

# A COMPARATIVE STUDY OF DCT AND DWT IMAGE COMPRESSION TECHNIQUES COMBINED WITH HUFFMAN CODING

Ashraf Maghari

(Received: 11-Apr.-2019, Revised: 27-May-2019, Accepted: 9-Jun.-2019)

## ABSTRACT

Image compression techniques have been widely used to store and transmit data which requires storage space and high transfer speed. The explosive growth of high-quality photos leads to the requirement of efficient technique to store and exchange data over the internet. In this paper, we present a comparative study to compare between the Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) algorithms in combination with Huffman algorithm; DCT-H and DWT-H. The comparison is based on five factors: Compression Ratio (CR), Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) and compression/decompression time. The experiments are conducted on five BMP gray-scale file images. We found out that DWT-H coding is comparable to DCT-H coding in term of CR and outperforms DCT-H in terms of MSE, PSNR and SSIM. The CR average results of the five test images for DCT-H and DWT-H are 2.36 and 3.17, respectively. Moreover, DCT-H has the average results of MSE = 13.19, PSNR = 37.15 and SSIM = 0.76, while WDT-H has the average results of MSE = 4.54, PSNR = 42.5 and SSIM = 0.85. On the other hand, DCT-H outperforms DWT-H in term of execution time for compression and decompression. DCT-H has an average compression time of 0.358s and an average decompression time of 0.122s, while WDT-H has 2.38s compression time and 2.13s decompression time.

## KEYWORDS

Image compression, DCT, DWT, Huffman coding, PSNR, SSIM.

## 1. INTRODUCTION

In the last decades, the demand for image compression has increased, particularly after the great development of camera devices and the proliferation of high-quality image and video exchange over the internet [1]. The applications based on images, such as medical imaging, cameras and video-on-demand systems contain large amounts of data to transmit. The main idea behind image compression is to reduce the size of the image in order to minimize the storage space and increase the transmission speed [2]. Data Compression (DC) is a technique that transforms the original data to its compact form by the recognition and utilization of patterns existing in the data. It should be able to inverse the data very approximately to original data [3]. DC techniques are crucially used in many real time applications, like satellite imagery, Geographical Information Systems (GISs), graphics, Wireless Sensor Networks (WSNs), ...etc. For example, an image without compression contains 1024 pixels of 24 bit with a size of 3MB. The image needs a transmission time of 7 minutes with an ISDN line and 64 Kbit/s. If the image is compressed at 10:1 compression ratio, its size drops to 300 KB and needs below 6 seconds transmission time [4]. The purpose of compression is to eliminate data redundancy and irrelevancy in order to decrease the storage and transmission costs while maintaining good quality.

There are two different categories of image compression techniques; lossy and lossless techniques. Lossy compression is a technique that transforms the original data into more efficient data and cannot reconstruct the original data without errors. It is also called transform coding [3]. Lossless compression is a technique that processes the original data without losing any information [5]. Examples of lossy techniques include Discrete Hartley Transform (DHT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), ...etc. [4]. Lossless compression techniques include Huffman coding, LZW, arithmetic coding, ...etc.

Transform coding converts the input data into another kind of representation in which the transformed values (coefficients) are encoded by compression techniques. DCT and DWT are the most widely used

transform coding techniques which have the ability to compress data using a smaller number of coefficients [3]. The major drawbacks, however, of DCT are preface of fake contouring effects and blocking artifacts at higher compression. Similarly, DWT requires huge computational resources.

Mostly, transform-based image compression algorithms follow three step processes: transformation, quantization and entropy encoding. The quantization may be scalar or vector and the quantized transformed coefficients are entropy-coded [6]. On the other hand, Huffman coding as a lossless algorithm has a good compression ratio and a fast compression time. A previous study showed that Huffman coding is better than RLE and Delta encoding techniques in term of compression time [7]. Huffman replaces fixed-length code words with variable-length code words, where low-frequency symbols are expressed with longer encodings and high-frequency symbols are expressed with shorter encodings. In this paper, we have analyzed and implemented DCT image compression technique combined with Huffman coding (DCT-H) and DWT image compression technique combined with Huffman coding (DWT-H). The two combined techniques have been evaluated using different performance metrics.

There are many comparative studies which have been conducted to compare between compression techniques using DCT and DWT [8], [9]. However, their experiments were conducted on only one or two images with different sizes and did not take into account the compression and decompression time. The main objective of this work is to empirically compare between DWT and DCT in combination with Huffman coding in terms of five factors: Compression Ratio (CR), Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) as well as compression and decompression time. This study will be helpful for upcoming researchers to approximately select and develop the required algorithms to be used in a particular situation. Our experiments are conducted using five different images with the same size (256×256). The size of (256×256) pixels is chosen to ease the implementation of DCT and WDT, where the image components are divided into (8×8) blocks for DCT implementation.

The remainder of the paper is organized as follows. We present a background in Section 2. The related work is described in Section 3. Section 4 presents the methodology used and the experimental setup. Results and discussion are explained in Section 5. Section 6 concludes our findings of this research.

## **2. BACKGROUND**

### **2.1 Lossy and Lossless Image Compression**

Lossless compression is commonly used with text files, where the input and output data are the same, before and after the compression process, while lossy compression is commonly used with different types of data, such as image, video and audio data. At lossy compression the input data and output data are not the same, which means that there is a loss of some data at the compression process so that when we perform the decompression process, we obtain a closer approximation of the input data [7].

### **2.2 Discrete Cosine Transform (DCT)**

Discrete Cosine Transform (DCT) is one of the lossy compression approaches which are commonly applied in photo compression as jpeg compression. DCT is very approximate to Discrete Fourier Transform (DFT), but DCT includes a basis of cosine functions and real number co-efficients. Both DFT and DCT transform data from a spatial domain into a frequency domain. An inverse function is used to reconstruct the image back [10]. The basic idea of DCT is to convert a signal into basic frequency components. The image is divided into several blocks. Then, the sum of cosine functions on different frequencies can be mathematically used to express each block of an image. For example, Joint Photographic Experts Group (JPEG) [11] is a well-known compression scheme based on DCT. For DCT, the image is first transformed into an appropriate format for image compression. All the image components are divided into (8×8) blocks. Every block is encoded using discrete cosine

transformation, which is used to exploit the spatial correlation between the pixels [2]. The following formulae in Equation (1) and Equation (3) describe the transformation function and its inverse.

