Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

ISSN No. 0976-5697

The Role and Challenges of Compression in Medical Image Communication

Dr. V. K. Bairagi
Associate Professor, Department of E&TC,
AISSMS's Institute of Information Technology,
Pune, India

Abstract: Information represented in a digitized image format is very easy to handle. These image files contain massive amounts of information, which requires efficient storage and transfer methods. There is requirement of investigation of quality issues, transfer methods, and storage mechanisms for such large size of these image files. Image compression requires less storage space for large image files. The transfer time is reduced for compressed files, while moving over networks. The aim of image compression techniques is to reduce the amount of data needed to accurately represent an image, such that this image can be economically transmitted or archived. In the field of medical imaging the use of computers is growing. Every day, a huge amount of data is produced from different medical imaging devices. Storage and transmission of this data becomes a problem, where bandwidth constraints are a major issue. This paper discusses the different types of image compression algorithm and the importance of image compression in medical communication. Some of the basic compression algorithms are used on test data and their performance is tested for wired communication.

Key words: Image, Compression, RLE, Communication, Lossless, Entropy.

1 INTRODUCTION

Compression is the art and science of representing information in a compact form. The internet has resulted in a network by which almost any form of wired or wireless communication is possible where compression is involved. The average net user communicates through text, images and video. Out of these, Text is the simplest and easiest, as it is quick and simple to use. Sending text requires very little bandwidth [1]. The problem with images and videos is that they require a large amount of bandwidth to send and receive data. Basically, still image data is a collection of 2-D arrays (one for each color plane) of values, representing intensity of the point in corresponding spatial location (pixel) [6].

To make this fact clear, let's see an example. An image, 512 pixel x 512 pixel x 24 bit, without compression, would require 0.75 MB of storage and 1.38 minutes for transmission, utilizing a high speed, 64 Kbit/s, ISDN line. If the image is compressed at a 10:1 compression ratio, the storage requirement is reduced to 0.075 MB and the transmission time drops to under 10 seconds [11][19][20]. Therefore, there is a need to decrease the size of the image that is to be sent or received. Increasing the bandwidth is another solution, but the cost sometimes makes this a less attractive solution. Also, as communication spectrum is limited, if bandwidth is increased then the number of channels gets reduced. By shrinking the size of the image, fewer pixels need to be stored and consequently, the file will take less time to load. The problem with this is that, if an image file is reduced then quality of the image is also reduced for lossy compression. Thus, two types of image compression exist: "Lossless" and "Lossy" compression [8]. This section provides a review on the importance of image compression methods and general issues which should be considered by the creators and managers of electronic records when considering image compression.

Computer graphics applications, particularly those generating digital photographs and other complex color images, can generate very large file sizes. Sources of images includes Image scanner, Digital camera, Video camera, Ultra-sound (US), Computer Tomography (CT), Magnetic resonance image (MRI), digital X-ray (XR), infrared image generators, etc. Issues of storage space and the requirement to rapidly transmit image data across networks and over the Internet have led to the development of a range of image compression techniques, to reduce the physical size of files [1] [2].

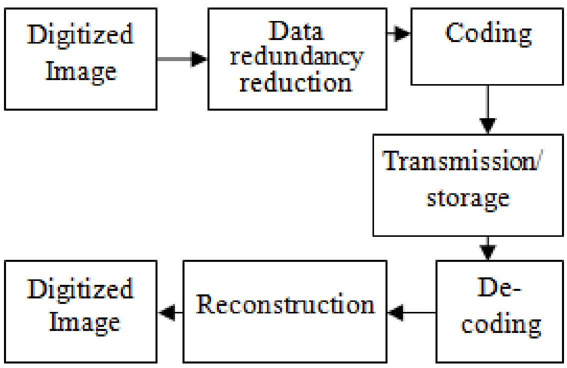

Fig :1 A generalized block diagram of any image compression technique

A generalized block diagram of any communication system where compression is used is shown in fig 1. The data having very less meaning or the data by removing which the quality of image will not be affected much is removed from the image. Then, the compression coding is applied over the image. Different algorithm such as run length coding (RLE), Lempel Ziv (LZ) Compressors, Huffman Encoding, DCT, CCITT Group 3 & 4, JPEG, JPEG 2000 are widely used for compression [8]. Even Microsoft-Windows is using compression software. Such compressed image data is then transmitted over the channel. At the receiving station, exactly the reverse procedure as that of the transmitting station is carried out. First, the decoding algorithm is used and the received data is then reconstructed to form the original image[3].

