6

P. Sreenivasulu\* and S. Varadarajan

# An Efficient Lossless ROI Image Compression Using Wavelet-Based Modified Region Growing Algorithm

https://doi.org/10.1515/jisys-2018-0180
Received April 13, 2018; previously published online November 14, 2018.

**Abstract:** Nowadays, medical imaging and telemedicine are increasingly being utilized on a huge scale. The expanding interest in storing and sending medical images brings a lack of adequate memory spaces and transmission bandwidth. To resolve these issues, compression was introduced. The main aim of lossless image compression is to improve accuracy, reduce the bit rate, and improve the compression efficiency for the storage and transmission of medical images while maintaining an acceptable image quality for diagnosis purposes. In this paper, we propose lossless medical image compression using wavelet transform and encoding method. Basically, the proposed image compression system comprises three modules: (i) segmentation, (ii) image compression, and (iii) image decompression. First, the input medical image is segmented into region of interest (ROI) and non-ROI using a modified region growing algorithm. Subsequently, the ROI is compressed by discrete cosine transform and set partitioning in hierarchical tree encoding method, and the non-ROI is compressed by discrete wavelet transform and merging-based Huffman encoding method. Finally, the compressed image combination of the compressed ROI and non-ROI is obtained. Then, in the decompression stage, the original medical image is extracted using the reverse procedure. The experimentation was carried out using different medical images, and the proposed method obtained better results compared to different other methods.

**Keywords:** Medical image, region of interest, modified region growing, merging-based Huffman encoding, SPIHT, DWT, DCT, non-ROI.

# 1 Introduction

Medical imaging has a great impact on the diagnosis, recognition, and surgical planning of diseases. Medical images contain a huge amount of data that help doctors in analyzing the condition efficiently and planning the diagnosis for patients. The storage of these medical images is a crucial task for hospitals because of storage requirements. These images are stored in digital form, which is easier to analyze. DICOM images are one of the examples of medical captured digital images [9]. Similarly, the process of medical diagnosis produces a huge amount of medical images such as those from computed tomography, magnetic resonance imaging (MRI), and electrocardiogram [20]. As these images take more space and bandwidth, their compression [11] is important for transmission. Image compression is a process of efficiently coding digital [1] images to reduce the number of bits required in representing an image. Due to the compression, images lose some information, which causes risk during treatment or diagnosis; thus, designing an efficient algorithm for compression and reconstruction is important to preserve image quality and reduce the computational time for transmission. Based on the loss of the data, image compression techniques are classified into two major categories: (i) lossless image compression and (ii) lossy compression [18]. Lossy techniques are used when loss can be

<sup>\*</sup>Corresponding author: P. Sreenivasulu, Research Scholar, Sri Venkateswara University, Tirupati, Andhra Pradesh, India, e-mail: psreenivasulu0682@gmail.com

S. Varadarajan: Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, Andhra Pradesh, India

accepted, and lossless techniques are used for applications that cannot afford any loss of information (e.g. in the medical field) [22].

Image modeling and coding are two separate phases conducted using the lossless method. In the modeling phase, a model is designed from the afforded image and applied for encoding. In the coding phase, statistical analysis is executed to find the symbols of the afforded image. Data loss is much less in the lossless compression strategy when equated to lossy compression due to the quantization nature of the strategy. Fractal coding-based image compression can be lossless or lossy compression; it is applied for the removal of redundancy from the original data after compression. The lossless compression scheme is a promising technique to save huge medical data for medical imaging systems [15]. To signify the image data in a compressed manner, an image compression system contains an encoder that uses the redundancies while the decoder is applied to rebuild the original image from the compressed data [12]. The digital image is usually characterized by three types of redundancies: psycho-visual redundancy, spatial redundancy, and coding redundancy [4]. The compression algorithms utilize these redundancies to compress the image.

The frequently utilized image compression method for medical images is IPEG, which unites discrete cosine transform (DCT) with Huffman coding in the earlier two decades. Due to the requirements for enhancing the visual quality in compressed medical images, the wavelets [discrete wavelet transform (DWT)] have had enormous successes in the field of image compression over the past 10 years [5]. Moreover, DCT, wavelet compression, fractal compression, vector quantization (VQ), and linear predictive coding are used in lossless compression [19]. Lossless image compression schemes often comprise two distinct and independent components: modeling and coding [16]. Generally, in addition to numerous other methods of image compression, wavelet transformation, VQ, neural network, and various encoding approaches are normally applied [17]. Some proposals were distinguished for applying video coding methodologies, such as H.264/MPEG-4 AVC [21] or the very current H.265/MPEG-H HEVC [14], to compress three-dimensional (3D) and 4D medical image datasets [23]. Moreover, the quick increase in the range and use of electronic imaging absolves awareness for the systematic design of an image compression system and for extending the image quality needed in dissimilar applications [8]. The dispute, on the other hand, is that although high compression rates are required, the usability of the reconstructed images depends on certain crucial features of the original images, which need to be protected after the compression process has been fulfilled [3].

