NEAR-LOSSLESS COMPRESSION SCHEME USING HAMMING CODES FOR NON-TEXTUAL IMPORTANT REGIONS IN DOCUMENT IMAGES

Prashant Paikrao
Research Scholar, Department of E&TC, SGGS Institute of Engineering and Technology, Nanded, (India).
E-mail: plpaikrao@gmail.com; 2019pec201@sggs.ac.in

Dharmapal Doye
Professor, SGGS Institute of Engineering and Technology, Nanded, (India).

Milind Bhalerao
Assistant Professor, SGGS Institute of Engineering and Technology, Nanded, (India).

Madhav Vaidya
Assistant Professor, SGGS Institute of Engineering and Technology, Nanded, (India).

Reception: 27/11/2022 **Acceptance**: 12/12/2022 **Publication**: 29/12/2022

Suggested citation:

Paikrao, P., Doye, D., Bhalerao, M., y Vaidya, M. (2022). Near-lossless compression scheme using hamming codes for non-textual important regions in document images. *3C TIC. Cuadernos de desarrollo aplicados a las TIC, 11*(2), 225-237. https://doi.org/10.17993/3ctic.2022.112.225-237

https://doi.org/10.17993/3ctic.2022.112.225-237

ABSTRACT

Working at Bell Labs in 1950, irritated with error-prone punched card readers, R W Hamming began working on error-correcting codes, which became the most used error-detecting and correcting approach in the field of channel coding in the future. Using this parity-based coding, two-bit error detection and one-bit error correction was achievable. Channel coding was expanded further to correct burst errors in data. Depending upon the use of the number of data bits 'd' and parity bits 'k' the code is specified as (n, k) code, here 'n' is the total length of the code (d+k). It means that 'k' parity bits are required to protect 'd' data bits, which also means that parity bits are redundant if the code word contains no errors. Due to the framed relationship between data bits and parity bits of the valid codewords, the parity bits can be easily computed, and hence the information represented by 'n' bits can be represented by 'd' bits. By removing these unnecessary bits, it is possible to produce the optimal (i.e., shortest length) representation of the image data. This work proposes a digital image compression technique based on Hamming codes. Lossless and near-lossless compression depending upon need can be achieved using several code specifications as mentioned here. The achieved compression ratio, computational cost, and time complexity of the suggested approach with various specifications are evaluated and compared, along with the quality of decompressed images.

KEYWORDS

Hamming code, Parity, Lossless Compression, Near Lossless Compression, Compression Ratio.

https://doi.org/10.17993/3ctic.2022.112.225-237

1. INTRODUCTION

Image coding and compression is used mainly for effective data storage and transmission over a network and in some cases for encryption. Image data is also coded for achieving compression to optimize the use of these resources. In digital image compression, depending upon the quality of decompressed image the compression algorithms employed are categorised in the categories like Lossless compression, Lossy compression, and Near-lossless compression. The data redundancy is a statistically quantifiable entity, it can be defined as R_D=1-1⁄CR, where the CR is the compression ratio represents the ratio of number of bits in compressed representation to number of bits in original representation. A compression ratio of 'C' (or 'C':1) means that the original data contains 'C' information bits for every 1 bit in the compressed data. The associated redundancy of 0.5 indicates that 50% of the data in the first data set is redundant. Three primary data redundancies can be found and used in digital image compression i.e. coding redundancy, interpixel redundancy, and psychovisual redundancy. When one or more of these redundancies are considered as a key component for reduced representation of data and accordingly these redundancies are encoded with some method the compression is achieved, Sayood (2017). There is no right or wrong decision when deciding between lossless and lossy image compression techniques. Depending on what suits your application the most, you can choose. Lossy compression is a fantastic option if you don't mind sacrificing image quality in exchange for smaller image sizes. However, if you want to compress photographs without sacrificing their quality or visual appeal, you must choose lossless compression Kumar and Chandana (2009). Based on a knowledge of visual perception, the irrelevant part of the data may be neglected, a lossy compression, includes a process for averaging or eliminating this unimportant information to reduce data size. In lossy compression the image quality is compromised but a significant amount of compression is possible. When the quality of decompressed image and integrity are crucial then lossy compression shouldn't be employed Ndjiki-Nya et.al (2007). Not all images react the same way to lossy compression. Due to the constantly changing nature of photographic images, some visual elements, including slight tone variations, may result in artefacts (unintended visual effects), but these effects may largely go unnoticed. While in line graphics or text in document images will more obviously show the lossy compression artefacts than other types of images. These may build up over generations, particularly if various compression algorithms are employed, so artefacts that were undetectable in one algorithm may turn out to be significant in another. So, in this scenario one should try to bridge the consequences of lossless and lossy compression algorithms. So, the near-lossless compression algorithm should be practiced in case of document images to optimise the compression ratio and the quality of reconstruction Ansari et al. (1998). One of the very famous Error detection and correction technique used in channel coding may be used for the digital image compression Caire et.al (2004) and Hu et.al (2000). In this paper use of Hamming codes with different specifications for various compression algorithms mentioned above is done and the compression is achieved.

