Journal of Information Hiding and Multimedia Signal Processing ©2025 ISSN 2073-4212
Ubiquitous International Volume **16**, Number **3**, September **2025**

RDJAT-Embed: Efficient Spatial Domain Data Hiding via Average Pixel Modulation and Tracing Arrays

Reynandriel Pramas Thandya[1], Dea Kristin Ginting[1], Rafif Aydin Ahmad[1], Stefanus Yosua Mamamoba[1], Ntivuguruzwa Jean De La Croix[1,2], Tohari Ahmad[1*], Kaberuka David[3]

[1]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[2]African Center of Excellence in Internet of Things, College of Science and Technology University of Rwanda, Kigali 3900, Rwanda
[3]School of Education, University of Kigali, Kigali 2611, Rwanda
5025231113@student.its.ac.id, 5025231040@student.its.ac.id, 5025231198@student.its.ac.id, 5025231066@student.its.ac.id, 7025221024@student.its.ac.id, tohari@its.ac.id, dkaberuka@uok.ac.rw

*Corresponding author: Tohari Ahmad
Received June 11, 2025, revised July 18, 2025, accepted July 21, 2025.

Abstract. *Steganography is essential for secure data transmission, as it conceals information within digital media to ensure confidentiality and integrity. However, many existing techniques face challenges in balancing embedding capacity, imperceptibility, and computational efficiency. This paper presents RDJAT-Embed, a novel steganography method based on a modified Difference Expansion (DE) technique that significantly enhances payload capacity while preserving high image quality. Unlike conventional DE methods that use pixel-pair differences, RDJAT-Embed embeds data by calculating the difference between each pixel and its group average, combined with a five-value grouping strategy, to minimize distortion and prevent pixel overflow or underflow. The method achieves an average peak signal-to-noise ratio of 53.683 decibels and a structural similarity index measure of nearly 0.999, outperforming state-of-the-art techniques in terms of imperceptibility and robustness.*

Keywords: data hiding; information hiding; information security; national data security; network infrastructure; steganography; cyber security

1. **Introduction.** The rapid development of cloud computing has transformed the internet into a significant platform for sharing multimedia content, including images [1], audio [2], and video [3]. However, this increased exchange of digital data also raises security concerns [4]. Sensitive information is often exposed to potential threats from malicious actors seeking to intercept, modify, or destroy it [5, 6]. To address these challenges, steganography, the practice of hiding secret data within digital media, has gained considerable attention from researchers [7, 8]. Unlike cryptography, which protects the content of a message, steganography conceals the existence of the message itself [9]. It can be applied to various media types and has historical roots dating back to ancient times when hidden messages were embedded in physical objects [10].

Among the various forms of steganography, image-based methods have gained prominence due to the widespread use of digital images and their large data storage capacity [11, 12, 13, 14, 15, 16]. Despite extensive research and application, key challenges remain,

particularly in achieving a balance between data embedding capacity and image quality [17, 18]. Increasing the payload may cause visible distortions in the stego image, making the hidden data more easily detectable [13]. Conversely, reducing the payload to preserve visual quality limits the amount of information that can be securely embedded [19]. To overcome these limitations, recent studies have focused on developing adaptive steganographic techniques that enhance robustness by tailoring the method to the characteristics of the cover image and communication channel [18]. Nevertheless, there is still a need for new approaches that can optimize data hiding while minimizing the tradeoff between payload size and image quality, as highlighted in state-of-the-art reviews [15, 16, 17].

Despite the growing number of advanced image steganography methods, many state-of-the-art algorithms continue to face fundamental challenges in optimizing the tradeoff between payload capacity and stego image quality. Techniques such as Difference Expansion (DE) [16], histogram shifting [20], most significant bit prediction (MSB) [21], and predictive embedding [5] have been widely explored to increase data hiding efficiency. Some approaches integrate auxiliary techniques, such as segmentation [22], transform domain embedding [23], and entropy coding [14], to enhance concealment performance and reduce perceptual distortion. While these methods have succeeded in improving payload size and maintaining acceptable visual quality under controlled conditions, they often struggle with scalability, robustness, and adaptability across diverse image types and embedding scenarios. A common limitation is that increasing the payload typically leads to visible artifacts or statistical deviations, making the stego image more susceptible to detection by steganalysis tools [24]. Conversely, methods that prioritize imperceptibility tend to impose strict limits on payload capacity, reducing the practical usability of the system. Furthermore, the increased complexity of multi-stage embedding pipelines can hinder implementation in real-world applications [5].

To address the limitations identified in existing image steganography methods, this study proposes a novel algorithm, **RDJAT-Embed**, which is built upon the DE technique. Unlike traditional DE methods that rely solely on pairwise pixel differences, the proposed approach embeds secret data by exploiting the difference between each pixel value and the average of its local pixel group. This group-based difference strategy increases the embedding space while preserving local image structure, thereby reducing the risk of visual distortion. As a result, RDJAT-Embed achieves a high payload capacity without compromising imperceptibility. By maintaining a delicate balance between embedding capacity, stego image quality, and resistance to detection, the proposed method offers a more effective and practical solution for secure data hiding digital images.

2. **Related works.** Numerous image steganography techniques have been developed to achieve an optimal balance between embedding capacity, imperceptibility, and robustness. This section reviews recent state-of-the-art approaches that aim to enhance data-hiding performance while minimizing perceptual and statistical distortions in stego images.

In the steganography of digital images, a commonly adopted strategy is the hybridization of the least significant bit (LSB) substitution with DE. The method in [25] divides an image into 3-pixel blocks, where the central pixel undergoes 3-bit LSB substitution, and the remaining two pixels embed data using an enhanced modulus-based PVD approach. Such integration improves capacity while maintaining reasonable image quality. De La Croix et al. [26] extended this idea by applying a modulus function in conjunction with PVD and LSB techniques, resulting in an irreversible steganographic scheme with enhanced embedding efficiency. Fibonacci-based methods have also been explored to increase data embedding flexibility [27]. This method proposed an algorithm to conceal two bits of secret data within three bits of a cover pixel, increasing capacity at a modest cost

to visual quality. DE-based methods remain a dominant area of exploration due to their capacity for reversible embedding. The classical DE approach, which conceals data in the expanded difference between pixel pairs, can lead to pixel overflow or degrade image fidelity.

Moreover, another notable technique is the multidirectional pixel value differencing with modulus function (MDPVDMF) method [25, 28, 29], which integrates PVD with a modulus function to embed data within non-overlapping 2×2 blocks of grayscale images. The method computes directional differences within each block to determine the embedding capacity and applies the modulus function to adjust pixel values accordingly. To ensure all stego pixel values remain within the valid intensity range [0, 255], post-embedding readjustments are performed, effectively mitigating common issues such as the falling-off boundary problem (FOBP) and the incorrect extraction problem (IEP). Despite its effectiveness, MDPVDMF has several limitations [29]. Its dependence on directional embedding and fixed block structures imposes constraints on flexibility, particularly in images with complex or non-uniform textures. Moreover, the need for post-processing increases computational complexity and may compromise imperceptibility if not carefully managed.

