International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Enhancement of Wang-Yin-Wang Improved Least Significant Bit Algorithm Applied in Image Steganography

Harry Chris John C. Decapia[1], John Christopher D. Ilacad[2], Mark Anthony S. Mercado[3], Jonathan C. Morano[4], Mark Christopher R. Blanco[5]

[1,2,3,4,5]CISTM, Pamantasan ng Lungsod ng Maynila, Philippines

Abstract

Researchers modified the development of Wang-Yin-Wang LSB Algorithm because it has not been able to maintain all the bits of the hidden image since there were insufficient pixel values in the carrier image. Use of LZW Compression algorithm for the hidden image is one way to solve this issue. The results showed that after compressing, there is now much capacity inside the carrier image to hold a bigger hidden image without making it look significantly different. This approach makes it easy to insert compressed data into a carrier image. Furthermore, besides improving capacity of carrier image as evidenced by test results, suggested method outperforms Wang-Yin-Wang's algorithm in terms of visual quality of the stego image.

Keywords: LSB Algorithm, LZW Compression, Stego-Image, Image Steganography

INTRODUCTION

In today's age of digital technology, personal information security and data protection is critical [1]. Thus, there is an increasing need for secure methods to transfer confidential data. Steganography appears as a popular technique for maintaining data confidentiality [2],[3]. For this reason, steganography has become more and more pronounced in hiding information within diverse media, making it difficult for unauthorized users to discover the original data behind it [3]-[5]. As stated by [6], the best steganographic media have a high degree of redundancy such as image files and audio files. While little redundant data can be observed in text files, text steganography is less common. Additionally, because they are relatively simpler than audios or videos, images are often the preferred choice of researchers who want to hide some information in them [7]

Capacity, imperceptibility, and robustness are three fundamental requirements that must be met by image steganography techniques [8]-[11]. Least Significant Bit (LSB) algorithm is a spatial domain image steganographic algorithm which updates pixel data through insertion or substitution of bits to hide secret messages inside them [12]. However, the LSB algorithm lacks much robustness and thus its susceptibility can be easily detected during an attack situation.

Wang-Yin-Wang's improvement focuses on increasing the security and robustness of the original LSB algorithm. One modification of their algorithm is to use the Python random.sample function to

randomize the selection of pixels in the carrier image during the embedding process. Their experiments showed that the improved algorithm performs better than the original LSB. However, certain modifications can still be made to add features and improve the Wang-Yin-Wang's algorithm.

According to [13], large volumes of hidden data may be identified, although small quantities may remain undetected. Furthermore, each pixel only has one bit for embedding the hidden message, and the hidden image size is limited by the number of pixels. With these, the study aims to solve the problems to further improve the algorithm in terms of capacity and imperceptibility. The researchers will utilize the Lempel–Ziv–Welch (LZW) compression algorithm to reduce the size of the hidden image.

LITERATURE REVIEW

Image Steganography

The objective of image steganography is to hide the message within a carrier image by modifying its properties [2]. It involves applying a cover object to conceal the original message image, a host object to hold the message image and a steganography algorithm that implants the message image in the cover object. The output is a stego image containing the secret embedded file. Then send this stego image to the receiver who can use de-steganography algorithms to extract the original contents [14], as illustrated in Figures 1-2. Image steganography has two processes: embedding and extraction. Embedding refers to hiding data in an image while extraction means recovering data from it [15]-[17]. The carrier image is referred to as an original one, and with hidden data as a stego image [3].

Figure. 1. Steganography at enders' side

Figure. 2. De-steganography at receiver side

Least Significant Bit Algorithm

The smallest bit in a binary string is referred to as the least significant bit (LSB). The least significant bit (LSB) helps protect secret data by incorporating it into pixel values of carrier images. [18] investigated the Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) of LSB, MSB and NHB algorithms by comparing them. It was shown that image quality was better for LSB algorithm than for the other two algorithms. Nonetheless, NHB algorithm has higher complexity than both LSB and MSB algorithms which makes it more secure. Moreover, embedding different types of secret data such as docx files, xlsx files, txt file, pdfs etc., within the carrier image is easy because of simplicity and efficiency of the LSB algorithm.

As per [19] Least significant bit (LSB) algorithm is a simple and representative spatial-domain algorithm. The LSB algorithm has low complexity and high embedding capacity and is easy to implement. However, its robustness against steganalysis is not very strong. Therefore, many improved LSB algorithms have been proposed to solve this problem.

Wang-Yin-Wang Algorithm

[20] implemented an enhanced LSB algorithm using Python and OpenCV. Python, known for its simplicity and flexibility in image processing.