a) Image file format

Different file formats are available like BMP, TIFF, and PNG etc, which are widely used in the field of image processing are shown in table 1. Each of these has special properties. Medical Imaging uses a special kind of file format to represent data, called DICOM file [4].

Table 1: Available file format

BMP (Windows Bitmap Format)	**GIF** (Graphics Interchange Formats)	**PNG** (Portable Network Graphics)	**TIFF** (Tagged Image File Format)	**DICOM** (Digital Imaging & Communications in Medicine)
Lacks features of others.	Quality factor is not good.	Supports large number of colors.	Capabilities could be added by the use of “tags”.	Carries the additional information than the TIFF.

b) DICOM Standard

Digital Imaging and Communications in Medicine (DICOM) is specific file format developed for medical image and their communication. It contains both Image data as well as patient information such as reports. DICOM format contains the stack of images along with header carrying some text information as shown in figure 2 [15].

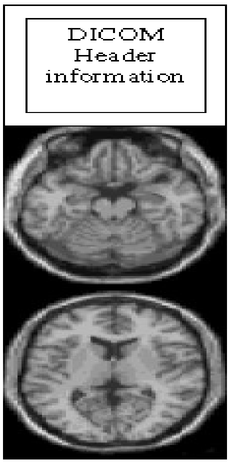

Fig2 : Typical DICOM file.

During DICOM image compression, there is also a need to carefully consider header information; hence hybrid combination of image compression, with text compression is required [16].

c) Image files format comparison

A comparison of some well known file format is shown in the table 2. Different images have been taken for test which includes monochrome, color, medical, nature images. The test database is shown in figure 3.

Fig 3: Test Images data base

BMP files are taken as reference and from it, various other files have been derived. The zip algorithm, which is freely available with windows, is applied on images. It is seen that file size gets reduced after compression. It is also seen, that percentage of compression is varying from image to image.

Table 2. Compression performance of file formats (in KBs)

Sr	File name	Entropy	.bmp	.tiff	DICOM	.zip
1	Ball	2.68	691	292	692	53
2	Cameraman	5.35	305	100	304	64
3	Flowers	7.41	70	71	70	54
4	Fruits	7.37	193	194	193	152
5	Graph	0.09	769	21	768	6
6	Medical	6.49	41	36	40	30
7	Man	7.08	769	718	768	368
8	Nature	7.44	5,626	5,680	5,626	1,908
9	Plane color	0	30	2	230	1

Definitely, due to compression, size of the image file gets reduced. This will reduce the storage and transmission requirement over the channel. For complete white plain image, the ZIP algorithm gives highest compression as compared to the others. This means the compression efficiency of the algorithm is data dependent.

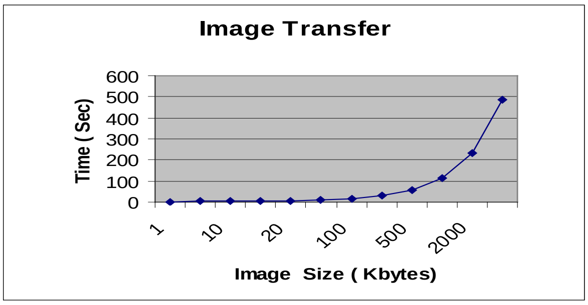

Fig 4: PC to PC communication,

There are two types of communication: wired and wireless. Compression is recommended for both types. The time required to transfer the image using PC to PC communication is shown in Fig 4. It is very clear that if we have smaller image size, definitely, transmission time gets reduced. One important observation is that, the graph of transmission time versus image file size is nonlinear. This is due to correlation-more number of bits is required for large images.