Furthermore, to enhance embedding flexibility and capacity, several recent steganographic algorithms have adopted generalized quantization range widths combined with multiple-based number conversion techniques [28, 30, 32, 33, 34]. This paradigm eliminates the power-of-two limitations commonly associated with traditional PVD methods, allowing for more adaptable data partitioning [30, 33]. By converting secret message bits into variable-length digit streams based on custom quantization intervals, these methods significantly improve embedding efficiency while maintaining a balance between imperceptibility and robustness.

Building on the limitations observed in prior works, this study proposes a novel image steganography algorithm, RDJAT-Embed, based on the DE principle. Unlike conventional DE techniques that rely on the difference between pixel pairs or superpixels, the RDJAT-Embed embeds secret data by computing the difference between each pixel and the average value of its local pixel group. This approach eliminates the need for complex mathematical transformations or fixed structural patterns, thereby increasing adaptability and reducing computational overhead. Furthermore, the method utilizes all pixel values within the safe intensity range of 5 to 249 as valid embedding candidates, ensuring that the resulting stego image remains free from overflow or underflow artifacts. The design also supports high imperceptibility, as the embedding process can produce stego pixels that are identical to the original pixels, effectively minimizing visual distortion. In contrast to block-based methods like MDPVDMF and superpixel-based, the RDJAT-Embed offers a more flexible and computationally efficient embedding framework.

3. **Motivation and contribution.**

3.1. **Motivation.**

The growing reliance on digital communication in multimedia environments has intensified the demand for secure and efficient data hiding techniques. Steganography, which conceals secret information within digital media, has emerged as a promising solution for secure communication. However, traditional steganographic methods often struggle to maintain a balance between embedding capacity, imperceptibility, and security. Techniques such as LSB substitution and PVD typically offer limited payload capacity to avoid visual distortion. At the same time, they are prone to statistical detectability and require computationally expensive operations, undermining their suitability for real-world deployment.

Moreover, several existing approaches suffer from practical limitations. High embedding rates may introduce noticeable artifacts, thereby increasing vulnerability to steganalysis techniques such as RS and SPAM. Some methods, such as the FOBP, face robustness issues or require multiple stego images for message extraction, making them less reliable and more challenging to implement in constrained environments. These challenges highlight the need for a steganographic technique that offers higher capacity, improved visual quality, and robustness, all while minimizing computational overhead.

3.2. **Contributions.** To address these challenges, this study presents RDJAT-Embed, a novel steganographic method based on a modified DE technique. This approach offers a lightweight, high-performance solution for secure data embedding, particularly suited to resource-constrained environments. The main contributions of this work are:

1. **Enhanced Embedding Capacity:** RDJAT-Embed replaces traditional pixel-pair differences with the difference between each pixel and its group average, enabling more flexible and efficient data embedding. A five-value grouping strategy optimizes payload distribution and reduces visual distortion.
2. **Improved Imperceptibility and Simplicity:** By restricting the embedding to pixels within the [5, 249] range, the method avoids overflow and underflow while maintaining high image quality. In many cases, the original pixel values are preserved, and the use of a tracing array (TRA) with precomputed group averages simplifies data extraction, eliminating the need for side-channel information or multiple images.
3. **Robustness and Adaptability:** RDJAT-Embed avoids embedding artifacts by adaptively using local pixel statistics, enhancing resilience to steganalysis. It requires no post-processing or pixel adjustment, reducing the risk of detection. The flexible grouping framework also allows adaptation to varying capacity and fidelity requirements.

4. **Methodology.** RDJAT-Embed enhances the performance of existing methods. Unlike the ordinary DE methods, this method uses the difference between pixel value and its group average value to conceal secret data. The technique groups pixels based on their value and then uses the average of each group to subtract the pixel value. The pixels of an image are grouped based on their value, with the size of the group corresponding to five values in an interval of five from zero to 254 (0–4, 5–9, until 249–254). RDJAT-Embed embeds the data when the differences (d) range between negative and positive four ($-4 \leq d \leq 4$). All pixels are used to embed data except pixels $p(i)$ less than five ($p(i) < 5$) and greater than 249 ($p(i) > 249$)). To avoid the problems of under/overflow, RDJAT-Embed only considers, if the difference condition is satisfied, the pixel values range from five to 249 ($5 \leq p(i) \leq 249$). It is essential to note that the underflow problem occurs when pixel values become negative, and the overflow issue occurs when pixel values exceed 255. Moreover, the modulus function is implemented in the proposed extraction algorithm to get secret data. To demonstrate the proposed scheme, the necessary steps for embedding and extracting secret data are presented below. Moreover, Figures **??** and **??** are provided to visualize how the proposed scheme works. To shed light on the deep steps taken in the embedding process, Algorithm 1 presents the pseudocodes of the process, and the extraction process is detailed in Algorithm 2.

4.1. **Embedding the secret data in a cover image.** The embedding algorithm constitutes a fundamental component of the proposed steganographic method, as it determines the systematic procedure through which the secret data is concealed within the cover image. To initiate the embedding process, the cover image is first transformed into a

multidimensional array p, enabling structured access to its pixel values. Concurrently, the secret message is extracted, converted into a binary or numerical format, and stored in an array b to facilitate efficient embedding. The following steps detail the algorithmic procedure for embedding the secret data into the cover image, ensuring imperceptibility and data integrity in the resulting stego image. The following are the detailed steps of the embedding process. (1) Get the bits of the secret message and store it in an array (b)

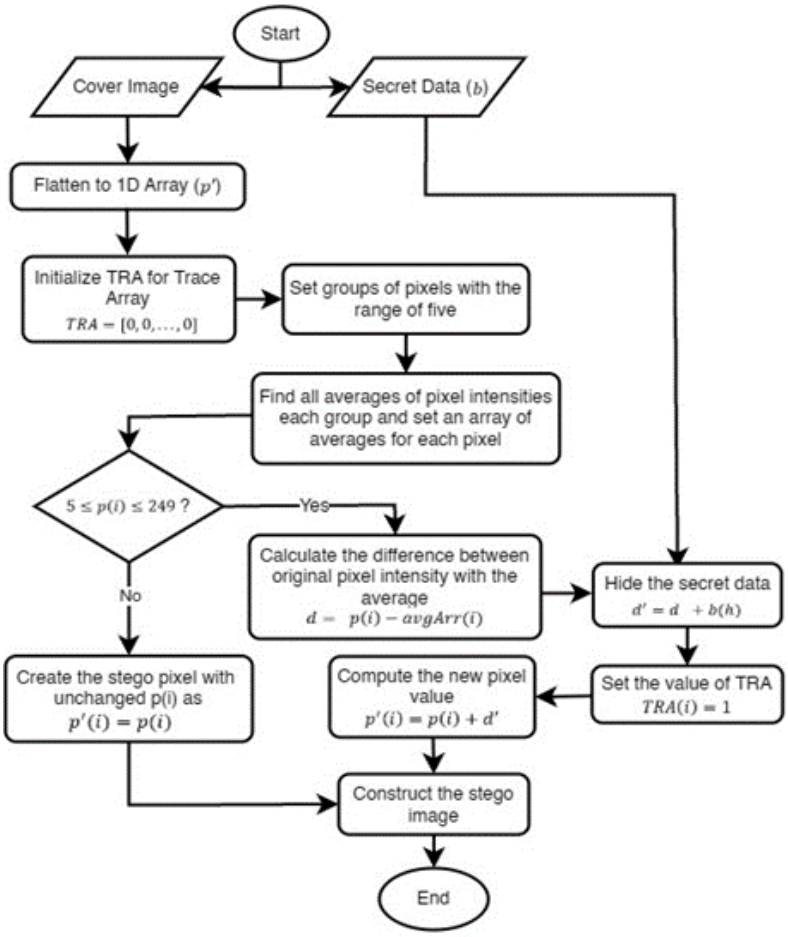

FIGURE 1. Embedding process flowchart

to facilitate its further fetching and use.

