Analysis of Run-Length Encoding Compression Effectiveness on Binary, Grayscale, and RGB Images

M. Athaullah Daffa Kusuma M (13522044)[1,2,3]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522044@std.stei.itb.ac.id, [2]athaullah@atha.pro, [3]daffakusumamulia@gmail.com

***Abstract*—Run-Length Encoding (RLE) is one of the simplest lossless data compression algorithms that works by replacing consecutive identical data sequences with a single value and its repetition count. This study aims to analyze the effectiveness of RLE compression on three types of digital images: binary, grayscale, and RGB. Compression efficiency is measured by calculating the percentage of space savings comparing original size to compressed size. It is hypothesized that RLE will achieve the highest compression efficiency on binary images due to the presence of only two intensity values, which is expected to create longer consecutive runs. Grayscale images are anticipated to show moderate and variable efficiency depending on image content and homogeneity. RGB images are expected to exhibit the lowest efficiency, potentially resulting in negative compression (size expansion) due to high color variation across pixels. This study aims to demonstrate that RLE is most suitable for images with large homogeneous regions and limited color palette, while it may prove ineffective for natural photographs or images with high color complexity.**

***Keywords*—Digital Image Processing, Image Compression, Lossless Compression, Run-Length Encoding**

I. Introduction

In the modern digital era, the volume of digital image data continues to grow exponentially. Images are utilized across various domains, including medical imaging, satellite imagery, digital photography, and multimedia applications. This rapid growth presents significant challenges in terms of storage capacity and data transmission bandwidth. Consequently, image compression has become an essential field of study in digital image processing, aiming to reduce file sizes while maintaining acceptable quality levels.

Image compression techniques are generally categorized into two main types: lossy and lossless compression. Lossy compression achieves higher compression ratios by permanently eliminating some data, which may result in quality degradation. In contrast, lossless compression preserves all original data, allowing perfect reconstruction of the original image. Lossless compression is particularly crucial in applications where data integrity is paramount, such as medical diagnostics, legal documentation, and scientific imaging.

Run-Length Encoding (RLE) is one of the oldest and simplest lossless compression algorithms. The fundamental principle of RLE involves replacing sequences of identical consecutive data values, known as "runs," with a pair consisting of the value and its repetition count. For instance, a sequence of fifteen consecutive white pixels would be encoded as a single entry containing the white pixel value and the count of fifteen. This approach is computationally efficient and straightforward to implement. However, the effectiveness of RLE is highly dependent on image characteristics, particularly the presence of homogeneous regions and repetitive patterns.

Digital images can be classified into several types based on their color representation. Binary images consist of only two intensity values, typically black and white, commonly used in document scanning and text recognition. Grayscale images contain 256 levels of gray intensity and are frequently used in medical imaging and scientific applications. RGB (Red, Green, Blue) images represent full-color images using three separate color channels, resulting in approximately 16.7 million possible color combinations. Each image type possesses distinct properties that may significantly influence compression performance.

Despite the widespread use of RLE in various applications and its inclusion in several image format standards such as BMP and TIFF, comprehensive comparative analyses of its effectiveness across different image types remain limited. This study aims to address this gap by analyzing RLE compression effectiveness on binary, grayscale, and RGB images. The primary objectives are to implement RLE compression algorithms suitable for each image type, measure and compare compression efficiency, and identify characteristics that influence RLE performance across different image categories.

II. Basic Theory

A. Digital Image Representation

A digital image can be represented as a two-dimensional discrete function $f(x, y)$ where x and y denote spatial coordinates and the function value corresponds to the pixel intensity (or color). For an image with width W and height H, the total number of pixels is

$$N = W \times H \quad (1)$$

In storage and processing, the image is typically treated as a matrix and may be converted into a one-dimensional sequence using a chosen scan order. The representation depends on the image type:

1. Binary Image
   Each pixel has only two possible values (commonly 0 and 1). Although the conceptual depth is 1 bit/pixel, in practice binary images may be stored using packed bits (1 bpp) or using 8-bit values depending on the file format and implementation.
2. Grayscale Image
   Each pixel is an intensity level. A common representation uses 8 bits/pixel, providing 256 intensity levels from 0 to 255.
3. RGB Image
   Each pixel consists of three color components (R,G,B). A common representation uses 8 bits per channel, resulting in 24 bits/pixel. This larger alphabet (more possible values) generally increases local variation between adjacent pixels.