### 2.2.1 Definition of DCT

Given a function  $f(i, j)$  (an  $M \times N$  block of an image), the 2D DCT transforms it into a new function  $F(u, v)$ . The general definition of the transform is:

$$F(u, v) = \frac{2C(u)C(v)}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \cos \frac{(2i+1)u\pi}{2M} * \cos \frac{(2j+1)v\pi}{2N} f(i, j) \quad (1)$$

where  $i, u = 0, 1, \dots, M-1, j, v = 0, 1, \dots, N-1$  and the constants  $C(u)$  and  $C(v)$  are defined as:

$$c(k) = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } k = 0 \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

The image block dimension is defined as  $M=N=8$ .

### 2.2.2 2D Discrete Cosine Transform (2D DCT)

The 2D transform function is defined as follows:

$$F(u, v) = \frac{C(u)C(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)u\pi}{16} * \cos \frac{(2j+1)v\pi}{16} f(i, j) \quad (3)$$

where  $i, u, j, v = 0, 1, \dots, 7$  and the constants  $C(u)$  and  $C(v)$  are determined by Equation (2).

2D Inverse DCT (2D IDCT):

The 2D inverse function is defined as follows:

$$f(i, j) = \sum_{u=0}^7 \sum_{v=0}^7 \frac{C(u)C(v)}{4} \cos \frac{(2i+1)u\pi}{16} * \cos \frac{(2j+1)v\pi}{16} F(u, v) \quad (4)$$

where  $i, u, j, v = 0, 1, \dots, 7$  and the constants  $C(u)$  and  $C(v)$  are determined by Equation (2).

After the transformation, most of the information is intense to a few low-frequency components. These components are then quantized in order to decrease the number of bits required to represent the image. The quantization step produces many zero components in the bit stream. Therefore, entropy encoding, such as Huffman encoding, can achieve better compression.

## 2.3 Image Compression by Wavelet Transform

Discrete Wavelet Transform (DWT) is another popular technique for lossy compression, which is commonly applied in photo-video compression area. The main idea of DWT is to decompose a signal into a set of basic functions which transforms a discrete time signal into a discrete wavelet representation. DWT is mostly based on two-dimensional discrete wavelet transform (2DDWT). It uses one-dimensional discrete wavelet transform (1D-DWT) row-wise to get low (L) and high (H) bands. Then, 1D-DWT can be applied column-wise to get four sub-bands, such as LL, LH, HL and HH. Furthermore, each of these four bands can then be divided into four sub-bands. Some schemes based on wavelet transform are discussed in [2].

On the other hand, the inverse of DWT (IDWT) should be capable of inversion at minimum approximately to the original signal. Equation (5) shows the expanded and translated family of wavelets  $\psi$  [12] which is an orthonormal basis of  $L^2(\mathbb{R})$ :

$$\left\{ \psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi \left( \frac{t - 2^j n}{2^j} \right) \right\}_{(j,n) \in \mathbb{Z}^2} \quad (5)$$

where  $\mathbb{Z}$  represents the set of integers. The scale of translations is changed along with the overall scale  $2^j$ , so as to keep movement in the lower resolution image in proportion [12]. The simplest wavelet transform is called Haar Wavelet Transform, which forms averages and differences of a given sequence of numbers. Theory of Haar WT algorithm can be found in [13].

## 2.4 Lossless Entropy Coding Techniques

Coding is a technique that assigns binary digits to the transformed and quantized output. Variable-length coding, also known as source coding, is used to have a less average length of bits per pixel for the image, in which it replaces each input symbol with a specific code word. The entropy represents the average amount of information contained per symbol in the source  $S$ . It is the average number of bits needed to represent the symbols in the source  $S$ . The entropy  $H(S)$  of an information source with alphabets  $S = \{s_1, s_2, \dots, s_n\}$  is:

$$H(S) = \sum_{i=1}^n p_i \log_2 \frac{1}{p_i} \quad (6)$$

where  $p_i$  is the probability that a symbol  $s_i$  will occur in  $S$ ,  $\log_2 \frac{1}{p_i}$  indicates the amount of information contained in  $s_i$ , which corresponds to the number of bits needed to encode  $s_i$ . The most widely used entropy coding methods in data compression literature are Huffman coding and Shannon–Fano coding.

## 2.5 Huffman Coding

Huffman coding is a famous compression coding technique, which replaces fixed-length code words with variable-length-code words, where low-frequency symbols are expressed with longer encodings and high-frequency symbols are expressed with shorter encodings [14]. It is a type of optimal prefix code, which is widely employed in lossless data compression. Huffman coding is uniquely decodable and consists of two components such as constructing Huffman tree from input sequence and traversing the tree to assign codes to characters. It is noticed that the two nodes at the same level of the tree will have the same code lengths. Huffman coding is still popular because of its simpler implementation, faster compression and lack of patent coverage. Several compression methods, like Deflate, JPEG, MP3, ...etc. use Huffman code as the back-end technique [3].

## 3. RELATED WORK

There are many studies which have been conducted on image compression techniques. Hnesh and Demirel [15] proposed a new hybrid model that integrated DWT and DCT-SVD techniques. They applied their model on jpeg images and compared their results to other experiments that they performed on original DCT and DCT-SVD separately. They reported that their model results were better than DCT and DCT-SVD in term of compression ratio.

Other efforts [16] present a new framework for compression of medical images based on compressive sensing (CS). The researchers proposed a framework that combined the discrete cosine transform (DCT) as well as the discrete wavelet transform (DWT) with CS. Their framework was applied on CT and MRI medical images. Their results showed that the CS-and DWT-based compression technique performs better than other compression techniques in terms of compression ratio and PSNR for MRI and CT images.

Sharma and Kaur [17] proposed a hybrid model, using DWT, DCT and Huffman coding, for the purpose of medical image compression. Their hybrid model aims to achieve higher compression rates by first applying DWT and DCT on RGB components. Then, quantization was applied to calculate probability index for each unique quantity in order to find out the unique binary code for each unique symbol to encoding. Huffman coding was then applied on the quantized components. Their results showed that their hybrid model can effectively improve the compression ratio and PSNR.

