A study on different Image Compression alogithm to develop an Efficient loseless Encoder

Report submitted to

Indian Institute of Information Technology, Allahabad
(An Institute of National Importance by Act of Parliament,
Established by MHRD, Government of India)

for Summer Internship 2019

By Yash Gupta

Bachelor of Technology
In
Electrical Engineering

Institute of Engineering & Technology
Affilated to Dr. A.P.J. Abdul Kalam Technical University
Jankipuram, Lucknow

CERTIFICATE

I hereby state that the project entitled "**A study on different Image Compression algorithm to develop an Efficient lossless Encoder**" is an authentic work under the guidance of **Dr. Mohammed Javed** (Assistant Professor, Department of Information Technology (IT), Indian Institute of Information Technology (IIIT-Allahabad)) and **Mr. Rajesh** (Research Scholar, Department of Information Technology (IT), Indian Institute of Information Technology (IIIT-Allahabad))

I Yash Gupta a 2nd year student of Electrical Engineering Department was working as Research Intern under **Dr. Mohammed Javed at IIIT Allahabad** for summer 2019.

Yash Gupta

EE – 2nd Year

ACKNOWLEDGEMENT

I would like to thank **Dr. Mohammed Javed** and **Mr. Rajesh** for their valuable time and efforts for giving me the opportunity to work on this project.

I would like to thank all my colleagues, PhD scholars of Computer Vision and Biometrics Lab, Security guard and other staff members for providing me with the opportunity and facilities including the internet connection and workspace to work on the Research project.

INDEX

INTRODUCTION

1.1 What is the need of Image Compression ?

The incessant expansion of multimedia applications, the needs and the requirements of the technologies used, involves enormous quantities of data. Compression is the technique that reduces the original amount of data to another less quantity leading to an efficient transmission and storage process. This is achieved by detecting and removing redundant information from the original data.

1.2 Techniques Involved :

Image compression techniques can be broadly classified into two categories namely **lossy and lossless compression** techniques. In some applications, such as medical diagnosis, We may need the reconstructed image to mirror the original image. This type of compression is called lossless image compression. In other applications, such as motion picture and television, a certain amount of information loss is allowed and this type of compression is referred to as lossy image compression. The latter has been widely studied in the domain literature and it is generally categorized as direct methods, which handles directly the image samples such as in Block Truncation Coding (BTC) and Vector Quantization (VQ) based techniques , and transform - based methods which use mathematical transforms to concentrate the energy of the image in a few number of coefficients such as principal component analysis (PCA) , discrete cosine transform (DCT) and discrete wavelet transform (DWT) . A small loss in performance is detected, and a computational efficiency is observed for the DCT-based methods compared to those using DWT . Based on empirical performance results , the main factors in image coding are the quantizer and entropy coder rather than the difference between the wavelet and the DCT transform. For this reason, the state-of-art image and video coding standards and multimedia devices prefer DCT over DWT . The DCT is a fundamental building block for several image and video processing applications. Thus, the DCT finds applications in several contemporary image and video compression standards, such as the JPEG .

IMAGES & IMAGE COMPRESSION

2.1 Types of Images:

Types of lossless images include:

RAW

Found in many DSLRs, and keeps all the light data received from the camera's sensor. For a professional, this great news. However, these files types tend to be quite large in size. Additionally, there are different versions of RAW, and you may need certain software to edit the files.

PNG

Compresses images to keep their small size by looking for patterns on a photo, and compressing them together. The compression is reversible, so once you open a PNG file, the image recovers exactly.

BMP

A format found exclusively to Microsoft. It's lossless, but not frequently used.

It should also be noted that converting a lossy photo back to lossless won't restore the photo's data.

In order to give the photo an even smaller size, lossy compression discards some parts of a photo. However, this doesn't mean the photo will look bad. Here are the two main types of lossy compression.

JPG

Also known as JPEG, this format gets rid of bits and pieces of a photo that you may notice depending upon the level of compression you apply. A normal amount of compression will not be noticeable, while extreme compression may be obvious.