Pseudocode of Wang-Yin-Wang Algorithm

Embedding Process

1. Read the hidden and carrier images and convert it to grayscale using OpenCV.
2. Convert the hidden image to a binary stream and obtain the length of that stream.
3. Acquire the data for the preparation of the embedding from the number of rows and columns of the carrier image.
4. Generate a list of all pixel coordinates of the carrier image.
5. Based on this list, generate a random list of position sequences using Python's random.sample function. This list contains the pixel positions of the binary stream to be embedded in the carrier image.
6. Obtain the bit-plane of each pixel and replace the least significant bit with the bit values of the binary stream to be embedded at the specified bit-plane.

Extraction Process

1. Retrieve the carrier image with the hidden image embedded in it and convert it to a grayscale image.
2. Obtain the number of rows and columns of the carrier image and generate a list containing all pixel coordinates.
3. Obtain a list of position sequences for the embedded data based on the list of random sequences used previously.
4. Extract the binary stream of the hidden image from the least significant bits of each pixel in the carrier image according to this position sequence list.
5. Convert the extracted binary stream back into an image by reorganizing it into pixel form and converting it to a grayscale image.

Lempel-Ziv-Welch (LZW Compression)

In the study of [21], the researchers conduct an analysis between two different techniques. The first one is the use of an LSB algorithm with no compression and encryption. In the second technique they encrypt the data and compress it. As a result, an LSB algorithm without compression and encryption has a low security level and lower image quality than the other one. According to [22]-[24], LZW has the capability to compress a file to one-third of its original size. Fixed length code words are assigned to variable length input symbol sequences using LZW coding. A "dictionary" or "codebook" with the source symbols to be encoded serves as the foundation for the coding. When new symbol sequences are added, the initial dictionary used for coding is expanded [25].

METHODOLOGY

In this study, Lempel-Ziv-Welch (LZW) Compression will be integrated into the hidden image of the Wang-Yin-Wang algorithm.

Proposed Algorithm

I. Process of Carrier Image before Embedding

1. Convert carrier image into grayscale using OpenCV.
2. Produce a list of all pixel coordinates based on the number of rows and columns of the carrier image.
3. Generate a random list of sequences based on the pixel coordinates using Python's random.sample function.
4. Select each pixel in the generated random list to avoid data loss or duplicate embedding.
5. Convert each of the pixels from the randomized list to binary stream.

II. Process of Hidden Image before Embedding

1. Convert hidden image into grayscale using OpenCV.
2. Get the binary stream of numbers of the hidden image and obtain the length of that stream.
3. Compress the binary stream of numbers of the hidden image using the LZW compression algorithm to reduce its size and obtain the length of that stream.

III. Embedding Process

1. Get the binary stream of the randomized list pixel of the carrier image and the compressed binary stream of the hidden image.
2. Embed each bit value from every byte in the binary streams of the hidden image into the Least Significant Bit (LSB) of each byte in the randomized binary streams of the carrier image.
3. Display result or the stego image.

IV. Extraction Process

1. Retrieve the stego image and convert it to a grayscale image.
2. Determine the dimensions (number of rows and columns) of the carrier image, and then create a list that includes all pixel coordinates.
3. Obtain a list of position sequences for the embedded data based on the list of random sequences used previously.
4. Extract the binary stream from the least significant bits of each pixel in the carrier image according to the position sequence list.
5. Decompress the binary stream using the LZW compression algorithm to get the decompressed binary stream.
6. Convert the decompressed binary stream back into an image by reorganizing it into pixel form and converting it to a grayscale image.
7. Display the Hidden Image.

Dataset

The carrier images for this study were specifically selected from the collection at the University of Southern California's Signal and Image Processing Institute. The chosen images are named 'Female' as illustrated in Figure 3 and 'Baboon' as shown in Figure 4, each available in three different resolutions: 1024x1024 pixels (low resolution), 2048x2048 pixels (medium resolution), and 4096x4096 pixels (high resolution). For the hidden image as shown in Figure 5, the researchers only use generated QR Code type of images but also come with 360x360 pixels for the low resolution, 720x720 for the medium resolution, and 1024x1024 for the high resolution.