All of the above studies applied different algorithms in image compression based on several factors to achieve a higher compression ratio. There are also some comparative studies that have been conducted to compare compression techniques. Barbhuiya et al. [8] conducted a comparative study on image compression using DCT (Discrete Cosine Transform) and DWT (Discrete Wavelet Transform). They also presented a compression algorithm using DWT and Inverse DWT and applied it on two image formats; JPEG (Joint Photographic Experts Group) and PNG (Portable Network Graphics) color images. Their results showed that the DWT technique outperforms the DCT technique in terms of Compression Ratio, Mean Square Error (MSE) and Peak Signal to Noise Ratio (PNSR).

Saroya and Kaur [9] proposed a comparative study between DCT and DWT algorithms based on the parameters mean square error (MSE) and peak signal to noise ratio (PSNR). Their experiments were conducted on one image (Lena) without showing any detailed results related to PSNR or MSE. Most of reviewed comparative studies did not report detailed results and did not take into account the compression and decompression time. This work, however, compares DWT and DCT in combination with Huffman coding in terms of four factors; Compression Ratio (CR), Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) as well as compression and decompression time.

#### 4. METHODOLOGY AND EXPERIMENTAL SETUP

In this section, we present the idea of image reconstruction, the compression and decompression models, discuss the performance metrics and then explain the experimental setup.

##### 4.1 Image Compression

The principle of image compression algorithms is reducing the redundancy in the image data and (or) producing a reconstructed image from the original image with the introduction of an error that is insignificant to the intended applications. The aim here is to obtain an acceptable digital image representation, while preserving the essential information contained in that particular dataset [18].

##### 4.2 Compression/Decompression Model (DCT-H)

Due to our methodology, we applied lossy and lossless algorithms on some test images. Figure 1 shows the framework followed to implement our compression/decompression model for DCT-H.

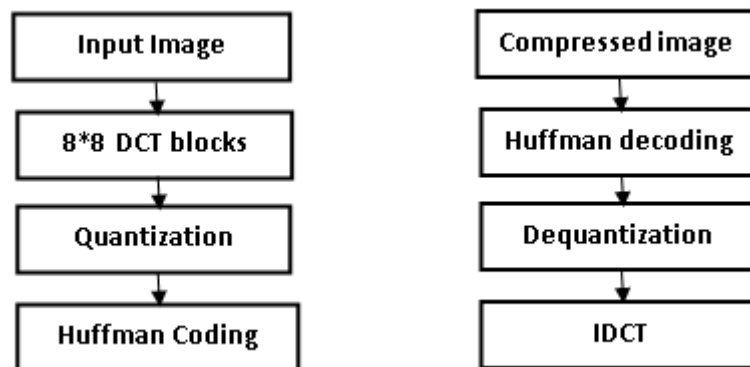


Figure 1. DCT-H, compression model (left) and decompression model (right).

Regarding the compression steps shown in Figure 1 (left), the input image is first loaded. Then, the loaded image is divided into blocks with 8\*8 pixels. Forward 2D DCT function is applied on each block to obtain the DCT coefficients which are the 64 basis functions of the 8\*8 pixel blocks from the original image. An example of 8\*8 image luminance block, with a range of 8-bit values  $f(i, j)$  [0, 255], is illustrated in Figure 2 (left) [12].

70	70	100	70	87	87	150	187	-80	-40	89	-73	44	32	53	-3
85	100	96	79	87	154	87	113	-135	-59	-26	6	14	-3	-13	-28
100	85	116	79	70	87	86	196	47	-76	66	-3	-108	-78	33	59
136	69	87	200	79	71	117	96	-2	10	-18	0	33	11	-21	1
161	70	87	200	103	71	96	113	-1	-9	-22	8	32	65	-36	-1
161	123	147	133	113	113	85	161	5	-20	28	-46	3	24	-30	24
146	147	175	100	103	103	163	187	6	-20	37	-28	12	-35	33	17
156	146	189	70	113	161	163	197	-5	-23	33	-30	17	-5	-4	20

Figure 2. An 8\*8 block  $f(i, j)$  from an image (left) and its corresponding DCT coefficients  $F(u, v)$  (right).

As shown in Figure 2 (right), except the DC and the first few AC components representing low spatial frequencies, most of the DCT coefficients  $F(u, v)$  have small magnitudes. This is because the pixel values in this block contain few high-spatial-frequency changes. We can see that the most information is accurately described by the first few components of the DCT coefficients. Therefore, the remaining components can be coarsely quantized, or even set to zero, with little signal distortion.

In the third step, quantization is applied on the DCT coefficients in order to minimize the number of output values to a high smaller set and discard the less important components. This example uses the luminance quantization table shown in Figure 3. The quantization step leads to high numbers of zeros repeated as shown in Figure 4. Therefore, in the last step of encoding, Huffman encoding can be applied to achieve better compression.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 3. The luminance quantization table.

-5	-4	9	-5	2	1	1	0
-11	-5	-2	0	1	0	0	-1
3	-6	4	0	-3	-1	0	1
0	1	-1	0	1	0	0	0
0	0	-1	0	0	1	0	0
0	-1	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 4. Quantization on DCT coefficients.

-80	-44	90	-80	48	40	51	0
-132	-60	-28	0	26	0	0	-55
42	-78	64	0	-120	-57	0	56
0	17	-22	0	51	0	0	0
0	0	-37	0	0	109	0	0
0	-35	55	-64	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 5. De-quantization results.

In our experiments, however, we use a uniform scalar quantizer with a scalar factor  $q = 20$ . This means that the DCT coefficients are divided by 20, which produces similar results to the results shown in Figure 4. See subsection “4.5 Experimental Setup”. Contrariwise, for the decompression model steps shown in Figure 1 (right), the compressed image file is loaded and then Huffman decoding is applied. After that, dequantization is carried out to retain the DCT coefficients as shown in Figure 5. In the last step, inverse discrete cosine transformation (IDCT) is applied to reconstruct an approximated image of the original image.

Figure 6 shows the reconstructed 8\*8 image block after applying IDCT. The figure shows that the reconstructed block is in close approximation of the original block, which generally cannot be visually noticed by the human eye in the image. To illustrate the quality of DCT compression, the error  $e = \text{original } 8*8 \text{ block (Figure 2, left) - reconstructed } 8*8 \text{ block (Figure 6)}$  is shown in Figure 7.