There are also other ways a JPG image's quality may be reduced. If you rotate the JPG too much, you'll notice a difference in quality. This is because the photo has to recompress itself with every rotation, losing some data in the process. There are however programs out there that rotate a JPG losslessly. The same degradation applies if you save a JPG multiple times.

GIF

GIF compresses files by reducing the number of colors it has. If the photo has more than 256 colors (the maximum amount of colors older computers could have) this format will make the image look less appealing. The best use for GIFs are for images that are animated.

Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. Lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. Lossy compression that produces negligible differences may be called visually lossless.

2.2 JPEG Image Compression System :

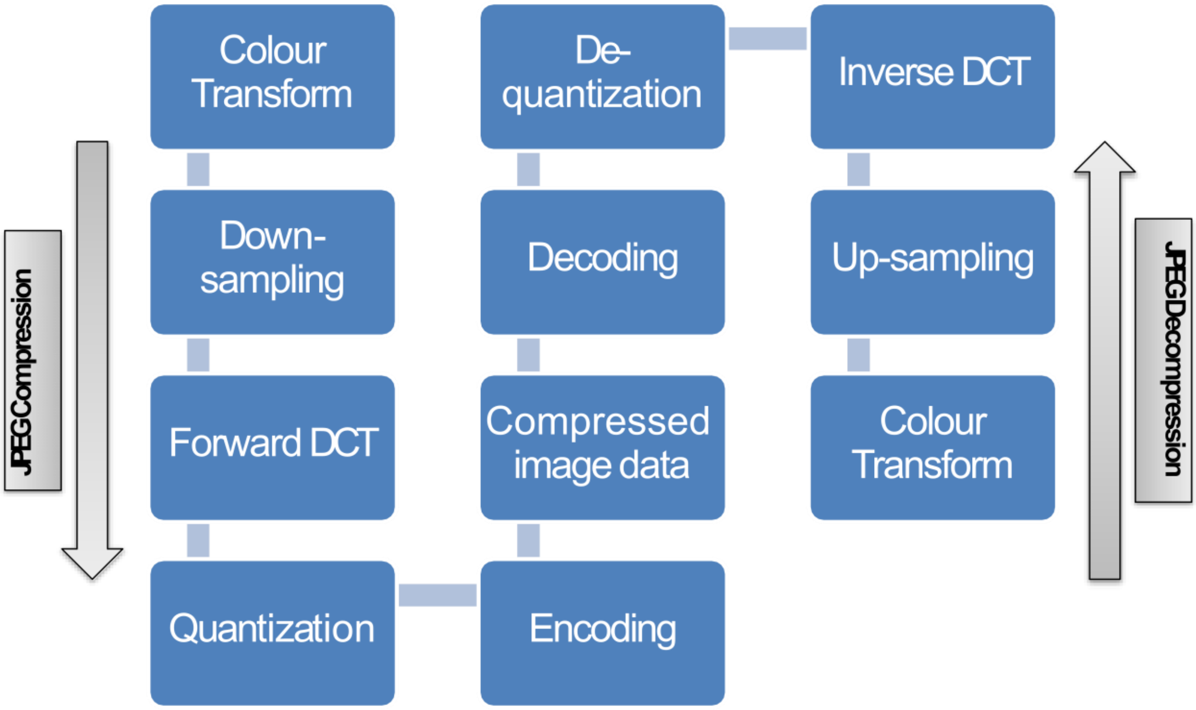

Fig: JPEG Schematic, This is the flowchart of JPEG Algorithm

Algorithm

- Splitting: Split the image into 8 x 8 non-overlapping pixel blocks. If the image cannot be divided into 8-by-8 blocks, then you can add in empty pixels around the edges, essentially zero-padding the image.

Original image

Image split into multiple 8x8 pixel blocks

Fig : This image shows that how the whole lena image is divided into (8x8) sub blocks

- Colour Space Transform from [R,G,B] to [Y,Cb,Cr] & Subsampling.
- DCT: Take the Discrete Cosine Transform (DCT) of each 8-by-8 block.

