ECE 513: Computer Assignment 5

DATA COMPRESSION AND ENCODING

Abhishek Balasubramaniam
CSU ID:832535894

Due Date: May 9, 2020

1 Introduction

Image compression is a type of data compression applied to digital images that encodes the original image with few bits. The objective of image compression is to reduce the redundancy of the image and to store or transmit data in an efficient form [1]. Algorithms or Systems that encode and decode images are called Image encoders, data compressors, or source encoders [2]. The block diagram of a Image Compression system is given in Figure 1. In this assignment we use the DWT transformed Pepper.mat image and apply suitable encoding scheme and study the Bpp and Snr ratio.

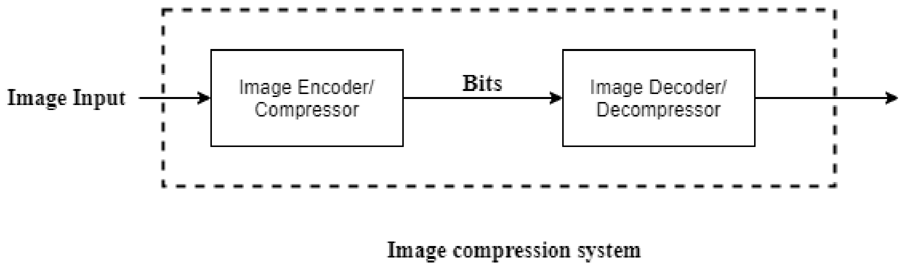


Figure 1: Block diagram of Image compression

2 Theory

2.1 Data compression

Data compression is the process of modifying, encoding or converting the bits structure of data in such a way that it consumes less space on disk [3]. It enables reducing the storage size of one or more data instances or elements. Data compression is also known as source coding or bit-rate reduction. Image compression is a form of data compression where we remove or group together certain parts of an image file in order to reduce its size [4].

There are two type of Data compression:

1. Lossless compression.

2. Lossy compression.

2.1.1 Lossless compression

Lossless compression is a method used to reduce the size of a file while maintaining the same quality as before it was compressed.Usually this is achieved by removing unnecessary meta data from JPEG and PNG files [5].

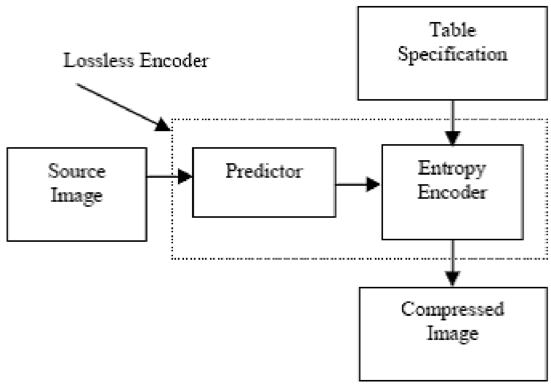


Figure 2: Block diagram of lossless Image compressor

2.1.2 Lossy compression

Lossy compression means that some data from the original image file is lost. This process is irreversible. The ”lossyness” of an image file may show up as jagged edges or pixelated areas. Lossy compression removes data from the original file, the resulting file often takes up much less disk space than the original [5].

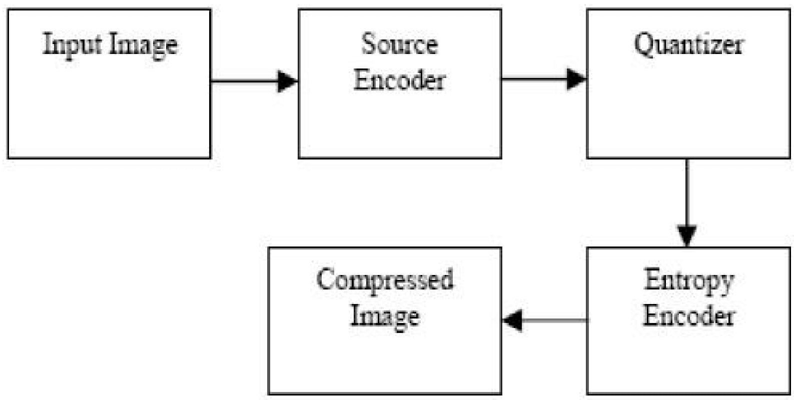


Figure 3: Block diagram of lossy Image compressor

The amount of compression achieved is typically described in two equivalent ways [6]:

1. Data Compression Ratio (DCR).
2. percentage savings.

2.2 Data Encoding

In order to compress an image there are a number of image encoding techniques [7]. They are:

1. **Pixel Encoding.** Eg: Huffman,entropy,PCM,etc.

