequence of identical values with a single value and its duplicates. In the context of image compression, it uses a pair value to represent consecutive pixels of the same color [20]. This technique is suitable for images that have large areas of solid, uniform color that contain consecutive pixels or pixel values that are identical with small variation, such as black and white images, icons, and line drawings [21].

## 3. THE PROPOSED ALGORITHM

The representation of an image pixel in the least number of binary bits is the responsibility of the image compression algorithm, where "context-aware" is the examination of each non-overlapping pixel block independently to analyze local pixel frequency and distribution. Furthermore, the "pixel size reduction" is a process of lossless re-indexing of the original pixels depending on their local occurrence rank values, which is used to map them to new indices. This prevents loss of data by enabling the use of optimized bit length. In the current era, the deep learning and self-supervised learning approaches have taken a lot of attention in the field of medical image processing, for example, a modern algorithm used context-aware self-supervised learning combined with graph neural networks (GNNs) to analyze complex structural relation through medical data. Although these learned algorithms

present powerful representation capabilities. They add more computational complexity unlike the localized block-based statistical approaches [22]. The proposed algorithm's construction was accurately designed for single-channel grayscale medical images only, and this requirement was largely confirmed. The algorithm's various steps were carried out directly in raw intensity matrices, including block partitioning, frequency ranking, and context-aware re-indexing operations, where the algorithm introduces no pre-processing color-space overhead. Beside that the statistical redundancy reduction was carried out only on native clinical grayscale data.

### 3.1 The compression steps

The operational steps of the suggested algorithm are outlined below.

Input: Grayscale image size  $m \times n$ .

Output: Compressed file with optimized pixel data and headers.

Step #1: Initialize

- Divide the image into non-overlapping blocks of pixels with a size of  $B \times B$ .
- Create a global pixel occurrence table (for the entire image) which is function as master reference index, in order to ensure uniform re-mapping throughout all blocks during both of compression and decompression, this table record the frequency of all unique pixel values and serves as fixed lookup key.

Step #2: Process each block independently

For each block in image

- 2.1 Calculate local occurrence frequency

Initialize local occurrence table.

For each pixel in the block.

Update local occurrence table [Pixel value].

- 2.2 Revalue pixels based on local frequency

Sort the local table by frequency descending.

Map each pixel in the block to its local revalued index.

For each pixel in the block.

Original value = pixel value.

Local rank = position of original value in sorted pixels.

Assign a new value based on local rank.

Pixel value revalued = local rank.

- 2.3 Calculate minimal bits needed for this block

Max frequency rank = Max (local occurrence table ranks).

Bits needed =  $\text{ceil} [\log_2 (\text{number of unique local pixel values})]$ .

Store bits needed as header information for this block.

Step #3: Store headers

- Create a header for each block

Block position (row, column)

Bits needed

Possibly, a local occurrence table or pattern index if needed

Step #4: Compress header

- Compress all headers using Lempel-Ziv-Welch (LZW) or any adaptive entropy encoder

Step #5: Write pixel data

For each block

For each pixel

Write the pixel value reevaluated in bits needed into the output stream.

End for

End for

Step #6: Final compression

Compress header and pixel data together as needed

Figure 1 visually illustrates the detailed sequential execution of the proposed algorithm.