(2) Iterate through the image array and sum the count of occurrences of each pixel value within five intervals. All the sums are saved in a one-dimensional array $count(idx)$.

(3) Count all the occurrences of each pixel value and store it in an array $countEach$ with the size of one by 255 (maximum value of pixel) and iterate through the array to multiply it with the pixel value (index). The next step consists of summing up using Equation (1), where i denotes the pixel value. The pixel index obtained from Equation (2) is then subtracted by one to ensure the value of the pixel ranges from 0 to 255, and it is used for the count and sum that define the index for the group of pixels. Variable $countEach(idx)$ represents the number of occurrences for the pixel value and $sum(idx)$ is the sum of the results of each group. Iterate through the array p to calculate the average for each group using Equation (3), considering $sum(idx)$, the group sum of the counts multiplied with each value at the group index idx, and the average value is rounded up using ceil function.

$$sum(idx) = sum(idx) + (i - 1) \times countEach(i) \tag{1}$$

$$idx = \lfloor p(i)/5 \rfloor + 1 \tag{2}$$

$$avgArr(i) = \lceil sum(idx)/count(idx) \rceil \tag{3}$$

Algorithm 1 Embedding steps

Notations:
Notation 1: Cover image $\rightarrow CI$
Notation 2: Cover image array of pixels $\rightarrow p$
Notation 3: Average array for each cover image pixel $\rightarrow avgArr$
Notation 4: Difference between cover pixel with average $\rightarrow d$
Notation 5: Secret data array $\rightarrow b$
Notation 6: Stego image array $\rightarrow p'$
Notation 7: Stego image $\rightarrow STI$
Notation 8: Tracing array $\rightarrow TRA$
Inputs: Cover image, Secret data
Outputs: Stego image, Average array, Tracing array

1: Start
2: Load the cover image CI
3: Load the secret data b
4: Reshape CI into one-dimensional array p
5: Initiate tracing array with length of p and set all to zero
6: Calculate count group of occurrences for each pixel value
7: Calculate sum group count multiplied by each pixel value
8: **for** $i = 0$ to $length(p)$ **do**
9: **if** $5 \leq p(i) \leq 249$ **then**
10: Calculate average
$$avgArr(i) = \lceil \frac{sum}{count} \rceil$$
11: Create stego pixel
$$p'(i) = p(i) + (p(i) - avgArr(i)) + b(h)$$
12: Set $TRA(i) = 1$
13: **else**
14: Create stego pixel with the value of original pixel
$$p'(i) = p(i)$$
15: **end if**
16: **end for**
17: Construct STI by reshaping p' into original-dimensional image
18: End the script

(4) Using a tracing array (TRA) Using a tracing array (TRA), we assign a value in the TRA variable as a key location for the embedded secret data. If the pixel is embedded with secret data, then bit one is used to identify the pixel containing secret data. Otherwise, the bit is zero. Computing the difference between the pixels and the average values using the relation in Equation (4), with d as the difference, $p(i)$ is the cover image pixel value at index i, and $avgArr(i)$ is the group average for the pixel at index i. The data concealment follows the relation in Equation (5) with $b(h)$, the secret data at index h, and d' is a new difference.

$$d = p(i) - avgArr(i) \tag{4}$$

$$d' = d + b(h) \tag{5}$$

(5) The pixels of the stego image are computed using Equation (6). For the pixel value falling within the range of 5 to 249, the data are concealed. Then, the pixel value becomes a sum of the cover pixel and the difference d' at the same index $p'(i)$. Otherwise, the stego pixel is identical to the original pixel at the same index.

$$p'(i) = \begin{cases} p(i) + d', & 5 \leq p(i) \leq 249 \\ p(i), & p(i) < 5 \text{ and } p(i) > 249 \end{cases} \tag{6}$$

(6) Give the tracing array, $TRA(i)$, value using Equation (7). Variable $TRA(i)$ represents the tracing array at the index of i. If the pixel value $p(i)$ falls within the range and is used to conceal secret data, then the TRA value at the same index as the pixel (i) is set to one. Otherwise, the TRA is unchanged (remains zero). Construct the stego image by reshaping the stego pixels array to its original dimensions.

$$TRA(i) = \begin{cases} 1, & 5 \leq p(i) \leq 249 \\ 0, & p(i) < 5 \text{ and } p(i) > 249 \end{cases} \tag{7}$$

4.2. **Practical example of embedding the secret data in a cover image.** To shed light on the proposed embedding process, the following scenarios demonstrate the steps taken from the secret bits and cover image to the final stego image.

(1) Scenario 1: For $p(i) = 10$, $avgArr(i) = 9$, and $b = 1$

$$d = 10 - 9 \tag{8}$$
$$d' = 1 + 1 \tag{9}$$
$$TRA(i) = 1 \tag{10}$$
$$p'(i) = 10 + 2 \tag{11}$$
$$p'(i) = 13 \tag{12}$$

(2) Scenario 2: $p(i) = 0$, $avgArr(i) = 4$, and $b = 1$

$$d = 0 - 4 \tag{13}$$
$$d = -4 \tag{14}$$
$$d' = -4 + 1 \tag{15}$$
$$d' = -3 \tag{16}$$
$$p'(i) = 0 + (-3) \tag{17}$$
$$p'(i) = -3 \tag{18}$$
$$TRA(i) = 0 \tag{19}$$

(3) Scenario 3: $p(i) = 254$, $avgArr(i) = 250$, and $b = 1$

$$d = 254 - 250 \tag{20}$$
$$d = 6 \tag{21}$$
$$d' = 6 + 1 \tag{22}$$
$$d' = 7 \tag{23}$$
$$p'(i) = 254 + 7 \tag{24}$$
$$p'(i) = 261 \tag{25}$$

4.3. **Hidden secret data and original cover image restoration.** The extraction process of the hidden secret data and the cover image used in concealment from the stego image is performed using the tracing array and the average array defined throughout the embedding process. Following Figure 2 and Algorithm 2, the steps involved in the extraction process are given below. Reshaping the stego image to a one-dimensional array

