

Medical Image Compression using DCT and DWT Techniques

Gullanar M. Hadi

*College of Engineering-Software Engineering Dept.
Salahaddin University-Erbil, Iraq
gullanarm@yahoo.com*

ABSTRACT

In this paper we used DCT (Discrete Cosine Transform) and DWT (Discrete Wavelet Transform) to achieve optimal compression ratio for two types of medical images (CT scan and MRI) with keeping good quality of these images. In 2D- DCT; the image is broken up into blocks of $n \times m$ pixels p_{xy} (with $n=m= 8$ typically), and then 2D- DCT is used to produce a block of $n \times m$ DCT coefficients for each block of pixels, the result coefficients are quantized, which must result in lossy but highly efficient compression ratio . The important feature of the DCT the feature that makes it so useful in data compression, is that it takes correlated input data and concentrates its energy in just the first few transform coefficients. For the same images we used DWT method, and then compare their results to choose the best method for compressing medical image with good quality. The compression algorithms programmed and tested to achieve the goal of reducing the file size (i.e., good compression ratio) using Matlab (R2008a).The conclusion part in this paper explain the results.

Keywords: Data Compression Techniques, Image Compression, DCT, DWT, Medical images.

1 Introduction

In computer science and information theory, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression is useful because it helps reduce resources usage, such as data storage space or transmission capacity. Because compressed data must be decompressed to use, this extra processing imposes computational or other costs through decompression. Compression can be either lossy or lossless as follows: [1] [2] Lossless compression reduces bits by identifying and eliminating statistical redundancy, where redundancy in information theory is the number of bits used to transmit a message minus the number of bits of actual information in the message. Lossless data compression make use of data compression algorithms that allows the exact original data to be reconstructed from the compressed data. No information is lost in lossless compression. This can be contrasted to lossy data compression, which does not allow the exact original data to be reconstructed from the compressed data. Also lossless compression is used when it is important that the original and the decompressed data be identical, or when no assumption can be made on whether certain deviation is uncritical. Typical examples are executable programs and source code.

Some image file formats, notably PNG, use only lossless compression, while others like TIFF and MNG may use either lossless or lossy methods. GIF uses a lossless compression method, but most GIF implementations are incapable of representing full color, so they quantize the image (often with dithering) to 256 or fewer colors before encoding as GIF. Color quantization is a lossy process, but reconstructing the color image and then re-quantizing it produces no additional loss. On other hand,

WebP is a new image format that provides lossless and lossy compression for images on the web. WebP lossless images are 28% smaller in size compared to PNGs. WebP lossy images are 25-34% smaller in size compared to JPEG images. [3] Lossless data compression is used in many applications. For example, it is used in the popular ZIP file format and in the UNIX tool gzip. It is also often used as a component within lossy data compression technologies.

Lossy compression reduces bits by identifying unnecessary information and removing it. The process of reducing the size of a data file is popularly referred to as data compression, although it's formal name is source coding (coding done at the source of the data before it is stored or transmitted). A lossy data compression method is one where compressing data and then decompressing it retrieves data that may well be different from the original, but is "close enough" to be useful in some way. Lossy data compression is used frequently on the Internet and especially in streaming media and telephony applications. These methods are typically referred to as codecs in this context. Most lossy data compression formats suffer from generation loss: repeatedly compressing and decompressing the file will cause it to progressively lose quality. This is in contrast with lossless data compression.

2 Why do We Need Image Compression?

In case of image compression: As an activity, digital image processing generally creates significant numbers of large files containing digital image data. Very often, these must be archived or exchange among different users and system. This call for efficient methods for the storage and transfer of digital image data files.

-Since digital images, by their nature, are quite data intensive, reducing their size can produce solutions that are more ambitious than would otherwise be practical .by eliminating redundant or unnecessary information, Image compression is the activity that addresses this aim.

-Digital image coding and compression is concerned with the minimization of the memory needed to represent and store a digital image. And the term data compression refers to the process of reducing the amount of data required to represent a given quantity of information. It is well known that raw digital images occupy a large amount of memory. For example 1024 x 1024 color image requires 3 MB of memory for its storage (1024 x 1024 x 3 bytes /pixel).

