

A Novel Steganography Algorithm based on Interval Type-1 Fuzzy Logic System

Vinita Yadav, Meenakshi Hooda*, Navita Dhaka

Department of Mathematics, Maharshi Dayanand University, Rohtak, India

Abstract In this paper, a novel image steganography algorithm *mshEdgeGrayFTI* is proposed, which combines the mamdani fuzzy inference system (FIS) and the least significant bit (LSB) technique for grayscale images. The proposed algorithm begins by removing the noise from the image using a Gaussian filter and then masking the image. The differences between the pixel intensities of two consecutive columns ($\text{diffColumn}_{i,j}$) and rows ($\text{diffRow}_{i,j}$) of the resultant image are taken as input variables. A Mamdani FIS determines the edge pixels where the confidential data can be hidden. The pixels are categorized into black, gray, and white. The LSB technique is then used to hide the confidential data in the identified edge pixels. Based on various evaluation metrics, the experimental results demonstrate that the proposed algorithm *mshEdgeGrayFTI* outperforms other methods.

Keywords Image steganography, Mamdani FIS, Fuzzy logic type-1, LSB.

DOI: 10.19139/soic-2310-5070-2650

1. Introduction

Over the years, information security has become one of the most crucial components of communication and information technology due to rapid developments in computers and technology. Therefore, we must take action to preserve confidential information. Confidential information can be protected using steganography or cryptography [1]. Although both technologies aim to protect confidential data, they have distinct concepts. Cryptography encodes the secret message so that it cannot be deciphered, while steganography hides the existence of confidential information. Steganography is a secret communication technique that hides secret information within trustworthy media such as images, text, audio, and video so that attackers cannot extract the information [2]. Image steganography is a method that is used to conceal secret data within an image, providing a secure and safe way to exchange information on the Internet. The ‘cover image’ refers to the image used to embed the secret data; the ‘payload’ refers to the embedded data; and the ‘stego image’ refers to the image that contains the embedded data. In steganography, several important aspects are considered, such as capacity, robustness, and security. Capacity refers to the number of secret message bits that can be concealed within the original image without degrading its authenticity, confidentiality, or integrity. It is measured in bits per pixel. Robustness indicates the extent to which the stego image can withstand modifications before the secret information is destroyed by an attacker. Security refers to the difficulty an attacker faces in detecting the presence of hidden information. The selection of embedding pixels determines the security of any steganography method, as pixels in noisy and textured regions are more challenging to model [3]. Image steganography can be categorized into the spatial and the transform (frequency) domains. In spatial domain steganography, the secret message is directly inserted into the cover image, whereas the transform

* Correspondence to: Meenakshi Hooda (Email: meenakshi.maths@mdurohtak.ac.in). Department of Mathematics, Maharshi Dayanand University, Rohtak 124001 (India).

domain technique modifies the transform domain coefficients obtained from the cover image [4]. LSB substitution is one of the spatial domain steganography approaches. LSB substitution is a traditional steganography technique known for its efficiency, as it offers low computational overhead and a high embedding capacity. This method involves modifying the LSBs of the pixels in the cover image to encode the secret message. On the receiving end, the recipient retrieves the hidden message by extracting the LSBs of the stego image [5]. The objective of this study is to propose a new steganography algorithm based on fuzzy logic. The proposed steganography algorithm, *mshEdgeGrayFTI*, provides high imperceptibility and security. Mamdani FIS is used to identify the edge pixels, while the LSB method is used for hiding and retrieving confidential data. The rest of the paper is organized as follows: section II discusses the research methodology. Section III presents the literature review related to this study. Section IV gives details the proposed algorithm *mshEdgeGrayFTI*. Section V illustrates the experimental analysis of *mshEdgeGrayFTI*. Finally, Section VI provides the conclusion of the paper.

2. Research Methodology

In this section, the basic concepts of Mamdani FIS and LSB substitution used in the algorithm *mshEdgeGrayFTI* are discussed.

2.1. Mamdani fuzzy inference system (FIS)

Fuzzy logic was first introduced by Lotfi Zadeh in 1965 [6]. In fuzzy logic, an FIS is a fundamental system that involves the maps of inputs to outputs using fuzzy sets and rules. An FIS mainly has three stages:

1. Fuzzification: Converts crisp values to fuzzy values.
2. Fuzzy inference engine: Evaluates the degree of match for the current fuzzy input and rule, determining which rules are relevant based on the input.
3. Defuzzification: Converts fuzzy values back to crisp values [7].

FIS has two types of models: Sugeno and Mamdani FIS, which include fuzzy logic operators, if-then rules, and membership functions. In this paper, we used the Mamdani FIS. In 1975, Ebrahim Mamdani introduced the first control system (Mamdani method) to regulate a combination of steam engines and fuzzy set theory Mamdani [8]. The Mamdani model utilizes fuzzy sets to represent inputs and outputs specified by membership functions. Figure 1 represents the basic structure of the Mamdani system [9].