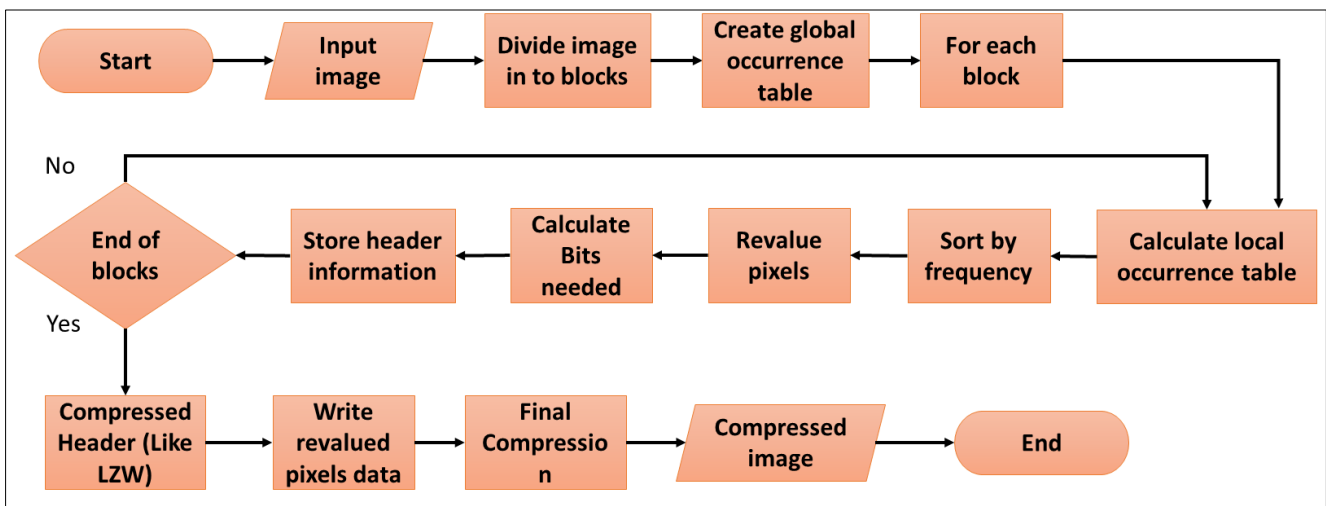


Figure 1. The flowchart of the compression algorithm

### 3.2 The decompression steps

In order to retrieve the original digital image back, the following decompression algorithm must be utilized.

Input: Compressed file containing compressed headers (with block info) and revalued pixel data (bit stream).

Output: Reconstructed grayscale medical image of size  $m \times n$ .

Step #1: Initialize

- Read the image dimensions  $(m, n)$  and block size  $B \times B$  from the main header.

- Initialize an empty image array  $\text{Image}[m][n]$  for single channel.

- Prepare to reconstruct each block independently.

Step #2: Decode Headers

- Decompress the header section using the same entropy decoding method (e.g., LZW).

- For each block header, extract: Block position (row,

column), bits needed, local occurrence table or reference pattern index (if stored).

- Store these header values in a list named (Block\_Header\_List).

Step #3: Read and Decode Block Data

For each block (in order from Block\_Header\_List):

- Extract block pixel data: Read  $B \times B$  pixel values from the bit stream using bits\_needed bits per pixel.
- Reconstruct local pixel ranks: The read values represent local ranks (from most to least frequent pixel).
- Rebuild the local occurrence table, there is for every distinct pixel value in the image, a global occurrence table that acts as a master reference index. The global table serves as the lookup key to map local ranks back to their initial intensity levels during decompression if just rank information is saved for the block. Ensuring consistency across all blocks without redundantly storing a full mapping table for each block.
- If only rank information was stored, reconstruct the mapping from ranks to original pixel values using the

global occurrence table or stored reference pattern.

- Map ranks back to original pixel values.

For each pixel in the block:

- revalued\_rank = read pixel value.
- original\_pixel\_value = local\_table[revalued\_rank].
- block\_pixel = original\_pixel\_value.
- Rebuild the block with the original pixel values.

Step #4: Place Block in Image

- Using the block's (row, column) position, insert the reconstructed block into the corresponding region of the output image:
- Image[row\*B: (row+1)\*B, column\*B : (column+1)\*B] = Reconstructed\_Block.

Step #5: Final Reconstruction

- Repeat the above steps for all blocks until the entire image is reconstructed.
- Merge all blocks to get the fully decompressed grayscale image.

Step #6: Output

- Save or display the reconstructed image.

Figure 2 shows the flowchart of the decompression algorithm.

