Computer Science

Topic: Image Compression

Research Question

"To what extent do wavelet-based compression algorithms perform better than DCT-based compression algorithms while compressing medical images?

Word Count: 3993

Contents

1.0 Introduction

A vital component of contemporary healthcare is medical imaging, which gives patients useful information for diagnosis and therapy. Healthcare professionals, however, face substantial challenges in storing, retrieving, and transmitting the vast amounts and complexity of medical pictures produced by contemporary imaging technologies. To overcome these difficulties, image compression techniques reduce the storage and transmission needs for medical images without compromising their quality. In the medical field, two commonly used compressional algorithms are wavelet based algorithms and DCT based compression.

1.1. Worthiness of this investigation

This investigation will address the question of whether wavelength-based compression algorithms perform better than DCT based compression algorithm while compressing medical images. The main objective of the investigation, however, is for medical companies to use the findings of this analysis and utilise the better compression algorithm for medical images.

1.2. What is an image?

An image is a picture composed of an array of pixels. These arrays of pixels control an image's clarity, also known as resolution. These pixels have a positive correlation with the resolution hence, as the amount of pixels increases, the clarity of the image will also increase. For example, an image that is high quality at 4k (2,160 pixels tall and 3,840 pixels wide) would usually use 25 megabytes (MB) of storage, while an HD image (1020 pixels tall and 1980 pixels wide) would only use one MB of storage. The storage taking high amounts of storage would mean that the image user will have to sacrifice their storage in return for a clear image. This problem also accounts for companies and corporations.

1.3 How are medical images different from ‘other’ images?

Medical images are used to diagnose and treat patients, typically in medical settings such as hospitals or clinics, while standard images are generally used for artistic or entertainment purposes. Medical images usually contain more detailed information than regular images as they are taken with specialized equipment to provide a more detailed look at the body. Medical images are also often processed using specialized software to enhance the image or to provide additional information.

1.4 Importance of storing medical images

Hospitals and other medical industry companies will have to store the medical images of their patients. The images have to be stored as they can be used to identify any anatomical and physiological abnormalities, chart the progress of treatment, and provide clinicians with a database of regular patient scans for later reference (Tech Target, by Megan Charles). However, the images stored take up much storage; hence medical companies have to finance many storage units to keep the current and future patients' images. To optimize the amount of storage used, medical companies could compress images that do not need high resolution to identify a diagnosis for a patient and decrease its resolution to save space and money.

1.5 Types of Compressions

Most commonly used compression for any file is Lossless and Lossy Compression.

Lossless compression reduces the file size of an image without sacrificing any of the original information. Images that must be stored or transmitted accurately, without losing quality, benefit from this type of compression.

On the other hand, Lossy compression is a technique that reduces the file size of an image by discarding some of the information from the original image. This type of compression can achieve a smaller file size, but the image's quality is sacrificed.

1.6 Why lossy compression is used?

Lossy compression is often better than lossless compression in scenarios where file size is more significant than image quality. Since the main goal for medical services is to help patients whilst being cost efficient, lossy compression is the most suitable compression technique for medical companies. This is because lossy compression can significantly reduce the size of an image file. While still providing an acceptable level of image quality.

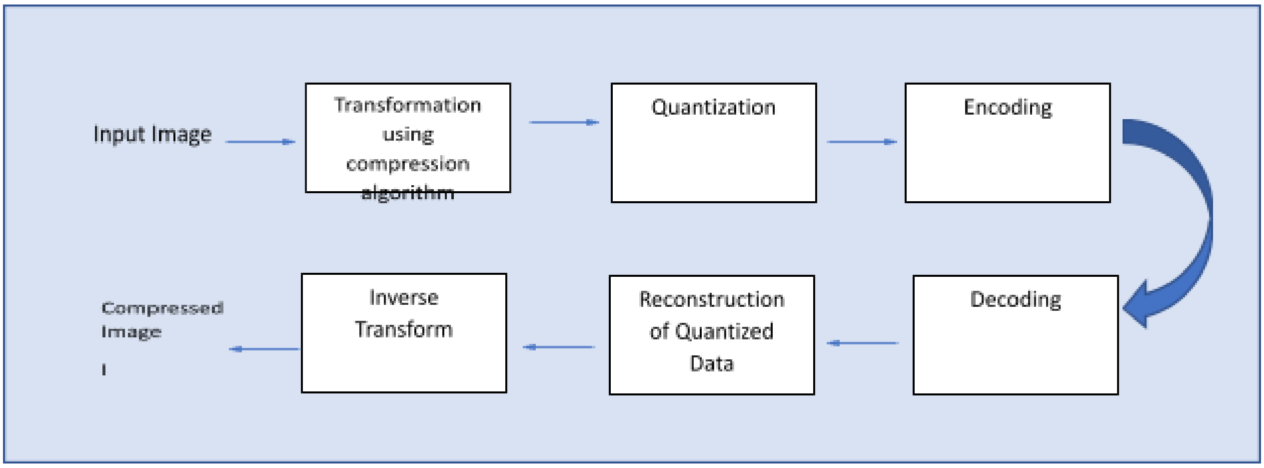


Figure 1.6 Lossy compression workflow

1.7 Importance of DCT based and wavelet based algorithms.

DCT (Discrete Cosine Transform) and wavelet-based algorithms are commonly used compression algorithm techniques. These techniques are significant because they allow for efficient compression of digital media, which makes it possible to store and transmit large amounts of data.

1.8 Plan of investigation

Methodology:

- Explanation of the implementation details of the two algorithms
- Description of the dataset and the evaluation metrics to be used for the comparison
- Description of the code for the analysis to be complete

Analysis of the data:

- Presentation of the results, including compression ratios and image quality measures.

- Comparison and analysis of the results to determine which algorithm performs better for medical image compression.

