ECE 513: Computer Assignment 2

IMAGE TRANSFORMS FOR DATA COMPRESSION

Name – Danish Gufran

CSU ID: 833161673

Due Date: April 1, 2021

1 Introduction

Image compression is used to reduce the image size and redundancy of the image data. The amount of data used to represent these images, therefore needs to be reduced. Image compression deals with redundancy, the number of bits needed to represent on image by removing redundant data. Decreasing the redundancy is the main aim of the image compression algorithms. Image compression technique, mostly used two-dimensional (2D) image compression. The goal of this Computer Assignment is to implement, study and compare two different image transforms, 2-D Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), for the purpose of image data compression. The peppers Image was used as a Mat file for compression and decompression of the images. Then the Signal to Noise ratio (SNR) of Original image to Reconstructed image is calculated this which will give us an insight into the compression ratio. Wavelet analyzer toolbox is used to check the reconstruction quality using DWT.

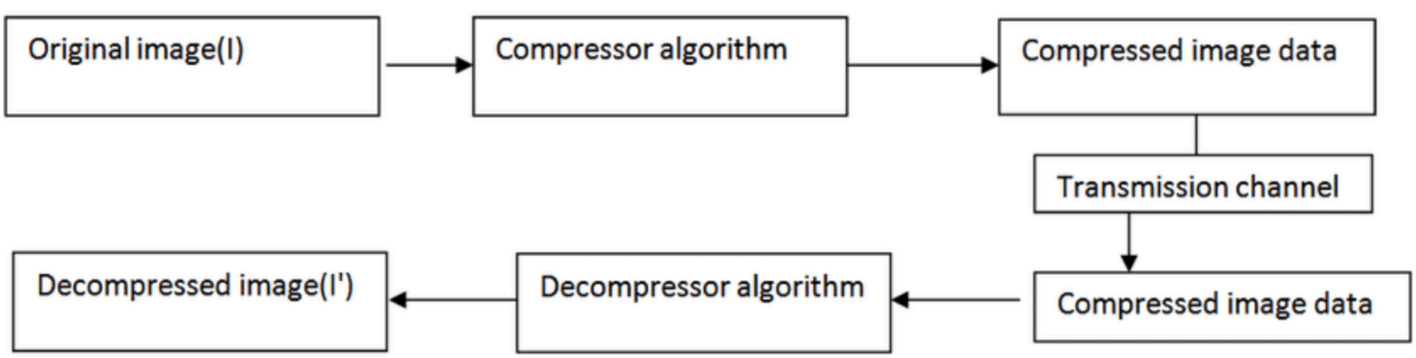

Fig.1: The block diagram of an Image Compression system

The block diagram shows a general representation of an image compression system. The figure explains that, first the original image is first loaded into the system and then a compressor algorithm is applied to it. This compressor is an image encoder to remove redundant data and retain important or crucial image data, this compresses the image therefore reducing its size. The method to retrieve the compressed data is to apply the decompressor algorithm to it, this is simply just an image decoder where the original image data is retracted from the encoder information, thereby presenting the original image.

2 Theory

We discuss 2 theories the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT). The said transforms are applied on MATLAB to the peppers image to understand their workings on data compression.

2.1 Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) is used to separate the image into pixels. DCT is used in signal, image processing especially for data compression because it has a strong energy compaction. DCT image compression may compress the image in NxN metric formation. The DCT transforms the image into the pixels. The pixel of image is transformed into the level of compression process. Then the image is transformed in to quantization process.

The DCT uses a cosine transform matrix as shown.

$$ct(m,n) = \begin{cases} \frac{1}{\sqrt{M}} & , m = 0, 0 \le n \le M-1 \\ \sqrt{\frac{2}{M}} \cos \frac{\pi(2n+1)m}{2M} & , 1 \le m \le M-1, 0 \le n \le M-1 \end{cases} \tag{1}$$

The DCT of a 2 dimensional image for x(m,n) of size N X N is as shown.

$$X(k,l) = \begin{cases} \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) & , k,l = 0,0 \quad m,n \in [0, N-1] \\ \frac{2}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) \cos \frac{(2m+1)k\pi}{2N} \cos \frac{(2n+1)k\pi}{2N} & , k,l \in [0, N-1] \quad m,n \in [0, N-1] \end{cases} \tag{2}$$

The Inverse Discrete Cosine Transform (IDCT) is as shown.

$$x(m,n) = \frac{1}{N}X(0,0) + \frac{2}{N}\sum \sum_{k=0,l=0}^{N-1} X(k,l)\cos\frac{(2m+1)k\pi}{2N}\cos\frac{(2n+1)l\pi}{2N} \qquad (3)$$

The DCT can be applied for multi-dimensional objects, in this assignment we will be focusing on the 2-Dimensional DCT for this data compression.

Application of DCT on the peppers image

The DCT is applied on the peppers image. The image is provided as a .mat file. We use two data blocks for data compression, that is 8x8 and 16x16. The results of each are compared at the end of this project. First the image is classified into non-overlapping blocks and the DCT is applied to it. The coefficients generated are stored into a matrix, now through following the energy-based scheme the dominant or critical coefficients are stores and the rest of the accumulated data is discarded. Mostly, the DCT coefficients closer to zero are discarded as these values do not seriously affect the image in any way during reconstruction these are negligible values.

Reconstruction of the image using DCT coefficients

The matrix generated from using the DCT is then used to calculate the IDCT (inverse) both the blocks, that is 8X8 and 16X16. The reconstructed image will have some errors as some of the coefficients were neglected during DCT.

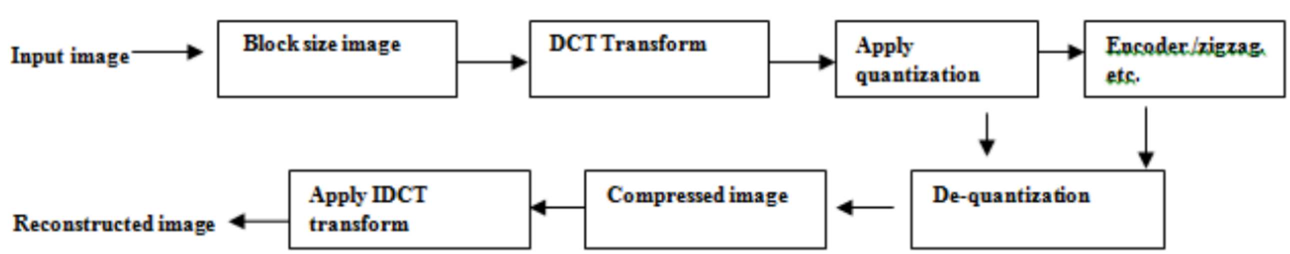

Fig.2: Block diagram of DCT and IDCT

The block diagram explains the entire DCT/IDCT process for image compression and reconstruction. The IDCT dequantizes the data from the matrix having the DCT coefficients and on applying the IDCT the reconstructed image is obtained. To verify the results in comparison from the original and the reconstructed image we use the signal to noise ratio as a method for comparison. The SNR value indicates the noise content in the reconstructed image this information is used to determine the image quality that has been reconstructed. A higher SNR value indicates a good reconstruction and a successful IDCT operation.

2.2 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is used to separate the image into pixels. The DWT has a standing feature that is employed in signal or image processing for lossless image compression. DWT is used in lossy and lossless image compression technique using a discrete signal. The wavelet series expansion of function f(x) ∈ L2 relative to wavelet (x) and scaling function ψ(x) is given by.

$$f(x) = \sum_m c_{n_o}(m)\varphi_{n0,m}(x) + \sum_{n=n_0}^{\infty}\sum_m d_n(m)\psi_{n,m}(x)$$

where jo is an arbitrary starting scale and the cn_o (m) is called approximation or scaling coefficients and d_n(m) is the wavelet coefficients. The above function maps a continuous variable into sequence of coefficients. If the expanded function is discrete, the resulting coefficients are called Discrete Wavelet Transform (DWT). In DWT, the coefficients of an image are calculated by decomposing the image. These coefficients are called sub-bands. The threshold of different filters are then compared with these coefficients.

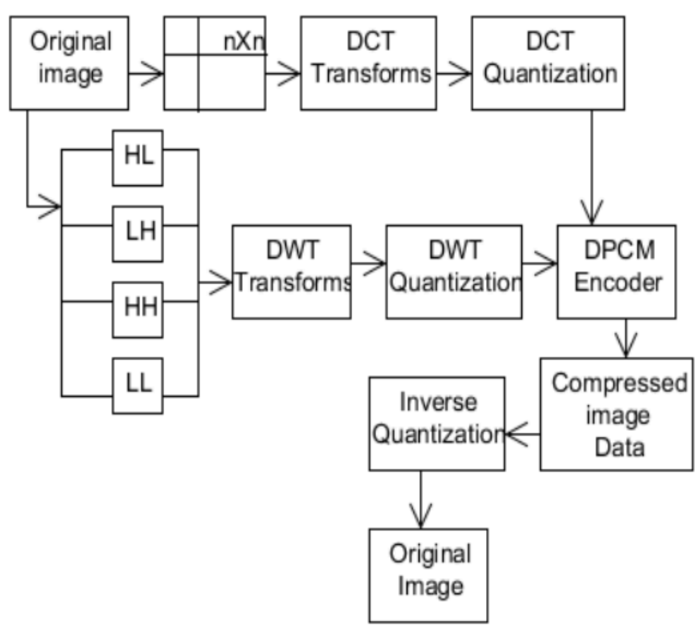

Fig.3: DWT Compression block diagram

The block demonstrates the applications of both DCT and DWT. In this portion we will focus on the DWT aspect for image compression. The DWT is used for lossy and lossless compression using a discrete signal. The L represents a low pass filter, where all the high frequencies are attenuated and the low frequencies are allowed to pass, this is utilized for a perfect reconstruction. The H represents high pass filter, where all the low frequencies are attenuated and the high frequencies are allowed to pass. The next step is to move on to the quantization process where this is repeated to obtain the best results. The DWT compression also retrieves a good SNR value, this is the primary reason for employing DWT as the compression technique is not only effective but also has a good reconstruction.

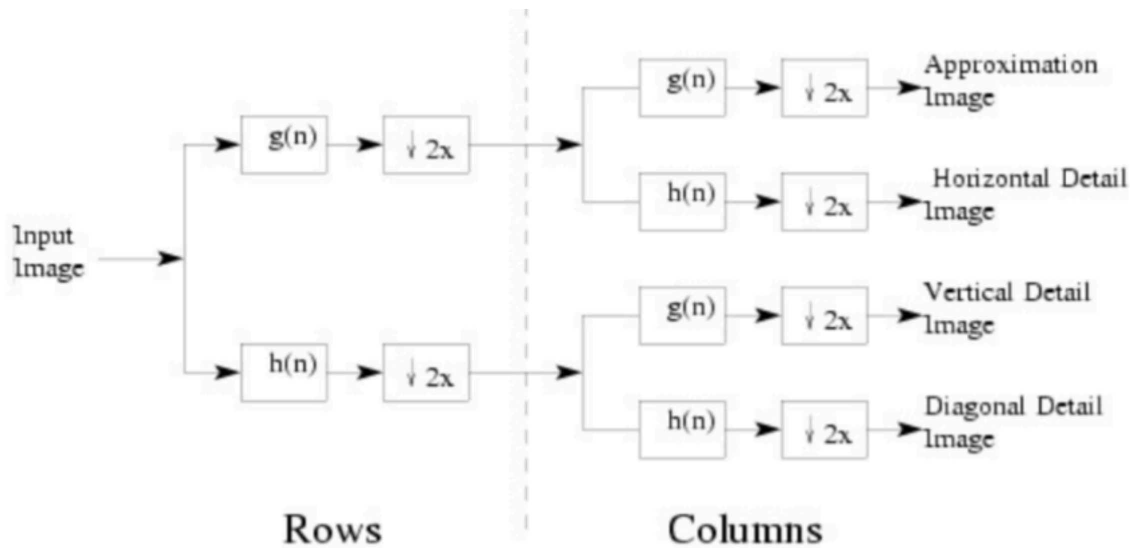

Fig.4: 2D DWT process

For 2D DWT we use separable wavelets obtained from 1-D wavelets. Let us consider a level 1 2D DWT of an NXN image. We can implement a 1-D DWT along the rows, leading to two “sub-images” of size N x N, followed by 1-D DWT along the columns of these two 2 images, resulting in four sub-images of size N x N.

The first sub-image that is obtained by low-pass filtering and subsampling along rows and columns gives the “ow-pass approximation. The second sub-image is obtained by low-pass filtering-subsampling along rows and high-pass filtering-subsampling along columns giving the first added details corresponding to the vertical edge details, the third and fourth sub-images similarly give the horizontal and diagonal edge details. Reconstruction from these sub-images can be done similar to the 1-D case. The process can be iterated on the low-pass approximation several times as in the 1-D case to obtain finer frequency resolution and perform multi-level 2-D DWT. This has different levels of compression according to the level of compression the ratio of compression also changes in this Assignment we are considering 2 level compression and 3 level compression using Daubechies (db), Symlet (sym) wavelets.

