Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 1, January 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v9i1.13186 p-ISSN : 2541-044X

Enhancing Least Significant Bit Steganography Image Fidelity Using Brotli Compression

Ariel Emmanuel Prayogo [1]*, Adhitya Nugraha [2], Fandy Novanto[3], Joshua Calvin Kurniawan[4]
[1,2,3,4]Universitas Dian Nuswantoro, Indonesia
[1]arielemmanuel4@gmail.com, [2]adhitya@dsn.dinus.ac.id, [3]fandy2832@gmail.com,
[4]kurniawanjoshua05@gmail.com

Submitted : Nov 21, 2023 | **Accepted** : Nov 24, 2023 | **Published** : Jan 1, 2024

Abstract: The rapid growth of technology has provided extensive convenience and openness in accessing information, yet this hasn't been balanced with an equivalent enhancement in information security. Steganography plays a crucial role in concealing and protecting data, with the Least Significant Bit (LSB) method being a commonly used algorithm that operates by substituting the least significant bits in the image pixels with the bits of the data to be hidden, aiming to preserve the image quality. This research aims to enhance the quality of the steganography result by embedding data with text data type using LSB by employing the Brotli compression technique, coupled with increasing image's capacity for embedding text data which will be used as data sample on this research. Brotli compression aims to reduce the size of the resulting stego image by combining embedded data that share identical values. Experiment results will be obtained by comparing the original image with the stego image using metrics like Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index (SSIM). The experiments successfully demonstrated that the integration of LSB with Brotli compression outperformed the regular LSB method, showing a 6.64% increase in PSNR, followed by a decrease in MSE by approximately 63.79%, and an increase in SSIM by around 0.0039%. This was accompanied by a continuous increase in compression values depending on the input data size. These results indicate that the integration of LSB with Brotli compression was successfully implemented to enhance the fidelity of the stego image.

Keywords: Brotli; Capacity; Compression; Fidelity; LSB; Stego Image; Text Data

INTRODUCTION

The rapid growth of technology has indeed facilitated the exchange of information(Masruri et al. , 2019). However, the security needs provided for information have diminished, often unable to balance the openness and advancement of transmitted information technology across various internet media(Nashat et al. , 2019). This imbalance has led to an increase of information flowing through the internet, becoming a route for attacks and cybercrimes by hackers, thereby escalating the necessity for information security(Masruri et al. , 2019; Utomo et al. , 2021). Therefore, effective method must be used to safeguard the confidentiality and security of information from unauthorized access.

This problem can be solved through the information system security technique called Steganography, a method of concealing data by embedding it within a cover or carrier media

*name of corresponding author

object(Utomo & Erwanto, 2019). Generally, steganography is known as a method of hiding messages within a cover medium, known only to the sender and recipient(Baby et al. , 2020). Thus, Steganography offer to ensure that unauthorized parties remain unaware of the existence of a secret message as the transmitted media appears non-suspicious(Baby et al., 2020; Utomo & Erwanto, 2019).

This research is important to finding solutions to enhance data message security by improving quality and providing bigger capacity. Therefore, this study aims to enhance steganography results using the LSB technique combined with the Brotli compression algorithm to increase efficiency in image's pixel capacity utilization and the quality of the resulting image. Through these combination of techniques, the aim is to produce high-accuracy images that maintain similarity to the original image. Experimentation will compare the original image with the stego image using metrics like Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index (SSIM). A stego image's quality can be deemed good if the PSNR metric value is above 40dB(Alifi & Suartana, 2020). These metric values serve as benchmarks to measure and evaluate the stego image's quality, particularly in terms of imperceptibility and robustness(Anshori & Dodu, 2019).

Although previous studies have explored LSB, there are still opportunity to further enhance optimal results. This research aims to use those opportunity and serve as a reference for future researchers to study deeper into similar topics. The primary difference in this study from prior research lies in applying compression methods to the LSB technique, where Brotli compression is employed on text data by compressing the overall data pattern to be embedded, resulting in a reduction in the size of large text data. Consequently, this indirectly increases image capacity as the space needed to accommodate a compressed messages decrease.