2. IMAGE COMPRESSION CONSIDERATIONS

There are number of factors to be considered when using compression algorithms. Most algorithms are particularly suited for specific circumstances, which must be understood if they are to be used effectively. For example, some are most efficient at compressing monochrome images, whilst others yield best results with complex color images [1],[8].

a) There are two main categories of Image compression algorithms:

- **Lossy compression** : There is loss in image quality, by removing some image information.
- **Lossless compression**: There is no loss in Image quality. They preserve all of the original image information while compression, and therefore do not degrade the quality of the image.

While achieving a higher compression ratio, one must note that the quality of the image is within the accepted limits [13]. The units "bits/pixel" here, mean average number of bits required to represent the compressed image. The compression ratio is defined as the Compression ratio = size of the output stream / size of the input stream.

A value of 0.5 means that, data occupies 50% of its original size after compression. Values greater than 1 mean that the output stream is bigger than the input stream (negative compression). The compression ratio should therefore, be as low as possible. The inverse of compression ratio is called *compression factor* [4]. The image after reconstruction should be visually & diagnostically accepted to readers and doctors.

b) **Redundancy** is the portion of data that can be removed when it is not needed or can be re-inserted to interpret the data when needed. It may be sub-classified into Special redundancy and temporal redundancy. Another type is perceptional redundancy but this is lossy type of compression [8].

c) **Entropy** is the basic measure of information. It is a measure of the average number of binary symbols needed to code the output of the source [6] [13], Entropy (H) can be given by following formula:

$$H = \sum_{i=1}^{m} p_i \log_2 p_i \text{ bits / source symbol}$$

{ *pi*= probability of occurrence of the symbol

Information is zero when p_i =1 and inversely, is arbitrarily large (∞) when $p_i \rightarrow 0$. This data is useful while deciding the optimal lossless compression factor, because entropy is related to redundancy R, in the image.

$$R = \sum_i P_i l_i - \sum_i [-P_i \log_2 P_i].$$

d) **Complexity** of the algorithm is one of the major things to be considered while choosing the algorithm as it is related with the time to encode & decode the image data. For online applications, the complexity of the algorithm should be sufficiently less so as to cope up with the transmission speed as it is related to the access time [6], [5]. It should be as less as possible and easy to implement. The complexity can be measured in terms of Big-O notations.

e) Compressing images allows users to send them using progressive transmission. Using conventional "sequential"

transmission, an image cannot be shown until complete transmission ends.
Now, we will discuss the some of the basic and widely used compression techniques.

3. LOSSLESS CODERS

Compression takes an input X and generates a representation Xc that hopefully requires fewer bits. There is a reconstruction algorithm that operates on the compressed representation Xc, to generate the reconstructed output Y. Based on the requirements of reconstruction, data compression schemes can be divided into two broad classes. One is lossless compression, in which Y is identical to X. Examples of lossless methods are Run Length coding (RLE), Huffman coding, Lempel/Ziv (LZ) algorithms, and Arithmetic coding. The other is lossy compression, which generally provides much more compression than lossless compression but allows Y to be different from X. There are several different ways in which image files can be compressed. For Internet use, the two most common compressed graphic image formats are the JPEG format and the GIF format. The JPEG method is more often used for photographs, while the GIF method is commonly used for line art and other images in which geometric shapes are relatively simple [17][18].
Other techniques for image compression include the use of fractals and wavelets. Both methods offer promise because they offer higher compression ratios than the JPEG or GIF methods for some types of images.
If data is losslessly compressed, the original data can be recovered exactly, from the compressed data. It is generally used for applications that cannot allow any difference between the original and reconstructed data.

1. Run Length Encoding:-

Run Length encoding, is the simplest data compression algorithm. It is more effective when the data sets is having single repeated character. Instate of storing same character for more times, the RLE stores its character and a number representing "How many times the character is repeated", For instance, the string "PPPPPRZZZZOOOOO" could be more efficiently represented as "*P6*Z4*O5".

