INTEK Jurnal Penelitian
Vol. 12, No. 1, pp. 34-41, April 2025

Reversible Data Hiding Algorithms in Securing Digital Image Artwork

I Made Aditya Putra Sadewa [1, a] Bambang Ari Wahyudi [2, *, b] Irma Palupi [3, c]
[1,2,3]Informatics, School of Computing, Telkom University, Jl. Telekomunikasi No. 1, Bandung, Jawa Barat, 40257, Indonesia
[*,b] bambangari@telkomuniversity.ac.id (Corresponding Author), [a]apsadewa@student.telkomuniversity.ac.id
[c]irmapalupi@telkomuniversity.ac.id

***Abstract*— In today's digital era, protecting digital artworks, particularly images, has become increasingly important to prevent copyright infringement and forgery. This paper proposes a novel method for embedding secret data into images using Reversible Data Hiding (RDH) techniques that leverage histogram shifting and random sub-blocks. The method is designed to maintain the visual integrity of the image while allowing the insertion of critical information, such as copyright metadata. The dataset used consists of 13 digital artworks sized 1280x720 pixels in PNG format, reflecting a diversity of textures and colors. Experimental results demonstrate that the proposed method achieves a high embedding capacity with PSNR values exceeding 37 dB, indicating excellent image quality post data insertion. Additionally, the method exhibits resilience against illegal modifications, with the ability to detect changes in images that have had data embedded. By integrating a PIN-based authentication system, the method enhances the security and integrity of the embedded information. This research makes a significant contribution to the field of digital artwork protection, offering an effective solution to preserve the authenticity and aesthetic value of images while enabling secure and reversible data insertion. The findings underscore the potential of RDH techniques in safeguarding sensitive information across various applications, ensuring that digital artworks can be both protected and enjoyed without compromising their quality.**

Keywords—authenticator; pin; steganography; RDH; SSIM; PSNR

I. Introduction

In the field of digital art, especially digital images, every work represents the unique creative expression of the creator [1]. This digital image art is not only a medium of artistic expression but also a valuable asset that can be easily accessed through various online platforms. However, this ease of access also brings serious challenges, such as the risk of copyright infringement, plagiarism, and counterfeiting. In this context, it is important to develop effective approaches to protect the authenticity and value of digital image artworks.

Watermarking allows embedding an image, text, PDF, audio, or video within a multimedia entity to prevent misuse and protect copyright [2]. Steganography, also known as hidden watermarking, is one of the reliable techniques to overcome these challenges. Steganography is the art of hiding some secret data by embedding it into an unclassified host medium [3]. This technique allows the insertion of secret information, such as copyright metadata or ownership marks, directly into a digital image while maintaining its visual quality. This is particularly relevant in the world of digital art, where the aesthetic and artistic value of the work must be maintained. One method of steganography is Reversible Data Hiding (RDH). This method allows information to be inserted into digital media without damaging the original visual elements. RDH typically utilizes approaches such as lossless compression [4], [5], [6], [7], [8] difference expansion [9], [10], [11], histogram shifting [12], [13], [14], prediction error expansion [15], [16], [17]. With these features, RDH becomes an ideal solution for protecting digital image art while ensuring its artistic integrity is preserved.

Huang et al. [18] proposed a new method for RDH in encrypted domains that uses stream encryption and permutation algorithms to maintain the correlation between neighboring pixels. The dataset used consists of twelve images of 512 x 512 pixels. The experimental results show varying embedding capacities, with high PSNR values reaching over 48 dB. Fadlan et al. [19]

DOI: http://dx.doi.org/10.31963/intek.v12i1.5174

proposed a modified RDH method using dynamic permutation to increase security against known plaintext attacks, by applying the Difference Histogram Shifting (DHS) algorithm to encrypted medical images. The dataset used consists of twelve medical images measuring 1024 x 1024 pixels. Tiwa et al. [20] proposed a digital diploma validation system using the RDH method with histogram shifting and random sub-block insertion techniques. The dataset used consists of 20 digital diplomas of 1122 x 792 pixels in PNG format, designed with various colors and patterns to test the compatibility of the system. The experimental results show good EC and high PSNR values, with PSNR values over 48 dB and SSIM above 0.99, indicating excellent image quality after data insertion.