LITERATURE REVIEW

Steganography

Steganography in images is one of the most popular techniques employed in implementing steganography(A. Ansari et al. , 2019). This is primarily due to the fact that every image comprises pixels that can be utilized as a medium for hiding data. This technique even offers higher effectiveness and security levels when the image used as a cover medium employs the RGB color format. Images in the RGB format have a larger pixel space compared to other formats like greyscale. This is because each pixel in an RGB image represents three color components: red, green, and blue. These three color components provide 8 bits per pixel, allowing for a greater capacity to embed data(Astuti et al. , 2020).

Least Significant Bit (LSB)

Many method exist in steganography for concealing data. One of the most widely used methods is the Least Significant Bit (LSB), where this technique alters the least significant bits of image pixels with bits from the data to be hidden, aiming to preserve the image quality(Mahdi & Khodher, 2021). LSB is frequently chosen because of its ease, simple and fast implementation(Ruang & Yuv, 2020). However, the results of steganography using this algorithm still have room for improvement in visual quality to significantly minimize potential damage to the image that might raise suspicion from unauthorized parties. Modifying this approach could solve the problem in enhancing the outcomes of LSB steganography(Adhi & Husada, 2022).

Brotli Compression

Compression Brotli is chosen to be integrated into the LSB technique as a modification in this research. Brotli compression is an algorithm developed by Google, which combines LZ77, Huffman, and second order context modeling algorithms(Antony Joans Kumar M. , Christopher Columbus C. , Ben George E. , 2023). This compression falls under the category of lossless compression, meaning it compresses without losing the original data, allowing the decompression results to match the data before compression(Reynaldo, V. , Wicaksana, A. , & Hansun, 2019). The combination of these three aspects makes Brotli a good compression algorithm for handling data with numerous identical patterns and characters. Brotli summarizes these identical characters into the same block, effectively reducing

*name of corresponding author

the size of the image(Reynaldo, V. , Wicaksana, A. , & Hansun, 2019). During the compression process, Brotli works by dividing text data into small blocks. Each block is then compressed separately. Subsequently, the compression process involves data analysis using context modeling methods to enhance the efficiency of the compression process. Next, an encoding algorithm is applied, which is a combination of entropy encoding, focusing on data compacting by representing information more efficiently, and Huffman coding, providing shorter codes for frequently occurring patterns or characters.

Brotli decompression process involves the use of Brotli decompression functions that include Huffman decoding, output buffering, and sliding window. The parameters used in the decompression function include block size and the compression quality level used during the data compression process. Huffman decoding is employed to restore the representations of symbols to their original values. Then, output buffering is used to store the decompressed data, and the sliding window is utilized to track context during the decompression process(Antony Joans Kumar M., Christopher Columbus C., Ben George E., 2023).

Measurement Matrix

After steganography is performed, values are needed to evaluate its results. The most commonly used matrix measurement tools are Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index (SSIM). MSE is used to indicate the average square error value obtained by comparing the difference between the original pixel values and the pixel values of the stego image. While PSNR is useful for displaying how well the quality of the generated image is. PSNR is defined as the ratio of the maximum value to the noise value obtained from the MSE matrix calculation. SSIM is used to reflect the level of similarity between the stego image and the original image(Adhi & Husada, 2022; Astuti et al., 2020). If the result approaches the value of 1, the similarity level is higher. MSE is implemented with the mathematical equation of (1).

$$MSE = \frac{1}{O\ x\ P} \sum_{k}^{O} \sum_{l}^{P} [c(k,l) - s(k,l)]^2 \qquad (1)$$

Based on the MSE equation above, O and P represent the dimensions of the images, while k and l are the coordinates of pixels in the original image and stego-image, while PSNR can be calculated using equation (2).

$$PSNR = 10 \log 10 \left(\frac{\mathrm{MAX}^2}{MSE} \right) = 20 \log 10 \left(\frac{\mathrm{MAX}}{\sqrt{MSE}} \right) \qquad (2)$$

In the PSNR equation, *MAX* represents the maximum pixel value of the image, while *MSE* is the mean square error value of the pixel and the SSIM can be obtained using equation (3).

