Ideal Huffman Code for Lossless Image Compression for Ubiquitous Access

T.Kavitha[1], K.Jayasankar[2]
[1]MVSR Engineering College, Hyderabad, India
[2]Vasavi College of Engineering, Hyderabad, India

Article Info

Article history:

Received May 22, 2018
Revised Jul 23, 2018
Accepted Aug 6, 2018

Keywords:

Average Length
Codeword
Compression
Entropy
Huffman code

ABSTRACT

Compression technique is adopted to solve various big data problems such as storage and transmission. The growth of cloud computing and smart phone industries has led to generation of huge volume of digital data. Digital data can be in various forms as audio, video, images and documents. These digital data are generally compressed and stored in cloud storage environment. Efficient storing and retrieval mechanism of digital data by adopting good compression technique will result in reducing cost. The compression technique is composed of lossy and lossless compression technique. Here we consider Lossless image compression technique, minimizing the number of bits for encoding will aid in improving the coding efficiency and high compression. Fixed length coding cannot assure in minimizing bit length. In order to minimize the bits variable Length codes with prefix-free codes nature are preferred. However the existing compression model presented induce high computing overhead, to address this issue, this work presents an ideal and efficient modified Huffman technique that improves compression factor up to 33.44% for Bi-level images and 32.578% for Half-tone Images. The average computation time both encoding and decoding shows an improvement of 20.73% for Bi-level images and 28.71% for Half-tone images. The proposed work has achieved overall 2% increase in coding efficiency, reduced memory usage of 0.435% for Bi-level images and 0.19% for Half-tone Images. The overall result achieved shows that the proposed model can be adopted to support ubiquitous access to digital data.

Corresponding Author:

T.Kavitha,
MVSR Engineering College,
Hyderabad, India.
Email: tkavitha_ece@mvsrec.edu.in

1. INTRODUCTION

The technological advancements has diffused a remarkable growth in the usage of multimedia data in various application uses. This growth is due to: 1) cheap availability of smart phone with processing capability; 2) wide availability of software application for editing and managing multimedia data; 3) the availability of access to internet allows user access to wide range of services, and 4) technological growth impacting the adoption of multimedia. The application that uses huge amount of multimedia data are entertainment organization, medical imaging, distance learning programming using multimedia and remote sensing etc. Digital multimedia data requires huge bandwidth for transmission and huge amount of storage space. For example, a 24 bits/pixels color images of size 640*480 pixel requires close to 1 megabyte of storage space. Despite the growth of technology of transmission and storage, the demand versus availability of storage and bandwidth is still an issue.

Cloud computing evolution has led to rapid changes in a way traditional services and user's operates. Many applications has adopted cloud computing platform to solve various computing and storage problems. The healthcare care service utilizes benefit offered by cloud provider and provides storage and remote access of user's digital multimedia data, such as CT-SCAN images, X-Ray [1] etc. which are stored on cloud platform. Digitization and real-time access to these multimedia data will aid in reducing the cost (i.e. reduce duplication of medical imaging, distribution cost and storage cost.

These medical data need to be stored with very high quality. Therefore efficient compression techniques are required. The compression techniques in cloud platform are directly related to cost. For example, 4 samples of PET-CT [1] medical multimedia data of a particular patient might require four plus gigabyte of storage space. Therefore adopting a good coding technique will aid in reducing cost of storage per bit and improve ubiquitous data access on smart devices [2].

Data transmission over data communication channel can induce cost overhead (e.g. increase in phone bills due to usage of modem, satellite data transmission, etc.). Therefore minimizing the total number of bits to be sent is most desired. This will aid in improving the speed of data transmission efficiency (i.e. real-time). The multimedia data such as images, videos, sound etc. requires large amount of bandwidth which can be reduced by using the right coding methods. A fixed length code does not guarantee of reducing the total code size to be sent. Since some characters in image data occurs more frequent than others, but still it requires the same size (bits) as frequently occurrence character.

Image compression technique has been proven to be an effective solution to solve bandwidth and storage issues. Digital images are majorly composed of spectral and spatial redundancy. Spectral redundancies exist due to relationship among different color planes and spatial redundancies exist due to relationship among neighborhood pixel parameter. By taking the benefits from these redundancies the coding (image compression) technique allows in reducing the number of bits needed in representing a multimedia data. The reconstruction process of an encoded data is known as decoding. The decoding technique is the reverse process of encoding. The goal of any compressing technique is to minimize the number of bits as probable, without affecting the pictorial quality of reconstructed multimedia data.

Digital Imaging and Communications in Medicine (DICOM), a medical imaging standard [1] is currently used mechanism for sharing data across service platforms. A digital multimedia data is generally composed of pixels of two dimensional arrays. These digital images are further classified into bi-tonal or bi-level which consist of two intensity levels (black and white), gray scale images which is also called as continuous-tone images where the pixel value is in range of [0-255], halftone images comprise of blurred discrete dots and the color images which consist of pixels corresponding to RGB (Red, Green and Blue) components. The image compression methods are generally classified in to two type's lossy and lossless techniques as shown in Figure 1.