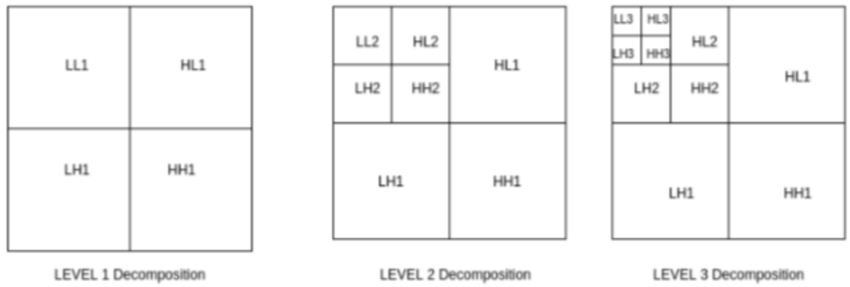

Fig.5: Levels of DWT decomposition

The above diagram shows 3 output formats for different level of decomposition for 2D DWT. In this Assignment we use MATLAB Wavelet analyzer toolbox 2D wave analyzer to view these outputs. We focus on the level 2 and level 3 decomposition used in the DWT Daubechies (db) and Symlet (sym) wavelets.

Signal to Noise Ratio (SNR)

The SNR values for the reconstructed images are calculated to determine the image quality of the reconstructed image. The formula for calculating the SNR is shown below.

$$SNR = 10log_{10}\frac{\sigma_o^2}{\sigma_e^2}$$

The σ_o^2 is the variance of the original image and σ_e^2 is the variance of the error image.

Algorithm

DCT Algorithm

1. Start
2. Load the peppers.mat file
3. Extract image and display
4. Apply 2D DCT for the blocks (8X8 or 16X16)
5. Calculate SNR
6. Create DCT without non overlapping block
7. Mask the data
8. Create DCT after applying energy threshold on blocks (8X8 or 16X16)
9. Calculate SNR
10. Perform IDCT
11. Calculate SNR
12. End

DWT Algorithm

1. Start
2. Load image into the wavelet analyzer
3. Perform level 2 Daubechies wavelet
4. Perform level 3 Daubechies wavelet
5. Plot horizontal and vertical histogram
6. Display compression and reconstruction data
7. Perform level 2 Symlet wavelet
8. Perform level 3 Symlet wavelet
9. Plot horizontal and vertical histogram
10. Display compression and reconstruction data
11. End

3 Results

3.1 Discrete Cosine Transform

The following is the original image extracted from the Peppers.mat file.

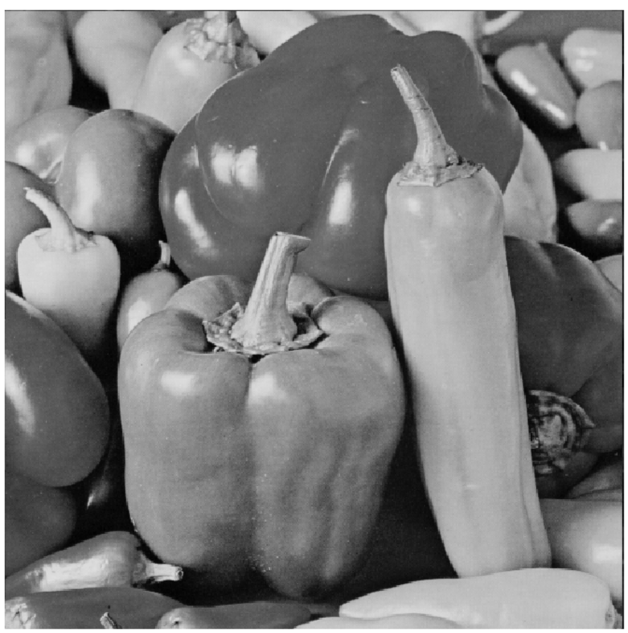

Fig.6: Original Image

The DCT is applied on the image without portioning the image with non-overlapping blocks.

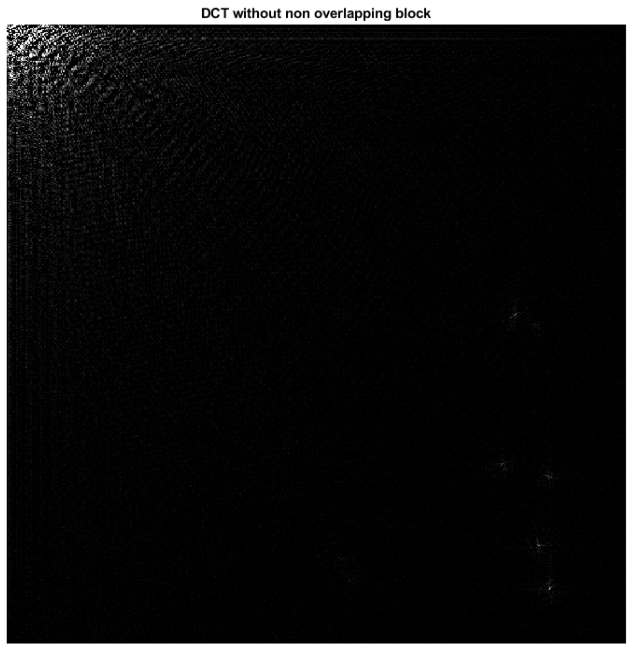

Fig.7: DCT without non-overlapping blocks

The Figure 7 shows that the DCT coefficients are not properly blocked. This leads to energy loss thus affecting the compression of the image and data loss. To over-come this the images should be partitioned into non-overlapping blocks of block size 8X8, 16X16 and above.

Performing DCT with Block Size 8X8

The below figures shows DCT of Pepper after performing partitioning of non- overlapping blocks with block size of 8x8. Where the first figure shows the DCT of the image without applying any threshold for the DCT coefficients where coefficient with energy level less than 90 % can also be seen and the second figure shows the DCT of the image with threshold for the DCT coefficients greater than 90 % of the energy saved and the rest removed.

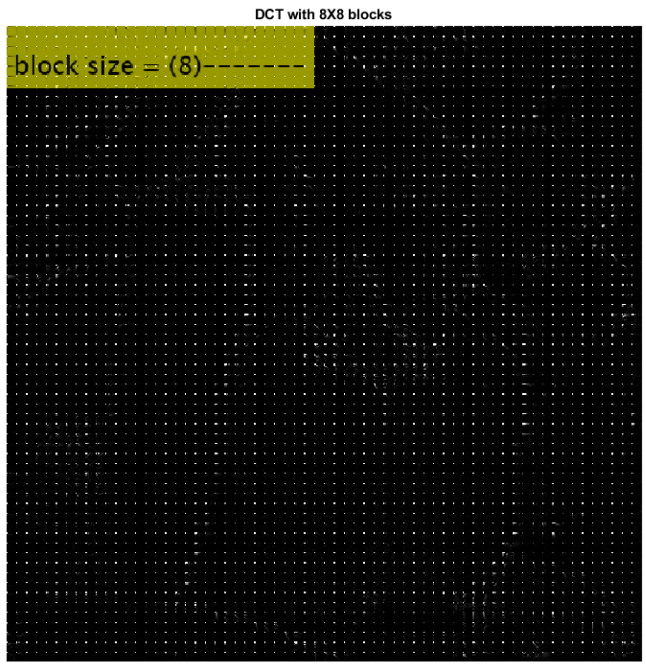

Fig.8 DCT without threshold

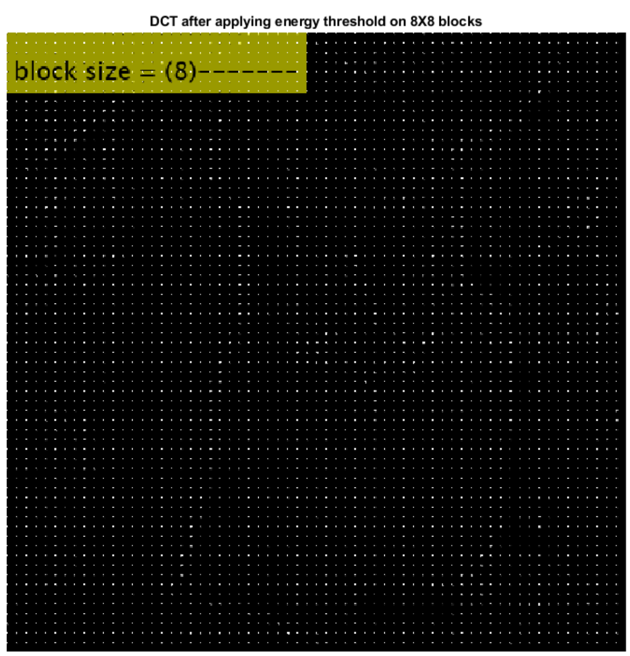

Fig.9 DCT with threshold

The next step is to reconstruct the image using IDCT to view and compare the results. We observe an SNR value of 12.229063. We considered the DCT coefficient with threshold image for reconstruction.

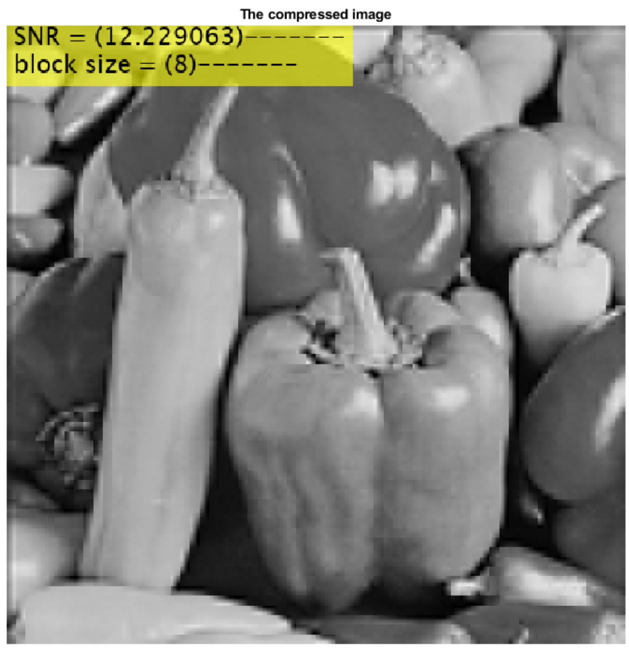

Fig.10: The compressed reconstructed image

Histogram

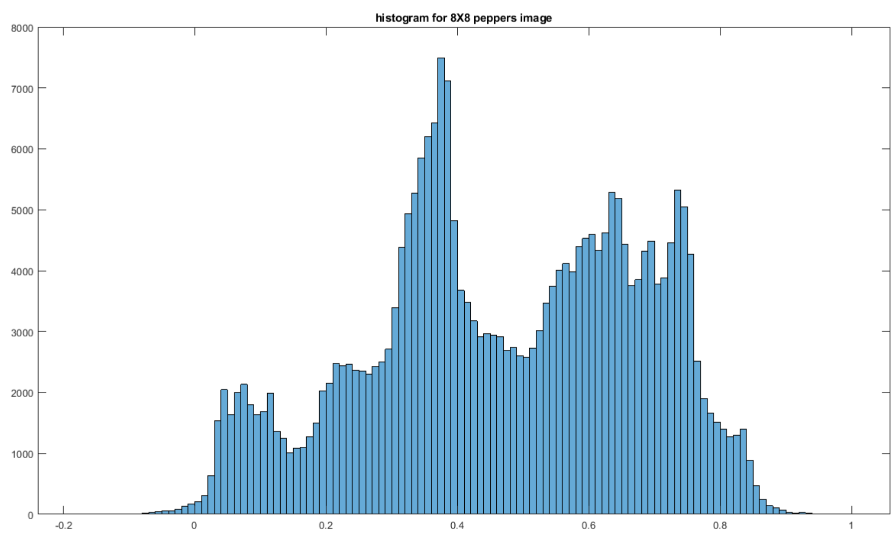

Fig.11: Histogram for 8x8

Performing DCT with Block Size 16X16

Performing the same operation using the block size as 16x16.

