Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061246

ARTICLE

FuzzyStego: An Adaptive Steganographic Scheme Using Fuzzy Logic for Optimizing Embeddable Areas in Spatial Domain Images

Mardhatillah Shevy Ananti[1], Adifa Widyadhani Chanda D'Layla[1], Ntivuguruzwa Jean De La Croix[1,2] and Tohari Ahmad[1,*]

[1]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia
[2]College of Science and Technology, University of Rwanda, Kigali, 3900, Rwanda

*Corresponding Author: Tohari Ahmad. Email: tohari@if.its.ac.id or tohari@its.ac.id

Received: 20 November 2024; Accepted: 24 February 2025; Published: 09 June 2025

ABSTRACT: In the evolving landscape of secure communication, steganography has become increasingly vital to secure the transmission of secret data through an insecure public network. Several steganographic algorithms have been proposed using digital images with a common objective of balancing a trade-off between the payload size and the quality of the stego image. In the existing steganographic works, a remarkable distortion of the stego image persists when the payload size is increased, making several existing works impractical to the current world of vast data. This paper introduces FuzzyStego, a novel approach designed to enhance the stego image's quality by minimizing the effect of the payload size on the stego image's quality. In line with the limitations of traditional methods like Pixel Value Differencing (PVD), Transform Domain Techniques, and Least Significant Bit (LSB) insertion, such as image quality degradation, vulnerability to processing attacks, and restricted capacity, FuzzyStego utilizes fuzzy logic to categorize pixels into intensity levels: Low (L), Medium-Low (ML), Medium (M), Medium-High (MH), and High (H). This classification enables adaptive data embedding, minimizing detectability by adjusting the hidden bit count according to the intensity levels. Experimental results show that FuzzyStego achieves an average Peak Signal-to-Noise Ratio (PSNR) of 58.638 decibels (dB) and a Structural Similarity Index Measure (SSIM) of almost 1.00, demonstrating its promising capability to preserve image quality while embedding data effectively.

KEYWORDS: Data hiding; digital images; fuzzy selection; information security; steganography

1 Introduction

Authors Steganography, a data hiding technique, has recently received significant attention in securing sensitive information by concealing information within various types of digital media, including audio [1], video [2], and images [3]. Steganography enhances the security of sensitive information by embedding it into seemingly innocuous content, rendering the hidden data undetectable to the casual observer [4]. In image steganography, the original image used to host the concealed data is the cover image [5]. In contrast, the modified image containing the hidden information is called the stego image [6]. This practice dates back centuries, with historical examples including using invisible ink or embedding messages in poetic texts. Modern steganography, however, has evolved significantly, employing advanced algorithms and computational techniques to embed data in digital images [7–10]. These algorithms subtly manipulate pixel values or use complex machine learning models to ensure the hidden data is imperceptible, thus enabling covert communication and robust data protection.

Despite its advancements, the field of image steganography faces several challenges, primarily centred around the balance between the quality of the stego image and its capacity to store data [11,12]. One of the fundamental problems is that as the amount of embedded data increases, the visual quality of the stego image tends to deteriorate, making it more susceptible to detection through visual inspection or steganalysis attacks [13–15]. This trade-off between data embedding capacity and image quality remains a critical issue. Additionally, current techniques often fail to perform consistently across different types of images, such as medical or natural images, further restricting their broader applicability. Many steganographic techniques, especially those using the Least Significant Bit (LSB) method [16,17], are susceptible to such attacks, compromising the hidden data's security and integrity [18]. However, significant progress has been made in developing image steganography techniques, but several limitations persist. The lack of adaptability across different image types to yield a high embeddable number of pixels, the challenge of maintaining high stego image quality while embedding substantial amounts of data, and the vulnerability to steganalysis attacks are vital areas that require further research in steganography.

To address the limitations highlighted in the state-of-the-art, this study presents a novel steganographic method named FuzzyStego. This method utilizes fuzzy logic to adaptively arrange the pixels based on their intensity levels with Low (L), Medium-Low (ML), Medium (M), Medium-High (MH), and High (H). The data embedding is done based on the intensity of the pixels depending on the pixel's classification from the fuzzy inference system. The embedding process considers a maximum of three least significant bits (LSB) for the most embeddable pixels and one LSB for the least embeddable pixel. The fuzzy paradigm makes FuzzyStego a method that embeds data based on the image's intensity's adaptiveness. By categorizing the pixels, FuzzyStego adapts the data embedding process according to the characteristics of each pixel, which helps optimize the number of bits embedded in each pixel, enhancing the payload capacity and imperceptibility, reflecting the stego image's quality. The key contributions of the FuzzyStego include:

- Embeddable pixel arrangement based on intensity using fuzzy logic to optimally select the pixels to host the data: Fuzzy logic is utilized to classify cover image pixels based on their intensity values. This classification identifies pixels that are more suitable for embedding secret bits. By adaptively selecting these embeddable pixels, the number of bits per pixel varies for enhanced security and efficiency of the embedding process.
- By extrapolating low and medium-intensity pixels, this approach allows for the embedding of multiple secret bits (two bits in medium-intensity pixels and three bits in low-intensity pixels) rather than the single bit concealed by algorithms in previous studies: FuzzyStego uses adaptive logic to determine which pixels can host three, two, or one bit. This approach increases the payload capacity without compromising the visual quality of the stego image. Therefore, FuzzyStego contributes to achieving a balance between high embedding capacity and image integrity.
- Introducing a cross-image type algorithm demonstrating nearly identical performance in general-purpose and medical imaging contexts: The proposed FuzzyStego introduces a cross-image type algorithm that performs consistently across general-purpose and medical imaging contexts. Utilizing adaptive logic to allocate bits per pixel significantly enhances payload capacity while maintaining image integrity.

The remaining part of this paper is divided into four sections. Section 2 examines the current advancements and outlines the gaps the proposed method, FuzzyStego, aims to fill, with further details provided in Section 3. Section 4 presents the experimental results and their interpretation, while Section 5 offers the conclusion.

2 Literature Review

Aligning with the general concern of steganography, which focuses on embedding hidden information within a host medium [6], recent advancements aim to enhance security and invisibility. Numerous techniques have been developed to improve the efficiency and imperceptibility of the concealed data [19–21]. Additionally, these methods address critical challenges related to detection and data integrity.

Siddiqui proposed a dynamic three-bit image steganography algorithm for medical and e-healthcare systems [19]. This approach introduced a new steganography algorithm specifically for images, such as MRI scans, which divide an image into three intensity regions, low, medium, and high, and embed data using one to three LSBs based on the region. The main objective of their method is to ensure that the hidden data remains undetectable using a Peak Signal Noise Ratio (PSNR), indicating visual distortion and successful concealment of information. Despite these advancements, there are some limitations to consider. The algorithm is designed for grayscale images, restricting its use in color image manipulation scenarios. While LSBs effectively maintain image quality, they also introduce vulnerabilities to steganalysis attacks. Optimizing the algorithm to use Fuzzy Logic will enhance data concealment and resistance to steganalysis.

Zaini proposed a steganographic approach that utilized the idea of difference expansion by employing pixel block differentiation to improve the quality of stego images [20] based on the adjacent pixels differencing from [22,23]. This method organizes pixels from the image into blocks of 31 and calculates the difference between them. The secret data is embedded if the difference between the pixels falls within a specific range of −10 and 10. This method aims to enhance imperceptibility while accommodating amounts of data.

Additionally, it utilizes a key to keep track of data positions for extraction and image retrieval. Although this technique shows promising outcomes, the fixed block size of 31 pixels may not be suitable for all types of images and payload sizes, potentially restricting its use. Moreover, it lacks consideration for resilience against steganalysis attacks. These challenges could be addressed by adopting a more adaptive scheme to distribute the data across the image and reduce the impact on image distortion by optimizing pixel differences, like large payloads. Ding, aiming to enhance the capacity of medical images to embed secret data, proposed an improved reversible data hiding approach using a difference expansion algorithm [24]. The proposed algorithm utilized the spaces between neighboring pixels by modifying their differences to embed data while preserving the image's appearance, referring to the previous paradigm in [25]. Despite the progress achieved with this technique, some drawbacks need to be addressed. One major issue is balancing the concealed image's quality and capacity to carry information. As more data is hidden within the image, its quality deteriorates, indicating that maintaining high image fidelity remains a concern across data loads. This can be resolved by using an adaptive threshold in the algorithm, which changes based on local image features, resulting in an improvement in capacity and reducing unnecessary distortions of the stego image.

Moreover, the study in [21] presents a novel technique for reversible data hiding in digital images by dividing the image into regions of varying complexity, namely smooth and rough areas. To maximize the data embedding capacity, the method embeds three bits per pixel in smoother regions, where redundancy is high, and one bit per pixel in rougher areas. This approach outperforms conventional prediction error expansion-based techniques by leveraging greater redundancy in smoother regions. A key innovation in this scheme is the pixel selection mechanism, which reduces the number of pixels that need to be shifted, thus preserving the visual quality of the stego image and reducing distortions. Experimental evaluations show that this method surpasses many existing techniques regarding rate-distortion performance, achieving better trade-offs between payload capacity and image quality.

However, while the technique shows significant promise, further improvements are needed to optimize its performance, particularly when handling larger datasets. As data volume increases, maintaining a balance

between high embedding capacity and the preservation of image quality becomes more challenging [26,27]. The method's capacity to handle substantial amounts of data without degrading the stego image's visual integrity remains a critical area for refinement. Recent advances in steganography have tackled these challenges by optimizing the balance between data embedding capacity and image quality [8,10]. However, achieving this balance in high payload scenarios, where more data is embedded, remains a persistent issue. Developing more refined algorithms that adapt to varying image characteristics is crucial while ensuring that the stego image's quality is not compromised. In this context, the current study introduces an advanced steganographic algorithm based on fuzzy logic for pixel intensity classification to refine the embedding process and maintain image quality, even with larger data payloads.

