A New Proposed Method For A Statistical Rules-Based Digital Image Compression

Khadhim Mahdi Hashi1, SALWA SHAKIR BAAWI 2, Bushra Kamil Hilal3

1Department of Computer Science, College of Education for Pure Science / University of Thi-Qar, Thi-Qar, Iraq
2Department of Computer Information Systems, College of Computer Science and Information Technology - University of Al-Qadisiyah, Diwaniyah, Iraq
3Department of Computer Science, College of Education for Pure Science / University of Thi-Qar, Diwaniyah, Iraq

1kadhimmehdi63@utq.edu.iq
2salwa.baawi@qu.edu.iq
3edpcompbushra@utq.edu.iq

Abstract—Image compression depends on data compression of digital images. Its central objective is to decrease the redundancy of the image data for reducing space and the cost of transmitting data in public communication channels. This research suggests a new compression technique based on the statistical rules. The proposed technique is one of the lossy compression techniques is based upon the statistics of the pixel values of the gray-scale image. The quality of these compressed images have been evaluated using some factors like the Image size before/after the compression process, Compression Ratio, (CR), and Peak Signal to Noise Ratio, (PSNR), Mean Square Error (MSE), and Mean Space Saving (MSS). Experimental results have demonstrated that the proposed technique provides sufficient higher compression with minimal to lose data.

Keywords—Image compression; statistical; compression ratio; Peak signal to noise ratio; lossy compression.

1. Introduction

The reason for increasing data storage and great rate transmission due to recent developments in the area of information technology that have contributed to the genesis of a massive amount of information every moment [1]. For instance, text, image, audio, video data is mostly passed from person to person or from place to place in a digital form [2]. It is often desirable to store data or transmits it for saving disk space, reducing the time needed for communication or the time needed for data transfer, and more. Data to be handled as well as software has been expanding, and the amount of information communicated between systems has also been continually increasing. Therefore, data compression technology is considered an important field for a variety of computing applications [3], [4].
Data compression refers to the art or science of information representation in a compact form. Compression is the process of eliminating data redundancy or converting data into a form that occupies less storage space [5]. Also, reducing the size of packets required to transfer data over digital communication networks. Thus, cuts down the time needed to transmit data while cutting down the occurrence of faults during the transport process [4], [6]. In the new domain, it is possible to filter out relevant information and discard information that is irrelevant or of lesser importance for image quality [2]. Although, the image is very fundamental and commonly used on the Internet, sometimes it tends to be big [7]. This is why image compression is so important.

The primary aim of data compression is intended to convert the entered data into other data of a smaller size to take up the least possible storage space while preserving the necessary information as possible. The general framework for image compression comprises two processes, namely, compression and decompression, as presented in Figure 1. It has been previously applied in [8].

![Figure 1. General information-theoretic framework for image compression](image)

The left side of Figure 1, presents the sender who compresses the data of the image file and then storage or transmits the modified “compressed image” to the receiver on the right side via the communication channel. This compression model is structured. In order to display an image in a reasonable amount of time, techniques must be incorporated or improved to reduce the image’s file size. The right side in Figure 1, inverses works to the sender side to recover as possible as the original image as. As noticed in Figure 1, the source file that is used as an input to the compression process is named the original (uncompressed) file. While the obtained file from the completed compression process is named the compressed file, which is used to restore the data to its original position again.

Data compression techniques can be classified into different categories [1]. These categories are based on the data type used, data quality that categorized into lossless compression and lossy compression and according to the application employed, as well as, the coding scheme applied are shown in Figure 2.

Many works have classified these techniques according to data types such as text (Jayapandiyan et al 2020), image (Agrwal et al. 2016), audio (Ali et al. 2019), or video (Shi and Sun 2017) as seen in [9]–[12]. While the techniques which are presented by (Helminger et al. 2020; Saidani et al. 2019) classified these techniques into lossless and lossy compression according to quality [13], [14]. Lossy compression is the kind of
encoding data technique that are utilizing estimates that inaccurate for describing the content. These methods are applying to decrease the size of data for treatment, storehouse, and transfer content. While lossless compression is the kind for compressing data algorithms that are permitting the original data to be ideally restored from the data is compressed [15]. In addition, other works offered by (Markas and Reif 1993; Aldemir et al. 2019; Saidani et al. 2019) have classified the technique according to application such as [14], [16], [17]. Finally, several works have done on different coding schemes. For instance, in Huffman coding [10], [18]–[22], introduced by (Agrwal et al. 2016; Hasan 2011; Anantha Babu 2016; Abdouleh 2012; Chukwuchekwu 2019; Nashat and Hassan 2016).

Several essential metrics are used for evaluating the quality of the algorithm used for data compression. The metrics can be described by the compression ratio, peak signal to noise ratio, mean square error, and mean space-saving. In addition, the time of the compression process and the time of the decompression process, for more details see in section 4. The main contribution of this research is to present a new method by statistical rule-based digital gray-scale image compression. The proposed method is a lossy compression through employing some statistical rules such as average filter, weighted average filter, and histogram-variance filter.