Lossless compression technique has been adopted in application where high-quality is required such as medical imaging, Property registration certificate etc. Lossy techniques provide much higher compression ratio than Lossless but the quality is not as good as lossless compression methodologies achieve. In [3] evaluated the performance of lossy and lossless compression scheme for medical data compression. Their model minimized the bandwidth, storage and transmission cost. However it induces high computing overhead. To address in [4] presented a HEVC coding technique to minimize computational complexity and reduce file size. However their model supports lossless compression only up to coding unit level [5] and to enable lossless compression requires huge computing platform.

In [6] presented a parallel implementation of Bit plane coding to achieve efficient compression ratio. This model achieves fine grained parallelism and speedups, however these model may not be applicable for ubiquitous computing on resource starved smart devices. To support data compression on such resource starved platform a simple compression technique must be developed. In [7] presented context aware probability based fixed code length encoding technique to address the issue of traditional arithmetic coder technique which requires priori knowledge of code word for renormalization process. The renormalization process is eliminated in fixed length coding, however it cannot assure reduction in number of bits for encoding.

A fixed length code does not guarantee of reducing the total code size to be sent. Since some characters in image data occurs more frequent than others, but still it requires the same size of bits as frequently occurrence character. Therefore prefix-free code for compression can be effective solution. In [8] presented an Variable-Length Codes (VLCs) and [9] presented an hybrid compression algorithm, both [8] and [9] have adopted Huffman code [10] to reduce total bit length and to support efficient lossless compression over wireless channel.

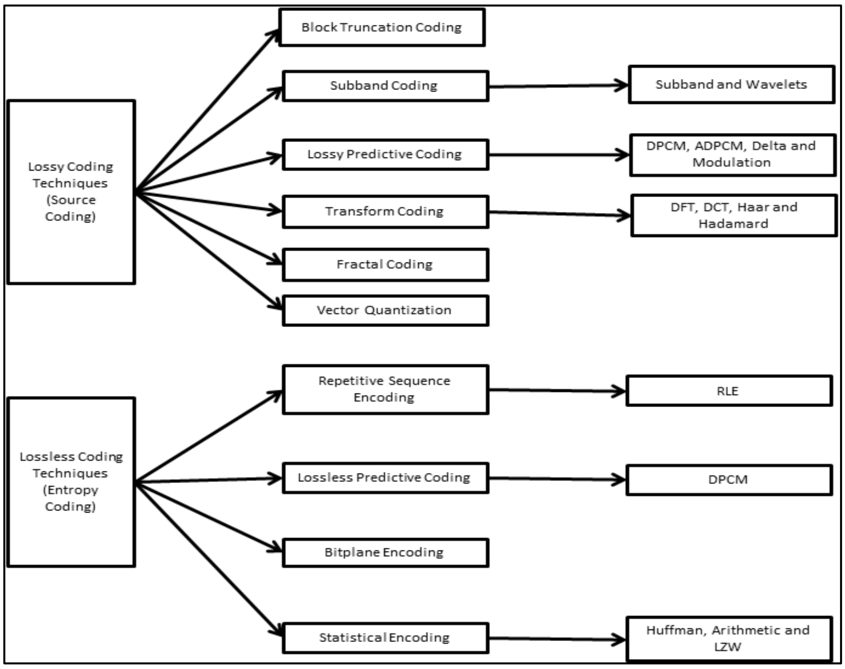
Lossy Coding Techniques (Source Coding)
Block Truncation Coding
Subband Coding
Subband and Wavelets
Lossy Predictive Coding
DPCM, ADPCM, Delta and Modulation
Transform Coding
DFT, DCT, Haar and Hadamard
Fractal Coding
Vector Quantization
Lossless Coding Techniques (Entropy Coding)
Repetitive Sequence Encoding
RLE
Lossless Predictive Coding
DPCM
Bitplane Encoding
Statistical Encoding
Huffman, Arithmetic and LZW

Figure 1. Taxonomy of Image Compression Technique

The author has presented efficient compression technique using modified Huffman, in extension to that work author has proposed few changes in this work. The codeword length for EOL=7 bits, one more 3-bit code is introduced, one 12-bit code is reduced to 7-bit and there are no 13-bit codes. In this work both encoding and decoding of images is carried which includes fill bits between the data bits and EOL symbol and the total number of bits transmitted for a scan line is made equal to multiples of 8 because of this appreciable compression is seen which is presented in this work.

This work presents an efficient image compression technique which is an optimized prefix-free codeword technique by modifying state-of-art Huffman code technique. The optimality proof of Proposed Ideal Huffman codeword technique over Huffman code is presented. The proposed codeword technique reduces the number of bits required for encoding and also reduces computation time (encoding and decoding) and memory requirement. The model achieves lossless compression with high compression ratio with good quality reconstructed images. The model can support ubiquitous access of data through smart devices.

The paper organization is as follows: The literature survey is presented in section two. The proposed models are presented in Section three. The simulation results and the experimental study are presented in the section four. The concluding remark is discussed in the last section.