$$SSIM\ (c,s) = \frac{(2\mu_c\mu_s + C_1)(2\sigma_{cs} + C_2)}{(\mu_c^2 + \mu_s^2 + C_1)(\sigma_c^2 + \sigma_s^2 + C_2)} \qquad (3)$$

In the SSIM equation, the variable c represents the cover image, s represents the stego image, μ symbolizes the average intensity value of the image, while σ represents the standard deviation value of the image. Thus, μ_c is the average intensity of the cover image, μ_s is the average intensity of the stego image, σ_c is the standard deviation of the cover image, and σ_s is the standard deviation of the stego image, while $C_1$and C_2 are constants containing pixel intensity weights to prevent division by zero.

State of The Art

The researchers also conducted a review of various relevant previous studies on the current research topic. By comprehensively analyzing studies conducted within the last five years, researchers could grasp the State-of-the-Art knowledge in this field, especially concerning improvements and enhancements in stego image quality using the LSB technique.

According to the research by (Nashat & Mamdouh, 2019) an increase in capacity and quality can be achieved by implementing the LSB method involving arithmetic operations and flipping LSB. The results showed that this method could offer better quality than regular LSB and a larger capacity. This research provides in-depth insights into addressing the shortcomings of LSB in terms of quality and capacity.

Another study regarding high-capacity adaptive steganography methods using LSB and Hamming code by (Wang et al. , 2020). is related to this topic. The study aimed to increase the capacity of secret data embedded into cover media while maintaining the image quality inconspicuous. The proposed method involved edge detection, especially in areas with the most significant sharpness, to blur potential image distortions, resulting in good visual quality in the image. Hamming code was employed to enhance the capacity of secret data embedding while minimizing changes to the stego image's appearance.

The research based on (Rustad et al. , 2021) explored LSB with adaptive patterns and LSB flipping techniques, similar to the previous method. The proposed method involved selecting the most optimal pattern to minimize error ratios caused by message embedding. Patterns with the least error rates were chosen as message embedding areas to achieve good image quality.

A study by (Carlo et al. , 2020) discussed the use of Huffman compression techniques with LSB and applied the Vigenère cipher cryptographic technique to provide additional security to text data. The research aimed to compress Huffman data encrypted by Vigenère and successfully demonstrated an improvement in stego image quality compared to traditional LSB techniques.

The research by (Horng, 2020) proposed an LSB steganography method combined with Quotient Value Differencing (QVD) on an image compressed using the AMBTC method. The primary objective was to increase the capacity of the hosting image. AMBTC compression is a lossy compression technique for greyscale images, dividing the image into non-overlapping blocks, each represented by two quantization levels and a bitmap matrix. This research proved that the proposed method could provide more capacity compared to other methods while maintaining good stego quality. However, this method involved lossy compression, where the resulting data may not fully recover, and the cover object used was in greyscale format.

Research by (Kumar & Swain, 2022) proposed a Reversible Data Hiding (RDH) method for greyscale images. The first method involved matching LSB with reversibility with a duplicate image. The second method involved four identical cover images to embed secret data using n-Rightmost Bit Replacement (n-RBR) and Modified Pixel Value Differencing (MPVD) techniques. This research aimed to enhance RDH hiding techniques to improve image capacity and quality.

METHOD

This research was conducted using a literature review and pure experimental research methods. Researchers aimed to investigate the issues they aimed to solve and define the problem through experiments. They also searched for references from previous scientific journals and articles on the internet.

Valuable guidance for method selection in this paper was obtained from the research by (Tayyeh & Al-jumaili, 2022). This research explored similar compression concepts, providing a solid foundation for developing this method as an addition to cover any remaining gaps. The study proposed a combination technique of Least Significant Bit (LSB) with the Deflate compression algorithm.

Deflate, comprising LZ77 and Huffman, was used to compress secret messages before being embedded into cover images. The LSB embedding in this research also involved using an XNOR gate to select color channels as insertion locations, thereby enhancing steganography performance. The study concluded successfully, improving image quality.