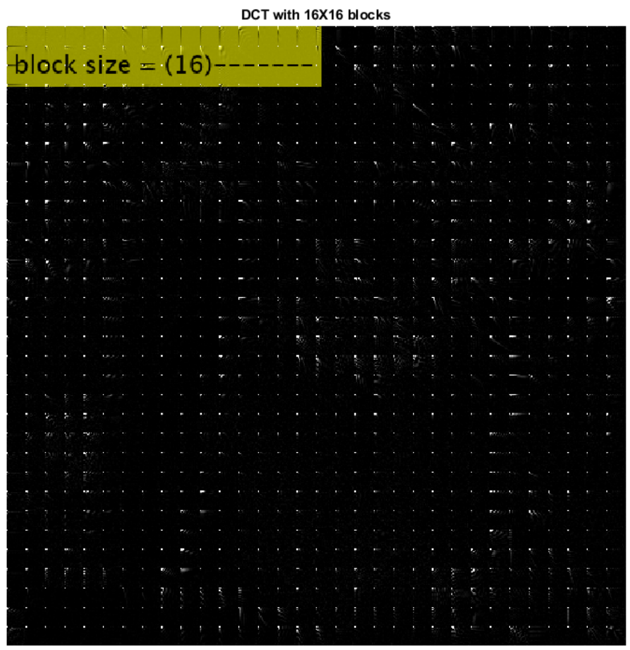

Fig.12 DCT without threshold (16x16)

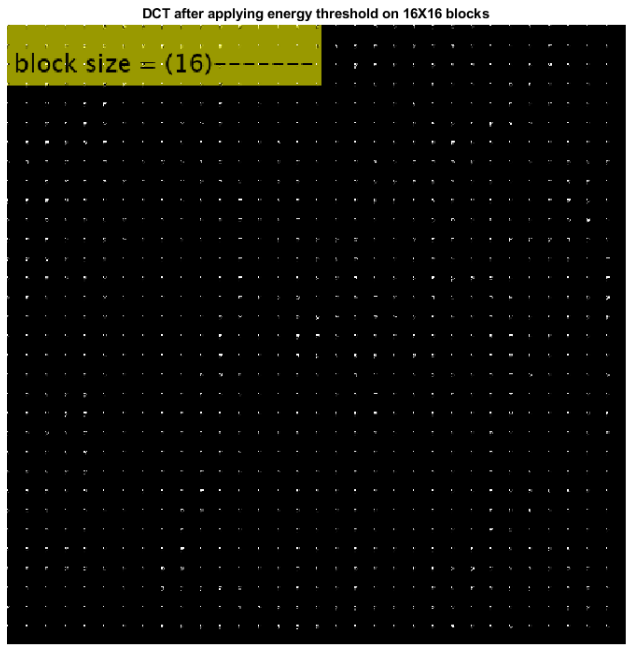

Fig.13 DCT with threshold (16x16)

The above figures shows DCT of Pepper after performing partitioning of non- overlapping blocks with block size of 16x16. Where the first figure shows the DCT of the image without applying any threshold for the DCT coefficients where coefficient with energy level less than 90 % can also be seen and the second figure shows the DCT of the image with threshold for the DCT coefficients greater than 90 % of the energy saved and the rest removed.

The next step is to reconstruct the image using IDCT to view and compare the results. We observe an SNR value of 11.354206. We considered the DCT coefficient with threshold image for reconstruction.

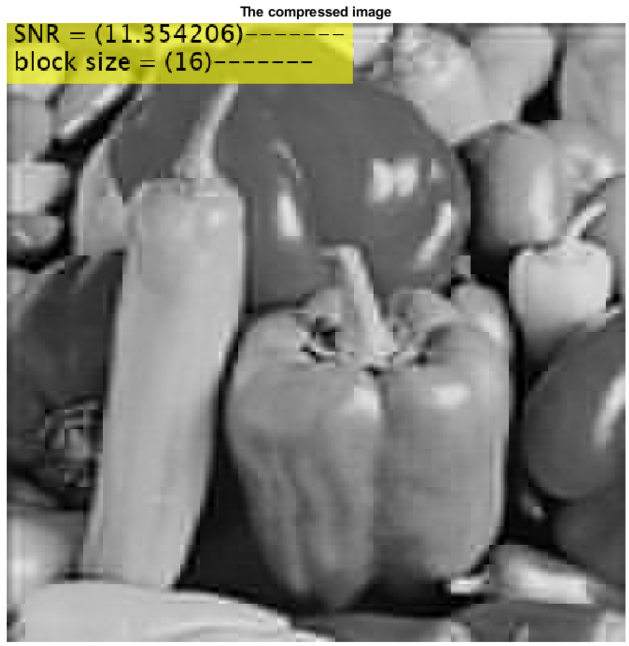

Fig.14: The compressed reconstructed image

Histogram

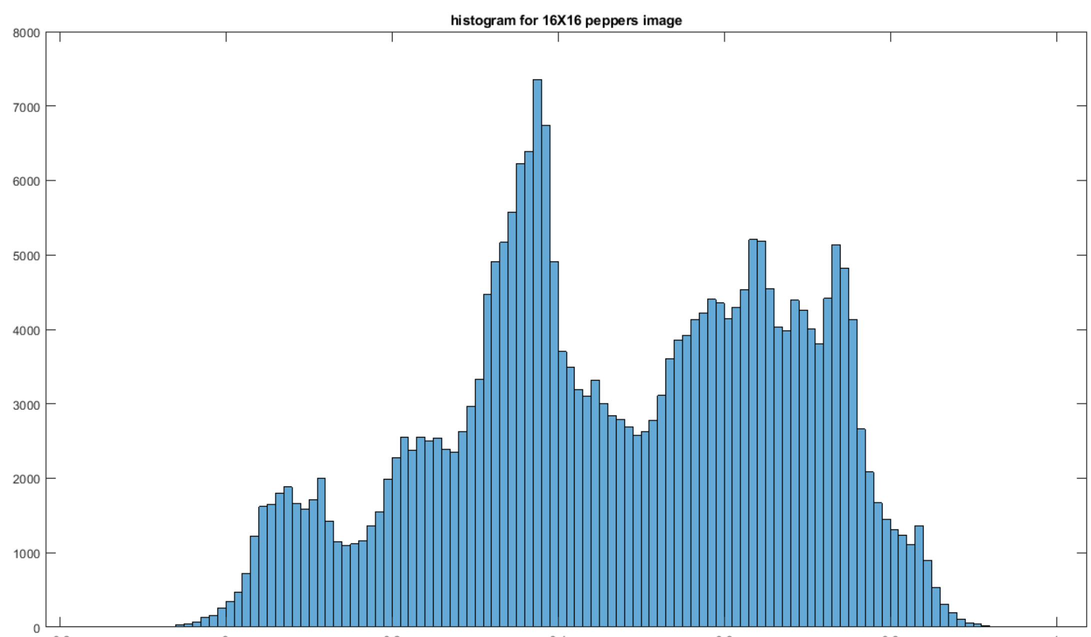

Fig.15: Histogram for 16x16

Comparing the two blocks

The original file was compressed using 8x8 and 16x16 partitions of non-overlapping block of the image. The original image size was 349Kb. The following is the comparison of the image after compression with the SNR and image size. The dimension the image were stored and kept constant for the purpose of comparison.

The tabular column

Block Size	**8x8**	**16x16**
SNR Value	12.229063	11.354206
Image Size	**134Kb**	**194Kb**

The information from this tabular column clearly indicates that the block size of 8x8 has a better compression as compared to the 16x16 block size. The 8x8 block size has a better SNR value and a lower image size which the ideal process for data compression. We were successfully able to reconstruct a better-quality image with lower image size. The original image size was 349Kb and the 8x8 block size image was 134Kb which is approximately 260% lower.

3.2 Discrete Wavelet Transform

DWT was performed on the Pepper Image that was provided using Daubechies wavelet and Symlet wavelet. 2 level and 3 level decomposition where performed. First we analyze the histogram of the original image. The histogram has been generated from the wavelet toolbox from MATLAB. A histogram is a bar graph-like representation of data that buckets a range of outcomes into columns along the x-axis. The y-axis represents the number count or percentage of occurrences in the data for each column and can be used to visualize data distributions.

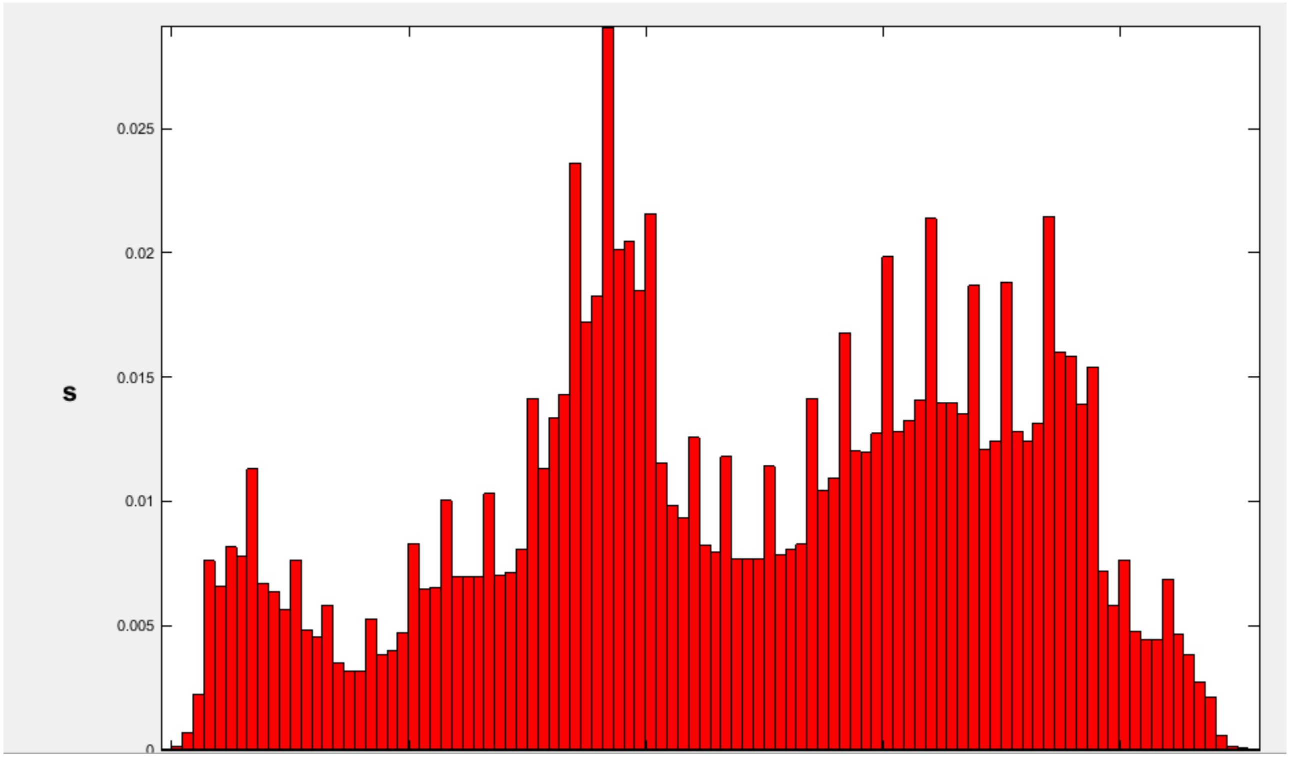

Fig.16: Histogram of the Original Image

Daubechies wavelet – Level 2 Decomposition

The wavelet analyzer toolbox from MATLAB is used to represent all the DWT wavelet decompositions. The following figures show the horizontal and vertical details of the peppers image.