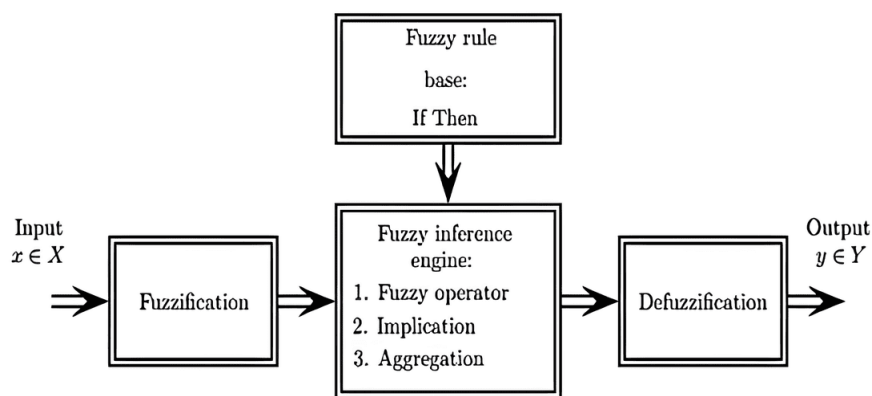


Figure 1. Mamdani system model

Here, $x \in X$ is a linguistic input variable, and $y \in Y$ is a linguistic output variable. During the fuzzification process,

an input value is first transformed into a membership value corresponding to each rule. The fuzzy operators are then applied to each rule, followed by the implication process. After that, all output values are aggregated according to each fuzzy rule. Finally, the crisp output value is determined by applying the defuzzification method [10].

2.2. Least significant bit (LSB)

LSB is the most widely used technique in image steganography. In this technique, the LSB of each pixel in the cover image is replaced with a bit of secret data. Many LSBs can be used to hide more bits of secret data. This technique is widely used due to its high hiding capacity and low computational complexity [11]. For example, if the secret bits '11' are to be hidden in a cover image pixel with binary value $(10011000)_2$, using two LSBs, then after embedding the secret bits, the stego image pixel will be $(10011011)_2$.

3. Literature Review

Chen et al. [12] introduced a steganography technique with high embedding capacity by utilizing different edge detection methods. They used a fuzzy complement and canny edge detector to extract edge segments from the original cover image. Specifically, the LSBs of both edge and non-edge pixels are used to hide different numbers of secret message bits. Hore et al. [13] compared measurement tools used to assess the quality of stego images, such as peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). Based on their findings, the SSIM measurement tool is recommended for all steganography images. Khodaei et al. [14] proposed a novel approach to conceal more secret bits in cover images by combining LSB and pixel value difference (PVD). Tseng et al. [15] proposed a block-based fuzzy edge detector technique that generates a greater number of edge pixels. Instead of using two parameters, they used only one variable for LSB replacement. Islam et al. [16] illustrated an edge-based strategy in which both strong and weak edges are utilized to embed the maximum number of secret message bits. Bai et al. [17] proposed a unique steganographic method using a different edge detector. In this scheme, the edge region is produced by replacing the last five LSBs of the cover image, while the three most significant bits (MSBs) of all pixels remain unaltered. This approach creates more hiding space compared to previous schemes. Santhi et al. [18] proposed a unique method for medical images, combining XOR and edge-based hiding techniques. A novel key is generated to enhance the security of the secret information using a sudoku puzzle design. Cheng et al. [19] illustrated a novel steganography approach to identify the edge region by using the MSBs. Alghamdi [20] proposed a new fuzzy logic-based approach to embed secret data, where pixel merging is done using four fuzzy rules. Dhargupta et al. [21] proposed a fuzzy type edge-finding approach to conceal secret information in edge areas, as variations in edge regions are more difficult to detect compared to smooth regions. The distance of an edge pixel to the nearest edge pixel determines the capacity of embedded bits. This technique has improved the PSNR value and quality of the stego image. Faydi et al. [22] proposed a novel method which compresses secret data using Lempel- ZIV-Welch (LZW) and embeds the compressed bits into the LSBs of edge pixels in the cover image, ensuring the pixels remain unchanged. A location address file records the pixel positions for retrieval. The method achieves a PSNR of infinity, SSIM and normalized cross correlation (NCC) of one and mean square error (MSE) and average difference (AD) of zero, ensuring perfect imperceptibility of the stego image. Feng et al. [23] proposed this study to enhance low-resolution digital images by first applying bilateral filtering for preprocessing. A rational function model with universal attributes is then developed for coordinate-free image reconstruction. Image features are extracted using generative adversarial network (GAN), SIFT and difference of gaussians (DOG), providing a foundation for reconstruction. The final enhancement is achieved through feature matching, bilateral regularization, pixel correction, wavelet transform and edge-adaptive processing. Ashraf et al. [24] proposes a steganographic approach called IT2FLS-LSB, which uses an interval type-2 fuzzy logic system to assess pixel similarity based on human perception for LSB embedding. It also proposes two additional methods—T1FLS-LSB and SM-LSB using Euclidean distance. The effectiveness of all methods is evaluated using PSNR, UQI, and SSIM on various images.

4. Proposed Work

This section presents a step-by-step explanation of the data embedding and extraction phases of the proposed algorithm *mshEdgeGrayFTI* for grayscale images. Mamdani FIS is employed to identify edge pixels in a grayscale image. The algorithm *mshEdgeGrayFTI* ensures accurate edge localization in the cover and stego image, ensuring that the secret data can be accurately embedded and extracted. Algorithm 1 represents the embedding phase of the proposed algorithm *mshEdgeGrayFTI*. This algorithm also enhances the security of the stego image by utilizing edge pixels for the data embedding phase.