Thus, making research on Deflate compression an excellent reference for this journal due to its resemblance to Brotli's compression abilities. However, in this study, Brotli also implements the principle of second-order context modeling according to (Antony Joans Kumar M. , Christopher Columbus C. , Ben George E. , 2023). This research will also implement the Least Significant Bit (LSB) approach, which will be integrated with Brotli compression techniques. The LSB method to be implemented in this research is also a regular LSB method without additional algorithms like the XNOR gate in the previous citation. The LSB method is often known as a message hiding technique in images or videos without significantly altering the visual quality, while Brotli compression is used to reduce text data size, thereby decreasing the resulting stego image size and improving visual quality.

Dataset

The dataset for embedding into cover images consists of text data with varying sizes: 1000 bytes, 2000 bytes, 3000 bytes, 4000 bytes, and 5000 bytes. This text data is generated by creating a TXT file and inputting message data that resembles human-readable messages, considering everyday language elements like spaces and other linguistic aspects, as these factors influence compression performance. This research will utilize three RGB cover images: 'Lena,' 'Baboon,' and 'Pepper,' each sized at 512 x 512 pixels with PNG file extensions.

Compression

At this stage, the inputted text data will undergo compression using the Brotli technique. The text data will be encoded using the Brotli algorithm, then converted from UTF-8 to base64, transforming the data into hexadecimal format, which will then be converted into binary. The resulting output will be compressed text data in binary form, occupying a smaller size, enabling images to efficiently accommodate it as it requires relatively less space. The compression data process is illustrated in Figure 1.

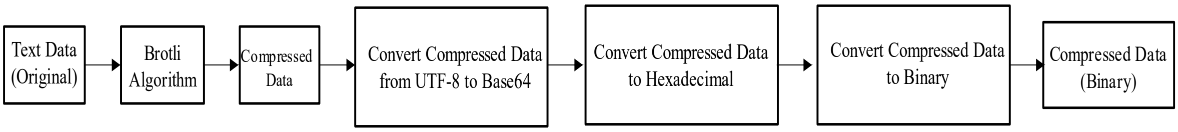

Text Data (Original)
Brotli Algorithm
Compressed Data
Convert Compressed Data from UTF-8 to Base64
Convert Compressed Data to Hexadecimal
Convert Compressed Data to Binary
Compressed Data (Binary)

Fig. 1 Brotli Compression Process

Embedding

In this embedding stage, the input will be using RGB color cover images, where this research will utilize the three cover data from the previously prepared dataset. The resulting output will be an RGB stego image in the same file format as the input. This stage will commence by selecting cover images that will serve as the container for the hidden message in PNG file format. Next, secret text data intended for embedding into these cover images will be inputted. The inputted text message will undergo Brotli compression, then encoded with base64, converting the data into hexadecimal, which will further be converted into binary to enable embedding using the Least Significant Bit (LSB) technique. The text message, now in binary form, will be embedded into the pixels of the cover image, where each pixel represents a certain number of bits to accommodate hidden data. This embedding is done by altering the least significant bit of the image pixels, replacing it with a bit from the text message. For instance, if an image has been transformed into binary code where each pixel represents a set of binary bits like **(10100010)**, **(11001100)**, **(10010011)**, and the text message formed into binary

*name of corresponding author

code is (**010**), the embedding process would result in **10100010**, **11001101**, **10010010**. Upon successful embedding, the output will be an RGB stego image in the PNG format, matching the input image format. The embedding process is shown in figure 2.