In this paper, we propose a medical image compression method using wavelet transform and a modified region growing (MRG) algorithm. For medical image compression, only a small part is useful out of the whole image. To find the important region, we first segment the image into two parts, region of interest (ROI) and non-ROI, using the MRG algorithm. Thereafter, we compress the most important part of ROI using DWT and set partitioning in hierarchical tree (SPIHT) encoding. Consequently, we compress the non-ROI using DCT and merging-based Huffman encoding (MHE) method. Then, we combine the ROI and non-ROI. Finally, we obtain the compressed image and then calculate the compression ratio to check the performance of the proposed approach. The reverse process is used for the decompression process. The main contributions of the research are as follows:

- A general medical image compression framework is developed by using wavelet transform and ROI detection scheme. Many previous algorithms can be considered to be the special cases of our approach.
- Under this framework, we can improve certain shortcomings such as information loss, preserve edges, and poor quality that are present in the existing image compression approaches, and obtain an improvement in compression ratio, peak signal-to-noise ratio (PSNR), cross-correlation, average difference, and normalized absolute error (NAE).
- For segmentation, to identify the ROI and non-ROI, we design a novel algorithm for segmentation namely, MRG. In addition, MRG provides an accurate ROI in standard low-resolution images with complete and fulfilled segmentation structure.

# 2 Literature Survey

For an image compression system, different researchers have suggested many approaches. Among them, a handful of significant studies are presented in this section. A developed medical image compression technique with lossless ROI has been examined by Zuo et al. [24]. Here, the image was first separated into two parts: ROI and non-ROI. A lossless compression algorithm was then used to the marked area of ROI, and an image restoration technique and the wavelet-based lossy compression algorithm were employed to the other area of the image.

Moreover, the equated approach of pixel and block level splitting for medical image compression and reconstruction has been proposed by Sunil et al. [15]. This strategy was implemented by separating the input image into subproblems. With this strategy, they divided the input image into subproblems, which were resolved by applying a compressed sensing method. After attaining the output of the compressed sensed image, this problem was passed to the iterative process of the problem resolver to decrease the time, computational complexity, and reconstruction error.

Additionally, wavelet-based volumetric medical image compression has been introduced by Bruylants et al. [6]. They thoroughly distinguished techniques that permit developing the performance of JPEG 2000 for volumetric medical image compression in their paper. For this purpose, they used a newly improved generic codec framework that supports IPEG 2000 with its volumetric extension (IP3D), different directional wavelet transforms, as well as a generic intra-band prediction mode. A thorough objective investigation of the performance-complexity trade-offs extended by these techniques on medical data was carried out. Moreover, they rendered a comparison of the proposed techniques to H.265/MPEG-H HEVC, which was currently the most state-of-the-art video codec available.

Image compression applying Shearlet coefficient and ROI detection has been implemented by Aneja et al. [2]. The input image was initially segmented into three regions, ROI, non-ROI, and background, applying histogram-based thresholding. Then, Shearlet transform was used to extract Shearlet coefficients for ROI and non-ROI. Afterward, the ROI was compressed by Huffman coding and the non-ROI was compressed by SPIHT coding. The non-relevant regions were directly changed to zero. Moreover, the inverse Shearlet transform and decoding techniques reconstruct the image, reversibly, up to the desired quality.

An ROI-based MRI brain image compression by applying the Peano space-filling curve (PSFC) has been invented by Devadoss et al. [7]. In their paper, they suggested an efficient approach for medical image compression established on the basis of applying the PSFC. In this method, the region comprising the most useful diagnostic features is covered in an ROI; pixels in the ROI are arranged applying PSFC and entropy encoded without any loss in quality. The remaining regions are covered as non-ROI and encoded with singular value decomposition followed by entropy encoding. The encoded ROI and non-ROI are associated to give the compressed output.

An efficient strategy for brain image (tissue) compression established on the position of the brain tumor has been discovered by Kumarganesh and Suganthi [10]. Here, computer-aided detection of brain tissue compression was established on the estimation of the location of the brain tumor. The suggested system detects and segments the brain tissues and brain tumor using mathematical morphological operations, Further, the brain tissue with the tumor is compressed by applying the lossless compression technique and the brain tissue without tumor is compressed by applying the lossy compression technique.

Additionally, a 3D separate descendant-based SPIHT algorithm for fast compression of high-resolution medical image probes has been discovered by Song et al. [13]. To render a fast compression algorithm for high-resolution medical image sequences, an efficient 3D separate descendant-based SPIHT algorithm was suggested in this study. To facilitate the transformation, 3D integer wavelet transform was applied at first. Established on an efficient spatial-temporal tree structure, which was contrived for the transformed coefficients, the authors suggested a fast coding strategy by dividing the descendant set into an offspring set and a leaves set. The suggested algorithm showed more selectivity in deciding the scanning and coding of the descendant sets, and hence the coding time was escalated.

# 3 Algorithm Used in the Proposed Approach

In this section, we explain the algorithm used in this paper. Thereafter, we explain in depth the working principles of the proposed image compression method.

# 3.1 Huffman Encoding Process

The Huffman coding technique is a strategy that takes a shot at the two data and image for compression. It is normally done in two phases. A statistical model is first extensively gathered; thereafter, in the second phase, the image data are encoded, as produced by the statistical model in the first phase. These codes are of variable code length utilizing an essential number of bits. This thought causes a lessening in the average code length and in this manner general size of compacted data is littler than the first.

- **Step 1:** Read the image in the workspace of Matlab.
- **Step 2:** Change the given color image to a gray-level image.
- Step 3: The probability of images are organized in diminishing request and lower probabilities are combined, and this progression proceeds until the point that only two probabilities are left and codes are relegated by deciding that the most astounding plausible image will have a shorter length code.

**Step 4:** Advance Huffman encoding is performed, i.e. mapping of the code words to the compared images will bring about packed data.

#### 3.2 MHE

The MHE algorithm consists of three significant steps: (i) Huffman code creation of original data, (ii) code conversion-based conditioning, and (iii) encoding. The following subsections explains the detailed process of the MHE algorithm.