Evaluation of findings:

- Summary of the key findings from the data
- Reflection on the advantages, disadvantages and the limitations of the compression algorithms.
- Final thoughts on the compression algorithms and which algorithm is better for compression of medical images.

2.0 Background

2.1 How do lossy and lossless compression algorithms work?

Lossy compression algorithms function by identifying patterns in the data that can be approximated or predicted with a lower level of accuracy without significantly affecting the final output quality. These algorithms typically rely on techniques such as quantization, which involves reducing the precision of specific data values. In addition, they use predictive coding, where neighbouring pixels in an image are used to predict a pixel's value based on the importance of adjacent pixels.

Lossless algorithms are primarily used for text compression and other data types where it is essential to maintain the content's quality. They work by identifying and exploiting patterns and redundancies in the data to represent it more efficiently. Lossless algorithms are primarily used for data types where it is essential to maintain content's quality.

2.2 Why lossy compression is chosen for the medical images.

Lossy compression is preferred for image compression relative to lossless compression. This is because lossy compression is suitable for removing information from an image without significantly affecting its perceived quality. The quality of the image being clear and understandable is highly critical to medical companies. Image clarity is essential because it allows staff to examine the images for any anomalies that might not have gotten noticed if the image was of low resolution.

2.3 Conceptual difference between DCT and wavelength-based algorithm.

The conceptual difference between Discrete Cosine Transform (DCT) and a wavelength-based algorithm is, DCT is a mathematical function that represents a signal in terms of frequencies. On the other hand, a wavelet-based algorithm analyzes and processes wavelets (mathematical functions) to represent signals as a sum of functions with different frequencies and scales.

2.4 How do the algorithms work?

DCT (Discrete Cosine Transformation):

DCT is a lossy compression algorithm; thus, it separates images into parts of differing frequencies. The least significant frequencies are discarded during a step called quantization. This step makes up a part of the compression process. In the decompression process, the image is retrieved with the essential frequencies remaining. The drawback is that it results in the output containing some distortion as some frequencies have now been left out.

The process of how DCT algorithm works. (ijser, Anitha s)

1. Pixels in the image are divided into 8x8 blocks.
2. Each block is subjected to the DCT from left to right, top to bottom
3. Quantization is used to compress each block.
4. Compressed image blocks are saved in a much smaller quantity of space.
5. Decompression is used to deconstruct the image.

DCT Equations: (Equations taken from (Collage of Redwoods, Ken Cabeen and Peter Gent))

This is how the equations are formed and the compression and decompression process of DCT works.

(1): The image's i,jth entry would be computed by the first equation. Where the matrix p represents an image by its x,yth elements. Since the image is broken into 8x8 blocks of pixels, N in this equation would be 8, because N is the size of the block that the DCT is done on.

$$D(i,j) = \frac{1}{4}C(i)C(j)\sum_{x=0}^{7}\sum_{y=0}^{7}p(x,y)\cos\left[\frac{(2x+1)i\pi}{16}\right]\cos\left[\frac{(2y+1)j\pi}{16}\right] \qquad 3$$

This transforms Equation (1) into equation (3) for a standard 8x8 image.

The DCT Matrix:

A matrix can be obtained from equation (1) using the following equation (4).

$$T_{i,j} = \left\{ \begin{array}{ll} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}}\cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{array} \right\} \qquad 4$$

This equation results in the following matrix for an 8x8 block:

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

(Collage of Redwoods, Ken Cabeen, and Peter Gent)

Performing the DCT:

Medical Images are primarily greyscale images. Thus the pixel values of black and white images range from 0 to 255, where 0 is pure black, and 255 is pure white.

Using a scale factor of 8 (for example, a 1024x1024 image), the corner of the picture on the left-hand side was chosen.

This led to the original matrix looking like this: (Collage of Redwoods, Ken Cabeen, and Peter Gent)

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

Due to DCT's design, the original block gets "leveled" by 128 for each entry, resulting in the following matrix.

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

In order to perform DCT, the matrix is multiplied.

$$D = TMT' \tag{5}$$

With Eq (5), matrix M will be multiplied by matrix T (on the left) first which will transform the rows. Then the T matrix transpose is multiplied to form columns. Which returns a new matrix named D.

Quantization

During the quantization step, the 8x8 blocks of DCT coefficients (i,j) are ready for compression. To calculate quantization, divide each element in D by the equivalent element in Q and then round to the nearest integer value.

$$C_{i,j} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right) \qquad 6$$

The lossy part of the compression takes place near the top left of matrix C. This is because the left top side of the image is the lowest frequency element while the right side is higher frequency, and because the human eye is most sensitive to lower frequencies, high frequencies are discarded. Lossy process would leave matrix C looking something like this.

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Decompression.

The originally used quantization matrix is multiplied by each element of C to reconstruct the image. This will result in matrix R.

$$R_{i,j} = Q_{i,j} \times C_{i,j} \qquad 7$$

For the final step of decompression, the Inverse Discrete Cosine Function is applied to matrix R which is rounded to the nearest integer, after 128 is added to the equation as the elements were "leveled off" in the previous steps. This will give us the decompressed final image.

$$N = round(T' \; R \; T) + 128 \qquad 8$$

Wavelet-Based Compression:

The idea is to transform the signal into the wavelet domain and perform a lossy compression on the wavelet coefficients, which represent the signal's high-frequency and low-frequency components.

Steps for the procedure.

1. Transform the signal into the wavelet domain.

The first step is to perform a wavelet transform on the signal. This transform decomposes the signal into multiple frequency sub-bands, each representing different frequency components of the signal. The wavelet transform also results in wavelet coefficients, which capture the detailed structure of the signal.

2. Quantize the wavelet coefficients.

The next step is to quantize the wavelet coefficients to reduce the amount of data needed to represent the signal. This is done by rounding the coefficients to a certain number of bits and mapping them to a finite set of values.