  Formula for 2D DCT is

$$\mathbf{G}_{u,v} = \frac{1}{4} \propto (u) \propto (v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} \cos \frac{(2x+1)}{16} \cos \frac{(2y+1)}{16}$$

- Quantization: quantize the DCT coefficients according to psycho-visually tuned quantization tables.

160	80	47	39	5	3	0	0
65	53	48	8	5	2	0	0
58	34	6	4	2	0	0	0
40	7	6	1	1	0	0	0
8	4	0	0	0	0	0	0
5	2	0	0	0	0	0	0
3	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT Coefficients F(u,v)

Quantizer

16	7	4	2	0	0	0	0
5	4	3	0	0	0	0	0
4	2	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	0	0	0	0	0	0	0

Quantized Coefficients **F**(u,v)

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

Quantization Table Q(u,v)

Each element of F(u,v) is divided by the corresponding element of Q(u,v) and then rounded off to the nearest integer to get the **F** (u,v) matrix.

Fig: Here an 8x8 DCT block of gray level lena image is taken which is then quantized by using quantization matrix.

- Serialization: by zigzag scanning pattern to exploit redundancy.

  In order to exploit the presence of the large number of zeros in the quantized matrix, a zigzag of the matrix is used.

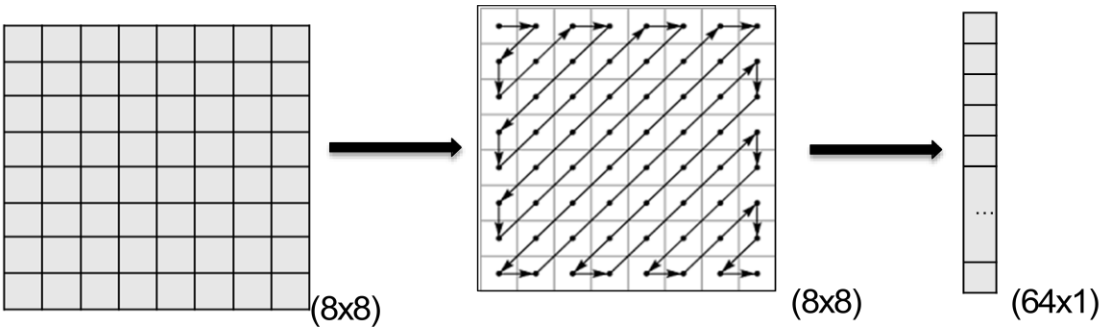

Fig : This diagram shows that how the 8x8 quantized block is converted into 64x1 vector using zig-zag scanning.

- Huffman coding

  Huffman coding is a compression technique used to reduce the number of bits needed to send or store a message. Huffman coding is based on the frequency of occurrence of a data item (pixel in images). The principle is to use a lower number of bits to encode the data that occurs more frequently.

LITERATURE REVIEW

Recent works in lossless Encoder was done in research paper "Color image compression algorithm based on the DCT transform combined to an adaptive block scanning" by Douak. This paper considers the design of a lossy image compression algorithm dedicated to color still images. After a preprocessing step (mean removing and RGB to YCbCr transformation), the DCT transform is applied and followed by an iterative phase (using the bisection method) including the thresholding , the quantization , dequantization, the inverse DCT, YCbCr to RGB transform and the mean recovering. This is done in order to guarantee that a desired quality (fixed in advance using the well known PSNR metric) is checked. For the aim to obtain the best possible compression ratio CR, then next step is the application of a proposed adaptive scanning providing, for each (nxn) DCT block a corresponding (nxn) vector containing the maximum possible run of zeros at its end. The last step is the application of a modified systematic lossless encoder which is based on TRE (two role encoder) . The efficiency of the proposed scheme is demonstrated by results, and was compared from the paper based on the "block truncation coding using pattern fitting principle".