2. **Predictive Encoding.** Eg: Delta modulation, 2-D DPCM, inter-frame method.

3. **Transform-based Encoding.** Eg: DCT-based, WT-based, Zonal encoding.

4. **Others.** Eg: Vector quantization (clustering), neural network-based, hybrid encoding.

There are three steps to encode an image using any encoding system. They are as follows:

1. Mapping.

   - Removes redundancies in the images and are invertible.

2. Quantization.

   - Mapped values are quantized using uniform or Llyod-Max quantizer.

3. Coding.

   - Optimal codewords are assigned to the quantized values.

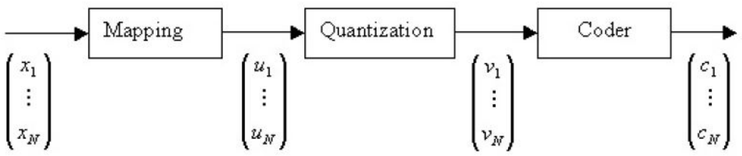


Figure 4: Block diagram of a typical encoding system

In this assignment we use pixel encoding in which each pixel is encoded while ignoring their inter-pixel dependencies.

The different methods used in pixel encoding are:

1. Entropy Encoding.

2. Run-length encoding.

3. Huffman coding.

2.2.1 Entropy Encoding

Entropy encoding is a method of lossless compression that is performed on an image after the quantization stage. It enables one to represent an image in a more efficient way with less memory needed for storage or transmission [8].

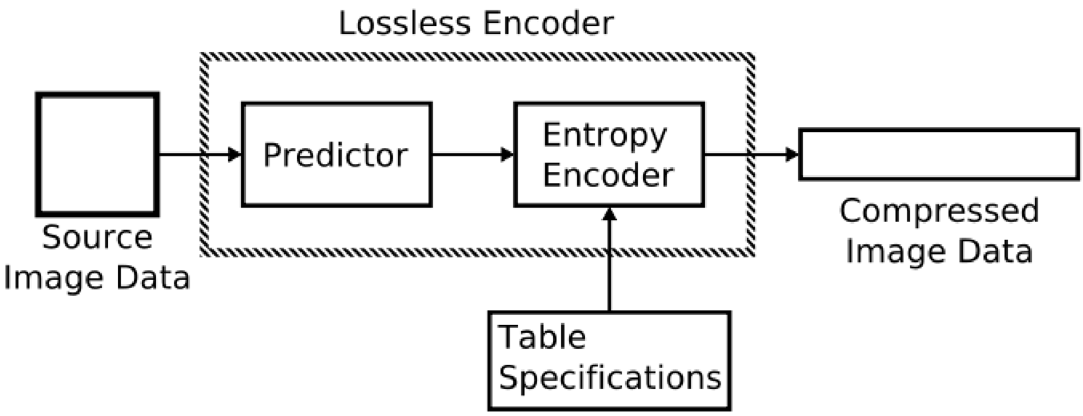


Figure 5: Block diagram of Entropy encoder

2.2.2 Run-length Encoding

Run-length encoding is a data compression algorithm that is supported by most bitmap file formats, such as TIFF, BMP, and PCX. RLE is suited for compressing any type of data regardless of its information content, but the content of the data will affect the compression ratio achieved by RLE. Although most RLE algorithms cannot achieve the high compression ratios of the more advanced compression methods, RLE is both easy to implement and quick to execute, making it a good alternative to either using a complex compression algorithm or leaving your image data uncompressed [9].