Algorithm 1 Embedding phase of algorithm *mshEdgeGrayFTI*

- 1: Read the image and convert it to a grayscale image, if necessary.
 - 2: Convolve it with a Gaussian filter to remove the noise from the image and mask each pixel of the image.
 - 3: Take $\text{diffColumn}_{i,j}$ and $\text{diffRow}_{i,j}$ as input variables and define their ranges to initialize the mamdani fuzzy inference system (FIS).
 - 4: Add membership functions “low” and “high” to the FIS using a triangular membership function (trimf). Initialize its parameters as the vector [a b c], where a and c define the feet, and b defines the peak of the membership function for each input variable, i.e., $\text{diffColumn}_{i,j}$ and $\text{diffRow}_{i,j}$. A membership function associated with a given fuzzy set maps an input value to its appropriate membership value.
 - 5: Add an output variable and define its membership functions by initializing parameters for the white, gray, and black fuzzy sets.
 - 6: Define fuzzy rules and implement the implication for each fuzzy rule.
 - 7: Evaluate the rules to return intermediate inference results in the form of bounded regions and calculate the centroid for each corresponding bounded region.
 - 8: Aggregate the output fuzzy set corresponding to each rule into a single fuzzy set and defuzzify the aggregated output using the center of sums (CoS) method.
 - 9: Take the secret message and convert it into binary form. Hide the secret message bits into the edge pixels of the cover image by using the k LSB method.
-

4.1. Data embedding phase

The detailed steps of the data embedding phase are described below:

Step 1: Select an image as the cover image (C). If the cover image is an RGB image, transform it into a grayscale image (G). G is a matrix of dimensions $P \times Q$, and each element $G_{i,j}$ represents the intensity of the pixels in the i^{th} row and j^{th} column of G, where $1 \leq i \leq P$ and $1 \leq j \leq Q$. The range of pixel intensity values is [0-255].

Step 2: Apply the Gaussian filter method to remove noise and mask the bits of each pixel in G using (1). This results obtain the masked image G' , whose elements are represented by $G'_{i,j}$.

$$G'_{i,j} = \text{int16} \left(\frac{G_{i,j}}{4} \right) \times 4 \quad (1)$$

Step 3: Compute the intensity differences between pixels in two consecutive columns ($\text{diffColumn}_{i,j}$) and two consecutive rows ($\text{diffRow}_{i,j}$) of the masked image G' , using (2) and (3), respectively.

$$\text{diffColumn}_{i,j} = \left| G'_{i,j} - G'_{i+1,j} \right| \quad (2)$$

where $1 \leq i \leq P-1$, $1 \leq j \leq Q$, and row-wise difference is $\{j, i\}$.

$$\text{diffRow}_{i,j} = \left| G'_{i,j} - G'_{i,j+1} \right| \quad (3)$$

where $1 \leq i \leq P$, $1 \leq j \leq Q-1$, and column-wise difference is $\{i, j\}$.

Step 4: Add “low” and “high” membership functions using triangular-shaped membership functions (trimf) to determine whether the difference between the intensity values of the pixels of two consecutive columns and rows is trivial or significant. These membership functions take the input values $\text{diffColumn}_{i,j}$ and $\text{diffRow}_{i,j}$ and compute fuzzy membership values for “low” and “high” within the range [0–1]. A higher output value indicates a greater degree of belonging.

- $\mu_{\text{low}}(\text{diffColumn}_{i,j})$ is the degree of membership of $\text{diffColumn}_{i,j}$ in the fuzzy set “low”, with trimf vector parameters [a, b, c] set to [0, 0, 255].
- $\mu_{\text{high}}(\text{diffColumn}_{i,j})$ is the degree of membership of $\text{diffColumn}_{i,j}$ in a fuzzy set “high”, with trimf vector parameters [a, b, c] set to [0, 255, 255].
- $\mu_{\text{low}}(\text{diffRow}_{i,j})$ is the degree of membership of $\text{diffRow}_{i,j}$ in the fuzzy set “low”, with trimf vector parameters [a, b, c] set to [0, 0, 255].
- $\mu_{\text{high}}(\text{diffRow}_{i,j})$ is the degree of membership of $\text{diffRow}_{i,j}$ in the fuzzy set “high”, with trimf vector parameters [a, b, c] set to [0, 255, 255].

Step 5: Next, the three fuzzy sets “black”, “gray”, and “white” are associated with the output variable $\text{fuzzyEdgeOut}_{i,j}$ using triangular-shaped membership functions (trimf).

- $\mu_{\text{black}}(\text{fuzzyEdgeOut}_{i,j})$ represents the degree of membership of $\text{fuzzyEdgeOut}_{i,j}$ in the fuzzy set “black”, with trimf vector parameters [a, b, c] set to [0, 0, 0.3].
- $\mu_{\text{gray}}(\text{fuzzyEdgeOut}_{i,j})$ represents the degree of membership of $\text{fuzzyEdgeOut}_{i,j}$ in the fuzzy set “gray”, with trimf vector parameters [a, b, c] set to [0.3, 0.3, 0.8].
- $\mu_{\text{white}}(\text{fuzzyEdgeOut}_{i,j})$ represents the degree of membership of $\text{fuzzyEdgeOut}_{i,j}$ in the fuzzy set “white”, with trimf vector parameters [a, b, c] set to [0.8, 1, 1].

The membership function shown in Figure 2 is a type of triangular membership function [25].