2. HAMMING CODES

When the channel is noisy or error-prone, the channel encoder and decoder are crucial to the overall encoding-decoding process. By adding a predetermined amount of redundancy to the source encoded data, it is possible to minimize the influence of channel noise. As the output of the source encoder is highly sensitive to transmission noise, it is based on the principle that enough controlled redundant bits must be added to the data being encoded to guarantee that a specific minimum number of bits will change during the transmission. Hamming demonstrated, showed that all single-bit errors can be detected and corrected if 3 bits of redundancy are added to a 4-bit code word, providing the Hamming distance between any two valid code words 3 bits. The 7-bit Hamming (7, 4) code word P1, P2, D3, P4, D5, D6, D7 for a 4-bit binary number b1,b2,b3,b4≅D3,D5,D6,D7 padded with parity bits p1,p2,p3≅P1,P2,P4 .

https://doi.org/10.17993/3ctic.2022.112.225-237

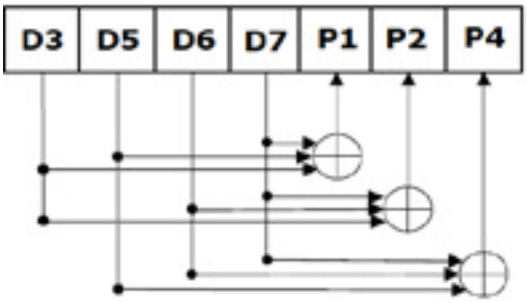

Fig. 1. Hamming Code

2.1. ERROR DETECTION

One of the most popular hamming code specifications used is (7,4); here 7 is total length of codeword and 4 is number of data bits.

Here (n-k) i.e., 7-4 =3 parity bits are used to encode 4-bit message to detect one bit error and correct one bit error.

In the following example, the first step of algorithm is to identify the position of the data bits and parity bits. All the bit positions at powers of 2 are marked as parity bits (e.g., 1, 2, 4, 8). Given below is the structure of 7-bit hamming code.

Here a data “0 1 0 1” is encoded using (7,4) even parity hamming code and an error is introduced in the fifth (D5) bit.

P1	P2	D3	P4	D5	D6	D7
1	1	0	1	1	0	1

Three parity checks and needed to be considered to determine whether there are any errors in the received hamming code.

Check 1: Here the parity of bits at position 1,3,5,7 should be verified

P1	D3	D5	D7
1	0	1	1

It is observed that the parity of above codeword is odd, it is concluded that the error is present and check1 is failed.

Check 2: Here the parity of bits at position 2,3,6,7 should be verified

P2	D3	D6	D7
1	0	0	1

It is observed that the parity of above codeword is even, then we will conclude the check2 is passed.

Check 3: Here the parity of bits at position 4,5,6,7 should be verified

P4	D5	D6	D7
1	1	0	1

https://doi.org/10.17993/3ctic.2022.112.225-237

It is observed that the parity of above codeword is odd, it is concluded that the error is present and check3 is failed.

So, from the above parity analysis, check1 and check3 are failed so we can clearly say that the received hamming code has errors.