2. LITTERATURE SURVEY

There have been several methodologies that have been proposed in recent time in order to improve the performance of compression, reduce the total number bits for encoding and reducing computing complexity for both lossy and lossless compression which are surveyed below.

In [7] presented a context aware binary arithmetic coding considering fixed coding length. Here they addressed the issue pertaining to renormalization overhead in computing for longer codeword of Apriori unknown length. This model is adaptive with respect to variable size of code. The model adopts variable size sliding window to compute the probability to find high precision code. Experiments are conducted and performance of their model is evaluated in terms of computation time, coding rate and PSNR (Peak-signal-to-noise-ratio). The outcome shows their model achieves significant performance improvement over MQ coder and M coder.

In [11] developed a model to construct dictionary for sparse multimedia data representation. They adopted dictionary based learning technique to build collaborative dictionaries for sparse multimedia data. They presented boosting based dictionaries which result in obtaining better sparse representation. Two optimization models are presented which improves generalization characteristics of trained dictionaries. Based on this dictionary, they developed a new compression technique. Experiments are conducted and outcome shows better performance than JPEG-2000 in terms of PSNR and Bit-error rate.

In [6] established an efficient compression technique to improve the speedup of compression. They presented a wavelet-based image coding method by adopting parallelization technique. This model is parallelized with GPU to achieve fine grained parallelism by adopting Bit plane coding. The model also presented smart memory management and thread to data mapping mechanism for better communication among inters threads. Experiment are conducted for high resolution images and outcomes shows it achieves better speedups and it also consume less power or energy.

In [12] constructed a codebook and presented vector quantization technique to determine the quality of compressed data. One of the widely used codebook design Linde–Buzo–Gray (LBG) tends to suffer from local minima problem and get trap to local minima for initial codebook generation. To address this, they designed a high quality initial codebook design model. The model divides the training vector into groups by utilizing variance and means characteristics parameter of training vectors. Then the initial codebook is generated by choosing codeword from each group. Experiments are conducted to evaluate the performance of their initial codebook generation over existing technique shows better performance in term of PSNR.

In [13] presented a model to speed up the encoding by adopting Vector quantization technique. Adopting Vector quantization for encoding is generally expensive in term of computing time. Since, it has to find closest codeword for each input vector from entire codebook. To address this time complexity, they presented a fast searching strategy by adopting multilevel elimination criteria. The multilevel elimination is derived from feature of norm, mean and variance. The multilevel features are optimized to speed up the encoding process. The elimination condition of level one is fixed to mean inequality bound due to its narrow search width, and selectivity order at level two and in level three of norm and variance inequalities bounds is optimized based on the location of input vectors and distribution of codeword in term of the features considered. Experiment outcome shows effectiveness of their model over existing techniques.

The wide research survey carried out in this work shows that reducing the bit size for encoding and transmission over data channel is most desired. Image compression technique plays a critical role in reducing size of a multimedia data. The major challenges of the all these state of art technique is to reduce data size and computing complexity. The overall survey shows that none of the work carried out so far had considered compression on resource starved device such as smart phone. The digital data access through smart phone device enables us to achieve ubiquitous computing. To achieve ubiquitous computing the compression technique should use minimum resource and maximize compression ratio without affecting the quality of reconstructed images. In next section we present an Ideal Huffman encoding technique.

3. PROPOSED IDEAL HUFFMAN ENCODING TECHNIQUE

Here we present an efficient image compression technique by optimizing Huffman encoding techniques. The codeword size is reduced without affecting the quality of image. An optimality proof of proposed Ideal Huffman codeword over existing Huffman code is presented.

The drawback of existing modified Huffman compression technique is that it requires large memory size (since the code length is around 183 and maximum length of code is 13) and compression ratios are low [14]. It also induces computation overhead due to larger code length. Minimizing number of code words and their corresponding lengths, without out effecting data loss is the major objective of proposed technique.

The proposed technique minimizes the codeword size for better efficiency and achieved better compression ratios and less encoding and decoding time.

Proposed code words are segmented into three segments. First segment comprises of multiples of 144 up to 1728 and contains a total of 24 codes (12 each for black and white). Next segment comprises of multiple of 13 up to 143 and contains a total of 22 codes (11 each for black and white). The last segment has the remaining 24 terminating codes (12 each for white and black). Additional details of proposed code words is available in [15], few codewords are changed compared to [15] . All segments put together we consider a total of 71 code words and the maximum length is observed to be 12 bits and very few number of 12-bit codes. The proposed code words can be uniquely decoded; reconstructed data is lossless and they satisfy Kraft McMillan inequality conditions. The memory consumption is reduced and the encoding and decoding is also reduced. The proposed codeword run length (RL) is computed and is represented as $RL = 144 * s_1 + 13 * s_2 + t$, where t is terminating code, $s_1$makeup codes, s_2 is makeup code.

3.1. Proposed Huffman Code Optimality Justification:

Let consider a set of code words represented using a binary tree, where right and left divisions map to either a 1 or 0 bit encoding operation. The character that is encoded will be a node on the tree, reachable in a sequence of right-left (i.e. 1 or 0 bit) changes from root to the character's nodes.