B. Image Compression and Redundancy

Digital image representation typically requires a large amount of memory, especially as resolution and color depth increase. Therefore, image compression is used to reduce redundancy in image representation so that the required storage space can be decreased [1]. In principle, an image can be compressed without reducing its visual quality, depending on the compression method applied.

In general, the objectives of image compression are:

1. To reduce storage requirements while maintaining the image's visual quality [2].
2. To represent an image with quality that is nearly the same as the original image, but in a more compact form.

Image compression methods are commonly categorized into two types:

1. Lossy Compression
   Lossy methods produce a compressed image that is visually similar to the original. Some information is removed during compression, but the loss is often acceptable to human visual perception. These methods generally aim to achieve a high compression ratio. Examples include JPEG compression and fractal image compression [1].
2. Lossless Compression
   Lossless methods always reconstruct the decompressed image exactly identical to the original, pixel by pixel. No information is lost during compression. Although the compression ratio is typically lower than lossy methods, the preserved quality remains high. Lossless compression is required for images that must not be degraded by compression, such as medical images and X-ray images. Examples include Huffman coding, Run-Length Encoding (RLE), and quantized coding [1].

C. Run-Length Encoding (RLE) Concept

Run-Length Encoding is a simple lossless compression technique that replaces a sequence of repeated values with a compact representation of that repetition. Run-Length Encoding (RLE) is well-suited for compressing images that contain groups of pixels with the same gray-level value, resulting in long sequences of identical values. In image RLE, compression is performed by forming a sequence of value pairs (p, q) for each row of pixels (i.e., the image is processed row by row) [1].

In each pair (p, q):

- The first value p represents the gray-level (pixel intensity).
- The second value q represents the number of consecutive pixels that have that gray-level value, referred to as the run length.

Because the decoding part just simply repeats each stored value p exactly q times, RLE preserves the original data without loss. However, its effectiveness depends strongly on how often identical values occur consecutively in the scan order; frequent value changes produce short runs (often $q = 1$), which can reduce compression efficiency or even increase the data size due to the overhead of storing (p, q) pairs.

D. RLE to Compress Binary, Grayscale, and RGB Images

The effectiveness of RLE depends on how often adjacent pixels are exactly equal after linearization.

1. Binary images
   With only two possible values, binary images often contain large uniform regions (e.g., document background). This increases the probability of long runs, which tends to improve compression performance.
2. Grayscale images
   Grayscale images have more possible intensity values, so exact equality between consecutive pixels is less frequent than in binary images. RLE performance becomes strongly dependent on image content whether smooth regions (e.g., sky, blurred backgrounds) that can still generate long runs, or textured/noisy regions that generate short runs that will reduce its effectiveness.
3. RGB images
   For RGB, exact matches require all three channels to match across consecutive pixels if RLE is applied to whole pixels. Natural images often exhibit high color variation, resulting in short runs and possible size expansion. In practice, RLE can be applied in different ways whether encode (R,G,B) triplets as symbols or

applying RLE separately to the R, G, and B channels.

D. Compression Performance Metrics

To evaluate RLE compression effectiveness, the original data size and the compressed data size are compared.

- S_o (original size) : the amount of memory required to store the image data before compression. It refers to the size of the raw pixel data (not including file headers/metadata), measured in bits or bytes.
- S_c (compressed size) : the amount of memory required to store the image data after RLE encoding.

Using S_o and S_c, here are the common performance metrics:

1. Compression Ratio
$$CR = \frac{S_o}{S_c} \quad (2)$$
If $CR > 1$, compression is achieved. But if $CR < 1$, the output expands.
2. Space Saving
$$SS\,(\%) = (1 - \frac{S_c}{S_o}) \times 100 \quad (3)$$
A positive SS indicates reduced storage, while a negative SS indicates an increase in size after compression.

III. Methodology

A. Image Preparation and Formatting

Experiments were conducted on three image categories: binary, grayscale, and RGB images. For each category, multiple images with different visual characteristics were included to observe how image content affects the number and length of runs produced by RLE. All images were processed at their original spatial resolution $W \times H$.

To ensure consistent processing across image types, each image was prepared as follows:

1. Binary images
Images were represented using two intensity values (0 and 1). If a source image was not originally binary, a thresholding step was applied to obtain a binary representation.
2. Grayscale images
Images were represented as single-channel intensity values (commonly 8-bit for 0–255). If the source image was RGB, it was converted to grayscale prior to encoding.
3. RGB images
Images were represented using three channels (R,G,B). RLE is applied per channel (channel-wise), meaning the R, G, and B planes are encoded separately.