3 Proposed Method: FuzzyStego Approach

This article presents a steganographic method that involves an adaptive approach to select the embeddable cover image pixels with a fuzzy logic-based paradigm. The embedding process, whose portrait is given in Fig. 1, starts by loading the cover image and the secret data in binary format. Next, the pixels in the cover image are classified into five intensity levels using predefined thresholds categorized as Low (≤50), Medium Low (51–100), Medium (101–150), Medium High (151–200), and High (201–255). The number of secret bits varies depending on the intensity level of each pixel in the image.

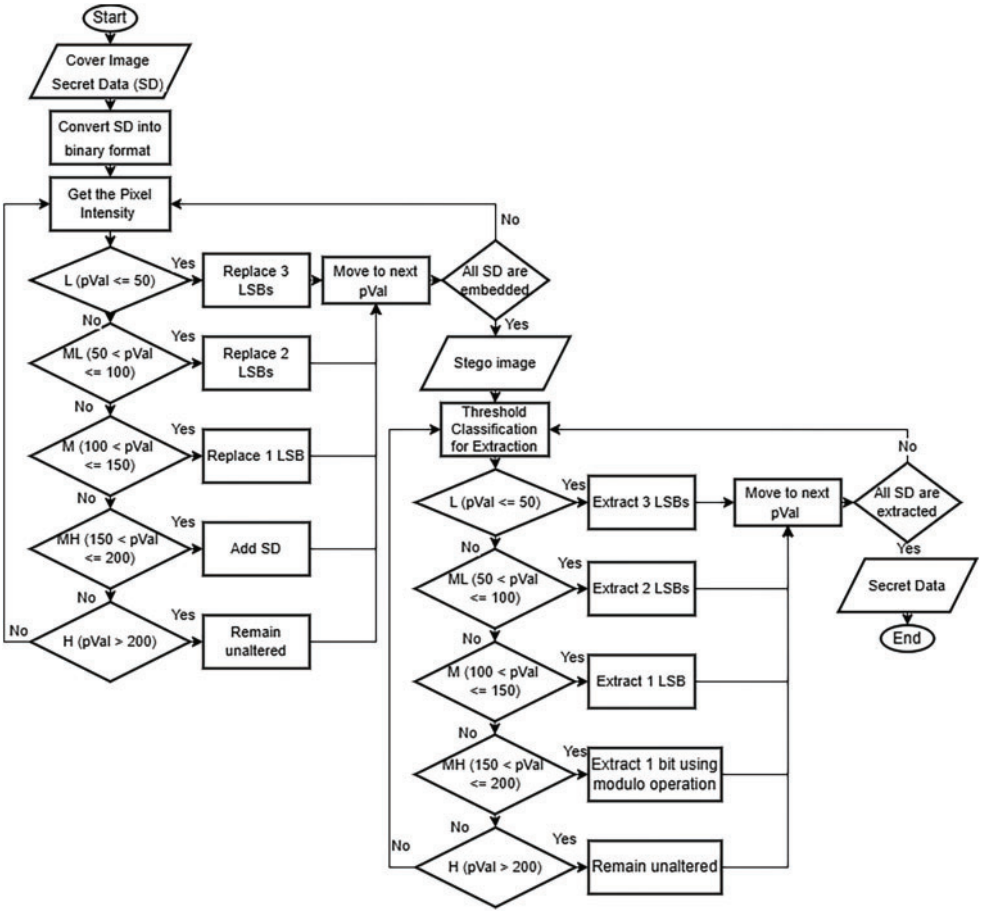

Figure 1: A general flowchart for the proposed FuzzyStego

The proposed FuzzyStego is mathematically expressed as $C = 3N_{low} + 2N_{med-low} + 1N_{med} + 1N_{med-high}$, for the embedding process with N_{low}, $N_{med-low}$, N_{med}, $N_{med-high}$ the pixel counts for the corresponding ranges, consider $p(i,j)$ as a pixel intensity at position (i,j) in the cover image. It is important to note that $p(i,j) \in [0,255]$. S also represents the secret data in binary form, and $LSB_k(p(i,j))$ represents the $k-$th least significant bit of the pixel $p(i,j)$. For the pixels satisfying the condition in Eq. (1), with the EmbedLSBs function is got by Eq. (2) taking $p'(i,j)$ as the pixel of the stego image, the embedding process follows the formula in Eq. (3). For the pixels satisfying the condition in Eq. (4), the embedding process follows Eq. (5), for the pixels under the condition in Eq. (6), the embedding process follows Eq. (7), for the pixels under the condition in Eq. (8), the embedding process follows Eq. (9), and for the pixels under the condition in Eq. (10), the stego pixels and the cover pixels are kept unaltered as of Eq. (11). The extraction process that considers b as 1, is expressed in Eq. (12) for the secret data and as Eq. (13) for the image pixels.

$$p(i,j) \leq 50 \tag{1}$$

$$EmbedLSBs\left(P\left(i,j\right),S,k\right) = \left\lfloor \frac{p(i,j)}{2^k} \right\rfloor .2^k + Bits(S,k) \tag{2}$$

$$p'\left(i,j\right) = EmbedLSBs(p\left(i,j\right),S,k;k=3) \tag{3}$$

$$51 \leq p(i,j) \leq 100 \tag{4}$$

$$p'\left(i,j\right) = EmbedLSBs(p\left(i,j\right),S,k;k=2) \tag{5}$$

$$101 \leq p(i,j) \leq 150 \tag{6}$$

$$p'\left(i,j\right) = EmbedLSBs(p\left(i,j\right),S,k;k=1) \tag{7}$$

$$151 \leq p(i,j) \leq 200 \tag{8}$$

$$p'\left(i,j\right) = \begin{cases} p\left(i,j\right)+1 \, if \, s \neq LSB_1\left(p\left(i,j\right)\right) \; and \; p(i,j) < 255, \\ p\left(i,j\right)-1 \, if \, s \neq LSB_1\left(p\left(i,j\right)\right) \; and \; p\left(i,j\right) = 255, \\ p\left(i,j\right) \quad otherwise. \end{cases} \tag{9}$$

$$201 \leq p(i,j) \leq 255 \tag{10}$$

$$p'\left(i,j\right) = p(i,j) \tag{11}$$

$$S = \begin{cases} \{LSB_1, LSB_2, LSB_3\}\left(p'(i,j)\right) & for \, p'(i,j) \leq 50+b \\ \{LSB_1, LSB_2\}\left(p'(i,j)\right) & for \, 50+b \leq p'(i,j) \leq 100+b \\ p'\left(i,j\right) mod \, 2 & for \, p'\left(i,j\right) > 150+b \end{cases} \tag{12}$$

$$p\left(i,j\right) = \begin{cases} \left\lfloor \frac{p'(i,j)}{2^k} \right\rfloor .2^k & for \, k = 3,2,1 (based \, on \, the \, pixel \, value) \\ p'\left(i,j\right) - S & for \, p'\left(i,j\right) > 150+b \end{cases} \tag{13}$$

The proposed FuzzyStego employs a Mamdani Fuzzy Inference System (FIS) as the core mechanism for pixel classification, ensuring an adaptive and practical approach to embedding data within a cover image. Mamdani FIS, known for its intuitive rule-based framework, is particularly suited for handling the uncertainty and variability inherent in pixel intensity values. Below, we detail the components of the FuzzyStego Mamdani FIS: Input Fuzzification, Fuzzy Rule Base, Fuzzy Inference, and Defuzzification.

Input Fuzzification: The grayscale intensity of each pixel $p(i,j)$ where $p\left(i,j\right) \in [0,255]$ is transformed into fuzzy linguistic terms based on predefined membership functions. These terms correspond to five intensity levels: Low (≤50), Medium Low (51–100), Medium (101–150), Medium High (151–200), and High (201–255). These membership functions, typically triangular, define the degree to which a pixel belongs to each intensity level.

Fuzzy Rule Base: The rule base is composed of intuitive if-then rules that facilitate the classification of pixel intensities into predefined categories. Each rule maps a specific intensity range of the pixel $p(i, j)$ to a corresponding category, ensuring systematic and adaptive classification. The following rules are followed to organize the pixels based on their intensities. If the intensity $p(i, j)$ falls within the Low range (0–50), it is classified into the Low category, allowing for higher data embedding due to the low perceptual sensitivity in this range. If the intensity $p(i, j)$ lies within the Medium Low range (51–100), it is categorized as Medium Low, permitting moderate embedding. For intensities in the Medium range (101–150), the pixel is assigned to the Medium category, balancing embedding capacity and visual quality. Pixels with intensities in the Medium High range (151–200) are classified as Medium High, where embedding is more restricted to minimize visual distortion. Finally, if the intensity $p(i, j)$ falls within the High range (201–255), it is assigned to the High category, ensuring no significant alterations are made to these visually sensitive pixels.

Fuzzy Inference: The Mamdani FIS uses a max-min inference approach, evaluating the rules based on the membership values of the input and determining the degree of truth for each rule. This step results in a fuzzy output set for each pixel.

Defuzzification: The output fuzzy set is converted into a crisp value using the centroid method, determining the most representative intensity level. This output guides the embedding process, defining how many bits can be embedded in each pixel.