3. Compress the quantized wavelet coefficients.

The quantized wavelet coefficients are then compressed using standard data compression techniques, such Huffman coding, arithmetic coding, or entropy coding, in this case Huffman coding. These techniques take advantage of the statistical properties of the coefficients to reduce the size of the data even further.

4. Reconstruct the signal.

The last step for the compressed wavelet coefficients are used to reconstruct the original signal by performing an inverse wavelet transform. This results in a signal that is similar to the original signal, but with some loss of detail due to the quantization and compression steps.

Overall, wavelet-based compression works by exploiting the properties of wavelets to decompose signals into frequency sub-bands, which can then be efficiently compressed and reconstructed back into the original signal. This results in high compression ratios, while preserving important details of the signal.

3.0 Investigation of the dataset:

3.1 The type of image will be used?

The images that will be used will be grayscale images because, in a medical environment, almost all the pictures taken are in grayscale image type. These grayscale photos will be saved in tif format and taken from the internet, however during the compression, the images will be transferred back to jpeg as DCT transformation is most compatible with JPEG-types files. The images picked for analysis will have different attributes (such as a very black and white CT scan, an x-ray of an arm with a cast on, and an x-ray of ribcages), since all of these photos have different traits from each other I believe some algorithms could be better at compressing different types of images.

3.2 Methodology

Gathering of data will be conducted through codes on MATLAB. The data that shall be gathered and juxtaposed between the two algorithms are: Peak Signal-to-Noise Ratio (PSNR)Value, the Compressions Ratio of the image, and the compressed Image Size

(Kilobytes). Then a chart on PSNR Comparison for DCT and Wavelet algorithms will be made along with the compressed image size and ratio for DCT and Wavelet algorithms.

Peak Signal-to-Noise Ratio (PSNR):

The PSNR compares the reconstructed image with the original one and calculates the difference between them in terms of the mean squared error. PSNR indicates to the user whether the compressed image is high or low quality. The result is in terms of decibels (dB) and the higher the decibel the higher the quality of the reconstructed image. The following mathematical formula is used to find PSNR:

$$PSNR = 20\log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

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

Compressions Ratio:

The compression ratio is a simple formula to identify the reduction in the size of data after compression. The higher the compression ratio the more efficient the compression algorithm. The formula for compression ratio is:

Original image size / Compressed original image size = Compression Ratio

The compressed image size will be measured in kilobytes as the original picture only covers a small amount of storage. The PSNR Value is the numerical difference between the value of pixels of the uncompressed image and the compressed image. PSNR has a positive correlation hence as the value of PSNR goes up the more identical the compressed picture is to the original, thus the quality of the picture also goes up.

The investigation will be conducted through these steps:

Methodology:

- Collect a sample of medical images and compress them using both wavelet-based and DCT-based algorithms
- Assess the image quality of the compressed images using PSNR
- Evaluate the compression efficiency of the algorithms by comparing the compressed file sizes
- Evaluate the Charts for the images.

The pictures chosen:

Three different pictures will be chosen for this analysis and each picture will have a different attribution therefor the results for each picture will differ (the pictures are cited in the bibliography). The first image: Img_1 is a head CT scan for a patient, and the image consists of a lot of black and white pixels. The second image: Img_2 is an x-ray image of an arm inside of a cast, this picture is a very common photo taken every day in medical centers also it is a picture that consists of some amount of black and white. Finally the third image: Img_3 is an x-ray image of a patient's ribcage, and almost has zero black and white pixels in the image.

4.0 Data Analysis:

4.1 Data collection

Running of the DCT algorithm and how the data was collected ("Discrete Cosine Transform- MATLAB and Simulink")

```
I = imread('Head_CT_scan.tif');
I = im2double(I);
T = dctmtx(8);
dct = @(block_struct) T * block_struct.data * T';
B = blockproc(I,[8 8],dct);
```

Lines 3-5: Compute the two-dimensional DCT of 8-by-8 blocks in the image. The function "dctmtx" returns the N-by-N DCT transform matrix. The reason "dctmtx" is used instead of other DCT functions is because: for small square inputs, such as 8x8 or 16x16, and since the images used will be fitting these square inputs the "dctmtx" function was chosen.

```
6    mask = [1   1   1   1   0   0   0   0
7            1   1   1   0   0   0   0   0
8            1   1   0   0   0   0   0   0
9            1   0   0   0   0   0   0   0
10           0   0   0   0   0   0   0   0
11           0   0   0   0   0   0   0   0
12           0   0   0   0   0   0   0   0
13           0   0   0   0   0   0   0   0];
14   B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);
```

Lines 6-14: From the 64 all but 10 of the DCT coefficients in each block from the upmost left corner are discarded. This is because the lowest frequency element of an image is at the left top side, whilst the right bottom side concise the higher frequency elements, and since the human eye is most sensitive to lower frequencies, high frequencies are discarded this way. This process is the lossy part of the function.

```
15   invdct = @(block_struct) T' * block_struct.data * T;
16   I2 = blockproc(B2,[8 8],invdct);
```

Lines 15,16: This is where the reconstruction of the image is performed using the two-dimensional inverse DCT of each block. This will give us the final decompressed image as I2.

```
17   imshow(I)
18   figure
19   imshow(I2)
```

Running of Wavelet algorithm and how the data was collected ("Haar Wavelet Image Compression")
:

```
1   clear all;
2   image_name='Lung.tif';
3   delta = 0.008;
```

Line 3: Delta is the value that governs the compression ratio and can be changed by the users as they see fit. In this investigation delta will be kept constant at 0.008 because compression is balanced at this value.

```
5   close all;
6   disp(delta)
7
8
9   if (delta>1 || delta<0)
10      error('harr_wt: Delta must be a value between 0 and 1');
11  end
```