The remaining of the article is organized as follows. The related work has been discussed in Section 2. Section 3 offers the methodology of the proposed model followed by the evaluation criteria in section 4. The experimental result and discussion in Section 5. Section 6 offerings the conclusion followed by references.
2. Related work

Many studies have been applied in the area of lossy image compression. A compression system must satisfy several requirements, including a higher compression ratio CR, image quality PSNR, and shorter time of encoding (compression) process. Therefore, many studies have devised methods that can fulfill all these requirements. Lossy compression can be included in three methods which are Fractal Transform (FCT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT) [23].

In [24], Veenadevi and Ananth (2011) have presented techniques of the fractal coding applied for the compression of satellite imageries. The compression in this study focused on three types of images namely standard Lena image, Satellite Rural image and Satellite Urban image. The experimental results showed that the compression ratio (CR) value of (3.2) and Peak Signal to Noise Ratio (PSNR) value of (11.91 dB).

Meanwhile, Zhang et al. (2014) have introduced a novel scheme of compressing encrypted images with the assistance of auxiliary information. In the compression phase, the original uncompressed images are the encrypted data in various DCT sub-bands are effectively compressed by using a quantization mechanism without revealing the original content, and an optimization method with ratio-distortion criteria is employed to select the quantization parameters according to the auxiliary information. The experimental results proved that the compression ratio (CR) value of (0.172) and peak signal to noise Ratio (PSNR) is (35.5 dB), and the mean square error is (18.24) [25].

In order to reduce the encoding time and get a better size and quality of images. Menassel et al. (2020) [26] and Menassel (2020) [27], have proposed new studies that improve the fractal image compression (FIC) method, and to overcome the long encoding time problem. These studies depended on the improving the performance of FIC by using the wolf pack algorithm and the bat-inspired algorithm (BAT). The proposed methods that use the application of metaheuristics have evaluated under several aspects which are encoding time, compression ratio (CR), peak signal to noise ratio (PSNR), and mean square error (MSE). The Results of this method not only provides improved coding time but also obtains better results in both file size and image quality.

3. Proposed Methodology

The proposed methodology consists of two parts, namely, the compression (encoding) and the decompression (decoding) processes, are provided in the following sections:

A. Compression (Encoding) Part

The workflow of the proposed compression process is shown in Figure 3. The original (Color/Grayscale) image data is the input of the proposed technique and the compressed image is the expected output.

As shown in Figure 3 above, the source image can be grayscale or color image. In this case, the (input) source image is a color image with any color, convert the source image from color image to a grayscale image. Six stages which depict the proposed compression technique:

1) Pre-processing stage: In this stage, one-dimensional sequences of pixels are obtained by path scanning; two-dimensional transformed grayscale image is converted into a one-dimensional sequence of pixels by using the zigzag method.
2) **Statistical method**: Applying the statistical rules-based on the pixels of the image using the proposed statistical method as shown in Figure 4 that depends upon using the statistical average filters in order to perform the first round of the compression process.

**Algorithm 1: Statistical method**

| Input: Variance filter value and array of one-dimension pixels for density for a grayscale image P\(p_0, \ldots, p_{n-1}\). |
| Output: V which is represented the vector pixels values after applying variance filter (first round of compression) and R vector of pixels frequencies. |

| Steps: |
|---------------------------|
| 1. Read Variance filter value (Vari); |
| 2. For each pixel \((p_i)\) in vector P |
| 3. Let \(\text{sum} = p_1, r = 0, j = i + 1\); |
| 4. While \((p_i - p_j) \leq \pm\text{Vari}\) |
| 4.1 \(\text{sum} = \text{sum} + p_j\); |
| 4.2 \(j = j + 1\); |
| 4.3 \(r = r + 1\); |
| 4.4 End while |
| 5. If \(\text{sum} \neq p_i\) then Apply the median filter on \((\text{sum} \text{ and } r)\); **End then**; |
| 6. If \(\text{sum} = p_i\) then Let median filter = \(p_i, r = 1\); **End then**; |
| 7. Add the median filter value to vector V; |
| 8. Add the r value to vector R; |
| 9. End for |
| 10. Print the vector V and vector R; |
| 11. End. |

Figure 4 Proposed Statistical Method
3) Compressing one’s values stage: The input of this stage is the output of the previous stage (statistical method vector R). The primary aim of this stage is to compress the frequencies that own a "1" value. Figure 5 presents the algorithm that illustrates the important steps of this stage.

**Algorithm 2: Compressing One’s values**

**Input:** Vector R (output of statistical method algorithm 1)

**Output:** Constructing vector R1, which is a vector of representing the second round of compression grayscale image.

**Steps:**
1. Read the vector R
2. Let Freq_one=1; // Frequencies of one’s value
3. Let i=2; // index for R vector
4. Let j=2; // index for R1 vector
5. N= length (R);
6. For i=1 to N
   7. While (( R[i]= 1) & R(i-1)=1) do
      8. Freq_one=Freq_one + 1;
      9. i=i + 1;
   10. End while
11. If Freq_one >1 then R1[j]=0, j=j+1,R1[j]=Freq_one, j=j+1, Freq_one=1;
12. End then;
13. Else
14. R1[j]=R[i-1];
15. j=j+1;
16. End Else
17. End while
18. End for
19. End.