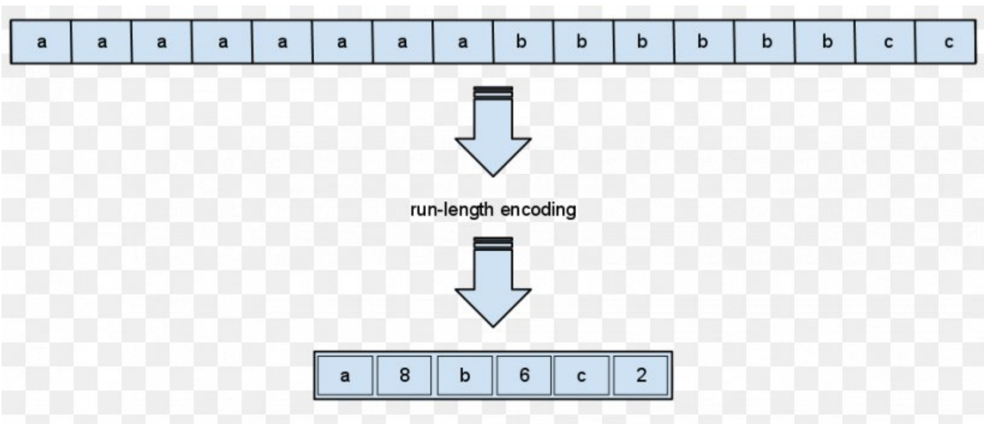


Figure 6: Block diagram of RLE Algorithm

2.2.3 Huffman coding

Huffman coding is an efficient method of compressing data without losing information. Huffman coding assigns codes to characters such that the length of the code depends on the relative frequency or weight of the corresponding character. Huffman codes are of variable-length, and prefix-free (no code is prefix of any other). Any prefix-free binary code can be visualized as a binary tree with the encoded characters stored at the leaves [10].

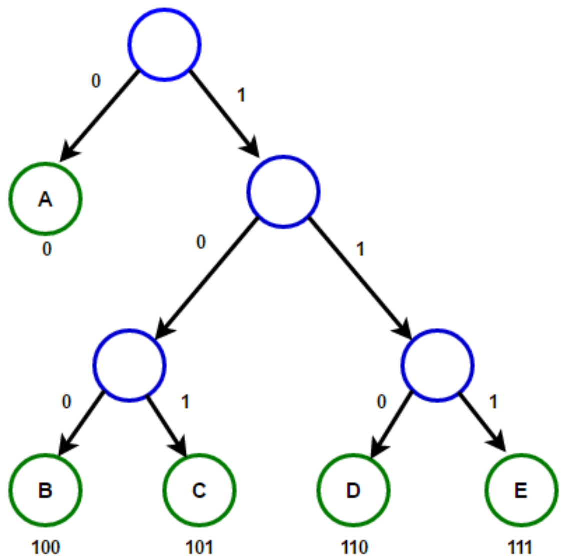


Figure 7: Block diagram of Huffman Encoding

The process of Huffman coding is described below [11]:

1. Create a leaf node for each symbol and add it to the priority queue.
2. While there is more than one node in the queue:
   (a) Remove the two nodes of highest priority from the queue.
   (b) Create a new internal node with these two nodes as children and with frequency equal to the sum of the two nodes' frequency.
   (c) Add the new node to the queue.
3. The remaining node is the root node and the Huffman tree is complete.

The following is the Pseudo-code of Huffman Encoding.

Algorithm 1 Pseudo-code for Huffman Encode

procedure $Huffman(C):$ // C is the set of n characters and related information

$n = C.size$

$Q = priorityqueue()$

for $i = 0$ **to** n **do**

  $n = node(C[i])$

  $Q.push(n)$

while $Q.size() \neq 1$ **do**

  $Z = newnode()$

  $Z.left = x = Q.pop$

  $Z.right = y = Q.pop$

  $Z.frequency = x.frequency + y.frequency$

  $Q.push(Z)$

return Q

end procedure

2.3 Encoding for DWT transformed image

The Huffman encoding is applied to a DWT transformed image. The compression and encoding is done using the 'WaveletAnalyzer' application of MATLAB. The threshold value and the bits per pixel values are changed and the resulting SNR are observed. The block diagram of a wavelet compression is as follows [12]:

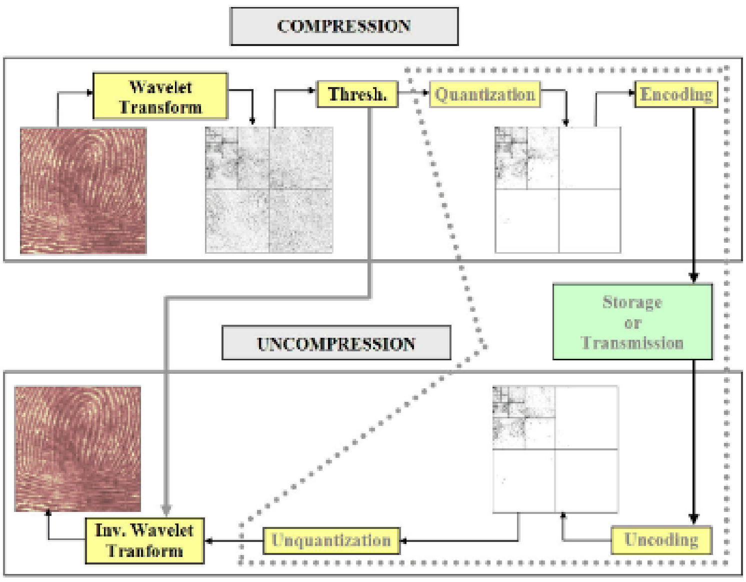


Figure 8: Block diagram of a wavelet Compression

We use the Bpp and Snr value observed from the 'WaveletAnalyzer' to check the performance of the compression and also plot them in a graph.

2.4 Signal to Noise Ratio:

Since the reconstructed image has some Noise in it the SNR for the original image and Reconstructed image is calculated using the following formula [13].

$$SNR = 10log_{10}\frac{\sigma_o^2}{\sigma_e^2} \tag{1}$$

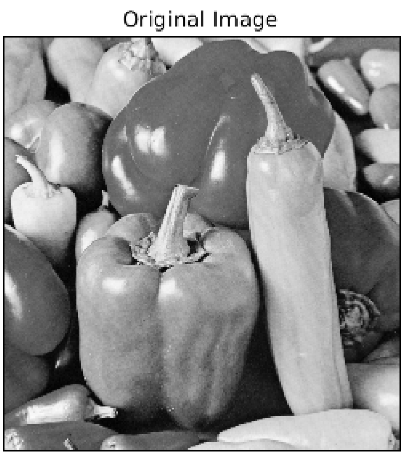
In 1 the σ_o^2 is the variance of the original image and σ_e^2 is the variance of the error image. Using this SNR value we can calculate the loss in the compression of the image as well as the quality of the image.

3 Result

The 'waveletAnalyzer' is used for compression and encoding using Huffman Encoding. The compression of the data is done using two type of wavelets namely the Daubechies Wavelets and Symlet Wavelet. For this process the Peppers image is used.

3.1 Daubechies Wavelets

The input image is loaded in gray scale format. Level 2 decomposition is applied on the image and then this image is used for encoding using Huffman encoding. The following are the input and output images.