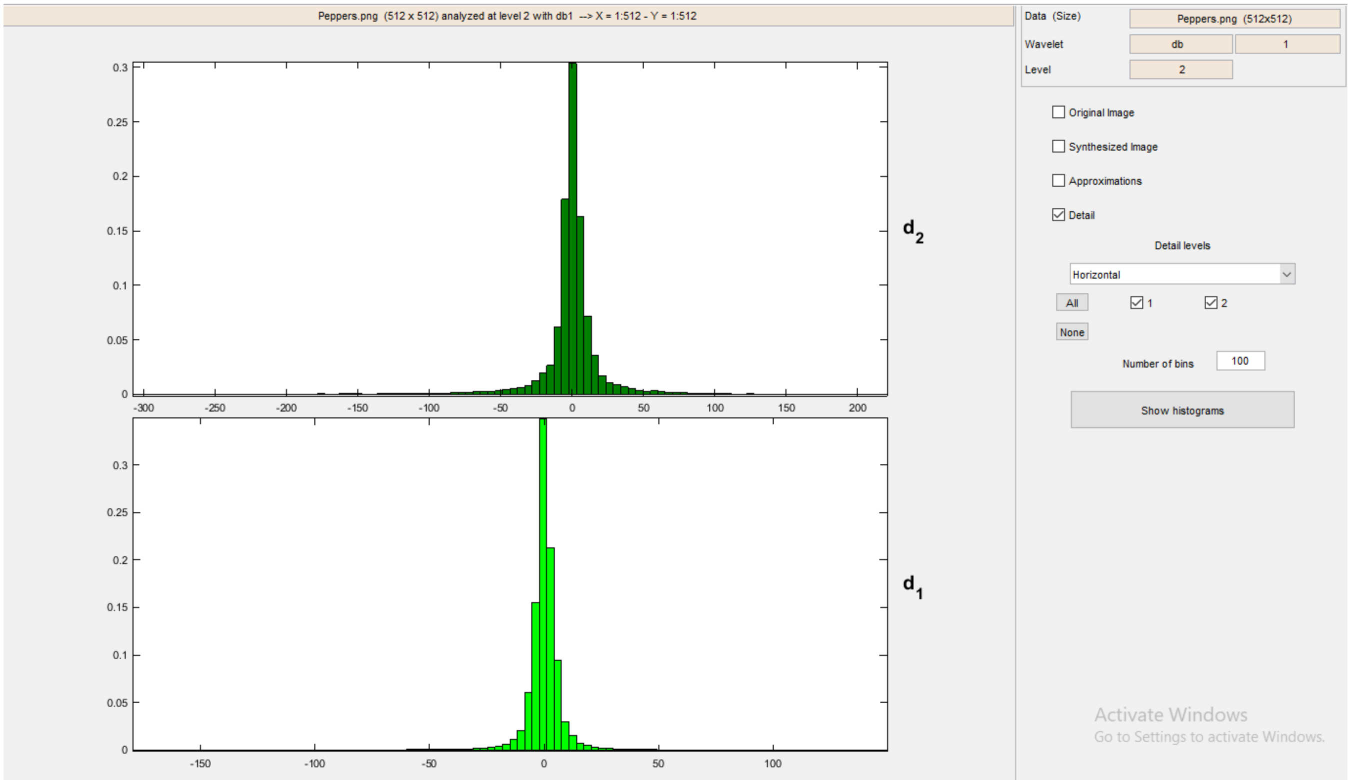

Fig.17: Db2 Horizontal details

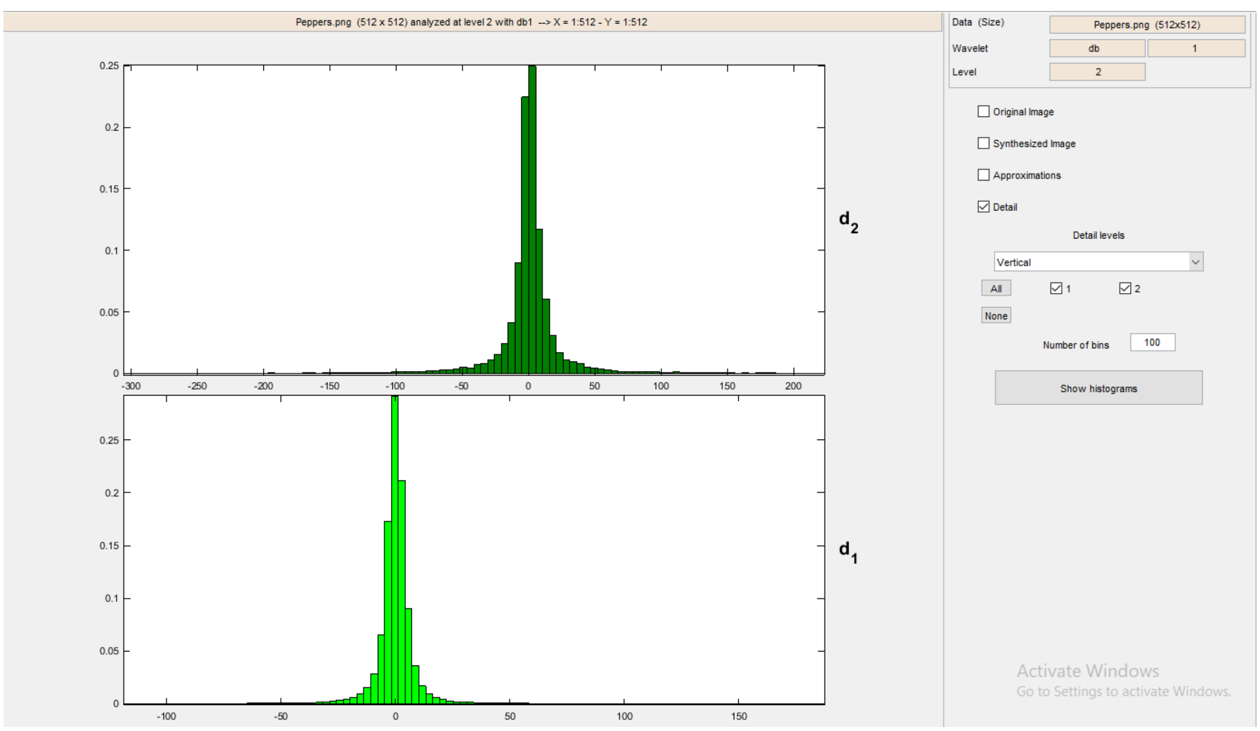

Fig.18: Db2 Vertical details

The histograms above provide information on the frequency-distribution of the synthesized image and both detail level of all the levels of image in vertical, diagonal, and horizontal domain. The histogram represents the wavelet detail coefficients of the original image. The two rows show the histograms of both level of decomposition.

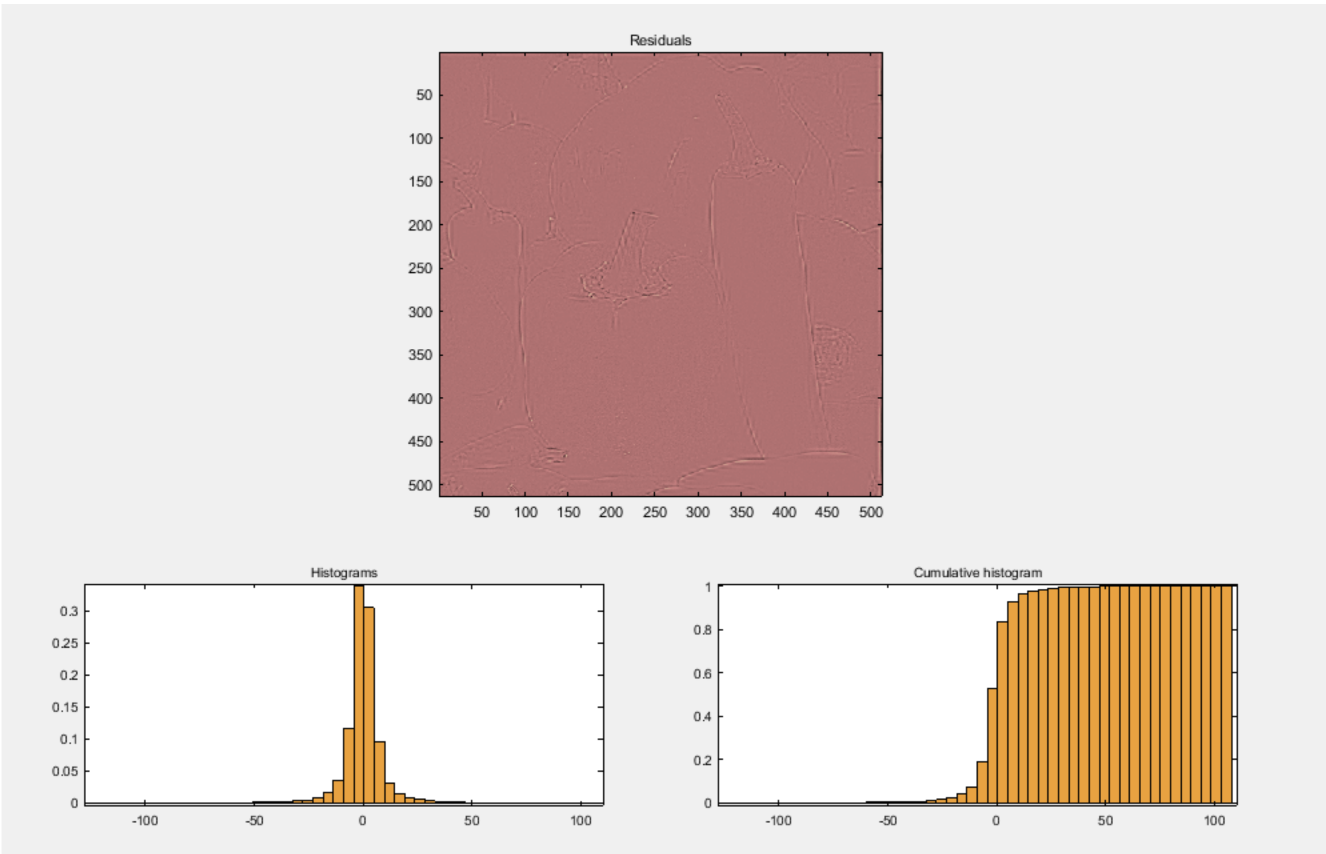

Fig.19: Decomposition Histogram

The next figure explains the retained energy and number of zeros in the compressed image to better understand the compression of each decomposition. The figure shows 99.24% of energy has been retained and 93.50% of the number of zeros have been removed. This constitutes a good compression.

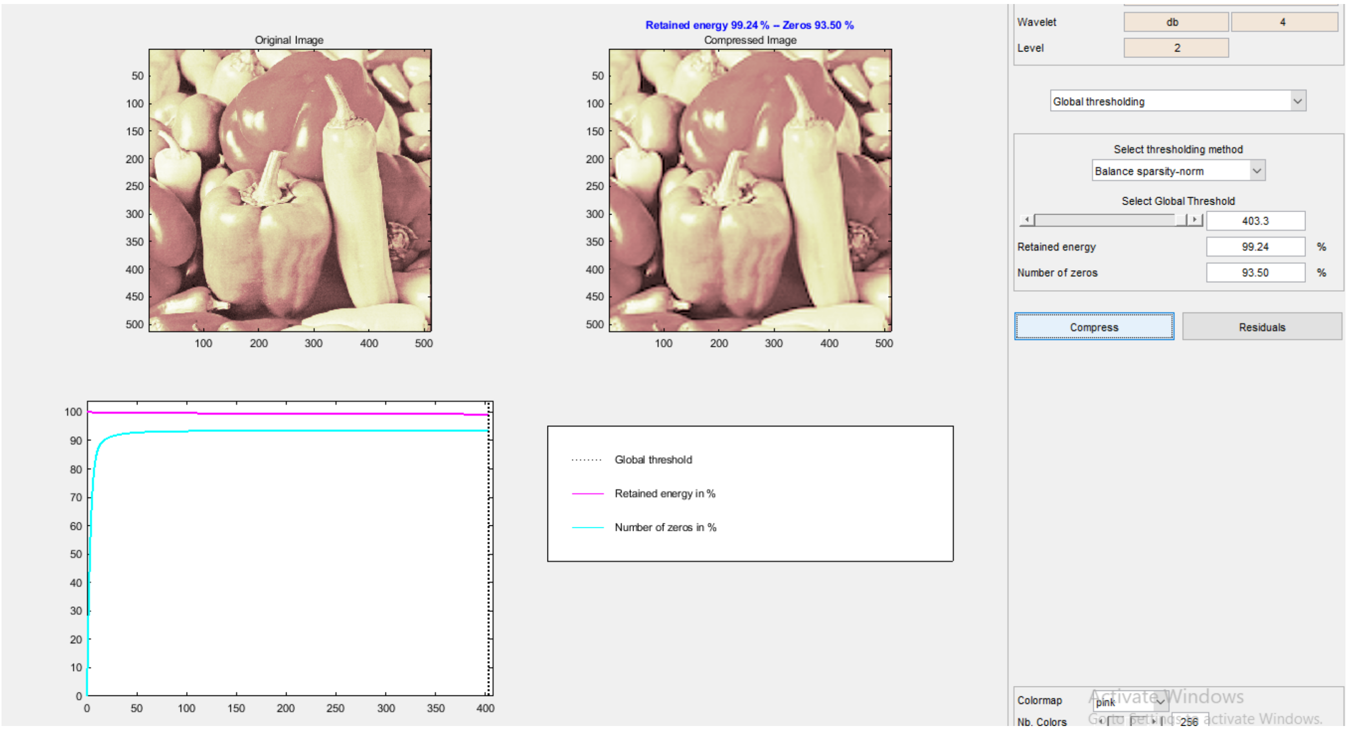

Fig.20: Db2 Compression statistics

Fig.21: Db2 Reconstruction DWT/IDWT Operation

The above figure shows successful reconstruction of the peppers image using the Daubechies Wavelet level 2 decomposition, The demonstration of DWT and IDWT has also been pictured.

Daubechies wavelet – Level 3 Decomposition

The following are the image obtained from 3 level decomposition using the toolbox.