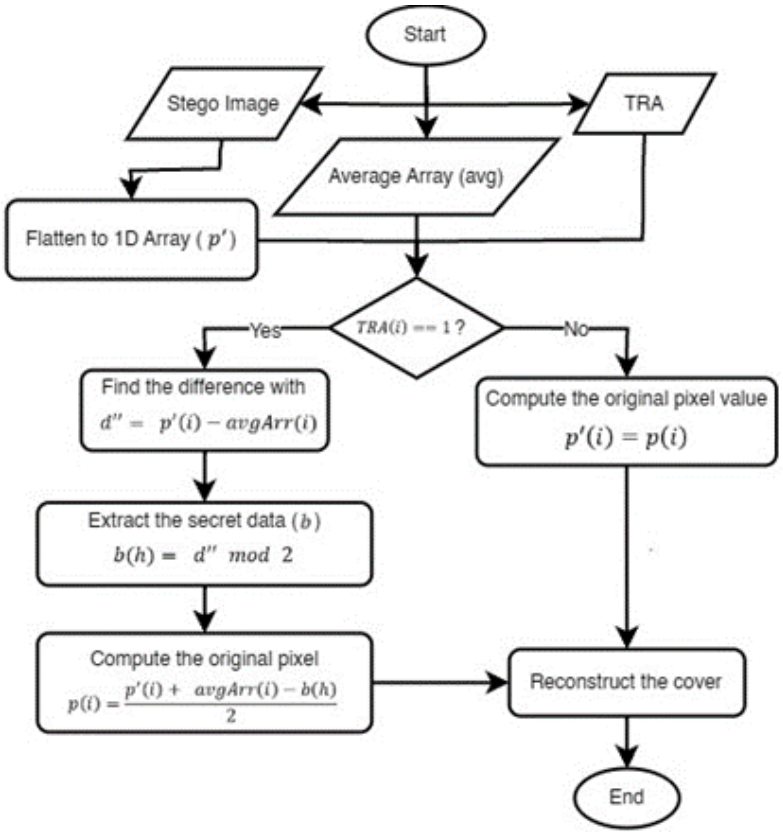

FIGURE 2. Extraction process flowchart

(stego array) and through the stego array, using the TRA, iterate, and if the $TRA(i)$ value is one, compute the difference between the stego pixel and the average array at the same index using Equation (26) considering $p'(i)$, the stego pixel at the index of i, $avgArr(i)$, the average group for the pixel at the index of i, and d'', the difference between them.

$$d'' = p'(i) - avgArr(i) \tag{26}$$

(1) To extract the secret bits, apply the modulus function of the d'' with two using Equation (27). Note that $s(h)$ is the extracted secret bits at the index h. The d'' is equivalent to $2d$ added with the secret bit, proofed using Equations (28)–(31). Therefore, the d'' value depends on the secret bit that has been embedded.

$$b(h) = d'' \bmod 2 \tag{27}$$
$$p'(i) = p(i) + (p(i) - avgArr(i)) + b(h) \tag{28}$$
$$d'' + avg(i) = p(i) + p(i) - avgArr(i) + b(h) \tag{29}$$
$$d'' = 2(p(i) - avgArr(i)) + b(h) \tag{30}$$
$$d'' = 2d + b(h) \tag{31}$$

(2) To extract the original cover image used for the embedding process, Equation (32) is used in combination with the TRA. If $TRA(i) = 1$, the original pixel is extracted by applying Equation (33). Otherwise, the original pixel is the same as the stego pixel. Note that $p(i)$ denotes the original pixel value at the index i, $avgArr(i)$ denotes the average value at the index i, $b(h)$ is secret data at the index h, and the tracing array value at the index i denoted with $TRA(i)$. The final step is to reconstruct the image by reshaping the pixel array to its original dimensions.

$$2p(i) = p'(i) + avgArr(i) - b(h) \tag{32}$$

$$p(i) = \begin{cases} \dfrac{p'(i) + avgArr(i) - b(h)}{2}, & TRA(i) = 1 \\ p'(i), & TRA(i) = 0 \end{cases} \tag{33}$$

4.4. **Practical example of the data and cover extraction process.** In this subsection, we provide detailed scenarios that may occur during the data and cover the extraction process. The d'' variable denotes the difference between the stego pixel value and its average, while the $b(h)$ represents the secret bit at index h.

(1) Scenario 1: $p'(i) = 20$, $avgArr(i) = 21$, and the $TRA(i) = 1$.

For this scenario, since the pixel value falls within the range that may have been used to embed the secret data and the $TRA(i)$ value indicates that the data has been embedded, we can use Equations (34)–(36). The final secret data is obtained with Equation (36). The recovery of the cover image's pixels consists of gathering all pixels obtained using Equations (37)–(39).

$$d'' = 20 - 21 \tag{34}$$
$$b(h) = -1 \bmod 2 \tag{35}$$
$$b(h) = 1 \tag{36}$$
$$p(i) = \frac{20 + 21 - 1}{2} \tag{37}$$
$$p(i) = \frac{40}{2} \tag{38}$$
$$p(i) = 20 \tag{39}$$

(2) Scenario 2: $p'(i) = 2$ or 255, $avgArr(i) = 1$ or 254, and $TRA(i) = 0$.

For this scenario, where $TRA(i) = 0$, based on the embedding conditions, the original cover image's pixel value is obtained by taking the actual pixel in the stego image, as the embedding process has not altered it previously.

5. Experimental results and analyses.

Algorithm 2 Extracting steps

Notations:

Notation 1: Stego image $\rightarrow STI$

Notation 2: Difference between stego pixel with average $\rightarrow d''$

Notation 3: Average group array $\rightarrow avgArr$

Notation 4: Recovered secret bit $\rightarrow s$

Notation 5: Tracing array $\rightarrow TRA$

Input 1: Stego Image

Input 2: Average array

Input 3: TRA

Output 1: Original cover image

Output 2: Secret data

1: Start
2: Load the stego image
3: Load the average array
4: Load the tracing array
5: Reshape STI to one dimensional array p'
6: Iterate stego image and average array while $i = 0;\ i \leq length(p')$
7: **if** $TRA(i)$ value is one **then**
8: Compute the difference (d'') by

$$d'' = p'(i) - avgArr(i)$$

9: Extract the secret bits at index h by

$$s(h) = d'' \bmod 2$$

10: Recover the cover image pixel by

$$p(i) = \frac{p'(i) + avgArr(i) - s(h)}{2}$$

11: **else**
12: Set original pixel value to stego pixel value:

$$p(i) = p'(i)$$

13: **end if**
14: Build the original cover image
15: End the script

5.1. **Experimental environment and dataset.** The experimentation of the proposed RDJAT-Embed was conducted on an Intel Core i5 processor operating at 2.3 GHz and 16 GB of RAM. The software used as the implementation platform is MATLAB R2024b due to its robustness in running the implementation program. The RDJAT-Embed is implemented as a steganography method using cover images from the SIPI image dataset [35] in Figure 3. These images have an exact resolution of 512 × 512. It is important to note that all experiments were conducted using grayscale images. This choice was motivated by the simplicity and efficiency of grayscale data, which resides in a single plane, making it computationally lighter and easier to analyze. Additionally, grayscale images provide a suitable baseline to evaluate the core performance of the proposed method with optimal memory and time usage. While the method is generalizable to RGB images, such an extension would require handling multiple channels and potential inter-channel dependencies, which increases complexity and computational load. The secret bits used

for embedding experiments are extracted from the Lorem Ipsum story [36]. The use of Lorem Ipsum is motivated by the need for a uniform distribution of bits, like that of existing benchmark steganographic algorithms.

