ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

LOSSLESS METHOD OF IMAGE COMPRESSION USING HUFFMAN CODING TECHNIQUES

Trupti S Bobade*, Anushri S. sastikar
[1]Department of electronics and telecommunication, J.D.I.E.T.Yavatmal, India
[2]Department of electronics and telecommunication, J.D.I.E.T.Yavatmal, India

Abstract

Usage of Image has *been* increasing and used in many applications. The need for an efficient technique for compression of images ever increasing because the raw images need large amounts of disk space seems to be a big disadvantage during storage & transmission. Image compression plays on important role in saving storage space and saving time while sending images over the network without excessively reducing the quality of the picture. There are many compression techniques, which are presented for better use of storage devices. A compression technique explained to achieve more compression ratio by reducing the number of source symbols. In this paper, Lossless method of image compression using a simple coding technique called Huffman's coding is explained.

Keywords: Data compression, Redundancy, Huffman Coding

Introduction

Any digital image can be obtained by sampling and quantities. This continuous tone picture requires an enormous storage. For instance, a 24 bit color image with 512x512 pixels will require 768Kbyte storage on a disk, and a picture size becomes twice of this size will not fit in a single floppy disk. To transmit such an image through a network at 28.8 Kbps modem would take almost 4 minutes [1]. The purpose for image compression is to reduce the amount of data required for representing sampled digital images and therefore, for reduce the space and transmission time.

Image compression plays a vital role in many important applications, such as remote-sensing application, image database and image communication. The images to be compressed are gray scale having pixel values between 0 to 255. There are two different techniques for compressing images lossless and lossy compression techniques. In the lossless compression techniques, information with respect to the image is not lost the reconstructed image from the compressed image is an identical to the original image. In the lossy compression, lost the information of image, the reconstructed image from the compressed image is not a identical to the original image but similar to original image.

In the paper, a lossless compression technique called Huffman's coding explained. The Huffman's algorithm is generating minimum redundancy codes compared toothier algorithms. The Huffman coding has mostly used in text compression, video compression, image compression and conferencing system. In the Huffman coding, technique collects all unique symbols from the source image and calculates its probability value for each symbol and sorts the symbols based on its probability value [1]. Further, from the lowest probability value symbol to the highest probability value symbol, to form a binary tree the two symbols get combined at a time. In the coding technique, allocates

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

zero to the left node and one to the right node starting from the root of the tree. To calculate Huffman's code for a specific symbol, all one and zero collected from the root to that specific node in the same order. The aim of this technique is to compress images by reducing the number of bits per pixel required to represent it and to decrease the transmission time for transmission of images and then reconstructing back by decoding the Huffman codes.

In the paper, the need for the compression is explained in first section, various types of data redundancies are explained in section 2. In section-3, types of compressions method are explained, in section-4. In section 5, the implementation of lossless method of compression (i.e. Huffman Coding) is done. The algorithm is developed and in section-6, the results were presented with explanation. Lastly, advantages, disadvantages and paper conclude with References.

II. NEED OF COMPRESSION

Why we required the compression technique is first thing to study. Now days in the social media exchanging of image files, videos through the network done in the process of exchanging the data we required large space and our speed gets slow because large file required more time to transmit any files. Example, to store a color image of a moderate size like 512*512pixels, needs 0.75MB of disk space another example, for 30mm digital slide with a resolution of 12micro-meter required 18MB space to stored this files and make available to the network there we needed the compression techniques. Image compression addresses the problem of reducing the amount of data required to represent any digital image. The main basic of the reduction process is the removal of redundant data. When image is receive at receiver side the compress image get decompress and reconstructed into its original image approximation to the original image.

Below example show important of image compression 1024 pixel×1024 pixel×24 bit an image without compression, would require 3.5 MB of storage and 7 minutes for transmission, utilizing a high speed, 64Kbits/s and ISDN line. If the image is compressed at a 10:1compression ratio, the storage requirement is reduced to300 KB and the transmission time drop to less than 6seconds.

III. VARIOUS TYPES OF REDUNDANCY

When data sets contain some type of redundancy that time reduction is possible. Compression of digital image is done by for reducing the total number of bits required to represent an image. Eliminating various types of redundancy exist for the pixel value. There are three types of data redundancies they are given below:

Coding Redundancy: Each pixel of the uncompressed image is coded by a fixed length. By using the variable length code schemes like Huffman's coding and arithmetic coding may produce the compression.