2. Huffman Coding

Huffman coding, developed by D.A. Huffman [5] is a classical data compression technique. It uses variable length coding. The codes for characters having a higher frequency of occurrence are shorter than those codes for characters having lower frequency. The character who's probability of occurrence is more , will be assigned short code and the character who's probability of occurrence is less , will be assigned long code

4. Limpel-Ziv-Welch(Lzw) Encoding

This original approach is given by J.Ziv and A. Lempel in 1977. T.Welch's refinements to the algorithm were published in 1984 [7]. LZW compression replaces strings of characters with single codes as shown. Compression occurs when a single code is output instead of a string of characters [5].

LZW works best for files containing lots of repetitive data e.g. text and monochrome images. But the files that are compressed, which do not contain any repetitive information at all, can even grow bigger!

5 Arithmetic Coding

Arithmetic coding is also a kind of statistical coding algorithm similar to Huffman coding. However, it uses a different approach to utilize symbol probabilities and performs better than Huffman coding. When the symbol probabilities are more arbitrary, arithmetic coding has a better compression ratio than Huffman coding [9].

4. LOSSY CODERS

Lossy codes are mostly transformed based coding, which uses a quantizer as one of the building blocks. Lossy coders mainly include Discrete Cosine Transform (DCT) or Wavelet based compressions, like Joint Photographic Expert Group (JPEG), JPEG2000, Embedded Zero Tree (EZW) and Set partitioning in Hierarchical Trees (SPIHT) [13]. A general transform coding scheme involves subdividing an NxN image into smaller nxn blocks and performing a unitary transform on each sub-image. The transform is a technique for converting a signal into elementary frequency components which are de-correlated from each other. Transform coding can be generalized into four stages: 1. Image Subdivision, 2. Image Transformation, 3. Coefficient Quantization, 4. Entropy coder [10].
Also, visually lossless compression can often be achieved by incorporating the Human Visual System (HVS) contrast sensitivity function in the quantization of the coefficients [6]. For a transform coding scheme, logical modeling is done in two steps: segmentation, in which the image is subdivided into bi-dimensional vectors (possibly of different sizes) and a transformation step, in which the chosen transform (e.g. DCT, Hadamard) is applied. Quantization can be performed in several ways [8].
Using lossy codes, one may achieve compression ratios upto 0.01 [6]. The compression ratio depends on, how much loss in image data is accepted to the user. This loss or distortion in the reconstructed image can be measured by some quality measures like MSE, PSNR, Correlation, NAE, MAE etc [13][14]. The JPEG and MPEG standards are examples of standards based on transform coding.

5. TEST AND OBSERVATIONS FOR BASIC LOSSLESS ALGORITHMS

I) In the medical imaging field, loss of even minute data would be a major issue if the loss is related to diagnostics [12]. Hence, only lossless algorithms are taken for consideration. Simple and basic algorithms are implemented like RLE, Huffman, LZW and arithmetic comparison.

Hybrid method is used to achieve more efficiency in compression. Here, we define a variable 'count' which would store the values according to the changes in the image pixels (randomness). Depending on the count value, suitable algorithms can be used for compression of images for efficient compression. The threshold value is calculated considering different sizes of images. Thus, one can apply the most suitable compression algorithm to an image so as to get maximum compression. Since LZW does not give even less than 50% of compression as seen in following table 3, it is not considered for the hybrid approach for images. Table 3 contains the result of RLE, Huffman, LZW and arithmetic coding applied to complete images.