Before applying RLE, each image was serialized using a row-wise scan order (left-to-right within a row, top-to-bottom across rows). This matches the common RLE approach of generating run pairs for each row of pixels.

B. RLE Encoding & Decoding Scheme

RLE compression was performed by generating a sequence of run pairs (p, q) along the scan order:

- p is the pixel value (gray-level) in the current channel.
- q is the run length, i.e., the number of consecutive pixels equal to p.

For binary and grayscale images, RLE is applied once on the single channel. But for RGB images, RLE is applied three times, producing three independent RLE streams:

- R channel RLE $[(P_{R_0}, Q_{R_0}),\ (P_{R_1},\ Q_{R_1}),\ \dots]$
- G channel RLE $[(P_{G_0}, Q_{G_0}),\ (P_{G_1},\ Q_{G_1}),\ \dots]$
- B channel RLE $[(P_{B_0}, Q_{B_0}),\ (P_{B_1},\ Q_{B_1}),\ \dots]$

Here's are the general RLE Encoding steps that used:

1. intialize
$p \leftarrow first\ value\ in\ the\ sequence,\ q \leftarrow 1$
2. for each next value x, if $x = p$ then $q \leftarrow q + 1$, else output (p, q) and set $p \leftarrow x,\ q \leftarrow 1$
3. Output the last (p, q)

Decoding is done by repeating each p value q times to reconstruct the original channel sequence, ensuring lossless recovery.

C. Size Computation

Compression effectiveness is evaluated by comparing:

- S_o (original size) is the storage required for the uncompressed raw pixel data (excluding file headers/metadata)
- S_c (compressed size) the storage required to store all RLE pairs (p, q) produced by encoding.

1. Original Size
Let b is the bit depth per pixel, for binary the b is 1 bit/pixel, for grayscale is 8 bits/pixel, and for RGB is 24 bits/pixel (8 bits times 3 channels), the Original Size can be count with formula (4)
$$S_o = W \times H \times b \quad (4)$$
2. Compressed Size
Let each run pair store p using b_p bits (for grayscale and each channels in RGB is 8 bits) and q using b_q bits (for this case we will using 8 bits, if q is more than 255, it will be stored in different pair), the Compressed Size can be count with formula (5)
$$S_c = K \times (b_p + b_q) \quad (5)$$
For RGB, K is $K_R + K_G + K_B$.

D. Performance Metrics

The following metrics are reported:

1. Compression Ratio

$$CR = \frac{S_o}{S_c} \quad (6)$$

2. Space Savings

$$SS(\%) = \left(1 - \frac{S_o}{S_c}\right) \times 100 \quad (7)$$

A positive value indicates storage reduction, and a negative value indicates size expansion.

E. Experimental Procedure

1. Select input images for each category (binary, grayscale, RGB)
2. Serialize pixels using row-wise scanning
3. Apply RLE
4. Compute S_o and S_c using the defined size model
5. Compute CR and SS for each image
6. Compare results across image types and analyze how image content influences run counts and run lengths

IV. Result and Analysis

For the first test case, we will use figure 4.1

Figure 4.1. Scenery Image (Source : https://www.pexels.com/photo/dock-under-cloudy-sky-in-front-of-mountain-206359/)

Figure 4.1 will be converted to grayscale and binary. Following result as the space saving result for figure 4.1.

Jenis Citra	Space Savings
Biner	98.77%
Grayscale	-16.14%
RGB	-24.14%

Figure 4.2. Space saving result for RLE Encoding figure 4.1

As we can see, the space savings for grayscale and RGB are negatives. That means the RLE Encoding makes the size of the Grayscale and RGB version of the image expand.

For the second test case, we will use figure 4.3

Figure 4.3. Rick Astley Image (Source : https://www.youtube.com/watch?v=dQw4w9WgXcQ)

Figure 4.3 will be converted to grayscale and binary. Following result as the space saving result for figure 4.3.

Jenis Citra	Space Savings
Biner	97.73%
Grayscale	-62.71%
RGB	-64.84%

Figure 4.4. Space saving result for RLE Encoding figure 4.3

As we can see, the space savings for grayscale and RGB are negatives. That means the RLE Encoding makes the size of the Grayscale and RGB version of the image expand.