The use of fuzzy logic in FuzzyStego is essential for optimizing the balance between embedding capacity and image quality through an adaptive pixel classification approach. Pixels are categorized into five intensity levels, each tailored to the sensitivity of the human visual system and the capacity for imperceptible data embedding. Dark pixels (Low intensity, ≤50) can tolerate significant alterations without visible artifacts, allowing up to 3LSBs to be modified, maximizing embedding capacity. In the Medium Low range (51–100), moderate distortion is acceptable, permitting the replacement of two LSBs while maintaining visual quality. Midtone regions (Medium intensity, 101–150), which are more sensitive to changes, allow only one LSB modification to minimize visible artifacts. Bright areas (Medium High intensity, 151–200) have limited tolerance for distortion, requiring conditional embedding strategies like incrementing or decrementing pixel values to ensure imperceptibility. High-intensity pixels (201–255) are left unaltered to preserve the integrity of visually critical regions. The Mamdani fuzzy inference system (FIS) is pivotal in enabling this adaptive adjustment of embedding strategies based on pixel intensity. The system achieves high imperceptibility by leveraging fuzzy rules and predefined thresholds, balancing data security with visual fidelity.

3.1 Data Embedding

The FuzzyStego method for embedding secret data consists of three stages: cover image classification, secret data embedding, and post-embedding conversion. These steps that follow the pseudocodes in Algorithm 1 ensure efficiency in embedding the secret data while preserving the visual integrity of the stego image. It adjusts its embedding capacity according to the intensity region to maintain image fidelity.

Algorithm 1: Embedding process

Input: Cover Image, Secret data
Load the Cover Image and Secret Data
Convert Secret Data into binary format
For each pixel in the Cover Image:
Get the pixel intensity
Repeat
If the pixel intensity is less than or equal to the low threshold then
Embed 3 bits into the 3 least significant bits of the pixel
Update the pixel value with the embedded bits
Increment the index for secret data bits by 3
Else if the pixel intensity is greater than the low threshold and less than or equal to the medium-low threshold then
Embed 2 bits into the 2 least significant bits of the pixel
Update the pixel value with the embedded bits
Increment the index for secret data bits by 2
Else if the pixel intensity is greater than the medium-low threshold and less than or equal to the medium threshold then
Embed 1 bit into the least significant bit of the pixel
Update the pixel value with the embedded bit
Increment the index for secret data bits by 1
Else if the pixel intensity is greater than the medium threshold and less than or equal to the medium-high threshold
then
Add 1 bit to the pixel value
Update the pixel value by adding the bit
Increment the index for secret data bits by 1
Else if the pixel intensity is greater than the medium-high threshold then
Do not embed any bits. Leave the pixel value unaltered
Until all secret data bits are embedded
Output: Stego Image

Step (1) Fuzzy-Based Cover Pixels Classification

In the first step, the cover image undergoes classification of its pixels according to their intensity levels. This plays a role in determining the capacity for embedding data without affecting image quality. The pixel intensity levels are divided into five levels: L, ML, M, MH, and H. The classification of pixels enables adaptive data embedding by allowing more secret bits to be hidden in pixels with intensity levels while making minimal adjustments to pixels with higher intensity to prevent noticeable deterioration in visual quality.

Step (2) Secret Data Embedding

After the pixels are categorized and labeled accordingly, the secret data are embedded into the cover image in the following process:

- L: For pixels with intensity Ip <= tL, 3LSBs are replaced to embed secret data, allowing for higher capacity without causing noticeable distortion to the human eye.

- ML: For pixels with intensity tL < Ip <= tML, 2LSBs are replaced to embed secret data, slightly decreasing the hiding capacity to uphold the visual quality of the stego image.
- M: Pixels with intensity tML < Ip <= tM, 1LSB replaced to embed secret data, making them more resilient to distortion and requiring fewer embedded bits.
- MH For pixels with intensity tM < Ip <= tMH, the binary value of the secret bit is added to the pixel value. This adjustment ensures minimal visual impact due to their sensitivity to changes.
- H: Pixels with intensity Ip > tMH, the binary value of the secret bit is added to the pixel value. The same applies to medium-high intensity; these pixels are sensitive to visuals, so fewer modifications are allowed.

3.2 Data and Cover Extraction

Like the embedding process, the extraction process is organized in a couple of steps. As given in the pseudocodes from Algorithm 2, the steps include stego preprocessing and extraction.

Algorithm 2: Extraction process

Input: Stego Image
Load the Stego Image
For each pixel in the Cover Image:
Get the pixel intensity
Repeat
If the pixel intensity is less than or equal to the low threshold then
Extract 3 bits from the 3 least significant bits of the pixel
Append the extracted bits to the list of secret bits
Update the Cover Image by resetting the least significant bits of the pixel
Else if the pixel intensity is greater than the low threshold and less than or equal to the medium-low threshold then
Extract 2 bits from the 2 least significant bits of the pixel
Append the extracted bits to the list of secret bits
Update the Cover Image by resetting the least significant bits of the pixel
Else if the pixel intensity is greater than the medium-low threshold and less than or equal to the medium threshold then
Extract 1 bit from the least significant bit of the pixel
Append the extracted bit to the list of secret bits
Update the Cover Image by resetting the least significant bit of the pixel
Else if the pixel intensity is greater than the medium threshold and less than or equal to the medium-high threshold
then
Extract 1 bit using the modulo operation
Update the pixel value by adding the bit
Increment the index for secret data bits by 1
Else if the pixel intensity is greater than the medium-high threshold then
Do not extract any bits. Leave the pixel value unaltered
Until all secret data bits are extracted
Output: Cover Image, Secret Data

Step (1) Preprocessing the Stego Image

Each pixel of the stego image is processed to identify any changes made by the embedded secret data. The intensity levels of the pixel values are classified as in the embedding process. This step is crucial as it helps determine how many secret bits were hidden in each pixel and how to restore the cover image. The stego image keeps its original dimensions through this process to preserve the integrity of the pixel structure.

Stage (2) Extracting the Data and the Cover Image

The extraction of the secret data and cover image starts by classifying each pixel based on intensity levels (L, ML, M, MH, and H) following the set of thresholds from the data embedding method:

- L: Extract 3LSBs from the pixel and treat them as secret bits, while the rest of the bits are used to reconstruct the cover image by resetting those 3LSBs to zero.
- ML: Extract 2LSBs, and the cover image is restored by setting this LSB to zero.
- M: Extract 1LSB and restore the cover image by setting this LSB to zero.
- MH: Extract the secret data using the modulo operation while restoring the cover image by subtracting the embedded secret bit value from the pixel value.
- H: No secret data is extracted, and the pixel value remains unaltered for the original cover image.

3.3 Dataset and Evaluation Metrics

The proposed FuzzyStego method was tested using images from the SIPI database [28]. A collection of experimental images often used in steganography research. These images are grayscales with a size of 512×512 pixels. To assess the method's effectiveness, random secret bits are generated using the Lorem Ipsum [29] and stored in a base-5 format with file sizes varying from 1 kb to 100 kb. Using test images sourced from the SIPI database and randomly generated secret data bits forms a strong foundation for testing the efficiency of the suggested steganographic method, evaluating its performance, and enabling comparisons with established procedures for a detailed analysis of the effectiveness and potential enhancements to by FuzzStego.

To evaluate the quality of images effectively after embedding secret data in them with the FuzzyStego method, we use PSNR and SSIM. In addition, we compute the embedding capacity to identify the embeddability of each image with FuzzyStego. The PSNR is calculated using Eq. (14), and the SSIM using Eq. (15). In these computations, the $C(i, j)$ stands for the cover image, while $S(i, j)$ refers to the stego image that contains the embedded data. The SSIM calculation also incorporates the parameters δ_i and δ_j, representative of average pixel intensities, and α_i and α_j indicative of intensity variations in horizontal and vertical directions. Moreover, $\alpha_{i,j}$ considers the covariance between these intensity variations, enabling SSIM to provide a comprehensive evaluation of the similarity in structure between the images.

$$PSNR = 10 \times \log_{10} \frac{255^2}{\frac{1}{a \times b} \sum_{i=1}^{a} \sum_{j=1}^{b} (C(i,j) - S(i,j))^2} \tag{14}$$

$$SSIM = \frac{(2\delta_i \delta_j + C_i)(2\alpha_{i,j} + S_i)}{(\delta_i{}^2 + \delta_j{}^2 + C_i)(\alpha_i{}^2 + \alpha_j{}^2 + S_i)} \tag{15}$$

4 Results and Discussions

4.1 Experimental Results

The data presented in Table 1 offers a detailed analysis of image quality metrics, specifically PSNR and SSIM, across different cover images and payload sizes (ranging from 1 to 100 kilobits (kb)). The data reveals

that aerial and fishing boats exhibit high PSNR values at 1 kb, indicating excellent quality even in smaller file sizes. However, as the payload size increases, PSNR values decline for all images, signaling quality loss due to the higher payload. Conversely, images like Airplane maintain more stable PSNR values at larger file sizes, suggesting they are more resistant to the quality degradation caused by the steganographic payload.