Table 3 Result as algorithms applied to images

SR. NO.	IMAGE (Size)	ACTUAL SIZE (IN KB)	COMPRESSED SIZE(KB) AND TIME(SEC) LZW		HUFFMAN		RLE		ARITHM-ETIC		OUR-APPROACH		COMPRESSION RATIOS LZW	HUFF-MAN	RLE	ARITHM-ETIC	COMBINED
			SIZE	TIME	SIZE	TIME	SIZE	TIME	SIZE	TIME	SIZE	TIME					
1	1 (200 x 200)	41	31.6	156	4.88	22.4	30.7	11.2	68.1	14.7	4.88	22.4	0.771	0.118	0.74	1.69	0.118
2	2 (200 x 200)	41	26.93	137.3	4.35	17.2	25.7	7.2	62.8	13.1	4.35	17.2	0.655	0.105	0.62	1.56	0.105
3	3 (200 x 200)	41	22.5	130.7	3.65	18.1	21	3.7	50.6	10.8	3.65	18.1	0.548	0.088	0.51	1.26	0.088
4	4 (200 x 200)	41	38	168.9	4.96	18.1	34.4	15.3	74.2	15	4.96	18.1	0.926	0.121	0.84	1.85	0.121
5	5 (200 x 200)	41	33.4	151.3	4.71	19.2	29.5	10.3	67.1	13.9	4.71	19.2	0.814	0.11	0.71	1.67	0.11
6	6 (200 x 200)	41	29.3	138.4	4.71	17.2	28.6	7.6	64.4	13.4	4.71	17.2	0.715	0.114	0.69	1.6	0.114
7	7 (200 x 200)	41	27.8	148.6	6.98	19.5	26.8	8.25	66.0	13.8	6.98	19.5	0.677	0.169	0.65	1.64	0.169
8	8 (200 x 200)	41	25.9	141	4.70	17	22.3	4.06	69.2	14.1	4.70	17	0.631	0.114	0.54	1.72	0.114
9	9 (200 x 200)	41	30.23	145	5.87	17.8	26.1	8.15	64.5	13.4	5.87	17.8	0.73	0.142	0.63	1.6	0.142
10	105 (100x100)	10.8	8.53	6.10	6.55	1.82	7.77	0.28	66.9	13.7	6.55	1.82	0.78	0.60	0.71	6.19	0.60
11	106 (100x100)	10.8	8.45	6.67	2.90	2.70	7.62	0.28	65.36	13.8	2.90	2.70	0.78	2.90	0.70	6.05	2.90
12	111 (512x512)	257	--	--	6.75	7.2	38.0	0.56	41.3	8.54	6.75	7.2	--	0.026	0.14	0.16	0.026
13	112 (512x512)	257	--	--	11.1	6.53	84.3	0.39	60.47	11.0	11.1	6.53	--	0.043	0.32	0.23	0.043
14	107 (1024x1024)	1Mb	--	--	90.2	21	323	0.98	66.12	12.8	66.12	12.8	--	0.088	0.31	0.064	0.064
15	109 (1024x1024)	1Mb	--	--	53.6	20	552	1.10	64.48	12.2	64.48	12.2	--	0.052	0.50	0.063	0.063

Images 105,106,111,112,107 and 109 are the resize version of image no 4,6,11,12,07,09.

II) Bit plane separation.

1) An 8 bit Medical Image is taken and the contribution from each bit-plane is observed. (Shown below in figure 5)

Algorithm:

1. Read the input image

2. Divide the Image in to 8 different planes depending on Bit position

3. Construct the image of each plane.

Major contribution in the image is due to the first three MSB bit planes, while the LSB bit plane does not contribute much in the image formation. Such bit planes are redundant information and may be removed for compression purposes. But this should be checked for every medical image.