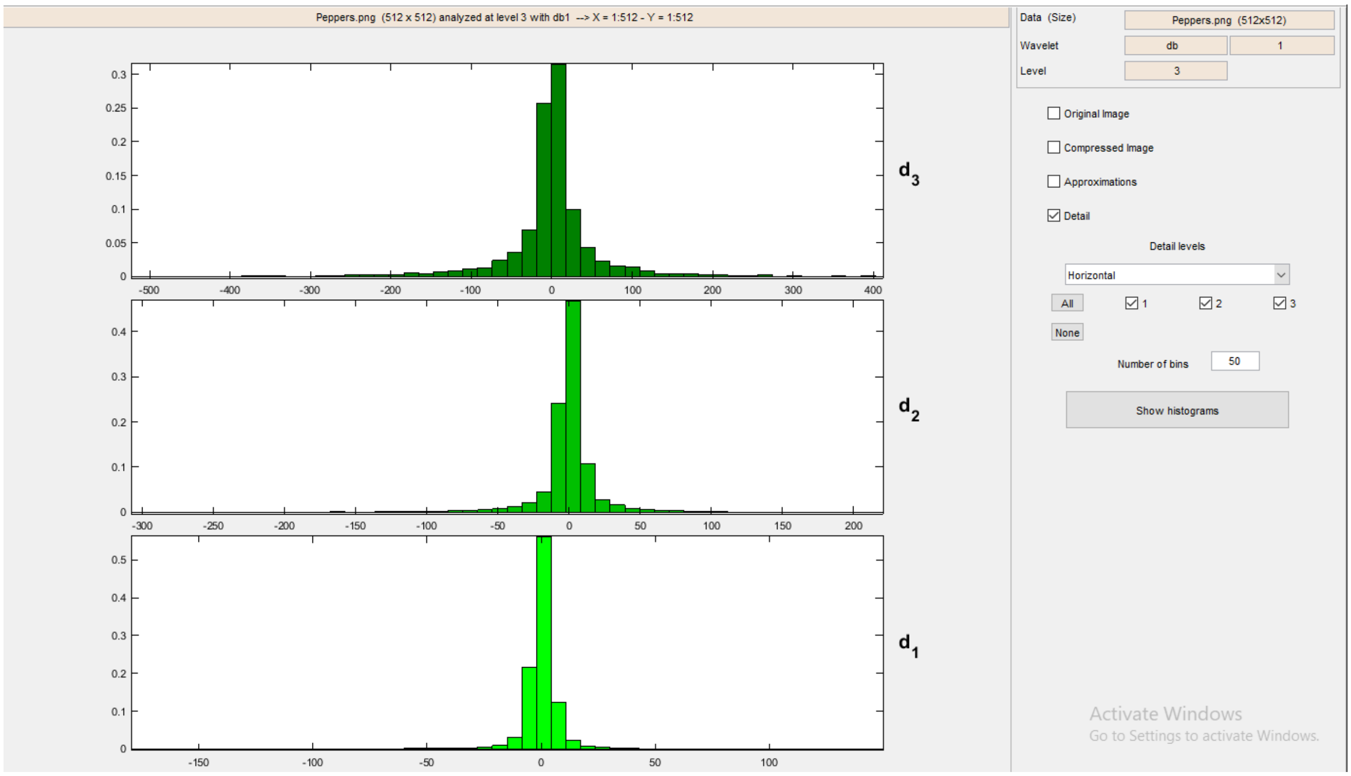

Fig.22: Db3 Horizontal Details

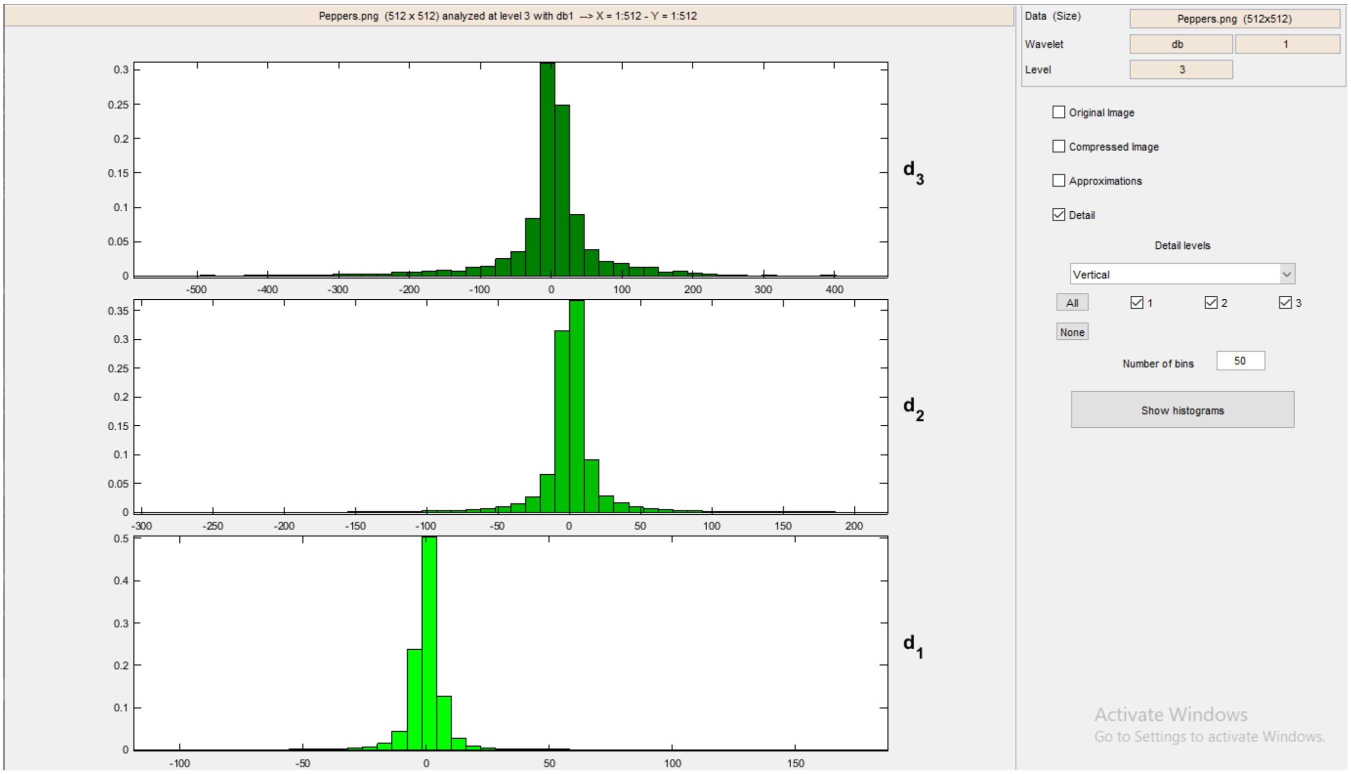

Fig.23: Db3 Vertical Details

The histograms above provide information on the frequency-distribution of the synthesized image and both detail level of all the levels of image in vertical, diagonal, and horizontal domain. The histogram represents the wavelet detail coefficients of the original image. The three rows show the histograms of three levels of decomposition.

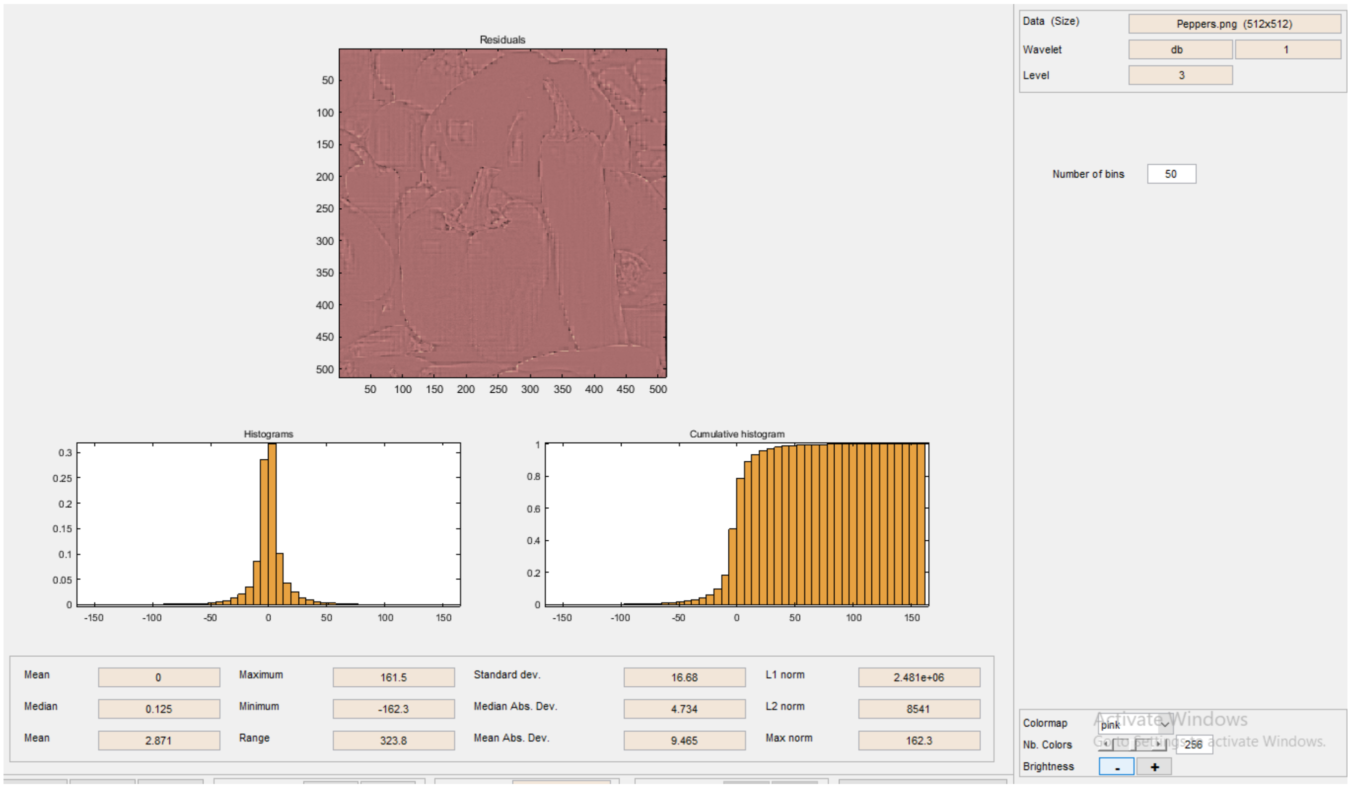

Fig.24: Decomposition Histogram

The next figure explains the retained energy and number of zeros in the compressed image to better understand the compression of each decomposition. The figure shows 98.39% of energy has been retained and 98.39% of the number of zeros have been removed. This constitutes a good compression.