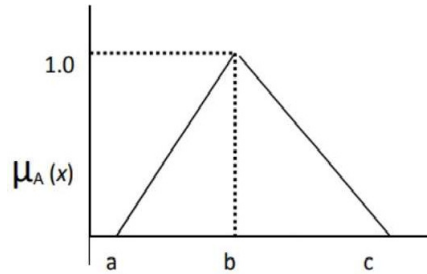


Figure 2. Triangular membership function for fuzzy type-1

Here, the degree of membership functions of the inputs $\text{trimf}(\text{diffColumn}_{i,j}[0, 0, 255])$ i.e. $\mu_{\text{low}}(\text{diffColumn}_{i,j})$, $\text{trimf}(\text{diffColumn}_{i,j}[0, 255, 255])$ i.e. $\mu_{\text{high}}(\text{diffColumn}_{i,j})$, $\text{trimf}(\text{diffRow}_{i,j}[0, 0, 255])$ i.e. $\mu_{\text{low}}(\text{diffRow}_{i,j})$, $\text{trimf}(\text{diffRow}_{i,j}[0, 255, 255])$ i.e. $\mu_{\text{high}}(\text{diffRow}_{i,j})$ are computed by using (4) [26].

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b \leq x \leq c \\ 0 & \text{if } x \geq c \end{cases} \quad (4)$$

Step 6: The four fuzzy rules are defined using the fuzzy sets of the input and output variables of the FIS as depicted in Figure 3.

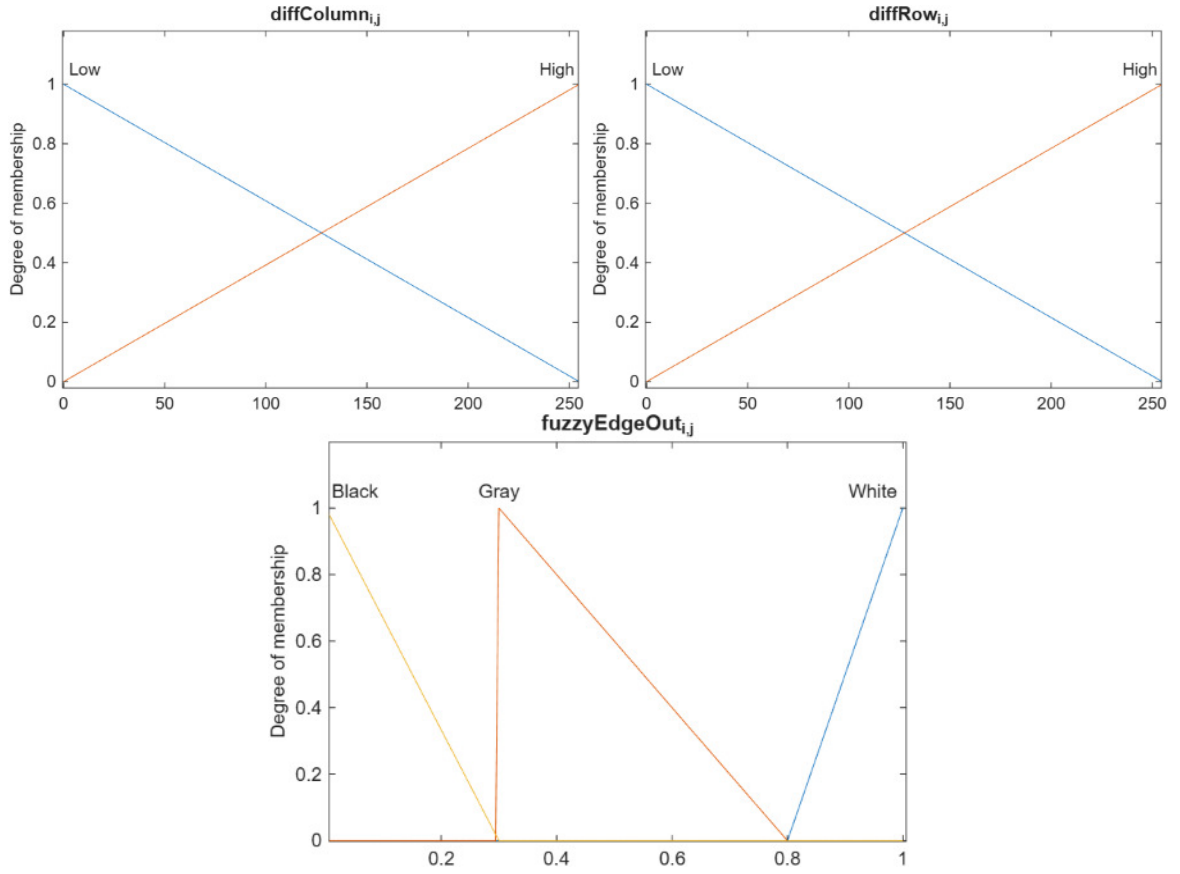


Figure 3. Input and output variables of FIS

Rule 1: IF $\text{diffColumn}_{i,j}$ is low and $\text{diffRow}_{i,j}$ is low, THEN $\text{fuzzyEdgeOut}_{i,j}$ is white.

Rule 2: IF $\text{diffColumn}_{i,j}$ is low and $\text{diffRow}_{i,j}$ is high, THEN $\text{fuzzyEdgeOut}_{i,j}$ is gray.

Rule 3: IF $\text{diffColumn}_{i,j}$ is high and $\text{diffRow}_{i,j}$ is low, THEN $\text{fuzzyEdgeOut}_{i,j}$ is gray.

Rule 4: IF $\text{diffColumn}_{i,j}$ is high and $\text{diffRow}_{i,j}$ is high, THEN $\text{fuzzyEdgeOut}_{i,j}$ is black.

Step 7: Evaluate fuzzy rules to obtain intermediate inference results in the form of a bounded region (A_k) and calculate the centroids (X_k) for each k^{th} rule.