During this stage computes the size of the compressing by counting the different size of the image before and after the second round of compression.

4) Replacing Maximum values stage: After constructing the vector (R1) in second round of compression will be expected R2 as output that represents the process of replacing maximum values by unused values in vector R1. The vector (R2) represents the third round of the grayscale image compression. Figure 6 explains the workflow of this stage.

**Algorithm 3: Replacing maximum values into unused values**

**Input:** Vector R1 (output of algorithm 2)

**Output:**
- Constructing vector R2, which is a decimal vector representing the third round of compression image (after the replacement process).
- Constructing vector U, which contains the unused values vector that reconstructed by maximum values.
- Constructing vector M owns maximum values that are exchanged with unused values.

**Steps:**
1. Read the vector R1
2. Let D= R1;
3. Let H = histogram (D); // Find the histogram
4. Let L= length (H); // represent the number of levels used.
5. Max_no_bits = ceil (\( \log_2(L) \));  // denotes the number of bits required to represent data.
6. Max_value = \( 2^{\text{Max_no_bits}} - 1 \);  // The biggest value that can be signified by using Max_no_bits.
7. Let \( j = 1 \);
8. For \( i = 0 \) to Max_value
9. While \( (i \text{ not belong to D}) \) do
10. \( U[j] = i \);
11. \( j = j + 1 \);
12. End while
13. End for
14. Let \( k = 1 \);
15. If \( (U = [ ] \text{ or Max(D)} \leq \text{Max_value}) \) then \( D1 = R1, \ M = [ ], U1 = [ ]; \) End then
16. Else
17. While \( (\text{Max(D)} > \text{Max_value}) \text{ & } (k \leq \text{length(U)}) \)
18. \( M[k] = \text{Max(D)}; \)
19. \( U1[k] = U[k]; \)
20. \( D[D = \text{Max(D)}] = U[k]; \)  // replace Max_value of D with unused value.
21. \( k = k + 1; \)
22. End while;
23. End if;
24. \( D1 = D; \)
25. Computing the size inputs and outputs
26. If size(output) > size(input) then \( D1 = R1, M = [ ], U1 = [ ]; \) End if
27. End.

Figure 6 Replacing maximum values into unused values

5) Reducing levels Stage: This stage represents the fourth round of compression process. This process can be performed by filtering the indexes in R2 vector that have larger numbers and ownered a low repetitions as seen in Figure 7 and Figure 8 that use for minimize one level algorithm. This stage aims to reduce the levels to half.

Algorithm 4: Reducing levels
Input: Vector (R2)
Output: vectors (R3, yy1, n1) after minimize levels
Y, y1 // indexes array
Steps:
1. Get \( F = \text{max_bits (R2)}; \)
2. Repeat
3. Call Minimize one level on (R2) (Figure 8) and saving their indexes on y and y1.
4. If \( (F > 1) \) then
5. compute size of the inputs and outputs
6. if (output size < input size)
7. \( R2 = yy1, n1; \)
8. Unit (output size > input size)
9. \( R2, yy1, n1 = R2 \)
10. Else
11. Let yy1 is empty matrix;
12. Let n1 is empty matrix;
13. End if
14. End.

Figure 7 Reducing levels algorithm
Algorithm 5: Minimize one level

**Input:** Vector (R2)

**Output:** vector R3 after minimize one level, vector yy1 for the indexes, n1 represents length of yy1, x,k and y2 represents indexes

**Steps:**
1. Get F= max_bits (R2);
2. If ( F>=1) then
3. x=find (R2> floor (F/2));
4. k=delta (x);
5. Y2=compressions ones (k);
6. R2 (R2>floor (F/2) = R2(R2> floor (f/2)- ceil (F/2));
7. Store the indexes of max_replace (y2) in yy1, and store the length of it in n1;
8. End then
9. Else
10. Let yy1is empty;
11. n1 is empty;
12. End.

Figure. 8 Minimize one level algorithm

6) **Constructed Compressed Image:** This is the last stage in the compression process and will be obtained the compressed image based on applying some instruction on the output of the fifth stage.

B. Decompression (Decoding) Process

The constructed compressed image that produced as an output from the last stage of the compressed part. Afterward, the compressed image is transferred to the receiver’s site through a public communication channel. On this side, the input image is recovered by reading the compressed image and applying the decompression algorithm based on statistical rules-based. The decompression part inverses the compression stages completely.