PROPOSED METHODOLOGY

5.1 Image Compression

We propose to compress image using Efficient Lossless Encoder algorithm.
The complete method can be divided into four major sub tasks

- 5.1.1) Color Transform
- 5.1.2) Forward DCT
- 5.1.3) Thresholding & Quantization
- 5.1.4) Modified Lossless Encoder

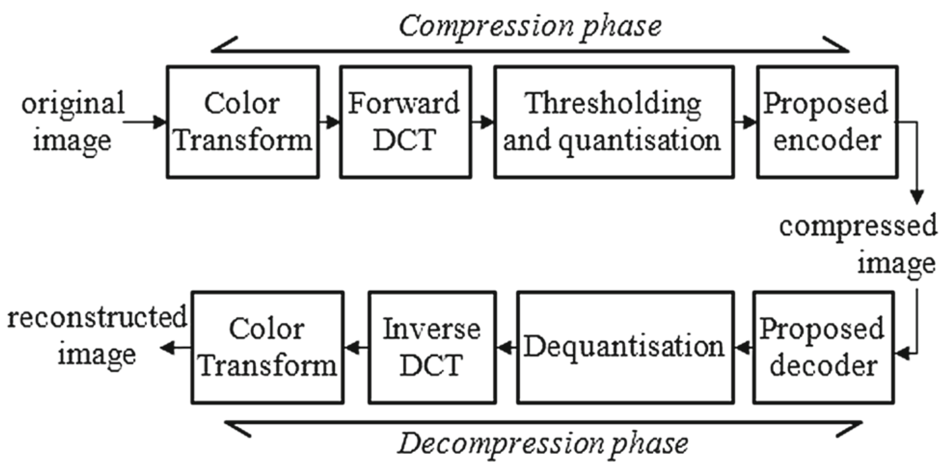

Fig : The proposed model to compress/decompress color image using an Efficient Lossless Encoder

5.1.1) Color Transform

First of all color transform from RGB to YCbCr is done so that the almost image energy of the transformed YCbCr image is contained in the Y plane. In this case, we can achieve an efficient compression that allows reaching high compression ratios in the Cb and Cr planes without loosing the quality of the whole compressed image.
Table 1 gives the energy distribution in RGB and YCbCr spaces of the used test images.

Image	ER (%)	EG	EB	EY(%)	ECb	ECr
Airplane	31.71	31.95	36.34	99.75	0.18	0.07
Peppers	55.04	33.79	11.17	90.56	5.87	3.57
Lena	60.30	19.06	20.64	89.37	0.83	9.80
Girl	31.95	33.04	35.01	99.91	0.05	0.04
Couple	50.24	27.30	22.46	94.51	1.31	4.18
House	35.62	29.92	34.46	98.55	0.39	1.06
Zelda	51.34	27.05	21.61	92.60	1.41	5.99
Average	45.17	28.87	25.96	95.04	1.44	3.53

Table 1 The energy distribution in RGB and YCbCr spaces of the used test images

5.1.2)Forward DCT

In DCT transform-domain image coding , each DCTcoefficient is related to all the pixels of the whole image; thus, it requires a large amount of computation. Accordingly, the image needs to be segmented into blocks .

In order to perform the DCT on the image, the Y, the Cb and the Cr planes should be divided into K of $n \times n$ (8 × 8, 16×16 or 32×32) blocks. Each block is DCT-transformed, quantized and thresholded.

5.1.3) Thresholding & Quantization

This algorithm applies the bisection method to repeat an iterative process to automatically select the threshold. After setting the goal of peak signal-to-noise ratio (GPSNR), the thresholding is given by the following steps:

- Step 1: Set a GPSNR and a boundary ε .
- Step 2: Determine the threshold selection range $[T_{max}, T_{min}]$ where $T_{min} = 0$ and T_{max} is the maximum value of DCT coefficients in a block.
- Step 3: Thresholding, quantization, inverse quantization, inverse DCT, and mean recovering are successively performed for DCT coefficients to obtain reconstructed image, where the Threshold
  $T = (T_{max} + T_{min})/2$.
- Step 4: Calculate PSNR , if **PSNR > GPSNR** , then let $T_{min} = T$; otherwise, let $T_{max} = T$.
- Step 5: Repeat Steps 3 and 4, until **abs(PSNR – GPSNR) < ε** is satisfied.