This research develops a data insertion method in digital images with a histogram shifting-based approach to maintain visual quality and data integrity. The process begins with the division of the image into sub-blocks, where the sub-blocks used for data insertion are randomly selected using a PIN as input to enhance security. Histogram shifting is used to modify the pixel values in certain blocks to enable binary data insertion without damaging the image. Pixels equal to 0 or 255 can cause overflow/underflow [18], and this approach includes mechanisms to handle distortions such as overflow and underflow. The validation stage is performed using the extracted image to obtain the index list S, which is then compared with the original image. If the index maps of the extracted image and the original image are identical, the method is considered successful in maintaining the integrity of the inserted data.

This method seeks to enhance the security and integrity of the hidden information by guaranteeing that only authorized persons can access and extract the embedded message. By integrating these two methodologies, we aim to build a robust architecture that provides secure embedding of data within the obfuscation medium while authenticating the identity of individuals who wish to access the data.

II. Research Methodology

The proposed method features three main stages: embedding, extraction, and validation. The difference from the previous method is that we have integrated an authentication method into Reversible Data Hiding (RDH).

Embedding is the process of inserting a message into an image. Extracting involves retrieving the previously inserted message and reconstructing the modified image pixels to closely approximate the original image before modification. Validating is an additional stage beyond the RDH method, designed to test the reliability of this approach. This stage is conducted by comparing the S-list of the embedded image with that of the reconstructed image, ensuring both the integrity of the inserted data and the authenticity of the image.

A. Dataset

The dataset used in this study consists of 13 unique digital artworks with a resolution of 1280x720 pixels in PNG format. All images in the dataset are in color, using the RGB (Red, Green, Blue) model, which allows full color analysis and processing. Red, Green, and Blue have a range of 0-255 [21]. The selection of a dataset with such resolution ensures visual quality that is detailed enough for the data embedding process, while the PNG format was chosen because it supports compression without data loss. In addition, the use of digital artwork reflects the diversity of textures, colors, and patterns, thus enabling evaluation of the method on different types of images with complex visual characteristics. The dataset used in this study was obtained from the open-source site Pixabay, which provides a wide collection of images for free [22]. The media available on this platform are freely licensed, making it possible to utilize them for research and testing purposes.

B. Embedding

The first step in the embedding process is to determine the block size of the image to be used. Once the block size is defined, the image is divided into 3x3 pixel-sized sub-blocks. If the image dimensions are insufficient to form 3x3 sub-blocks, the image is modified by adjusting its pixel size. This adjustment ensures the image can be

DOI: http://dx.doi.org/10.31963/intek.v12i1.5174

processed with a consistent sub-block structure, allowing each part of the image to be processed effectively in the subsequent stages without losing essential information. This modification maintains the compatibility of the image throughout the processing workflow. Figure 1 shows the flow of the embedding stage.