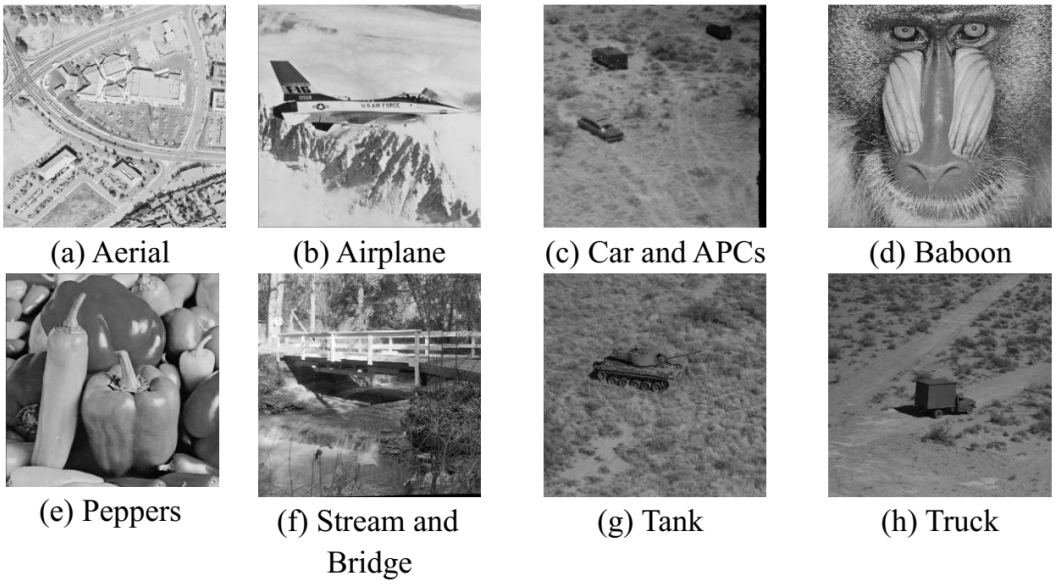

FIGURE 3. Sample test cover images

5.2. **Evaluation metrics.** To comprehensively assess the performance of the RDJAT-Embed, four standard evaluation metrics are employed: Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and Structural Similarity Index Measure (SSIM). These metrics collectively evaluate both the visual quality and embedding efficiency of the stego-images. The PSNR is a widely used metric for quantifying the distortion introduced during data compression or data embedding. As defined in Equation (40), PSNR measures the ratio between the maximum possible signal power and the power of the noise introduced by the embedding process, expressed in decibels (dB). Higher PSNR values correspond to better visual quality and minimal distortion. MSE, defined in Equation (41), calculates the average of the squared differences between the pixel intensities of the original cover image and the corresponding stego image. Lower MSE values indicate reduced distortion due to embedding. SSIM, computed using Equation (42), evaluates the perceptual quality of the stego image by measuring structural similarity in terms of luminance, contrast, and structural components. SSIM values range from 0 to 1, where values closer to 1 indicate higher structural fidelity between the cover and stego images. It is crucial to indicate that the cover image is denoted as p, the stego image as p', and the mean pixel intensities as μ_p and $\mu_{p'}$, with the variance intensities represented by σ_p and $\sigma_{p'}$, and the covariance by $\sigma_{pp'}$.

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right) \tag{40}$$

$$MSE = \frac{1}{h \times w}\sum_{i=1}^{h}\sum_{j=1}^{w}\left(p_{(i,j)} - p'_{(i,j)}\right)^2 \tag{41}$$

$$SSIM(p,p') = \frac{(2\mu_p\mu_{p'} + C_1)(2\sigma_{pp'} + C_1)}{(\mu_p^2 + \mu_{p'}^2 + C_1)(\sigma_p^2 + \sigma_{p'}^2 + C_1)} \tag{42}$$

5.3. **Results.**

5.3.1. *Obtained results.* The results presented in Table 1 provide a detailed assessment of the PSNR in dB achieved by the proposed RDJAT-Embed method across ten standard grayscale images and eleven payload levels ranging from 1 kb to 100 kb. The obtained results, as reported in the table, demonstrate the scientific robustness and practical effectiveness of the RDJAT-Embed in preserving stego image quality during data embedding. At low payloads, all images achieve PSNR values above 68 dB, which is relatively high, indicating negligible distortion and near-lossless embedding. As the payload increases, the PSNR values gradually decrease, as expected, due to the increased amount of embedded data. However, the RDJAT-Embed consistently maintains PSNR values above 48 dB, even at the highest tested payload (100 kb), far surpassing the standard acceptability threshold of 30 dB. Notably, for Baboon and Pepper images, which are traditionally challenging for data hiding due to their textural variability, the PSNR values remain above 48 dB at 100 kb, indicating that RDJAT-Embed is resilient to image complexity and provides adaptive embedding that minimizes distortion. For the Aerial image, the highest PSNR value, ranging between 68.851 and 48.460 dB, is yielded, showing a promising record for the quality of the stego image.

Table 1. Obtained PSNR in dB for the proposed RDJAT-Embed

Image	1kb	10kb	20kb	30kb	40kb	50kb	60kb	70kb	80kb	90kb	100kb
Aerial	68.851	58.468	55.521	53.703	52.481	51.484	50.695	50.018	49.438	48.917	48.460
Airplane	68.173	58.768	55.575	53.830	52.640	51.713	50.950	50.260	49.655	49.117	48.636
Baboon	68.311	58.416	55.409	53.602	52.369	51.388	50.596	49.928	49.362	48.850	48.399
Car and APCs	69.184	58.909	55.879	54.078	52.850	51.842	51.066	50.352	49.778	49.245	48.773
Fishing Boat	68.779	58.746	55.686	53.866	52.636	51.642	50.845	50.162	49.599	49.085	48.636
Pepper	68.012	58.342	55.351	53.597	52.380	51.418	50.619	49.929	49.350	48.861	48.433
Stream and bridge	69.443	59.406	56.386	54.639	53.316	52.338	51.522	50.904	50.346	49.815	49.356
Tank	68.773	58.711	55.675	53.956	52.741	51.777	50.978	50.304	49.742	49.244	48.775
Truck	68.900	59.087	56.115	54.367	53.126	52.150	51.331	50.687	50.082	49.561	49.103

Table 2. MSE results from all the test images and payload sizes for the RDJAT-Embed

Image	1kb	10kb	20kb	30kb	40kb	50kb	60kb	70kb	80kb	90kb	100kb
Aerial	0.008	0.093	0.182	0.277	0.367	0.462	0.554	0.648	0.740	0.835	0.927
Airplane	0.010	0.086	0.180	0.269	0.354	0.438	0.523	0.612	0.704	0.797	0.890
Baboon	0.010	0.094	0.187	0.284	0.377	0.472	0.567	0.661	0.753	0.847	0.940
Car and APCs	0.008	0.084	0.168	0.254	0.337	0.426	0.509	0.600	0.684	0.774	0.863
Fishing Boat	0.009	0.087	0.176	0.267	0.354	0.446	0.535	0.626	0.713	0.803	0.890
Pepper	0.010	0.095	0.190	0.284	0.376	0.469	0.564	0.661	0.755	0.845	0.933
Stream and bridge	0.007	0.075	0.149	0.223	0.303	0.380	0.458	0.528	0.600	0.679	0.754
Tank	0.009	0.087	0.176	0.262	0.346	0.432	0.519	0.606	0.690	0.774	0.862
Truck	0.008	0.080	0.159	0.238	0.317	0.396	0.479	0.555	0.638	0.719	0.799