For the third test case, we will use figure 4.5

Figure 4.5. Kabosu, Shiba Inu (Doge) (Source : https://kabosu112.exblog.jp/iv/detail/?s=9944144&i=201002%2F12%2F90%2Fa0126590_22301391.jpg)

Figure 4.5 will be converted to grayscale and binary. Following result as the space saving result for figure 4.5.

Jenis Citra	Space Savings
Biner	98.55%
Grayscale	-44.15%
RGB	-48.40%

Figure 4.6. Space saving result for RLE Encoding figure 4.5

As we can see, the space savings for grayscale and RGB are negatives. That means the RLE Encoding makes the size of the Grayscale and RGB version of the image expand.

For the final test case, we will use figure 4.7

Figure 4.7. Japanese Singer Noriyuki Makihara (Source : https://www.youtube.com/watch?v=naz0-szzYXk)

Figure 4.7 will be converted to grayscale and binary. Following result as the space saving result for figure 4.7.

Jenis Citra	Space Savings
Biner	97.36%
Grayscale	-9.05%
RGB	-12.06%

Figure 4.8. Space saving result for RLE Encoding figure 4.7

As we can see, the space savings for grayscale and RGB are negatives. That means the RLE Encoding makes the size of the Grayscale and RGB version of the image expand.

The reason RLE achieved very high savings on the binary test image (approximately 90%) is that binary images contain only two possible pixel values, which commonly form large homogeneous regions such as uniform backgrounds and solid foreground objects. When the image is scanned row-wise, these homogeneous areas generate long runs of identical values, causing the number of stored run pairs (p, q) to be much smaller than the total number of pixels. As a result, the overhead of storing run lengths is amortized across many repeated pixels, yielding a small compressed size S_c relative to the original size S_o therefore a large positive space savings.

The reason the grayscale test case produced negative compression is that grayscale images have a much larger set of possible intensity values (typically 256 levels), making it less likely for adjacent pixels to be exactly identical. Even in visually smooth regions, grayscale images often contain gradual intensity changes (shading and gradients) that break runs into short segments. This increases the total number of runs K and leads to many (p, q) pairs with small run lengths. Under these conditions, the additional storage required for the run descriptors outweighs any reduction from grouping repeated values, resulting in $S_c > S_o$ and negative $SS(\%)$.

That goes to the RGB too. The reason the RGB test case also resulted in negative compression, even with channel-wise RLE, is that natural color images typically exhibit high local variation in each channel due to texture, edges, illumination differences, and sensor noise. Because the R, G, and B channels are encoded independently, the compressed size depends on the sum of runs across channels $(K_R + K_G + K_B)$. When each channel changes frequently along the scan direction, the run counts become large and the encoded output contains a high number of short runs. Consequently, the combined overhead of storing (p, q) pairs for three channels becomes substantial, often exceeding the original 24-bit/pixel storage, which explains why the RGB compressed size expands rather than shrinks.

V. Conclusion

Run-Length Encoding (RLE) provides very high compression efficiency on binary images, with the test case showing approximately 90% space savings. This behavior occurs because binary images typically contain large homogeneous regions and only two possible pixel values, which produce long consecutive runs and a small number of stored (p, q) pairs.

For grayscale and RGB (channel-wise) images, the results show negative compression, meaning the RLE output is larger than the original data. The main cause is the higher intensity and color variability in these image types, which leads to frequent pixel changes along the scan order and generates many short runs. In such conditions, the overhead of storing run descriptors (p, q) dominates and outweighs any compression benefit.

Overall, RLE is most suitable for images with large uniform areas and limited value variation (e.g., binary documents, simple graphics), while it is generally ineffective for images with gradients, textures, or natural photographic content. Future improvements may include alternative scan patterns, preprocessing such as quantization to increase repeated values, or combining RLE with additional coding methods to reduce run-representation overhead.

VI. ACKNOWLEDGMENT

I would like to thank God Almighty who has blessed me so that the work on this paper can run smoothly from start to finish. Then, I would like to thank my family, including my parents, who have supported me in studying at ITB. I also thank all the lecturers and assistants, especially Mr. Dr. Ir. Rinaldi Munir, M.T. for his guidance in studying the IF4073 Digital Image Processing Subject.

REFERENCES

[1] Munir, Rinaldi. 2025. "Pemampatan Citra (Bagian 1)", https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2025-2026/26-Image-Compression-Bagian1-2025.pdf. [Accessed: Dec. 24, 2025]

[2] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. New York, NY, USA: Pearson, 2018.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025

M. Athaullah Daffa Kusuma M (13522044)