Step 8: The center of sums (CoS) defuzzification method calculates a single crisp output value x^* based on the weighted sum of the intermediate results A_k and X_k . The formula for CoS is:

$$x^* = \frac{\sum_{k=1}^N A_k * \bar{X}_k}{\sum_{k=1}^N A_k}, \quad (5)$$

where A_k represents the bounded area for the k^{th} rule, N is the total number of rules fired, and X_k is the center of the area.

We have set the defuzzified value as 0.8 for the upper threshold. This means that all pixels with a defuzzified crisp value in the range of $[0-0.8]$ are identified as edge pixels and selected for data embedding.

Step 9: Convert the secret message to ASCII format and then convert it to binary form. The message bits (msgBits)

are hidden in the identified edge pixels of G using (6).

$$S_{i,j} = \text{Replace}(\text{kLSBs}(G_{ij}), \text{k msgBits}) \quad (6)$$

where, $S_{i,j}$ is the edge pixel of the stego image (S) obtained after embedding the data, $G_{i,j}$ is the edge pixel of G , $\text{LSBs}(G_{i,j})$ are the least significant bits (LSBs) of the current edge pixel.

4.2. Data extracting phase

At the receiver side, the extraction phase of the *mshEdgeGrayFTI* algorithm, as described in Algorithm 2, is used to extract confidential data from the stego image. The steps up to the identification of edge pixels (steps 1–8) are similar to those in the embedding phase and are applied to the stego image. Then, the k LSBs of the identified stego pixels ($S_{i,j}$) are extracted to recover the embedded secret message using (7).

$$\text{msgBits} = k \text{LSBs}(S_{i,j}) \quad (7)$$

Similarly, all msgBits from the edge pixels of the stego image are extracted through this procedure and combined.

Algorithm 2 Extracting phase of algorithm

- 1: At the receiver's end, read the stego image and follow the embedding algorithm from steps 1 to 8 to identify the edge pixels.
 - 2: Extracted the last k LSBs of the stego image. Combine all the bits and convert them from binary to ASCII format. Finally, transform the ASCII values into characters to reveal the secret message.
-

4.3. Example of the embedding and extracting phases of algorithm *mshEdgeGrayFTI*

Data embedding phase

Step 1: Take a block of 8×8 pixels from the gray scale image (G) as shown in Table 1. $G_{i,j}$ represents the pixel intensity value of the pixel in the i^{th} row and j^{th} column.

Table 1. 8×8 PIXELS BLOCK OF GRAY IMAGE(G)

160	172	181	183	181	181	184	184
162	185	207	213	204	197	204	212
166	185	208	224	225	225	225	226
175	185	202	220	233	239	237	233
194	196	203	214	227	235	238	237
208	213	219	226	231	237	242	244
207	217	231	238	239	238	240	243
212	221	231	238	238	237	235	235

Step 2: Mask the bits of each pixel value of the gray scale cover image using (1) to obtain the masked image (G') as shown in Table 2.

Table 2. MASKED PIXELS BLOCK(G') OF GRAY IMAGE(G)

160	172	180	180	180	180	184	184
160	184	204	212	204	196	204	212
162	184	208	224	224	224	224	224
172	184	200	220	232	236	236	232
192	196	200	212	224	232	236	236
208	212	214	224	228	236	240	244
204	216	228	236	236	236	240	240
212	220	228	236	236	236	232	232

Step 3: Calculate $\text{diffColumn}_{i,j}$ and $\text{diffRow}_{i,j}$ for the masked image (G') using (2) and (3).

$$\text{diffColumn}_{2,6} = |G'_{2,6} - G'_{3,6}| = |196 - 224| = 28$$

$$\text{diffRow}_{2,6} = |G'_{2,6} - G'_{2,7}| = |196 - 204| = 8$$

Step 4: Take $\text{diffColumn}_{2,6}$ and $\text{diffRow}_{2,6}$ as input variables and compute the fuzzy membership values for low and high fuzzy sets associated with these inputs using (4).

$$\mu_{\text{low}}(28) = \left| \frac{255-28}{255} \right| = 0.890$$

$$\mu_{\text{high}}(28) = \left| \frac{28}{255} \right| = 0.109$$

$$\mu_{\text{low}}(8) = \left| \frac{255-8}{255} \right| = 0.960$$

$$\mu_{\text{high}}(8) = \left| \frac{8}{255} \right| = 0.031$$

Step 5: The fuzzy sets Black, Gray, and White are associated with the variable $\text{fuzzyEdgeOut}_{i,j}$ using trimf. The parameters are initialized with the vectors given below for the different triangular membership functions.

$$\text{Black} = [0 \ 0 \ 0.3]$$

$$\text{Gray} = [0.3 \ 0.3 \ 0.8]$$

$$\text{White} = [0.8 \ 1 \ 1]$$

Step 6: The implication is implemented for the four fuzzy rules as shown in Figure 4 and the intermediate results are inferred for each rule as given below:

$$\text{Rule 1 : } \min(\mu_{\text{low}}(28), \mu_{\text{low}}(8)) = \min(0.890, 0.960) = 0.890$$

$$\text{Rule 2 : } \min(\mu_{\text{low}}(28), \mu_{\text{high}}(8)) = \min(0.890, 0.031) = 0.031$$

$$\text{Rule 3 : } \min(\mu_{\text{high}}(28), \mu_{\text{low}}(8)) = \min(0.109, 0.960) = 0.109$$

$$\text{Rule 4 : } \min(\mu_{\text{high}}(28), \mu_{\text{high}}(8)) = \min(0.109, 0.031) = 0.031$$

Step 7: The evaluation of the rules returns intermediate inference results in the form of bounded regions (A_k) and calculates the centroid (X_k) for each k^{th} rule, as shown in Figure 5.