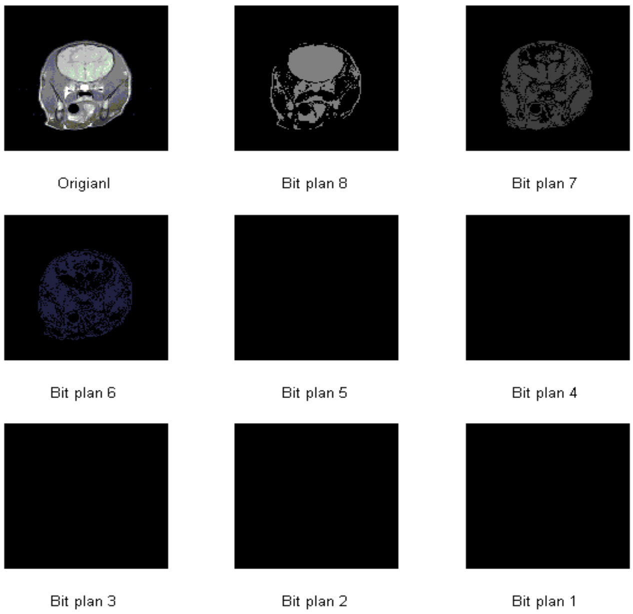

Fig 5: Bit plane separation (*Original and *bit plane…spelling in the images above)

III) Applying algorithms on Bit-Planes

The basic algorithms are applied to the bit planes of the image for observation. RLE, Huffman and LZ only are considered for bit planes.

Table 4 contains the result of RLE, Huffman coding and LZW coding applied to bit planes of images. Both result tables give us the idea about the compressed size of image, time taken by algorithm to achieve compression and also the compression ratio for each image and plane. Result table 4 also tells about the randomness in white and black pixels of the image. It is calculated by incremental counter if there is any change in pixel value as compared to previous one. Its normalized value is taken and then percentage is calculated.

Depending on this result table, we develop the hybrid algorithm which effectively applies the compression algorithm to images so as to achieve efficient compression. Figure 6 shows the images database, while figure 7 shows the developed software.

Table 4 Result as algorithms applied to bit-planes of images

SR NO.	IMAGE	PLANE	RANDOM-NESS	COMPRESSED SIZE				HYBRID IMAGE SIZE	HYBRID CR
				HUFFMAN	RLE	LZW	HYBRID		
1	1.bmp 40.1KB	1	44.0225	10.0kb	6.3kb	5.13kb	5.13kb	26.34kb	0.658
		2	43.7475	10.0kb	6.28kb	5.17kb	5.17kb		
		3	42.8275	10.0kb	6.23kb	4.67kb	4.67kb		
		4	38.2975	9.77kb	5.79kb	4.67kb	4.67kb		
		5	26.9475	8.25kb	5.54kb	2.26kb	2.26kb		
		6	18.1075	7.32kb	4.29kb	1,77kb	1,77kb		
		7	11.9225	6.50kb	3.41kb	1.54kb	1.54kb		
		8	5.8475	5.21kb	2.16kb	1.13kb	1.13kb		
2	2.bmp 40.1KB	1	38.7875	9.15kb	6.81kb	6.12kb	6.12kb	25.35	0.632
		2	37.0275	9.12kb	6.65kb	5.86kb	5.86kb		
		3	34.1325	9.00kb	6.44kb	5.67kb	5.67kb		
		4	27.2275	8.16kb	5.48kb	2.68kb	2.68kb		
		5	19.7075	7,38kb	4.42kb	1.94kb	1.94kb		
		6	14.1475	6.55kb	3.59kb	1.26kb	1.26kb		
		7	7.4375	5.40kb	2.29kb	963bytes	963bytes		
		8	4.1275	4.57kb	1.49kb	858bytes	858bytes		
3	3.bmp 40.1KB	1	25.5775	7.69kb	3.93KB	3.02kb	3.02kb	20.82	0.519
		2	25.2725	7.65kb	3.86KB	2,97kb	2,97kb		
		3	24.8075	7.60kb	3.82KB	2,92kb	2,92kb		
		4	24.3575	7.53kb	3.72KB	2.89kb	2.89kb		
		5	23.0425	7.37kb	3.60KB	2.79kb	2.79kb		
		6	20.8925	7.12kb	3.37KB	2.53kb	2.53kb		
		7	16.7975	6.70kb	2.94KB	2.16kb	2.16kb		
		8	7.7425	5.56kb	1.88KB	1.54kb	1.54kb		
4	5.bmp 40.1KB	1	42.265	9.75kb	6.03kb	5.85kb	5.85kb	36.25	0.904
		2	42.21	9.69kb	6.03kb	5.87kb	5.87kb		
		3	41.9775	9.7kb	5.98kb	5.78kb	5.78kb		
		4	41.75	9.67kb	5.99kb	5.75kb	5.75kb		
		5	36.22	9.60kb	5.73kb	5.79kb	5.79kb		
		6	20.5425	8.43kb	5.46kb	4.01kb	4.01kb		
		7	12.305	7.19kb	3.13kb	2.38kb	2.38kb		
		8	2.6275	4.47kb	1.38kb	822bytes	822bytes		