#### A) Huffman code creation of original data

- Arrange the probabilities of the symbols (nodes) in descending order.
- Using the symbols of two lowest probabilities,  $P_A$  and  $P_B$ , create a new node of which these two probabilities are branches, the new node being labeled with the arithmetic sum of these two probabilities.
- Repeat the process using the new code instead of the original two, until only one node is left.
- Label each upper branch with a "0" and the lower member of each pair with a "1" or vice versa.
- The code for each of the original symbols is then determined by proceeding from the root of the tree to the required leaf, noting the branch label of each node traversed.

#### B) Code conversion of a condition-based sequence

After generating a Huffman code for each symbol or original data (unsigned 8-bit integer value), the code conversion of the condition-based sequence is done. The process of code conversion of the conditionbased sequence is as follows: initially, the original data and their code words are taken. Thereafter, the code conversion process is done by merging two symbols, i.e. number of times the selected combination of two symbols is repeated. Then, the merging process is based on the following observations:

- First, the merging process is applied on the selected symbols, which have to satisfy the condition, i.e. the selected symbols that satisfied this condition criterion (>1) then qualified for the merging process.
- Secondly, the pair that same combination of first digit of the selected pair that should not come.
- Third, the bit length of the first position of the pair should be lesser than the bit length of the second position of the pair.
- If the above three conditions are satisfied, the first position of the symbol is repeated twice and that new pair replaces the selected pair or old pair.
- Similarly, the above process is repeated for all selected code words.

#### C) Encoding

The encoding process is done based on the combination of the symbol used in the code conversion of the merging-based sequence and the preceding symbol in the original data. The encoding process is as follows: at first, the combination of the symbols used for the code conversion process and the preceding symbol of the combination of the symbols used for the code conversion process are checked to decide

whether the code formed using the code conversion process is to be considered or not. Then, three conditions (explained in the above steps) are applied and each symbol is checked in order to encode the original data. After this verification, a code is formed for the original data. The final code is the encoded data based on the MHE technique.

#### 3.2.1 Example of the MHE Algorithm

The complete process of MHE is explained by the following example. Assume "(215) (145) (215) (145) (126) (215) (51) (45) (215) (126)" are original data and the Huffman code is formed for each symbol in the original data. The formation of the code is shown in Figure 1.

Figure 1 is explained as follows: initially, the repeated letters are considered for one time and mark the frequency of the letter in the original data, i.e. the numeric term 3 below the unsigned integer value "215" represents the symbol "215" repeated three times in the original data. Thereafter, two symbols with the least frequency are taken, and zero and one are assigned to them and represent the total frequency of the two symbols below it. Similarly, this process is done by taking the next two symbols, and zeros and ones are assigned until the last step. The Huffman code is then formed for each symbol by considering the corresponding branches of zeros and ones from the last step to the first. From Figure 2, the Huffman code formed for the symbol "215" is 1; the Huffman code formed for the symbol "145" is 000; the Huffman code formed for the symbol "126" is 01; the Huffman code formed for the symbol "51" is 0011; and the Huffman code formed for the symbol "45" is 0010. Eventually, the Huffman code for the original data "(215) (145) (215) (145) (126) (215) (51) (45) (215) (126)" is "1000100001100110010101." Figure 3 shows the direction of the Huffman code formed for each letter.

After generating the Huffman code for each letter, the code conversion of the condition-based sequence is done by arranging the letters with the least length first. The code conversion of the merging-based sequence process is used to compress the data. In Figure 3, we obtain the frequency combination of two symbols first and the merging process is applied only on the satisfying combination. Based on the merging conditions, at first, the combination of the symbol "(215) (145)" is repeated at least two times that it only qualified for the merging process. Second, the pair of that same combination of the first digit of the selected pair "(215) (145)" should not repeat. Third, the bit length of the first position of the pair (1) should be lesser than the bit

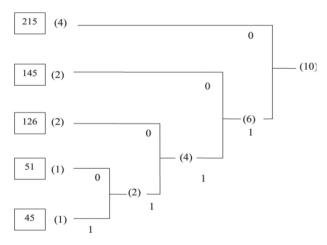


Figure 1: Formation of Huffman Code.

Original	215	145	215	145	126	215	51	45	215	126
Modified	215	215	215	215	126	215	51	45	215	126
Compressed	1	1	1	1	01	1	0011	0010	1	01

Figure 2: After the Code Conversion Process.

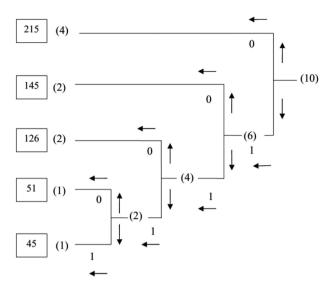


Figure 3: Direction of Huffman Code Formation.

length of the second position of the pair (000). Now, the pair "(215) (145)" is qualified for the merging process. Then, the merging process is done by step 4 in the code conversion section. The new code (215) (215) is replaced ahead for (215) (145). Similarly, the code conversion process is done for all the original data. Finally, the MHE-based encoded data for the original data "(215) (145) (215) (145) (126) (215) (51) (45) (215) (126)" is "111101100110010101." Table 1 shows the symbols with their codes after the code conversion process.