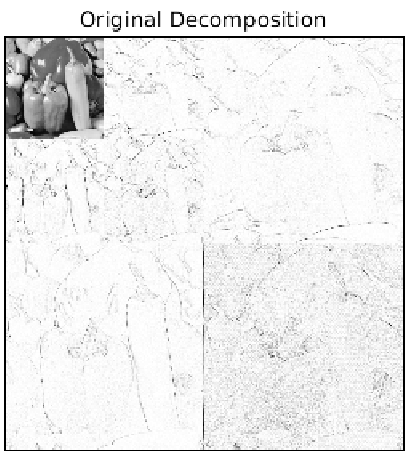


Figure 9: Original images and Decomposition

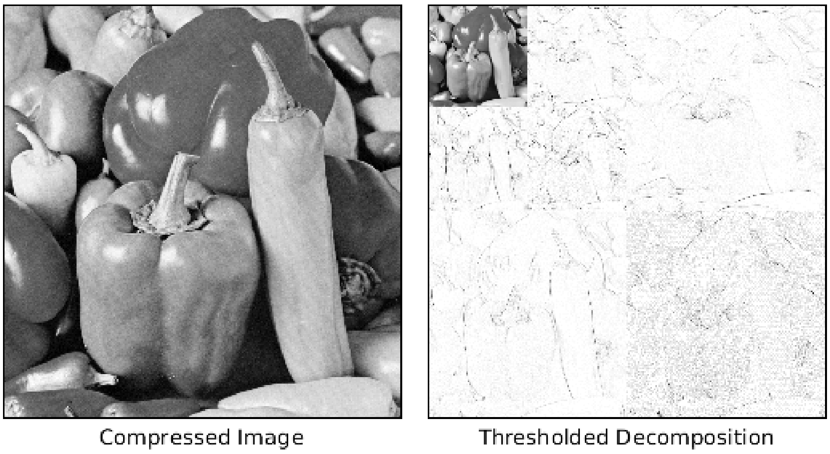


Figure 10: Encoded images and Decomposition

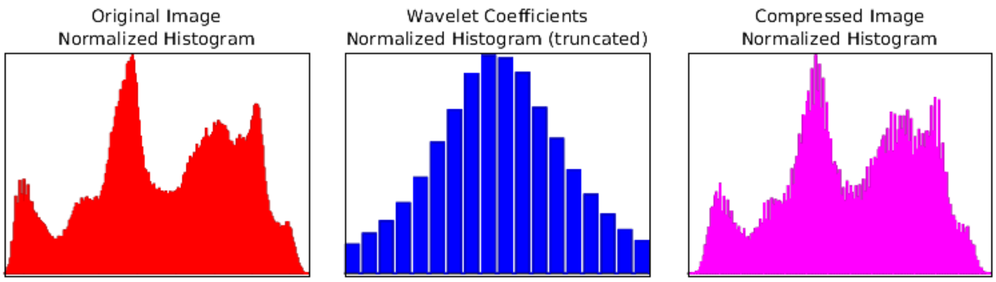


Figure 11: Histogram

The following table has the observed input BPP, SNR, BPP, and Compression ratio.

Input BPP	SNR	Observed BPP	Compression ratio (%)
8	38.03	2.9685	37.11
6	38.03	2.9685	37.11
2	37.33	2.8546	35.68
1	35.8	1.3275	16.59
0.5	28.91	0.73752	9.22

Table 1: Bpp and Snr analysis table.

The above mentioned SNR and BPP is used to analyze the compression characteristics. The value of SNR and Observed BPP after compression remains same for values from 8 to 6 input BPP. The SNR ratio and observed BPP goes down for input BPP values below 2. The 12 show the snr vs bpp values.

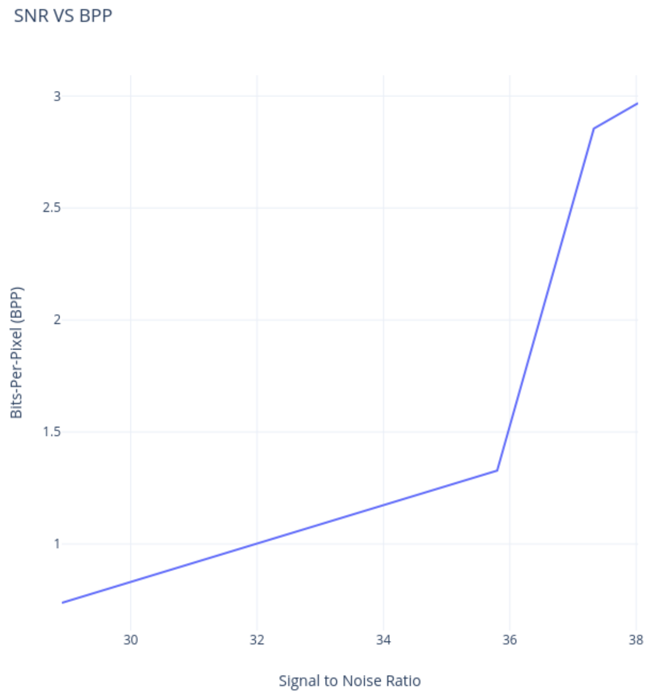


Figure 12: snr vs bpp for Daubechies Wavelets

3.2 Symlet Wavelet

The input image is loaded in gray scale format. Level 2 decomposition is applied on the image and then this image is used for encoding using Huffman encoding. The following are the input and output images.

