STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
*Stat., Optim. Inf. Comput.*, Vol. **16**, December 2026, pp 5799–5814.
Published online in International Academic Press (www.IAPress.org)

# An Interval Type-2 Fuzzy Logic-Guided Adaptive LSB Steganography Method for Secure Data Hiding

Vinita Yadav, Meenakshi Hooda*, Navita Dhaka

*Department of Mathematics, Maharashi Dayanand University, Rohtak, India*

**Abstract** This study introduces *mshEdgeGrayFT2*, an edge-based image steganography approach that combines an Interval Type-2 Fuzzy Logic System (IT2FLS) with LSB embedding. Horizontal and vertical intensity variations between neighboring pixels are considered as inputs to the fuzzy system for determining appropriate edge regions for secret-data embedding. Upper and lower membership functions are used to represent variations in the degree of membership, allowing uncertainty in local pixel differences to be considered during edge selection. The effectiveness of the method is examined at different embedding levels using PSNR, MSE, and SSIM. The results are analyzed to study the trade-off between data-hiding capacity and the visual quality of the resulting stego images.

**Keywords** Image steganography, Interval type 2 fuzzy logic system, Data hiding, Least Significant Bit

**DOI:** 10.19139/soic-2310-5070-4809

## 1. Introduction

*The growth* of communication networks, advances in data technologies, and the widespread availability of World Wide Web services have greatly improved the access to various types of information. However, this progress also presents challenges to keeping sensitive data secure. As a result, steganography has become a valuable technique to hide confidential information in other media, such as images, text, audio, and video [1]. Image steganography is particularly effective in protecting our privacy because it takes advantage of the human visual system (HVS) and reduces the cost of image storage and transmission [2]. Image steganography techniques are generally classified into spatial and frequency domain methods. The spatial domain methods hide the secret message by altering specific pixels in the image. In contrast, the frequency domain methods involve changing the image's representation (e.g., using a discrete wavelet transform) before embedding the message using standard techniques [3]. Steganography involves two main steps: embedding and extraction. Embedding involves hiding confidential data within the pixels of an image, whereas extraction retrieves the concealed information. The image before embedding is termed the cover image, while the image after embedding is referred to as the stego image [4]. When designing a digital data-hiding system, several important factors must be considered: Imperceptibility, Embedding capacity, and Robustness. Imperceptibility is the property in which the hidden message should be impossible for a person to detect, and the stego image should look just like the original image. Embedding capacity refers to how much secret information can be hidden in the image without affecting its quality. Robustness indicates how hard it is to remove the hidden information without damaging the original image [5]. The security of a steganography technique depends on which pixels are selected for embedding. Pixels in areas with texture or noise are better choices because they are harder to detect. Edge pixels are handy, as their intensity differs from neighbouring pixels due to abrupt changes in the gradient. These sharp variations make edges more challenging to model than pixels in smoother