# 3.3 SPIHT Encoding

A more efficient implementation of EZW (embedded zero wavelet) is the SPIHT algorithm [11, 17], which was represented by Shapiro. After using the wavelet transform to an image, the SPIHT algorithm partitions the decomposed wavelet into important and insignificant partitions established based on the following function:

$$S_n(T) = \begin{cases} 1 & \max_{(i,j) \in T_M} \left\{ \left| C_{i,j} \right| \right\} \ge 2^n \\ o & \text{otherwise} \end{cases}$$
 (1)

Here,  $S_n(T)$  is a crucial set of coordinates T and  $C_{i,j}$  is the coefficient value at coordinate (i, j). There are two passes in the algorithm: the sorting pass and the refinement pass. The SPIHT encoding process applies three lists: LIP (list of insignificant pixels), comprising individual coefficients that have magnitudes smaller than the thresholds; LIS (list of insignificant sets), comprising the set of wavelet coefficients determined by tree structures and determined to have magnitudes smaller than the threshold; and LSP (list of significant pixels), which is a list of pixels determined to have magnitudes larger than the threshold (significant). On the above three lists, the sorting pass is executed. The maximum number of bits needed to present the largest coefficient in the spatial orientation tree is found and represented by  $n_{max}$ :

$$n_{\max} = \left[\log_2\left(\max_{i,j}\{|C_{i,j}|\}\right)\right]. \tag{2}$$

Those coordinates of the pixels that persist in the LIP are tested for significance by applying Eq. (2), from the sorting pass. The result is sent to the output and out of it the significant will be transferred to the LSP as well as having their sign bit output. Sets in the LIS will also get their significance examined, and, if determined significant, will be neglected and partitioned into subsets. Subsets with only one coefficient and determined to be significant will be eliminated and separated into subsets. Subsets having only one coefficient and determined to be crucial will be inserted to the LSP; otherwise, they will be inserted to the LIP. The *n*th most significant bit of the coefficients in the LSP is the final output in the refinement pass. The value of *n* is decremented,

Table 1: Segmentation Results Using the MRG Method.

Segmentation using modified region growing algorithm								
Input medical images	ROI	NON-ROI						
	0							
	•							

and the sorting and refinement passes are used again. These passes will continue until either the desired rate is attained or n = 0, and all nodes in the LSP have all their bit output. The latter case will cause an almost exact reconstruction so all the coefficients have been processed completely. At any time, the bit rate can be controlled exactly by the SPIHT algorithm as the output developed is in single bits and the algorithm can be completed. In terms of processing time, the decoding process follows the encoding exactly and is almost symmetrical.

#### 3.4 Wavelet Transform

DWT is used to transform the image from its spatial domain into its frequency domain. In time, the wavelets are signals that are local in scale and normally have an irregular shape. A wavelet is a waveform of effectively determined duration that has an average value of zero. The term "wavelet" is derived from the fact that it integrates to zero; it waves up and down across the axis. For compact signal representation, several wavelets also display a property ideal. A signal can be decomposed into many shifted and scaled representations of the original mother wavelet. In component wavelets, a wavelet transform can be applied to decompose a signal. The images are decomposed into low-low (LL), low-high (LH), high-low (HL), and high-high (HH) elements, where LL is the approximate coefficient and the remaining three are detailed coefficients. Once this is caused, the coefficients of the wavelets can be decimated to neglect some of the details. For the frequency domain transforms, we utilize Haar-DWT with this suggested method. There are two operations utilizing a 2D Haar-DWT. One is the horizontal operation and the other is the vertical one. The detailed operations of 2D Haar-DWT are stated as follows [5].

Step 1: First, scan the pixels from left to right in subtraction operations on neighboring pixels and store the sum horizontal direction. Repeat this operation until all the rows are worked. The pixel sums present the low-frequency part (denoted as symbol L), while the pixel differences present the high-frequency part of the original image (denoted as symbol H).

Step 2: Second, scan the pixels from top to bottom in the vertical direction. Perform the addition and subtraction operations on neighboring pixels, and then store the sum on the top and the difference on the bottom. Repeat this operation until all the columns are processed. Finally, four sub-bands are obtained, denoted as LL, HL, LH, and HH, respectively. The LL sub-band contains the low-frequency information and it looks the same as the original image. Similarly, other bands, HL, LH, and HH, contain the high-frequency information. A wavelet function  $\Psi(t)$  has two main properties:

$$\int_{-\infty}^{0} \Psi(t) dt = 0.$$
 (3)

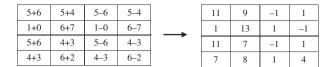
That is, the function is oscillatory or has a wavy appearance:

$$\int_{-\infty}^{0} |\Psi(t)|^2 dt < \infty. \tag{4}$$

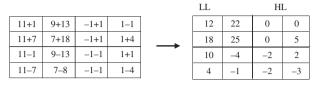
Let us consider an example for a better understanding of the 2D Haar-DWT of an image. Figure 4 illustrates the pixel-by-pixel representation of a  $4 \times 4$  image. Figure 5 shows the output of the 2D Haar-DWT.

5	6	5	4
1	0	6	7
5	6	4	3
4	3	6	2

**Figure 4:** Pixel by Pixel Representation of a  $4 \times 4$  Image.



Horizontal operations



Vertical operations LH HH

Figure 5: Example of 2D-Haar DWT of an Image.