Table 1: Obtained PSNR and SSIM results

Images	20 kb		40 kb		60 kb		80 kb		100 kb	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Aerial	59.615	0.999	55.866	0.999	53.564	0.999	51.682	0.999	51.840	0.999
Airplane	55.252	0.999	58.212	0.999	58.212	0.999	58.212	0.999	58.212	0.999
Car and APCs	51.193	0.997	48.144	0.995	46.697	0.993	45.831	0.990	45.268	0.988
Fishing Boat	57.458	0.999	52.216	0.999	48.005	0.996	45.304	0.991	44.424	0.992
Pixel ruler	58.274	1.000	55.849	1.000	53.915	1.000	52.425	1.000	53.557	1.000
Stream and bridge	56.385	0.999	53.095	0.999	49.408	0.997	47.898	0.997	46.733	0.997
Tank	56.191	0.999	52.273	0.999	49.727	0.998	48.133	0.997	47.773	0.997
Truck	52.017	0.999	49.168	0.998	47.453	0.997	46.107	0.995	45.517	0.995
Peppers	52.561	0.998	49.202	0.996	47.195	0.995	45.714	0.993	44.418	0.990
Barbara	55.702	0.999	52.134	0.997	50.759	0.997	49.402	0.995	48.188	0.994
Zelda	54.094	0.997	51.055	0.995	49.253	0.993	48.276	0.992	47.672	0.991
Baboon	51.428	0.99	49.224	0.999	47.740	0.999	46.726	0.998	46.164	0.998

Regarding structural quality, the SSIM values in Table 1 show that most images retain their structural integrity even with high payload steganography, as SSIM scores remain close to 1.0. This indicates that the proposed FuzzyStego method preserves vital structural elements such as edges and textures despite pixel-level degradation from the embedded data. Images like Aerial and Pixel ruler maintain nearly perfect SSIM values across all payload sizes, while images like Car, APCs, and Peppers show slight reductions in SSIM. However, these values still suggest minimal perceptual quality loss. The strength of the FuzzyStego method lies in its ability to effectively balance both PSNR and SSIM, preserving high visual and structural quality while embedding hidden data. It minimizes the adverse effects of data embedding, ensuring the hidden information remains secure without significantly degrading the overall image quality.

Fig. 2 presents a scatter plot to illustrate the relationship between the SSIM and the Embedding Capacity (EC) for the test images under the FuzzyStego, highlighting the efficiency of the proposed method. Most images exhibit SSIM values close to 1, indicating excellent structural preservation, with minimal perceptual differences between the original and stego images. This implies that even with embedded data, the visual integrity of the images remains almost intact. Images like Aerial and Pixel ruler demonstrate higher EC values (around 0.7), meaning they can embed substantial data while retaining near-perfect visual quality. In contrast, images such as Baboon and Zelda show lower EC values (around 0.2–0.4), suggesting they have a more limited capacity for data embedding without experiencing slight structural degradation, as reflected in their slightly lower SSIM values (though still above 0.994). This indicates that FuzzyStego effectively balances embedding capacity and visual quality across different images.

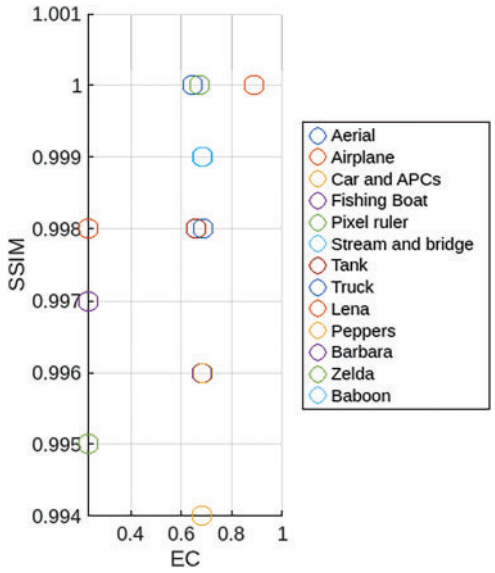

Figure 2: Illustration of the relationship between the SSIM and the EC

The scatter illustration in Fig. 3 presents the relationship between the PSNR and the EC, highlighting the performance of the proposed FuzzyStego technique. The PSNR, which reflects the quality of an image after steganographic embedding, remains consistently high across various images, with values ranging from 50 dB to 59 dB. This demonstrates FuzzyStego's ability to preserve image quality. Additionally, the EC varies from 0.1 to 0.8, indicating that FuzzyStego efficiently utilizes the image's capacity for data hiding while maintaining low distortion. Images such as Airplane cars and APCs achieve a remarkable balance, displaying high EC and PSNR values, which underscores the method's adaptability and efficiency.

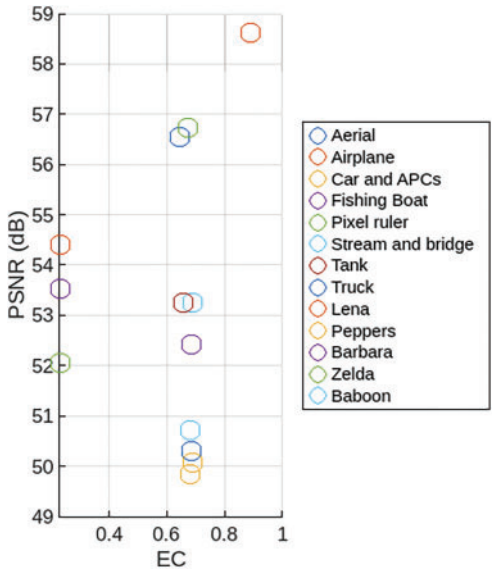

Figure 3: Illustration of the relationship between the PSNR and the EC

To demonstrate the cross-dataset performance of the proposed FuzzyStego method, Table 2 compares its PSNR and SSIM across medical images from [30,31] and general-purpose images from [32] datasets. The results confirm FuzzyStego's adaptability and efficiency in embedding data into various cover images while maintaining high image quality and structural integrity. FuzzyStego achieves promising PSNR and SSIM values for medical images, even for larger payloads. For instance, in the Brain image, the PSNR decreases from 65.482 dB (1 kb) to 47.887 dB (100 kb) while maintaining an SSIM above 0.97, reflecting excellent perceptual quality. Similar trends are observed for other medical images, such as Hand, Leg, and Head, with SSIM values consistently exceeding 0.95 across all payload sizes. This demonstrates FuzzyStego's ability to embed large amounts of data in medical images without significantly degrading visual quality or structural similarity. For general-purpose images, FuzzyStego also delivers robust performance. The Fountain image, for example, maintains a PSNR of 63.040 dB for 1 kb and 40.127 dB for 100 kb, with an SSIM consistently above 0.98. Similar performance is noted for images such as Car, Bridge, and Cars, where the SSIM values remain near 1.0 for smaller payloads and above 0.97 for larger ones. This indicates that FuzzyStego preserves general-purpose images' visual and structural quality across varying payload sizes.

Table 2: Results of the FuzzyStego under a cross-dataset experiment

Image type	Cover image	1 kb		20 kb		40 kb		80 kb		100 kb	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Medical Images from [30,31]	Brain	65.482	0.999	52.921	0.985	49.765	0.969	48.907	0.980	47.887	0.979
	Hand	65.184	1.000	49.051	0.995	44.837	0.991	41.393	0.983	40.426	0.980
	Leg	61.315	1.000	46.882	0.992	44.109	0.983	41.643	0.966	40.852	0.957
	Head	66.025	1.000	52.073	0.988	49.761	0.978	46.680	0.960	45.541	0.947
General Purpose Images from [32]	Fountain	63.040	1.000	44.019	0.997	40.686	0.990	40.128	0.988	40.127	0.987
	Car	60.809	1.000	44.420	0.991	41.223	0.983	40.989	0.986	40.988	0.985
	Bridge	69.706	1.000	44.612	0.992	42.203	0.988	42.033	0.988	42.033	0.988
	Cars	56.322	0.999	41.688	0.983	39.956	0.975	39.611	0.977	39.610	0.977

To further evaluate the proposed FuzzyStego method's practicality, we examine the embedding and extraction times, providing context with the computational environment used: an Intel® Core™ i7-1165G7 processor (2.80 GHz) and 16 GB RAM. The results in Table 3 show that embedding times increase with payload size while extraction times remain stable across different payloads and cover images. Smaller payload sizes, such as 1 kb, embedding times are minimal, ranging from 0.004 s for the "Airplane" and "Truck" images to 0.167. As payload sizes increase to 100 kb, embedding times naturally rise due to the increased data being processed. For instance, the embedding time for the "Aerial" image grows from 0.019 s at 1 kb to 1.0879 s at 100 kb, while the "Barbara" image increases from 0.155 s to 0.8785 s over the same range. It is also important to note that the extraction times remain consistent across all payload sizes, indicating the scalability and efficiency of the FuzzyStego algorithm for data retrieval. For example, the "Aerial" image shows only a slight variation in extraction time, from 0.390 s to 0.3909 s, regardless of the payload size. Similarly, images like "Truck" and "Airplane" maintain extraction times below 0.5 s, underscoring the algorithm's reliability. Specific images, such as "Airplane" and "Truck," exhibit consistently faster embedding and extraction times, particularly at smaller payloads. For instance, the "Truck" image requires only 0.004 s for embedding and 0.453 s for extraction at 1 kb.