Fired Rule(k)	Bounded Region	Area (A_k)	Centroid (X_k)
1		0.0987	1.0076
2		0.0149	0.5531
3		0.0475	0.5583
4		0.00914	0.1589

Figure 5. Area and centroid of each bounded region

Step 8: Next, apply the center-of-sum (CoS) method to find a single crisp value using (5). The final crisp output is shown in Figure 6.

$$x^* = \frac{\sum_{k=1}^4 A_k \cdot \overline{X_k}}{\sum_{k=1}^4 A_k} = 0.76$$

Since the single defuzzified crisp value falls within the threshold of [0-0.8], the current pixel is marked as an edge pixel and is used to embed one of the secret bits.

Step 9: Let the binary secret message be $(01000111100)_2$ and we want to embed $k=2$ bits, i.e. 00 in the edge pixel. Let $G_{2,6}=197$ be the edge pixel, and the binary representation of 197 is $(11000101)_2$. Replace 2 LSBs of $G_{2,6}$ i.e. 01 with 00 by using (6) and we get $S_{2,6}$ with binary representation $(11000100)_2$ i.e 196. Similarly, embed all bits of the secret message in the edge pixels identified in the grayscale image to obtain the stego image.

Data extracting phase

At the receiver side, after detecting edge pixels, retrieve confidential data from the stego image using (7). i.e., 2 LSBs of $S_{2,6}$ is 2 bits of secret message.

Similarly, extract all secret bits from the stego image and concatenate them to obtain the secret message in binary form. Convert this binary message to ASCII format to obtain the secret message in readable form.

5. Experiment results and discussion

Experimental simulations are conducted to show the integrity of the proposed *mshEdgeGrayFTI* algorithm for image steganography. We have implemented the proposed algorithm on three different cover images of size $256 \times 256 \times 3$ pixels, as shown in Figure 7. To assess the perceptual transparency of the proposed steganographic method, two quality metrics are used: the peak signal-to-noise ratio (PSNR), the mean squared error (MSE) and structural similarity index measure (SSIM) [24] as shown in Table 3. For a better study of the proposed steganographic method, we embedded a different number of secret data bits (k) in the k LSBs of the selected pixels. In all cover images, the value of k is chosen as [1, 2, 3, 4]. To assess the effectiveness of the proposed *mshEdgeGrayFTI* steganographic approach, the threshold values (Th) are used.

Table 4 presents the PSNR and MSE results for $k = [1, 2, 3, 4]$ and $Th = 0.8$ for the three cover images (Lena, Baboon and Airplane). A total of 65,536 secret bits were embedded in the cover images. The results show that the PSNR values are significantly higher for $k = 1$ compared to $k = [2, 3, 4]$. Moreover, as the number of embedded bits increases, PSNR decreases for all images. However, even at the minimum PSNR values, the results remain relatively high, ensuring that the visual differences between the cover and stego images remain imperceptible to the human eye, as shown in Figure 8.

In Table 5, a comparative analysis has been performed based on the tests presented in previous studies, including the OPAP [28], IP [29], ARIP [27] and the proposed *mshEdgeGrayFTI* steganographic method. Chan et al. [28] proposed the OPAP method, an extension of the basic LSB technique [30], which allowed embedding of up to 65,536 bits while achieving relatively higher PSNR values for the three images. Yang [29] proposed an LSB substitution technique using a raster scan and the Inverted Pattern (IP) approach, where parts of secret data were inverted while others remained unchanged before embedding. Amirtharajan et al. [27] introduced an adaptive LSB embedding method, where 256×256 cover images were divided into 4×4 pixel blocks, and encrypted secret messages were embedded within each block. Table 5 shows that the proposed *mshEdgeGrayFTI* method achieves superior PSNR and MSE values across all three cover images compared to previous approaches. A total of 65,536 bits of secret data were embedded in the cover images of size $256 \times 256 \times 3$ pixels, and a comparison was made regarding the hiding data size (bits) PSNR (dB) and MSE for the Lena, Baboon and Airplane images. The *mshEdgeGrayFTI* method, which utilizes a k -bit embedding approach, has been implemented on three different cover images, and the quality of the resulting stego images has been evaluated.

Table 3. IMAGE QUALITY METRICES

Metrics	Formula
PSNR	$10 \cdot \log_{10} \left(\frac{[255]^2}{MSE} \right) \text{ dB}$
MSE	$\frac{\sum_{i=1}^H \sum_{j=1}^W (C(i,j) - C'(i,j))^2}{H \cdot W}$
SSIM(x, y)	$\frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$

Figure 9 shows the resistance of the *mshEdgeGrayFTI* technique against the PDH analysis by examining the pixel difference between the original and stego images. First, the differences between successive pixels in the original image are calculated, falling within the range of -255 to +255. The same process is applied to the stego image, and the PDH plot is then generated by plotting these difference values against their frequency along the x and y axes. A PDH plot for an original image is typically smooth. If the PDH plot of the stego image appears smooth without irregular variations, steganography is not detected.