#### 3.5 DCT

In multimedia standards, DCT is an orthogonal transformation that is very widely applied in image compression and is widely accepted. DCT belongs to a family of 16 trigonometric transformations. The type 2 DCT transmutes a block of image size  $N \times N$  having pixel intensities  $S(n_1, n_2)$  into a transform array of coefficients  $S(K_1, K_2)$ , depicted by the following equation:

$$S(K_1, K_2) = \sqrt{\frac{4}{N^2}} C(K_1) C(K_2) \sum_{n_1 = 0}^{N-1} \sum_{n_2 = 0}^{N-1} S(n_1, n_2) \cos\left(\frac{\pi (2n_1 + 1)K_1}{2N}\right) \cos\left(\frac{\pi (2n_2 + 1)K_2}{2N}\right),$$
 (5)

where  $K_1, K_2, n_1, n_2 = 0, 1, ..., N - 1$ , and

$$C(K) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k = 0\\ 1 & \text{otherwise} \end{cases}$$
 (6)

The transformed array  $S(K_1, K_2)$  found through Eq. (5) is also of the size  $N \times N$ , same as that of the original image block. In the directions of  $n_1$  and  $n_2$ , it should be mentioned here that the transform-domain indices  $K_1$  and  $K_2$  denote the spatial frequencies, respectively.  $K_1 = K_2 = 0$  corresponds to the average or the direct current component, and all the remaining ones are the alternating current components that correspond to higher spatial frequencies as  $K_1$  and  $K_2$  increase.

# 4 Proposed Image Compression Methodology

The main aim of the proposed methodology is to compress the input image using multiple phases. Figure 6 shows the details of the proposed methodology. The first phase is segmentation, which segments the magnetic resonance image into ROI and non-ROI. Magnetic resonance images are provided to the system for the purpose of compression. The second phase is compression. In this paper, the compression is done using wavelet transform and encoding methods. The final phase is decompression, which is a reverse process of compression.

## 4.1 Segmentation of ROI and non-ROI

The ROI is the most significant portion of a medical image. It comprises the most valuable data of the medical image and should not undergo any modification. There can be numerous disjoint ROIs in a medical image and numerous manners exist to describe the ROI in a medical image. Here, a segmentation algorithm is used

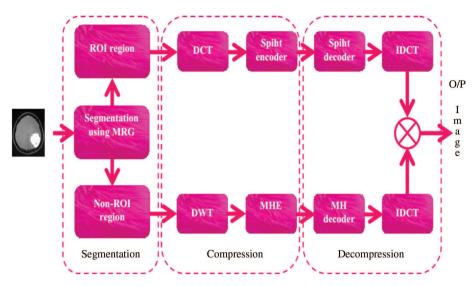


Figure 6: Overall Diagram of a Proposed Methodology.

to detect the ROI from the input image. Segmentation is one of the most important parts of the complete image processing cycle, and it forms a very helpful and an essential part of object detection. In this paper, to segment the ROI from the input image, we utilize the MRG algorithm. Consider the input image I(i, j) that has a size of 256  $\times$  256. Initially, we segment the ROI from the input image. The projected MRG technique segments the input image with respect to a point called a seed. In the region growing segmentation, the chief point is to regulate the initial seed points. A seed point is the commencement stage for region growing, and its selection is significant for the segmentation solution. The technique of mathematical morphology is engaged in order to attain an initial seed point. The detailed procedure of the projected region growing-based image segmentation procedure is elucidated below.

**Step 1:** Consider the input image I(i, j), which has a size of 256  $\times$  256. Here, first, we split the image into a number of blocks,  $B^i$ . Each block has one center pixel and a number of neighborhood pixels.

**Step 2:** Then, we set the intensity threshold,  $(T^{IN})$ .

**Step 3:** For every block  $B^i$ , proceed with the subsequent processes in step 7 until the number of blocks reached the total number of blocks for an image.

**Step 3(a):** Find the histogram H of each pixel in  $B^i$ .

**Step 3(b):** Regulate the most frequent histogram of the  $B^{ith}$  block and signify it as  $F^H$ .

**Step 3(c):** Prefer any pixel according to  $F^H$  and allocate that pixel as seed point that has the intensity  $IN_p$ .

**Step 3(d):** Deliberate the adjacent pixel having the intensity  $IN_n$ .

**Step 3(e):** Find the intensity difference of those pixels *p* and *n*:

$$D_{IN} = ||IN_p - IN_n||. (7)$$

**Step 3(f):** If  $D_{IN} \leq T^{IN}$ , then add the consistent pixel to the region and the region is grown; else, move to step 3(h).

Step 3(g): Check whether all pixels are added to the region. If true, go to step 2, then go to step 3(h).

Step 3(h): Re-estimate the region and detect the new seed points and do the procedure in step 3(a).

**Step 4:** Stop the whole procedure.

With the help of this MRG procedure, the input images are segmented. The segmented image output is displayed in Figure 7.

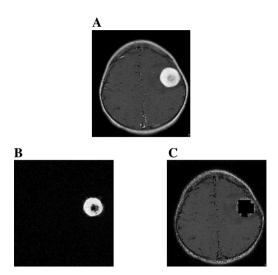


Figure 7: Segmentation Output. (A) Input image. (B) Segmented ROI image. (C) Segmented non-ROI image.

# 4.2 Image Compression Algorithm

The primary intention of our research is to compress the medical image efficiently. The objective of image compression is to reduce the irrelevance and redundancy of image data in order to store or transmit data in an efficient form. Here, segmented regions are effectively compressed by employing wavelet transform and encoding techniques. The proposed image compression algorithm is given below:

Input: original image  $O_I[i,j]$ .

Output: compressed image  $C_I[i.j]$ .

#### Step 1: Segmentation using region growing algorithm

The first step of image compression is segmentation. For segmentation, in this paper, we utilize an MRG algorithm. Initially, the input magnetic resonance image I[i,j] is clustered into two segments such as ROI image  $I_{ROI}[i,j]$  and non-ROI image  $I_{non-ROI}[i,j]$  using the MRG algorithm. After the ROI detection process, the major task is to compress the input image or medical image I[i,j] with the lossless version. To obtain this result, we have individually compressed each segmented images as [ROI]<sub>c</sub> and [non-ROI]<sub>c</sub>.