The MSE results presented in Table 2 provide detailed insight into the distortion performance of the proposed RDJAT-Embed method across various payload sizes and image types. The MSE values remain consistently low, even as the payload increases from 1 kb to 100 kb, indicating minimal pixel-level distortion. At the highest payload, the MSE ranges from 0.754 for the Stream and bridge image to 0.940 for Baboon, which demonstrates the method's robustness even in the presence of complex textures and structural variations. Similarly, images such as Tank and Fishing Boat exhibit MSE values of 0.862 and 0.890, respectively, further emphasizing the RDJAT-Embed's capacity to maintain

visual quality in diverse scenarios. At the lowest payload of 1 kb, MSE values are negligible, not exceeding 0.010 across all images, reflecting the excellent reversibility and imperceptibility of the proposed algorithm.

The average PSNR results obtained using the proposed RDJAT-Embed method, as illustrated in Figure 4, reveal a consistently high level of visual fidelity across ten standard test images. The PSNR values range from 53.299 dB for Pepper to 54.316 dB for Stream and Bridge, demonstrating that all cover images maintain superior quality even after embedding. For Baboon and Car and APCs, the achieved values are 53.330 dB and 53.814 dB, respectively, confirming the method's robustness in handling high-frequency textures and fine-grained image details. These results indicate that the RDJAT-Embed algorithm effectively preserves structural information during the embedding process, achieving an overall average PSNR of approximately 53.76 dB, well above the commonly accepted threshold for imperceptibility in data hiding applications. Moreover, the scientific significance of these results lies in the RDJAT-Embed's ability to maintain high PSNR values with minimal variation across diverse image content.

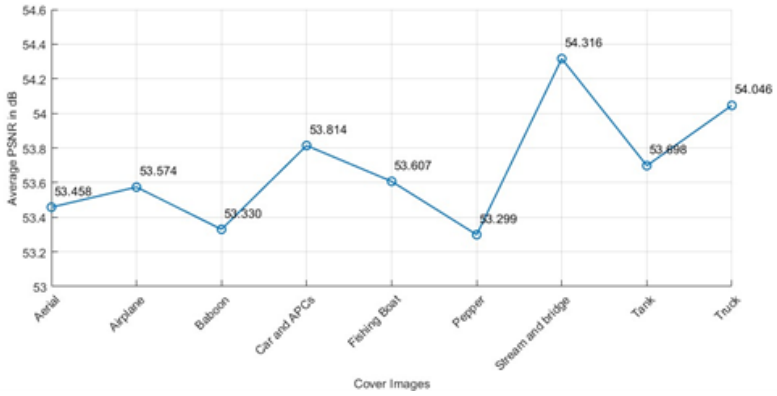

FIGURE 4. Average PSNR in dB for all tested cover images

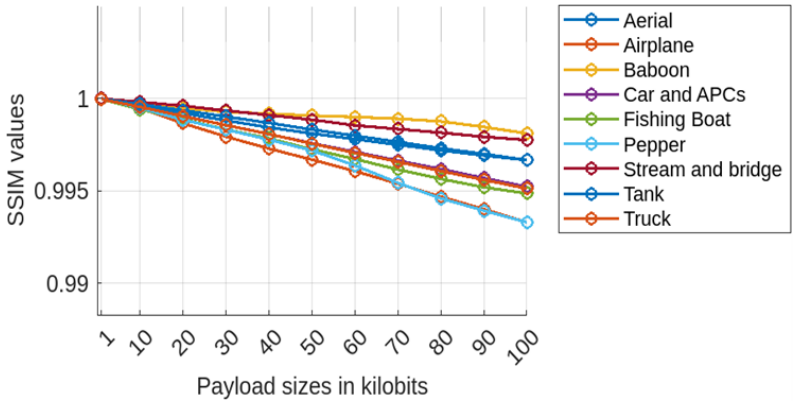

FIGURE 5. Obtained SSIM results for the RDJAT-Embed

The minimal distortion observed in both smooth and textured regions reflects the adaptability and precision of the RDJAT-Embed. To highlight the contribution of the proposed RDJAT-Embed, Figure 5 illustrates the SSIM performance of the proposed RDJAT-Embed method where all tested images consistently maintain SSIM values above 0.99, indicating relatively excellent structural similarity between the original cover and the stego images. This confirms that RDJAT-Embed effectively preserves luminance, contrast, and structural details even under increasing embedding loads. The RDJAT-Embed

performs particularly well on Car and APCs, as well as Fishing Boat and Truck, which sustain the highest SSIM scores across various payloads. Even complex and high-texture images, such as Baboon and Pepper, retain SSIM values close to 0.995 at the maximum payload, demonstrating that the method adapts well to diverse visual content without compromising perceptual quality. The minimal and gradual decline in SSIM further highlights RDJAT-Embed's capability to balance embedding capacity and imperceptibility.

Figure 6 also presents the average SSIM values for all tested cover images under the RDJAT-Embed method, reinforcing the method's ability to maintain exceptional structural fidelity during data embedding. All images consistently achieve SSIM scores of 0.997 or higher, with Baboon, Stream and Bridge, and Tank attaining the highest average SSIM of 0.999 despite their complex textures and structural variations.