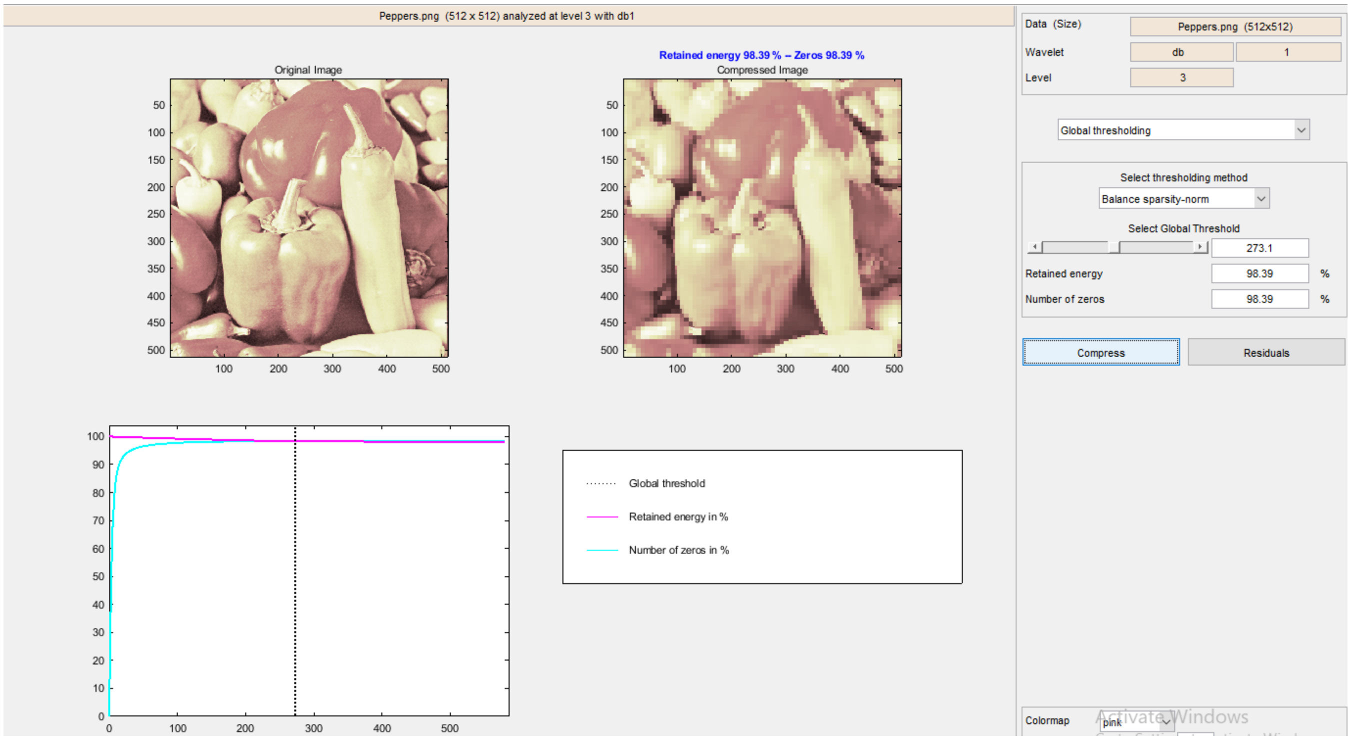

Fig.25: Db3 Compression Statistics

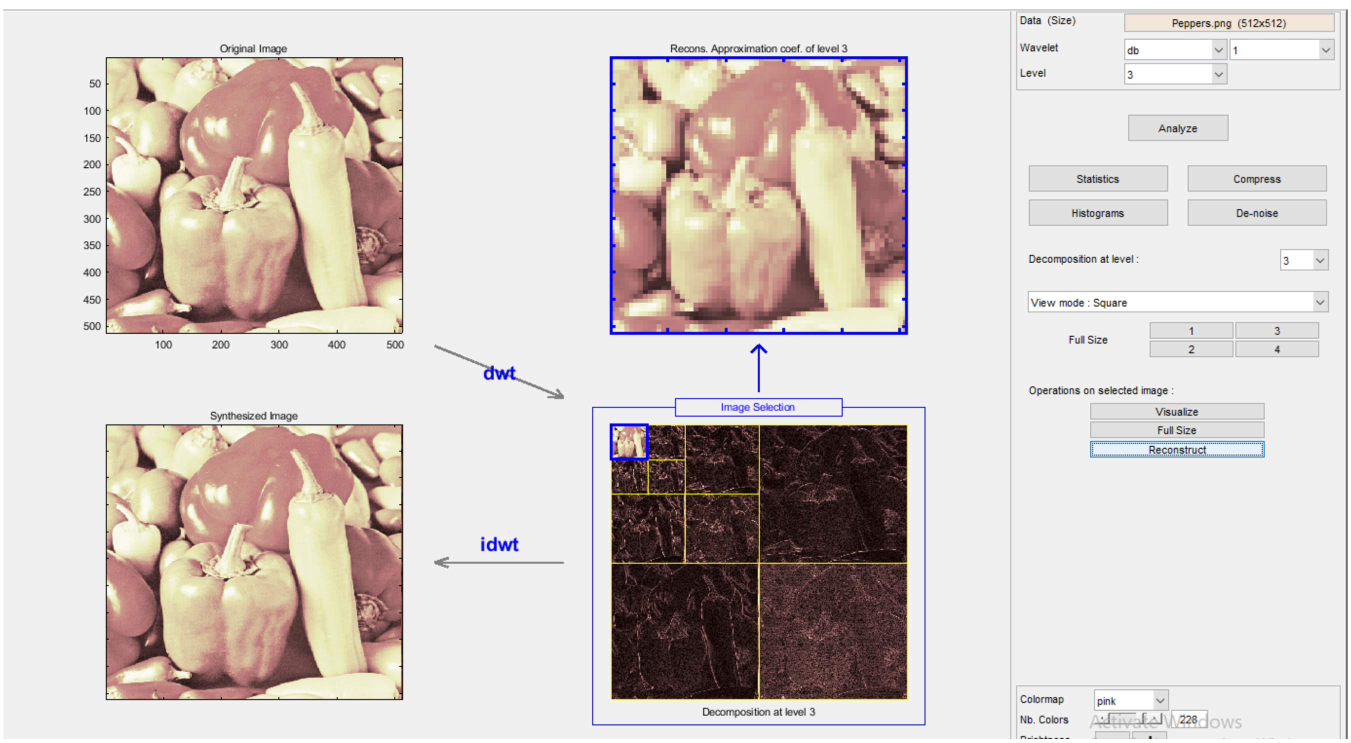

Fig.26: Db3 Reconstruction DWT/IDWT Operation

The above figure shows successful reconstruction of the peppers image using the Daubechies Wavelet level 3 decomposition, The demonstration of DWT and IDWT has also been pictured.

Symlet wavelet – Level 2 Decomposition

The following are the image obtained from 2 level decomposition using the toolbox.

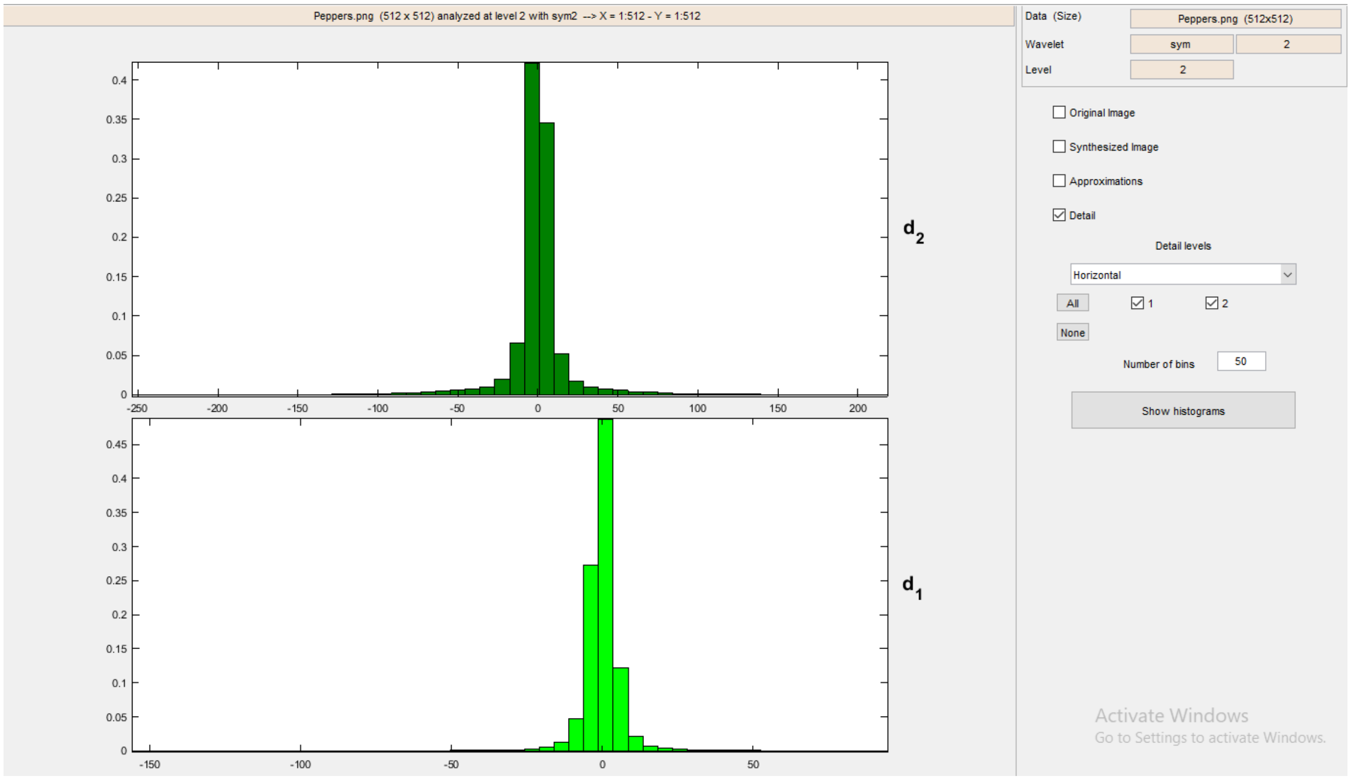

Fig.27: Sym2 Horizontal Details

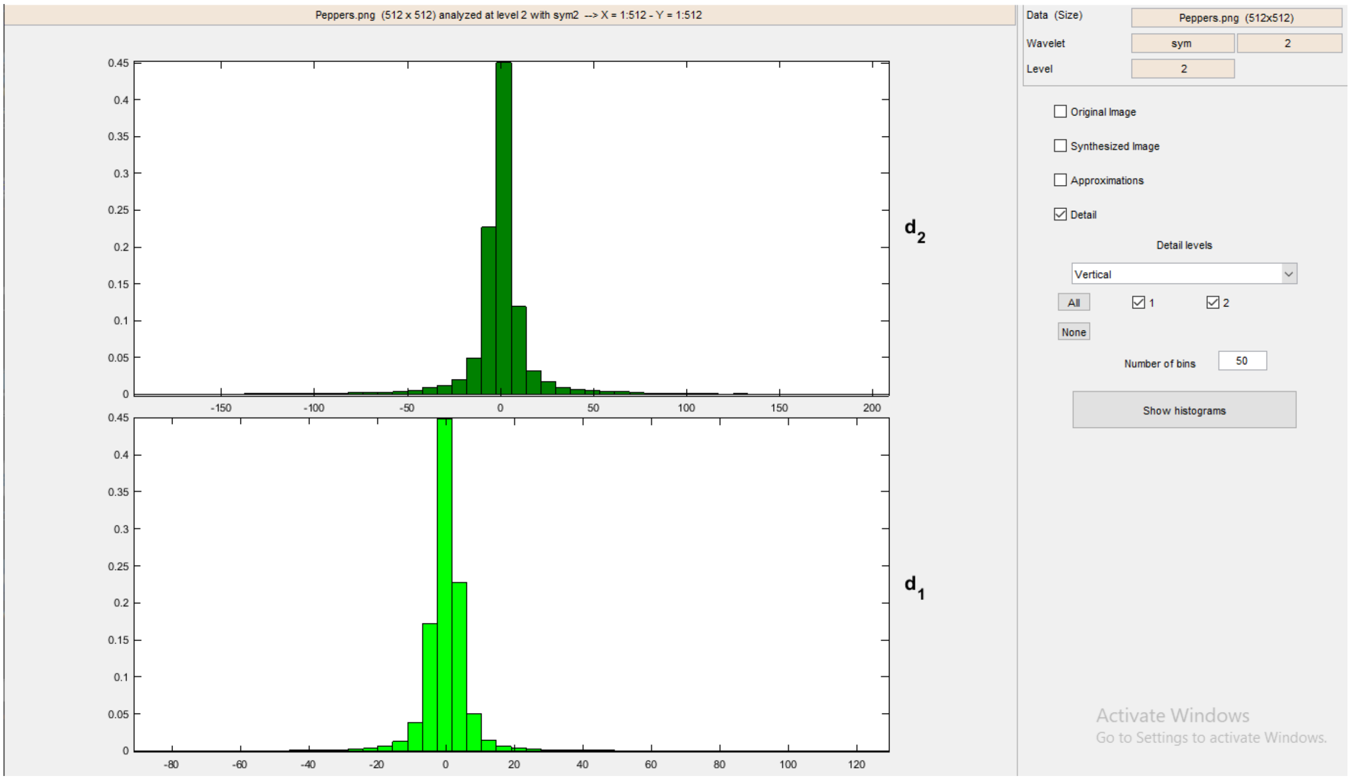

Fig.28: Sym2 Vertical Details

The histograms above provide information on the frequency-distribution of the synthesized image and both detail level of all the levels of image in vertical, diagonal, and horizontal domain. The histogram represents the wavelet detail coefficients of the original image. The two rows show the histograms of both level of decomposition.

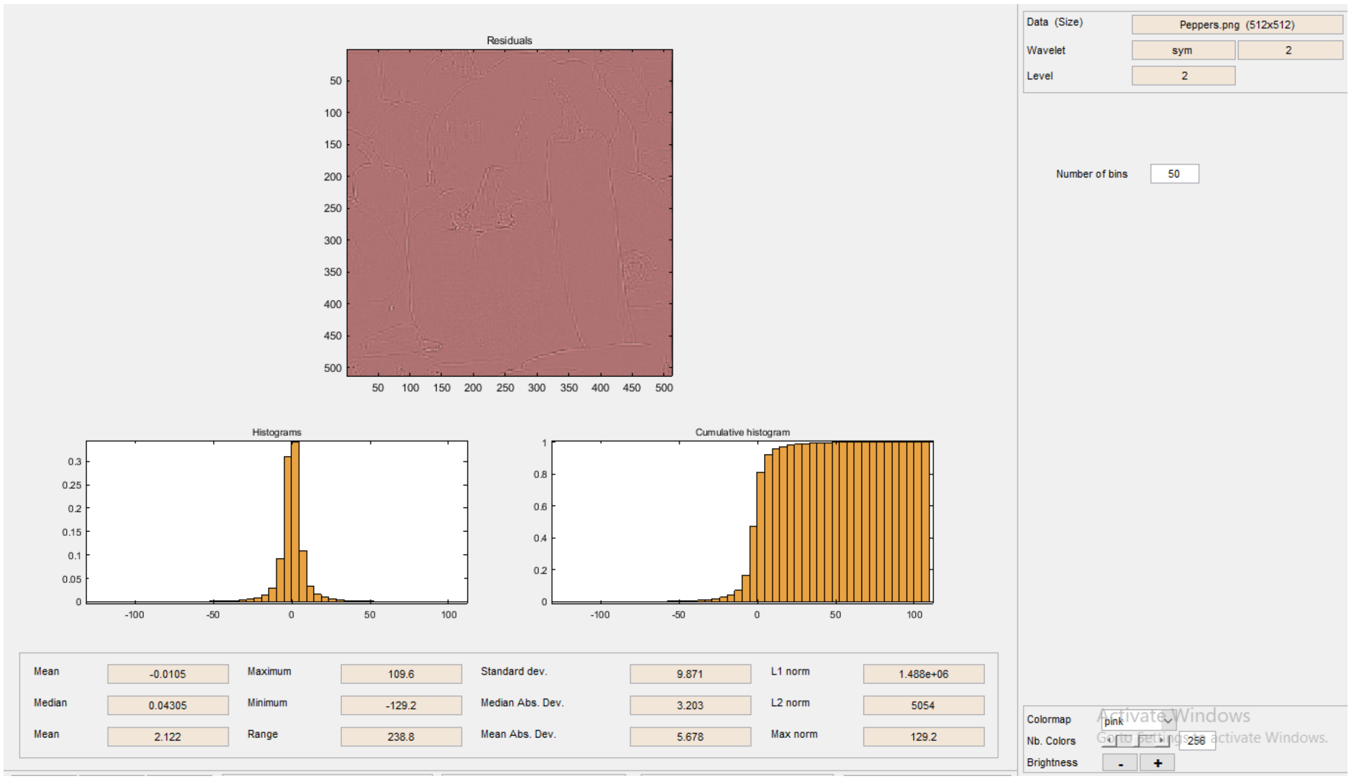

Fig.29: Decomposition Histogram

The next figure explains the retained energy and number of zeros in the compressed image to better understand the compression of each decomposition. The figure shows 99.35% of energy has been retained and 93.64% of the number of zeros have been removed. This constitutes a good compression.