3 Error Metrics

This article demonstrates the compression and uncompressing of a grayscale image (i.e. Medical images and other type) and computed MSE and PSNR error values using the 'spiht' compression method. See wcompress function in matlab. [5]

Two measures are commonly used to quantify the error between two images: the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) which is expressed in decibels to achieve desirable compression ratios. The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error

The mathematical formulae for the two are: [1], [4], and [5]

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - I'(i,j))^2 \quad (1)$$

$$PSNR = 20 * \log_{10} (255 / \sqrt{MSE}) \quad (2)$$

Where $I(i, j)$ is the original image, $I'(i, j)$ is the approximated version (which is actually the decompressed image) and M, N are the dimensions of the images. A lower value for MSE means lesser error, and as seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction. So, if you find a compression scheme having a lower MSE (and a high PSNR), you can recognize that it is a better one. [4]

3.1 Mean Square Error (MSE)

Mean square error is a criterion for an estimator: the choice is the one that minimizes the sum of squared errors due to bias and due to variance. The average of the square of the difference between the desired response and the actual system output. As a loss function, MSE is called squared error loss. MSE measures the average of the square of the "error. The MSE is the second moment (about the origin) of the error, and thus incorporates both the variance of the estimator and its bias. For an unbiased estimator, the MSE is the variance. In an analogy to standard deviation, taking the square root of MSE yields the root mean squared error or RMSE. Which has the same units as the quantity being estimated. For an unbiased estimator, the RMSE is the square root of the variance, known as the standard error. [4]

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i,j) - K(i,j)\|^2 \tag{3}$$

Where $m \times n$ is the image size and $I(i, j)$ is the input image and $K(i, j)$ is the retrieved image, where the simple image statistics (sample mean, and sample variance) represented as follows:

The sample mean (m_A) of an image A ($N \times M$): [6]

$$m_A = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A(i, j) \tag{4}$$

$$variance = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i, j) - m_A)^2 \tag{5}$$

The sample standard deviation is, $\sqrt{variance}$.

3.2 Peak Signal-to-Noise Ratio (PSNR):

It is the ratio between the maximum possible power of a signal and the power of corrupting noise .Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. The PSNR is most commonly used as a measure of quality of reconstruction in image compression etc. It is most easily defined via the mean squared error (MSE) which for two $m \times n$ monochrome images I and K where one of the images is considered noisy.[1],[4],[5]

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \tag{6}$$

$$PSNR = 20 * \log_{10} (255 / \text{sqrt}(MSE))$$

Here, MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with B bits per sample, MAX_I is $2^B - 1$. Typical values for the PSNR in Lossy image and video compression are between 30 and 50 dB, where higher is better. PSNR is computed by measuring the pixel difference

between the original image and compressed image. Values for PSNR range between infinity for identical images, to 0 for images that have no commonality. PSNR decreases as the compression ratio increases for an image. For more detail see ref. [1] p/465

4 2D Discrete Cosine Transform (DCT) for Image Compression

The important feature of the DCT, the feature that makes it so useful in data compression, is that: it takes correlated input data and concentrates its energy in just the first few transform coefficients. If the input data consists of correlated quantities, then most of the n transform coefficients produced by the DCT are zeros or small numbers, and only a few are large (normally the first ones). See equation (7).

In this paper, however, we discussed image compression which is based on the two-dimensional correlation of pixels (a pixel tends to resemble all its near neighbors, not just those in its row) see equation (8). This is why practical image compression methods use the DCT in two dimensions. The image is broken up into blocks of $n \times m$ pixels (with $n=m= 8$ typically), and Equation (8) is used to produce a block of $n \times m$ DCT coefficients (DC coefficient) for each block of pixels. The coefficients are then quantized, which results in lossy but highly efficient compression. [7][8]

where

$$G_f = \sqrt{\frac{2}{n}} C_f \sum_{t=0}^{n-1} p_t \cos \left[\frac{(2t+1)f\pi}{2n} \right], \tag{7}$$