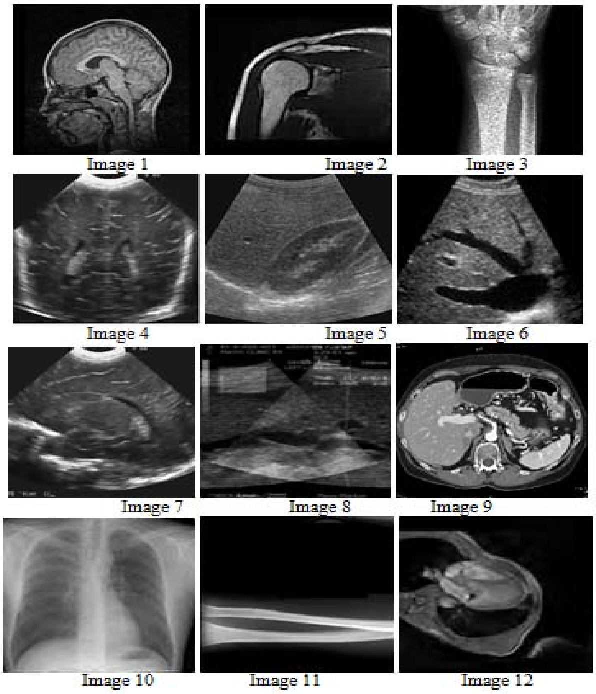

Fig 6: Image database

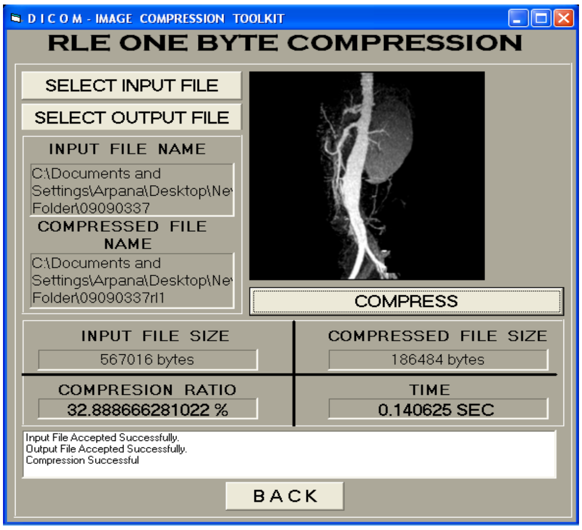

Fig 7: Developed software

6. DISCUSSION AND CONCLUSIONS

Thus, from result table 3 it is concluded that, compression using Huffman coding algorithm is more than RLE for the images. From the result table in Table 4, as applied to bit planes of images, it can be concluded that Huffman coding does not give good results as compared to RLE and LZW coding. Also, it is seen that as randomness of planes decreases, Huffman coding gives better results, as compared to most randomly changing pixel planes. Thus, it is concluded that Huffman coding is applicable to those images that contain less variation in pixel values. It is also seen that LZW coding gives good results for bit plane images rather than being applied on a complete image, but it is not applicable for very large sized images, as it is very time-consuming. Also, arithmetic coding is efficient for images with size greater than 512x512.

Medical images must be compressed using lossless compression algorithms only, which must be fast in encoding as well as in decoding to support real-time applications such as telemedicine. It is recommended that, algorithms should only be used in the circumstances for which they are most efficient. It is also strongly recommended that archival master versions of images should only be created and stored using lossless algorithms in the case of medical imaging. Achieving a compression ratio such that the bit rate equals entropy of the image data will be the future research direction.