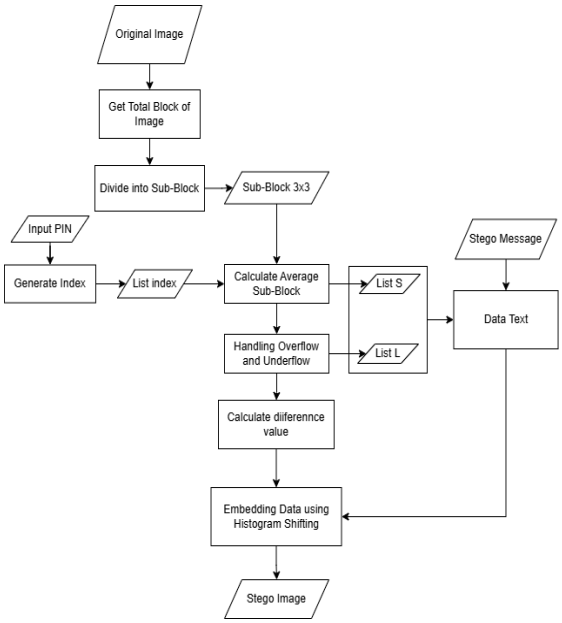

Figure 1. Embedding Process

Each sub-block is denoted by $C_{i,j}$ (for $1 \leq i \leq N$. $1 \leq j \leq 9$), where N represents the number of sub-blocks. In this embedding method, the PIN establishes the initial value (seed), which is subsequently utilized to generate a random seed. The result of the random seed will function as a list index that determines the location of the sub-block. The PIN is an essential element that ensures consistency in the random number generation process. After obtaining the list of sub-blocks through a random generator, we check the probability value of each sub-block. Only sub-blocks with a probability above the threshold will be selected. This requirement serves to randomly filter out potential locations while remaining consistent with the same PIN input. The value of a used in this research is 0.99. The selected sub-block will perform calculations using equation (1). This calculation aims to produce a list S, which will be used as the average pixel value of the sub-block.

$$S = \left\{ append\left(\left| \frac{sum(C_i)}{9} \right| \right) if\ prob_i > a \right. \quad (1)$$

To prevent overflow/underflow of pixels, modifications are made using formula (2). The next step is to convert the pixel values in the sub-block based on a certain range of values. Pixel value {1, 254} will be converted into binary value '0', while pixel value {0, 255} will be converted into binary value '1'. The results of this conversion are then organized into a new list called list L.

$$C'_{i,j} = \begin{cases} 254\ if\ C_{i,j} = 255 \\ 1\ if\ C_{i,j} = 0 \\ C_{i,j}\ otherwise \end{cases} \quad (2)$$

After modifying the pixel values, the secret message is inserted into the encrypted image by manipulating the histogram difference. The calculation of the difference value in each block is done based on Equation (3) to determine the message insertion location.

$$D_{i,j} = C'_{i,j} - C'_{i,1} \quad (3)$$

Equation (4) is used in the final stage of the embedding process, serving to insert the message bits into the image.

$$C''_{i,j} = \begin{cases} C'_{i,j} - 1\ if\ D_{i,j} < -1 \\ C'_{i,j} - b\ if\ D_{i,j} = -1 \\ C'_{i,j} + b\ if\ D_{i,j} = 0 \\ C'_{i,j} + 1\ if\ D_{i,j} > 0 \end{cases} \quad (4)$$

The embedding process is performed only if the pixel difference value in the sub-block meets a certain condition, which is equal to -1 or 0.

C. Extracting

The extraction process aims to retrieve the data that has been inserted into the image while ensuring that the data remains intact and error-free. This process is crucial to maintain the accuracy and integrity of the hidden data, ensuring that the extracted information matches the original data before insertion. Figure 2 shows the flow of the extraction process.

DOI: http://dx.doi.org/10.31963/intek.v12i1.5174

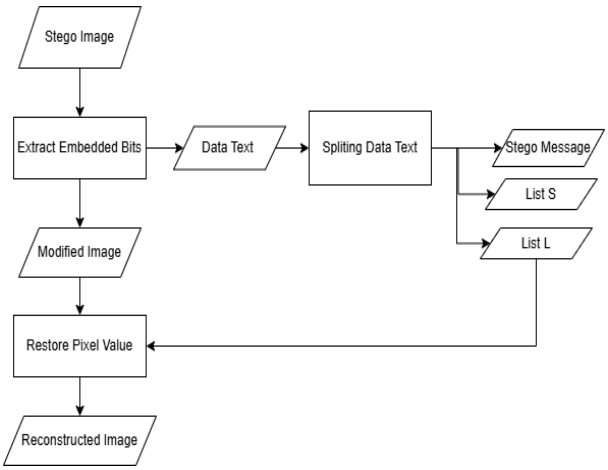

Figure 2. Extracting Process

The data bit extraction process is performed using formula (5). After that, data splitting is performed to separate the inserted information into several parts, namely the stego message, L list, and S list. The stego message itself contains important information, such as the name of the digital artwork creator, the year of creation, and the title of the image.