Line 5-11: Lines 9-11 makes checks that delta will be between 0 to 1 so there will at least be some compression.

```
14  H1=[0.5 0 0 0 0.5 0 0 0;0.5 0 0 0 -0.5 0 0 0;0 0.5 0 0 0 0.5 0 0 ;0 0.5 0 0 0 -0.5 0 0 ;0 0 0.5 0 0 0 0.5 0;0 0 0.5 0 0 0 -0.5 0;0 0 0 0.5 0 0 0 0.5;0 0 0 0.5 0 0 0 -0.5;]
15  H2=[0.5 0 0.5 0 0 0 0 0;0.5 0 -0.5 0 0 0 0 0;0 0.5 0 0.5 0 0 0 0;0 0.5 0 -0.5 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1;];
16  H3=[0.5 0.5 0 0 0 0 0 0;0.5 -0.5 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 0 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1;];
```

Lines 14-16: These (H1, H2, H3) are the transformation matrices for the wavelet transform.

```
18  H1o = (H1.*(2^0.5));
19  H2o = (H2.*(2^0.5));
20  H3o = (H3.*(2^0.5));
```

Lines 18-20: To get an orthonormal column of each matrix, the columns of H1,H2,H3 are normalized to a length of 1

```
22   Ho=normc(H1o*H2o*H3o);
23   H = H1*H2*H3;
24   x=double(imread(image_name));
25
26   len=length(size(x));
27
28   if len~=2
29       error('Input image must be a grey image');
30   end
```

Line 22-30: Line 22 is the resultant transformation matrix. The function imread reads an image file and returns it as an image array. The double function then converts the data type of the image array to double, which is a higher-precision data type. The result is stored in the variable x. And if the length is not equal to two then it prints that the image must be a grey image. This is because for the matric to work the parameters of the image should be two-dimensional (such as 512x512).

```
32   yo = zeros(size(x));
33   y = zeros(size(x));
34   [r,c]=size(x);
```

```
37   for i=0:8:r-8
38       for j=0:8:c-8
39           p=i+1;
40           q=j+1;
41           yo(p:p+7,q:q+7)=(Ho')*x(p:p+7,q:q+7)*Ho;
42           y(p:p+7,q:q+7)=(H')*x(p:p+7,q:q+7)*H;
43       end
44   end
45
46
47   figure;
48   imshow(x/255);
49
50
51   n1=nnz(y);
```

Line 51: Is the number of non-zero elements that consist within ‘y’.

```
53  zo=yo;
54  m=max(max(yo));
55  yo=yo/m;
56  yo(abs(yo)<delta)=0;
57  yo=yo*m;
58
59
60  z=y;
61  y=y/m;
62  y(abs(y)<delta)=0;
63  y=y*m;
```

Lines 53-63: Values within +delta and -delta in 'y' are replaced by zeros(These are the commands that result in compression).

```
67  n2=nnz(y);
68
69
70  for i=0:8:r-8
71      for j=0:8:c-8
72          p=i+1;
73          q=j+1;
74          zo(p:p+7,q:q+7)=Ho*yo(p:p+7,q:q+7)*Ho';
75          z(p:p+7,q:q+7)=inv(H')*y(p:p+7,q:q+7)*inv(H);
76      end
77  end
```

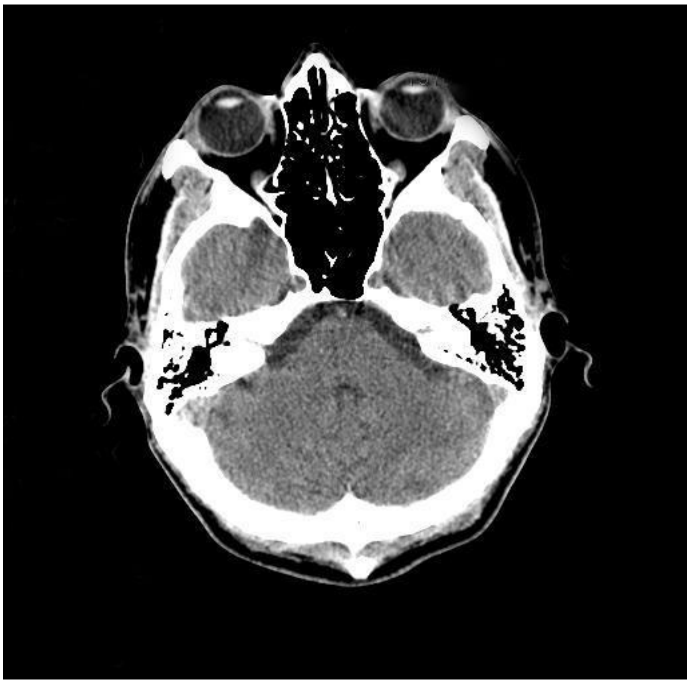
Line 67-77: In line 67, n2 holds the number of non-zero elements in the updated 'y'. While the rest of lines 70-77 are the code for the inverse DWT of the image.

```
79  figure;
80  subplot(121);
81  imshow(x/255);
82  title("original image");
83  subplot(122)
84  imshow(z/255);
85  title("compressed image");
86  imwrite(x/255,'orginal.tif');
87  imwrite(z/255,'compressed.tif')
88
89
90  compression_ratio = n2/n1;
```

Lines 79-90: Line 81 represents the final compressed image, while lines 86 and 87 automate the saving of the original and compressed image in the same folder in which the code resides. The final line of code give the compression ratio as the number of non-zero elements after updated 'y' (n2) / number of non-zero elements in 'y' (n1)

4.2 Tables and Observation

Img_1 'Head_CT_Scan'
Original Image: Image size = 86.9 Kilobytes. Image Dimension = 512x512 pixels.