Figure.3. Female

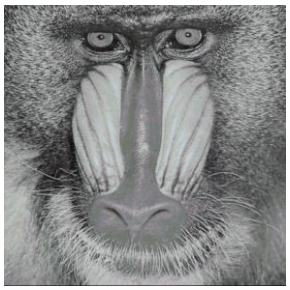

Figure.4. Baboon

Figure.5. Hidden Image

Metrics

The mean-square error (MSE) and the peak signal-to-noise ratio (PSNR) are used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error as in (1). The lower the value of MSE, the lower the error.

$$PSNR = 10\, log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (1)$$

MAX is the maximum number of values that can be entered for a pixel value for a picture or video, often 255, and MSE is the average of the Mean Squared Error (MSE), which is the total squares of the discrepancies between the source and the image in compression. MSE is determined as in (2)

$$MSE = \frac{1}{mn}\sum_{i=1}^{M}\sum_{j=1}^{N}[f(i,j) - g(i,j)]^2 \quad (2)$$

As the PSNR value increases, the variance decreases, indicating a higher quality of the image. In general, when 30dB or more of PSNR is present, there is barely any difference between the two pictures to the perception of sight [13]. If every pixel in the carrier image is utilized to embed the hidden image, the following formula in (3) can be used to determine the largest hidden image size that can be embedded using the traditional LSB method:

*Capacity = (Image width * Image height)/8 (Bytes)* (3)

RESULTS AND DISCUSSION

Capacity

Table 1. Compression Result

Hidden Image (px)	Uncompressed Bits	Compressed Bits
360 x 360	129,600	29,334
720 x 720	518,400	74,710
1024 x 1024	2,073,600	210,194

The results demonstrate in Table 1 that as a hidden image is compressed, the maximum image size that can be inserted into the carrier image's LSB will increase. This indicates that the compressed data can be hidden more easily within the carrier image because there is now more room available, a larger hidden image can be embedded within the carrier image without significantly altering its appearance. As a result, compression can improve the carrier image's ability to hide concealed data.

Imperceptibility

The stego image quality is determined by this criterion and indicates how closely the stego image matches the carrier image visual. An increase in the PSNR quality measure shows that the result has less distortion visual.

Table 2. PSNR Result of Wang-Yin-Wang's Algorithm

Hidden Image (px)	Carrier Image (px)	Female (dB)	Baboon (dB)
360 x 360	1024 x 1024	50.3168	50.4398
720 x 720	2048 x 2048	50.7862	51.3218
1024 x 1024	4096 x 4096	51.0923	50.3186

Table 3. PSNR Result of Proposed Algorithm

Hidden Image (px)	Carrier Image (px)	Female (dB)	Baboon (dB)
360 x 360	1024 x 1024	55.3152	55.3119
720 x 720	2048 x 2048	58.0726	59.5409
1024 x 1024	4096 x 4096	59.1410	57.6534

The results of the Wang-Yin-Wang LSB approach and the proposed technique to embed payloads of varying sizes in sample images of female and baboon by the size of three distinct resolutions are displayed in the Tables 2-3. The results show that in every examined situation, the improved LSB approach yields greater PSNR values. Additionally, the PSNR values vary depending on which image is evaluated using each method.

CONCLUSION AND RECOMMENDATION

The goal of this study was to improve the Wang-Yin-Wang LSB Algorithm research by an enhancement. The advancement that is used in the suggested improvements is the use of LZW compression. The method produced good results when applied in the existing algorithm, as predicted theoretically. The improved LSB method outperforms the current algorithm, as demonstrated by the

testing results. LZW compression gives the carrier image more capacity to accommodate a larger concealed image without significantly altering its appearance. In addition, the stego-image has less visual distortion as it produced a greater PSNR values as shown in Table 3. Consequently, AI in Artificial Intelligence should be used to make image steganography more secure, reliable, and productive, as it involves embedding, encryption and detection avoidance. Yet the ethical implications and potential abuse of these advancements must also be taken into consideration.

ACKNOWLEDGEMENT

This research's success is due to the outstanding assistance and inspiration offered by several individuals and organizations. We want to express our deepest gratitude to everyone who inspired and encouraged us on this journey.

We would like to thank our thesis supervisor, Mr. Mark Anthony S. Mercado, for his great assistance and input, which helped us develop our study. We are particularly appreciative of the panelists, Mr. Jonathan Morano and Mr. Mark Christopher Blanco, for providing informative input during the defense, which improved the quality of our work. Finally, we want to thank everyone who helped make this research a success, no matter the size of their contribution. Your help has been irreplaceable and greatly appreciated.