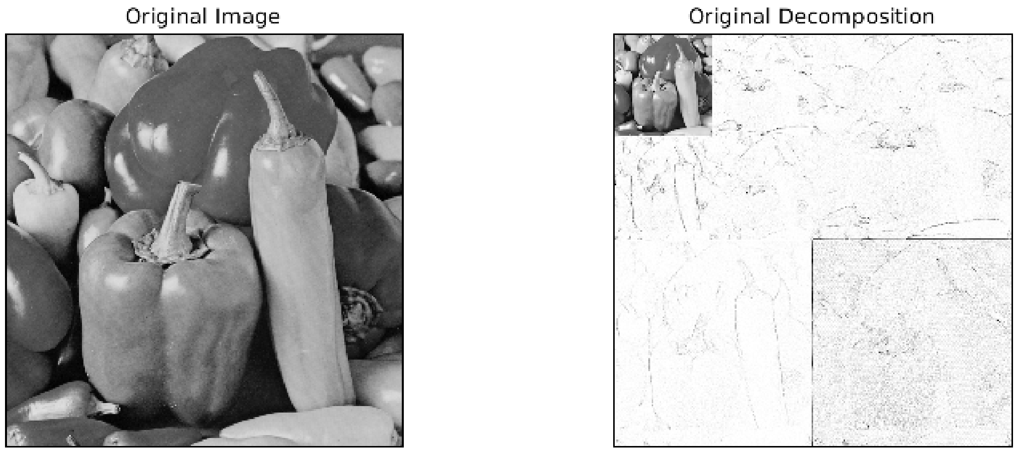


Figure 13: Original images and Decomposition

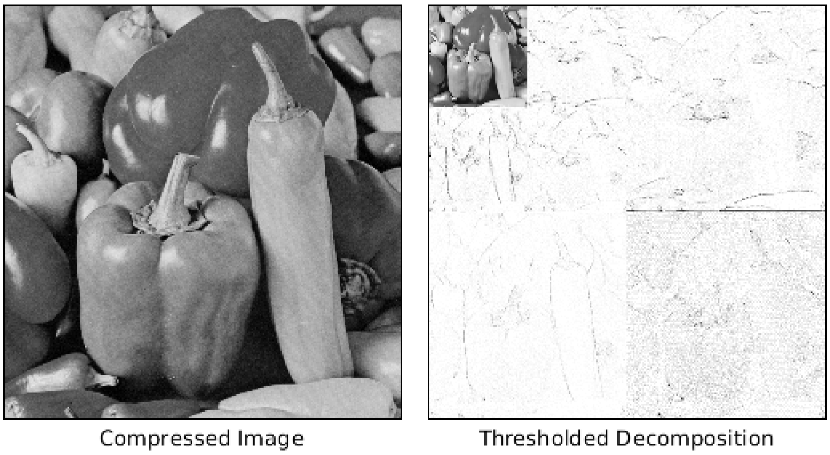


Figure 14: Encoded images and Decomposition

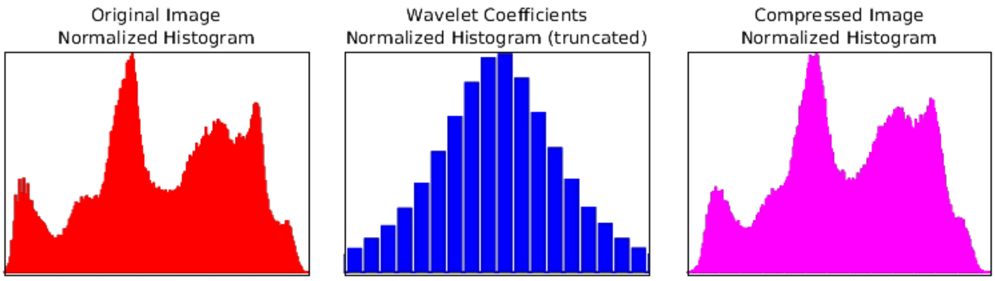


Figure 15: Histogram

Input BPP	SNR	Observed BPP	Compression ratio (%)
8	37.83	3.045	38.06
6	37.83	3.045	38.06
2	37.17	2.829	35.36
1	36.82	1.8175	22.72
0.5	30.55	0.77341	9.67

Table 2: Bpp and Snr analysis table.

The above mentioned SNR and BPP is used to analyze the compression characteristics. The value of SNR and Observed BPP after compression remains same for values from 8 to 6 input BPP. The SNR ratio and observed BPP goes down for input BPP values below 2. The 12 show the snr vs bpp values.