Psycho-visual Redundancy: It is a redundancy proportional to different sensitivities to all images signals from human eyes. By eliminating some less relative important information, which is not sensitive to human eyes.

Inter-pixel Redundancy: It is a redundancy which depends upon to statistical dependencies among the pixels, between neighboring pixels.

Coding Redundancy:

Consider a gray level's image having n pixel and L is the number of grey level within the image range from 0 to L-1 and the number of pixels with gray level r_k is $n_k$therefore the probability of occurring gray level *is*

r_k being $p_r(r_k)$. To represent the gray level r_k used no. of bits is l(r_k) ,then the average no. of bits required to represent each pixel is:

Lavg = $\sum_{k=0}^{l-1} l(r_k) p_r(r)$(1.2)

Where,

Pr(r_k)=n_k / n (1.4)

k= 0, 1, 2,……….., L-1

Hence the number of bits required to represent the whole image is n x Lavg. When lavg is minimized achieved the maximal ratio, i.e. when l (r_k), the Length of gray level representation function, leading to minimal Lavg is found. Coding the gray levels in such a way that the Lavg is not minimized resulting in an image containing code redundancy.

Generally, coding redundancy is presented when the codes (whose lengths are represented here by $l(r_k)$function) assigned to gray levels don't take full advantage of gray level's probability (Pr (r_k) function). Therefore, it almost always presents when an image's gray levels are represented with a straight or natural binary code [1]. A natural binary coding of their gray levels assigns the same number of bits to both the most and the least probable values, thus failing to minimize equation 1.2 and resulting in coding redundancy**.**

Example of Coding Redundancy: An 8-level image has the gray level distribution shown in table I. If natural 3-bit binary code see code 1, and lr_kin table I)is used to represent 8 possible gray levels, Lavg is 3- bits, because lr_k= 3 bits for all r_k. If code 2 in table I is used however, the average number of bits required to code the image is reduced to:

Lavg = 2(0.19) + 2(0.25) +2(0.21) + 3(0.16) + 4(0.08) + 5(0.06) + 6(0.03) + 6(0.02)

= 2.7 bits.

From equation of compression ratio (n2/n1) the resulting compression ratio *CR* is 3/2.7 or 1.11.Thus approximately 10% of the data resulting from the use of code 1 is redundant. The exact level of redundancy can be determined from equation (1.0).

r_k	$p_r(r_k)$	Code1	l1(r_k)	Code2	l2(r_k)
0	0.19	000	3	11	2
1/7	0.25	001	3	01	2
2/7	0.21	010	3	10	2
3/7	0.16	011	3	001	3
4/7	0.08	100	3	0001	4
5/7	0.06	101	3	00001	5
6/7	0.03	110	3	000001	6
1	0.02	111	3	000000	6

Table I: Example of Variable Length Coding

RD= 1-1/1.1 = 0.099=9.9%

Clearly 9.9% data in First Data set a redundant, which is to be removed to achieve compression.

Reduction of Coding Redundancy: To reduce this redundancy from an image we go through the Huffman technique where we are, assigning fewer bits to the more likely gray levels than to the less probable ones achieves

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

data compression. This process commonly is referred to as variable length coding. There are several favorable and near favorable techniques for constructs such as code like Huffman's coding, Arithmetic coding, etc.

Inter pixel Redundancy: Another important form of data redundancy inter pixel redundancy, which is directly related to the inter pixel correlations within the image. Because the value of any given pixel can be reasonable predicted from the value of its neighbors, the information carried by individual pixels is small. Much of the visual contribution of a single pixel in an image is redundant; it could have been the guest on the basis of its neighbor's values. A variety of names, including spatial redundancy, geometric redundancy, and interface redundancies have been coined to refer to these inter pixel dependencies. In order to reduce the inter pixel redundancies in an image, the 2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient but usually non-visual format.

For example, the differences between adjacent pixels can be used to represent an image. Transformations of this type are referred to as mappings. They are called reversible if the original image elements can be reconstructed from the transformed data set [1, 2].