In proposed technique, every character that is encoded is considered to be a leaf node and the tree formed contains no occurrences of characters to be encoded. In other words every character to be encoded has an independent tree and no characters to be encoded are present in trees formed. This operation enables in character wise efficient encoding/decoding operation demarcation. The proposed technique is adopted to determinate the optimality i.e. to encode an image with set of characters and occurrence of each character presence with least number of bits. For instance if we consider a fixed code length of 8 bits, then 40 character of an image will require 320 bits size. To address this, here we minimize the number of bits by finding a new code words that can encode an image in less than 320 bits. The significant factor is to use long code for occasional appearing characters and shorter code for repeatedly occurring characters.

Therefore minimizing the bits required for encoding is most desired. Let U_xbe the codeword length for x^thcharacter in bits and V_x is the rate of occurrence/frequency of x^thcharacter to be encoded. Considering binary tree construction of code words, U_x\ also represents number of sub level of each character of a binary coded tree. Codeword Traversal (CT) of a given code tree is measured as follows.

$$CT(H) = \sum_{x=1}^{K} V_x \times U_x^H \qquad (1)$$

Where variable K represents total number of characters to be encoded, H is the Huffman tree and its corresponding code word lengths are U_x^H. The code word lengths U_x^Hcan be expressed as a function of probability of occurrence of x^{th}character and its corresponding code word W. i.e. $U_x^H = f(P_x^H, W_x^H)$.

If TBL represents the total length of encoded words of K characters, its computation is formulated as:

$$TBL(H) = \sum_{x=1}^{K} K \times V_x \times U_x^H \qquad (2)$$

For example, a 4 bit code 0100 will have a leaf at fourth level of the tree and which is a code of that given character. The aim of the proposed method is to minimize $TBL(H)$ to achieve efficient compression. In Equation (2) number of characters K and their occurrences $V_x$are constant and data dependent. Efficient establishment of code words i.e. $U_x$would reduce $TBL(H)$. The objective of proposed method can be defined as

$$\min_{\forall x \in K\ establish\ U_x} TBL(H) \qquad (3)$$

Let $H_{huffman}$ represent the existing Huffman code tree [14] and proposed tree is represented usingH_{Ideal}. To achieve our objective we need to prove that $TBL(H_{Ideal}) < TBL(H_{huffman})$. i.e

$$\sum_{x=1}^{K} K \times V_x \times U_x^{H_{Ideal}} < \sum_{x=1}^{K} K \times V_x \times U_x^{H_{huffman}} \qquad (4)$$

As K and $V_x$are constant for the same data, Equation (4) can be expressed as

$$\sum_{x=1}^{K} U_x^{H_{Ideal}} < \sum_{x=1}^{K} U_x^{H_{huffman}} \qquad (5)$$

Using functional representation we can state that

$$\sum_{x=1}^{K} f\left(P_x^{H_{Ideal}}, W_x^{H_{Ideal}}\right) < \sum_{x=1}^{K} f\left(P_x^{H_{huffman}}, W_x^{H_{huffman}}\right) \qquad (6)$$

In Equation (6) the number of code words in $W_x^{H_{Ideal}}$is 71 and $W_x^{H_{huffman}}$is 183. In maximum size of $H_{huffman}$is 13 whereas maximum size of $H_{Ideal}$is 12. As probabilities do not change the time required to traverse $H_{huffman}$ trees are far greater than H_{Ideal} due to number of code words and their average lengths are far larger when compared to proposed code words.

Therefore it can be concluded that $TBL(H_{Ideal}) < TBL(H_{huffman})$

3.2. Length of Huffman Code:

Huffman coding procedure generates optimum code, but the average length of any code will depend on size of the alphabet and the probabilities of symbols of source 'S'. According to Huffman, if a source 'S' has an entropy H(s) , if we have an optimal Huffman code for the source 'S' with average length $\mathbb{L}_{\rightarrow}$, the Average length is bounded below by the entropy and above by entropy plus 1 bit[16].

$$H(s) \leq \mathbb{L}_{\rightarrow} \leq H(s) + 1 \qquad (7)$$

The difference between H(s) and $\mathbb{L}_{\rightarrow}$is given by the Kraft McMillan inequality

$$H(s) - \mathbb{L}_{\rightarrow} \quad \leq \; log_2 \sum_{i=1}^{K} 2^{-li} \qquad (8)$$

where K is the number of symbols/codes.
As the code is an optimal code, according to Kraft McMillan $\sum_{i=1}^{K} 2^{-li} \leq 1$, therefore

$$H(s) - \mathbb{L}_{\rightarrow} \;\leq\; 0 \qquad (9)$$

which is called as redundancy.
The coding efficiency CE of proposed code word is evaluated using following equation [17]

$$CE = \frac{H(s)}{\mathbb{L}_{\rightarrow}} \qquad (10)$$

Where $H(s)$ is the entropy and $\mathbb{L}_{\rightarrow}$ is average length of code words, the $\mathbb{L}_{\rightarrow}$ is computed using the following equation, where $P(a_i)$ are the probabilities and l_i code lengths.

$$\mathbb{L}_{\rightarrow} = \sum_{i=0}^{K} P(a_i)\, l_i \qquad (11)$$

where K is the number of symbols.