	DCT Compression	Wavelet Compression
Image		
File Size (Kilobytes)	77.9	49.0
Dimensions (Pixels)	512x512	512x512
Compression	1.12	1.77

Ratio (86.9/Compressed image size in KB)		
PSNR Value (dB)	28.2	27.8

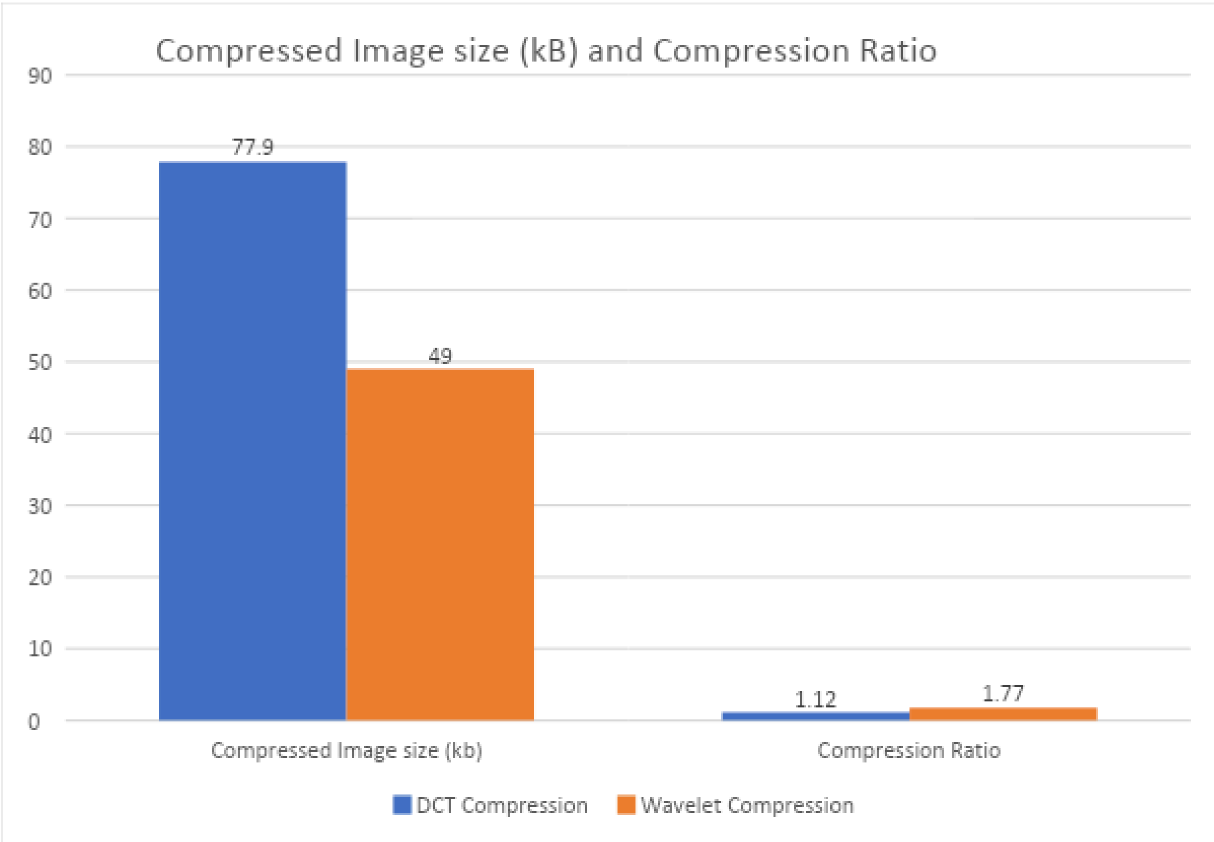


*Figure 4.2.1 “Img_1”

Img_2 ‘Arm_Cast’
Original Image: Image size = 77.9 Kilobytes. Image Dimension = 1024x1024 pixels.

	DCT Compression	Wavelet Compression
Image		
File Size (Kilobytes)	69.4	45.6
Dimensions (Pixels)	1024x1024	1024x1024
Compression Ratio (77.9/Compressed image size in KB)	1.12	1.71
PSNR Value (dB)	26.0	26.1

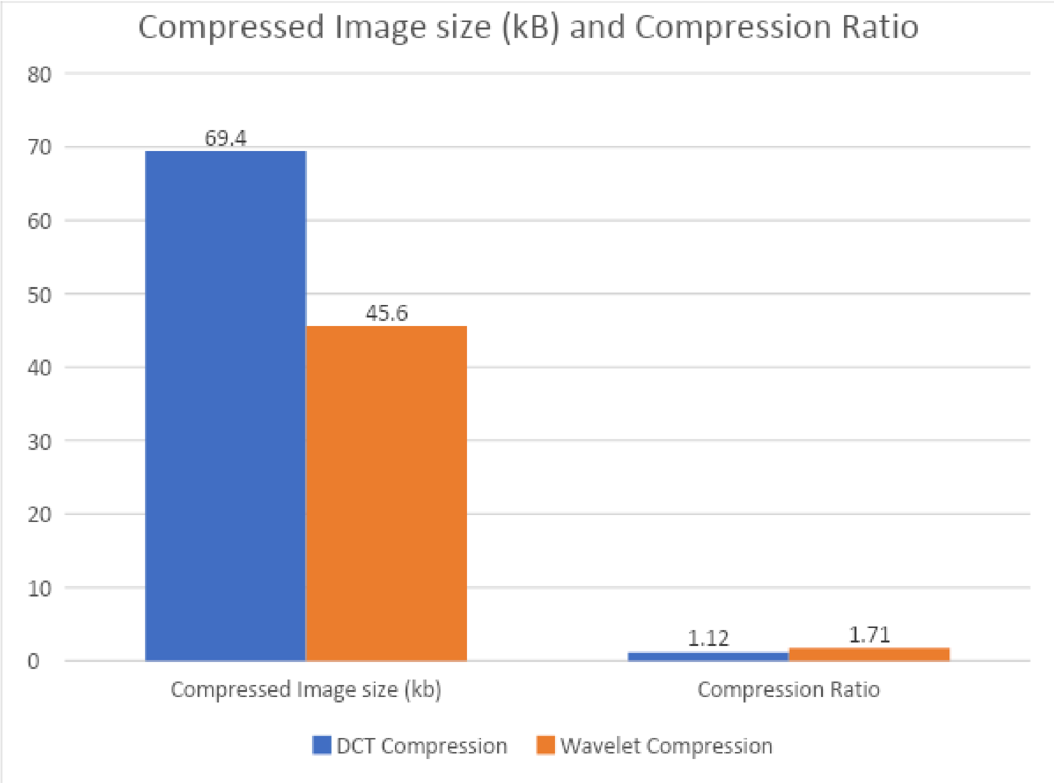
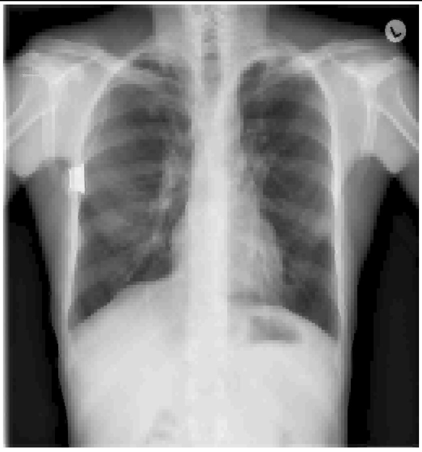