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

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

regions [6]. Researchers have introduced many traditional edge detectors such as Sobel, Robert, Laplacian, Prewitt, Fuzzy, and Canny operators. Among these, the canny edge detector (CED) and fuzzy logic edge detection (FLED) are widely used in academic research and industry. Fuzzy logic is a soft computing method that simulates human reasoning by addressing uncertainty and imprecision in data. It is especially beneficial for edge detection in images, as it can more effectively handle unclear and noisy variations in pixel intensity compared to traditional methods. In 1965, Zadeh introduced the concept of fuzzy sets, which are elements that can belong to a set to a certain degree between 0 and 1. This capability makes fuzzy logic more suitable for processing vague or imperfect data [7]. The main challenge with fuzzy sets is the lack of optimization techniques to address certain issues. To overcome this, the concept of L-fuzzy sets was introduced [8]. This advancement in fuzzy logic led to the development of fuzzy logic systems, now widely used in system modeling and control. Fujita et al. [9] introduced Fuzzy Tree-Decomposition and Fuzzy Path-Decomposition as extensions of classical tree- and path-decomposition concepts to fuzzy graphs. Their approach incorporates edge membership degrees to account for uncertainty and provides a framework for analyzing graph-width parameters in fuzzy environments. Mamdani and Baaklini [10] introduced the fuzzy logic controller, which has become a fundamental tool in many applications. Moreover, James C. Bezdek developed the fuzzy c-means algorithm, which has become an important technique in clustering and pattern recognition [11]. In 1975, Zadeh proposed type-2 fuzzy sets as an extension of conventional type-1 fuzzy sets[12]. Type-2 fuzzy sets enhance the ability to manage linguistic and numerical uncertainties. Later, Karnik and Mendel [13] developed a type-2 fuzzy logic system(T2FLS) incorporating specific rules to manage uncertainties within the membership function. However, despite its effectiveness, type-2 fuzzy logic systems are computationally complex. To simplify this issue, researchers developed interval type-2 fuzzy logic systems, which reduce computational demands while maintaining the benefits of type-2 fuzzy logic [14]-[15]. Fujita et al. [16] stigated Plithogenic Fuzzy Control as a generalization of classical fuzzy control by incorporating attribute-based membership and contradiction functions. The proposed framework integrates Upside-Down Logic to model truth–falsity reversals and provides a mechanism for handling ambiguity and inversion in reasoning and control processes. Gerardo et al. [17] use the Mamdani interval singleton type-2 fuzzy logic system, combined with the central composite design technique, to refine fuzzy rule base development. This approach enhances the evaluation of dimensional attributes in manufacturing quality. Yadav et al. [18] compared the results of algorithms *mshEdgeGrayFT1* [18] and *mshEdgeGrayFT2*, which are based on fuzzy logic type-1 and type-2. Wang et al. [20] developed a Genetic Algorithm (GA) based LSB steganographic technique. Chan et al. [21] introduced a steganographic approach that performs the LSB technique for embedding and incorporates an optimal pixel adjustment procedure (OPAP) to minimize distortion between the original and the stego images. Demirci et al. [22] introduced the similarity measure-based LSB technique in which the smoothness of each pixel in a color image is first calculated using a similarity relation matrix and normalized to the maximum gray level, mapping the three-dimensional color space into one dimension. The edges are detected by assigning pixels lower than thresholds as edge pixels. Karakis et al. [23] introduced two novel image steganography techniques utilizing fuzzy logic and similarity measures to identify non-sequential least significant bits (LSB) for embedding data within image pixels. Ashraf. [24] utilizes a Mamdani inference engine with IT2 MFs to compute the IT2 fuzzy similarity between image pixels based on a set of IT2 fuzzy rules.

Although fuzzy logic and edge-based LSB embedding have previously been investigated, uncertainty in local intensity variations remains an important consideration when identifying suitable regions for data embedding. Unlike a Type-1 fuzzy system, which assigns a single membership grade to each input, the proposed *mshEdgeGrayFT2* method employs an Interval Type-2 Fuzzy Logic System (IT2FLS) with upper and lower membership functions to represent uncertainty in horizontal and vertical pixel differences. This enables a more flexible evaluation of ambiguous local intensity transitions for fuzzy-guided pixel selection rather than relying solely on fixed gradient thresholds or precise Type-1 membership grades. By prioritizing locally varying image regions for LSB embedding, the method is intended to support relatively high-capacity data hiding while maintaining visual similarity between cover and stego images. Potential applications include confidential image-based communication, copyright or ownership information embedding, and metadata hiding. Applications in sensitive domains, such as medical or mission-critical imaging, would require additional domain-specific evaluation of robustness, information preservation, and resistance to steganalysis. The main contributions of this work are: (i) the use of horizontal and vertical masked pixel differences as inputs to an IT2FLS for identifying

suitable embedding regions; (ii) the use of upper and lower membership functions and a fuzzy rule base to model uncertainty in edge-pixel selection; (iii) integration of the resulting edge-selection mechanism with multi-bit LSB embedding; and (iv) experimental investigation of the capacity–imperceptibility relationship using PSNR, MSE, and SSIM at different embedding levels.

## 2. Fuzzy logic system

The design of T2 FLS focuses mainly on managing uncertainty in information to improve system performance for specific applications. However, the construction of type-2 fuzzy rules remains the same as in type-1 FLS, with the key distinction being the nature of the membership functions. This section introduces the fundamentals of IT2 FLS. To understand an IT2 FLS, it is essential first to discuss T2 FSs and IT2 FSs. Therefore, this section aims to provide readers with a basic understanding of the general concepts and principles of IT2 FLS.