The average length of code and above conditions of lower bound and upper bound on average length of code, coding efficiency, are evaluated in the proposed work. In the next section experimental results to prove our claim are presented.

4. SIMULATION RESULTS AND ANALYSIS

The experiment is carried out on windows 10 operating system, Intel I-7 class processor, 8 Gb RAM with 4GB dedicated CUDA graphic card. The algorithm of both proposed and existing approach is implemented in C# 6.0 programming language using Microsoft Visual Studio Dot net framework 4.0 and simulation is conducted using Matlab 2016b framework. The experiments are carried out for multimedia document and images obtained from [18]. The performance evaluation of proposed model over existing model is carried out in term of Compression Ratio (CR), Encoding Time (ET), Decoding Time (DT) and Memory Utilized (MU) for both bi-level and half tone images. This work further carried performance evaluation for entropy, code length and coding efficiency for bi-level images.

For first case study, let us consider performance evaluation for bi-level images. The bi-level images are obtained from [18], the first 8 images are considered for evaluation. All images have same dimension (W*H) of 1728*2339 and same resolution of 200 dpi. The result obtained is tabulated in Table 1 and 2. It is seen from Table 1 that proposed model achieve better compression ratio than existing model. The image 4 has the least compression ratio and image 2 has the highest compression ratio for both proposed and existing model. An average compression ratio performance improvement of 33.44% is achieved by proposed model over existing approach.

Achieving better compression ratio may not guarantee ubiquitous access through resource starved smart devices. To support multimedia data access on such platform the algorithm should be designed in such a way that it should minimize memory usage and reduce computing time. Experiments are conducted to evaluate the performance in term of memory usage and computation time for both proposed and existing approaches which is tabulated in Table 1 and 2. The proposed model utilizes less memory when compared to existing model. An average reduction of 0.435% memory is achieved by proposed over existing model.

Table 1. Experiments are conducted to evaluate the performance in term of memory usage and computation time for both proposed and existing approaches

Images	Compression Ratio		Memory Utilized	
Bi-level	Proposed	Existing	Proposed	Existing
1	18.13377123	12.00986996	460891328	464162816
2	19.71413215	13.00445728	443153408	457236480
3	10.53515521	7.043233753	461783040	455630848
4	6.371446652	4.285723824	461991936	462417920
5	10.0346674	6.711374835	455626752	456781824
6	13.34339838	8.885209025	450818048	451346432
7	6.469949349	4.35160719	447696896	447467520
8	10.9054496	7.278028177	446146880	448917504
Avg	11.93849625	7.946188005	453513536	455495168

The computation time of both proposed and existing approach is evaluated in term of encoding time, decoding time and total execution time which are shown in Table 2. An average encoding time of 0.055 seconds and 0.059 seconds is achieved by proposed and existing approach respectively. An average reduction of 6.95% encoding time is achieved by proposed model over existing approaches. An average decoding time of 0.0034 seconds and 0.0146 seconds is achieved by proposed and existing approach respectively. An average decoding time reduction of 76.56% is achieved by proposed model over existing approaches. An average total computing time of 0.0587 seconds and 0.074 seconds is achieved by proposed and existing approach respectively. An average computing time reduction of 20.73% is achieved by proposed model over existing approaches.

Table 2. The computation time of both proposed and existing approach is evaluated in term of encoding time, decoding time and total execution time

Images	Encoding Time		Decoding Time		Total Time	
Bi-level	Proposed	Existing	Proposed	Existing	Proposed	Existing
1	0.04563	0.031343	0.003401	0.005081	0.04903	0.036425
2	0.025628	0.027699	0.002439	0.002735	0.028068	0.030434
3	0.042783	0.042792	0.002511	0.002782	0.045294	0.045574
4	0.066193	0.073051	0.003225	0.052285	0.069418	0.125336
5	0.085812	0.086576	0.002817	0.002817	0.088629	0.089393
6	0.032241	0.035526	0.002642	0.019647	0.034883	0.055174
7	0.063866	0.097745	0.002399	0.003447	0.066264	0.101192
8	0.080464	0.080972	0.008078	0.028583	0.088542	0.109555
Avg	0.055327	0.059463	0.003439	0.014672	0.058766	0.074135

The overall outcome shows that the proposed achieves significant performance improvement over existing approaches in term of compression ratio, encoding, decoding time, total computing time and memory utilization.

The user generally stores various form of scanned multimedia data which is composed of just text or it may consist of image with text in it. To evaluate performance of proposed approach efficiency on such data, the second case study, the Half tone image obtained from [14] are considered. Experiments are conducted for 7 images obtained from [14].All images have same dimension (W*H) of 1728*2339 and same resolution of 200 dpi. The result obtained is tabulated in Table 3 and 4. It is seen from Table 3 that proposed model achieve better compression ratio than existing model. The image 13 has the least compression ratio and image 17 has the highest compression ratio for both proposed and existing model.