2.2. ERROR CORRECTION

They must be repaired because it was discovered that the received code contains an error. Use the next few steps to fix the mistakes:

The error checks word has become:

check1	check2	check3
1	0	1

The error is in the fifth data bit, according to the decimal value of this error-checking word, which was calculated as "1 0 1" (the binary representation of 5). Simply invert the fifth data bit to make it correct.

So, the correct data will be:

P1	P2	D3	P4	D5	D6	D7
1	1	0	1	0	0	1

In the same fashion other specifications of code like (3,1), (6,3), (7,3), and so on can be implemented and error detection and error correction of messages using the parity bits can be accomplished. There has never been a way for error checking and correcting that is more effective than Hamming codes, so, it is still widely used channel encoding. It offers an effective balance between error detection and correction, in addition to that one may verify its application in data compression field, which is being discussed in the next topic.

3. PROPOSED WORK

The document image compression, if various logical regions of the document image are segmented and appropriate compression algorithm is used for those regions, then the trade-off between compression ratio and the quality of decompressed image may be resolved a bit. Lossy compression techniques for photographs, Near-lossless compression for Figures, Tables and other text-like important regions, and Lossless compression for the Text contents Shanmugasundaram *et.al* (2011).

3.1. THE EXISTING LOSSLESS ENCODING ALGORITHM

i. Reshaping: 2D to 1D conversion of image data

ii. Resizing: Divide the data in terms of 7-bit chunks

iii. Checking Hamming encodability of newly formed chunks; valid and invalid hamming codes (7,4)

iv. Lossless Encoding of data

v. beginning with ‘0’+codeword (7 bit), if the chunk is invalid codeword

vi. beginning with ‘1’+encoded codeword (4 bit), if the chunk is valid codeword

vii. If the size of the coded data is smaller than the input data, then compression is achieved.

The following algorithms are the step by step modification in the existing work of Hu *et.al* (2000). Here, the near-lossless encoding algorithm based on the idea of bit plane slicing is presented as Algorithm 1. It is presumptive that the least significant bit (LSB) in an image contains the least

https://doi.org/10.17993/3ctic.2022.112.225-237

significant information, and that if this information is removed from a gray-level image, a little visual degradation will be caused.

3.2. ALGORITHM 1: BIT PLANE SLICING BASED LOSSY ENCODING

i. Perform bit-plane slicing: remove LSB (assuming that LSB plane consists of least information)
ii. Reshaping: 2D to 1D conversion of image data
iii. Resizing: Divide the data in terms of 7-bit chunks
iv. Checking Hamming encodability of new chunks; valid and invalid hamming codes (7,4)
v. Lossless Encoding of data
vi. beginning with '0'+codeword (7 bit), if the chunk is invalid codeword
vii. beginning with '1'+encoded codeword (4 bit), if the chunk is valid codeword
viii. If the size of the coded data is smaller than the input data, then compression is achieved.

The second approach, which also uses (8,4) Hamming codes for lossless compression, has the additional capability of detecting and correcting error in the eighth bit, which is set or reset based on the even or odd parity of the entire 7-bit codeword in (7,4) variant. This technique offers the lossless compression in addition to the additional 1-bit detection and correction capabilities.

3.3. ALGORITHM 2: LOSSLESS ENCODING USING (8,4) HAMMING CODE SPECIFICATION

i. Reshaping: 2D to 1D conversion of grey image data
ii. Checking Hamming encodability of newly formed chunks; valid and invalid hamming codes (8,4)
iii. Lossless Encoding of data
iv. beginning with '0'+codeword (8 bit), if the chunk is invalid codeword
v. beginning with '1'+encoded codeword (4 bit), if the chunk is valid codeword
vi. If the size of the coded data is smaller than the input data, then compression is achieved.