70	60	106	94	62	103	146	176
85	101	85	75	102	127	93	144
98	99	92	102	74	98	89	167
132	53	111	180	55	70	106	145
173	57	114	207	111	89	84	90
164	123	131	135	133	92	85	162
141	159	169	73	106	101	149	224
150	141	195	79	107	147	210	153

Figure 6. The reconstructed 8\*8 image block.

0	-1	11	4	-15	27	-6	-31
2	-14	24	-23	-4	-11	-3	29
4	16	-24	20	24	1	11	-49
-12	13	-27	-7	-8	-18	12	23
-3	0	16	-2	-20	21	0	-1
5	-12	6	27	-3	2	14	-37
6	5	-6	-9	6	14	-47	44

Figure 7. Error of DCT compression.

### 4.3 Compression/Decompression Model (DWT-H)

Similar to DCT-H, Figure 8 portrays the framework used to implement the compression/decompression model for Discrete Wavelet Transformation combined with Huffman coding (DWT-H). The objective of the wavelet transform is to decompose the input signal into components some of which can be thresholded away. Moreover, the original image can be approximately reconstructed using these components. Two-level 2D DWT is applied on the input image using the MATLAB function “dwt2(input, ‘haar’)” to obtain the most significant components that occur in the upper left side of the image (LL). Figure 9 shows the two-level DWT of 256\*256 image. LL band holds the approximated version of the original image and represents the general trend of pixel values of the input image. The other components can be efficiently encoded (after quantization) or even discarded, because they have little significant information.

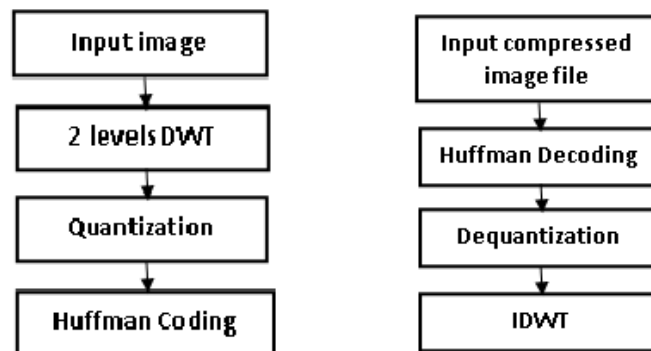


Figure 8. DWT-H, compression model (left) and decompression model (right).

After one level of 2D DWT is complete, the transformed image contains four subbands; LL, HL, LH and HH, standing for low-low, high-low and so on, as Figure 9 shows. The LL subband can be further decomposed to yield yet another level of decomposition (two-level DWT). This process can be continued until the desired number of decomposition levels is reached or the LL component only has a single element (left). Figure 10 shows an example of 16\*16 gray image block (up) and its one-level 2D DWT (down). The two-level 2D DWT is also shown in Figure 11. We can see that, except the LL components, the other components have small values which can be coarsely quantized and coded with little signal distortion. In the third step of the compression model of DWT-H shown in Figure 8 (left), quantization is applied to the DWT components in order to minimize the number of output values. Finally, in the last step, Huffman encoding is applied for compression purpose. Contrariwise, for the decompression steps shown in Figure 8 (right), Huffman decoding is applied on the compressed image. Then, dequantization is carried out to retain the decoded DWT components. In the last step, inverse wavelet transformation (IDWT) is applied to reconstruct the image.

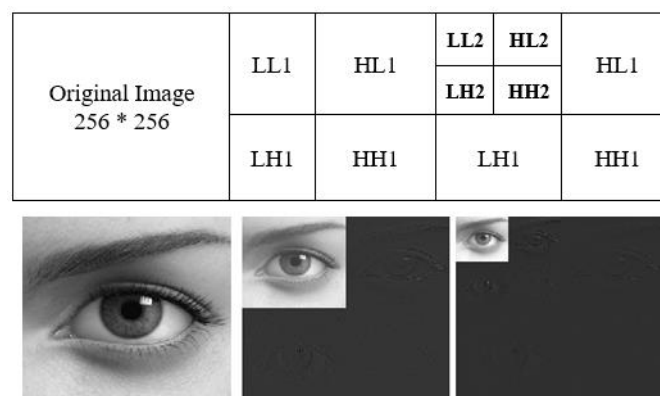


Figure 9. Two-level DWT transformation.

### 4.4 Evaluation Metrics

To evaluate the performance of both combined compression techniques; i.e., DCT-H and DWT-H,

five performance metrics have been used; Compression Ratio (CR), Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) as well as compression time and decompression time. They are described as follows:

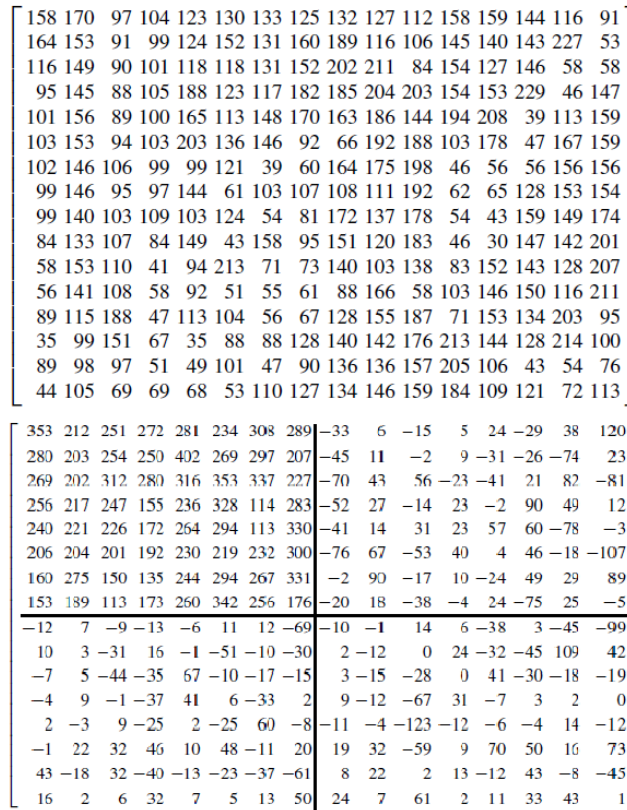


Figure 10. A 16\*16 block from a gray image (up) and its one-level 2D DWT (down).