4. **Evaluation Criteria**

Image compression is a range of image processes that means lessening the size of the image file without influence the quality of the image. Criteria (factors) for measuring the strength and weakness of the data compression algorithms are used. These factors are used to develop the compression system. Therefore, the relative importance of each factor depends on its application. Suppose that I1 and I2 represent the image before and after compression respectively. These factors are described below:

A. **Mean Square Error**

The Mean Square Error (MSE) is the cumulative difference between the compressed image and original image. Equation (1) is a formula to compute MSE [28]:

$$MSE = \frac{\sum_{m,n}[I_2(m,n) - I_1(m,n)]^2}{mn}$$  \hspace{1cm} (1)

Where, m and n are pixel co-ordinate $I_2$ – compressed image pixel, while $I_1$ represent the original image pixel. The mean square error (MSE) is calculated by obtaining the average square error. Then its result will be used to calculate PSNR to evaluate the resolution and quality of the compressed image as seen the following subsection.
C. **Peak Signal-to-Noise Ratio**

The peak signal-to-noise ratio (PSNR) is an expression between the largest possible power of the signal and distorting noise that affects its representation quality. Commonly speaking, the higher the bit-rate (i.e., more bits per value) in compressed storage, the higher the quality (i.e., higher PSNR) of the reconstructed image after decompression. Equation (2) represents the formula of PSNR [29]:

\[
PSNR = \frac{10 \log_{10}(\text{intensity}(\text{max}))^2}{\text{MSE}}
\]

For (8 bits) pixel gray scale, the intensity (max) value is 255,

\[
PSNR = \frac{10 \log_{10}255^2}{\text{MSE}}
\]

D. **Time of Compression**

The running time of both compression and decompression are comparing to other compressors in order to evaluate the speed of the compression process. This result computes via (byte per second) [30].

E. **Compression Ratio**

The compression ratio (CR) means the amount of difference between the uncompressed and compressed image size. For example, if the compression ratio is 0.4, that means 40% of the image is occupied from the source image size [31]. The following formula can be considered to compute the compression ratio (4).

\[
CR = \frac{\text{Size of compressed file}}{\text{Size of uncompressed file}} \times 100
\]

Sometimes, this value is also called bpp (bit per pixel) with the image.

F. **Histogram Analysis**

One way to discover a good compression method is to analyze the histogram of all reconstructed images and then compare them with original. It represents the number of pixels that have colors in the images color space.

5. **Experimental Result and Discussion**

The suggested method for lossy image compression is implemented on MATLAB (R2018b) on Intel core i5 processor with 8 GB RAM. The simple’s input RGB and gray images are of different sizes. The decompression image quality through proposed method is measured by its PNSR, CR, MSE, and the time of each compression and decompression processes. The results of the experiments are presented in the following subsections:

The proposed method has been applied on 22 images with different resolutions and computed the image size of (original and compressed) and the percentage of CR, PSNR, and MSE. As well as, the time of compression and decompression process. The results have been summarized in Table 1, Table 2, and Table 3.
Table 1. Experimental results of proposed method (average filter) based on size image (original and Compressed), CR, PSNR, MSE, and the time of compression and decompression.

| Image No. | Image Size (KB) | CR Original | CR Compressed | PSNR Original | PSNR Compressed | MSE Original | MSE Compressed | Compression Time (second) | Decompression Time (second) |
|-----------|-----------------|-------------|---------------|---------------|-----------------|--------------|-----------------|---------------------------|-----------------------------|
| g1        | 386             | 71.5        | 5.412         | 38.97         | 8.225           | 0.5215       | 1.1584          |
| g2        | 137             | 7.49        | 18.39         | 54.01         | 0.258           | 0.0868       | 0.1882          |
| g3        | 134             | 10.23       | 13.1          | 42.25         | 3.865           | 0.1000       | 0.1487          |
| g4        | 133             | 10.18       | 13.11         | 43.90         | 2.647           | 0.0968       | 0.2048          |
| g5        | 199             | 32.5        | 6.12          | 39.99         | 6.513           | 0.1973       | 0.6311          |
| g6        | 410             | 60.86       | 6.74          | 41.01         | 5.142           | 0.3870       | 1.3450          |
| g7        | 650             | 90.04       | 7.228         | 39.82         | 6.765           | 0.6254       | 2.5803          |
| g8        | 387             | 41.8        | 9.26          | 40.63         | 5.614           | 0.3116       | 0.6484          |
| g9        | 237             | 38.99       | 6.08          | 45.90         | 1.794           | 0.2155       | 0.6224          |
| g10       | 178             | 14.4        | 12.35         | 37.89         | 10.557          | 0.1340       | 0.2096          |
| g11       | 586             | 93.6        | 6.26          | 37.72         | 10.979          | 0.6306       | 1.7361          |
| g12       | 577             | 75.5        | 7.63          | 37.39         | 11.836          | 0.5521       | 1.2085          |
| g13       | 335             | 96.8        | 3.46          | 37.40         | 11.810          | 0.5060       | 2.4326          |
| g14       | 301             | 70.4        | 4.27          | 37.33         | 12.021          | 0.4092       | 1.0673          |
| g15       | 1001            | 65.7        | 15.21         | 37.94         | 6.892           | 0.6380       | 1.3859          |
| g16       | 601             | 87.9        | 6.83          | 36.87         | 13.359          | 0.6180       | 1.5100          |
| g17       | 901             | 81.4        | 11.06         | 37.17         | 12.475          | 0.6964       | 1.7182          |
| g18       | 214             | 64          | 3.35          | 37.44         | 11.719          | 0.3527       | 1.1193          |
| g19       | 336             | 78.38       | 4.289         | 38.15         | 9.943           | 0.4508       | 1.7221          |
| g20       | 149             | 25.5        | 5.84          | 37.28         | 12.161          | 0.1665       | 0.3202          |
| g21       | 65              | 25.067      | 2.595         | 34.81         | 21.442          | 0.1816       | 0.3390          |
| g22       | 257             | 75.608      | 3.399         | 35.70         | 17.479          | 0.4029       | 1.5295          |
| Average   | 7.8174          | 39.5935     | 9.249         | 0.3974        | 1.1205          |