*Figure 4.2.2 “Img_2”

Img_3 ‘Ribcage’
Original Image: Image size = 135 Kilobytes. Image Dimension = 1024x1024 pixels.

	DCT Compression	Wavelet Compression

Image		
File Size (Kilobytes)	67.2	37.8
Dimensions (Pixels)	1024x1024	1024x1024
Compression Ratio (135/Compressed image size in KB)	2.01	3.62
PSNR Value (dB)	25.9	25.4

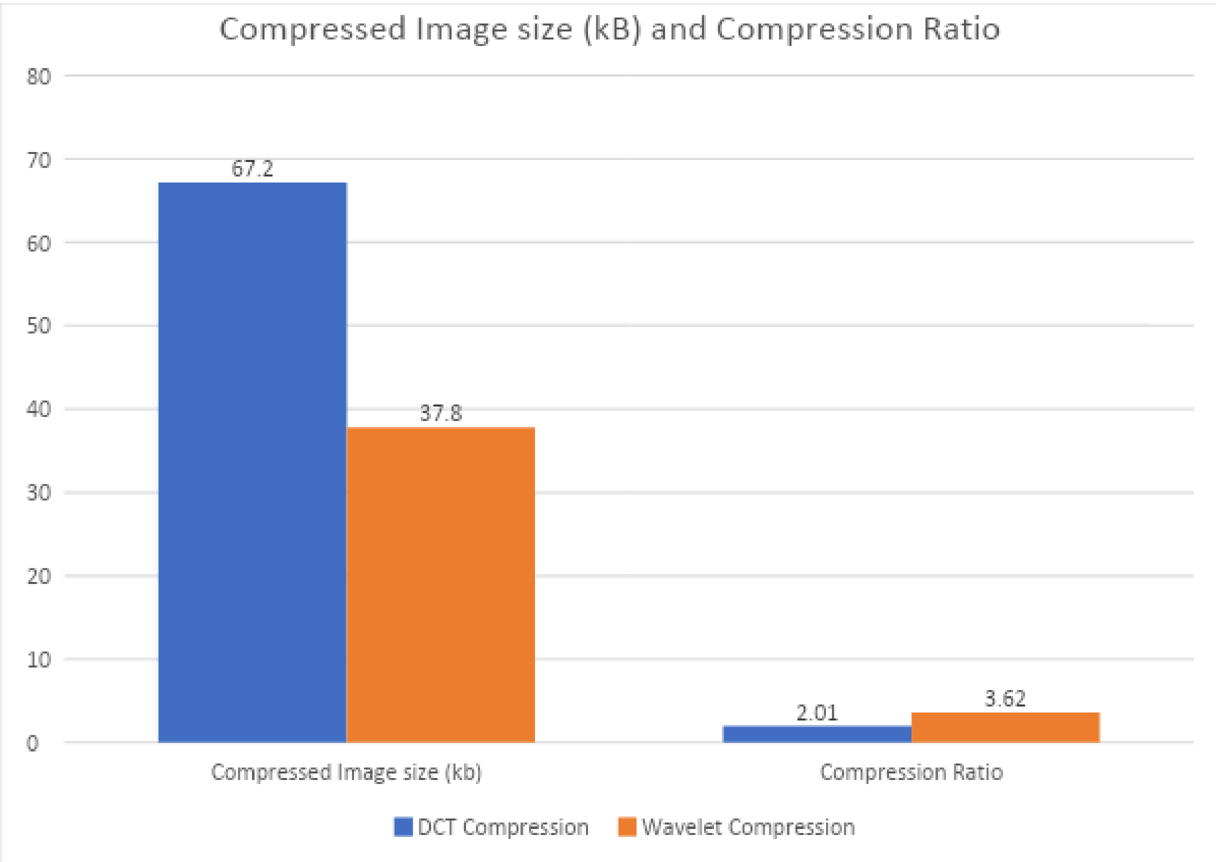


*Figure 4.2.3 “Img_3”

5. Evaluation

5.1 Evaluation of Data gathered.

The data analysis from above had multiple different findings which could end the argument on whether wavelet compression performs better than DCT based algorithm. The three main findings from the analysis of the data above is:

1. Wavelet compression is more efficient as its compression ratio is always higher than the DCT compression ratio in every scenario.

2. Wavelet compression is also more storage-friendly as it is always less compared to the DCT algorithm.

3. DCT compression's picture clarity is almost always higher than wavelets, as understood by the PSNR value.

4. Both algorithms perform better with grey pixel (pixels that value closer to 127) dominated images.

With the logic that "wavelet compression is better at two things whilst DCT is better at only 1", it should be obvious that wavelet compression performs better than DCT-based compression. However, this is false as different scenarios require different algorithms for the compression of different medical images.