$$b^* = \begin{cases} 0 \; if \; C''_{i,j} - \; C''_{i,1} = 0, -1 \\ 1 \; if \; C''_{i,j} - \; C''_{i,1} = 1, -2 \end{cases} \qquad (5)$$

The next step is to restore the pixels in the image that have been modified during the previous embedding stage, using formula (6).

$$C'^*_{i,j} = \begin{cases} C''_{i,j} - \; 1 \; if \; C''_{i,j} - \; C''_{i,1} > 0 \\ C''_{i,j} + 1 \; if \; C''_{i,j} - \; C''_{i,1} < -1 \\ \quad C''_{i,j} \; otherwise \end{cases} \qquad (6)$$

List L is used as the main reference in the overall image pixel restoration process. Using formula (7), this list helps to ensure that every pixel that has been modified in the previous stage can be restored to its original state. This process is important to maintain visual integrity and ensure that the image remains true to its original form after going through various stages of manipulation.

$$C'^*_{i,j} = \begin{cases} 0 \; if \; C'^*_{i,j} \; = 1 \\ 255 \; if \; C'^*_{i,j} = 254 \\ \quad C'^*_{i,j} \; otherwise \end{cases} \qquad (7)$$

The extraction stage produces two main outputs. The first output is the extracted data, which serves to verify the compatibility between the embedding and extraction processes, ensuring that both processes have performed as expected without loss of information. The second output is the restored image, which is saved for use in the testing phase of the validation process.

D. Validating

Validation is done to test whether the modified and restored images have pixel quality that is identical to the original image. This validation process can be seen in Figure 3.

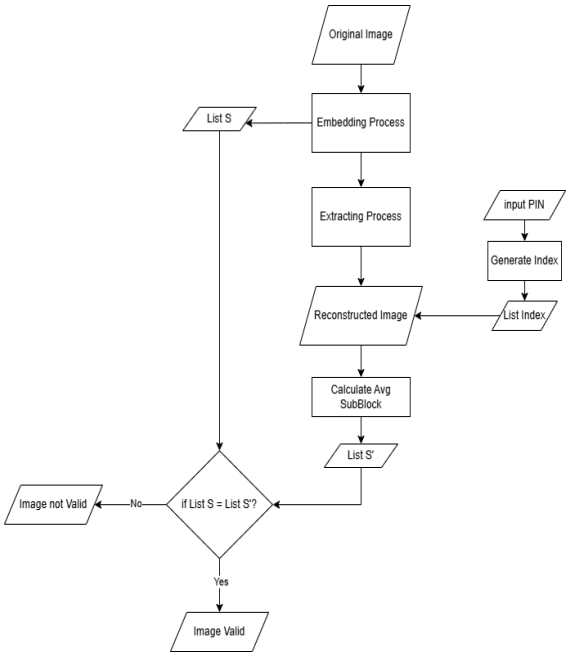

Figure 3. Validating Process

In this stage, the restored image from the extraction process goes through an S-list search using the same PIN as in the embedding stage. Next, equation (1) is applied to process the list. After that, it is checked whether the S-list generated from the restored image is exactly the same as the S-list from the original image. If the results are the

DOI: http://dx.doi.org/10.31963/intek.v12i1.5174

same, the image is considered valid as it has identical pixel quality to the original image, indicating the success of the embedding, extraction, and restoration process as a whole.

III. Results and Discussion

In this section, we carry out three main stages of analysis. The first stage is to test the image quality to ensure visual integrity and ensure that the data embedding process does not significantly reduce the quality of the media. The second stage is to test the security aspects by applying various attack strategies on the image, to evaluate the system against image modification attempts. The third stage detects the storage capacity of each dataset image.

A. Image Quality

This section analyzes image quality using two main metrics, namely Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). PSNR is used to measure the degree of numerical distortion [23] between the original image and the reconstructed image, while SSIM is used to evaluate the structural similarity, luminance, and contrast between the two images. Table 1 shows the visual quality of the reconstructed images.