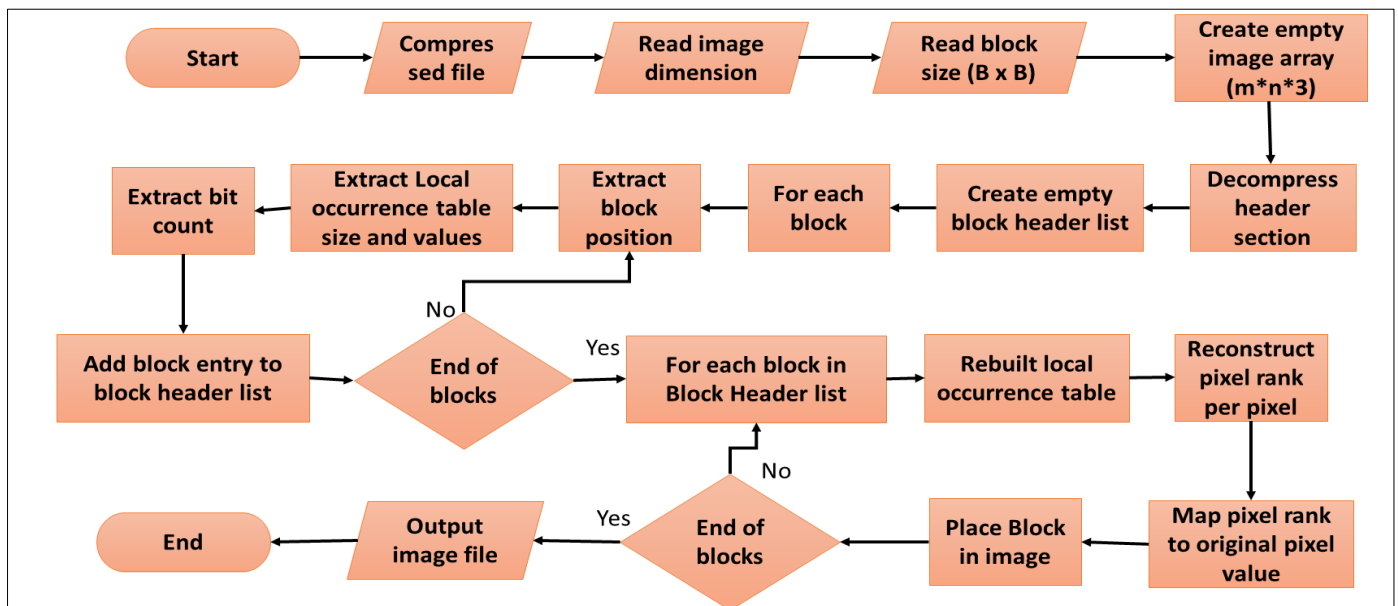


Figure 2. The flow chart of the decompression algorithm

Table 1. Compressed file structure and data components

Data Component	Description	Data Type / Unit
Main Header	Image dimensions ( $m \times n$ ) and Block size ( $B \times B$ )	Metadata
Global Table	Frequency of all unique pixels in the entire image	Reference Array
Block Headers	Bit-count and Local Mapping info for each block	Compressed (LZW)
Pixel Data	Re-indexed values based on local rank	Variable Bit-stream

Table 1 explains the data components and file structure, the global pixel distribution and a compressed header with each block's local mapping details and optimized bit-length are stored in the compressed file to guarantee lossless reconstruction.

#### 4. RESULT AND DISCUSSION

The effectiveness of the proposed algorithm is evaluated by applying it to several different medical images (see Figure 3).

As a lossless compression algorithm, the original image is identical to the decoded compressed image, and no degradation evaluation is required. A compression ratio was adopted as a performance metric, the compression ratio is defined as the ratio of the original image size to the compressed image size. For that, the only performance metric used to evaluate the algorithm is the compression ratio. The performance evaluation process also takes into account the use of different pixel block sizes ( $8 \times 8$  and  $4 \times 4$ ); see the Tables 2 and 3.

Tables 2 and 3 reveals information about the original image size and compressed image size utilizing two pixels block sizes ( $8 \times 8$  and  $4 \times 4$ ); these information can be summarized as follows.

1- High compression efficiency with  $4 \times 4$  blocks

Focusing on the compression ratio, the  $(4 \times 4)$  block consistently outperforms the  $(8 \times 8)$  block, which gives an indication that the  $(4 \times 4)$  block provides stronger compression, and this algorithm is pixel block size sensitive. When converting from an  $(8 \times 8)$  to a  $(4 \times 4)$  block, the compression ratio commonly improves by (0.7 to 1.1).

2- Most compressible and least compressible images

The most compressible image in the dataset is the “knee Magnetic Resonance Imaging (MRI) 2” image which has the highest compression ratio for both block sizes (2.757 for  $8 \times 8$  and 3.7434 for  $4 \times 4$ ).