An example of this could be seen with the second image 'Arm_Cast' in the 'Data Analysis section. If the 'Arm_Cast' picture was used and the patient had a problem on their arm, such as a cracked bone it could easily have been spotted by both the compressed images as the patient's arms clarity is good in both pictures. In this scenario, it would have been best to use Wavelet compression as it would have saved more storage. However, if the patient were to have been injured in their palm where it is unclear on the wavelet picture, it would have best been to sacrifice some storage in return for clarity with DCT-based compression as the palm of the patient is fully visible with DCT-based compression.

5.2 Advantages, Disadvantages and Limitations of the compression algorithms.

Advantages of using wavelet-based algorithms:

1. Efficient compression: Wavelet-based algorithms compress images without losing diagnostic information or visual quality. Multi-resolution representation allows wavelets to efficiently compress high-frequency (e.g. edges and fine details) and low-frequency (e.g. smooth regions) information. Medical centers need effective compression because it saves storage and money. In a computer with insufficient RAM, loading a high-storage picture may also surpass the available memory, forcing the computer to use the hard drive as virtual memory, slowing the system. With the high compressed picture, the amount of ram used is decreased thus saving memory.

2. Low computational complexity: Wavelet-based algorithms have low computational complexity, meaning they require less processing power compared to other compression techniques, which makes them useful in a resource-limited environment such as in less economically developed countries, or for the personal computers of medical center staff. This would be beneficial for staff as it would be cost-effective for the staff as they are now not forced to buy high-end computers just to compress images

3. Robustness: Data that is missing or corrupted can still be handled using wavelet-based algorithms because they are robust to errors in the wavelet algorithms. This makes them useful as technical limitations or other factors may lead to incomplete or corrupted data in medical imaging applications. Incomplete or corrupted data could be catastrophic for a sector in the medical industry. This could lead to longer queues for patients to wait, wrong diagnoses for patients or at the worst case scenario even patient’s deaths.

Advantages of using DCT compression algorithm:

1. Simplistic Picture Output: During the compression, only the lower frequencies of the image are taken into consideration while creating the matrix for the compression, this creates a picture that is both simplified and easier to understand for the eye. Simplistic picture can help improve communication as the main diagnosis can be easily seen by an untrained eye with the DCT compression done on an image such as an x-ray. It also lowers the risk of miscommunication and medical errors that could be made by the medical staff, which could be fatal for the patient.

2. Fast Decoding: The rapid decoding time of wavelet-based algorithms makes them suitable for real-time applications like medical imaging and telemedicine. This will increase the rapidness of procedures, for example on a busy day in a medical center, patients wouldn’t need to wait for a longer amount of time as the faster compression rate makes the procedure the patients have to go through also more quickly.

3. Widely supported: Many image processing software and hardware support DCT as an image compression algorithm. This means that medical companies can use DCT compressed images across different platforms, without worrying about compatibility issues. Also since DCT compression has been standardized by various international organizations, including the International Organization for Standardization (ISO), making it a widely accepted and trusted compression algorithm.

Disadvantages of Wavelet compression algorithm:

Complexity of implementation: Wavelet-based algorithms can be more difficult to implement and optimize than other compression methods, which can be a barrier to adoption in some medical settings.

Lack of standardization: There are many different wavelet-based compression algorithms and variations, which can make it difficult to standardize compression practices across different medical settings and applications. Since the compression technique is not standardized this can make it difficult to compare compressed images between 2 different medical centers

Disadvantages of DCT compression algorithm:

Lack of spatial information preservation: Wavelet-based compression more effectively preserves spatial information than DCT-based compression, which treats each block of pixels independently. This is a disadvantage as these images contain important details that may affect the diagnostic accuracy of medical professionals.

Poor performance with higher-dimensional data: Compared to wavelet-based compression, DCT-based compression may perform as well with higher-dimensional data, such as 3D medical imaging. This is a disadvantage as some medical images are 3d and 4d such as MRI scans and PET scans, which limits the usage of DCT compression.

Limitations of this experiment:

Limited sample size: The small sample size used in the experiment may not be practical. This is because the quantity and complexity of the medical images used in the investigation could limit the generalizability of the findings and make drawing significant inferences difficult. It would have been easier to draw conclusions about which compression algorithm is better if a larger number of images were analysed.

5.3 Conclusion

Conclusion:

Both techniques have their own strengths and weaknesses and can be used effectively in medical image compression

DCT compression is widely used in image and video compression and is supported by many image processing software and hardware, making it a highly compatible compression algorithm. It is known for its fast compression speed, and it can provide good image quality at high compression ratios wit good PSNR values. DCT compression can be an excellent choice for applications that require high speed and high compression ratios, such as telemedicine and remote medical imaging.

On the other hand, Wavelet compression delivers more flexibility to the user by providing higher efficient compression ratios whilst preserving most of the image details. It offers a compression algorithm to computers with low processing power, making it well-suited for all medical centers despite if they lack in financial resources. Wavelet-based compression is also useful for extreme situations such as if there are any missing data or corrupted data, wavelet compression can still be implemented hence minimizing errors.

Overall, both DCT and wavelet-based compression techniques have their own advantages and disadvantages, and the choice between them depends on the specific needs of the application. For medical image compression, the choice should be based on the requirements of the particular application, such as the image quality, compression ratio, speed, and available hardware.

6. Appendix

6.1 Original Image

Because the images are distorted when I insert them into word I have created a google docs so it is easier to get a better look at the images as well:

https://drive.google.com/drive/folders/17vR93-k6mdhoomP2zwIsIvhjvxfcGVOH?usp=share_link.