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0, \\ 1, & f > 0, \end{cases} \text{ for } f = 0, 1, \dots, n-1$$

$$G_{ij} = \frac{2}{\sqrt{mn}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} p_{xy} \cos \left[\frac{(2y+1)j\pi}{2m} \right] \cos \left[\frac{(2x+1)i\pi}{2n} \right]. \tag{8}$$

For $0 \leq i \leq n-1$ and $0 \leq j \leq m-1$ and for C_i and C_j defined by Equation (7). The first coefficient G_{00} is again termed the “DC coefficient” and the remaining coefficients are called the “AC coefficients.” The decoder reconstructs a block of quantized data values by computing the IDCT whose definition is: [7], [9]

$$p_{xy} = \frac{2}{\sqrt{mn}} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} C_i C_j G_{ij} \cos \left[\frac{(2x+1)i\pi}{2n} \right] \cos \left[\frac{(2y+1)j\pi}{2m} \right] \tag{9}$$

We now show one way to compress an entire image with the DCT in several steps as follows:

1. The image is divided into k blocks of 8×8 pixels each. The pixels are denoted by P_{xy} . If the number of image rows (columns) is not divisible by 8, the bottom row (rightmost column) is duplicated as many times as needed.
2. The DCT in two dimensions [Equation (8)] is applied to each block B_i , also see java code (figure 1).

We illustrate the performance of the DCT in two dimensions by applying it to one block of 8×8 values. The results are DCT coefficients, then quantized these coefficients after that we calculate the size of the quantized image to get compression ratio (1- compressed image size/original image size). see Table 1(a,b,c,d) [1], and the java code (figure 1) to calculate 2D DCT results.

Table 1: Two -Dimensional DCT of a Block of Correlated Data [taken from Ref. [1]]

12	10	8	10	12	10	8	11	81	0	0	0	0	0	0	0	0
11	12	10	8	10	12	10	8	0	1.57	0.61	1.90	0.38	<u>1.81</u>	0.20	<u>0.32</u>	
8	11	12	10	8	10	12	10	0	<u>0.61</u>	0.71	0.35	0	0.07	0	0.02	
10	8	11	12	10	8	10	12	0	<u>1.90</u>	<u>0.35</u>	4.76	0.77	<u>3.39</u>	0.25	<u>0.54</u>	
12	10	8	11	12	10	8	10	0	<u>0.38</u>	0	<u>0.77</u>	8.00	0.51	0	0.07	
10	12	10	8	11	12	10	8	0	<u>1.81</u>	<u>0.07</u>	<u>3.39</u>	<u>0.51</u>	1.57	0.56	0.25	
8	10	12	10	8	11	12	10	0	<u>0.20</u>	0	<u>0.25</u>	0	<u>0.56</u>	<u>0.71</u>	0.29	
10	8	10	12	10	8	11	12	0	<u>0.32</u>	<u>0.02</u>	<u>0.54</u>	<u>0.07</u>	<u>0.25</u>	<u>0.29</u>	<u>0.90</u>	
(a) Original data								(b) DCT coefficients								
81	0	0	0	0	0	0	0	12.29	10.26	7.92	9.93	11.51	9.94	8.18	10.97	
0	2	1	2	0	2	0	0	10.90	12.06	10.07	7.68	10.30	11.64	10.17	8.18	
0	1	1	0	0	0	0	0	7.83	11.39	12.19	9.62	8.28	10.10	11.64	9.94	
0	2	0	5	1	3	0	1	10.15	7.74	11.16	11.96	9.90	8.28	10.30	11.51	
0	0	0	1	8	1	0	0	12.21	10.08	8.15	11.38	11.96	9.62	7.68	9.93	
0	2	0	3	1	2	1	0	10.09	12.10	9.30	8.15	11.16	12.19	10.07	7.92	
0	0	0	0	0	1	1	0	7.87	9.50	12.10	10.08	7.74	11.39	12.06	10.26	
0	0	0	1	0	0	0	1	9.66	7.87	10.09	12.21	10.15	7.83	10.90	12.29	
(c) Quantized								(d) Reconstructed data (good)								

```
// java code to calculate 2D DCT for image A (8 x 8).
public class DCT2D {
    public static void main(String[] a) throws Exception{
        double PI=3.14159265;
        int A[][]={{12, 10, 8, 10, 12, 10, 8, 11},
                  {11, 12, 10, 8, 10, 12, 10, 8},
                  {8, 11, 12, 10, 8, 10, 12, 10},
                  {10, 8, 11, 12, 10, 8, 10, 12},
                  {12, 10, 8, 11, 12, 10, 8, 10},
                  {10, 12, 10, 8, 11, 12, 10, 8},
                  {8, 10, 12, 10, 8, 11, 12, 10},
                  {10, 8, 10, 12, 10, 8, 11, 12}};

        double G=0;
        int i,j,u,v;

        for(i = 0; i <= 7; i++) {
            for(j = 0; j <= 7; j++) {
                G = 0.0;
                for(u = 0; u <= 7; u++) {
                    for(v = 0; v <= 7; v++) {
                        double x = 1;

                        double y = 1;
                        if (i == 0)
                            x = 1/Math.sqrt(2.0);
                        if (j == 0)
                            y = 1/Math.sqrt(2.0);
                        G=G+(0.25*x*y*Math.cos(((PI*i)/(2*8))*(2*u + 1))*Math.cos(((PI*j)/(2*8))*(2*v + 1))*A[u][v]);

                        if(G<-1)
                            System.out.print("G["+i+"]"+"["+j+"]= "+(0)+"\n");
                        else
                            if(G>-1)
                                G=Math.round(G);
                                System.out.print("G["+i+"]"+"["+j+"]= "+(G)+"\n");}}
                    System.out.print("\n\n\n\n"); } }
            }
    }
}
```