While the “Cervical Spine” image has the lowest compression ratio (1.4037 for  $8 \times 8$  and 2.068 for  $4 \times 4$ ) suggesting that the image has more complex structures and is harder to compress effectively.

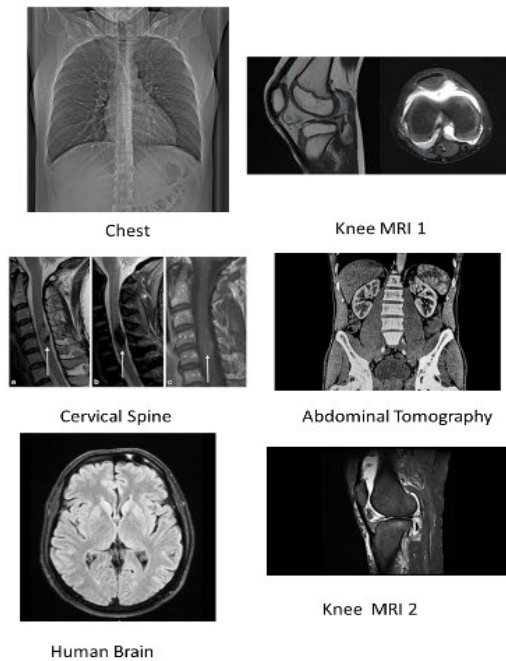


Figure 3. Dataset medical images

Table 2. Compression ratio  $(4 \times 4)$  pixel block

Image Name	Original Image Size	Compressed Image Size	Compression Ratio
Chest	2852640	1271248	2.244
Knee MRI 1	401856	141632	2.8373
Abdominal tomography	402600	138560	2.9056
Cervical Spine	402288	194528	2.068
Human Brain	2903328	889824	3.2628
Knee MRI 2	402192	107440	3.7434

Table 3. Compression ratio  $(8 \times 8)$  pixel block

Image Name	Original Image Size	Compressed Image Size	Compression Ratio
Chest	2852640	1716160	1.6622
Knee MRI 1	401856	203392	1.9758
Abdominal tomography	402600	197504	2.0384
Cervical Spine	402288	286592	1.4037
Human Brain	2903328	1225728	2.3687
Knee MRI 2	402192	145856	2.7575

Table 4. Compression ratio at  $(8 \times 8)$  and  $(4 \times 4)$  pixel blocks and Huffman compression technique

Image Name	Proposed Algorithm		Huffman
	compression ratio $(8 \times 8)$	compression ratio $(4 \times 4)$	compression ratio
Chest	1.6622	2.244	1.1635
Knee MRI 1	1.9758	2.8373	1.2208
Abdominal tomography	2.0384	2.9056	1.4964
Cervical Spine	1.4037	2.068	1.082
Human Brain	2.3687	3.2628	1.4026
Knee MRI 2	2.7575	3.7434	1.7747

Note: MRI = Magnetic Resonance Imaging.

Table 4 shows the values of the compression ratio for the proposed algorithm (using two-block sizes  $4 \times 4$  and  $8 \times 8$ ) and the Huffman compression technique. There are a number of distinct trends to note.

For all the images of the dataset, the compression ratio produced by the Huffman technique is significantly lower than the compression ratio produced by the proposed algorithm, whether using  $8 \times 8$  or  $4 \times 4$  blocks. That indicates that the proposed technique achieves more efficient size reduction than standard Huffman coding.

#### 4.1 Complexity and overhead analysis

As the suggested algorithm test its performance in block sizes of  $(4 \times 4)$  and  $(8 \times 8)$ , and the  $(4 \times 4)$  block configuration requires four times of header as  $(8 \times 8)$  headers needed, that leads to higher header overhead as a result of the number of blocks per image compared to  $(8 \times 8)$  configuration. Furthermore, as the suggested re-indexing approach depend on simple frequency sorting instead of complex iterative transforms; it is classified as computationally efficient. The confirming lossless recovery was verified for all test images using the MD5 (pixel-by-pixel difference) between the original and the decompression images.