Table 3: Embedding and extraction times in seconds for the proposed FuzzStego

Cover image	Payload size in kb							
	1		40		80		100	
	Time in seconds							
	Embedding	**Extraction**	**Embedding**	**Extraction**	**Embedding**	**Extraction**	**Embedding**	**Extraction**
Aerial	0.019	0.392	0.618	0.390	1.102	0.390	1.087	0.390
Airplane	0.004	0.099	0.293	0.111	0.285	0.098	0.292	0.097
Car and APCs	0.005	0.438	0.074	0.441	0.178	0.443	0.243	0.440
Fishing Boat	0.010	0.433	0.429	0.480	0.511	0.428	0.573	0.429
Pixel ruler	0.009	0.395	0.521	0.400	1.052	0.392	1.166	0.407
Stream and bridge	0.010	0.433	0.251	0.437	0.346	0.449	0.417	0.432
Tank	0.008	0.434	0.264	0.436	0.403	0.428	0.450	0.435
Truck	0.004	0.453	0.055	0.443	0.110	0.453	0.182	0.501
Peppers	0.008	0.432	0.226	0.425	0.472	0.429	0.543	0.420
Barbara	0.155	0.651	0.457	0.648	0.776	0.668	0.878	0.658
Zelda	0.153	0.673	0.331	0.665	0.502	0.673	0.577	0.672
Baboon	0.013	0.438	0.228	0.428	0.436	0.445	0.550	0.449

The scalability and real-world applicability of FuzzyStego are demonstrated by the high similarity observed between the cover and stego images across all tested payload sizes. This is further supported by the close resemblance of their histograms, even at larger payload capacities, underscoring FuzzyStego's suitability for modern, interconnected systems. Fig. 4 displays the histograms for two sample test images (one being a general purpose image, another being a medical image) of both cover and stego images generated using the FuzzyStego method. A detailed comparison of these histograms reveals a high degree of similarity between the cover and stego images, underscoring FuzzyStego's ability to embed data effectively while maintaining the visual integrity of the original image. The minimal differences in the histogram distributions between the cover and stego images highlight the method's efficiency in concealing the embedded information, making the stego images virtually indistinguishable from their cover counterparts. This characteristic confirms the robustness of FuzzyStego and emphasizes its suitability for real-world applications.

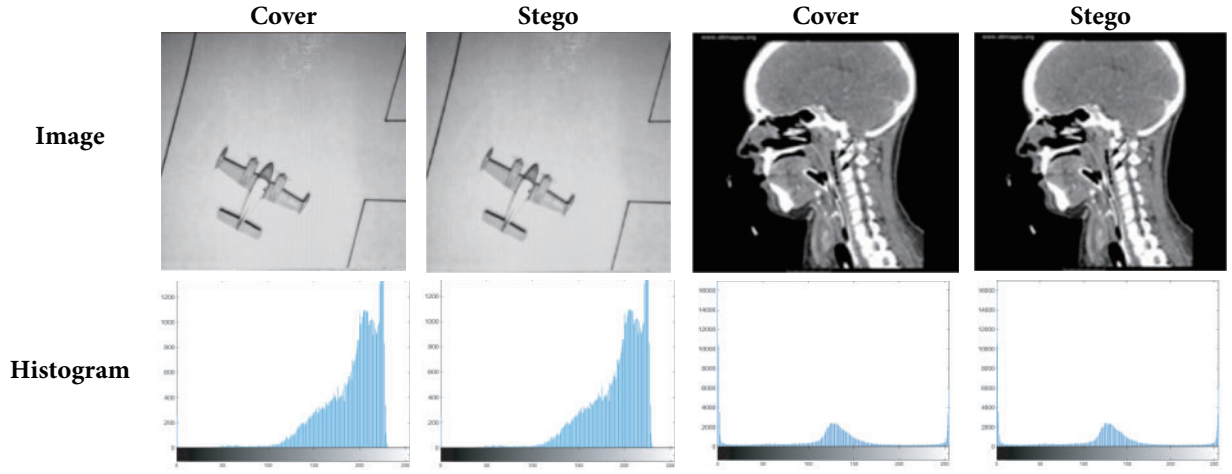

Figure 4: Histograms for cover and stego images using FuzzyStego

4.2 FuzzyStego Security Analysis

To assess the robustness of FuzzyStego against steganalysis and its resilience to compression attacks, Table 4 presents the detection accuracy of stego images generated using FuzzyStego, tested against the steganalysis algorithms in [15,33]. Stego images are created using the FuzzyStego algorithm, with payload sizes of 60, 80, and 100 kb. The results in Table 1 indicate the algorithm's robustness, with detection accuracy consistently remaining below 30% across all test scenarios. The highest detection accuracy, recorded with preprocessed images subjected to a strong steganalysis attack, is just 29.33%, highlighting the effectiveness of FuzzyStego in resisting steganalysis.

Table 4: Detection accuracy in percentage (%) of the proposed method by steganalysis attacks

Staganalysis algorithm	Payload capacity		
	60 kb	80 kb	100 kb
Algorithm in [15]	17.991	20.490	26.048
Algorithm in [33]	17.818	22.999	29.337

Additionally, Table 5 includes the average PSNR and SSIM values for the FuzzyStego applied to both compressed cover and stego images to evaluate the resilience of FuzzyStego against compression attacks. The compression technique used is Huffman Coding, a commonly applied compression algorithm known for its relatively high distortion. The PSNR values of the compressed stego images range from 33.736 (for the "Stream and Bridge" image) to 41.143 (for the "Pixel Ruler" image). These values indicate that the stego images retain good quality even after compression. Higher PSNR values, such as the 41.143 achieved for the "Pixel Ruler" image, suggest minimal distortion, whereas lower values, like 33.736 for "Stream and Bridge," imply slightly more distortion. However, these values are still within an acceptable range for most practical applications. Similarly, the SSIM values, which measure the structural similarity between the original and compressed images, remain consistently high across all test images. SSIM values range from 0.878 (for the "Tank" image) to 0.993 (for the "Pixel Ruler" image), with most values exceeding 0.9. An SSIM above 0.9 typically signifies that the compressed image preserves its structural features quite well, further supporting the robustness of FuzzyStego in maintaining the integrity of stego images after compression.

Table 5: Average PSNR and SSIM of the FuzzyStego under huffman compression

Cover image	PSNR in dB	SSIM
Aerial	34.530	0.943
Airplane	38.993	0.940
Car and APCs	35.109	0.895
Fishing Boat	35.121	0.912
Pixel ruler	41.143	0.993
Stream and bridge	33.736	0.945
Tank	34.409	0.878
Truck	34.961	0.901

4.3 Comparative Analysis of FuzzyStego and Existing Methods

4.3.1 Rationale and Scientific Basis of the Selection of the Comparison Methods

The selection of comparison methods for the FuzzyStego approach is guided by the need to evaluate its performance against diverse techniques that exhibit unique strengths in data embedding, extraction, and steganographic robustness. Each comparison method emphasizes attributes like embedding capacity, imperceptibility, computational efficiency, and security, aligning with the core objectives of FuzzyStego. The methods proposed [7,8,34–36] are selected due to their ability to utilize pixel value differences for adaptive data embedding. These methods embed secret bits by analyzing consecutive pixel pairs and their differences, ensuring high capacity in smooth areas and minimal distortion in edge regions. By comparing with the method in [10], the adaptability of FuzzyStego in varying intensity regions can be effectively assessed, as both approaches prioritize adaptive embedding. However, FuzzyStego's fuzzy-based classification introduces finer-grained embedding control to enhance the stego image's visual quality.

Moreover, the Genetic Algorithm-Enhanced Pixel Value Differencing (GA-IPVD) technique proposed in [26] is selected as a robust benchmark because it uses genetic algorithms to optimize pixel selection for data embedding. FuzzyStego's fuzzy logic-based classification provides an alternative to GA's optimization by focusing on pixel intensity levels. Comparing these methods highlights FuzzyStego's capacity to balance imperceptibility and simplicity without requiring iterative optimization. Therefore, the selected methods for comparison reflect a broad spectrum of strategies in steganography, from pixel-based adaptations to

complexity-driven and optimization-enhanced techniques. By comparing FuzzyStego against these diverse methodologies, its scientific contributions, including intensity-sensitive embedding and fuzzy-based classification, validated in capacity and imperceptibility enhancements and computational efficiency evidenced by the obtained experimental results.

4.3.2 Evaluating the Trade-off between Payload Capacity and Image Quality

Fig. 5 compares the proposed FuzzyStego method with the existing methods, showing the PSNR values achieved for the Boat and Baboon, the commonly used cover images for all the techniques involved in the comparison. Based on the figure data, the proposed FuzzyStego demonstrates strong performance, achieving PSNR values of 52.43 dB for the Boat image and 50.72 dB for the Baboon image. These values are significantly higher than those obtained in the state-of-the-art methods, such as those from [7,26], which fall below 42 dB for both images. This highlights FuzzyStego's superior ability to preserve image quality after embedding. When compared with methods from [8,10,27], FuzzyStego remains competitive. For the Boat image, FuzzyStego's PSNR (52.43 dB) outperforms that of [8] (46.30 dB), demonstrating its ability to embed data with less visual distortion. While the methods from [10,27] achieve higher PSNR values (58.70 dB and 59.11 dB, respectively), FuzzyStego still strikes a commendable balance between image quality and embedding efficiency. FuzzyStego outperforms the methods from [7,8,26] for the Baboon image, all showing PSNR values below 48 dB, indicating more noticeable image degradation. In contrast, FuzzyStego maintains a PSNR of 50.72 dB, preserving better visual quality. While the methods [10,27] achieve slightly higher PSNR values (54.27 and 55.45 dB, respectively), FuzzyStego's results outperform.