Table 5. COMPARISON OF PROPOSED ALGORITHM WITH PREVIOUS SCHEME

Transversing Method	Lena		Baboon		Airplane	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
OPAP [28]	51.1251	0.5018	51.1446	0.4996	51.1350	0.5007
IP [29]	51.6171	0.4481	51.5622	0.4538	51.6501	0.4447
ARIP [27]	51.7499	0.4346	51.1446	0.4996	51.6501	0.4447
<i>mshEdgeGrayFTI</i>	58.1276	0.1203	53.7274	0.2674	58.6648	0.0765

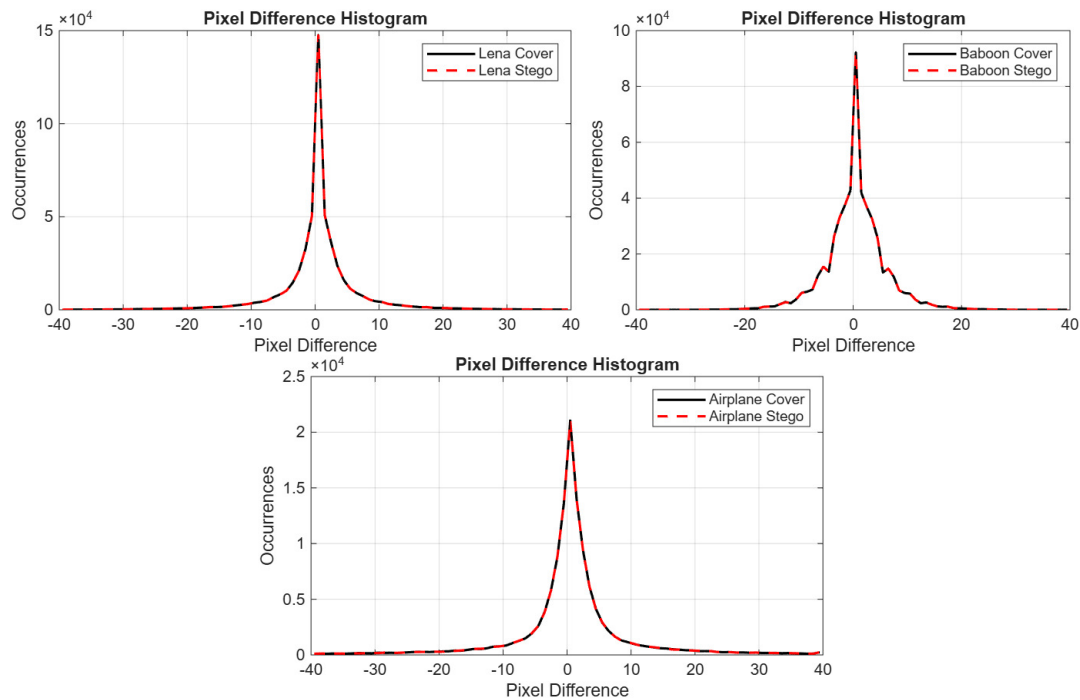


Figure 9. Pixel difference histogram

The proposed method, as detailed in section 4, is uniformly applied to all test images illustrated in Figure 10. A total of 1,96,608 bits are embedded in each test image of size $512 \times 512 \times 3$. The same embedding procedure is repeated for each image. Table 6 presents the results for different values of $k=[1, 2, 3]$ bits. These results are compared with the Type-1 Fuzzy Logic approach originally developed in [24] and modified using a standard procedure.

Figure 10. Test images of size $512 \times 512 \times 3$ Table 6. Comparison of Previous studies and *mshEdgeGrayFTI*

Image	Previous studies [24]						<i>mshEdgeGrayFTI</i>					
	k=1		k=2		k=3		k=1		k=2		k=3	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	51.8292	0.9973	45.8211	0.9896	39.5811	0.9617	66.3374	1.0000	59.8423	0.9999	53.3154	0.9996
Peppers	51.3590	0.9962	45.2655	0.9833	38.9900	0.9409	65.1016	0.9998	58.4918	0.9996	51.9018	0.9989
Earth	51.3538	0.9973	45.3733	0.9896	39.1242	0.9608	60.4417	0.9999	53.9246	0.9994	47.3715	0.9985

6. Conclusion

In this paper, we illustrate a new algorithm, *mshEdgeGrayFTI*, in which Mamdani FIS is applied for the detection of edge pixels, and the LSB technique is used to embed confidential data in edge pixels according to a specified range interval. The experimental results demonstrate that the proposed technique outperforms existing methods in terms of embedding capacity and imperceptibility. Additionally, *mshEdgeGrayFTI* effectively withstands the pixel difference histogram (PDH) analysis, enhancing its security against detection. In the future, we will further improve the embedding capacity of the stego image by using fuzzy logic type-2 so that it can hide a greater number of secret bits.

Acknowledge

This work was supported by CSIR, New Delhi, under the Junior Research Fellowship CSIR-Award No: 09/0382(15463)/2022-EMR-1) through the first author and CSIR, New Delhi, under junior research fellowship CSIR-Award No: 09/0382(15462)/2022-EMR-1) through the third author.

Conflicts of interest

The authors have no conflict of interest in any aspect.