It is computationally expensive to scan the complete image to determine if the codewords (its grey levels) are valid or invalid hamming codewords. Therefore, a novel technique for eliminating this avoidable routine is being tested, which involves discovering the valid codewords beforehand and simply classifying image gray-levels as valid or invalid codewords by comparing with them. This process is similar to quantization but here the quantization levels are neither equidistant nor generated by some programming language function. Even though this quantization inevitably results in information loss, the compression ratio will be improved Li *et.al* (2002). These quantized gray-levels offers spectral compression at the same time the probabilities of the resultant gray-levels also get changed. The image with changed gray-level probabilities is further feed to probability-based coding like Huffman's coding and added compression is achieved Jasmi *et.al* (2015) and Huffman (1952). The algorithms 3 is subsequently modified in algorithm 4 and 5 using (8,4) code for lossless compression with 16 and 32 quantization levels respectively. After the successful compression and decompression, the quality of decompressed image using quantization approach is computed based on the parameters like Correlation of input output images Dhawan (2011), its Mean Squared Error (MSE), Signal to Noise Ratio (SNR), Structural Similarity Index Metric (SSIM) to compute the retainment of structural properties of the input images, Compression Ratio (CR) achieved and the Computational Time (CT).

3.4. ALGORITHM 3: FAST NEAR-LOSSLESS ENCODING USING (7,4) HAMMING CODE SPECIFICATION AND QUANTIZATION

i. Reshaping: 2D to 1D conversion of image data
ii. Perform bit-plane slicing: remove LSB (considering that it LSB plane consists of the least information) OR
Resizing: Divide the data in terms of 7-bit chunks

https://doi.org/10.17993/3ctic.2022.112.225-237

iii. Identify the ‘16’ valid codewords and enlist them in an array valcod
iv. valcod = [0 15 22 25 37 42 51 60 67 76 85 90 102 105 112 127]
v. Lossy step: Quantize the entire image pixels to grey levels in valcod variables
vi. If size of coded data is smaller than input data, then compression is achieved.

3.5. ALGORITHM 4: FAST NEAR-LOSSLESS ENCODING USING (8,4) HAMMING CODE SPECIFICATION AND 16 LEVEL QUANTIZATION

i. Reshaping: 2D to 1D conversion of image data
ii. Identify the ‘16’ valid codewords and enlist them in an array valcod16
iii. Valcod8 = [0 15 51 60 85 90 102 105 150 153 165 170 195 204 240 255]
iv. Lossy step: Quantize the entire image pixels to grey levels in valcod16 variables
v. If size of coded data is smaller than input data, then compression is achieved.

3.6. ALGORITHM 5: FAST NEAR-LOSSLESS ENCODING USING (8,4) HAMMING CODE SPECIFICATION AND 32 LEVEL QUANTIZATION

i. Reshaping: 2D to 1D conversion of image data
ii. Identify the ‘total 32’ valid even, odd conjugate codewords along-with its complements and enlist them in an array valcod32
iii. valcodeoc = [0 15 23 24 36 43 51 60 66 77 85 90 102 105 119 129 136 142 150 153 165 170 178 189 195 204 212 219 231 232 240 255]
iv. Lossy step: Quantize the entire image pixels using the 32 grey-levels in valcod32 variable
v. If the size of the coded data is smaller than the input data, then compression is achieved.

4. RESULTS

The experimentation carried out on a dataset consisting of three classes of images like graphs, diagrams, and equations, which are the text-like regions in the document image. As discussed above the existing Hamming code-based algorithms is implemented and executed over the dataset. Performance of these algorithms is evaluated by means of the performance metrics like compression ratio and computation time. The algorithms are implemented using MATLABR2020b software on a 64-bit, 2.11 GHz processor with 8 GB RAM computer system. The algorithm offers compression generally but, in some times, (4/30) it failed to achieve the compression. The average compression ratio achieved is 1.2 : 1, which is not significant considering the computation cost of the algorithm. The regular scanned document takes computation time up to 10 min. So, the images in the dataset are resized to 512 X 512 pixel size and further computation time is observed. For this size of images, the computation time results 21.30 sec./ image. This time is also higher considering the size of image. To improve the CR and minimizing the computation cost the bit-plane slicing based lossy compression algorithm (Algorithm 1) is proposed, it has enhanced the CR a bit and the CT is also halved. Further to avoid information loss a (8,4) and need of image resizing, the Hamming code-based algorithm (Algorithm 2) is tested. It further enhanced the CR but the CT once again increased up to original algorithm. This performance of mentioned algorithms is presented using Table 1 below.