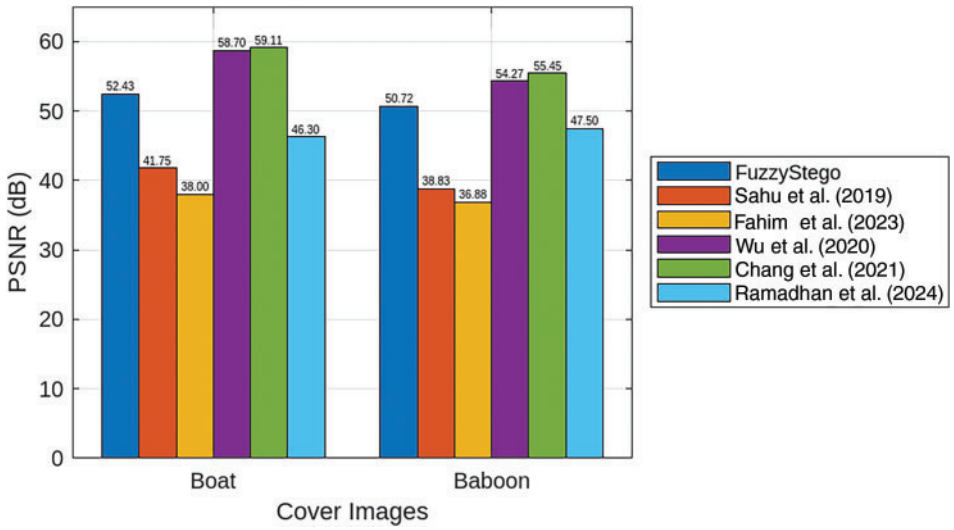

Figure 5: PSNR Comparison between the FuzzyStego and the existing algorithms [7,8,10,26,27]

4.3.3 Evaluating the Maximum Payload Capacity

Fig. 6 highlights the significant advantages of the proposed FuzzyStego method over state-of-the-art algorithms by showcasing the maximum embedding capacities achieved for the Boat and Baboon cover images. FuzzyStego consistently demonstrates superior performance, achieving the highest embedding capacity in all cases. This efficiency supports FuzzyStego's suitability for applications requiring high-capacity steganography.

FuzzyStego achieves an embedding capacity of 1,079,052 bits for the Boat cover image, far exceeding the capacities of other methods. The method in [7], while ranking second with 824,789 bits, still falls short by approximately 23.6%. The methods in [10,26] exhibit significantly lower capacities of 34,059 bytes and 499,992 bytes, representing reductions of over 96.8% and 53.7%, respectively, compared to FuzzyStego. These results emphasize the substantial performance gap between FuzzyStego and the alternatives. Similarly, FuzzyStego achieves the highest embedding capacity for the Baboon cover image, reaching 1,078,190 bytes. The method in [7] follows with 793,183 bits, approximately 26.5% lower. The methods in [10,26] perform poorly, with capacities of 17,582 and 499,995 bits, reflecting reductions of 98.4% and 53.7%, respectively.

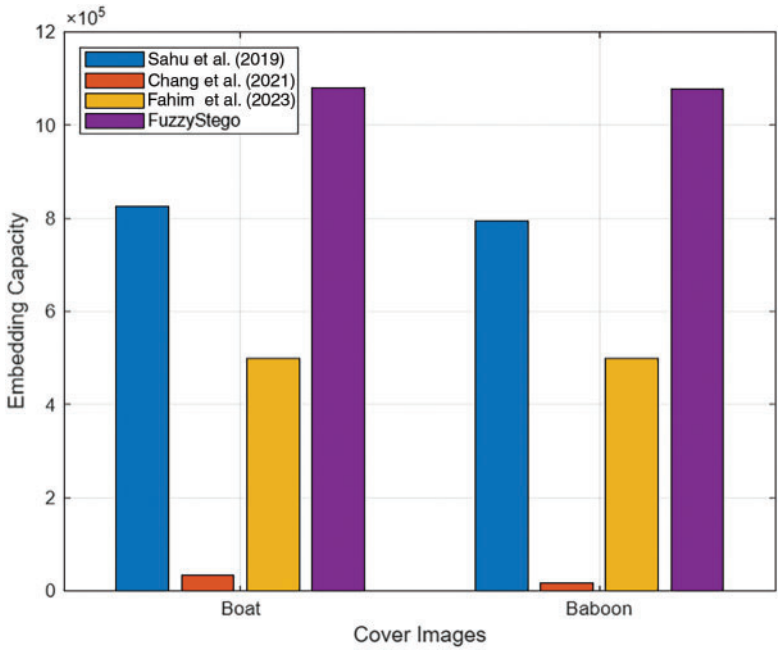

Figure 6: Embedding capacity comparison between the FuzzyStego and the existing methods [7,10,26]

4.3.4 Quantitative Evidence Highlighting the Superiority of FuzzyStegoover Existing Techniques

Fig. 7 provides a comparative analysis highlighting the quantitative superiority of FuzzyStego over existing steganographic techniques, namely the PVD-based method in [34], the LSB-based method in [35], and the Discrete Cosine Transform (DCT)-based method in [36]. This evaluation uses PSNR as the metric, with higher PSNR values indicating better imperceptibility and image quality after embedding. For the Peppers image, FuzzyStego demonstrates a PSNR of 50.07 dB, surpassing the PVD-based method in [34] (41.55 dB) and the LSB-based method in [35] (34.24 dB). Although the DCT-based method in [36] reports a slightly higher PSNR of 45.09 dB, FuzzyStego effectively balances imperceptibility and embedding robustness, offering a practical and adaptable steganographic solution.

These results illustrate FuzzyStego's robust performance across different images and steganographic paradigms. By integrating a fuzzy logic-based adaptive approach, FuzzyStego surpasses the PVD and LSB techniques and provides comparable results to the DCT-based method, emphasizing its versatility and effectiveness in secure data embedding.

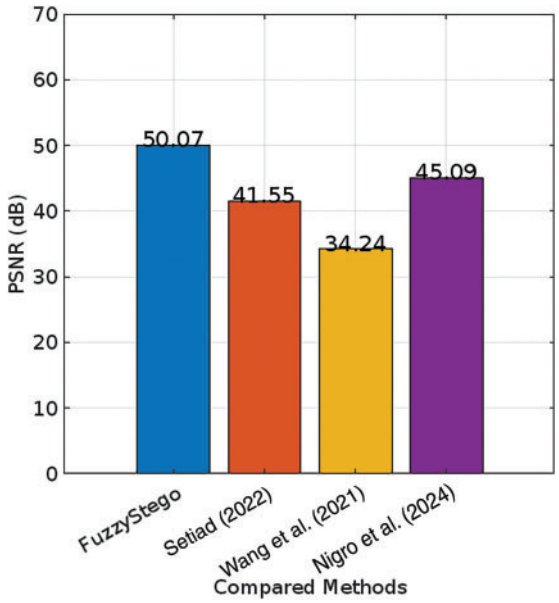

Figure 7: Quantitative comparisons of the FuzzyStego and the existing methods under a same cover image, Peppers [35–37]

4.4 Ablation Study for a Comparative Analysis of FuzzyStego and the NoFuzzy Methods

This study's ablation experiments compare the proposed FuzzyStego with the LSB method without fuzzy logic. Fig. 8 illustrates the comparative performance of the fully proposed FuzzyStego against its version without fuzzy logic to adaptively select the embeddable pixels (labeled as "NoFuzzy"). Fig. 8a presents the PSNR results as a function of the payload size in kb, and Fig. 8b includes the SSIM as a function of payload size under two general purposes and medical images. These metrics, critical for evaluating the quality and integrity of stego images, highlight the FuzzyStego method's superiority over the NoFuzzy in preserving image quality and structure.

Fig. 8a identifies that the PSNR results show the consistent advantage of FuzzyStego over the LSB method. At low payload sizes (≤20 kb), FuzzyStego maintains high PSNR values, exceeding 65 dB for all test images. For example, the "Aerial" and "Airplane" images achieve PSNR values of around 70 dB, whereas the LSB method drops below 60 dB, indicating noticeable degradation. As the payload increases to medium levels (20–60 kb), FuzzyStego gradually reduces PSNR but remains above 50 dB, highlighting its scalability. On the other hand, the NoFuzzy method presents a steeper drop, with PSNR values often falling below 40 dB for images like "Brain" and "Hand." At higher payload sizes (≥80 kb), FuzzyStego continues to demonstrate robust performance, sustaining PSNR values above 40 dB, while the NoFuzzy approach shows low quality stego images, with PSNR values dropping below 30 dB. This resilience demonstrates FuzzyStego's ability to handle large data payloads while preserving image quality, a critical requirement for real-world applications.