### *2.1. T2 FSs and IT2 FSs*

A type-2 fuzzy set (T2 FS) $\tilde{A}$ is characterized by a membership function $\mu_{\tilde{A}}(x,u)$, defined over variables $x$ and $u$. Here, $x$ belongs to the universe of discourse $X$, while $u$ denotes its primary membership function [25]. Consequently, the T2 FS $\tilde{A}$ can be expressed as given in (1).

$$\tilde{A} = \{(x,u), \mu_{\tilde{A}}(x,u) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{1}$$

When $\mu_{\tilde{A}}(x,u) = 1$ holds for the secondary membership function, the T2 FS simplifies to an IT2 FS [26], defined in (2).

$$\tilde{A} = \{(x,u), \mu_{\tilde{A}}(x,u) = 1 \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{2}$$

The graphical depiction of an IT2 FS is illustrated in Figure 1, where $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$ denote the lower membership function (LMF) and upper membership function (UMF). The area between these two functions is known as the Footprint of Uncertainty (FOU), given by (3).

$$\text{FOU} = \left[\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x)\right] \tag{3}$$

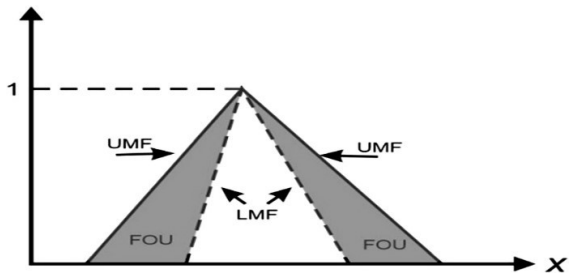



Figure 1. Interval Type-2 Fuzzy Set

### *2.2. Interval type-2 fuzzy logic systems (IT2 FLS)*

In the proposed implementation, Mamdani inference is employed and the interval Type-2 output is type-reduced using the Karnik–Mendel/Enhanced Karnik–Mendel procedure before obtaining the final crisp output. The same inference parameters are used during both embedding and extraction. An IT2 FLS comprises four main components: fuzzification, rule evaluation, type reduction, and defuzzification. The process follows these sequential steps:

• Fuzzification: Converts a crisp input to an IT2 FS using IT2 MFs.
• Rule evaluation: The inference engine applies if-then rules to process the input IT2 FSs and generates an output IT2 FS.
• Type reduction: Transforming the resulting IT2 FS into a T1 FS.
• Defuzzification: Converts the type-reduced set into a final crisp numerical value.

## 3. Proposed steganographic method

In this work, an image steganography method, *mshEdgeGrayFT2*, is proposed by integrating an Interval Type-2 Fuzzy Logic System (IT2FLS) with LSB embedding. The proposed method analyzes local intensity variations and generates a fuzzy suitability value for each candidate pixel. The fuzzy output is used to identify the primary fuzzy-edge regions and to establish the priority of candidate pixels for secret-data embedding. The number of least significant bits used for embedding is controlled by the embedding level $m$, where $m \in \{1, 2, 3, 4\}$. A pixel is considered suitable for embedding when the final crisp output satisfies

$$C_\beta \leq T_h,$$

where $T_h = 0.8$ is the nominal threshold used in the present experiments.

### *3.1. mshEdgeGrayFT2 embedding procedure*

In the proposed algorithm, each pixel in the grayscale image is processed using a 3×3 window to compute the defuzzified value. A pixel is selected for data hiding when its defuzzified value is less than 0.8. The proposed algorithm consists of six main steps: (i) computing pixel differences in both vertical and horizontal directions, (ii) applying interval type-2 fuzzy logic, (iii) utilizing a rule-based inference system, (iv) performing type reduction, (v) executing defuzzification, and (vi) embedding the data. The embedding process of ***mshEdgeGrayFT2*** is depicted in Figure 2. A detailed description of each step of ***mshEdgeGrayFT2*** is presented in the proceeding subsections.

#### *3.1.1. Computing pixel differences in both vertical and horizontal directions*

(a) Select an image as a cover image (C). If the cover image is an RGB image, transform it to a grayscale image (Z). Z is a matrix of dimensions P × Q, and each element $Z_{i,j}$ represents the pixel intensity at the $i^{th}$ row and $j^{th}$ column of Z, where $1 \leq i \leq P$ and $1 \leq j \leq Q$. The range of pixel intensity values is [0-255]. In the proposed algorithm, a 3×3 sample matrix is taken as given below:

| | | |
|---|---|---|
| 59 | 147 | 134 |
| 118 | 67 | 156 |
| 111 | 178 | 97 |

$$Z_{2,2} = 67, Z_{2,3} = 156, Z_{3,2} = 178$$

(b) The masking operation clears the $m$ least significant bits of each pixel before the computation of local intensity differences. This operation ensures that the pixel characteristics used by the fuzzy system are not affected by the information subsequently embedded within these $m$ least significant bits. For illustration, let the embedding level be $m = 2$. Before calculating the horizontal and vertical differences, the two least significant bits of each pixel are cleared using (4)