In Table 2, we present the experimental results of our statistical (Weighted average filter) proposed method that applied on twenty two grayscale images based on the size of the image before and after compression, compression ratio (CR), compression time, and decompression time.

Table 2. Experimental results of proposed method (weighted average filter) based on size image (original and Compressed), CR, PSNR, MSE, and the time of compression and decompression.

| Image No. | Image Size (KB) | CR Original | CR Compressed | PSNR Original | PSNR Compressed | MSE Original | MSE Compressed | Compression Time (second) | Decompression Time (second) |
|-----------|-----------------|-------------|---------------|---------------|-----------------|--------------|-----------------|---------------------------|-----------------------------|
| g1        | 386             | 71.5        | 5.412         | 38.97         | 8.225           | 0.6035       | 1.1225          |
| g2        | 137             | 7.49        | 18.39         | 54.01         | 0.258           | 0.0905       | 0.2500          |
| g3        | 134             | 10.23       | 13.1          | 42.25         | 3.865           | 0.1366       | 0.1226          |
| g4        | 133             | 71.5        | 5.41          | 43.90         | 2.647           | 0.1180       | 0.1997          |
| g5        | 199             | 7.4         | 18.39         | 39.99         | 6.513           | 0.2380       | 0.6292          |
| g6        | 410             | 10.2        | 13.100        | 41.01         | 5.142           | 0.4631       | 1.4143          |
Meanwhile, in Table 3, we present the experimental results of our statistical (Histogram variance filter) proposed method that applied on twenty two grayscale images based on the size of the image before and after compression, compression ratio (CR), compression time, and decompression time.

Table 3. Experimental results of proposed method (Histogram variance filter) based on size image (original and compressed), CR, PSNR, MSE, and the time of compression and decompression.

| Image No. | Image Size (KB) | CR | PSNR | MSE | Compression time (second) | Decompression time (second) |
|-----------|----------------|----|------|-----|---------------------------|----------------------------|
| g7        | 650            | 10.1 | 13.114 | 39.82 | 6.765                     | 0.7883                     | 2.164                     |
| g8        | 387            | 32.5 | 40.63 | 5.614 | 0.3727                     | 0.6429                     |
| g9        | 237            | 60.8 | 45.59 | 1.794 | 0.2531                     | 0.6378                     |
| g10       | 178            | 90   | 7.228 | 37.89 | 10.557                    | 0.1641                     | 0.2182                    |
| g11       | 586            | 41.8 | 9.262 | 37.72 | 0.3727                     | 0.8052                     | 1.7305                    |
| g12       | 577            | 38.9 | 6.087 | 37.39 | 11.836                    | 0.7479                     | 1.2101                    |
| g13       | 335            | 14.4 | 12.359 | 37.40 | 11.810                    | 0.6629                     | 2.6509                    |
| g14       | 301            | 93.6 | 6.260 | 37.33 | 12.021                    | 0.5292                     | 1.1544                    |
| g15       | 1001           | 75.5 | 7.634 | 39.74 | 6.892                     | 0.8101                     | 1.3618                    |
| g16       | 601            | 96.8 | 3.463 | 36.87 | 13.359                    | 0.8426                     | 1.5368                    |
| g17       | 901            | 70.4 | 4.275 | 37.17 | 12.475                    | 0.8531                     | 1.6958                    |
| g18       | 214            | 65.7 | 15.215 | 37.44 | 11.719                    | 0.4661                     | 1.1978                    |
| g19       | 336            | 87.9 | 6.834 | 38.15 | 9.943                     | 0.5568                     | 1.8296                    |
| g20       | 149            | 81.4 | 11.060 | 37.28 | 12.161                    | 0.2401                     | 0.3059                    |
| g21       | 65             | 64   | 3.358 | 34.81 | 21.442                    | 0.2226                     | 0.3403                    |
| g22       | 257            | 78.3 | 4.289 | 35.70 | 17.479                    | 0.5696                     | 1.5216                    |