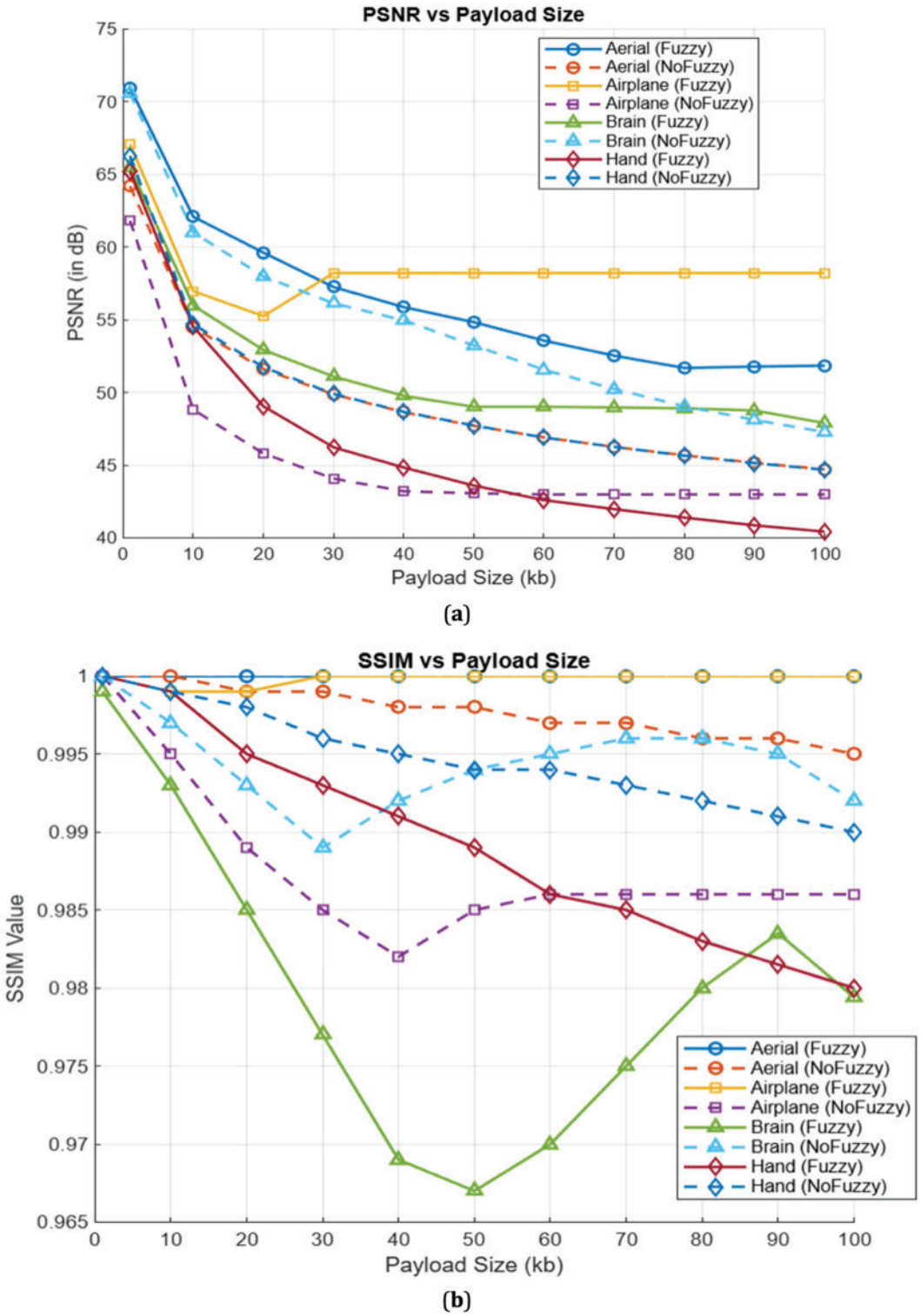

Figure 8: Ablation experiments results. **(a)** PSNR results; **(b)** SSIM results

Moreover, Fig. 8b illustrates that the SSIM results further validate FuzzyStego's superior performance in maintaining the structural similarity between the cover and stego images. FuzzyStego achieves SSIM values consistently above 0.98, reflecting its strong structural fidelity compared to the NoFuzzy. While the SSIM values for FuzzyStego show a minor decrease with increasing payload, they remain within acceptable limits for practical use. In contrast, the NoFuzzy approach suffers a significant reduction in SSIM, especially at larger payloads. Considering the specific images, it is noticed that the "Brain" image's SSIM drops below 0.97 at payloads of 50 kb and beyond, indicating a marked loss in structural integrity. FuzzyStego demonstrates remarkable consistency, with images like "Aerial" and "Airplane" showing near-flat SSIM curves, even at large payload sizes.

On the other hand, the NoFuzzy method's SSIM curves exhibit steep downward trends, further underscoring its limitations in preserving image structure. The NoFuzzy method's simplistic embedding strategy results in significant quality degradation and structural distortion as payload sizes increase. FuzzyStego's intelligent embedding mechanism ensures robust scalability, maintaining high image quality and structural integrity levels, even at large payloads.

4.5 Discussions

The findings of this study highlight the innovative and robust capabilities of the FuzzyStego method, particularly in balancing image quality and embedding efficiency across diverse scenarios. The evaluation of image quality metrics such as PSNR and SSIM in Table 1 reveals significant insights. Images like Aerial and Fishing Boat for smaller payloads demonstrate excellent PSNR values, indicating superior quality preservation. Images like Airplane exhibit remarkable resilience to quality degradation at larger payloads, suggesting an inherent adaptability in their texture profiles for data embedding. The structural integrity analysis using SSIM demonstrates FuzzyStego's strength in preserving essential image features even at high payload capacities. Images like Aerial and Pixel Ruler maintain near-perfect SSIM values, signifying minimal perceptual quality loss. Although images such as Car and APCs exhibit slightly reduced SSIM values, the decrease is negligible, confirming FuzzyStego's capability to embed data securely while retaining structural integrity. This robust preservation of structural features ensures that the stego images are nearly indistinguishable from their original counterparts, reinforcing the method's suitability for high-quality applications.

The cross-dataset analysis reaffirms FuzzyStego's versatility in handling medical and general-purpose images. In medical applications, where maintaining structural integrity is paramount, FuzzyStego achieves exceptional PSNR and SSIM values, even for larger payloads. General-purpose images like Fountain also exhibit robust performance, maintaining SSIM values consistently above 0.97 across payload sizes, demonstrating FuzzyStego's ability to deliver superior quality across diverse use cases. Furthermore, the computational efficiency of FuzzyStego, evaluated through embedding and extraction time, further highlights its practicality. Embedding times increase predictably with payload size, yet extraction times remain stable, showcasing the method's scalability and reliability to highlight FuzzyStego's efficiency in real-time applications.

Moreover, based on the current performance of FuzzyStego in spatial images, it can be effectively extended to audio and video steganography by adapting its principles to the unique characteristics of these media types. For audio steganography, the method may use the amplitude values of audio signals, classifying them into intensity levels like pixel intensities in images. Audio signals, represented as normalized waveforms (e.g., [−1, 1]) or digital formats (e.g., [0,255]), benefiting the features of fuzzy logic, can be categorized into L, ML, M, MH, and H amplitude levels. The fuzzy rules can guide the embedding process by determining the number of bits to embed based on the amplitude level. Silent or quiet portions can tolerate higher embedding, while louder or peak amplitude sections, more perceptually sensitive, allow for minimal embedding [37].

This adaptive embedding strategy may be a promising solution to be worked on in future works in audio steganography to maximize the data-hiding capacity, making it suitable for applications such as secure communication in voice recordings and music tracks.

Furthermore, the FuzzyStego approach, which integrates spatial and temporal adaptations to optimize data embedding, can also be recommended for video steganography. Utilizing the fuzzy logic paradigm proposed in FuzzyStego, a video frame can be treated as an image, with pixels classified into intensity levels using the Mamdani FIS and predefined thresholds. The temporal redundancy between consecutive frames may be analyzed to introduce motion-based fuzzy rules. Low-motion regions, such as static backgrounds, can accommodate higher embedding capacity by modifying multiple least significant bits, whereas high-motion areas, involving moving objects or rapid transitions, are modified minimally to preserve visual fidelity. By combining spatial and temporal fuzzy logic adaptations, the FuzzyStego method can achieve promising relatively high embedding capacity and imperceptibility in video files, making it a robust solution for secure video communication.

Conclusively, this article demonstrates the effectiveness of FuzzyStego as a superior steganographic method compared to existing approaches. FuzzyStego achieves an optimal balance between embedding capacity, imperceptibility, and computational efficiency using fuzzy logic for detailed pixel classification. Its ability to maintain high visual quality in stego images while securely embedding data highlights its reliability and adaptability. This approach establishes FuzzyStego as a robust and versatile solution for modern steganographic applications, meeting the growing need for secure and efficient data-hiding techniques.

5 Conclusion

This article introduces FuzzyStego, a new technique developed to enhance stego image quality by reducing the impact of payload size. Traditional methods, such as PVD, Transform Domain Techniques, and LSB insertion, often encounter limitations like image degradation, susceptibility to processing attacks, and constrained embedding capacity. The proposed method addresses these challenges by applying fuzzy logic to categorize pixels into five intensity levels: L, ML, M, MH, and H. This pixel classification enables adaptive data embedding, with the number of hidden bits adjusted based on intensity levels to minimize detectability. The results demonstrate that the proposed FuzzyStego achieves a promising PSNR and SSIM, which reaches a maximum of 1, showing its effectiveness in maintaining high image quality while embedding data.

To expand the applicability of the proposed FuzzyStego method, currently focused on image steganography, future research could investigate its performance in other domains, such as audio and video steganography. While this study primarily targets image-based applications, adapting the method to these multimedia formats could reveal its broader versatility and effectiveness. Audio and video files present unique challenges, such as larger sizes and more intricate structures, but FuzzyStego's adaptability suggests it can maintain high levels of security and efficiency across these domains. Given its demonstrated scalability and robustness, FuzzyStego has the potential to advance the field of steganography, contributing to real-world applications like secure communication and digital content protection.

Acknowledgement: The authors thank the NCC laboratory staff and the Information Security Research Group members at Institut Teknologi Sepuluh Nopember for their technical support and valuable discussions throughout this study.

Funding Statement: This study was funded by Institut Teknologi Sepuluh Nopember.