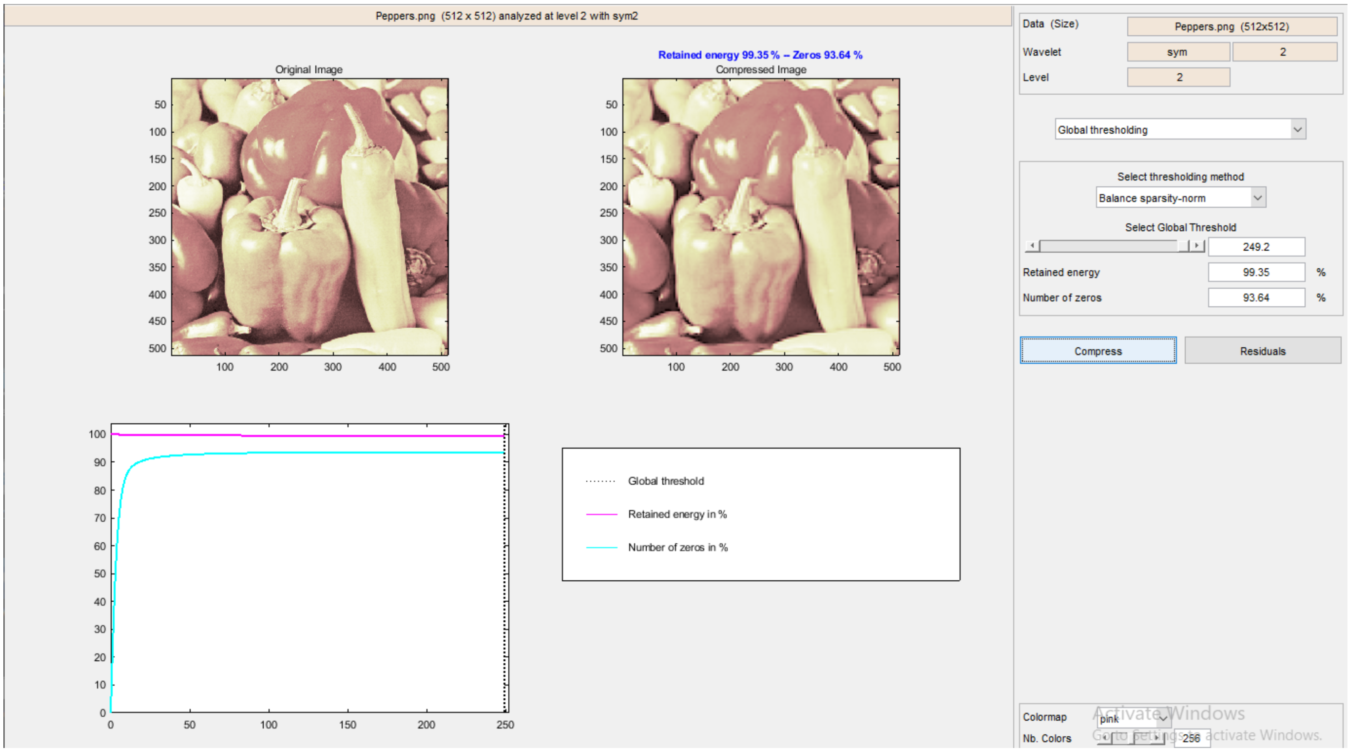

Fig.30: Sym2 Compression Statistics

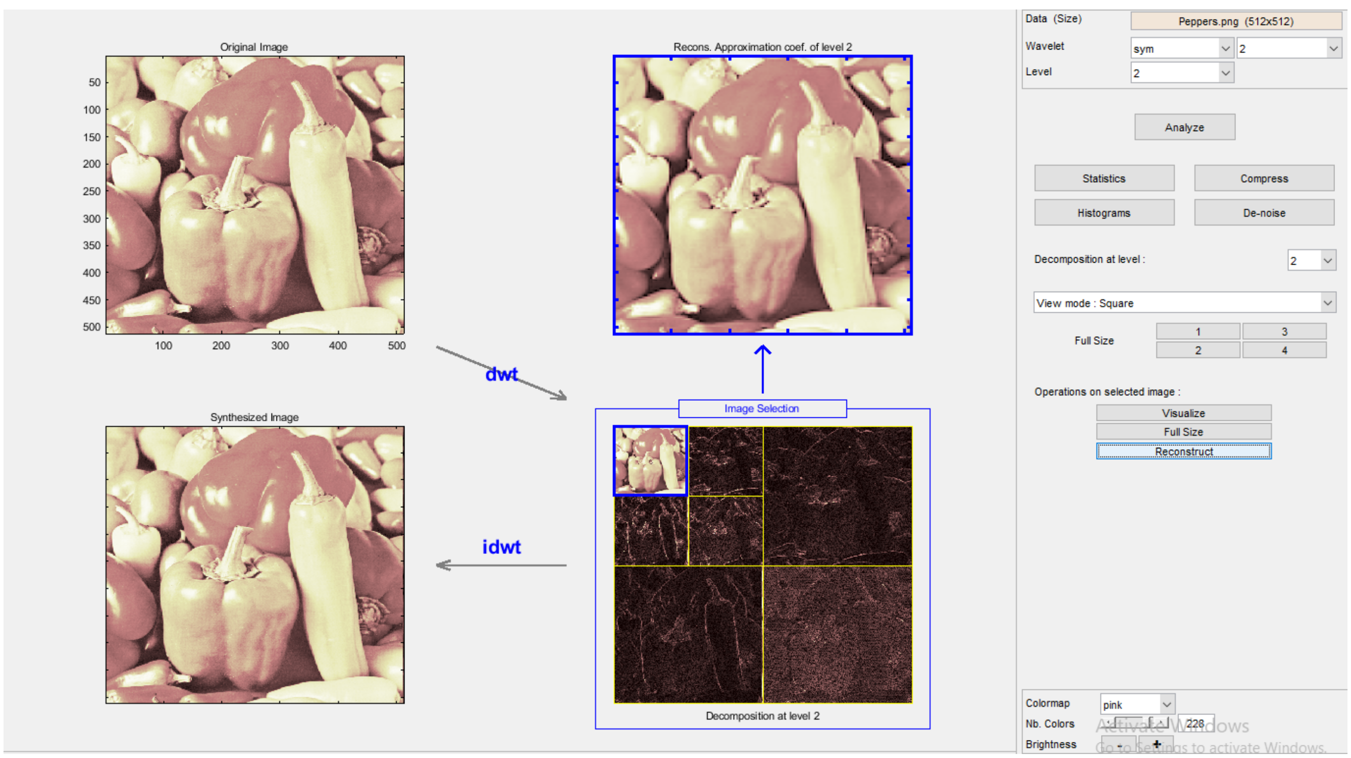

Fig.31: Sym2 Reconstruction DWT/IDWT Operation

The above figure shows successful reconstruction of the peppers image using the Symlet Wavelet level 2 decomposition, The demonstration of DWT and IDWT has also been pictured.

Symlet wavelet – Level 3 Decomposition

The following are the image obtained from 3 level decomposition using the toolbox.

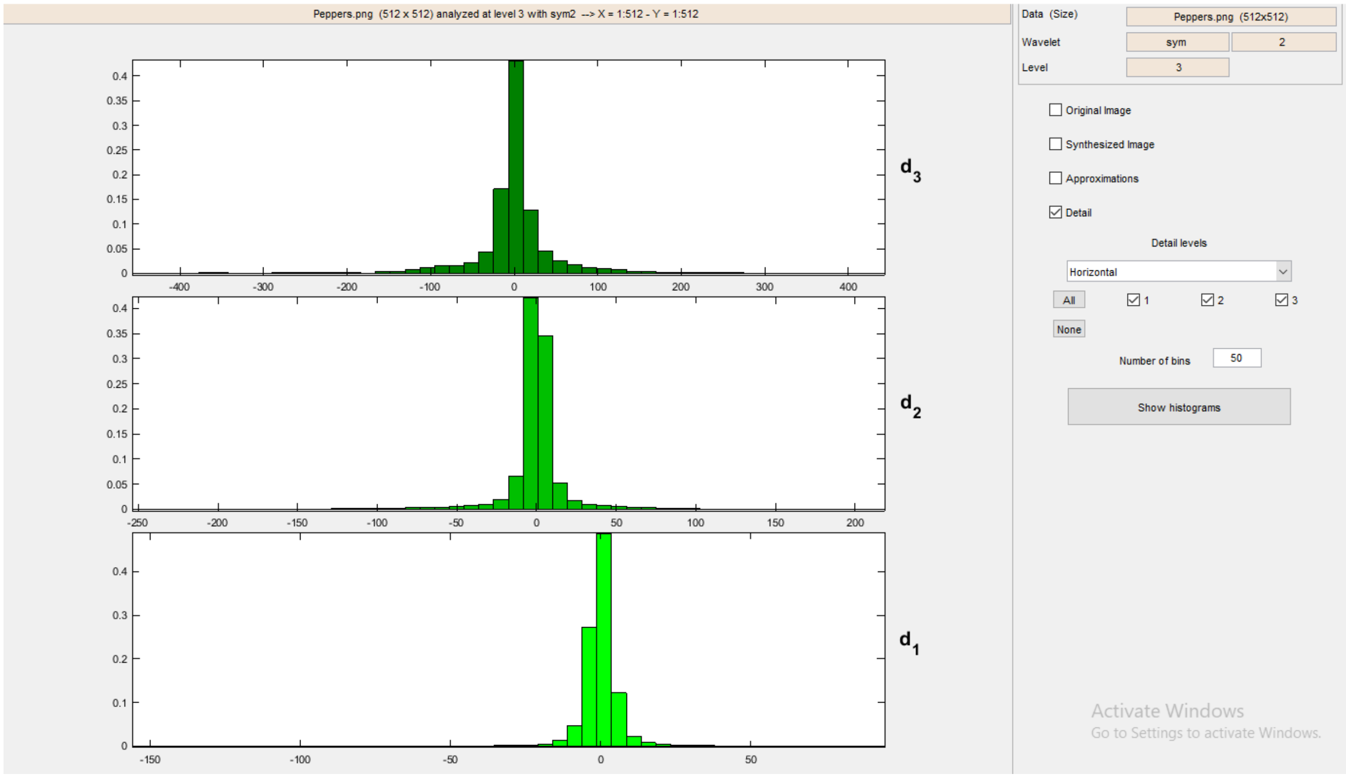

Fig.32: Sym3 Horizontal Details

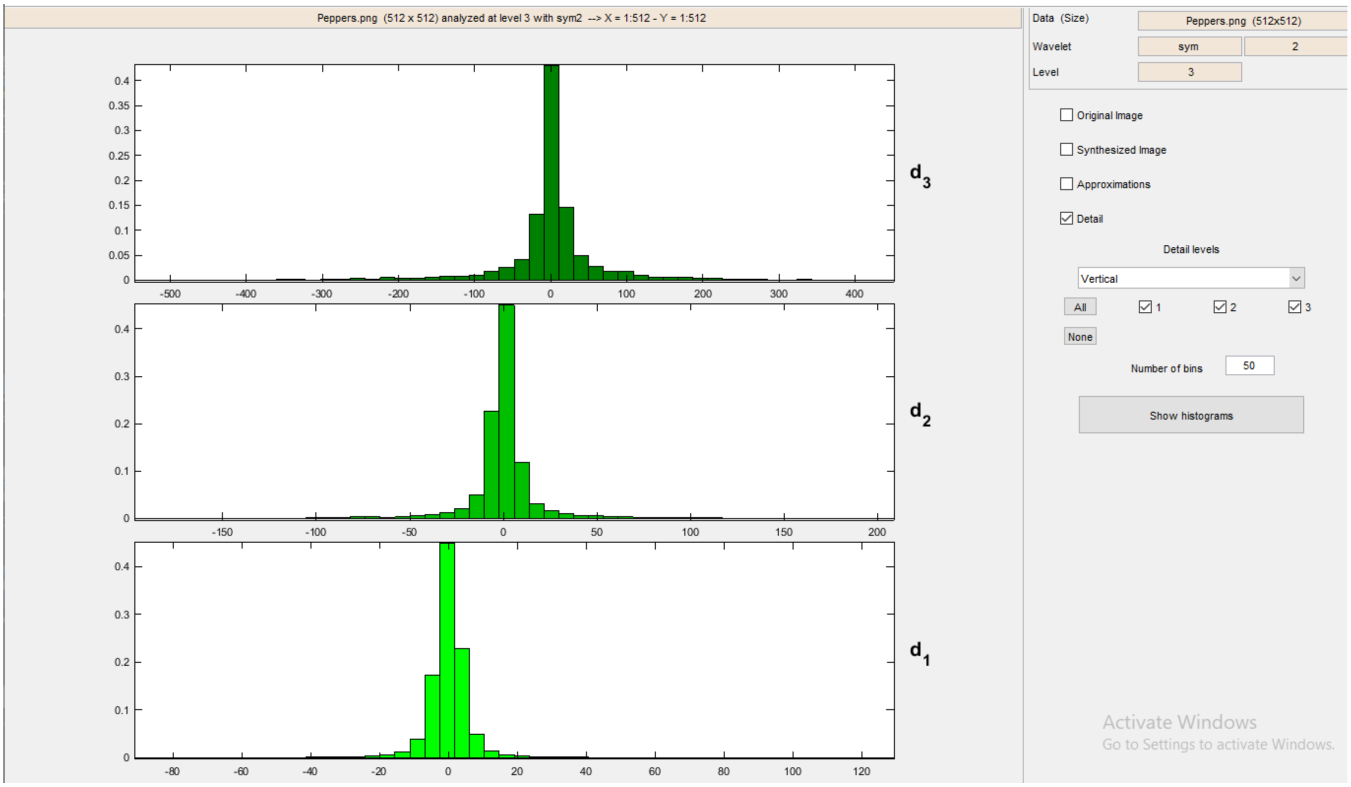

Fig.33: Sym3 Vertical Details

The histograms above provide information on the frequency-distribution of the synthesized image and both detail level of all the levels of image in vertical, diagonal, and horizontal domain. The histogram represents the wavelet detail coefficients of the original image. The three rows show the histograms of three levels of decomposition.