DCT coefficients with the absolute value less than the threshold T are discarded, and then the quantization is carried out. Here [.] represents rounding to nearest integer.

$$C_{DCT}^{NZQ} = \left\lfloor 1 + \frac{\mathrm{C}_{\mathrm{DCT}}^{\mathrm{NZ}} - [\mathrm{C}_{\mathrm{DCT}}^{\mathrm{NZ}}]_{\mathrm{min}}}{[C_{DCT}^{NZ}]_{max} - [C_{DCT}^{NZ}]_{min}} \right\rfloor$$

C^{NZ}_{DCT} =non zero DCT value

Min C^{NZ}_{DCT} = min value of DCT in an image

Max C^{NZ}_{DCT} = max value of DCT in an image

Q is the number of bits used for quantization

This formula is only used in non-zero values of DCT matrix.

5.1.4) Modified Lossless Encoder

The proposed approach is a simple and efficient encoding method. It combines the adaptive scan technique and the run length encoder (RLE). After the thresholding process and the quantization step, the proposed lossless encoder encodes the resulted DCT coefficients. Firstly, the DCT coefficients block is ordered and transformed in a vector following a scan. Several scan orders can be found in the literature such as zigzag and hilbert. In this study, four scans are used, namely zigzag, horizontal, vertical and hilbert as an adaptive block scanning. To encode efficiently the DCT coefficients vector, for each scan, the proposed encoder generates two vectors. The first vector is the nonzero DCT block coefficients (NZ) vector. The second is an index vector called $I\,DX_i$ ($i = 1 \cdots 4$) that contains the numbers of zeros that precede a NZ DCT coefficient. A scan is characterized by its $I\,DX_i$ vector, which is represented by its maximum value. Since that each scan will be coded according to a number of bits, which depends of maximum value of its $I\,DX_i$ vector, the retained scan is the scan that has the minimum value among the four $I\,DX_i$ vectors. This retained scan uses certainly the smallest encoder to encode the run of zeros sequence that precedes a nonzero (NZ) DCT coefficient. For this, a field, called *AS* of two (2) bits ("00," "01," "10" or "11"), is used to represent the selected scan order. As an example, if the zigzag scan is retained, *AS* is affected by the value "00." As depicted in Fig. 2, the encoded image bitstream is composed of the *Global Header* defining the control bits, and two other vectors including the NZ coefficients and the correspondent IDX_i vectors.

Each DCT block is encoded independent of the other blocks. For this, the *Global Header* is required to control the coding and the decoding process. This *Global Header* consists of three parts:

– The first one is the vector that contains the sizes L_i of the NZ coefficients of each block. The parameter L_i is coded with P bits.

– The second part is the vector that refers to the best scan AS_i of each block. To encode the four scans of the adaptive block scanning, two (2) bits are used.

– The last part is the vector that contains the parameter q_i coded by P bits. This parameter corresponds to the number of bits necessary to code the IDX_i vector of each block.

The parameters P and q_i are calculated as follows:

$$P = \log2(n \times n)$$

$$q_i = [\ \log2(\max(IDX_i))\ ]$$

where [.] represents the smallest integer greater than or equal.

So each N x N block is represented by :

Fig : Representation of one block encoded data.

	L	AS	q	NZ	IDX
SIZE:	**P bits**	**2 bits**	**P bits**	**(Q x L) bits**	**(q x L) bits**

Whole image is represented by :