Reduction of Inter pixel Redundancy:

To reduce the inter pixel redundancy; we use various techniques such as:

1. Run length coding.
2. Delta's compression.
3. Constant area coding.
4. Predictive coding.

Psycho visual Redundancy:

Human perception of the information about an image normally does not involve quantitative analysis of every pixel or luminance value in the image. In general, an observer searches for distinguishing features such as edges or textural regions and mentally combines them into recognizable groupings. The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process. Thus eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than information in normal visual processing. This information is said to be psycho visually redundant. It can be eliminated without significantly impairing the quality of image perception. Psycho visual redundancies are fundamentally different from the coding redundancy and inter pixel redundancy. Unlike coding redundancy and inter pixel redundancy, psycho visual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing. Since the elimination of psycho, visual redundant data results in a loss of quantitative information. Thus it is an irreversible process.

Reduction of Psycho visual Redundancy:

Original source		**Source reduction**			
S	**P**	**1**	**2**	**3**	**4**
a_1	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
a_2	**0.4**	**0.4**	**0.4**	**0.4**	**0.7**
a_3	**0.1**	**0.1**	**0.1**	**0.3**	
a_4	**0.1**	**0.1**	**0.2**		
a_5	**0.05**	**0.1**			
a_6	**0.05**				

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Table II: Huffman Source Reductions S-source, P-probability

To reduce psycho visual redundancies we use quantizes. Since the elimination of psycho, visually redundant data results in a loss of quantitative information. It is commonly referred to as quantization. As it is an irreversible operation (visual information is lost) quantization results in lossy data compression.

IV. TYPES OF COMPRESSION METHOD:

The image compressions are broadly classified under two categories depending whether or not an exact replica of the original image could be reconstructed using the compressed image.

These are:

1. Lossless compression
2. Lossy compression

Lossless compression technique:

In lossless compression techniques, the original image can be perfectly recovered from the compressed (encoded) image. These are also called noiseless since they do not add noise to the signal (image) [5]. It is as well known as entropy coding since it uses statistics/decomposition techniques to eliminate/minimize redundancy. Lossless compression is used only for a few applications with stringent requirements such as medical imaging.

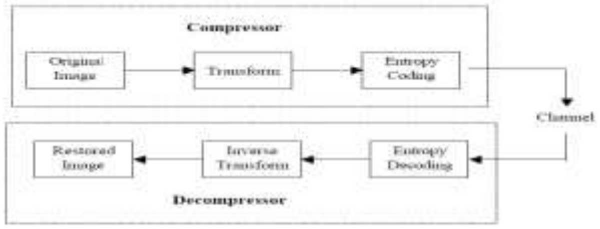


Figure:**Lossless image compression**

Following techniques are included in lossless compression:

1. Run length encoding
2. Huffman encoding
3. LZW coding
4. Area coding

Lossy compression technique:

Lossy schemes provide much higher compression ratios than Lossless schemes. Lossy schemes are widely used since the quality of the reconstructed images is adequate in most applications. By this scheme, the decompressed image is not an identical to the original image, but reasonably close to it.

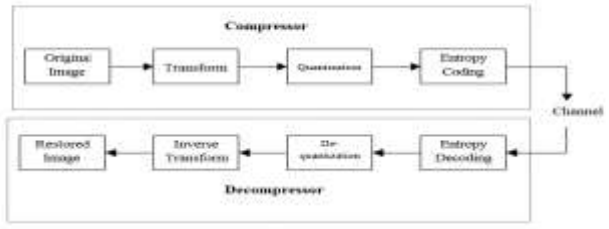


Figure: lossy image compression

As shown in above the outline of lossy compression techniques. The quantization process results in loss of information. The entropy coding after the quantization step, however, is lossless. The decoding is a reverse process. Firstly, entropy decoding is applied to compressed data to get the quantized data. Secondly, dequantization is applied to it &finally the inverse transformation to get the reconstructed image.

Major performance considerations of a lossy compression scheme include:

1. Compression ratio
2. Signal - to – noise ratio
3. Speed of encoding & decoding.

Lossy compression techniques include following schemes:

1. Transformation coding
2. Vector quantization

Lossless Compression Techniques

Run Length Encoding:

This is a very simple compression method used for sequential data. It is very useful in case of repetitive data. This technique replaces sequences of identical symbols (pixels), called runs by shorter symbols. The run -length code for a gray scale image is represented by a sequence {Vi,Ri } where Vi is the intensity of pixel, and Ri refers to the number of consecutive pixels with the intensity Vi as shown in the figure. If both Vi and Ri are represented by one byte, this span of 12 pixels is coded using eight bytes yielding a compression ratio of 1: 5.