$$Z_{i,j}^{(m)} = \left\lfloor \frac{Z_{i,j}}{2^m} \right\rfloor 2^m, \tag{4}$$

For the central pixel and its neighbouring pixels,

$$Z_{2,2} = 67, \qquad Z_{3,2} = 178, \qquad Z_{2,3} = 156$$

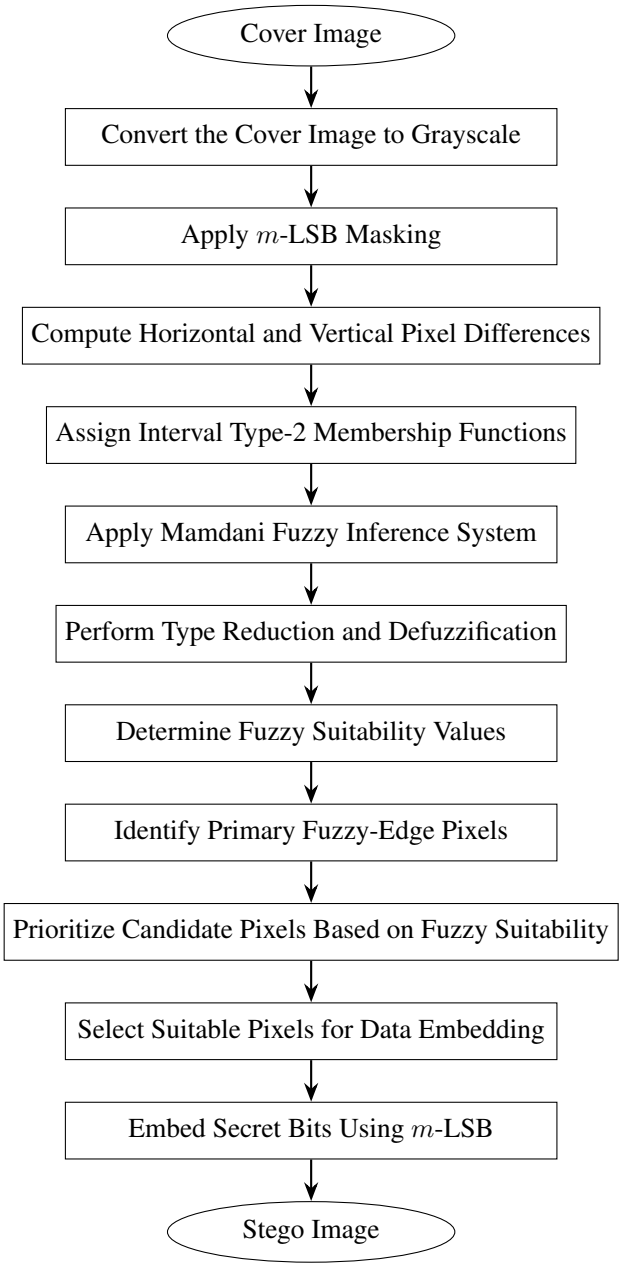



Figure 2. Embedding procedure of the proposed *mshEdgeGrayFT2* algorithm.

.

After $m = 2$ LSB masking,

$$Z_{2,2}^{(2)} = \left\lfloor \frac{67}{4} \right\rfloor 4 = 64,$$

$$Z_{3,2}^{(2)} = \left\lfloor \frac{178}{4} \right\rfloor 4 = 176,$$

and

$$Z_{2,3}^{(2)} = \left\lfloor \frac{156}{4} \right\rfloor 4 = 156.$$

(c) Therefore, the vertical and horizontal differences are

$$D_{2,2} = |Z_{2,2}^{(2)} - Z_{3,2}^{(2)}| = |64 - 176| = 112,$$

and

$$D'_{2,2} = |Z_{2,2}^{(2)} - Z_{2,3}^{(2)}| = |64 - 156| = 92.$$

Thus, the two inputs supplied to the interval Type-2 fuzzy inference system for the central pixel are

$$(D_{2,2}, D'_{2,2}) = (112, 92).$$

### *3.1.2. Assignment of IT2 Membership Functions to Pixel Differences*

In this step, the vertical and horizontal differences between each pixel pairs are assigned IT2 MFs, which we call $D_{i,j}$ and $D'_{i,j}$, corresponding to the linguistic terms Low(L) and High (H). Various IT2 MFs, including triangular, Gaussian, trapezoidal, etc., can be used to handle these linguistic terms. In our algorithm, triangular membership functions(TMFs) are used because each linguistic value is defined as an interval, with its left and right endpoints determined by experts [27]. The IT2 MFs are defined in (5), with its lower membership function (LMF) and upper membership function (UMF) specified in (6) and (7) [28].