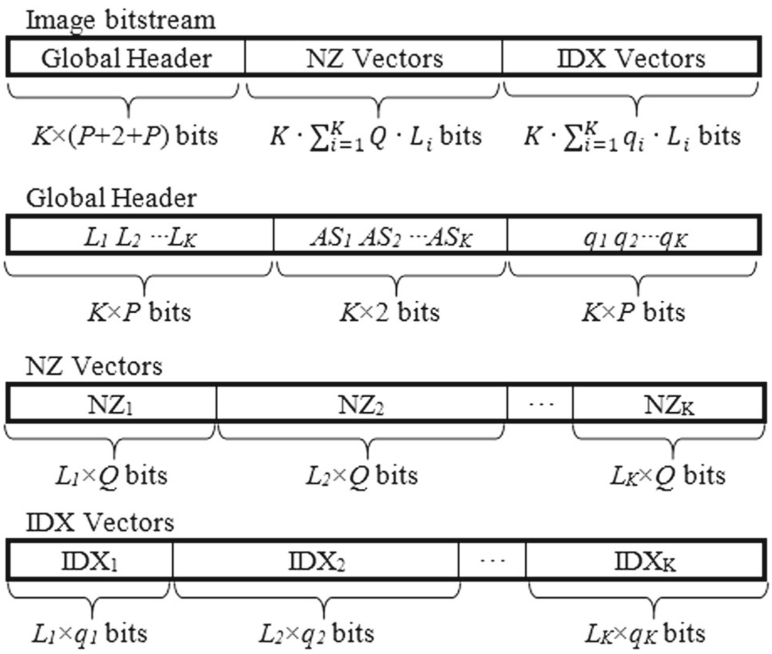

Fig: The image bitstream format of

5.2 Illustrative Example

In order to clarify the idea, for example, the 8×8 intensities block *B* depicted from Lena 256 gray levels image of size 512 × 512.

$$B = \begin{matrix} 163 & 157 & 159 & 160 & 159 & 154 & 156 & 157 \\ 161 & 155 & 155 & 157 & 157 & 157 & 157 & 152 \\ 159 & 159 & 157 & 157 & 159 & 154 & 158 & 154 \\ 160 & 159 & 158 & 158 & 156 & 155 & 157 & 158 \\ 161 & 158 & 160 & 157 & 156 & 153 & 156 & 171 \\ 159 & 155 & 154 & 158 & 156 & 155 & 154 & 160 \\ 157 & 156 & 151 & 160 & 157 & 156 & 156 & 155 \\ 158 & 158 & 158 & 159 & 155 & 156 & 154 & 171 \end{matrix}$$

Fig: 8x8 block from lena gray level image

Its rounded DCT transform is B_{DCT} :

$$B_{DCT} = \begin{matrix} 1259 & 3 & 7 & -3 & 10 & 0 & 0 & -1 \\ 0 & 7 & -4 & 6 & -4 & 4 & -2 & 5 \\ 0 & 0 & -3 & 2 & 1 & 1 & 0 & -2 \\ -1 & 0 & 0 & 0 & 4 & 1 & 1 & 0 \\ 8 & -2 & 6 & -9 & 3 & -3 & 4 & 0 \\ -2 & 3 & -5 & 4 & -2 & 2 & -5 & 0 \\ 2 & 0 & 1 & -3 & 1 & -2 & 1 & 1 \\ 2 & -1 & 1 & -3 & 1 & -1 & -1 & 2 \end{matrix}$$

Fig: 8x8 DCT block from lena gray level image

After this DCT block is thresholded using bisection method and then quantize. This block is thresholded with a threshold *Thr* = 3 and quantized with a *Q* = 7 bits uniform quantizer. The resulted block is given by:

$$C = \begin{matrix} 97 & 0 & 27 & 0 & 28 & 0 & 0 & 0 \\ 0 & 27 & 27 & 27 & 27 & 27 & 0 & 27 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 \\ 27 & 0 & 27 & 26 & 0 & 0 & 27 & 0 \\ 0 & 0 & 27 & 27 & 0 & 0 & 27 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Fig: 8x8 quantized and thresholded block

Now four type of scans are done in this block and the best scan is noted.