Table 1. Comparison of purely Hamming code-based algorithms based on CR and CT.

Sr. No.	Algorithm	CR (CR:1)	CT (sec/ image)
1	Algorithm 0	1.20	21.30
2	Algorithm 1	1.24	10.32
3	Algorithm 2	1.29	19.15

https://doi.org/10.17993/3ctic.2022.112.225-237

As both the parameters CR and CT offered by these algorithms are not attractive, the quantization-based algorithms (Algorithm 3, Algorithm 4, and Algorithm 5) are proposed, avoiding traversing throughout the image for checking the validity of codewords. These algorithms have achieved the significant CR along with 100 times less CT. The quality of decompressed image using quantization approaches is computed based on the parameters like Correlation of input output images, its Mean Squared Error, Signal to Noise Ratio, Structural Similarity Index Metric to compute the retainment of structural properties of the images, Compression Ratio achieved and the Computational Time. The results are presented in Table 2.

Table 2. Comparison of quantization-based algorithms.

	Algorithm 3	Algorithm 4	Algorithm 5
Correlation (0-1)	0.9420	0.9629	0.9658
MSE (bits/image)	11.22	37.80	12.13
PSNR (Ratio)	39.19	33.22	39.17
SSIM (0-1)	0.9295	0.9308	0.9519
CR (CR:1)	3.04	3.31	3.31
CT (sec/image)	0.2575	0.1132	0.1314

It can be observed from the above table that Algorithm 5 performs better than others considering the mentioned parameters except MSE and CT; the MSE of algorithm 3, that is less 1 bit/image lesser than Algorithm 5 and CT of Algorithm 4 is least, that is about 0.02 sec lesser than Algorithm 5. The Correlation and SSIM offered by this algorithm are highest, which is very significant while proposing the near-lossless algorithm. The performance of quantization-based approach computed over mentioned three classes using the image quality metrics discussed is presented below. The average values for these classes are shown to get overall idea about performance of algorithm.

4.1. CORRELATION

The 2D correlation between the input image and the decompressed result is calculated, it ranges from 0-1, where value '1' represent that both the images are identical. The values near to '1' signifies the near-lossless compression. The performance is displayed using Fig. 2.

https://doi.org/10.17993/3ctic.2022.112.225-237

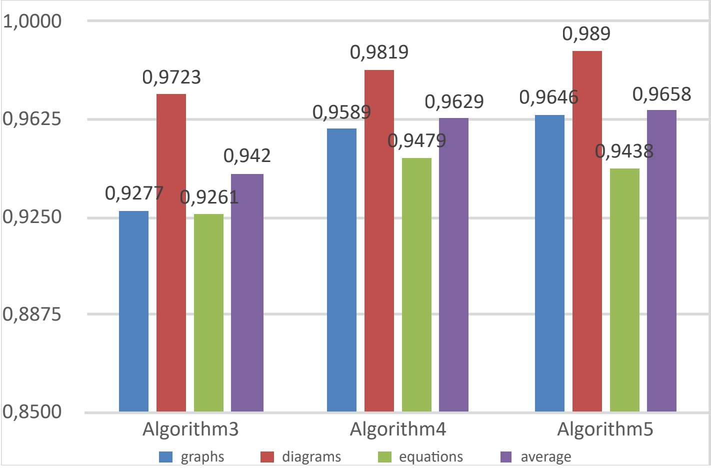

Fig. 2. Comparison of algorithms based on Correlation.

4.2. MSE

The most used estimator of image quality is MSE refers to the metric giving cumulative squared error between the original and compressed image. Near zero values are desirable. The performance is displayed using Fig. 3.