Figure 1: java code to calculate 2D DCT

5 Wavelet Transform

This article demonstrates the compression and uncompression of a grayscale image (i.e. Medical images of CT and MRI types) and computed MSE and PSNR error values using the 'spiht' compression method. See wcompress function details in Matlab. [5]

Where, set partitioning in hierarchical trees (SPIHT) is an image compression algorithm that exploits the inherent similarities across the sub bands in a wavelet decomposition of an image. It is the powerful wavelet-based image compression method. The algorithm codes the most important

wavelet transform coefficients first, and transmits the bits so that an increasingly refined copy of the original image can be obtained progressively. For more details see Ref. [10].

The structure of the wavelet transform based compression is shown in figure 2 below: [9] [11]

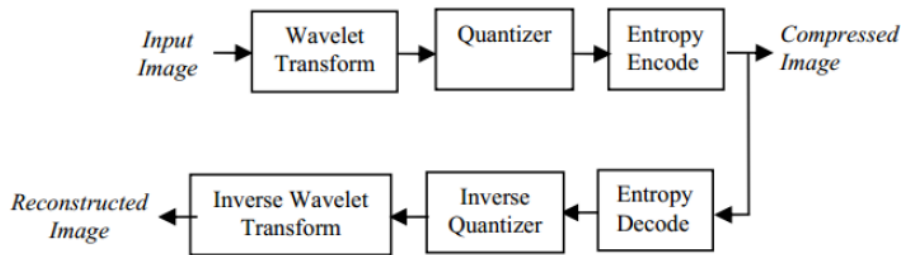
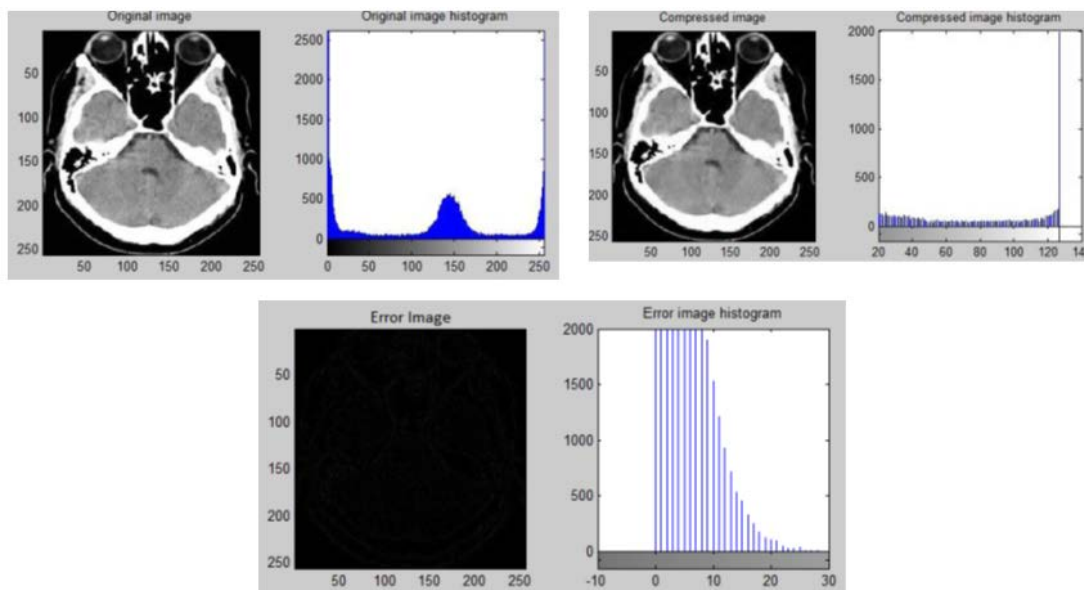