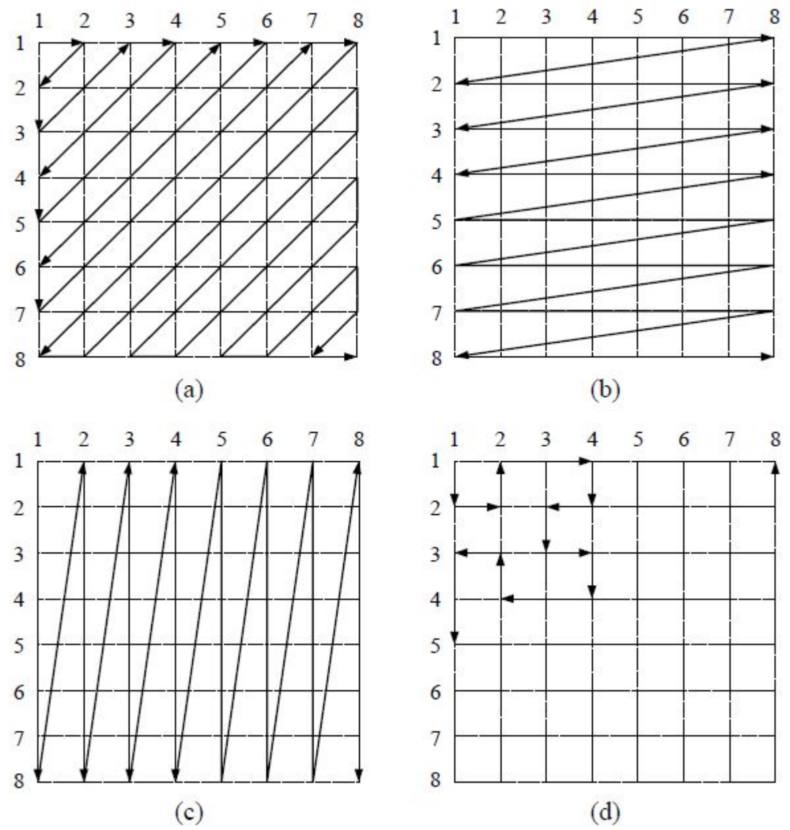

Fig : This diagram shows four types of scanning a) zig-zag scan b) horizontal scan c) vertical scan d) Hilbert scan

The different generated scans are:

1. **Zigzag scan** (max$(IDX1)$ = 9):

0 0 0 27 27 0 27 0 0 27 0 0 27 28 0 27 0 0 0 0 0 27 0 0 27 0 0 0 27 26 27 0 0 0 0 27 0 0 0 27 0 0 0 0 0 0 0 0 27 0 0 27

NZ: 27 27 27 27 27 28 27 27 27 27 26 27 27 27 27 27

IDX_1 : 3 0 1 2 2 0 1 6 2 4 0 0 4 3 **9** 2

2. **Horizontal scan** (max$(IDX2)$ = 12):

0 27 0 28 0 0 0 27 27 27 27 27 0 27 0 0 0 0 0 0 0 0 0 0 0 27 0 0 0 27 0 27 26 0 0 27 0 0 0 27 27 0 0 27

NZ: 27 28 27 27 27 27 27 27 27 27 27 26 27 27 27 27

IDX_2 : 1 1 4 0 0 0 0 1 **12** 3 1 0 2 3 0 2

3. **Vertical scan** (max$(IDX3)$ = 10):

0 0 0 27 0 0 0 27 0 0 0 0 0 0 27 27 0 0 27 27 0 0 0 27 0 0 26 27 0 0 28 27 0 27 0 0 0 0 0 27 0 0 0 0 0 0 0 0 0 0 27 27 0 0 0 27

NZ: 27 27 27 27 27 27 27 26 27 28 27 27 27 27 27 27

IDX_3 : 3 4 6 0 2 0 3 2 0 2 0 1 5 **10** 0 3

4. **Hilbert scan** (max($IDX4$ = 13):

0 27 0 27 0 27 27 0 0 0 0 0 0 0 27 0 0 0 0 0 0 0 0 0 0 27 27 27 26 0 0 0 0 0 0 0 0 0 0 0 0 0 27 27 0 0 0 0 0 27 0 0 27 27 28 0 0 0 27

NZ: 27 27 27 27 27 27 27 27 26 27 27 27 27 27 28 27

IDX_4 : 1 1 1 0 8 11 0 0 0 **13** 0 6 2 0 0 3

We note that, in these scans, the DC coefficient is not considered. For this block, the best scan is the zigzag scan because the smallest value of the maximums (bolded values) of the four index vectors IDX_i is 9. We have also:

– $Q = 7$,

– $P = log2\,(8 \times 8) = 6$ bits.