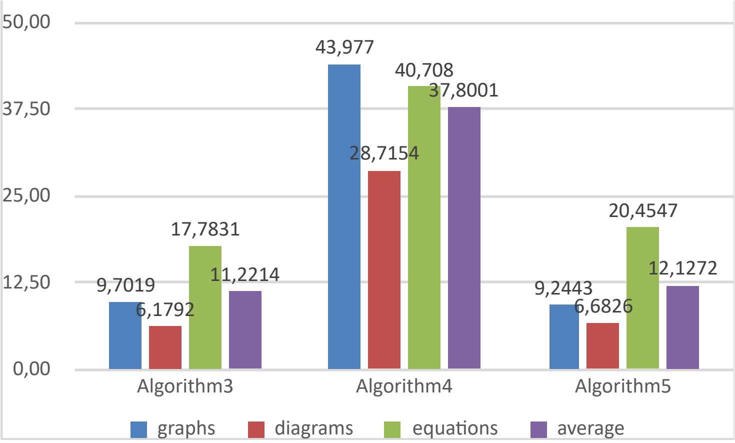

Fig. 3. Comparison of algorithms based on MSE.

4.3. PSNR

The Peak Signal to Noise Ratio is positive indicator of performance of compression algorithms. The performance is displayed using Fig. 4, here the Algorithm 5 offers the highest PSNR value, which is a good indicator of its applicability.

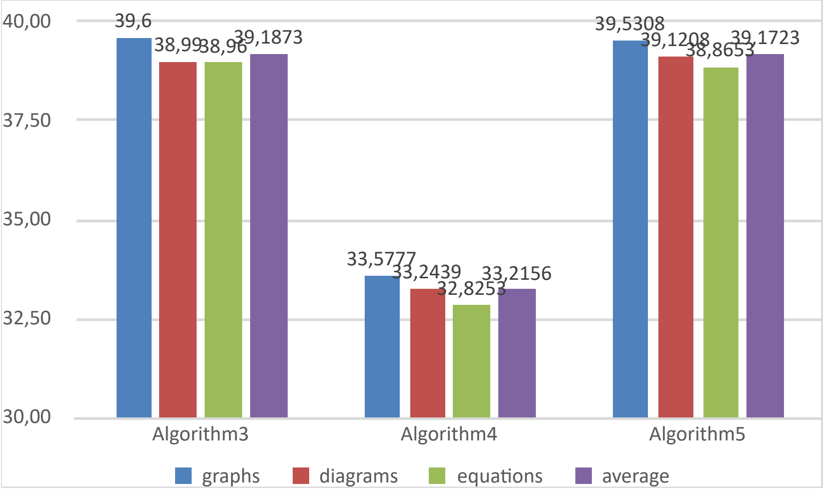

Fig. 4. Comparison of algorithms based on SNR.

4.4. SSIM

The Structural Similarity Index Measure (SSIM), a perception-based measure, considers image degradation as a perceived change in structural information. These techniques differ from others like Mean Squared Error (MSE) and Signal to Noise Ratio that include evaluate absolute errors (SNR). According to the theory behind structural information, pixels are highly interdependent, when they are spatially close to one another. These dependencies carry important details about how the elements in an image are arranged. In our experimentation Algorithm 5 has the highest SSIM, which is another good indicator of the quality of a compression algorithm. The performance is displayed using Fig. 5.

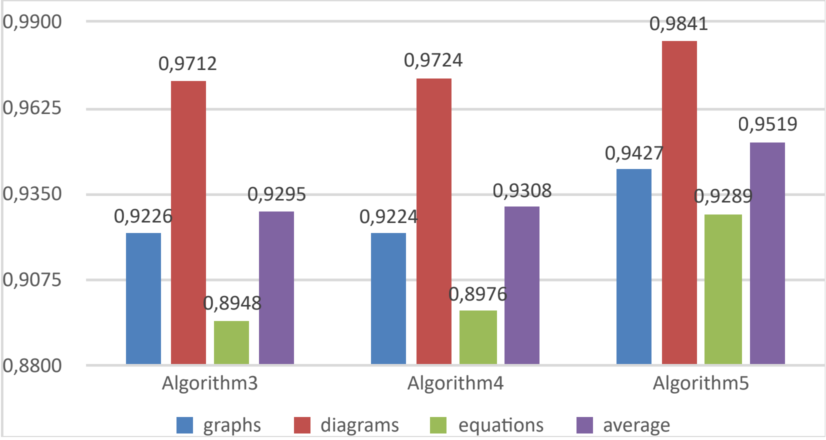

Fig. 5. Comparison of algorithms based on SSIM.