An average compression ratio of 1.61% and 1.08% is achieved for proposed and existing approaches respectively. An average compression ratio performance improvement of 32.578% is achieved by proposed model over existing approach.

Table 3. Experiments are conducted for Half-tone Images also to evaluate the performance in terms of memory usage and computation time for both proposed and existing approaches

Images	Compression Ratio		Memory Utilized	
Half Tone	Proposed	Existing	Proposed	Existing
11	0.695582544	0.470711486	450809856	454340608
12	2.075501752	1.395822175	440967168	448059008
13	0.644204385	0.436092239	449675264	449232896
14	2.189180108	1.471817463	457629696	459351168
15	1.05286834	0.712055561	443772928	441708544
16	1.914592216	1.284225092	453025792	449961984
17	2.706861682	1.833576089	457592832	457089024
Avg	1.611255861	1.086328586	450496219.4	451391890.3

Experiments are conducted for Half-tone Images also to evaluate the performance in terms of memory usage and computation time for both proposed and existing approaches which is tabulated in Table 3 and 4 for half tone images. The proposed model utilizes less memory when compared to existing model. An average reduction of 0.19% memory is achieved by proposed over existing models. The computation time of both proposed and existing approach is evaluated in terms of encoding time, decoding time and total execution time which are shown in Table 4.

Table 4. The computation time of both proposed and existing approach is evaluated in terms of encoding time, decoding time and total execution time

Images	Encoding Time		Decoding Time		Total Time	
Half Tone	Proposed	Existing	Proposed	Existing	Proposed	Existing
11	0.023428	0.025051	0.00275	0.00596	0.026178	0.031011
12	0.008898	0.01096	0.002141	0.010973	0.01104	0.021933
13	0.036263	0.026678	0.003871	0.010692	0.040133	0.037371
14	0.009292	0.00925	0.005175	0.005725	0.014467	0.014976
15	0.028917	0.030166	0.002925	0.006034	0.031842	0.0362
16	0.009318	0.010014	0.002162	0.05401	0.01148	0.064024
17	0.437284	0.607159	0.002941	0.009798	0.447082	0.6101
AVG	0.079057	0.102754	0.003138	0.014742	0.083174	0.116516

An average encoding time of 0.079 seconds and 0.102 seconds is achieved by proposed and existing approach respectively. An average reduction of 23.06% encoding time is achieved by proposed model over existing approaches. An average decoding time of 0.0031 seconds and 0.0147 seconds is achieved by proposed and existing approach respectively. An average decoding time reduction of 78.71% is achieved by proposed model over existing approaches. An average total computing time of 0.083 seconds and 0.11 seconds is achieved by proposed and existing approach respectively. An average computing time reduction of 28.61% is achieved by proposed model over existing approaches.

The overall outcome shows that the proposed achieves significant performance improvement over existing approaches in term of compression ratio, encoding time, decoding time, total computing time and memory utilization.

Third study, is carried by considering only run lengths of white and black pixels without any fill bits. Experiments are conducted for Bi-level images obtained from [18] to evaluate performance of following metric which is tabulated in Table 5, Entropy $H(s)$ is the measure of average information content per symbol, Coding Efficiency $(CE)$is defined as the ratio of entropy to average length, and Average Length $\mathbb{L}_{\rightarrow}$ of the code is the number of bits per symbol as shown in Table 5. Average entropy of 9.969 is obtained considering all bi-level images. The $\mathbb{L}_{\rightarrow}$ is computed using Equation (12).An average codeword length of 12.74 bits/sample and 13.038 bits/symbol is achieved for proposed and existing model respectively. An average improvement of 0.294bits/symbol is achieved by proposed $\mathbb{L}_{\rightarrow}$ over existing $\mathbb{L}_{\rightarrow}$. The coding efficiency (CE) is computed using Equation (12). The proposed model achieves better coding efficiency over existing model. An average improvement of 1.99% is achieved with proposed model over existing model for all Images except for Image 7, the reason being it is having huge different combinations of run lengths. For existing method the codewords are formed from two tables and for proposed the codewords are formed from three tables. For few run lengths the length of the codeword will be more for proposed than existing.

Table 5. Experiments are conducted for Bi-level images obtained from [18] to evaluate performance of following metric

Bi-level Image	Entropy	$\mathbb{L}_{\rightarrow}$(Proposed)	$\mathbb{L}_{\rightarrow}$(Existing)	CE(Proposed) %	CE(Existing) %	$\mathbb{L}_{\rightarrow}$(Difference)	% Improvement
1	9.17	10.7514	11.3047	85.294	81.1191	0.553337	4.894735
2	12.5	17.0002	17.5197	73.5091	71.3294	0.5195	2.965233
3	9.371	11.8112	12.2704	79.3425	76.3732	0.4592	3.742339
4	7.921	9.2704	9.2908	85.4408	85.2532	0.0204	0.219572
5	9.18	11.0983	11.2793	82.7136	81.3863	0.181	1.60471
6	9.939	13.8788	14.1414	71.6135	70.2837	0.2626	1.856959
7	9.05	10.3255	10.1466	87.6432	89.1885	-0.1789	-1.76315
8	12.63	17.8137	18.3508	70.8893	68.8144	0.53711	2.926901
AVG	9.969	12.74369	13.038	79.5557	77.9685	0.294281	1.995177