Average: **7.8174**  **39.5935**  **9.249**  **0.4788**  **1.0879**
Figure 9 presents compressed image, the histogram of the compressed image was similar to that of the original image and its histogram. This means that the proposed technique can compressed with minimum changes in the compressed image.

| Image No. | Image Size (KB) | CR | PSNR | MSE | Time (second) | Compression | decompression |
|-----------|-----------------|----|------|-----|---------------|-------------|---------------|
| Original  | Compressed      |    |      |     |               |             |               |
| g8        | 387             | 39.2 | 9.88 | 38.97| 8.241         | 0.4998      | 0.6784        |
| g9        | 237             | 39.02| 6.08 | 44.82| 2.140         | 0.3307      | 0.6124        |
| g10       | 178             | 14.4 | 12.33| 36.56| 14.330        | 0.1984      | 0.2095        |
| g11       | 586             | 93.9 | 6.23 | 36.25| 15.405        | 0.9497      | 1.763         |
| g12       | 577             | 76.3 | 7.55 | 35.70| 17.463        | 0.9029      | 1.2194        |
| g13       | 335             | 96.9 | 3.46 | 35.84| 16.911        | 0.7662      | 2.715         |
| g14       | 301             | 70.5 | 4.27 | 35.62| 17.789        | 0.6540      | 1.1568        |
| g15       | 1001            | 66.07| 15.14| 38.24| 9.745         | 0.9638      | 1.3849        |
| g16       | 601             | 88.1 | 6.82 | 35.15| 19.857        | 1.0469      | 1.5282        |
| g17       | 901             | 81.8 | 1    | 35.38| 18.803        | 0.9989      | 1.7416        |
| g18       | 214             | 64   | 3.35 | 36   | 16.333        | 0.6386      | 1.0835        |
| g19       | 336             | 78.34| 4.29 | 36.62| 14.133        | 0.7233      | 1.6906        |
| g20       | 149             | 25.6 | 5.83 | 35.57| 18.030        | 0.2843      | 0.3056        |
| g21       | 65              | 25.089| 2.592| 33.45| 29.357        | 0.2578      | 0.3443        |
| g22       | 257             | 75.743| 3.393| 34.29| 24.211        | 0.7089      | 1.4965        |
| Average   | 7.8174          | 38.069| 13.291| 0.5869| 1.0750        |             |               |

Figure 9 Images and histogram before and after compression process
Based on the experimental results in the Tables above, this research compares the performance of our statistical proposed methods to two standard methods: JPEG and JPEG2000. Table III shows the comparison through the main parameters which are used for evaluating each of these methods. Parameters are compression ratio (CR), peak signal-to-noise ratio (PSNR), and mean square error (MSE).

Table IV. CR, PSNR, and MSE of the proposed statistical methods compared with that of other methods

| Methods              | CR      | PSNR   | MSE     |
|----------------------|---------|--------|---------|
| JPG                  | 5.7860  | 38.69  | 10.486  |
| JPG2000              | 4.443   | 43.18  | 3.4212  |
| Average              | 7.8174  | 39.595  | 9.249  |
| weighted average     | 7.8174  | 39.595  | 9.249  |
| Histogram variance   | 7.8171  | 38.0695 | 13.2914 |

The result in Table IV proves that the proposed methods based on statistical rule (average or weighted average) achieves an average CR value of 7.8174, an average PSNR of 39.5935, an average MSE of 9.249 and an average of 0.82. Meanwhile, the result in Table IV demonstrates that the proposed method based on statistical rule (Histogram variance) realizes an average CR value of 7.8171, an average PSNR of 38.0695, an average MSE of 13.2914 and an average of 0.82. Experimental results illustrated that our statistical proposed method is outperformed the standard JPEG method in each parameter were used to evaluating as seen in Figure 10.

Although, our statistical proposed method achieved an increase in compression ratio (7.8174) in each of average and weighted average methods and (7.8171) in the histogram variance method compared to JPEG2000 that produced compression ratio is (4.443). However, our statistical proposed method cannot yield outperform in each of PSNR and MSE compared to JPEG2000 as seen in Figure 10.

Figure 10 presents compare our proposed statistical rule-based method to standard methods: JPEG and JPEG2000 through the core parameters which are CR, PSNR, and MSE.

![Figure 10. Shows the parameters CR, PSNR, and MSE of the proposed statistical methods compared to standard methods (JPEG and JPEG2000)](image-url)
On the other hand, based on the related works, compares the performance of our proposed statistical methods to the methods discussed in related work in section 2 under different aspects which are CR, PSNR, MSE, and time of compression (second), as shown in Table 5.