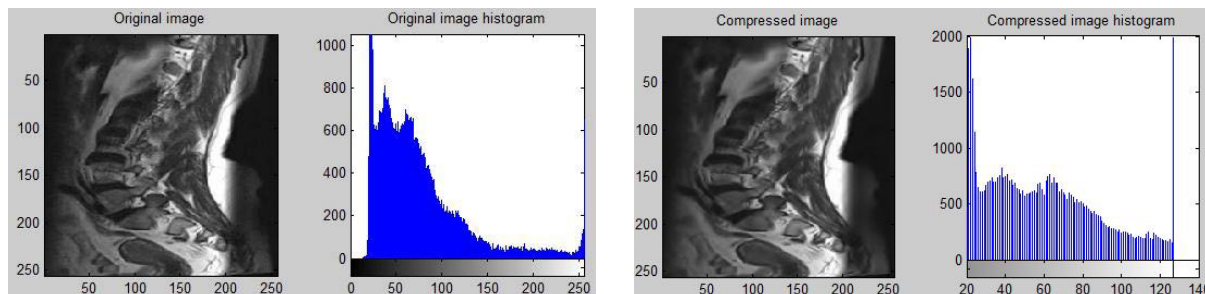
Figure 2: The structure of the wavelet transform based compression

6 The Results

6.1 Wavelet method results for CT-Scan medical image of brain (.jpg)



6.2 Wavelet method results for MRI medical image of Spine (.jpg)



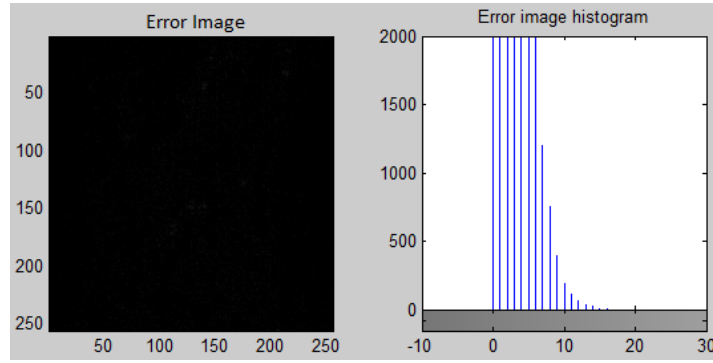

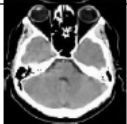
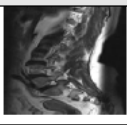