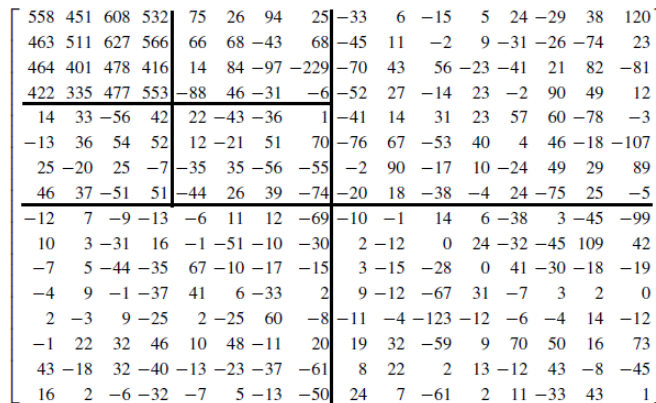


Figure 11. Two-level 2D DWT of the 16\*16 block shown in Figure 10.

**4.4.1 Compression Ratio**

The Compression Ratio (CR) is defined as the ratio of the number of bytes of the original image to that of the compressed image. It can also be described as the ratio of the size of the original image to the size of the compressed image. The formula of CR is given in Equation (7) [19]. The higher the ratio is, the better is the compression technique.

$$CR = \frac{\text{Number of bytes of original image}}{\text{Number of bytes of compresses image}} \tag{7}$$

**4.4.2 Peak Signal-to-Noise Ratio (PSNR)**

PSNR is a measure factor for the quality of the compressed image to the original image. It determines



the error between the original and the reconstructed image. A high PSNR value means a high-quality image. The objective of this factor is to measure partially the human visual response to image quality [20]. The mathematical expression of PSNR is given in Equation ( ) [18].

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (8)$$

where:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I_0(i, j) - I_r(i, j)\|^2$$

$m, n = \text{image size} > 0$ ,  $I_0 = \text{original value}$  and  $I_r = \text{compressed value}$ .

The Mean Square Error (MSE) [19] is the cumulative squared error between the original image and the compressed image. A low value of MSE means a higher value of PSNR. A high PSNR provides a better image quality after the reconstruction of the image.

#### 4.4.3 Structural Similarity Index Measure (SSIM)

SSIM is a well-known quality metric used to measure the structural distortion between two images. It was developed by Wang et al. [21] and is recently used by many researchers [22], [23], [24] for image quality assessment. As opposed to PSNR, some studies revealed that PSNR performs badly in discriminating structural content in images, since various types of degradations applied to the same image can yield the same value of PSNR [25]. Generally, the value of PSNR can be predicted from SSIM and *vice-versa* [26]. SSIM is designed to compare between the original and the distorted images ( $x, y$ ) using three factors: luminance, contrast and structural factors. It is defined as:

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (9)$$

$$\text{where, } l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \text{ and } s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

where  $\mu_x, \mu_y, \sigma_x, \sigma_y$ , and  $\sigma_{xy}$  are the local means, standard deviations and cross-covariance for images  $x$  and  $y$ .

The first term  $l(x, y)$  is the luminance comparison function, which measures the closeness of the two images' mean luminance ( $\mu_x$  and  $\mu_y$ ). The second term  $c(x, y)$  is the contrast comparison function, which measures the closeness of the contrast of the two images. Here, the contrast is measured by the standard deviation  $\sigma_x$  and  $\sigma_y$ . The third term  $s(x, y)$  is the structure comparison function, which measures the correlation coefficient between the two images  $x$  and  $y$ . Note that  $\sigma_{xy}$  is the covariance between  $x$  and  $y$ . The positive values of the SSIM index are in the range [0-1]. A value of 0 means no correlation between images and 1 means  $x = y$ . The parameters  $\alpha, \beta$  and  $\gamma$  indicate the importance of the three components, which are set to 1.  $C_1, C_2$  and  $C_3$  are constants and  $C_3 = C_2/2$ . So, the SSIM simplifies to:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (10)$$

#### 4.5 Experimental Setup

We have used MATLAB to conduct the experiments on a CPU @ 2.50GHz with an Intel(R) Core

(TM) i5-2450M processor. There are different images used for experimentation. We choose as examples five different BMP gray images to conduct our experiments. BMP (BitMap) is standard file format for Microsoft Windows, which can be stored uncompressed. The five gray images have the same size of  $256 \times 256 = 65536$  bytes (66 KB). The size of  $256 \times 256$  pixels is chosen to ease implementing DCT and WDT, where the image is split into  $8 \times 8$  blocks, which are then transformed into the frequency plane using Fast Discrete Cosine Transform (FDCT). For DWT, the 2D input image of size  $256 \times 256$  is subjected to two-level decomposition using discrete Haar wavelet transform function. The scalar factor ( $q$ ) for uniform quantization is set to 10, which affects the quantization steps for Huffman coding. Quantization is the main source of losing information. So, we choose a

small value for  $q$  to obtain a small MSE. For DCT, a block image size of  $8 \times 8$  is used with a scalar factor  $q = 20$ . There are no precise rules for selecting the scalar factor  $q$  values. The trade-off between quality and CR can be controlled by the scalar factor  $q$  which will define the size of the frequency components [18]. Different scalar values (10 and 20) were chosen for DWT and DCT during the evaluation process to have suitable compression ratio and MSE. To determine the execution time for compression and decompression processes, we use `tic` and `toc` MATLAB functions together in order to measure the amount of time MATLAB takes and display the time in seconds.

## 5. RESULTS AND DISCUSSION

### 5.1 Results

Figure 12 shows the test images and their corresponding resulting compressed images using DWT-H and DCT-H. The experimental results with the comparison factors have been arranged in Table 1. Table 1 shows the results of the five comparison factors obtained by applying DCT and Huffman coding. It shows the compression ratio (CR), PSNR, MSE, SSIM, compression time and decompression time. Table 2 shows the same five factors for DWT combined with Huffman coding (DWT-H). Regarding the performance metrics, the results show that DWT-H outperforms DCT-H in that DWT-H produced mostly higher CR, higher PSNR, higher SSIM and lower MSE than DCT-H, as graphically shown in Figure 13. Performance metric values obtained are as shown in Table 1 and 2.

In contrast, for the compression/decompression time, the results show that DCT-H is less time-consuming than DWT-H. The compression/decompression time values obtained are shown graphically in Figure 14.

Table 1. The comparison factor results of DCT-H.