#### 4.2 Comparing with JPEG and JPEG2000

For expanding the performance evaluation of the suggested algorithm, the comparison was made with the JPEG and JPEG2000 standards (as shown in Table 5). For ensuring statistical consistency, these baselines were create using the same set of medical images. The number in Table 5 shows a clear advantage of the proposed block-based re-indexing in most cases overcoming the conventional frequency – based transforms, especially in capturing localized statistics of MRI scans.

Table 5. Comparison with JPEG and JPEG2000

Image Name	Proposed $(4 \times 4)$	Huffman	JPEG (Q = 70)	JPEG2000 (CR = 2.5)
Chest	2.244	1.163	2.555	2.5
Knee MRI 1	2.837	1.22	1.85	2.5
Cervical Spine	2.068	1.082	1.843	2.5
Human Brain	3.262	1.402	1.209	2.5
Knee MRI 2	3.743	1.774	0.809	2.5

### 4.3 Evaluate the proposed algorithm on standard clinical Digital Imaging and Communication in Medicine medical modalities

The proposed algorithm has been subject to another examination to further validate the empirical resilience compressing ten images from the Digital Imaging and Communication in Medicine (DICOM) dataset using the following evaluation metrics: Compression Ratio (C.R.), Bits Per Pixel (bpp), Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index (SSIM).

Below Figure 4 presents a set of medical images taken from DICOM database for evaluate the proposed algorithm and Tables 5 and 6 demonstrate the test result using both of  $(4 \times 4)$  and  $(8 \times 8)$  block sizes.

The experimental data show the block size is an architectural trade-off inherently. That it relates to compression performance where it shows stability and Tables

6 and 7 robustness at a block size of  $(8 \times 8)$  where the CR has range of  $(2.07 - 2.20)$  and improves in bit-rate value  $(3.63$  to  $3.87)$  bpp across most of the clinical images. On the other hand, when swapping to a block size  $(4 \times 4)$  an improvement in the CR values was notice. Where its range is  $2.80$  to  $3.12$ , and the bit rate show a promising value where it's getting down to range of  $2.57$  to  $2.85$  bpp.

The above algorithm facts are highly justified as the local statistical correlation are vastly maximized within the pixel neighborhoods for smaller block dimensions. Leading to minimizing the number of unique intensity values, which dynamically optimizes the header representation and permits tighter re-indexing. The most important thing to be highlighted throughout all the recorded values is that the evaluation metrics achieved  $MSE = 0$ ,  $SSIM = 1$  and an infinite value for PSNR. These values demonstrate the algorithm lossless data integrity of the context-aware mechanism, which is essential for accurate medical diagnostics.

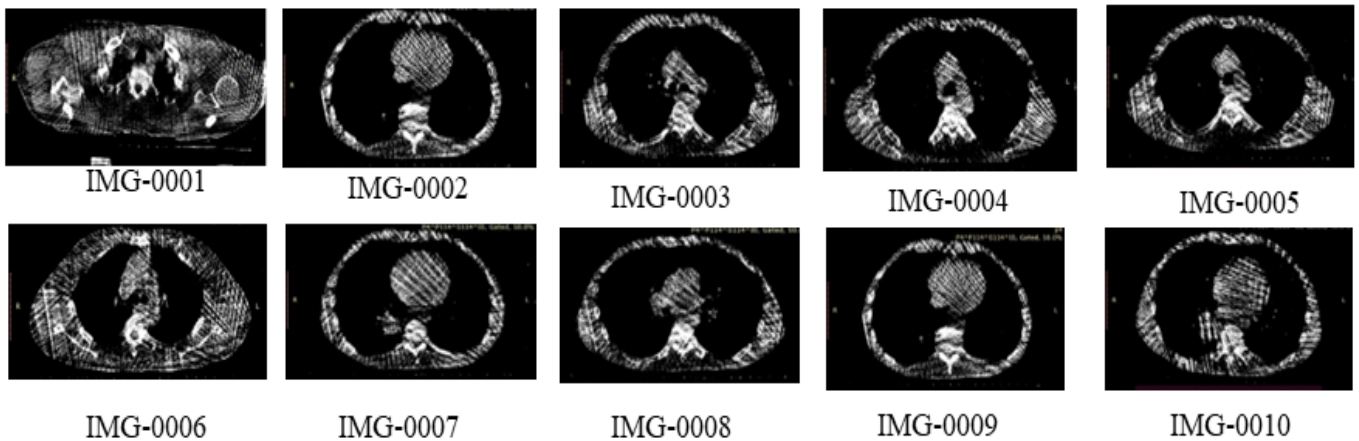


Figure 4. A set of Digital Imaging and Communication in Medicine (DICOM) dataset medical images