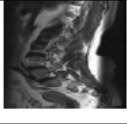
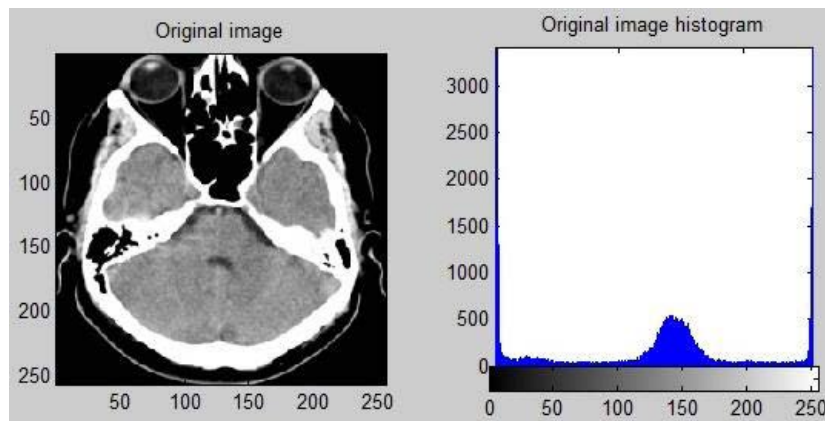


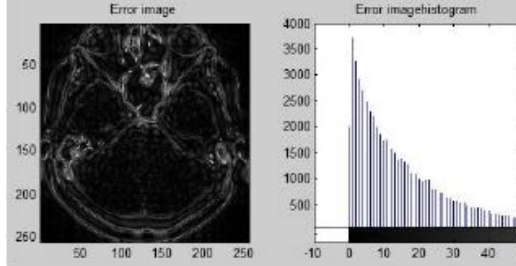
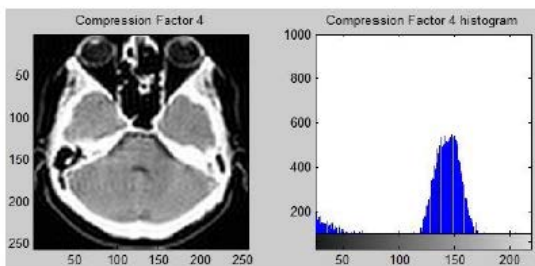
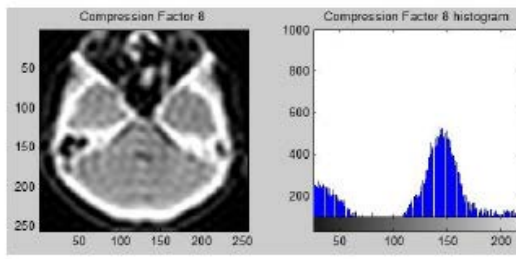
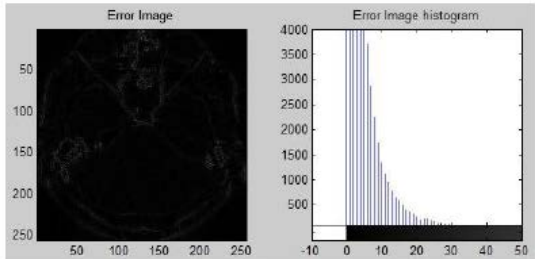
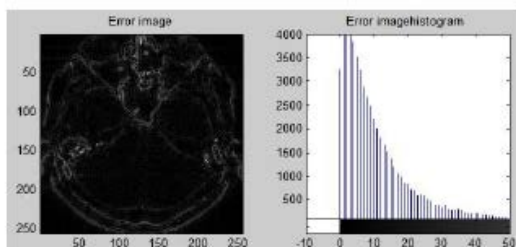
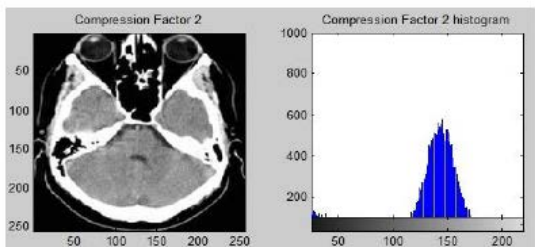
Table-1 results using Wavelet method for CT-Scan and MRI images