Author Contributions: The authors confirm their contribution to the paper as follows: conceptualization, Mardhatillah Shevy Ananti, Adifa Widyadhani Chanda D'Layla, Ntivuguruzwa Jean De La Croix and Tohari Ahmad; methodology,

Mardhatillah Shevy Ananti, Adifa Widyadhani Chanda D'Layla, Ntivuguruzwa Jean De La Croix and Tohari Ahmad; software development, Mardhatillah Shevy Ananti, Adifa Widyadhani Chanda D'Layla and Ntivuguruzwa Jean De La Croix; formal analysis, Mardhatillah Shevy Ananti, Adifa Widyadhani Chanda D'Layla and Ntivuguruzwa Jean De La Croix; original draft writing, and visualization, Mardhatillah Shevy Ananti, Adifa Widyadhani Chanda D'Layla and Ntivuguruzwa Jean De La Croix; review and editing of the manuscript, Adifa Widyadhani Chanda D'Layla, Ntivuguruzwa Jean De La Croix and Tohari Ahmad; supervision, Tohari Ahmad; project administration, Tohari Ahmad; acquisition of funding, Tohari Ahmad. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data is available on request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Glossary

APCs	Armored Personel Carriers
C	Mathematical Expression of FuzzyStego
$C(i, j)$	Cover Image
dB	Decibel
DCT	Discrete Cosine Transform
EC	Embedding Capacity
Eq.	Equation
FIS	Fuzzy Inference System
GA	Genetic Algorithm
GA-IPVD	Genetic Algorithm-Enhanced Pixel Value Differencing
H	High Intensity
Ip	Image Pixel
kb	Kilobit
L	Low Intensity
LSB	Least Significant Bit
$LSB_k(p(i, j))$	K^{th} LSB of the pixel at $p(i, j)$
M	Medium Intensity
MH	Medium High Intensity
ML	Medium Low Intensity
PSNR	Peak Signal-to-Noise Ratio
PVD	Pixel Value Differencing
$p(i, j)$	Pixel at position (*i,j*) in the cover image
$p'(i, j)$	Pixel at position (*i,j*) in the stego image
S	Secret data
SIPI	Signal and Image Processing Institute
SSIM	Structural Similarity Index Measure
$S(i, j)$	Stego Image
tL	Low Threshold
tM	Medium Threshold
tH	High Threshold
$1N_{med}$	One pixel count for the M range
$1N_{med-High}$	One pixel count for the MH range
$2N_{med-Low}$	Two pixel counts for the ML range
$3N_{Low}$	Three pixel counts for the L range

References

1. Zhang X, Li C, Tian L. Advanced audio coding steganography algorithm with distortion minimization model based on audio beat. Comput Electr Eng. 2023;106(6):108580. doi:10.1016/j.compeleceng.2023.108580.
2. Debnath S, Mohapatra RK, Dash R. Secret data sharing through coverless video steganography based on bit plane segmentation. J Inf Secur Appl. 2023;78(4):103612. doi:10.1016/j.jisa.2023.103612.
3. Song B, Wei P, Wu S, Lin Y, Zhou W. A survey on deep-learning-based image steganography. Expert Syst Appl. 2024;254(7):124390. doi:10.1016/j.eswa.2024.124390.
4. Hu K, Wang M, Ma X, Chen J, Wang X, Wang X. Learning-based image steganography and watermarking: a survey. Expert Syst Appl. 2024;249(2s):123715. doi:10.1016/j.eswa.2024.123715.
5. Rahman S, Uddin J, Zakarya M, Hussain H, Khan AA, Ahmed A, et al. A comprehensive study of digital image steganographic techniques. IEEE Access. 2023;11(5):6770–91. doi:10.1109/access.2023.3237393.
6. Mandal PC, Mukherjee I, Paul G, Chatterji BN. Digital image steganography: a literature survey. Inf Sci. 2022;609(12):1451–88. doi:10.1016/j.ins.2022.07.120.
7. Sahu AK, Swain G. An optimal information hiding approach based on pixel value differencing and modulus function. Wirel Pers Commun. 2019;108(1):159–74. doi:10.1007/s11277-019-06393-z.
8. Wu H, Li X, Zhao Y, Ni R. Improved PPVO-based high-fidelity reversible data hiding. Signal Process. 2020;167(8):107264. doi:10.1016/j.sigpro.2019.107264.
9. Weng S, Shi Y, Hong W, Yao Y. Dynamic improved pixel value ordering reversible data hiding. Inf Sci. 2019;489(8):136–54. doi:10.1016/j.ins.2019.03.032.
10. Chang J, Ding F, Li X, Zhu G. Hybrid prediction-based pixel-value-ordering method for reversible data hiding. J Vis Commun Image Represent. 2021;77(6):103097. doi:10.1016/j.jvcir.2021.103097.
11. Khan M, Rasheed A. A high-capacity and robust steganography algorithm for quantum images. Chin J Phys. 2023;85(6):89–103. doi:10.1016/j.cjph.2023.06.016.
12. Li Q, Yan B, Li H, Chen N. Separable reversible data hiding in encrypted images with improved security and capacity. Multimed Tools Appl. 2018;77(23):30749–68. doi:10.1007/s11042-018-6187-y.
13. De La Croix NJ, Ahmad T. Toward secret data location via fuzzy logic and convolutional neural network. Egypt Inform J. 2023;24(3):100385. doi:10.1016/j.eij.2023.05.010.
14. Reinel TS, Brayan AH, Alejandro BM, Alejandro MR, Daniel AG, Alejandro AJ, et al. GBRAS-net: a convolutional neural network architecture for spatial image steganalysis. IEEE Access. 2021;9:14340–50. doi:10.1109/access.2021.3052494.
15. De La Croix Ntivuguruzwa J, Ahmad T. A convolutional neural network to detect possible hidden data in spatial domain images. Cybersecurity. 2023;6(1):23. doi:10.1186/s42400-023-00156-x.
16. Mohammed HA, Saffar NFH. LSB based image steganography using McEliece cryptosystem. Mater Today Proc. 2021;40(1):271. doi:10.1016/j.matpr.2021.07.182.
17. Ali UAME, Ali E, Sohrawordi M, Sultan MN. A LSB based image steganography using random pixel and bit selection for high payload. Int J Math Sci Comput. 2021;7(3):24–31. doi:10.5815/ijmsc.2021.03.03.
18. Hu X, Fu Z, Zhang X, Chen Y. Invisible and steganalysis-resistant deep image hiding based on one-way adversarial invertible networks. IEEE Trans Circuits Syst Video Technol. 2024;34(7):6128–43. doi:10.1109/TCSVT.2023.3348291.
19. Siddiqui GF, Iqbal MZ, Saleem K, Saeed Z, Ahmed A, Hameed IA, et al. A dynamic three-bit image steganography algorithm for medical and e-healthcare systems. IEEE Access. 2020;8:181893–903. doi:10.1109/access.2020.3028315.
20. Zaini AFR, De La Croix NJ, Ahmad T. A steganographic approach based on pixel blocks differencing to enhance the quality of the stego image. In: Conference on Information Communications Technology and Society (ICTAS); 2024 Mar 7–8; 2024; Durban, South Africa. p. 63–8. doi:10.1109/ICTAS59620.2024.10507107.
21. Cao F, An B, Yao H, Tang Z. Local complexity based adaptive embedding mechanism for reversible data hiding in digital images. Multimed Tools Appl. 2019;78(7):7911–26. doi:10.1007/s11042-018-6031-4.
22. Chen CC, Chang CC, Chen K. High-capacity reversible data hiding in encrypted image based on Huffman coding and differences of high nibbles of pixels. J Vis Commun Image Represent. 2021;76(8):103060. doi:10.1016/j.jvcir.2021.103060.

23. Bai X, Chen Y, Duan G, Feng C, Zhang W. A data hiding scheme based on the difference of image interpolation algorithms. J Inf Secur Appl. 2022;65:103068. doi:10.1016/j.jisa.2021.103068.
24. Ding W, Zhang H, Reulke R, Wang Y. Reversible image data hiding based on scalable difference expansion. Pattern Recognit Lett. 2022;159(4):116–24. doi:10.1016/j.patrec.2022.05.014.
25. Tian J. Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol. 2003;13(8):890–6. doi:10.1109/TCSVT.2003.815962.
26. Fahim A, Raslan Y. Optimized steganography techniques based on PVDS and genetic algorithm. Alex Eng J. 2023;85(3):245–60. doi:10.1016/j.aej.2023.11.013.
27. Ramadhan IF, Anandha RDA, D'Layla AWC, De La Croix NJ, Ahmad T. Image steganography using customized differences between the neighboring pixels. In: 2024 7th International Conference on Informatics and Computational Sciences (ICICoS); 2024 Jul 17–18; Semarang, Indonesia. p. 496–501. doi:10.1109/ICICoS62600.2024.10636936.
28. Signal and Image Processing Institute (USC). Volume 3: miscellaneous. [cited 2024 May 1]. Available from: https://sipi.usc.edu/database/database.php?volume=misc.
29. Lorem Ipsum. The standard lorem ipsum passage. [cited 2024 May 1]. Available from: https://www.lipsum.com/.
30. DICOM Library. Free online medical knowledge exchange portal—DICOM Library. [cited 2024 May 1]. Available from: https://www.dicomlibrary.com/.
31. Scott Mader K. CT medical images. [cited 2024 May 1]. Available from: https://www.kaggle.com/datasets/kmader/siim-medical-images.
32. Bas P, Filler T, Pevný T. Break our steganographic system: the ins and outs of organizing BOSS. In: Filler T, Pevný T, Craver S, Ker A, editors. Information hiding. Berlin/Heidelberg: Springer; 2011. p. 59–70. doi: 10.1007/978-3-642-24178-9_5.
33. Croix NJDL, Ahmad T, Han F. Enhancing secret data detection using convolutional neural networks with fuzzy edge detection. IEEE Access. 2023;11(1):131001–16. doi:10.1109/access.2023.3334650.
34. Wu DC, Shih ZN. Image steganography by pixel-value differencing using general quantization ranges. Comput Model Eng Sci. 2024;141(1):353–83. doi:10.32604/cmes.2024.050813.
35. Setiadi DRIM. Improved payload capacity in LSB image steganography uses dilated hybrid edge detection. J King Saud Univ Comput Inf Sci. 2022;34(2):104–14. doi:10.1016/j.jksuci.2019.12.007.
36. Wang X, Liu C, Jiang D. A novel triple-image encryption and hiding algorithm based on chaos, compressive sensing and 3D DCT. Inf Sci. 2021;574(1):505–27. doi:10.1016/j.ins.2021.06.032.
37. Nigro M, Krishnan S. Trends in audio scene source counting and analysis. Mach Learn Appl. 2024;18(12):100593. doi:10.1016/j.mlwa.2024.100593.