Table 1. Visual Quality of Reconstructed Image

No.	Image	PSNR (dB)	SSIM
1	Valley	45.45	0.9945
2	Tundra	50.45	0.9977
3	Lake	47.20	0.9955
4	Horses	42.76	0.9915
5	Rural	47.89	0.9962
6	Fishing	38.72	0.9884
7	Morning Fog	49.03	0.9980
8	Nature’s Dew	39.45	0.9876
9	Heaven	45.70	0.9956
10	Giant Mountain	49.44	0.9968
11	Storm	48.38	0.9976
12	Iceberg	54.50	0.9986
13	Waterfall	37.22	0.9811

The image quality test results show that the PSNR values range from 37.22 dB to 54.50 dB, while the SSIM values range from 0.9811 to 0.9986. In general, a high PSNR value indicates that the resulting image quality has a low level of distortion compared to the original image. In this test, the highest PSNR value of 54.50 dB indicates excellent image reconstruction quality, with little noise or insignificant changes. In contrast, the lowest PSNR value of 37.22 dB is still within the acceptable range, although it shows a greater degree of distortion compared to the other results. In terms of SSIM, values ranging from 0.9811 to 0.9986 indicate that the test images have a very high structural similarity with the original image. The highest SSIM value of 0.9986 reflects that the structure, luminance, and contrast of the image are almost identical to the original image, while the lowest value of 0.9811 still shows good visual quality despite the slight differences.

PSNR is generally considered poor if its value is less than 20 dB, and is considered excellent if it exceeds 30 dB [24]. On the other hand, SSIM values range from 0 to 1, where larger values indicate smaller image distortion, while a value of 1 indicates that the two images are identical [25]. Overall, the combination of PSNR and SSIM values in this test shows that the method used is able to maintain image quality well, both in terms of distortion reduction and visual structure preservation. These results prove that the tested system has a reliable performance in producing high-quality images.

B. Security Analysis

This security analysis aims to test the system's robustness against modifications to the embedded image by detecting changes to the inserted message. The test is conducted through three experimental scenarios:

1. The first experiment involves making simple scribbles on the embedded image to simulate small changes to the visual content.

DOI: http://dx.doi.org/10.31963/intek.v12i1.5174

2. The second experiment added noise evenly to the embedded image to measure the system's tolerance to random noise.
3. The third experiment modifies the brightness and contrast levels of the embedded image to evaluate the sensitivity of the system to changes in visual parameters.

Table 2. Security Analysis of Experiments 1-3

No.	Image	Experiment 1	Experiment 2	Experiment 3
1	Valley	Not Detected	Detected	Detected
2	Tundra	Detected	Detected	Detected
3	Lake	Detected	Detected	Detected
4	Horses	Not Detected	Detected	Detected
5	Rural	Detected	Detected	Detected
6	Fishing	Detected	Detected	Detected
7	Morning Fog	Detected	Detected	Detected
8	Nature's Dew	Detected	Detected	Detected
9	Heaven	Not Detected	Detected	Detected
10	Giant Mountain	Detected	Detected	Detected
11	Storm	Not Detected	Detected	Detected
12	Iceberg	Detected	Detected	Detected
13	Waterfall	Detected	Detected	Detected

Table 2 presents the test results of various experimental scenarios applied to the developed method. The detected label indicates that the system successfully detected the modification of the image through the difference between the extracted message and the original message. Conversely, the not detected label indicates that the system did not detect any modification in the modified image. The test results show that in the second experiment with the addition of noise and the third experiment with the modification of brightness and contrast, the system was able to detect all modifications to the images from the dataset used. However, in the first experimental scenario with simple graffiti, there were some images that were not detected as modified.

Based on these results, it can be concluded that the developed method has good robustness against significant modifications, such as noise and changes in visual parameters, but shows weakness against simple modifications, such as light scribbles on images.

C. Capacity Analysis

This data embedding capacity analysis plays an important role in evaluating the efficiency of the method used, especially in maintaining a balance between the amount of data that can be embedded and the resulting image quality. In addition, this analysis also aims to determine the average number of bits that can be inserted in a 1280x720 pixel RGB image.