To show that Average length is bounded below by the entropy and above by entropy plus 1 bit, simulation is carried on all bi-level images. The lower bound is always satisfied but the upper bound may not be satisfying for some type of sources where especially the source has huge number of symbols with wide variation of probabilities. To derive how much it varies from minimum required number of bits, the difference between $\mathbb{L}_{\rightarrow} - H(s)$ with respect to $H(s)$ can be computed [17]. This parameter conveys that we would need x% of more bits than the entropy $H(s)$. The results are shown in 6 where the bounds on Average length is satisfying for few Images and other Images need extra bits than minimum with respect to entropy $H(s)$.

$$x\% = \frac{\mathbb{L}_{\rightarrow} - H(s)}{H(s)} * 100 \tag{12}$$

The parameters evaluated in Table 5 and table 6(a) & (b) are applicable only for Bi-level images it cannot be carried for Half-tone Images as the pixel values are not 1 and 0.The results shows that bounds on Average Length is satisfying for Images 1,4,5,7 and not satisfying for Images 2, 3, 6 and 8 but for all Images the x% of bits required more than entropy is less for proposed method than existing method. A maximum of 12.77% reduction in number of bits is achieved for Image 2 and a minimum of 1.09% for Image 4.On an Average 6.32409 % less bits are required for coding.

For Image 7 the proposed method has consumed 2% more bits than existing because that Image [18] consists of complex Run lengths of white and black pixels. For existing method the codewords are formed from two tables and for proposed the codewords are formed from three tables, for some run lengths the length of the codeword will be more for proposed than existing.

Table 6(a). Experiments are conducted for Bi-level images

Image Bi-level	White entropy $H_w(s)$	Black entropy $H_b(s)$	Avg. Len Exist (white)	Avg. Len Exist (black)	Avg. Len Prop (white)	Avg. Len Prop (black)
1	5.4715	3.6988	6.3857	4.9191	6.1033	4.6481
2	7.9476	4.5491	9.8887	7.631	10.033	6.9672
3	5.7137	3.6576	7.2859	4.9845	6.9354	4.8758
4	4.7373	3.1834	5.3118	3.979	5.3557	3.9147
5	5.7654	3.4144	6.7593	4.52	6.6553	4.443
6	6.1976	3.7415	9.4878	4.6536	9.2482	4.6306
7	5.917	3.1326	6.7928	3.3538	6.9755	3.35
8	6.8624	5.7656	8.9731	9.3777	9.0602	8.7535

Table 6(b). Experiments are conducted for Bi-level images

Image Bi-level	% of bits req. more than white $H_w(s)$ Exist	% of bits req. more than black $H_b(s)$ Exist	% of bits req. more than white $H_w(s)$ Prop	% of bits req. more than black $H_b(s)$ Prop	% of total bits req. more than H(s) Exist	% of total bits req. more than H(s) Prop
1	16.708	32.991	11.547	25.665	49.69	37.21
2	24.423	67.747	26.239	53.156	92.17	79.39
3	27.516	36.278	21.382	33.306	63.79	54.68
4	12.127	24.992	13.054	22.972	37.12	36.02
5	17.239	32.381	15.435	30.125	49.62	45.56
6	53.088	24.378	49.222	23.763	77.46	72.98
7	14.801	7.0612	17.889	6.9399	21.86	24.82
8	30.757	62.649	32.027	51.823	93.40	83.84
Avg	24.583	36.06	23.349	30.969	60.64	54.32

The overall result achieved shows the efficiency of proposed approach over existing approach. The model designed can be considered for future ubiquitous access through smart device. In next section the conclusion and future work are discussed.

5. CONCLUSION

The compression technique is composed of lossy and lossless compression technique. The lossy achieve higher compression ratio over lossless. However it may not applicable where quality of reconstructed image requirement is high such as in medical imaging, scanned document transmission and any of the wireless communication. Here we considered Lossless image compression technique where an ideal Huffman code is proposed.

The study shows that using proposed approaches for digital data image compression can reduce number of bits for encoding and achieve better compression ratio up to 33.44% and 32.578% compared in comparison to existing approaches for Bi-level and Half- tone images respectively. The evaluation was conducted only for scanned images of bi-level and half tone images. The proposed approaches reduce memory consumption up to 0.435% and 0.19% over existing approaches for Bi-level and Half- tone images respectively. The proposed approaches reduce encoding, decoding and total computation time up to 6.95%, 76.56% and 20.73% for bi-level images respectively and 23.06%, 78.71% and 28.61% respectively for half tone images.