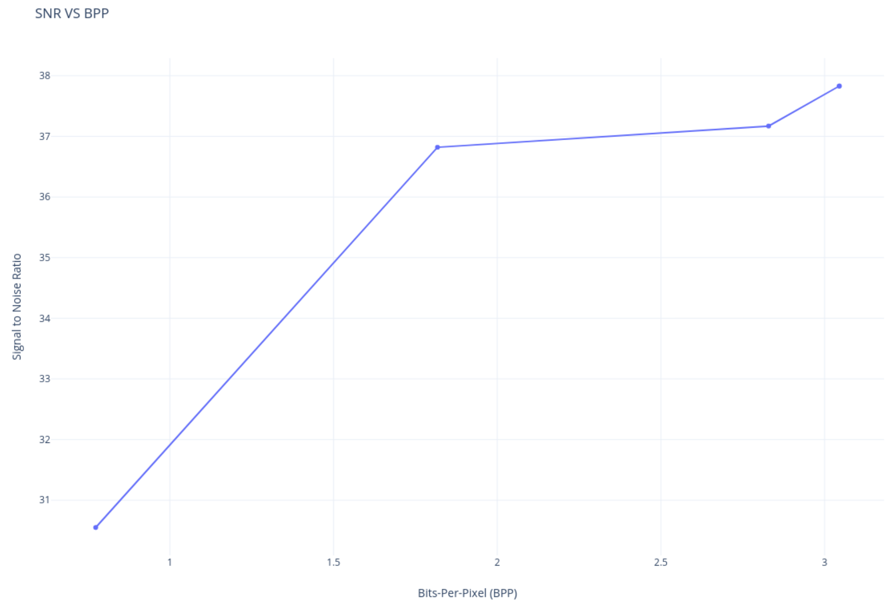


Figure 16: snr vs bpp for Symlet Wavelets

4 Discussions

From observing the above given graphs and image output we can see that finding the optimal input BPP increases the quality of the compressed image both visually and with SNR value. As the input BPP increase above 2 BPP the SNR, compressed BPP and compression ratio remains constant for both Daubechies Wavelets and Symlet Wavelet. The compressed image quality decreases below 0.5 BPP and there is a lot of data loss observed.

5 Conclusion

Thus the Huffman encoding is used for data compression on images using DWT transformed images. The Huffman encoding performed similar on both Daubechies Wavelets and Symlet Wavelet which can be seen with the output SNR recorded. The image also consist of all the important features of the image which gives us the idea of Huffman encoding is a lossless technique. The only drawback for Huffman encoding is that the codes have variable lengths.

References

[1] A. Skodras, C. Christopoulos, and T. Ebrahimi. "The JPEG 2000 still image compression standard." In: *IEEE Signal Processing Magazine* 18.5 (2001), pp. 36–58.

[2] Andrew E. Yagle. *Images, Compression, and Coding*. URL: https://www.eecs.umich.edu/courses/eecs206/archive/spring02/lab.dir/Lab5/lab5_v3_0_release.pdf.

[3] David Hemmendinger. *Data Compression*. URL: https://www.britannica.com/technology/data-compression.

[4] Abir Hussain, Ali Al-Fayadh, and Naeem Radi. "Image Compression Techniques: A Survey in Lossless and Lossy algorithms." In: *Neurocomputing* 300 (Mar. 2018). DOI: 10.1016/j.neucom.2018.02.094.

[5] K. A. Ramya and M. Pushpa. "A Survey on Lossless and Lossy Data Compression Methods." In: *International Journal of Computer Science and Engineering Communications* 4 (Mar. 2016), pp. 1277–1280.

[6] Schmitz. *INFORMATION AND COMPRESSION*. URL: https://courses.engr.illinois.edu/ece110/sp2020/content/courseNotes/files/?information.

[7] Ritu Mudgil Sonal Chawla Meenakshi Beri. "Image Compression Techniques: A Review." In: *International Journal of Computer Science and Mobile Computing* 3 (Aug. 2014), pp. 291–296. DOI: https://www.ijcsmc.com/docs/papers/August2014/V3I8201466.pdf.

[8] Wikipedia. *Entropy encoding*. URL: https://en.wikipedia.org/wiki/Entropy_encoding.

[9] Scott Robinson. *Run-Length Encoding*. URL: https://stackabuse.com/run-length-encoding/.

[10] Jimin Khim Karleigh Moore. *Huffman Code*. URL: https://brilliant.org/wiki/huffman-encoding/.

[11] Prof. M.R.Azimi. *Lectures 25 26*. URL: https://www.engr.colostate.edu/ECE513/SP09/lectures/lectures25_26.pdf.

[12] MathWorks. *2-D Wavelet Compression*. URL: https://www.mathworks.com/help/wavelet/ug/two-dimensional-true-compression.html.

[13] Wikipedia. *Signal-to-noise ratio*. URL: https://en.wikipedia.org/wiki/Signal-to-noise_ratio.