For the parameter q and L (P bits):

– $q1 = [\ log2\,(9)\ ] = 4$ bits coded with P bits as "000100,"

– $L1 = 16$ coded with P bits as "010000."

The number of bits needed to encode this block is:

$$\boldsymbol{m = P + 2 + P + (Q \times L1) + (q1 \times L1) = 190 \text{ bits.}}$$

The *Global Header* starts with the sequence: "010000 00 000100."

Here we are not considering DC coefficient , they are encoded by differential DC coding.

If we compare with other methods like :

1. RLE (Run Length Encoder):

0 1 27 1 0 1 28 1 0 4 27 5 0 1 27 1 0 12 27 1 0 3 27 1 0 1 27 1 26 1 0 2 27 1 0 3 27 2 0 2 27 1 (using raster scan)

The number of bits needed is: **7 × 42 = 294 bits.**

2. CDABS (paper by Douak):

155 1 155 1 155 155 8 155 11 155 155 155 154 13 155 155 6 155 2 155 155 156 3 155

The number of bits needed is: **15*(block Header)* + (7 + 1) × 24 = 207 bits.**

3. dLUT (differential Lookup Table):

NN: 27 27 27 27 27 28 27 27 27 27 26 27 27 27 27 27

dLUT : 4 1 2 3 3 1 2 7 3 5 1 1 5 4 10 3

The number of bits needed is: **14*(block Header)* + 7 × 16 + 4 × 16 = 190 bits.**

where NN is the non-null DCT coefficients and dLUT is the differential lookup table.

RESULTS

Results is been taken out on different images with different quantization width Q and different size of blocks (8x8, 16x16, 32x32). Quality of image was judged by PSNR value and compressed data was judged by bpp (bits per pixel).

Q	NxN	PSNR	bpp
7	16	31.9966	1.1454
8	16	32.6961	1.2354
9	16	32.9473	1.3254

Table 2: Shows the results of Lena Image for different quantization width (Q) & same 16x16 block size.

Q	NxN	PSNR	bpp
8	8	33.0072	1.2886
8	16	32.6961	1.2354
8	32	31.8225	1.3022

Table 3: Shows the results of Lena Image for same quantization width & different block size.

SUMMARY AND CONCLUSION

The project entitled **A study on different Image Compression algorithm to develop an Efficient Lossless Encoder** was implemented on MATLAB 2019a. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to compress image in lossless manner.

This project helped me in gaining valuable information and practical knowledge on several topics like Data compression, Encoder, MATLAB, JPEG Algorithm ,Bisection based Quantization etc . The entire system is secured. Also the project helped me in understanding the development phases of a project and software development life cycle. I learned how to test different features of a project.

This project has given me great satisfaction in having designed an application which can be implemented into products by simple modifications. There is a scope for further development in our project to a great extend as discussed in future scope.

FUTURE SCOPE

- In future we are working now on the enhancement of the obtained decompressed images quality by the application (as a post processing step) of the block-artifacts removing algorithms.
- Also we are trying to increase more our Encoding Algorithm part so that it can compressed more in smaller bits losslessly.

REFERENCES

1. https://www.ece.ucdavis.edu/cerl/reliablejpeg/compression/

2. F. Douak, R. Benzid, and N. Benoudjit, “Color image compression algorithm based on the DCT transform combined to an adaptive block scanning,” AEU - International Journal of Electronics and and Communications, vol. 65, no. 1, pp. 16–26, Jan. 2011.

3. https://en.wikipedia.org/wiki/JPEG

4. G. K. Wallace, “The JPEG still picture compression standard,” IEEE

5. Benzid R, Marir F, Bouguechal N. Electrocardiogram compression method based on the adaptive wavelet coefficients quantization combined to a modified two-role encoder. IEEE Signal Processing Letters 2007

6. Yang C, Lin J, Tsai W. Color image compression by moment-preserving and block truncation coding techniques. IEEE Transactions on Communications 1997

.