The experiments are further carried to reduce average length of code and for verifying the bounds on average length including coding efficiency. It is proved that a reduction of 0.294 bits/symbol on average length and 1.9% improvement in terms of coding efficiency is achieved by proposed model over existing approach. The average length and coding efficiency for Image 7 for proposed is less than existing approach, for raw data but when we transmit the encoded data along with fill bits and EOL for real time application the compression factors, memory consumption and total computation time are improved appreciably compared to the existing shown in Table 1, 2, 3 and 4.The bounds on average length shows an overall improvement of 6.32409 % less bits are required than minimum compared with existing.

The overall results achieved show that the proposed model can support ubiquitous access of digital data stored in cloud platform through smart devices. The model will aid in reducing the overall cost of storage and processing of digital data on cloud platform. The future design would further consider developing a better compression and consider evaluation under digital data transmission under wireless environment.

REFERENCES

[1] Abdolvahab Ehsani Rad, Mohd Shafry Mohd Rahim, Alireza Norouzi," Digital Dental X-Ray Image Segmentation and Feature Extraction", TELKOMNIKA, Vol. 11, No. 6, June 2013, pp. 3109 ~ 3114. e-ISSN: 2087-278X

[2] Xiaofeng Wu, Shigang Hu, Zhiming Li, Zhijun Tang, Jin Li, Jin Zhao, "Ideal Huffman Code for Lossless Image Compression for Ubiquitous Access", TELKOMNIKA Indonesian Journal of Electrical Engineering Vol.12, No.3, March 2014, pp. 1895 ~ 1905.

[3] S. Parikh, H. Kalva and V. Adzic, "Evaluation of HEVC compression for high bit depth medical images," IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, pp. 311-314, 2016.

[4] D. Anand M, Dr V Mathivananr, "Quantization Encoding Algorithm Based Satellite Image Compression", Indonesian Journal of Electrical Engineering and Computer Science Vol. 8, No. 3, December 2017, pp. 740 ~ 742

[5] Mengmeng Zhang*1, Jianfeng Qu1, Huihui Bai2," Fast Intra Prediction Mode Decision Algorithm for HEVC", TELKOMNIKA, Vol. 11, No. 10, October 2013, pp. 5703 ~ 5710 ISSN: 2302-4046

[6] P. Enfedaque; F. Auli-Llinas; J. C. Moure, "GPU Implementation of Bitplane Coding with Parallel Coefficient Processing for High Performance Image Compression," in IEEE Transactions on Parallel and Distributed Systems , vol.PP, no.99, pp.1-1. 2017.

[7] F. Aulí-Llinàs, "Context-Adaptive Binary Arithmetic Coding With Fixed-Length Codewords," in IEEE Transactions on Multimedia, vol. 17, no. 8, pp. 1385-1390, 2015.

[8] Enyan Sun, Xuanjing Shen, Haipeng Chen, Chuanyun Wang, "Distributed Image Compression and Transmission Scheme in Wireless Multimedia Sensor Networks", TELKOMNIKA Indonesian Journal of Electrical Engineering Vol.12, No.1, January 2014, pp. 661 ~ 668.

[9] H. C. Kuo and Y. L. Lin, "A Hybrid Algorithm for Effective Lossless Compression of Video Display Frames," in IEEE Transactions on Multimedia, vol. 14, no. 3, pp. 500-509, 2012.

[10] D. A. Huffman, "A method for the construction of minimum redundancycodes," Proc. I.R.E., vol. 40, no. 9, pp. 1098–1102, Sep. 1952.

[11] M. Nejati, S. Samavi, N. Karimi, S. M. R. Soroushmehr and K. Najarian, "Boosted Dictionary Learning for Image Compression," in IEEE Transactions on Image Processing, vol. 25, no. 10, pp. 4900-4915, 2016.

[12] X. Ma, Z. Pan, Y. Li and J. Fang, "High-quality initial codebook design method of vector quantisation using grouping strategy," in IET Image Processing, vol. 9, no. 11, pp. 986-992, 2015.

[13] Y. F. Xie, J. H. Liu, C. F. Zhang, L. S. Kong and J. L. Yi, "Codewords Distribution-Based Optimal Combination of Equal-Average Equal-Variance Equal-Norm Nearest Neighbor Fast Search Algorithm for Vector Quantization Encoding," in IEEE Transactions on Image Processing, vol. 25, no. 12, pp. 5806-5813, 2016.

[14] K. Wakabayashi, "Research and Events that Permitted Facsimile Use to Explode in Japan," 2009 IEEE Globecom Workshops, Honolulu, HI, 2009, pp. 1-6.

[15] T. Kavitha and Dr. K. Jaya Sankar, "An Efficient Compression Technique for ITU-T Group 3 Coded Images Using Variable Length Codes with Reduced Average Length", 2016 IEEE International Conference on India International Conference On Information Processing (IICIP-2016), pp. 1-6.

[16] Khalid Sayood, "Introduction to Data compression", Morgan Koffman publishers, third edition, San Francisco, CA, 2005.

[17] David salmon "Data compression:, The complete reference" CA,USA,3rd edition,2004.

[18] Standard Test Images. Compiled by Mike Waken, University ofMichigan--ww.ece.rice.edu/~wakin/images/.