References

1. D.C. Wu and W.H. Tsai, *A steganographic method for images by pixel-value differencing*, Pattern Recognition Letters, vol. 24, no. 9-10, pp. 1613–1626, 2003.
2. H. S. Yusuf and H. Hagsras, *High payload image steganography method using fuzzy logic and edge detection*, Int. J. Comput. Sci. Trends Technol., vol. 8, no. 4, pp. 123–134, 2020.
3. H. Dadgostar and F. Afsari, *Image steganography based on interval-valued intuitionistic fuzzy edge detection and modified LSB*, Journal of Information Security and Applications, vol. 30, pp. 94–104, 2016.
4. A. Divya and S. Thenmozhi, "Steganography: various techniques in spatial and transform domain," *International Journal of Advanced Scientific Research and Management*, vol. 1, no. 3, pp. 81–89, 2016.
5. S. Kumar, A. Singh, and M. Kumar, *Information hiding with adaptive steganography based on novel fuzzy edge identification*, Defence Technology, vol. 15, no. 2, pp. 162–169, 2019.
6. L. A. Zadeh, *Fuzzy sets*, Information and Control, 1965.
7. A. M. Alawad, F. D. A. Rahman, O. O. Khalifa, and N. A. Malek, *Fuzzy logic based edge detection method for image processing*, International Journal of Electrical and Computer Engineering, vol. 8, no. 3, pp. 1863, 2018.
8. S. Rane *et al.*, *Performance of Mamdani Fuzzy Inference System for Tracking Multiple Targets Using Autopilot System*, International Journal of Electrical and Electronics, vol. 5, no. 1, pp. 34–42, 2017.
9. H. Herpratiwi, M. Maftuh, W. Firdaus, A. Tohir, M. I. Daulay, and R. Rahim, *Implementation and Analysis of Fuzzy Mamdani Logic Algorithm from Digital Platform and Electronic Resource*, TEM Journal, vol. 11, no. 3, pp. 1028–1033, 2022.
10. H. Espitia, J. Soriano, I. Machón, and H. López, *Design methodology for the implementation of fuzzy inference systems based on boolean relations*, Electronics, vol. 8, no. 11, pp. 1243, 2019.
11. G. Viji and J. Balamurugan, *LSB Steganography in Color and Grayscale Images without using the Transformation*, Bonfring International Journal of Advances in Image Processing, vol. 1, pp. 11, 2011.
12. W. J. Chen, C. C. Chang, and T. H. N. Le, *High payload steganography mechanism using hybrid edge detector*, Expert Systems with Applications, vol. 37, no. 4, pp. 3292–3301, 2010.
13. A. Hore and D. Ziou, *Image quality metrics: PSNR vs. SSIM*, in Proc. 20th International Conference on Pattern Recognition, pp. 2366–2369, 2010.
14. M. Khodaei and K. Faez, "New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing," *IET Image Processing*, vol. 6, no. 6, pp. 677–686, 2012.
15. H. W. Tseng and H. S. Leng, "High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion," *IET Image Processing*, vol. 8, no. 11, pp. 647–654, 2014.
16. S. Islam, M. R. Modi, and P. Gupta, "Edge-based image steganography," *EURASIP Journal on Information Security*, vol. 2014, pp. 1–14, 2014.
17. J. Bai, C. C. Chang, T. S. Nguyen, C. Zhu, and Y. Liu, "A high payload steganographic algorithm based on edge detection," *Displays*, vol. 46, pp. 42–51, 2017.

18. B. Santhi and B. Dheeptha, "A novel edge based embedding in medical images based on unique key generated using sudoku puzzle design," *SpringerPlus*, vol. 5, pp. 1–16, 2016.
19. C. C. Chang, T. S. Nguyen, and T. Y. Chien, "An efficient steganography scheme based on edge detection for high payload," *J. Inf. Hiding Multim. Signal Process.*, vol. 8, no. 5, pp. 967–979, 2017.
20. A. A. Alghamdi, "Computerized steganographic technique using fuzzy logic," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018.
21. S. Dhargupta, A. Chakraborty, S. K. Ghosal, S. Saha, and R. Sarkar, "Fuzzy edge detection based steganography using modified Gaussian distribution," *Multimedia Tools and Applications*, vol. 78, pp. 17589–17606, 2019.
22. S. N. M. Al-Faydi, S. K. Ahmed, and H. N. Y. Al-Talb, "Improved LSB image steganography with high imperceptibility based on cover-stego matching," *IET Image Processing*, vol. 17, no. 7, pp. 2072–2082, 2023.
23. X. Feng, C. Zhu, and Z. Ge, "Research on low resolution digital image reconstruction method based on rational function model," *IAENG International Journal of Computer Science*, vol. 51, no. 2, 2024.
24. Z. Ashraf, M. L. Roy, P. K. Muhuri, and Q. M. D. Lohani, "Interval type-2 fuzzy logic system based similarity evaluation for image steganography," *Heliyon*, vol. 6, no. 5, 2020.
25. A. Salimi, M. Subaşı, L. Buldu, and Ç. Karataş, "Prediction of flow length in injection molding for engineering plastics by fuzzy logic under different processing conditions," *Iranian Polymer Journal*, vol. 22, pp. 33–41, 2013.
26. M. H. Azam, M. H. Hasan, S. Hassan, and S. J. Abdulkadir, "Fuzzy type-1 triangular membership function approximation using fuzzy C-means," in *Proc. 2020 Int. Conf. on Computational Intelligence (ICCI)*, pp. 115–120, 2020.
27. R. Amirtharajan and J. B. B. Rayappan, "An intelligent chaotic embedding approach to enhance stego-image quality," *Information Sciences*, vol. 193, pp. 115–124, 2012.
28. C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.
29. C. H. Yang, "Inverted pattern approach to improve image quality of information hiding by LSB substitution," *Pattern Recognition*, vol. 41, no. 8, pp. 2674–2683, 2008.
30. R. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," *Pattern Recognition*, vol. 34, no. 3, pp. 671–683, 2001.