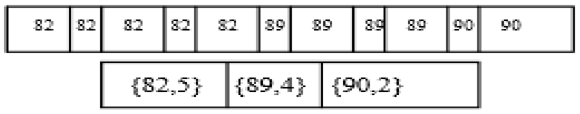


Figure: Run –Length Encoding

Huffman Encoding:

This is a general technique for coding symbols based on their statistical occurrence frequencies (probabilities). The pixels within the image are treated as symbols. The symbols that occur more frequently are assigned a smaller number of bits, while the symbols that occur less frequently are assigned a relatively larger number of bits. Huffman code is a prefix code. This means that the (binary) code of any symbol is not the prefix of the code of any other symbol. [5] Most image coding standards use lossy techniques in the earlier stages of compression and use Huffman coding as the final step.

LZW Coding:

LZW (Lempel- Ziv – Welch) is a dictionary based coding. Dictionary based coding can be static or dynamic. In static dictionary coding, dictionary is fixed during the encoding and decoding processes. In dynamic dictionary coding, the dictionary is updated on the fly. LZW is widely used in computer industry and is implemented as the compress command on UNIX.

Area Coding:

Area coding is an enhanced form of run length coding, reflecting the two-dimensional character of images. This is a significant advance over the other lossless methods. For coding an image it does not make too much sense to interpret it as a sequential stream, as it is, in fact an array of sequences, building up a two-dimensional object. The algorithms for area coding try to find rectangular regions with the same characteristics. These regions are coded in a descriptive form of an element with two points and a certain structure. This type of coding can be highly effective, but it bears the problem of a nonlinear method, which cannot be implemented in hardware. Therefore, the performance in terms of compression time is not competitive, although the compression ratio is.

Lossy Compression Techniques:

Transformation Coding:

In this coding scheme, transforms such as DFT (Discrete Fourier Transform) and DCT (Discrete Cosine Transform) are used to change the pixels in the first image into frequency-domain coefficients (called transform coefficients). These coefficients have several desirable properties. One is the energy compaction property that results in most of the energy of the original data being concentrated in only a few of the significant transform coefficients. This is the basis of achieving the compression. Only those few significant coefficients are selected, and the remaining is discarded. The selected coefficients are considered for further quantization and entropy encoding. DCT coding has been the most common approach to transform coding. It is also adopted in the JPEG image compression standard.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Vector Quantization:

The basic idea behind this technique is to develop a dictionary of Fixed-size vectors, called code vectors. A vector is usually a block of pixel values. A given image is next partitioned into on-overlapping blocks (vectors) called image vectors. Next for each in the dictionary is determined, and its index in the dictionary is used as the encoding of the original image vector. Thus, each image is represented by a sequence of indices that can be further entropy coded.

V. COMPRESSION AND DECOMPRESSION USING HUFFMAN CODING

Huffman Encoding:

Huffman Coding First Huffman coding algorithm was developed by David Huffman in 1951. Huffman coding is an entropy encoding algorithm used for lossless data compression. In this algorithm fixed, length codes are replaced by variable length codes. When using variable-length code words, it is desirable to create a prefix code, avoiding the need for a separator to determine codeword boundaries. Huffman Coding uses such prefix code. Huffman procedure works as follows:

1. Symbols with a high frequency are expressed using shorter encoding than symbols, which occur less frequently [4].

2. The two symbols that occur least frequently will have the same length. The Huffman algorithm uses the greedy approach, i.e. at each step the algorithm chooses the best available option. A binary tree is built up from the bottom up. To see how Huffman Coding works, let's take an example. Assume that the characters in a file to be compressed to have the following frequencies:

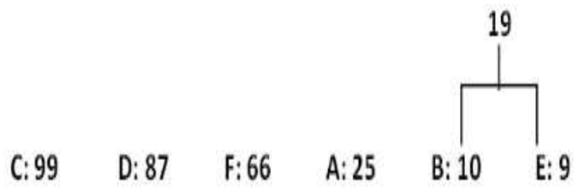
A: 25 B: 10 C: 99 D: 87 E: 9 F: 66

The processing of building this tree is:

1. Create a list of leaf nodes for each symbol and arrange the nodes in the order from highest to lowest.

C: 99 D: 87 F: 66 A: 25 B: 10 E: 9

3. Select two leaf nodes with the lowest frequency. Create a parent node with these two nodes and assign the frequency equal to the sum of the frequencies of two child nodes.