6.2 DCT algorithm compressed images.

6.3 Wavelet algorithm compressed images.

6.4 DCT algorithm code.

```
I = imread('Ribcage.tif');
I = im2double(I);
T = dctmtx(8);
dct = @(block_struct) T * block_struct.data * T';
B = blockproc(I,[8 8],dct);
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   0   0   0   0   0   0
        1   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);
invdct = @(block_struct) T' * block_struct.data * T;
I2 = blockproc(B2,[8 8],invdct);
imshow(I)
figure
imshow(I2)
```

6.4 Wavelet algorithm code.

```
clear all;
image_name='Arm.tif';
delta = 0.005;

close all;
disp(delta)


if (delta>1 || delta<0)
    error('harr_wt: Delta must be a value between 0 and 1');
end


H1=[0.5 0 0 0 0.5 0 0 0;0.5 0 0 0 -0.5 0 0 0;0 0.5 0 0 0 0.5 0 0 ;0 0.5 0 0 0 -0.5 0 0 ;0 0 0.5 0 0 0 0.5 0;0 0 0.5 0 0 0 -0.5 0;0 0 0 0.5 0 0 0 0.5;0 0 0 0.5 0 0 0 -0.5;];
H2=[0.5 0 0.5 0 0 0 0 0;0.5 0 -0.5 0 0 0 0 0;0 0.5 0 0.5 0 0 0 0;0 0.5 0 -0.5 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1;];
H3=[0.5 0.5 0 0 0 0 0 0;0.5 -0.5 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 0 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1;];

H1o = (H1.*(2^0.5));
H2o = (H2.*(2^0.5));
H3o = (H3.*(2^0.5));

Ho=normc(H1o*H2o*H3o);
H = H1*H2*H3;
x=double(imread(image_name));

len=length(size(x));

if len~=2
    error('Input image must be a grey image');
end

yo = zeros(size(x));
y = zeros(size(x));
[r,c]=size(x);


for i=0:8:r-8
    for j=0:8:c-8
        p=i+1;
        q=j+1;
```

```
        yo(p:p+7,q:q+7)=(Ho')*x(p:p+7,q:q+7)*Ho;
        y(p:p+7,q:q+7)=(H')*x(p:p+7,q:q+7)*H;
    end
end


figure;
imshow(x/255);


n1=nnz(y);

zo=yo;
m=max(max(yo));
yo=yo/m;
yo(abs(yo)<delta)=0;
yo=yo*m;


z=y;
y=y/m;
y(abs(y)<delta)=0;
y=y*m;



n2=nnz(y);


for i=0:8:r-8
    for j=0:8:c-8
        p=i+1;
        q=j+1;
        zo(p:p+7,q:q+7)=Ho*yo(p:p+7,q:q+7)*Ho';
        z(p:p+7,q:q+7)=inv(H')*y(p:p+7,q:q+7)*inv(H);
    end
end
```

```
figure;
subplot(121);
imshow(x/255);
title("original image");
subplot(122)
imshow(z/255);
title("compressed image");
imwrite(x/255,'orginal.tif');
imwrite(z/255,'compressed.tif');


compression_ratio = n2/n1;
```

w7. Bibliography

- Chang, Chein-I, et al. "A comparative study of DCT and wavelet-based image compression in radiography." Journal of digital imaging 15.4 (2002): 245-252.
- “Discrete Cosine Transform- MATLAB and Simulink.” *Discrete Cosine Transform- MATLAB & Simulink*, www.mathworks.com/help/images/discrete-cosine-transform.html.
- “Haar Wavelet Image Compression.” *Haar Wavelet Image Compression - File Exchange - MATLAB Central*, 17 Feb. 2023, www.mathworks.com/matlabcentral/fileexchange/57392-haar-wavelet-image-compression.

- “How to Calculate PSNR of Compressed Images, and How to Compare PSNR...” *How to Calculate PSNR of Compressed Images, and How to Compare PSNR of Images Compressed by Two Different Techniques. - MATLAB Answers - MATLAB Central*, 4 Oct. 2012, www.mathworks.com/matlabcentral/answers/49906-how-to-calculate-psnr-of-compressed-images-and-how-to-compare-psnr-of-images-compressed-by-two-diff.
- Ken Cabeen and Peter Gent. “Image Compression and the Discrete Cosine Transform” N
- Li, Lihua, et al. "Compression of medical images using wavelet transforms." Proceedings of the 1999 International Conference on Image Processing. IEEE, 1999.
- “Lossy Data Compression: JPEG.” *Lossy Data Compression: JPEG*, cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/dct.htm.
- Nawaz, Saqib Ali, et al. “Advance Hybrid Medical Watermarking Algorithm Using Speeded up Robust Features and Discrete Cosine Transform.” *PubMed Central (PMC)*, 4 June 2020, https://doi.org/10.1371/journal.pone.0232902.
- “Peak Signal-to-noise Ratio (PSNR) - MATLAB Psnr.” *Peak Signal-to-noise Ratio (PSNR) - MATLAB Psnr*, www.mathworks.com/help/images/ref/psnr.html.
- Prod-Images-Static.Radiopaedia.Org, 2023, https://prod-images-static.radiopaedia.org/images/40967943/6139c4f7e8399384b60912f35fa123_jumbo.jpeg. Accessed 17 Feb 2023.
- “‘Routine Head Scan’ on Me.” Flickr, 12 Aug. 2005, www.flickr.com/photos/ciscel/124548696.

- Tariq, Syed Muhammad, and Mudassar Raza. "Comparison of DCT and wavelet based compression techniques for CT scan images." arXiv preprint arXiv:1301.0496 (2013).
- Tohumoglu, Gulay. “Wavelets and Their Usage on the Medical Image Compression with a New Algorithm.” NSIP, Jan. 1999.
- Wu, Wei, et al. "DCT Versus Wavelet: A comparison of image compression methods." Proceedings of the International Conference on Natural Computation. IEEE, 2005.