| Methods                              | Image         | CR   | PSNR | MSE   | Compression time (second) |
|--------------------------------------|---------------|------|------|-------|---------------------------|
| Average                             | Lena 128x128  | 1.986| 38.06| 10.144| 0.0873                    |
| weighted average                     |               | 1.986| 38.06| 10.144| 0.0873                    |
| Histogram variance                   |               | 1.986| 36.89| 13.306| 0.0924                    |
| BIA (Bat Inspired Algorithm) [26]    | Lena 128x128  | 1.486| 32.909| 9.799 | 33.376                    |
| [24]                                 |               | 3.2  | 11.91| /     | /                         |
| Average                             | Lena 256x256  | 2.595| 34.81| 21.442| 0.1816                    |
| weighted average                     |               | 2.595| 34.81| 21.442| 0.2226                    |
| Histogram variance                   |               | 2.592| 33.45| 29.357| 0.2578                    |
| BIA (Bat Inspired Algorithm) [26]    | Lena 256x256  | 1.486| 32.909| 9.799 | 732.345                   |
| [24]                                 |               | 3.2  | 11.91| /     | /                         |
| Average                             | Lena 512x512  | 3.399| 35.70| 17.479| 0.4029                    |
| weighted average                     |               | 3.399| 35.70| 17.479| 0.5696                    |
| Histogram variance                   |               | 3.393| 34.29| 24.211| 0.7089                    |
| [27]                                 |               | 1.320| 30.389| 16.118| 0.518                     |
| [25]                                 |               | 0.92 | 31.95| 18.24 | /                         |

As shown in Table 5, all methods have applied to the Lena image transformed into gray level with different sizes (128x128, 256x256, and 512 x 512). The experimental results of PSNR lead us to our statistical proposed methods is clearly outperforming over the image compression approaches accomplished by Veendevi and Ananth (2011), Zhang et al. (2014), Menassel et al. (2020), and Menassel (2020). The results of CR of our statistical proposed methods also are excelling over the image compressions approaches proposed by Zhang et al. (2014), Menassel et al. (2020), and Menassel (2020). Furthermore, our proposed statistical methods achieved in a time of are outdoing the approaches of image compression introduced by Veendevi and Ananth (2011), Menassel et al. (2020) ( for Lena image with size 128x128); also, Menassel et al. (2020) ( for Lena image with size 256x256); Zhang et al. (2014), and Menassel (2020) ( for Lena image with size 512x512).

6. Conclusion
In this paper, we have presented a new proposed method for a statistical rules-based digital image compression. Our proposed compression method evaluated by using various grayscale images and different sizes, under various parameters (CR, PSNR, MSE, and compression time). Results demonstrated the efficiency of the proposed algorithm and compared it with previous methods. Therefore, our statistical method offers better results in quality (PSNR), CR, and encoding time.

In future work, we are planning to improve our statistical compression for different architectures and datasets. We will also further improve the statistical rules of our compression on the data sets with relatively high compression metrics.
References

[1] J. Uthayakumar, T. Vengattaraman, and P. Dhavachelvan, “A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. Preprint, 2018, doi: 10.1016/j.jksuci.2018.05.006.

[2] H. W. Tseng and C. C. Chang, “High capacity data hiding in JPEG-compressed images,” *Informatica*, vol. 15, no. 1, pp. 127–142, 2004, doi: 10.15388/Informatica.2004.050.

[3] M. Senthilkumar and V. Mathivanan, “Analysis of Data Compression Techniques using Huffman Coding and Arithmetic Coding,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 6, no. 5, pp. 930–936, 2016.

[4] K. Sayood, *Introduction to Data Compression*, Fifth. Morgan Kaufmann is an imprint of Elsevier, 2017.

[5] J. E. N. A. S. Joseph N. Srikanth, “A Novel Approach of Modified Run Length Encoding Scheme for High Speed Data Communication Application,” *Int. J. Sci. Res.*, vol. 2, no. 12, pp. 293–296, 2013.

[6] G. E. Blelloch, *Introduction to Data Compression*. 2013.

[7] A. Kumar and P. Singh, “An image compression algorithm for gray scale image,” *2011 IEEE*, pp. 342–346, 2012, doi: 10.1109/etncc.2011.6255917.

[8] D. S. Prashantha, N. K N, K. H, A. S. R, and L. C, “an Approach for Multispectral Image Compression,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 5, no. 3, pp. 237–240, 2020, doi: 10.33564/ijeast.2020.v05i03.036.

[9] J. R. Jayapandiyan, C. Kavitha, and K. Sakthivel, “Optimal Secret Text Compression Technique for Steganographic Encoding by Dynamic Ranking Algorithm,” in *Journal of Physics: Conference Series*, 2020, vol. 1427, no. 1, doi: 10.1088/1742-6596/1427/1/012005.

[10] S. L. Agrwal, M. Sharma, D. Kumari, and S. K. Gupta, “Improved image compression technique using IWT-DCT transformation,” *Proc. 2016 2nd Int. Conf. Next Gener. Comput. Technol. NGCT 2016*, no. October, pp. 683–686, 2017, doi: 10.1109/NGCT.2016.7877499.

[11] A. H. Ali, A. N. Abbas, L. E. George, and M. R. Mokhtar, “Image and audio fractal compression: Comprehensive review, enhancements and research directions,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 3, pp. 1564–1570, 2019, doi: 10.11591/ijeecs.v15.i3.pp1564-1570.