Now add the parent node in the list and remove the two child nodes from the list. And repeat this step until you have only one node left.

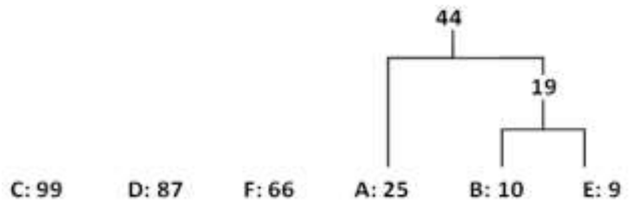


Figure: Huffman encodin(1)

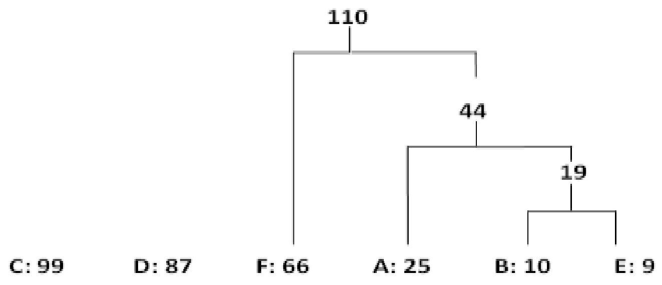


Figure: Huffman encoding (2)

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

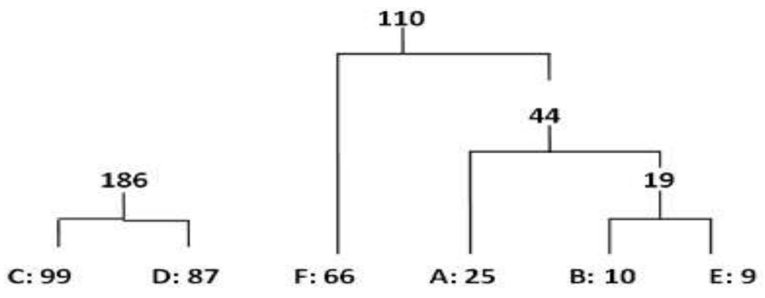


Figure: Huffman encoding (3)

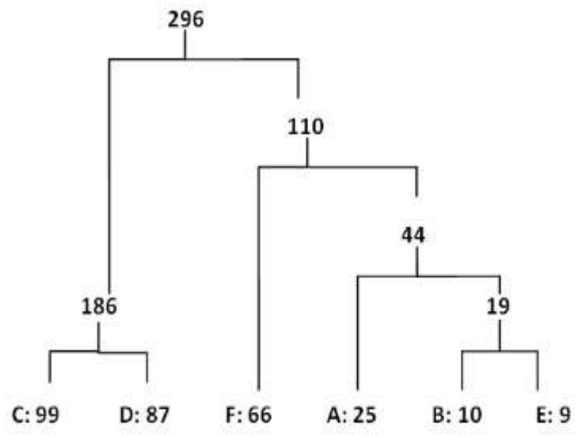


Figure: Huffman encoding (4)

3. Now label each edge. The left child to each parent is labeled with the digit 0 and right child with 1. The code word for each source letter is the sequence of labels along the path from the root to the leaf node representing the letter.

Huffman Codes are shown below in the table

C	00
D	01
F	10
A	110
B	1110
E	1111

Table 1: Huffman Codes.

Huffman decoding:

After the code has been created, coding and/or decoding are accomplished. The code itself is an instantaneous uniquely decodable block code. It is called a block code, because each source symbol is mapped into a fixed sequence of code symbols. It is instantaneous, because each code word in a string of code symbols can be decoded without referencing succeeding symbols. It is uniquely decodable, because any string of code symbols can be decoded in only one way. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner.

For the binary code of table, a left-to-right scans of the encoded string 01101101110111100 reveal that the first valid code word is 01, which is the code for symbol D. The next valid code is 10, which corresponds to symbol F. Valid code for the symbol A is 110; valid code for the symbols B is 1110, valid code for the symbol E is Continuing in this manner reveals the completely decoded message DFABEC, so in this manner the original image or data can be decompressed using Huffman decoding as explained above.