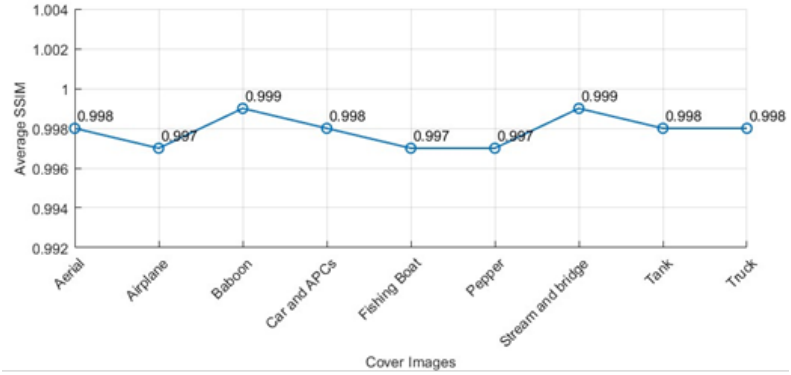

FIGURE 6. Obtained average SSIM for all cover images under RDJAT-Embed

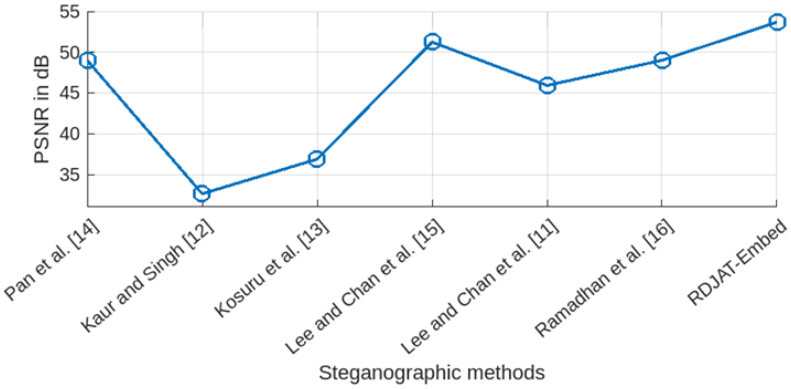

FIGURE 7. Comparison of the PSNR to the state-of-the-art algorithms

5.4. **Results comparison.** To provide a comprehensive comparison of the proposed RDJAT-Embed method against a selection of state-of-the-art steganographic techniques, Figure 7 illustrates the average PSNR values obtained against those reported in existing methods. The RDJAT-Embed achieves the highest average PSNR of 53.683 dB, indicating exceptional preservation of visual quality after data embedding. This value surpasses the threshold of 50 dB, which is widely considered to represent imperceptible visual distortion to the human eye, thus demonstrating the model's ability to embed data with minimal perceptual degradation. In comparison, the methods by Pan et al. [14], and by Ramadhan et al. [16] record an average PSNR about 49 dB, both reflecting moderate fidelity but still

lower than RDJAT-Embed. Lee and Chan [11] and Lee and Chan [15] report average PSNR values around 45.920 dB and 51.240 dB, respectively, with the latter approaching high fidelity but remaining about 2.440 dB below the proposed method. Meanwhile, Kosuru et al. [13] and Kaur and Singh [12] show substantially lower average PSNR values about 36.880 dB and 32.630 dB, respectively, signifying notable distortion and reduced imperceptibility. The outperformance of RDJAT-Embed in terms of average PSNR is a direct result of its robust embedding mechanism, which utilizes residual difference joint adaptive thresholding to balance capacity and quality. This enables RDJAT-Embed to maintain image structural integrity and noise resistance even at higher payloads. In addition to outperforming existing techniques by a margin of up to 21.050 dB, RDJAT-Embed establishes a new benchmark for fidelity-aware steganography, making it highly applicable in sensitive domains.

6. **Conclusion.** This study introduces RDJAT-Embed, a lightweight and effective steganography method that enhances data embedding capacity while preserving image quality. By employing a modified DE technique based on group averages rather than pixel pairs, RDJAT-Embed achieves high imperceptibility, with an average PSNR of 53.683 dB, a low MSE, and a near-perfect SSIM. Its five-value pixel grouping strategy prevents overflow and underflow, maintaining visual fidelity across stego images. Unlike many existing approaches, RDJAT-Embed requires only a single stego image and a simple tracing array for data extraction, offering improved computational efficiency and reduced complexity. The method also demonstrates robustness against common steganalysis attacks, making it suitable for secure, real-world applications. Comparative evaluations confirm that RDJAT-Embed outperforms several state-of-the-art techniques in balancing payload capacity, imperceptibility, and efficiency.

Despite its strengths, RDJAT-Embed excludes extreme pixel values, which may limit its performance in grayscale or binary images with dominant dark or bright regions. Future research can address this limitation by exploring adaptive group sizing to improve pixel coverage and embedding flexibility.

Acknowledgement. This research was supported by the Institut Teknologi Sepuluh Nopember (ITS), Indonesia. The authors thank the members of the Cyber Security Research Group, Net-Centric Computing Laboratory, Department of Informatics, ITS for the support and discussion.

REFERENCES

[1] M. H. Noor Azam, F. Ridzuan, M. N. S. Mohd Sayuti, A. H. Azni, N. H. Zakaria, and V. Potdar, "A systematic review on cover selection methods for steganography: Trend analysis, novel classification and analysis of the elements," *Comput. Sci. Rev.*, vol. 56, p. 100726, May 2025, doi: 10.1016/j.cosrev.2025.100726.

[2] A. A. AlSabhany, A. H. Ali, F. Ridzuan, A. H. Azni, and M. R. Mokhtar, "Digital audio steganography: Systematic review, classification, and analysis of the current state of the art," *Comput. Sci. Rev.*, vol. 38, p. 100316, Nov. 2020, doi: 10.1016/j.cosrev.2020.100316.

[3] J. Li, M. Zhang, K. Niu, Y. Zhang, Y. Ke, and X. Yang, "High-Security HEVC Video Steganography Method Using the Motion Vector Prediction Index and Motion Vector Difference," *Tsinghua Sci. Technol.*, vol. 30, no. 2, pp. 813–829, Apr. 2025, doi: 10.26599/TST.2024.9010016.

[4] S. Rahman, J. Uddin, M. Zakarya, H. Hussain, A. A. Khan, and A. Ahmed, "A Comprehensive Study of Digital Image Steganographic Techniques," *IEEE Access*, vol. 11, pp. 6770–6791, 2023, doi: 10.1109/ACCESS.2023.3237393.

[5] P. C. Mandal, I. Mukherjee, G. Paul, and B. N. Chatterji, "Digital image steganography: A literature survey," *Inf. Sci. (N. Y.)*, vol. 609, pp. 1451–1488, Sep. 2022, doi: 10.1016/j.ins.2022.07.120.

[6] Q. Zhang, Y. Zhang, Y. Ma, Y. Liu, and X. Luo, "Digital Image Steganographer Identification: A Comprehensive Survey," *Comput. Mater. Continua*, vol. 81, no. 1, pp. 105–131, 2024, doi: 10.32604/cmc.2024.055735.

[7] H. Zheng, C. Zhou, X. Li, Z. Guo, and T. Wang, "A Novel Steganography-Based Pattern for Print Matter Anti-Counterfeiting by Smartphone Cameras," *Sensors*, vol. 22, no. 9, p. 3394, Apr. 2022, doi: 10.3390/s22093394.

[8] T. Wang, H. Cheng, X. Liu, Y. Xu, F. Chen, and M. Wang, J. Chen, "Lossless image steganography: Regard steganography as super-resolution," *Inf. Process. Manag.*, vol. 61, no. 4, p. 103719, Jul. 2024, doi: 10.1016/j.ipm.2024.103719.

[9] D. T. Firdaus, N. J. De La Croix, and T. Ahmad, "AudioSecure: An open-source code to secure data using interpolation and multi-layering techniques within audio covers," *Software Impacts*, vol. 22, p. 100707, Nov. 2024, doi: 10.1016/j.simpa.2024.100707.

[10] F. Kabir, T. K. Araghi, and D. Megías, "Privacy-preserving protocol for high-frequency smart meters using reversible watermarking and Paillier encryption," *Comput. Electr. Eng.*, vol. 119, p. 109497, Oct. 2024, doi: 10.1016/j.compeleceng.2024.109497.

[11] C.-F. Lee and K.-C. Chan, "A Novel Dual Image Reversible Data Hiding Scheme Based on Vector Coordinate with Triangular Order Coding," *IEEE Access*, vol. 12, pp. 90794–90814, 2024, doi: 10.1109/ACCESS.2024.3421545.

[12] R. Kaur and B. Singh, "A robust and imperceptible n-Ary based image steganography in DCT domain for secure communication," *Multimed. Tools Appl.*, vol. 83, no. 7, pp. 20357–20386, Aug. 2023, doi: 10.1007/s11042-023-16330-9.