#### **Step 2: Computation of** $[ROI]_c$

- To compute [ROI]<sub>c</sub>, DCT is first applied to select DCT coefficients; then, SPIHT is applied to bit streams with the aim of doing lossless compression in DCT coefficients.
- After the cosine transform, the SPIHT encoding algorithm is used to generate a compressed ROI image [ROI]<sub>c</sub>. The compressed ROI image [ROI]<sub>c</sub> is present in the bit stream format. The SPIHT encoder converts the image components into the bit stream. SPIHT works by partitioning the wavelet-decomposed image into significant and insignificant partitions based on the following function:

$$S_n(\tau) = \begin{cases} 1, & \text{if } \max_{(i,j) \in \tau} \left\{ c_{i,j} \right\} \ge 2^n \\ 0, & \text{Otherwise} \end{cases}$$
(8)

where  $2^n$  is the threshold,  $S_n(\tau)$  is a significant set of coordinates, and  $\tau$  and  $c_{i,j}$  are the coefficient values at coordinates i, j. In this algorithm, three ordered lists are used to store the significance information during set partitioning: LIS, LIP, and LSP. The detailed explanation of the SPIHT procedure is given in Section 3.1. Finally, we obtain the compressed ROI image [ROI]<sub>c</sub> from the encoding process.

#### Step 3: Computation of [non-ROI]<sub>c</sub>

- To compute [non-ROI]<sub>c</sub>, we first decompose the image  $I_{\text{non-ROI}}[i, j]$  into sub-bands using DWT, which improves the compression ratio. The divided sub-bands are HH, LL, HL, and LH, From the four sub-bands. the HL sub-band is chosen for the encoding process.
- Thereafter, the non-ROI image  $I_{\text{non-ROI}}[i,j]$  is encoded using MHE, which is used to compress the bit stream. The detailed explanation of the MHE approach is explained in Section 3.2. Finally, we obtain the compressed non-ROI image, [non-ROI]<sub>c</sub>.

# Step 4: Evaluate compression ratio $I^{\text{Com}}[i,j]$ using [ROI]<sub>c</sub> and [non-ROI]<sub>c</sub>

From the above steps (2) and (3), we individually calculate the compression ratio using Eq. (9). Finally, the compressed medical image or input image  $I^{\text{Com}}[i,j]$  is obtained from Eq. (10).

Compression ratio 
$$=$$
  $\frac{\text{Size of original image}}{\text{Size of compressed bit stream}}$ . (9)

$$I^{\text{Com}}[i,j] = \frac{[\text{ROI}]_c + [\text{non-ROI}]_c}{2}.$$
 (10)

# 4.3 Image De-compression Algorithm

The decompression process is exactly the inverse of the compression process. First, the bit stream of the ROI mask code is decoded using the SPIHT decoding decompression algorithm, and that of the non-ROI code is decompressed by the corresponding modified Huffman decoding method. The proposed image de-compression stage involves following significant steps:

- Input: compressed bit stream I<sup>Com</sup>[i.j].
- Output: compressed image I<sup>Dec</sup>[i.i].

#### **Step 1: Calculation of [non-ROI]** Dec

The modified Huffman decoding and inverse DWT are used to compute [non-ROI] Dec from the compressed image [non-ROI]<sub>c</sub>. The reverse operation of merge Huffman coding and DWT is performed to obtain  $[\text{non-ROI}]_{dc}$ .

#### Step 2: Calculation of [ROI] $_{dc}$

Here, the decompressed ROI image  $[ROI]^{Dec}$  is calculated from the compressed image  $[ROI]_c$  through the reverse operation of DCT and SPIHT decoder. Finally, [ROI] Dec is obtained using inverse DCT and SPIHT decoder.

## Step 4: Calculate decompressed image $I^{Dec}[i, j]$ using OR operation

To calculate the decompressed image  $I^{Dec}[i, j]$ , we merge the three decompressed regions as in steps 1 and 2. Finally, the resultant decompressed image  $I^{\text{Dec}}[i,j]$  is obtained using logical OR operation as Eq. (11):

$$I^{\text{Dec}}[i,j] = [\text{ROI}]^{\text{Dec}} ||[\text{non-ROI}]^{\text{Dec}},$$
(11)

where  $I^{\text{Dec}}[i, j]$  is the decompressed output image, [ROI]  $^{\text{Dec}}$  is the decompressed ROI, and  $[\text{non-ROI}]^{\text{Dec}}$  is the decompressed non-ROI.

# 5 Results and Discussion

This section presents the results obtained from the experimentation and the detailed discussion about the results. The proposed approach of image compression for medical image datasets and the results are evaluated with the compression ratio, PSNR, average difference, cross-correlation, and NAE. The proposed image compression technique is performed on a Windows machine with the following configurations: Intel (R) Core i5 processor, 3.20 GHz, 4 GB RAM, and Microsoft Windows 7 Professional operating system. We used Matlab (latest version, 7.12) for this proposed technique.

#### 5.1 Evaluation Matrices

In this paper, the performance is analyzed based on famous metrics such as compression ratio, PSNR, cross-correlation, average difference, and NAE, defined as follows:

**Compression ratio:** The image compression ratio is the ratio between the uncompressed size and compressed size, as follows:

$$Compression ratio = \frac{Size of original image}{Size of compressed bit stream}.$$
 (12)

**PSNR:** The description of PSNR is given in the following formulas:

$$PSNR = 10 \log \frac{E_{\max}^2 \times D_i[i,j]_w \times D_i[i,j]_h}{\sum \left(O_i[i,j]_{xy} - D_i[i,j]_{xy}\right)},$$
(13)

where  $D_i[i.j]_w$  and  $D_i[i.j]_h$  are the width and height of the decompressed image,  $O_I[i.j]_{xy}$  is the original image pixel value at coordinate (x, y),  $D_i[i.j]_{xy}$  is the decompressed image pixel value at coordinate (x, y), and  $E_{\text{max}}^2$  is the largest energy of the image pixels (i.e.  $E_{\text{max}} = 255$  for 256 gray-level images).