At first, we have as much as the compressor does a probability distribution. The compressor made a code table. The decompressed doesn't use this method though. It instead keeps the whole Huffman binary tree, and of course a pointer to the root to do the recursion process. In our implementation, we'll make the tree as usual, and then you'll store a pointer to last node in the list, which is the root. Then the process can start. We'll navigate the tree by using the pointers to the children whom node has. This process is done by a recursive function which accepts as a parameter a pointer to the current node, and returns the symbol.

VI. ALGORITHM FOR HUFFMAN CODING AND DECODING

Step1- Read the image on to the mat lab.

Step2- Convert the given color image into the grey level image.

Step3- Call a function which will find the symbols (i.e. pixel value which is non-repeated).

Step4- Call a function which will calculate the probability of each symbol.

Step5- Probabilities of symbols are arranged in decreasing order, and lower probabilities are merged and this step is continued until only two probabilities are left, and codes are assigned according to rule that: the highest probable symbol will have a shorter length code.

Step6- Further Huffman encoding is performed i.e. mapping of the code words to the corresponding symbols will result in a compressed data.

Step7- The original image is reconstructed, i.e. decompression is done by using Huffman decoding.

Step8- Generate a tree equivalent to the encoding tree.

Step9- Read input character wise and left to the table II until last element is reached in the table II.

Step10-Output the character encodes in the leaf and returns to the root, and continues the step9 until all the codes of corresponding symbols are known.

VII. APPLICATION

1. Arithmetic coding can be viewed as a generalization of Huffman's coding; indeed, in practice arithmetic coding is often preceded by Huffman coding, as it is easier to find an arithmetic code for a binary input than for a non-binary input.[3]
2. Huffman coding is in wide use because of its simplicity, high speed.
3. Used for 3d medical image compression
4. It also provides a level of security against unlawful monitoring. [5]

VIII. ADVANTAGES AND DISADVANTAGES

Advantages

1. Algorithm is easy to implement.
2. Produce a lossless compression of images [3].
3. The great advantage of Huffman's coding is that, although each character is coded with a different number of bits, the receiver will automatically determine the character whatever their order.
4. Provide the more efficient results than other methods of compression.
5. Compression and decompression speed more as comparing to arithmetic and LWZ

Disadvantages:

1. Efficiency depends on the accuracy of the statistical model used and type of image.
2. Algorithm varies with different formats, but few get any better than 8:1 compression.
3. Compression of image files that contain long runs of identical pixels by Huffman is not as efficient when compared to RLE [3].
4. The Huffman encoding process is usually done in two passes. During the first pass, a statistical model is built, and then in the second pass the image data is encoded based on the generated model. From here we can see that Huffman encoding is a relatively slow process as time is required to build the statistical model in order to archive an efficient compression rate.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

5. It is required to send Huffman table at the beginning of the compressed file, otherwise the decompresses will not be able to decode it. This causes overhead.

IX. CONCLUSION

Huffman coding is an efficient technique for image compression and decompression to some extent. The compression of images for storing and transmitting images can be done so that it indicates that there is no loss of information during transmission. So other methods of image compression can be carried out as namely JPEG method, Entropy coding, etc...As a future work, more focus shall be on improvement of compression ratio.

REFERENCE

[1] jagadish h. pujarandlohitm. Kadlaskar,.‘ A new lossless method of image compression and decompression using Huffman coding techniques’, Department of EEE, B V B College of Engg. & Tech., Hubli, India-580 031

[2] Prof. Rajendra Kumar Patel, ‘An Efficient Image Compression Technique Based on Arithmetic Coding’Assistant Professor, Department of CSE, Patel College of Science &Technology, Bhopal, International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)Volume-2 Number-4 Issue-6 December-2012

[3] MamtaSharma,‘Compression Using Huffman Coding’ ,S.L. Bawa D.A.V. college ,IJCSNS VOL.10 No.5, May 2010

[4] AnmolJyotMaan‘Analysis and Comparison of Algorithms for Lossless Data Compression’, Hyderabad, INDIA, ISSN 0974-2239 Volume 3, Number 3 (2013), pp. 139-146

[5] Sonal, dineshkumar, ‘A study of various image compression Techniques’, Department of computer science & engineering, Guru jhambheswar university of science and technology, hisar

[6] Handout by Julie Zelenski with minor edits by Keith Schwarz, ‘Huffman Encoding and Data Compression’