Table 3. Embedding Capacity (EC)

No.	Image	EC
1	Valley	275.642
2	Tundra	336.990
3	Lake	365.951
4	Horses	321.968
5	Rural	310.115
6	Fishing	219.996
7	Morning Fog	447.171
8	Nature's Dew	525.734
9	Heaven	371.182
10	Giant Mountain	488.991
11	Storm	438.923
12	Iceberg	469.970
13	Waterfall	209.157

The dataset used consists of digital artwork images with a wide range of color variations, which significantly affect the data EC, as shown in Table 3. The high color variation in these digital artworks contributes to pixel

DOI: http://dx.doi.org/10.31963/intek.v12i1.5174

intensity variations, directly impacting the capacity for data insertion. To observe the EC, we count the number of neighboring pixels that have an intensity value difference between 0 and -1 [19]. The test results show that the data EC varies across images, with the lowest EC value being 209,157 and the highest EC value reaching 525,734. This variation in EC can be attributed to the diverse color distribution and visual complexity of the images, which create differing opportunities for embedding data without compromising image quality.

IV. Conclusion

This research successfully developed an effective Reversible Data Hiding (RDH) method for protecting digital artwork, with results showing that the visual quality of the image is maintained, as evidenced by PSNR values ranging from 37.22 dB to 54.50 dB and SSIM reaching 0.9986. The method also shows good resistance to image modification, although there is a weakness in detecting small changes, such as light scribbles. Moreover, the data insertion capacity varies between 209,157 bits to 525,734 bits, depending on the visual complexity of the image. Overall, this research makes a significant contribution to the protection of copyright and authenticity of digital artworks, and opens up opportunities for further development in secure and efficient data insertion techniques. Recommendations for future research include improving the detection of minor modifications by means such as marking each pixel in the image to monitor the pixel changes that occur. This approach is important as it allows the identification of very small changes. By marking the pixels, the system will be able to detect the difference between the original image and the modified image, even if the changes are not visible. This can also increase the robustness of the method against attacks that attempt to hide modifications, thereby increasing the reliability and accuracy of the method in preserving the authenticity of digital images.

Acknowledgement

We would like to thank all those who have contributed to this research. Thank you to Telkom University for providing facilities and support for this research. We are also grateful to fellow researchers who have provided valuable inputs during the research process. Finally, we appreciate all the resources available online that have helped in data collection and references for this research.

References

[1] X. Han, Y. Wu, and R. Wan, "A Method for Style Transfer from Artistic Images Based on Depth Extraction Generative Adversarial Network," *Appl. Sci.*, vol. 13, no. 2, 2023, doi: 10.3390/app13020867.

[2] Ramyashree, P. S. Venugopala, S. Raghavendra, and B. Ashwini, "CrypticCare: A Strategic Approach to Telemedicine Security Using LSB and DCT Steganography for Enhancing the Patient Data Protection," *IEEE Access*, vol. 12, no. July, pp. 101166–101183, 2024, doi: 10.1109/ACCESS.2024.3430546.

[3] L. Liu, L. Tang, and W. Zheng, "Lossless Image Steganography Based on Invertible Neural Networks," *Entropy*, vol. 24, no. 12, pp. 10816–10825, 2022, doi: 10.3390/e24121762.

[4] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," *Secur. Watermarking Multimed. Contents III*, vol. 4314, no. October, pp. 197–208, 2001, doi: 10.1117/12.435400.

[5] W. Zhang, B. Chen, and N. Yu, "Improving various reversible data hiding schemes via optimal codes for binary covers," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 2991–3003, 2012, doi: 10.1109/TIP.2012.2187667.

[6] W. Zhang, X. Hu, X. Li, and N. Yu, "Recursive histogram modification: Establishing equivalency between reversible data hiding and lossless data compression," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2775–2785, 2013, doi: 10.1109/TIP.2013.2257814.

[7] C. Qin, C. C. Chang, and Y. C. Chen, "Efficient reversible data hiding for VQ-compressed images based on index mapping mechanism," *Signal Processing*, vol. 93, no. 9, pp. 2687–2695, 2013, doi: 10.1016/j.sigpro.2013.03.036.