**Cross-correlation:** The cross-correlation between the original image  $O_I[i,j]$  and decompressed image  $D_i[i,j]$  is

$$O_{I}[i,j] \circ D_{i}[i,j] = \sum_{i=0}^{M-1} \sum_{i=0}^{N-1} O_{I}[i,j] D_{i}[i,j].$$
(14)

**Average difference:** The average difference (AD) of the input and decompressed image is given by

$$AD = \frac{1}{MN} \sum_{i=0}^{M} \sum_{j=0}^{N} \{O_I[i,j] - D_i[i,j]\}. \tag{15}$$

**NAE:** The NAE is given by

$$NAE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |[O_I(i,j) \cdot D_i(i,j)]|}{\sum_{i=1}^{M} \sum_{j=1}^{N} |O_I(i,j)|}.$$
 (16)

### 5.2 Experimental Results

In this paper, we proposed a medical image compression approach. At first, the input magnetic resonance image is segmented into two regions (ROI and non-ROI) using the MRG algorithm. Then, we compress the ROI and non-ROI separately. Finally, we calculate the compressed image from the input image. Here, we analyze the performance of the proposed system using two phases: segmentation and compression. The visual results of the segmentation phase are given in Table 1, and the Possibilistic Fuzzy C-means clustering (PFCM)-based segmentation results are given in Table 2.

 Table 2: Segmentation Result Using the PFCM Method.

Segmentation using PFCM							
Input medical images	ROI	NON-ROI					

# 5.3 Comparative Analysis

In this section, we compare our proposed image compression approach with the PFCM-based image compression approach. In our proposed approach, at first, we segment the input image into two regions using

Table 3: Comparative Analysis of Proposed against Existing Method for the Compression Stage.

Images	Compression ratio		PSNR		Cross correlation		NAE		Average difference	
	Proposed	PFCM	Proposed	PFCM	Proposed	PFCM	Proposed	PFCM	Proposed	PFCM
Image 1	3.88	3.58	39.73	31.00	0.86	0.7358	0.134	0.38	5.6031	0.0370
Image 2	1.68	3.39	38.65	30.78	0.87	0.6929	0.138	0.37	6.17	0.0349
Image 3	2.30	2.16	44.29	36.05	0.87	0.6685	0.092	0.19	4.59	0.0051
Image 4	2.50	1.02	41.98	32.68	0.88	0.7468	0.128	0.30	6.249	0.0568
Image 5	2.46	1.05	46.69	34.55	1.12	0.684	0.101	0.31	3.176	0.047

Table 4: Comparative Analysis of Proposed against Existing Method for Segmentation Stage.

Images			MRG			PFCM	
	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy	
Image 1	0.386317	0.99917	0.953521	1	0.993465	0.99369	
lmage 2	0.411155	0.99719	0.952301	0.9796	0.993048	0.99273	
Image 3	0.253367	0.99485	0.769744	1	0.994797	0.99485	
Image 4	0.525656	0.99900	0.962127	1	0.991914	0.99212	
Image 5	0.379811	0.99925	0.973220	0.9997	0.981384	0.98254	

the MRG algorithm. Then, we compress the ROI using DWT with SPIHT encoding and the non-ROI using DCT with modified Huffman encoding method. To prove the effectiveness of the proposed approach, the proposed method is compared with PFCM-based image compression. Here, for segmentation, the PFCM algorithm is used. PFCM is a combination of fuzzy C-means and possibilistic C-means. Table 3 shows the performance of the proposed approach against existing metrics.

Table 3 shows the comparative analysis of the proposed against existing methods for the compression stage. For experimentation, here we used five magnetic resonance images. When analyzing Table 1, our proposed approach achieves the maximum PSNR of 39.73 dB for using image 1, 38.65 dB for using image 2, 44.29 dB for using image 3, 41.98 dB for using image 4, and 46.69 dB for using image 5. Consequently, the PFCM-based compression obtains the maximum PSNR of 31.00 dB for using image 1, 30.78 dB for using image 2, 36.05 dB for using image 3, 32.68 dB for using image 4, and 34.55 dB for using image 5. Similarly, our proposed approach achieves the maximum compression ratio of 3.88 dB for using image 1, 1.68 dB for using image 2, 2.30 dB for using image 3, 2.50 dB for using image 4, and 2.46 dB for using image 5. Consequently, PFCM-based compression obtains the maximum compression of 3.58 dB for using image 1, 3.39 dB for using image 2, 2.16 dB for using image 3, 1.02 dB for using image 4, and 1.05 dB for using image 5. From the table, we clearly understand that our proposed approach achieves the maximum PSNR and compression ratio compared to PFCM-based compression. Table 2 shows the performance of the segmentation stage. In this proposed approach, we used the MRG algorithm for segmentation. When analyzing Table 4, our proposed approach achieves the maximum accuracy of 99.36% for using image 1, 99.27% for using image 2, 99.48% for using image 3, 99.21% for using image 4, and 98.25% for using image 5. From the result, we clearly understand that our proposed approach achieves the better result.