Image name	CR	PSNR	MSE	SSIM	Decode Time (s)	Code Time (s)
Camera man	1.72	36.04	16.3	0.575	0.12	1.14
Rice	2.56	38.04	10.28	0.772	0.12	0.2
MRI	3.37	39.37	7.57	0.898	0.11	0.15
Eye	2.66	36.7	13.9	0.734	0.14	0.16
Tiger	1.5	35.6	17.9	0.822	0.12	0.14
<b>Average results</b>	2.36	37.15	13.19	0.76	0.122	0.358

Table 2. The comparison factor results of DWT-H.

Image name	CR	PSNR	MSE	SSIM	Decode Time	Code Time
Camera man	1.96	40.8	5.4	0.712	2.44	2.7
Rice	1.97	41.7	4.4	0.856	2.4	2.7
MRI	8.24	49.6	0.72	0.993	0.65	0.7
Eye	2.1	40.1	6.4	0.854	2.27	2.64
tiger	1.6	40.5	5.8	0.851	2.9	3.17
<b>Average results</b>	3.17	42.5	4.54	0.85	2.13	2.38

### 5.2 DISCUSSION

In this study and due to our comparison study, we applied DCT and DWT algorithms in combination with Huffman algorithm. For the DCT transform, the image is split into  $8 \times 8$  blocks which are then transformed into the frequency plane using Fast Discrete Cosine Transform (FDCT). After the transform, most of information is concentrated to a few low-frequency components. These components are then quantized in order to reduce the number of bits needed to represent the image. This quantization step will lower the quality of the image by reducing the precision of the components. Regarding the DWT algorithm, as a result of applying DWT on the input image, it is divided into four non-overlapping multi-resolution sub-bands; LL, LH, HL and HH. The sub-band LL represents the coarse-scale DWT coefficients while the sub-bands LH, HL and HH represent the fine-scale DWT coefficients. Then, sub-band quantization and coding are used to reduce the number of bits needed to represent the image. This step will lower the quality of the image but not like DCT, because in DCT, the information is concentrated on fewer components that can be affected more sensitively by

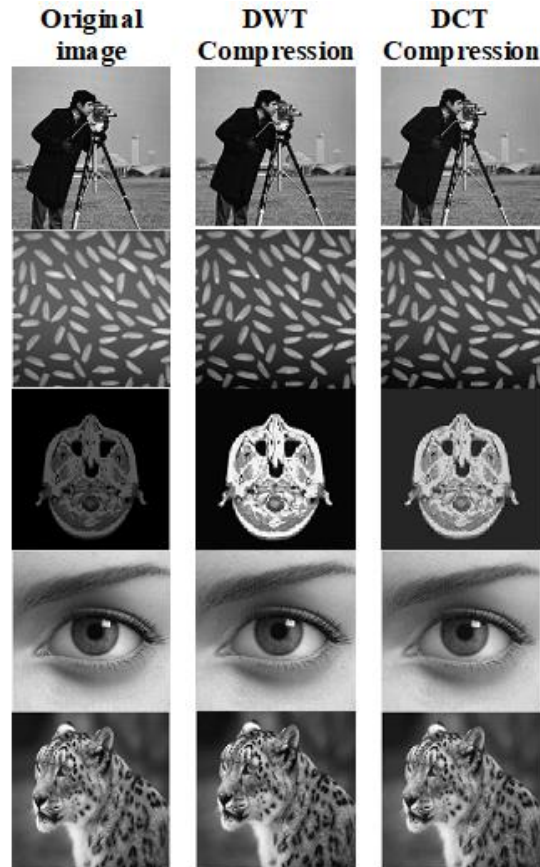


Figure 12. The five test images (Camera man, Rice, MRI, Eye, Tiger) and the corresponding compressed images using DCT-H and DWT-H.

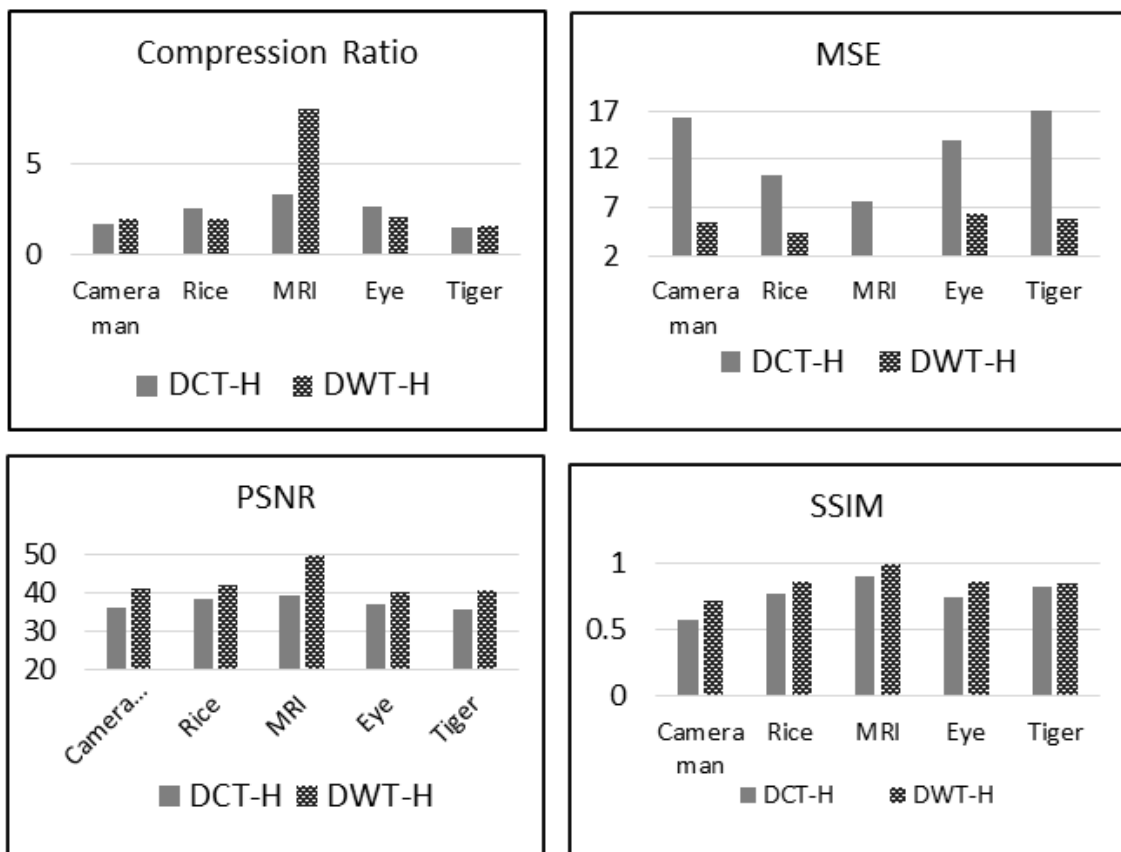


Figure 13. DCT-H and DWT-H: Compression Ratio, MSE, PSNR and SSIM.