Table 6. Evaluation metrics of Digital Imaging and Communication in Medicine (DICOM) image data base at block size  $(4 \times 4)$

Image Name	Original Size (Bits)	Compressed Size (Bits)	CR	bpp	MSE	PSNR	SSIM
IMG-0001	2097152	860208	2.437959191	3.281433105	0	65535	1
IMG-0002	2097152	925448	2.266093827	3.530303955	0	65535	1
IMG-0003	2097152	892944	2.34858177	3.406311035	0	65535	1
IMG-0004	2097152	883648	2.373288911	3.370849609	0	65535	1
IMG-0005	2097152	891816	2.351552338	3.402008057	0	65535	1
IMG-0006	2097152	885496	2.368335938	3.37789917	0	65535	1
IMG-0007	2097152	930800	2.253064031	3.550720215	0	65535	1
IMG-0008	2097152	893112	2.348139987	3.406951904	0	65535	1
IMG-0009	2097152	1103976	1.899635499	4.211334229	0	65535	1
IMG-0010	2097152	1236840	1.695572588	4.718170166	0	65535	1

Note: CR = Compression Ratio; bpp = Bits Per Pixel; MSE = Mean Squared Error; PSNR = Peak Signal to Noise Ratio; SSIM = Structural Similarity Index.

Table 7. Evaluation metrics of Digital Imaging and Communication in Medicine (DICOM) image data base at block size  $(8 \times 8)$

Image Name	Original Size (Bits)	Compressed Size (Bits)	CR	bpp	MSE	PSNR	SSIM
IMG-0001	2097152	860208	2.437959191	3.281433105	0	65535	1
IMG-0002	2097152	925448	2.266093827	3.530303955	0	65535	1
IMG-0003	2097152	892944	2.34858177	3.406311035	0	65535	1
IMG-0004	2097152	883648	2.373288911	3.370849609	0	65535	1
IMG-0005	2097152	891816	2.351552338	3.402008057	0	65535	1
IMG-0006	2097152	885496	2.368335938	3.37789917	0	65535	1
IMG-0007	2097152	930800	2.253064031	3.550720215	0	65535	1
IMG-0008	2097152	893112	2.348139987	3.406951904	0	65535	1
IMG-0009	2097152	1103976	1.899635499	4.211334229	0	65535	1
IMG-0010	2097152	1236840	1.695572588	4.718170166	0	65535	1

Note: CR = Compression Ratio; bpp = Bits Per Pixel; MSE = Mean Squared Error; PSNR = Peak Signal to Noise Ratio; SSIM = Structural Similarity Index.

#### 4.4 Clinical workflow and real-world deployment (Picture Archiving, and Communication Systems and telemedicine)

Modern healthcare information systems, including Hospital Information System (HIS), Picture Archiving, and Communication Systems (PACS), where their neutral work is to manage and create a huge number of medical imaging data such as DICOM files, to handle such a task they rely on two essential requirements: high transmission bandwidth and large storage capacity. The integration of the suggested algorithm into these medical imaging information systems in smoothly and continuously way. Algorithm structure depend on the Global Occurrence Table with optimized header design in the course of telemedicine data transmission, the communication overhead is reduced by the use of a compressed bitstream as a limited bandwidth existed commonly. Additionally, the enabling of rapid, localized, chunk-based analyzing depends on small block structures like  $(4 \times 4)$  and  $(8 \times 8)$ . Within modern healthcare information systems, these capabilities play vital role in supporting efficient real-time clinical workflows. Allowing for quick random access to particular regions of interest (ROI) and progressive image rendering without need to decompress the whole medical volume. Consequently, a lossless reconstruction of the compressed image is guaranteed by low-complexity header deployment while essential diagnostic details are maintained for accurate medical decision.