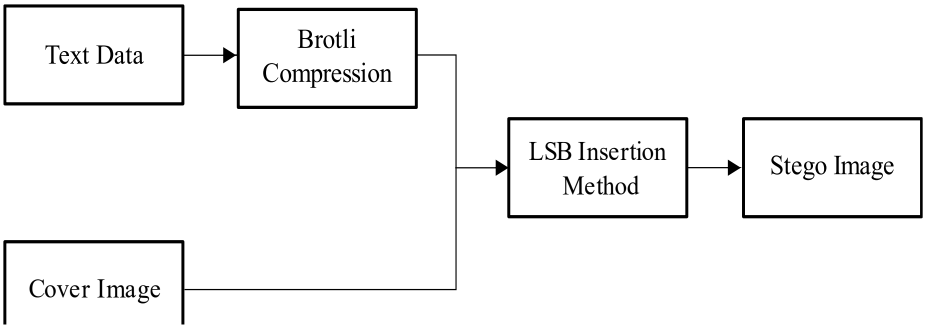


Fig. 2 Embedding Process

Extraction

In this stage, extraction will be performed on the stego image that contains the hidden message. The process begins by inputting the stego image, which will undergo extraction to retrieve the concealed text message within it. Extraction involves identifying the placement of the message data within the least significant bits of the cover image. These bits will be extracted and grouped into binary code. The outcome will then be converted into hexadecimal form and subsequently decoded from base64 to obtain the compressed text data. Therefore, it's necessary to decompress this binary data using the Brotli algorithm. The output obtained will be the original text data, the same as it was before the insertion process. The extraction process can be observed in Figure 3.

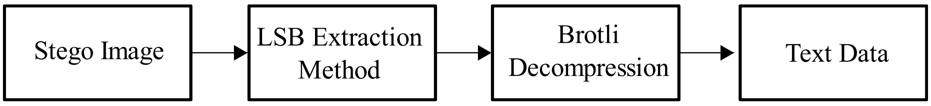


Fig. 3 Extraction Process

The results of this method's experiments will be measured and evaluated by comparing the original image with the resulting stego image using metrics such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index (SSIM). Utilizing these metrics allows for an easier understanding of the evaluation of the stego image's quality and the extent of improvement. A smaller MSE value indicates a better quality image because MSE represents the damage level to the image. Conversely, in PSNR, a higher value implies a clearer image quality. If the PSNR value is above 40dB, the stego image can be considered in good condition(Alifi & Suartana, 2020). SSIM reflects how similar the stego image is to the original one; the closer the SSIM value is to 1, the higher the similarity between the images (Carlo et al. , 2020). SSIM values typically range from -1 to 1.

RESULT

In this testing, two different evaluations will be conducted for each cover images "Lena," "Baboon," and "Pepper." Each of these images will be embedded with text data of varying sizes ranging from 1000 bytes to 5000 bytes, where the text data will undergo compression before embedding. The test results for each cover image will be presented separately. These tables will contain both the results with compression and without compression. This will allow for a comparison of how much Brotli compression contributes to improving the quality and maintaining similarity to the original image. However, before executing the experiments, the data will be compared so that the size to be embedded into the image reduces. The results of the data size after the compression using Brotli will be displayed in Table 1.

Table 1
Brotli Compression Result on Text Data

Size Before Compression (Byte)	Size After Compression (Byte)	Compressed
1000	413	58.67%
2000	718	64,10%
3000	1054	64,84%
4000	1372	65,70%
5000	1672	66,55%

Table 1 displays various compression results for each dataset that will be used. The highest compression ratio is found at 66.55% for data sized at 5000 bytes. Furthermore, this experiment aims to prove that Brotli compression can provide significantly better results when embedding larger-sized data. After successfully compressing the data, the small-sized data mentioned above will be converted to hexadecimal and then to binary to be embedded in pixels of the image using the LSB technique, so here is a comparison of pixel usage between regular LSB and Brotli LSB as shown in Table 2.

Table 2
Pixel Usage Comparison Between Regular LSB and LSB Brotli

Data Size (Bytes)	Regular LSB Pixel Usage	LSB – Brotli Pixel Usage
1000	2667	551
2000	5334	958
3000	8000	1406
4000	10667	1830

Table 2 displays a comparison of how many pixels are used in the image to accommodate data ranging from 1000 to 5000 bytes. In the table, it is found that the difference in pixel usage between regular LSB and LSB Brotli is quite significant. Additionally, it can be observed that with an embedded data size of 5000, the usage of pixels is very small, requiring only 2230 pixels. Thus, it can be said that brotli LSB successfully reduces pixel usage, allowing the cover image to have more capacity.

Overall, the results of the embedding from both methods will be presented in tabular form, including parameters such as Data Length in Bytes, Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index (SSIM). The calculations for these parameters will be computed using the previously stated mathematical equations. The testing outcomes will be showcased in comparison tables in Table 3 for the Lena cover, Table 4 for the Baboon cover, and Table 5 for the Pepper cover.