REFERENCES

1. Evsutin, O., Melman, A., & Meshcheryakov, R. (2020). Digital steganography and watermarking for digital images: A review of current research directions. IEEE Access, 8, 166589-166611.
2. Tiwari, A., Shankar, G., & Jain, B. B. (2021). Comparative Analysis of Different Steganography Technique for Image Security. International Journal of Engineering Trends and Applications (IJETA, 8(2), 6-9.
3. Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image steganography: A review of the recent advances. *IEEE access*, *9*, 23409-23423.
4. Sultana, S., Khanam, A., Islam, M. R., Nitu, A. M., Uddin, M. P., Afjal, M. I., & Rabbi, M. F. (2018). A modified filtering approach of LSB image steganography using stream builder along with AES encryption. HBRP recent trends in information technology and its applications, 1(2), 1-10.
5. Milosav, P., Milosavljević, M., & Banjac, Z. (2023). Steganographic Method in Selected Areas of the Stego-Carrier in the Spatial Domain. *Symmetry*, *15*(5), 1015.
6. Hussain, M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T., & Jung, K. H. (2018). Image steganography in spatial domain: A survey. *Signal Processing: Image Communication*, *65*, 46-66.
7. Ansari, A. S., Mohammadi, M. S., & Parvez, M. T. (2019). A comparative study of recent steganography techniques for multiple image formats. International Journal of Computer Network and Information Security, 11(1), 11-25.
8. Bai, J., Chang, C. C., Nguyen, T. S., Zhu, C., & Liu, Y. (2017). A high payload steganographic algorithm based on edge detection. Displays, 46, 42-51.
9. Vanmathi, C., & Prabu, S. (2018). Image steganography using fuzzy logic and chaotic for large payload and high imperceptibility. International Journal of Fuzzy Systems, 20, 460-473.
10. Ghosal, S. K., Chatterjee, A., & Sarkar, R. (2021). Image steganography based on Kirsch edge detection. Multimedia Systems, 27(1), 73-87.

11. Singla, D., & Juneja, M. (2014, March). An analysis of edge-based image steganography techniques in spatial domain. In 2014 Recent Advances in Engineering and Computational Sciences (RAECS) (pp. 1-5). IEEE.
12. Poornima, R., & Iswarya, R. J. (2013). An overview of digital image steganography. International Journal of Computer Science & Engineering Survey (IJCSES), 4(1), 23-31.
13. Al-Huwais, N. H., Atiyah, Y. A., Parvin, S., & Gawanmeh, A. (2020, October). An Improved Least Significant Bit Image Steganography Method. In *2020 Fourth International Conference on Multimedia Computing, Networking and Applications (MCNA)* (pp. 90-96). IEEE.
14. Akhtar, N., Khan, S., & Johri, P. (2014, February). An improved inverted LSB image steganography. In 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) (pp. 749-755). IEEE.
15. Ashraf, Z., Roy, M. L., Muhuri, P. K., & Lohani, Q. D. (2020). Interval type-2 fuzzy logic system-based similarity evaluation for image steganography. Heliyon, 6(5).
16. Juneja, M., & Sandhu, P. S. (2013, January). An improved LSB based Steganography with enhanced Security and Embedding/Extraction. In *3rd International Conference on Intelligent Computational Systems, Hong Kong China* (pp. 29-34).
17. Ilaga, K. R., Sari, C. A., & Rachmawanto, E. H. (2018). A high result for image security using crypto-stegano based on ECB mode and LSB encryption. *Journal of Applied Intelligent System*, *3*(1), 28-38.
18. Wai, Y. Y., & Myat, E. E. (2018). Comparison of LSB, MSB and New Hybrid (NHB) of steganography in digital image. International Journal of Engineering Trends and Applications, 5(4), 16-19.
19. Wang, J., Cheng, M., Wu, P., & Chen, B. (2019). A survey on digital image steganography. Journal of Information Hiding and Privacy Protection, 1(2), 87.
20. Wang, S., Yin, H., & Wang, X. (2023). Research on the Improvement of LSB-based Image Steganography Algorithm. Academic Journal of Science and Technology, 5(3), 222-224.
21. AbdelWahab, O. F., Hussein, A. I., Hamed, H. F., Kelash, H. M., Khalaf, A. A., & Ali, H. M. (2019). Hiding data in images using steganography techniques with compression algorithms. TELKOMNIKA (Telecommunication Computing Electronics and Control), 17(3), 1168-1175.
22. Jayasankar, U., Thirumal, V., & Ponnurangam, D. (2021). A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. Journal of King Saud University-Computer and Information Sciences, 33(2), 119-140.
23. Ruxanayasmin, B., Krishna, B. A., & Subhashini, T. (2013). Implementation of data compression techniques in mobile ad hoc networks. *International Journal of Computer Applications*, *80*(8).
24. Kausar, S., Habib, M., Shabir, M. Y., Ullah, A., Xu, H., Mehmood, R., ... & Iqbal, M. S. (2020). Secure and efficient data transfer using spreading and assimilation in MANET. *Software: Practice and Experience*, *50*(11), 2095-2109.
25. Rao, S., Prof, N.A., Rao, K., & Prof, A. (2020). A NEW APPROACH TO INCREASE LZW ALGORITHM COMPRESSION RATIO. *International Journal of Engineering Applied Sciences and Technology*.