[13] S. N. V. J. D. Kosuru, A. Pradhan, K. A. Basith, R. Sonar, and G. Swain, "Digital Image Steganography With Error Correction on Extracted Data," *IEEE Access*, vol. 11, pp. 80945–80957, 2023, doi: 10.1109/ACCESS.2023.3300918.

[14] P. Pan, Z. Wu, C. Yang, and B. Zhao, "Double-Matrix Decomposition Image Steganography Scheme Based on Wavelet Transform with Multi-Region Coverage," *Entropy*, vol. 24, no. 2, p. 246, Feb. 2022, doi: 10.3390/e24020246.

[15] C.-F. Lee and K.-C. Chan, "Improved Dual-image Quality with Reversible Data Hiding Using Translocation and Switching Strategy," *Comput. Syst. Sci. Eng.*, vol. 44, no. 2, pp. 1551–1566, 2023, doi: 10.32604/csse.2023.026294.

[16] I. F. Ramadhan, Rr. D. A. Anandha, A. W. C. D'Layla, N. J. De La Croix, and T. Ahmad, "Image Steganography using Customized Differences between the Neighboring Pixels," in *Proc. 7th Int. Conf. Informatics Comput. Sci. (ICICoS)*, IEEE, Jul. 2024, pp. 496–501, doi: 10.1109/ICICoS62600.2024.10636936.

[17] Y. Bai, G. Xing, H. Wu, Z. Rao, C. Ma, S. Wang, X. Liu, Y. Zhou, J. Tang, K. Huang, and J. Kang, "Backdoor Attack and Defense on Deep Learning: A Survey," *IEEE Trans. Comput. Soc. Syst.*, vol. 12, no. 1, pp. 404–434, Feb. 2025, doi: 10.1109/TCSS.2024.3482723.

[18] M. Driss, L. Berriche, S. Ben Atitallah, and S. Rekik, "Steganography in IoT: A Comprehensive Survey on Approaches, Challenges, and Future Directions," *IEEE Access*, vol. 13, pp. 74844–74875, 2025, doi: 10.1109/ACCESS.2025.3564120.

[19] J. D. L. C. Ntivuguruzwa and T. Ahmad, "FuzConvSteganalysis: Steganalysis via fuzzy logic and convolutional neural network," *SoftwareX*, vol. 26, 2024, doi: 10.1016/j.softx.2024.101713.

[20] C. T. Huang, N. S. Shongwe, and C. Y. Weng, "Enhanced Embedding Capacity for Data Hiding Approach Based on Pixel Value Differencing and Pixel Shifting Technology," *Electronics*, vol. 12, no. 5, Mar. 2023, doi: 10.3390/electronics12051200.

[21] Z.-M. Yeh, C.-C. Lin, Y. Lin, C.-C. Chang, and C.-C. Chang, "High-Capacity Reversible Data Hiding in Encrypted Hyperspectral Images Using MSB Prediction and Arithmetic Coding," *J. Inf. Hiding Multimed. Signal Process.*, vol. 16, no. 2, p. 30, 2025.

[22] J. Zhang, K. Chen, C. Qin, W. Zhang, and N. Yu, "AAS: Automatic Virtual Data Augmentation for Deep Image Steganalysis," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 4, pp. 3515–3527, Jul. 2024, doi: 10.1109/TDSC.2023.3333913.

[23] Y. Yousfi and J. Fridrich, "An Intriguing Struggle of CNNs in JPEG Steganalysis and the OneHot Solution," *IEEE Signal Process. Lett.*, vol. 27, pp. 830–834, 2020, doi: 10.1109/LSP.2020.2993959.

[24] J. D. L. C. Ntivuguruzwa and T. Ahmad, "A convolutional neural network to detect possible hidden data in spatial domain images," *Cybersecurity*, vol. 6, no. 1, p. 23, Sep. 2023, doi: 10.1186/s42400-023-00156-x.

[25] H.-H. Liu, P.-C. Su, and M.-H. Hsu, "An Improved Steganography Method Based on Least-Significant-Bit Substitution and Pixel-Value Differencing," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 11, Nov. 2020, doi: 10.3837/tiis.2020.11.016.

[26] N. J. De La Croix, C. C. Islamy, and T. Ahmad, "Reversible Data Hiding using Pixel-Value-Ordering and Difference Expansion in Digital Images," in *Proc. IEEE Int. Conf. Commun., Netw. Satellite (COMNETSAT)*, 2022, pp. 33–38, doi: 10.1109/COMNETSAT56033.2022.9994516.

[27] A. A. Abdulla, S. A. Jassim, and H. Sellahewa, "Efficient high-capacity steganography technique," in *Proc. SPIE 8755, Mobile Multimedia/Image Processing, Security, and Applications*, 875508, May 2023, doi: 10.1117/12.2018994.

[28] P.-H. Kim, E.-J. Yoon, K.-W. Ryu, and K.-H. Jung, "Data-Hiding Scheme Using Multidirectional Pixel-Value Differencing on Colour Images," *Secur. Commun. Netw.*, vol. 2019, pp. 1–11, Oct. 2019, doi: 10.1155/2019/9038650.

[29] A. K. Sahu, G. Swain, M. Sahu, and J. Hemalatha, "Multi-directional block based PVD and modulus function image steganography to avoid FOBP and IEP," *J. Inf. Secur. Appl.*, vol. 58, p. 102808, May 2021, doi: 10.1016/j.jisa.2021.102808.

[30] A. Fahim and Y. Raslan, "Optimized steganography techniques based on PVDS and genetic algorithm," *Alexandria Eng. J.*, vol. 85, pp. 245–260, Dec. 2023, doi: 10.1016/j.aej.2023.11.013.

[31] J. Jiang, Z. Wang, and X. Zhang, "Image-to-Image Steganography based on multimodal generative model," *Signal Process.*, vol. 238, p. 110106, Jan. 2026, doi: 10.1016/j.sigpro.2025.110106.

[32] Y. Chen, H. Wang, Y. Cui, G. Shen, C. Gu, Y. Liu, and H. Wu, "A multi-level additive distortion method for security improvement in palette image steganography," *J. Vis. Commun. Image Represent.*, vol. 110, p. 104463, Jul. 2025, doi: 10.1016/j.jvcir.2025.104463.

[33] D.-C. Wu and Z.-N. Shih, "Image Steganography by Pixel-Value Differencing Using General Quantization Ranges," *Comput. Model. Eng. Sci.*, vol. 141, no. 1, pp. 353–383, 2024, doi: 10.32604/cmes.2024.050813.

[34] J. Gopika Rajan and R. S. Ganesh, "Dynamic pixel shuffling and hash LSB steganography with RC4 encryption: A robust data security framework," *Expert Syst. Appl.*, vol. 279, p. 127403, Jun. 2025, doi: 10.1016/j.eswa.2025.127403.

[35] Viterbi School of Engineering, "Signal and Image Processing Institute (USC), University of Southern California," [Online]. Available: https://sipi.usc.edu/database/database.php?volume=misc. Accessed: May 01, 2024.

[36] Lorem Ipsum, "The Standard Lorem Ipsum Passage," [Online]. Available: https://www.lipsum.com. Accessed: May 01, 2024.