Table 3
Comparison of Regular LSB and Brotli LSB Using Lena Cover 512x512

REGULAR LSB WITHOUT COMPRESSION						LSB WITH BROTLI COMPRESSION				
Data	**1000**	**2000**	**3000**	**4000**	**5000**	**1000**	**2000**	**3000**	**4000**	**5000**
PSNR(dB)	71,02360	68,03810	66,26639	65,04269	64,07386	74,79060	72,43805	70,76183	69,65938	68,79760
MSE(dB)	0,00514	0,01022	0,01536	0,02036	0,02545	0,00216	0,00371	0,00546	0,00703	0,00858
SSIM	0,999979	0,999934	0,999887	0,999843	0,999801	0,999995	0,999987	0,999975	0,999963	0,999948

Table 3 is a comparison between regular LSB and the proposed LSB method in this paper concerning the first cover, Lena, sized at 512 x 512 pixels. According to this table, all metric values show improvement. From 1000 bytes to 5000 bytes of data, the difference in PSNR reduction continues to decrease. For the 5000-byte data, LSB Brotli managed to achieve a PSNR around 68.79760 dB, whereas Regular LSB only attained a PSNR value of 64.07386 dB..

Table 4
Comparison of Regular LSB and Brotli LSB Using Baboon Cover 512x512

REGULAR LSB WITHOUT COMPRESSION						LSB WITH BROTLI COMPRESSION				
Data	**1000**	**2000**	**3000**	**4000**	**5000**	**1000**	**2000**	**3000**	**4000**	**5000**
PSNR(dB)	71,03760	68,05217	66,33122	65,10607	64,12318	74,95489	72,47243	70,90066	69,64057	68,82213
MSE(dB)	0,00512	0,01018	0,01513	0,02007	0,02516	0,00208	0,00368	0,00528	0,00706	0,00853
SSIM	0,999999	0,999997	0,999996	0,999994	0,999992	0,999999	0,999999	0,999999	0,999998	0,999998

Table 4 presents a comparison between regular LSB and the proposed LSB method in this paper regarding the second cover, Baboon, sized at 512 x 512 pixels. According to this table, all metric values show improvement, even with a consistently excellent SSIM value around 0.99999. For data ranging from 1000 bytes to 5000 bytes, LSB Brotli on the Baboon cover provided a decreasing PSNR with a diminishing difference.

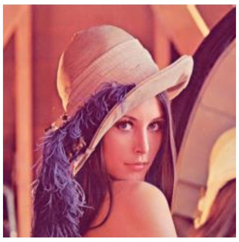
Table 5
Comparison of Regular LSB and Brotli LSB Using Pepper Cover 512x512

REGULAR LSB WITHOUT COMPRESSION						LSB WITH BROTLI COMPRESSION				
Data	**1000**	**2000**	**3000**	**4000**	**5000**	**1000**	**2000**	**3000**	**4000**	**5000**
PSNR(dB)	71,06031	68,00686	66,23487	65,00085	64,04919	74,79828	72,45895	70,83738	69,67748	68,81696
MSE(dB)	0,00509	0,01029	0,01536	0,02056	0,02560	0,00215	0,00369	0,00536	0,00700	0,00854
SSIM	0,999992	0,999978	0,999887	0,999949	0,999931	0,999998	0,999995	0,999991	0,999987	0,999982

Table 5 compares regular LSB with the proposed LSB method in this paper for the third cover, Pepper, sized at 512 x 512 pixels. According to this table, all metric values exhibit improvement. PSNR and SSIM are within an optimal range, with SSIM maintaining a value around 0.9999, and PSNR achieving a value of 68.81696 dB for data sized at 5000 bytes, clearly surpassing the regular LSB method.

Table 6 will display a comparison between images. In this table, only images that have had 5000 bytes of data embedded will be compared, as 5000 bytes is the largest amount of data experimented in this research and yielded the most impressive results. In this table, it is found that changes in images with embedded data are completely imperceptible to the naked eye when compared to the original images. So it can be said that the stego image has good quality and the damage it possesses is very minimal.