#### 4.5 Statistical and mechanism analysis of block size performance

The analyzing the balance between header overhead and local entropy is considered essential to understand the reason for why the  $4 \times 4$  block overcomes the  $8 \times 8$  block. Grayscale medical image characterized (from a mathematical view) with high spatial correlation within ultra-localized neighborhoods. To obtain (per block) fewer unique values and local frequency concentration, it must minimize local pixel variance by reducing block size to  $4 \times 4$ . This effectively lowers the required bit depth for pixel values. Despite choosing a  $4 \times 4$  block size, which it increases the number of blocks four times that of  $8 \times 8$ , which increases total header overhead. This cost can be compensated by the massive bit saving achieved by encoding the highly compressed local pixel arrays. It is also worth noting that this approach can be applied for homogeneous medical textures in a perfect way, it faces a difficulty in highly chaotic natural images. Consequently, and this local-rank method is specifically tailored for medical dataset.

### 5. CONCLUSION

The suggested context-aware, block-based size reduction algorithm shows promising compression performance on tested medical images. According to experimental data, the proposed re-indexing process consistently outperforms regular Huffman coding in taking advantage of local statistical redundancy.

Especially the algorithm performance on the standard DICOM dataset show a clear architectural trade-off based on block dimensions. where a block size of  $8 \times 8$  has compression ratio range of 2.06 to 2.20 with bit-rate range of (3.62 to 3.86) while the  $(4 \times 4)$  block size achievement is significantly

enhanced the performance by have the range of compression ratio range of (2.80 to 3.12) and lowering the bit-rate to (2.56 to 2.85) bpp.

Regardless of the block configuration. The perfect reconstruction of the compressed image is confirmed by the mathematical metrics, by having the  $MSE = 0$  and  $SSIM = 1$  for all the candidate test images.

Within the evaluated validation scope, the algorithm demonstrated encouraging stability. While these definitive findings suggest that the method is well-suited for high-fidelity grayscale medical image storage, future work will focus on broader validation across larger clinical archives to fully establish its generalizability.

### ACKNOWLEDGMENT

The authors would like to express their gratitude to editor and reviewers, as they provide great guidance to improve this manuscript.

### CONFLICT OF INTEREST

Regarding the publishing of this manuscript, the authors disclose no conflicts of interest.

### REFERENCES

- [1] George, L.E., Sultan, B.A. (2014). Image compression based on wavelet, polynomial and quadtree. *Journal of Applied Computer Science & Mathematics*, 5(2): 15-20. [https://www.jacsm.ro/view/?pid=11\\_2](https://www.jacsm.ro/view/?pid=11_2).
- [2] Kaur, K., Chopra, V., Kaur, H. (2012). Image compression of medical image using VQ - Huffman coding technique. *International Journal of Research in Business and Technology*, 1(1): 36-43. <https://doi.org/10.17722/ijrbt.v1i1.6>
- [3] Chen, Y.Y., Tai, S.C. (2004). Embedded medical image compression using DCT based sub band decomposition and modified SPIHT data organization. In *Proceeding of the Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04)*, Taichung, Taiwan, pp. 1-8. <https://doi.org/10.1109/BIBE.2004.1317339>
- [4] Gosai, P. (2017). A survey: Various techniques of lossy image compression. *International Journal of Novel Research and Development*, 2(4): 58-59.
- [5] Khafaji, C.A., George, L.E. (2013). Fast loss less compression of medical images based on polynomial. *International Journal of Computer Application*, 70(15): 23-32. <https://doi.org/10.5120/12039-7999>
- [6] Starosolski, R. (2020). Hybrid adaptive lossless image compression based on discrete wavelet transform. *Entropy*, 22(7): 751. <https://doi.org/10.3390/e22070751>
- [7] Matos, L.M.O., Neves, A.J.R., Pinho, A.J. (2015). Lossy-to-lossless compression of biomedical images based on image decomposition. In *Application of Digital Signal Processing through Practical Approach*. <https://doi.org/10.5772/60650>
- [8] Shantagiri, P.V., Saravanan, K.N. (2013). Pixel size reduction loss-less image compression algorithm. *International Journal of Computer Science & Information Technology*, 5(2): 87-95.