[8] Z. Tang, S. Xu, H. Yao, C. Qin, and X. Zhang, "Reversible data hiding with differential compression in encrypted image," *Multimed. Tools Appl.*, vol. 78, no. 8, pp. 9691–9715, 2019, doi: 10.1007/s11042-018-6567-3.

[9] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, 2003, doi: 10.1109/TCSVT.2003.815962.

[10] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, 2004, doi: 10.1109/TIP.2004.828418.

[11] L. Kamstra and H. J. A. M. Heijmans, "Reversible data

embedding into images using wavelet techniques and sorting," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2082–2090, 2005, doi: 10.1109/TIP.2005.859373.

[12] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–361, 2006, doi: 10.1109/TCSVT.2006.869964.

[13] X. Li, W. Zhang, X. Gui, and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 7, pp. 1091–1100, 2013, doi: 10.1109/TIFS.2013.2261062.

[14] Z. Tang, S. Xu, D. Ye, J. Wang, X. Zhang, and C. Yu, "Real-time reversible data hiding with shifting block histogram of pixel differences in encrypted image," *J. Real-Time Image Process.*, vol. 16, no. 3, pp. 709–724, 2019, doi: 10.1007/s11554-018-0838-0.

[15] B. Ou, X. Li, Y. Zhao, R. Ni, and Y. Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, 2013, doi: 10.1109/TIP.2013.2281422.

[16] Q. Ying, Z. Qian, X. Zhang, and D. Ye, "Reversible Data Hiding with Image Enhancement Using Histogram Shifting," *IEEE Access*, vol. 7, pp. 46506–46521, 2019, doi: 10.1109/ACCESS.2019.2909560.

[17] D. M. Thodi, J. J. Rodríguez, and S. Member, "Expansion Embedding Techniques for Reversible Watermarking," vol. 16, no. 3, pp. 721–730, 2007.

[18] F. Huang, J. Huang, and Y. Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 12, pp. 2777–2789, 2016, doi: 10.1109/TIFS.2016.2598528.

[19] M. F. Putranto, A. M. Barmawi, and B. A. Wahyudi, "Permutation Modification of Reversible Data Hiding Using Difference Histogram Shifting in Encrypted Medical Image," *Procedia Comput. Sci.*, vol. 135, pp. 727–735, 2018, doi: 10.1016/j.procs.2018.08.212.

[20] T. Ramdhani, B. A. Wahyudi, and P. E. Yunanto, "Validation System with Reversible Data Hiding in Digital Diplomas," *2022 10th Int. Conf. Inf. Commun. Technol. ICoICT 2022*, pp. 124–128, 2022, doi: 10.1109/ICoICT55009.2022.9914852.

[21] S. Rahman, J. Uddin, H. U. Khan, H. Hussain, A. A. Khan, and M. Zakarya, "A Novel Steganography Technique for Digital Images Using the Least Significant Bit Substitution Method," *IEEE Access*, vol. 10, no. November, pp. 124053–124075, 2022, doi: 10.1109/ACCESS.2022.3224745.

[22] Pixabay, "Pixabay," 2024. [Online]. Available: https://www.pixabay.com. [Accessed: 10-Jul-2024].

[23] B. V. Gajera, S. R. Kapil, D. Ziaei, J. Mangalagiri, E. Siegel, and D. Chapman, "CT-Scan Denoising Using a Charbonnier Loss Generative Adversarial Network," *IEEE Access*, vol. 9, pp. 84093–84109, 2021, doi: 10.1109/ACCESS.2021.3087424.

[24] Y. H. Chuang, B. S. Lin, Y. X. Chen, and H. J. Shiu, "Steganography in RGB Images Using Adjacent Mean," *IEEE Access*, vol. 9, pp. 164256–164274, 2021, doi: 10.1109/ACCESS.2021.3132424.

[25] A. K. Panigrahy *et al.*, "A Faster and Robust Artificial Neural Network Based Image Encryption Technique With Improved SSIM," *IEEE Access*, vol. 12, no. January, pp. 10818–10833, 2024, doi: 10.1109/ACCESS.2024.3353294.