4.5. COMPRESSION RATIO

Here, the average compression ratio for the considered images, ranges from 3.04 : 1 to 3.31 : 1, algorithm 5 having the highest value and algorithm 3 having the lowest. The compression ratio is a useful indicator of decompression effectiveness, the higher value indicates a better algorithm. The performance is displayed using Fig.6.

https://doi.org/10.17993/3ctic.2022.112.225-237

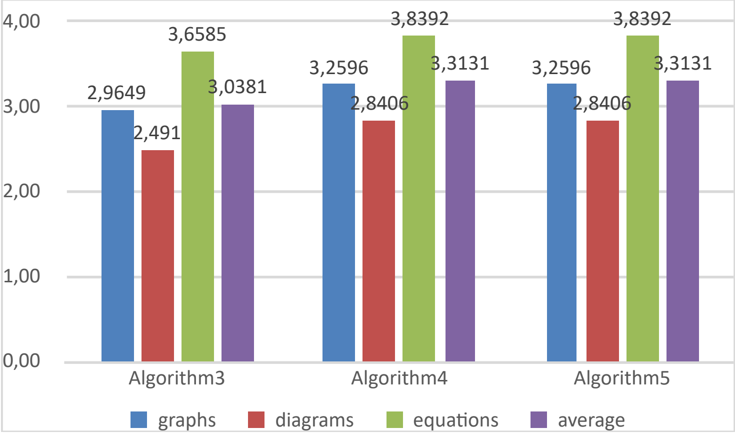

Fig. 6. Comparison based on Compression Ratio.

4.6. COMPUTATION TIME

Finally, computational time is used to study the time complexity of the algorithm; in this case, Algorithm 4 tends to be the most efficient and requires the least amount of time. The performance is displayed using Fig. 7.

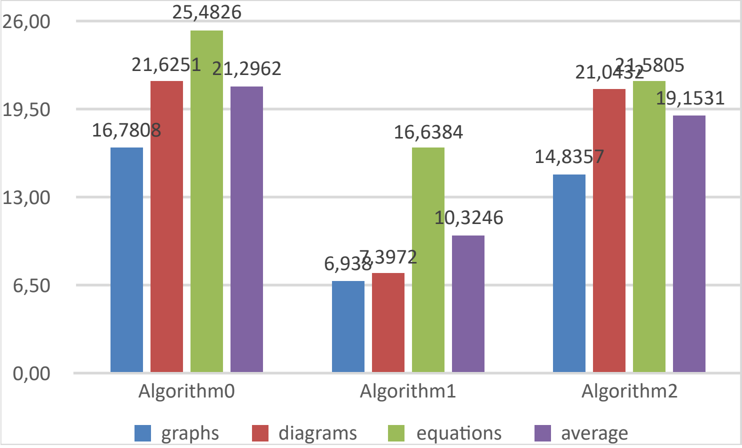

Fig. 7. Comparison based on Computation Time.

Algorithm 5 having highest CR, executes faster and performs better in terms of quality of decompressed image. So, one can conclude that the Algorithm 5 tends to the most optimized algorithm for Near-lossless compression of Document Images.

5. CONCLUSIONS

Compression offers an attractive option for efficiently storing large amounts of data. Document Images are multi-tone images, separate algorithm for different regions may improve the compression outcomes. Lossy algorithms are best for regions like photographs, lossless are necessary in case of sensitive regions like text and the non-textual but text consisting important regions like figures, plots and equations needed to be treated with near-lossless techniques. They are advantageous for maintaining the trade-off between compression ratio and quality of reconstructed image. In this work the channel coding algorithm like Huffman codes is implemented with its different specifications, the work is further optimized based on time complexity and quality of decompressed image. Considering the higher Correlation, PSNR and SSIM values obtained, one can understand that the quality of the decompressed images is really good, and the algorithms can be considered as near-lossless algorithms. It is also concluded that the (8, 4) Hamming code with even and odd conjugate complementary codes and based 32 quantization levels works best amongst the other discussed.

https://doi.org/10.17993/3ctic.2022.112.225-237

6. FUTURE SCOPE

After observing the performance of various hamming code-based algorithms, one may be encouraged to practice them and do some modification. So, the first thing that may be tried is to make use of (3,1) and (4,1) specifications of Hamming code. Further, after the quantization to newly achieved gray levels is done, one can implement probability-based coding scheme like Huffman or Arithmetic codes to represent the data more effectively.