# 6 Conclusion

Image compression is a noteworthy component in reducing broadcast as well as capacity costs. The whole image compression frameworks are useful in their related fields and everyday novel compression framework is rising, which outfit retrieved compression ratio. The proposed strategy can be utilized as a base model for studies. In this paper, ROI lossless image compression using wavelet transform with encoding method is implemented. Here, the segmentation is performed based on the MRG algorithm, ROI compression is made using DWT with SPIHT encoding method, and non-ROI compression is made using DCT with MHE algorithm.

From the outcomes, the proposed method has demonstrated a high PSNR and high compression ratio when contrasted with the existing method. The performance of the segmentation stage was also analyzed according to sensitivity, specificity, and accuracy.

# **Bibliography**

- [1] A. Alarabevyat, S. Al-Hashemi, T. Khdour, M. Hjouj Btoush, S. Bani-Ahmad and R. Al Hashemi, Lossless image compression technique using combination methods, J. Softw. Eng. Appl. 5 (2012), 752-763.
- [2] R. Aneja and A. H. Siddiqi, Hybrid image compression using Shearlet coefficients and region of interest detection, J. Med. Imaging Health Inform. 6 (2016), 506-517.
- [3] M. A. Ansari and R. S. Anan, Recent trends in image compression and its application in telemedicine and teleconsultation, in: National Systems Conference, pp. 59-64, 2008.
- [4] A. Avramovic, Lossless compression of medical images based on gradient edge detection, in: 19<sup>th</sup> Telecommunications forum (TELFOR), pp. 22-24, 2011.
- [5] M. Beladgham, I. Boucli Hacene, A. Taleb-Ahmed and M. Khélif, MRI images compression using curvelets transforms, AIP Conf. Proc. 1019 (2008), 249.
- [6] T. Bruylants, A. Munteanu and P. Schelkens, Wavelet based volumetric medical image compression, J. Signal Process. Image Commun. 31 (2015), 112-133.
- [7] C. P. Devadoss, B. Sankaragomathi and T. Monica, Region of interest based MRI brain image compression using Peano space filling curve, J. Current Signal Transduct. Ther. 11 (2016), 114-120.
- [8] S. Grgic, M. Grgic and B. Zovko-Cihlar, Performance analysis of image compression using wavelets, IEEE Trans. Indust. Electron. 48 (2001), 682-695.
- [9] A. Kumar Gupta, M. Dyer, A. Hirsch, S. Nooshabadi and D. Taubman, Design of a single chip block coder for the EBCOT engine in JPEG2000, in: Proceedings of the 48th Midwest Symposium on Circuits and Systems, Cincinnati, OH, USA, pp. 63-66, 2005.
- [10] S. Kumarganesh and M. Suganthi, An efficient approach for brain image (tissue) compression based on the position of the brain tumor, Int. J. Imaging Syst. Technol. 26 (2016), 237-242.
- [11] H.-C. Kuo and Y.-L. Lin, A hybrid algorithm for effective lossless compression of video display frames, IEEE Trans. Multimed. 14 (2012), 500-509.
- [12] R. Loganathan and Y. S. Kumaraswamy, Medical image compression using biorthogonal spline wavelet with different decomposition, Int. J. Comput. Sci. Eng. 2 (2010), 3003-3006.
- [13] X. Song, Q. Huang and S. Chang, Three-dimensional separate descendant-based SPIHT algorithm for fast compression of high-resolution medical image sequences, J. IET Image Process. 11 (2017), 80-87.
- [14] G. J. Sullivan, J.-R. Ohm, W. Han and T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, IEEE Trans. Circuits and Syst. Video Technol. 22 (2012), 1649-1668.
- [15] H. Sunil, Sharanabasaweshwar and G. Hiremath, A combined scheme of pixel and block level splitting for medical image compression and reconstruction, Alexandria Enq. J. (2017), https://doi.org/10.1016/j.aej.2017.03.001.
- [16] I. Sutskever, J. Martens and G. Hinton, Generating text with recurrent neural networks, in: International Conference on Machine Learning, Bellevue, WA, USA, pp. 1-8, 2011.
- [17] M. F. Ukrit, A. Umamageswar and G. R. Suresh, A survey on lossless compression for medical images, Int. J. Comput. Appl. 31 (2011), 47-50.
- [18] I. S. Uzun and A. Amira, Real-time 2-D wavelet transform implementation for HDTV compression, Real-Time Image Process. **11** (2005), 151-165.
- [19] G. Vijayvargiya, S. Silakari and R. Pandey, A survey: various techniques of image compression, Int. J. Comput. Sci. Inform. Secur. 11 (2013), 1-6.
- [20] A. Weinlich, J. Rehm, P. Amon, A. Hutter and A. Kaup, Massively Parallel Lossless Compression of Medical Images Using *Least-Squares Prediction and Arithmetic Coding*, IEEE, pp. 1680–1685, 2013.
- [21] T. Wiegand, G. J. Sullivan, G. Bjntegaard and A. Luthra, Overview of the h.264/avc video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (2003), 560-576.
- [22] P. Wu, K. Xie, H. Yu, Y. Zheng and W. M. Yu, A new preprocessing algorithm used in color image compression, J. Adv. Future Comput. Control Syst. 1 (2012), 465-471.
- [23] H. Yu, Z. Lin and F. Pan, Applications and improvement of H.264 in medical video compression, IEEE Trans. Circuits Syst. **52** (2005), 2707–2716.
- [24] Z. Zuo, X. Lan, L. Deng, S. Yao and X. Wang, An improved medical image compression technique with lossless region of interest, Int. J. Light Electron Optics 126 (2015), 2825-2831.