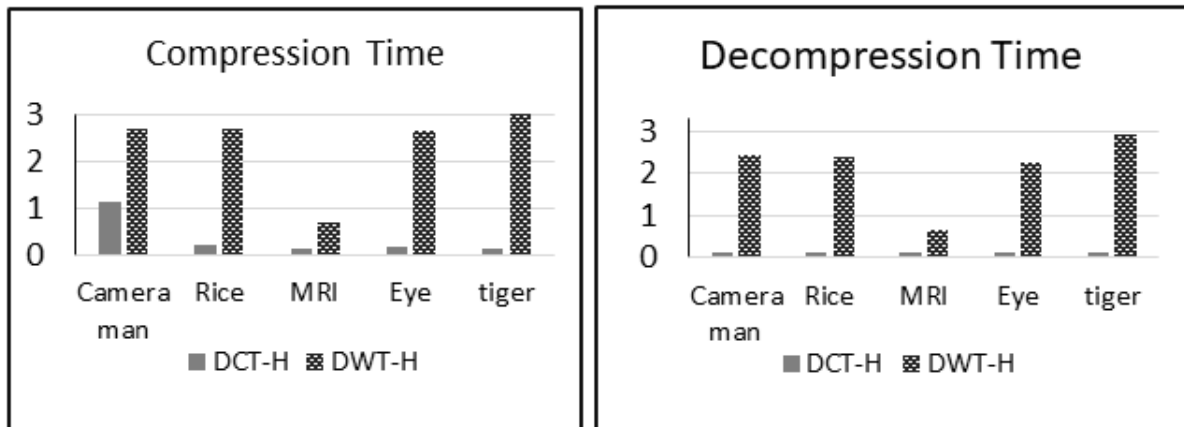


Figure 14: Compression and decompression time of DCT-H and DWT-H.

quantization than in DWT. Moreover, DWT avoids blocking artifacts that can happen by dividing the input image into blocks as in DCT [9].

The results illustrate that DWT combined with Huffman (DWT-H) coding give mostly higher compression ratio, higher PSNR and higher SSIM than DCT combined with Huffman coding (DCT-H). At the same time, we found that compression and decompression time of DCT-H was less than in DWT-H. Our appreciation for the appearance of these results is based on the extent of similarity in output values, when DCT or DWT algorithms are applied on the test images. This means that when DCT or DWT output coefficients are mostly very small, then the quantization does not cause big loss in the information. The quantization is used to reduce the number of distinct output values to a much smaller set to efficiently represent the source output using any coding techniques such as Huffman coding [12]. Using a large scalar value for quantization gives high compression ratio and low-quality image reconstruction. So, in our experiments, we used a constant small scalar value ( $q$ ) for quantization ( $q=10$  for DWT,  $q=20$  for DCT) in order to obtain small MSE and high-quality reconstruction. According to our observations, we conclude that using DWT with Huffman algorithm achieves better compression ratio, PSNR and SSIM than using DCT with Huffman algorithm. Hence, performance of DWT combined with Huffman coding is comparatively higher than that of DCT combined with Huffman coding. Our results are consistent with [8], where DWT algorithm performed much better than DCT algorithm in terms of Compression Ratio and Peak Signal-to-Noise Ratio (PSNR). On the other hand, DCT-H is less time-consuming than WDT-H.

## 6. CONCLUSION

In this paper, we performed a comparative study based on DCT and DWT in combination with Huffman coding in terms of Compression Ratio (CR), Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) as well as compression and decompression time. Our experiments were conducted on five different BMP file images; Camera man, Rice, MRI, Eye and Tiger. The experimental results showed that using DWT with Huffman algorithm (DWT-H) achieves a comparable compression ratio with higher PSNR and less MSE than DCT with Huffman algorithm (DCT-H). At the same time, we found out that DWT-H is time-consuming in terms of compression and decompression. The average CR metric value of the five images was 2.36 for DCT-H and 3.17 for DWT-H. Regarding MSE, PSNR and SSIM, the average result values of DCT-H were: MSE = 13.19, PSNR = 37.15 and SSIM = 0.76. WDT-H has the average result values of MSE = 4.54, PSNR = 42.5 and SSIM = 0.85. On the other hand, DCT-H outperforms DWT-H in terms of compression and decompression execution times. DCT-H has an average execution time of 0.358s for compression and an average execution time 0.122s for decompression, while WDT-H has 2.38s compression time and 2.13s decompression time. In future works, we will extend our experimental study to other image compression algorithms and use different coding techniques.