- <https://doi.org/10.5121/ijcsit.2013.5208>
- [9] Al-Khafaji, G.K., Aal-Hassani, U.S., Yahya, R.I. (2024). Lossless medical image compression using 1D/2D liner polynomial coding techniques. In 2024 International Conference on High Performance Computing and Smart System (HORA), Istanbul, Turkiye, pp. 1-4. <https://doi.org/10.1109/HORA61326.2024.10550510>
- [10] Wang, H., Fu, N., Chen, Z., Liu, S. (2023). Dynamic kernel-based adaptive spatial aggregation for learned image compression. arXiv:2308.08723. <https://doi.org/10.48550/arXiv.2308.08723>
- [11] Li, S., Dai, W., Fang, Y., Zheng, Z., Fei, W., Xiong, H., Zhang, W. (2024). Revisiting learned image compression with statistical measurement of latent representations. IEEE Transactions on Circuits and Systems for Video Technology, 34(4): 2891-2907. <https://doi.org/10.1109/TCSVT.2023.3300316>
- [12] Raid, A.M., Khedr, W.M., El-Dosuky, M.A., Ahmed, W. (2014). JPEG image compression using discrete cosine transform - A survey. International Journal of Computer Science & Engineering Survey, 5(2): 39-47. <https://doi.org/10.5121/ijcses.2014.5204>
- [13] Nivedita, Singh, P., Jindal, S. (2012). A comparative study of DCT and DWT-SPIHT. IJCEM International Journal of Computational Engineering and Management, 15(2): 26-32.
- [14] Brisam, A.A., Mosa, Q.O. (2021). Compression technique for the JPEG image standard by using compression algorithm. Journal of Al-Qadisiyah for Computer Science and Mathematics, 13(2): 1-10. <https://doi.org/10.29304/jqcm.2021.13.2.787>
- [15] Min, Q., Wang, X., Huang, B., Zhou, Z. (2022). Lossless medical image compression based on anatomical information and deep neural network. Biomedical Signal Processing and Control, 74: 103885. <https://doi.org/10.1016/j.bspc.2022.103499>
- [16] Zhang, Y., Jiang, J., Zhang, G. (2021). Compression of remotely sensed astronomical image using wavelet-based compressed sensing in deep space exploration. Remote Sensing, 13(2): 1-16. <https://doi.org/10.3390/rs13020288>
- [17] Lafif, I.H., Abdulredha, S.H., Hassan, S.K. (2024). Discrete wavelet transform based image processing: A review. Al-Nahrain Journal of Science, 7(3): 109-125. <https://doi.org/10.22401/ANJS.27.3.13>
- [18] Mathur, M.K., Loonker, S., Saxena, D. (2012). Lossless Huffman coding technique for image compression and reconstruction using binary tree. International Journal of Computer Technology and Application, 3(1): 76-79.
- [19] Singh, S., Singh, H. (2011). Improved adaptive Huffman compression algorithm. IJCT International Journal of Computer and Technology, 1(1): 16-22. <https://doi.org/10.24297/ijct.v1i1.2602>
- [20] Al-Shweiki, J., Handan, N., Al Kaabneh, K. (2019). Comparative study between different image compression algorithm. International Journal of Science and Applied Information Technology, 8(6): 49-53. <https://doi.org/10.30534/ijcsait/2019/06862019>
- [21] Birajdar, A., Agarwal, H., Bolia, M., Gupte, V. (2019). Image compression using run length encoding and its optimisation. In 2019 Global Conference for Advancement in Technology (GCAT), Bangalore, India, pp. 1-6. <https://doi.org/10.1109/GCAT47503.2019.8978464>
- [22] Sun, L., Yu, K., Batmanghelich, K. (2022). Context-aware self-supervised learning for medical images using graph neural network. arXiv:2207.02957. <https://doi.org/10.48550/arXiv.2207.02957>

## NOMENCLATURE

CR	Compression Ratio
bpp	Bits Per Pixel
MSE	Mean Squared Error
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity Index
RLE	Run - Length -Encoding
RGB	Red, Green, Blue
DCT	Discrete Cosine Transform
Y	Luminance / Luma
Cb	Chrominance Blue
Cr	Chrominance Red
GNN	Graph Neural Network
JPEG	Joint Photographic Experts Group
LZW	Lempel-Ziv-Welch
MRI	Magnetic Resonance Imaging