Table 6
Comparison of Original Images and Stego Images Embedded with 5000 Bytes of Data Text

Data	Image	Original Image	LSB Embedded Image	LSB-Brotli Embedded Image
5000 Bytes	Lena			
	Baboon			
	Pepper			

DISCUSSIONS

The research conducted in this paper resulted in a significant improvement in metric values across all cover images. When averaging the overall percentage from the three cover images, it can be concluded that there is a PSNR improvement of around 6.64%, followed by a substantial decrease in MSE, dropping by about 63.79%. Additionally, the SSIM value also increased by 0.0039%, maintaining a consistent value of 0.9999 even with the largest input data. This SSIM value demonstrates the success of this method in enhancing one of the fundamental concepts of steganography, Imperceptibility, where the stego image must have minimal visual differences and remain undetectable to the human eye (A. S. Ansari et al. , 2020).

The difference in values among the three different cover images could be influenced by the quality and distribution of color components within these cover images. Furthermore, from this testing, a pattern emerged showing that the larger the data input into the cover image, the more significant the compression applied, resulting in efficient usage of Brotli compression with larger-sized data. The input data is a contributing factor—if the embedded data contains many similar characters, more compression is applied, leading to significantly improved resulting quality. As previously explained, Brotli performs compression by identifying repetitive structures or patterns in the data and condenses them into a more concise form according to the references within the Brotli algorithm.

There are limitations to this study. The paper focused only on testing with image files, while steganography encompasses various types and mediums for embedding data. However, this study concentrated specifically on embedding within images, especially RGB-formatted images. The study did not test with exceptionally large data, limiting the method's applicability. Other constraints include using only 512 x 512-sized images as the embedding media, without exploring larger sizes that might be more effective. In Table 5, it is found that no changes are noticeable when observed with the naked eye. However, changes may become noticeable when using much larger amounts of data, but in this research, the tested data was limited to 5000 bytes.

CONCLUSION

The results from the writing above lead to the conclusion that the integration of LSB with Brotli compression was successfully achieved and provide good results compared to regular LSB. This demonstrates that this method has the capability to enhance image quality after text data embedding. The testing provided a clear comparison where Brotli compression significantly improved the quality of the cover image after embedding. Moreover, the integration of LSB with Brotli compression resulted in a 6.64% increase in PSNR, around 63.79% decrease in MSE, and an SSIM improvement around 0.0039%, maintaining an SSIM value of 0.9999. Testing also revealed that the highest compression ratio was achieved with the largest data size, specifically 5000 bytes in this paper, reaching a compression ratio of 66.55%. This aligns with the PSNR results for data ranging from 1000 to 5000 bytes, which consistently decreased. Thus, it can be said that Brotli compression performs well with large input data. However, this is also dependent on the embedded data. If the embedded data contains many identical characters or patterns, it induces greater compression.

REFERENCES

Adhi, C., & Husada, B. (2022). *Indonesian Journal of Mathematics and Natural Sciences*. *45*(1), 30–37.

Alifi, M. B., & Suartana, I. M. (2020). *Implementasi Teknik Steganografi pada Gambar JPEG dan PNG dengan menggunakan Metode Adaptive Minimum Error Least Significant Bit Replacement (AMELSBR)*. *02*, 113–119.

Anindita, F. (2019). *POSITION DETECTION OF SECRET MESSAGES FROM LSB-BASED*. *6*(1), 608–615.

Ansari, A., Mohammadi, M. S., & Parvez, M. T. (2019). *A Comparative Study of Recent Steganography Techniques for Multiple Image Formats*. *January*, 10–25. https://doi.org/10.5815/ijcnis.2019.01.02

Ansari, A. S., Mohammadi, M. S., & Parvez, M. T. (2020). *A Multiple-Format Steganography Algorithm for Color Images*. *8*. https://doi.org/10.1109/ACCESS.2020.2991130

Anshori, Y., & Dodu, A. Y. E. (2019). *SATIN – Sains dan Teknologi Informasi Aplikasi Steganografi pada Media Citra Digital Menggunakan Metode Least Significant Bit (LSB)*. *5*(1), 1–10.