## REFERENCES

- [1] M. Gupta and A. K. Garg, "Analysis of Image Compression Algorithm Using DCT," *Int. J. Eng. Res. Appl.*, vol. 2, no. 1, pp. 515–521, 2012.
- [2] T. Sheltami, M. Musaddiq and E. Shakshuki, "Data Compression Techniques in Wireless Sensor Networks," *Futur. Gener. Comput. Syst.*, vol. 64, pp. 151–162, 2016.
- [3] J. Uthayakumar, T. Vengattaraman and P. Dhavachelvan, "A Survey on Data Compression Techniques: From the Perspective of Data Quality, Coding Schemes, Data Type and Applications," *J. King Saud Univ. Inf. Sci.*, 2018.
- [4] A. Mr, S. Subramanian and S. Mr, "Comparing PSNR of Different Image Transforms (DCT, DFT, DWT, DHT, DTT)," 2018.
- [5] Nashar Luthfi Sugara, T. W. Purboyo and A. L. Prasasti, "Comparative Analysis of Image Compression Using Huffman and DCT Method on JPG Image," *J. Eng. Appl. Sci.*, vol. 13, pp. 4447-4452, 2018.
- [6] G. Suseela and Y. A. V. Phamila, "Energy Efficient Image Coding Techniques for Low Power Sensor Nodes: A Review," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 2961–2972, 2018.
- [7] W. Z. Wahba and A. Y. A. Maghari, "Lossless Image Compression Techniques: Comparative Study," *Int. Res. J. Eng. Technol.*, vol. 3, no. 2, pp. 1–9, 2016.
- [8] A. H. M. J. I. Barbhuiya, T. A. Laskar and K. Hemachandran, "An Approach for Color Image Compression of JPEG and PNG Images Using DCT and DWT," *Proc. of the International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 129–133, 2014.
- [9] N. Saroya and P. Kaur, "Analysis of Image Compression Algorithm Using DCT and DWT Transforms," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 2, 2014.
- [10] T. Kumar and R. Kumar, "Medical Image Compression Using Hybrid Techniques of DWT, DCT and Huffman Coding," *Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng.*, vol. 3, no. 2, pp. 54–60, 2015.
- [11] G. K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [12] Ze-Nian Li, Drew, Jiangchuan Liu and Mark S. Drew, *Fundamentals of Multimedia*, 2<sup>nd</sup> Ed., Springer International Publishing, 2014.
- [13] C. Wang, R. Xiong, H. He, Y. Zhang and W. Shen, "Comparison of Decomposition Levels for Wavelet Transform-based Energy Management in a Plug-in Hybrid Electric Vehicle," *Journal of Cleaner Production*, vol. 210, pp. 1085–1097, 2019.
- [14] A. J. Maan, "Analysis and Comparison of Algorithms for Lossless Data Compression," *Int. J. Inf. Comput. Technol.*, vol. 3, no. 3, pp. 139–146, 2013.
- [15] A. M. G. Hnesh and H. Demirel, "DWT-DCT-SVD Based Hybrid Lossy Image Compression Technique," *Proc. of the International Conference on Image Processing, Applications and Systems (IPAS)*, pp. 1–5, 2016.
- [16] R. Kher and Y. Patel, "Medical Image Compression Framework Based on Compressive Sensing, DCT and DWT," *Biology, Engineering and Medicine*, vol. 2, no. 2, pp. 1–4, 2017.
- [17] S. Sharma and S. Kaur, "Image Compression Using Hybrid of DWT, DCT and Huffman Coding," *Int. J. Sci. Emerg. Technol. with Latest Trends*, vol. 5, no. 1, pp. 19–23, 2013.
- [18] O. Ghorbel, "DCT & DWT Image Compression Algorithms in Wireless Sensor Networks: Comparative Study and Performance Analysis," *Int. J. Wirel. Mob. Networks*, 2013.
- [19] A. Katharotiya, S. Patel and M. Goyani, "Comparative Analysis between DCT and DWT Techniques of Image Compression," *J. Inf. Eng. Appl.*, vol. 1, no. 2, pp. 9–17, 2011.
- [20] R. Monika, S. Dhanalakshmi and S. Sreejith, "Coefficient Random Permutation Based Compressed Sensing for Medical Image Compression," *Advances in Electronics, Communication and Computing*, Springer, pp. 529–536, 2018.
- [21] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli et al., "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

- [22] R. Reisenhofer, S. Bosse, G. Kutyniok and T. Wiegand, "A Haar Wavelet-based Perceptual Similarity Index for Image Quality Assessment," *Signal Processing: Image Communication*, vol. 61, pp. 33–43, 2018.
- [23] K. Ma, Z. Duanmu, H. Yeganeh and Z. Wang, "Multi-exposure Image Fusion by Optimizing a Structural Similarity Index," *IEEE Transactions on Comput. Imaging*, vol. 4, no. 1, pp. 60–72, 2018.
- [24] S. Pistonesi, J. Martinez, S. M. Ojeda and R. Vallejos, "Structural Similarity Metrics for Quality Image Fusion Assessment: Algorithms," *Image Process. Line*, vol. 8, pp. 345–368, 2018.
- [25] Z. Wang and A. C. Bovik, "Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [26] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," *Proc. of the 20<sup>th</sup> International Conference on Pattern Recognition*, pp. 2366–2369, 2010.

### ملخص البحث:

أقصد شاع استخدام تقنيات ضغط الصور لتخزين البيانات ونقلها، الأمر الذي يتطلب حيزاً تخزينياً كبيراً وسرعة نقل عالية. ويؤدي النمو السريع للصور عالية الجودة إلى تنامي الطلب على تقنيات فعالة لتخزين البيانات وتبادلها عبر الإنترنت. في هذه الورقة، نقدم دراسة مقارنة بين خوارزميات تقنيتي DCT و DWT مع استخدام ترميز هوفمان. والمقارنة في هذه الدراسة مبنية على خمسة عوامل هي: معدل الضغط (CR)، ومتوسط مربع الخطأ (MSE)، وأعلى نسب الإشارة إلى الضجيج (PSNR)، ومقياس عامل التشابه البنيوي (SSIM)، وزمن الضغط/إزالة الضغط.

أجريت التجارب على خمس صور تنتمي إلى ملفات رمادية التدرج. وبينت النتائج أن الترميز باستخدام تقنية DWT يضاهي مثيله باستخدام تقنية DCT من حيث معدل الضغط (CR)، ويتقدم عليه فيما يتعلق بمتوسط مربع الخطأ وأعلى نسب الإشارة إلى الضجيج ومقياس عامل التشابه البنيوي. وكانت نتائج الصور الخمس المفحوصة فيما يخص معدل الضغط في المعدل 2.36 و 3.17 على الترتيب لكل من تقنية DCT وتقنية DWT. علاوة على ذلك، كان معدل النتائج لتقنية DCT كمايلي:  $MSE=13.19$ ؛  $PSNR=37.15$ ؛  $SSIM=0.76$ . وعلى الجانب الآخر، كان معدل النتائج لتقنية DWT على النحو الآتي:  $MSE=4.54$ ؛  $PSNR=42.5$ ؛  $SSIM=0.85$ .

من جانب آخر، تتفوق تقنية DCT على تقنية DWT من حيث زمن تنفيذ الضغط وإزالة الضغط، بحيث كان معدل النتائج لتقنية DCT 0.358 ثانية لزمن الضغط و 0.122 ثانية لزمن إزالة الضغط؛ في حين كان زمن الضغط وزمن إزالة الضغط 2.38 ثانية و 2.13 ثانية على الترتيب لتقنية DWT.