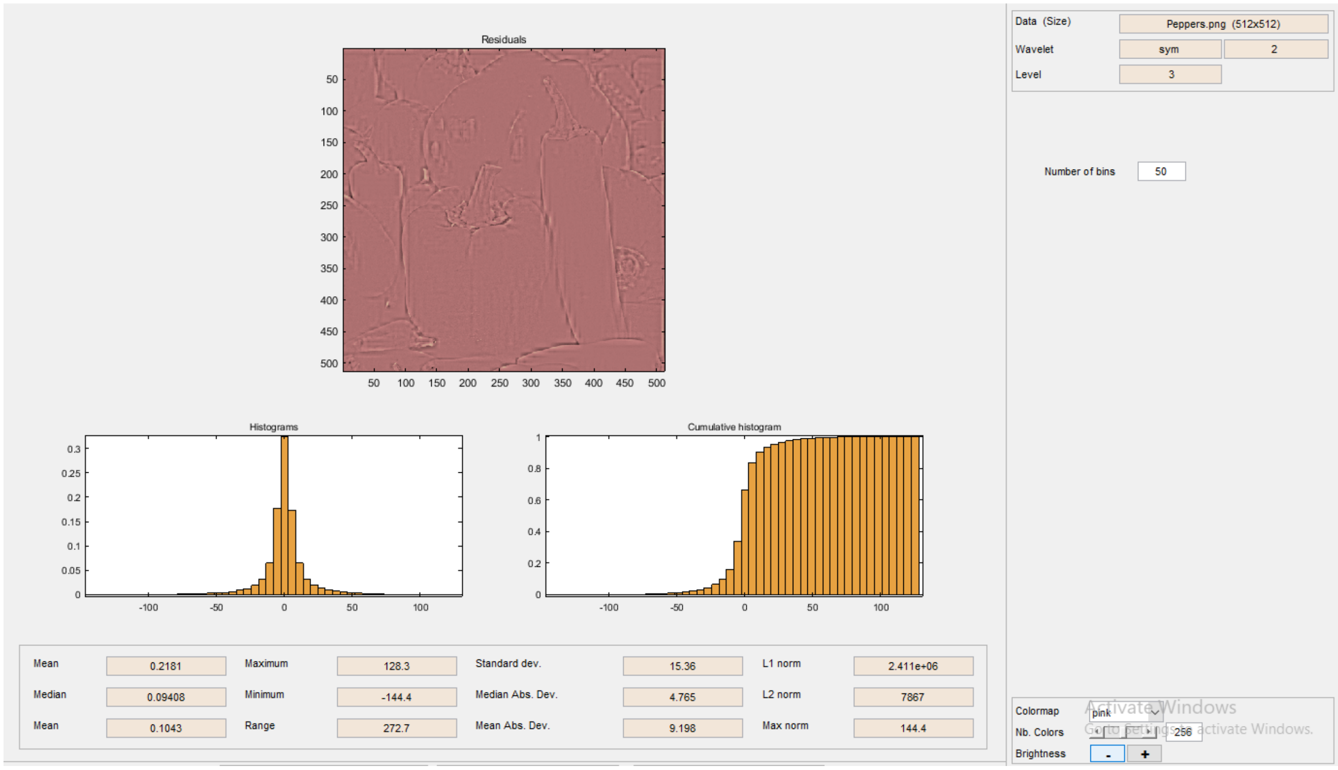

Fig.34: Decomposition Histogram

The next figure explains the retained energy and number of zeros in the compressed image to better understand the compression of each decomposition. The figure shows 98.51% of energy has been retained and 98.36% of the number of zeros have been removed. This constitutes a good compression.

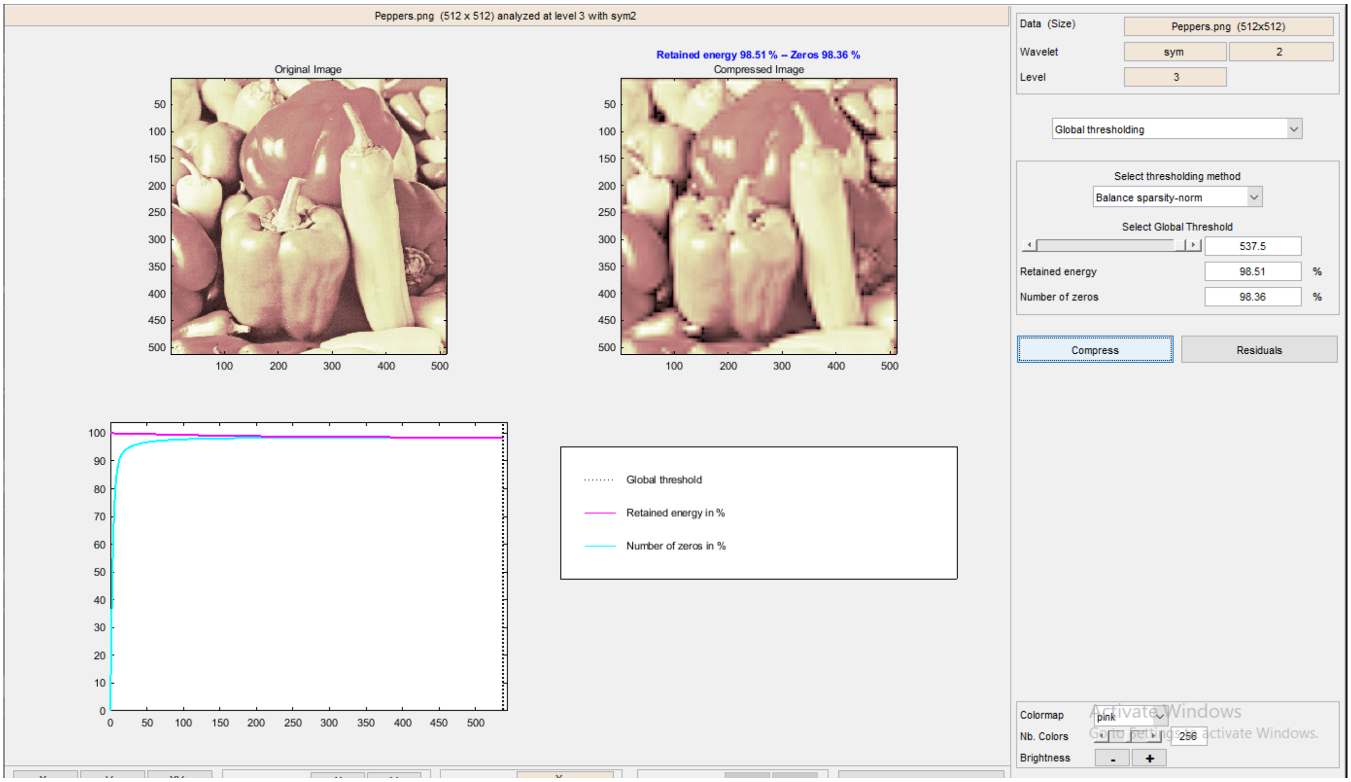

Fig.35: Sym3 Compression Statistics

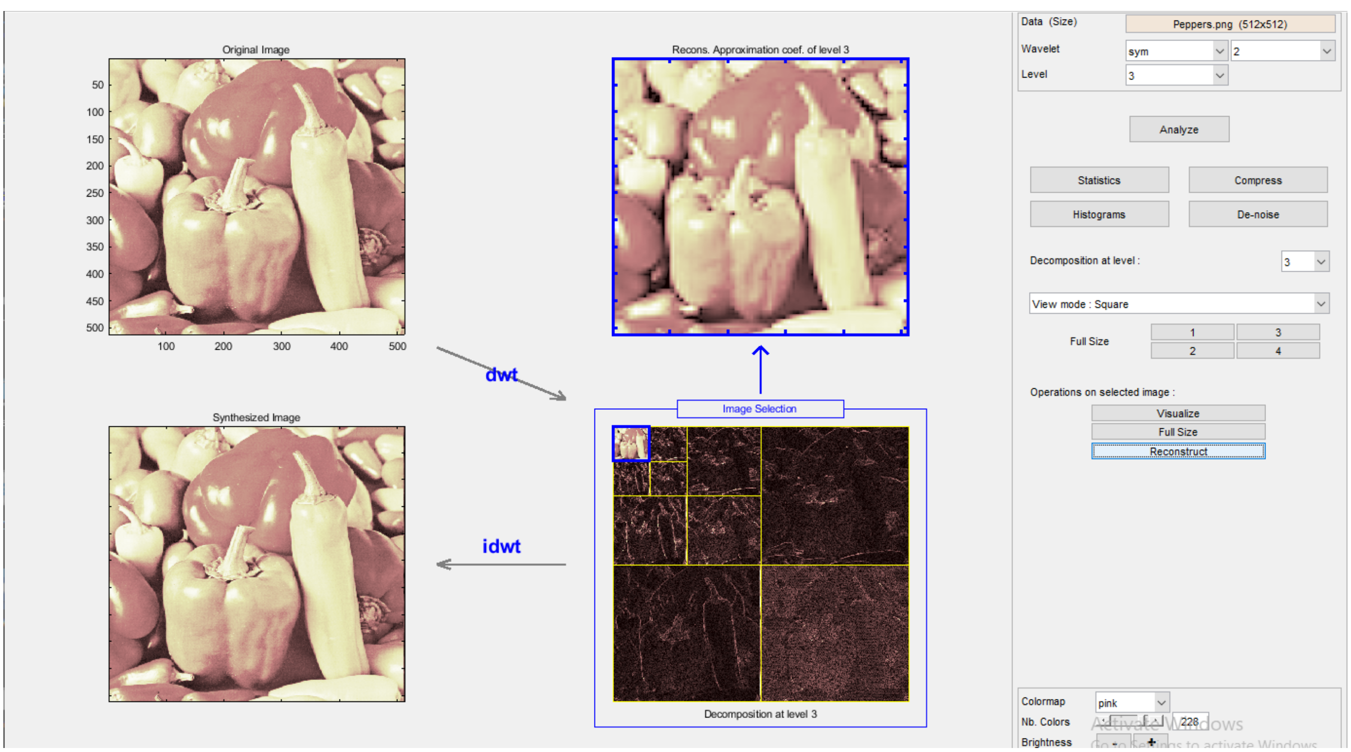

Fig.36: Sym3 Reconstruction DWT/IDWT Operation

The above figure shows successful reconstruction of the peppers image using the Symlet Wavelet level 2 decomposition, The demonstration of DWT and IDWT has also been pictured.

Tabular Column

DWT	Db2	Db3	Sym2	Sym3
Energy Retained	**99.24%**	**98.39%**	**99.35%**	**98.51%**
Number of Zeros	**93.50%**	**98.39%**	**93.64%**	**98.36%**

Conclusion

DCT and DWT was successfully applied on the peppers image provided and the results for both the transforms where compared. In DCT implementing partitioning of image using blocks gave better results than applying DCT throughout the image without partitioning. The energy-based threshold should be set high in order to get better performance. The coefficient close to zero should be removed. In DWT applying Level 3 compression will give us better compression than Level 2 compression but as levels increase complexity and data loss also increases. The image compression of DWT is better than DCT. Thus, using Discrete wavelet transforms for image compression will be more optimal than that of using Discrete Cosine transforms. But for applications where the data output image should be clear DCT can be used by finding the optimal threshold factor.

References

[1] Dr. M.R. Azimi Lecture Slides

[2] http://www.eecs.umich.edu/courses/eecs206/archive/spring02/lab.dir/Lab5/lab5v30release.pdf [3] R. J. Cintra and F. M. Bayer, ”A DCT Approximation for Image Compression,” in

IEEE Signal Processing Letters, vol. 18, no. 10, pp. 579-582, Oct. 2011.

[4] A. S. Lewis and G. Knowles, ”Image compression using the 2-D wavelet transform,” in IEEE Transactions on Image Processing, vol. 1, no. 2, pp. 244-250, April 1992.

[5] A. Skodras, C. Christopoulos and T. Ebrahimi, ”The JPEG 2000 still image compres- sion standard,” in IEEE Signal Processing Magazine, vol. 18, no. 5, pp. 36-58, Sept. 2001.

[6] Mathworks/ Matlab Example/ Image Compression with the Discrete Cosine Trans-

form.

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

[8] Michel Misiti, Yves Misiti, Georges Oppenheim, Jean-Michel Poggi, "Wavelet Tool- box User Guide," Version 3, Mathworks.

[9] https://en.wikipedia.org/wiki/Signal-to-noiseratio