	Image type	Size(pixels) Resize image	Size(Kb) .jpeg	MSE	Psnr	Compression Ratio*	
1	CT -Origin medical(Brain. jpg)	256 x 256	12.9	-	-		
	Compressed Brain image	256 x 256	7.32	33.669	32.858	43.25	
2	MRI-Spine Origin	256 x 256	9.38				
	Spine- compressed	256 x 256	1.07	9.23	38.476	88.59	

*CR = (original image size - compressed image size)/original image size*100%

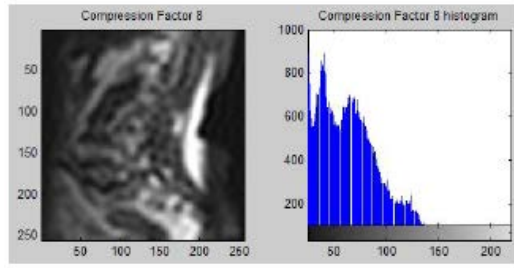
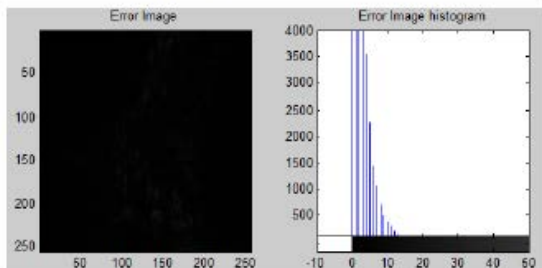
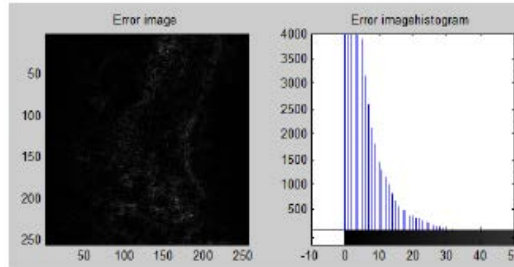
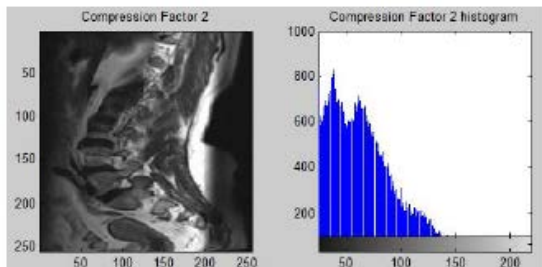
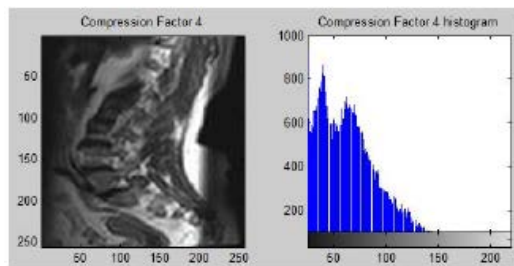
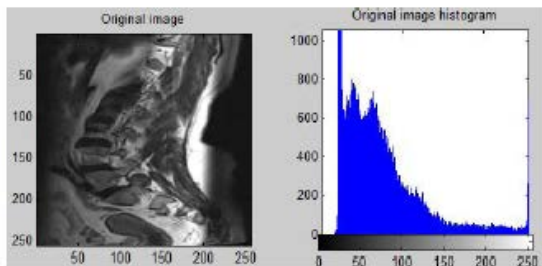
6.3 Results using Discrete Cosine Transform (DCT) method for CT Brain image





(a)

(b)



(c)

(d)

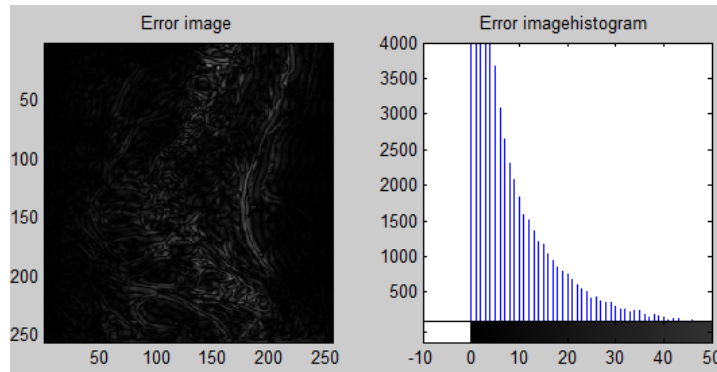



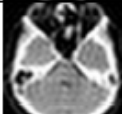
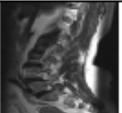
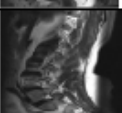
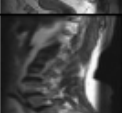
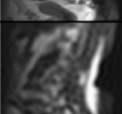