[12] Y. Q. Shi and H. Sun, *Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards, second edition*. 2017.

[13] L. Helminger, A. Djelouah, M. Gross, and C. Schroers, “Lossy Image Compression with Normalizing Flows,” vol. 1, pp. 1–12, 2020.

[14] A. Saidani, J. Xiang, and D. Mansouri, “A New Lossless Compression Scheme for WSNs Using RLE Algorithm,” *2019 20th Asia-Pacific Netw. Oper. Manag. Symp. Manag. a Cyber-Physical World, APNOMS 2019*, pp. 1–6, 2019, doi: 10.23919/APNOMS.2019.8893093.
[15] M. Alslehat, A. Aldulaimi, and M. Otiar, “fg,” *Int. J. Sci. Appl. Inf. Technol.*, vol. 8, no. 6, pp. 30–35, 2019.

[16] E. Aldemir, G. Tohumoglu, and M. A. Selver, “Binary medical image compression using the volumetric run-length approach,” *Imaging Sci. J.*, vol. 67, no. 3, pp. 123–135, 2019, doi: 10.1080/13682199.2019.1565695.

[17] T. Markas and J. Reif, “Multispectral image compression algorithms,” *Data Compression Conf. Proc.*, no. August, pp. 391–400, 1993, doi: 10.1109/DCC.1993.253110.

[18] M. R. Hasan, “Data Compression using Huffman based LZW Encoding Technique,” *Int. J. Sci. Eng. Res.*, vol. Volume 2, no. 11, pp. 1–7, 2011.

[19] S. Anantha Babu, P. Eswaran, and C. Senthil Kumar, “Lossless Compression Algorithm Using Improved RLC for Grayscale Image,” *Arab. J. Sci. Eng.*, vol. 41, no. 8, pp. 3061–3070, 2016, doi: 10.1007/s13369-016-2082-x.

[20] M. K. Abdouleleh, A. Masmoudi, and M. S. Bouhlel, “A New Method Which Combines Arithmetic Coding with RLE for Lossless Image Compression,” *J. Softw. Eng. Appl.*, vol. 05, no. 01, pp. 41–44, 2012, doi: 10.4236/jsea.2012.51007.

[21] N. Chukwuchekwa, B. Akinwola, and J. Onojo, “Improved DCT based Digital Image Watermarking Technique using Particle Swarm Optimization Algorithm,” no. February, 2019.

[22] A. A. Nashat and N. M. H. Hassan, “Image compression based upon Wavelet Transform and a statistical threshold,” *Proc. - 2016 Int. Conf. Optoelectron. Image Process. ICOIP 2016*, no. January, pp. 20–24, 2016, doi: 10.1109/OPTIP.2016.7528492.

[23] A. Kulkarni and A. Junnarkar, “Gray-Scale Image Compression Techniques: A Review,” *Int. J. Comput. Appl.*, vol. 131, no. 13, pp. 22–25, 2015, doi: 10.5120/jca2015907519.

[24] V. S. V and A. G. Ananth, “Fractal Image Compression of Satellite Imageries,” *Int. J. Comput. Appl.*, vol. 30, no. 3, pp. 33–36, 2011, doi: 10.5120/3621-5056.

[25] X. Zhang, Y. Ren, and Z. Qian, “Compressing encrypted images with auxiliary information,” *IEEE Trans. Multimed.*, vol. 16, no. 5, pp. 1327–1336, 2014, doi: 10.1109/TMM.2014.2315974.

[26] R. Menassel, I. Gaba, and K. Titi, “Introducing BAT inspired algorithm to improve fractal image compression,” *Int. J. Comput. Appl.*, vol. 42, no. 7, pp. 697–704, 2020, doi: 10.1080/1206212X.2019.1638631.

[27] R. Menassel, “Optimization of Fractal Image Compression,” *Fractal Anal. - Sel. Examples*, no. July, 2020, doi: 10.5772/intechopen.93051.

[28] L. Zhuang and Y. Guan, “Image Enhancement via Subimage Histogram Equalization Based on Mean and Variance,” *Comput. Intell. Neurosci.*, vol. 2017, 2017, doi: 10.1155/2017/6029892.

[29] A. Kaur, S. Gupta, L. Sahi, and S. Padda, “COMPREHENSIVE STUDY OF IMAGE COMPRESSION,” *J. Crit. Rev.*, vol. 7, no. 17, pp. 2382–2388, 2020.

[30] D. Tao, S. Di, Z. Chen, and F. Cappello, “Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled
Quantization,” Proc. - 2017 IEEE 31st Int. Parallel Distrib. Process. Symp. IPDPS 2017, pp. 1129–1139, 2017, doi: 10.1109/IPDPS.2017.115.

[31] V. Chaurasia and V. Chaurasia, “Statistical feature extraction based technique for fast fractal image compression,” J. Vis. Commun. Image Represent., vol. 41, pp. 87–95, 2016, doi: 10.1016/j.jvcir.2016.09.008.