REFERENCES

[1] Ansari, Rashid, Nasir D. Memon, and Ersan Ceran. "Near-lossless image compression techniques." Journal of Electronic Imaging 7, no. 3 (1998): 486-494.

[2] Caire, Giuseppe, Shlomo Shamai, and Sergio Verdú. "Noiseless data compression with low-density parity-check codes", DIMACS Series in Discrete Mathematics and Theoretical Computer Science 66, 263-284, (2004).

[3] Hu, Yu-Chen, and Chin-Chen Chang, "A new lossless compression scheme based on Huffman coding scheme for image compression", Signal Processing: Image Communication 16.4, 367-372, (2000).

[4] Sampath Kumar, Chandana, "Comparative Study of Lossless Compression Scheme Based on Huffman Coding for Medical Images", Recent Trends in Science & Technology, (2009).

[5] Shanmugasundaram, Senthil, and Robert Lourdusamy. "A comparative study of text compression algorithms." International Journal of Wisdom Based Computing, no. 3,68-76 (2011).

[6] Li, Robert Y., Jung Kim, and N. Al-Shamakhi. "Image compression using transformed vector quantization." Image and Vision Computing 20, no. 1 (2002): 37-45.

[7] Dhawan, Sachin. "A review of image compression and comparison of its algorithms." International Journal of Electronics & Communication Technology, IJECT 2, no. 1 (2011): 22-26.

[8] Jasmi, R. Praisline, B. Perumal, and M. Pallikonda Rajasekaran. "Comparison of image compression techniques using huffman coding, DWT and fractal algorithm." In 2015 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-5. IEEE, 2015.

[9] Pandya, M. K. "Data compression: efficiency of varied compression techniques." Formal Report, Brunel University (2000).

[10] Huffman, David A. "A method for the construction of minimum-redundancy codes." Proceedings of the IRE 40.9, 1098-1101, (1952).

[11] P. Ndjiki-Nya, M. Barrado and T. Wiegand, "Efficient Full-Reference Assessment of Image and Video Quality," 2007 IEEE International Conference on Image Processing, 125-128, (2007).

[12] Sayood, Khalid, "Introduction to data compression", 5th edn. Morgan Kaufmann, (2017).

AUTHORS BIOGRAPHY

Mr. Prashant Laxmanrao Paikrao is Ph.D. Research Scholar in Electronics & Telecommunication Engineering department of SGGSIE&T, Nanded (M.S.), India. He is an active member of IE, ISTE and IETE. His area of interest is Image Processing, Digital Logic Design, Fuzzy Logic.

https://doi.org/10.17993/3ctic.2022.112.225-237

Dr. Dharmapal Dronacharya Doye is Professor in Department of Electronics & Telecommunication Engineering of SGGSIE&T, Nanded (M.S.), India. He is an active member of IE, ISTE and IETE. His area of interest is Image Processing, Video Processing, Artificial Intelligence.

Dr. Milind Vithalrao Bhalerao is Assistant Professor and Head of Electronics & Telecommunication Engineering Department of SGGSIE&T, Nanded (M.S.), India. He is an active member of ISTE and IETE. His area of interest is Image Processing, Robotics, Artificial Intelligence.

Dr. Madhav Vithalrao Vaidya is Assistant Professor in Information Technology Department of SGGSIE&T, Nanded (M.S.), India. He is an active member of IEEE, ISTE and IETE. His area of interest is Data Mining, Image Processing, Pattern Recognition, Blockchain Technology, and Internet of Things.

https://doi.org/10.17993/3ctic.2022.112.225-237