Table-2 results using DCT method for CT and MRI images

	Image type	Size(pixels)	Size(Kb) jpeg	MSE	Psnr	Compression Ratio*	
1	Origin medical-CT brain	256 x 256	12.9				
	Compression Factor 2	256 x 256	12.2	5.4649	40.755	5.4	
	Compression Factor 4	256 x 256	10.2	13.5615	36.8077	20.93	
	Compression Factor 8	256 x 256	7.77	23.5266	34.4152	39.76	
				MSE	Psnr		

2	Spine image origin	256 x 256	9.38	-	-	-	
	Spine- Compression Factor 2	256 x 256	9.05	2.0287	45.0586	3.5	
	Spine- Compression Factor 4	256 x 256	7.32	5.8727	40.4424	21.96	
	Spine- Compression Factor 8	256 x 256	5.46	9.9926	38.134	41.79	

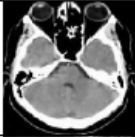
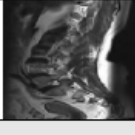

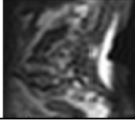
7 Conclusion

According to the results in the table below, the conclusions are:

- 1- Wavelet compression method for both CT and MRI images is better than DCT compression method.
- 2- In MRI images, the compression ratio is high using wavelet method, it is twice than the compression ratio of CT-Scan images.

- 3- The MSE and PSNR are approximately the same in both compression methods (wavelet and DCT) for MRI and CT scan images.
- 4- The quality of compressed images using wavelet is better than using DCT method and the error image is less than the error image using DCT method

Note: the size of all images used in this article is resize to 256 x 256 pixels

Image Type	Size(pixels)	Compression method	Compression ratio	MSE	PSNR	
CT	256 x 256	Wavelet	43.25	33.669	32.858	
		DCT	39.76	23.526	34.415	
MRI	256 x 256	Wavelet	88.59	9.23	38.476	
		DCT	41.79	9.99	38.134	

REFERENCES

- [1]. David Salomon A Concise Introduction to Data Compression. © Springer-Verlag London Limited 2008.
- [2]. Khalid Sayood Introduction to Data Compression.3rd edition. 2006 by Elsevier inc,
- [3]. Lossless and Transparency Encoding in WebP, https://developers.google.com/speed/webp/docs/webp_lossless_alpha_study?csw=1
- [4]. Prabhakar, Telagrapu, V .Jagan Naveen etal. Image compression using DCT and wavelet transformations, international journal of signal processing, image processing and pattern recognition , vol.4,No.3, September ,2011.
- [5]. Matlab the language of technical computing, version 7.6.0.324(R2008a), wcompress function.
- [6]. <http://eeweb.poly.edu/~onur/lectures/lecture2.pdf>. Onur G. Guleryuz, Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, NY 18
- [7]. David Salomon, Giovanni Motta, D. Bryant. Handbook of Data Compression, 5th edition. Springer-Verlag London Limited 2010.
- [8]. Rao K.R., Yip P.C. eds. The transform and data compression handbook 2001.CRC Press. Boca Raton London New York Washington, D.C

- [9]. Digital image processing using MATLAB by Rafael C. Gonzalez ,Richard E. Wood Steven L.Eddins,2004, Pearson Prentice Hall

- [10]. <http://www.cipr.rpi.edu/research/SPIHT/spiht1.html>

- [11]. M. Mozammel Hoque Chowdhury and Amina Khatun. Image Compression Using Discrete Wavelet Transform. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012