7. REFERENCES

[1] Adrian Brown, “Digital Archives Analyst”, Digital Preservation Guidance Note -05, the national Archival, Issue: 1, 2003, pp 1-9.
[2] C.E. Shannon, W. Weaver, “The Mathematical Theory of Communication”, University of Illinois Press, Urbana, 1949, pp 1-4.
[3] Cecile Delgorge, “JPEG 2000, an adapted compression method for ultrasound images"? A comparative study”. 2001, pp 1-4.
[4] D. Solomon, G. Motta, “Handbook of Data Compression”, 5th edition, Springer Publication, 2010, pp 1-51.
[5] Gerald Schaefer, Roman Starosolski and Shao Ying Zhu, “An evaluation of lossless compression algorithms for medical infrared Images”, IEEE conference proceeding EMB 27th Conference, 2005, pp 1673-1678.
[6] Giuseppe Placidi , “Adaptive compression algorithm from projections: Application on medical greyscale images”, Journal of Computers in Biology and Medicine 39, Elsevier Publication, 2009, pp 993 – 999.
[7] Guy E. Blelloch, Introduction to Data Compression, Carnegie Mellon University, October 16. 2001, page 1-10.
[8] I. Baeza, A. Verdoy , “ROI-based procedures for progressive transmission of digital images: A comparison”, Journal of Mathematical & Computer Modelling, Elsevie Publication, , 2009, pp 849_859.
[9] K. Sayood, “*Introduction to Data Compression*”, 3rd edition, Elsevier Publication, 2010, pp 473-512.
[10] V K Bairagi, A M Sapkal, “Automated Region Based Hybrid Compression for DICOM MRI Images for Telemedicine Applications”, The IET Science, Measurement & Technology**,** Vol 6, No 4, July 2012, pp. 247 – 253
[11] Min Liu, “Digital Image and Video Compression Techniques”, EDC385G: Multimedia Authoring, 2003, pp 1-4.
[12] V K Bairagi, “Symmetry Based Biomedical Image Compression”, Springer Journal of Digital Imaging. Vol 28, No 6, pp 718-726, Dec 2015.
[13] Shaou-Gang Miaou,Fu-Sheng Ke, and Shu-Ching Chen, “A Lossless Compression Method for Medical Image Sequences Using JPEG-LS and Interframe Coding”, IEEE Transactions On Information Technology In Biomedicine, Vol. 13, No. 5, 2009, pp 1-4.
[14] V K Bairagi, A M Sapkal, M S Gaikwad, “The Role of Transforms in Image Compression”, Springer Journal of Institute of Engineers (India), Series B., Vol 94, No 2, June 2013.Pp 135-140.
[15] V K Bairagi, A M Sapkal, “The role of DICOM technology in Telemedicine”, Invertis International Journal of science and Technology, 2011, Vol 4, No 2, pp 72-77.
[16] V. K. Bairagi, A. N. Gaikwad, M. Kanaskar, “DICOM medical data compression for telemedicine in rural areas”, *Journal of Advances in Engineering Science* Sect. C (2), 2008, PP 1-6
[17] A M Sapkal, V K Bairagi, “Telemedicine In India: Challenges and Role of Image Compression”, American Scientific Publishers: *Journal of* Medical Imaging And Health Informatics. Vol. 1, No 4, Dec 2011, pp300–306.

[18] P. Bhor, R. Gargote, R. Vhorkate, R.U.Yawle, V. K. Bairagi, "A No Reference Image Blur Detection Using Cumulative probability Blur Detection(CPBD) Metric", International Journal of Innovative Science and Modern Engineering, Vol 1, No 5, April 2013, pp 76-80.
[19] http://jan.ucc.nau.edu/lrm22/pixels2bytes/calculator.htm
[20] https://www.expedient.com/file-transfer-time-calculator/