Antony Joans Kumar M., Christopher Columbus C., Ben George E., S. C. C. (2023). *Ensuring Secure and Efficient Multi-Cloud Storage with Brotli Compression and Hierarchical Data Protection Compression and Hierarchical Data Protection*. 0–11.

Astuti, E. Z., Setiadi, D. R. I. M., Rachmawanto, E. H., Sari, C. A., & Sarker, M. K. (2020). *LSB-based Bit Flipping Methods for Color Image Steganography LSB-based Bit Flipping Methods for Color Image*. https://doi.org/10.1088/1742-6596/1501/1/012019

Baby, M., Sutera, M., Magdalena, R., & Hidayat, B. (2020). *STEGANALISIS MENGGUNAKAN METODE WAVELET DAN SVM DENGAN PENYISIPAN TEKS MELALUI APLIKASI STEGANOGRAPHY ANDROID STEGANALYSIS USES WAVELET AND SVM METHODS WITH TEXT MESSAGE INSERTION THROUGH THE ANDROID STEGANOGRAPHY APPLICATION*. *7*(2), 4244–4250.

Carlo, J., Arroyo, T., Journal, I., Arroyo, J. C. T., Espadero, J. A., Ganas, M. A., Ardeña, R. F., & Vilchez, R. N. (2020). *An Efficient Least Significant Bit Image Steganography with Secret Writing and Compression Techniques*. 3280–3286.

Horng, J. (2020). *Steganography Using Quotient Value Differencing and LSB Substitution for AMBTC Compressed Images*. 129347–129358. https://doi.org/10.1109/ACCESS.2020.3009232

Kumar, A., & Swain, G. (2022). High fidelity based reversible data hiding using modified LSB matching and pixel difference. *Journal of King Saud University - Computer and Information Sciences*, *34*(4), 1395–1409. https://doi.org/10.1016/j.jksuci.2019.07.004

Mahdi, S. A., & Khodher, M. A. (2021). *An Improved Method for Combine (LSB and MSB) Based on Color Image RGB*. *39*(01), 231–242.

Masruri, N. H., Sunyoto, A., Informatika, M. T., Amikom, U., Jl, Y., Road, R., & Catur, C. (2019). *Meningkatkan Keamanan Pesan Menggunakan Enkripsi Arnold Cat Map Dan Steganografi Pixel Value Differencing*. 113–118.

Nashat, D., & Mamdouh, L. (2019). *An efficient steganographic technique for hiding data*. *6*.

Reynaldo, V., Wicaksana, A., & Hansun, S. (2019). *Brotli data compression on moodle-based e-learning server*. *10*(11), 963–970. https://doi.org/10.24507/icicelb.10.11.963

Ruang, M., & Yuv, W. (2020). *STEGANOGRAFI CITRA LSB DENGAN METODE REGION GROWING*. *December 2019*.

Rustad, S., Ignatius, D. R., Setiadi, M., Syukur, A., & Andono, P. N. (2021). Inverted LSB image steganography using adaptive pattern to improve imperceptibility. *Journal of King Saud University - Computer and Information Sciences*, *xxxx*. https://doi.org/10.1016/j.jksuci.2020.12.017

Tayyeh, H. K., & Al-jumaili, A. S. A. (2022). *A combination of least significant bit and deflate compression for image steganography*. *12*(1), 358–364. https://doi.org/10.11591/ijece.v12i1.pp358-364

Utomo, Y. B., & Erwanto, D. (2019). *Analisa Teknik Steganografi dan Steganalysis Pada File Multimedia Menggunakan Net Tools dan Hex Editor*. *3*(1), 16–22.

Utomo, Y. B., Mukminna, H., Elektro, T., Teknik, F., Islam, U., & Kediri, K. (2021). *Penerapan Teknik Steganalysis Menggunakan Metode Chi Square Attack Pada Stego Image Berformat Jpeg Berbasis Android*. *1*(1), 51–58.

Wang, Y., Tang, M., & Wang, Z. (2020). Optik High-capacity adaptive steganography based on LSB and Hamming code. *Optik - International Journal for Light and Electron Optics*, *213*(January), 164685. https://doi.org/10.1016/j.ijleo.2020.164685