$$\mu_{\tilde{A}_i}(x) = [\underline{\mu}_{\tilde{A}_i}(x), \bar{\mu}_{\tilde{A}_i}(x)] \tag{5}$$

For an input x, the membership values for a triangular MF are

$$\underline{\mu}_{\tilde{A}_i}(x; a', b, c') = \begin{cases} 0 & \text{if} \quad x \le a' \\ \frac{x-a'}{b-a'} & \text{if} \quad a' \le x \le b \\ \frac{c'-x}{c'-b} & \text{if} \quad b \le x \le c' \\ 0 & \text{if} \quad x \ge c' \end{cases} \tag{6}$$

$$\bar{\mu}_{\tilde{A}_i}(x; a, b, c) = \begin{cases} 0 & \text{if} \quad x \le a \\ \frac{x-a}{b-a} & \text{if} \quad a \le x \le b \\ \frac{c-x}{c-b} & \text{if} \quad b \le x \le c \\ 0 & \text{if} \quad x \ge c \end{cases} \tag{7}$$

In (6) and (7), the five input parameters $a; a'; b; c; c'$ are defined, where $b \in [a, c]$ and $b \in [a', c']$ are utilized in LMF $\underline{\mu}_{\tilde{A}_i}(x; a', b, c')$ and UMF $\bar{\mu}_{\tilde{A}_i}(x; a', b, c')$. The input variables are $D_{i,j}$ and $D'_{i,j}$. When the vertical and horizontal difference $D_{i,j}$ and $D'_{i,j}$ fall in the interval [0 0 255], [0 255 255], they will be assigned the linguistic terms L and H. The upper membership functions of low and high TMFs are defined by the parameters [0 0 255] and [0 255 255], and the lower membership functions of low and high TMFs are [0 0 205] and [51 255 255]. Table 1 represents the LMF and UMF for inputs 112 and 92 using (6) and (7).

### *3.1.3. Rules and inference engine*

Next, we define four fuzzy rules using the "AND" operator and "IF-THEN" condition that establish the relationship between inputs and outputs. The output $(C_{B_{i,j}})$ is implemented using three triangular membership functions representing the linguistic values: Black, Gray, and White. The fuzzy rules are given in Table 2. The implication results of the four fuzzy rules with respect to UMF and LMF are given in Figure 3 and the intermediate results

Table 1. Calculation of upper and lower membership function

| Membership Function | **Parameters [a, b, c]** | **Calculation** | **Result** |
|---|---|---|---|
| UMF (Upper Membership Function) | [0, 0, 255] | $\bar{\mu}_{low}(112) = \frac{255-112}{255-0}$ | 0.5607 |
| | [0, 255, 255] | $\bar{\mu}_{high}(112) = \frac{112-0}{255-0}$ | 0.4392 |
| | [0, 0, 255] | $\bar{\mu}_{low}(92) = \frac{255-92}{255-0}$ | 0.6392 |
| | [0, 255, 255] | $\bar{\mu}_{high}(92) = \frac{92-0}{255-0}$ | 0.3607 |
| LMF (Lower Membership Function) | [0, 0, 205] | $\underline{\mu}_{low}(112) = \frac{205-112}{205-0}$ | 0.4536 |
| | [50, 255, 255] | $\underline{\mu}_{high}(112) = \frac{112-50}{255-50}$ | 0.3024 |
| | [0, 0, 205] | $\underline{\mu}_{low}(92) = \frac{205-92}{205-0}$ | 0.5512 |
| | [50, 255, 255] | $\underline{\mu}_{high}(92) = \frac{92-50}{255-50}$ | 0.2048 |

inferred for each fuzzy rule are given in Table 3.The fuzzy rules evaluate the local intensity variations and produce an interval type-2 fuzzy output representing the suitability of the corresponding pixel for data embedding. The resulting crisp values are subsequently used to determine the embedding priority of the candidate pixels.

Table 2. Fuzzy Rules

| **Rule** | **Condition** | **Output** |
|---|---|---|
| $R^1$ | IF $D_{i,j}$ is Low AND $D'_{i,j}$ is Low | $\tilde{C}_B$ is White |
| $R^2$ | IF $D_{i,j}$ is Low AND $D'_{i,j}$ is High | $\tilde{C}_B$ is Gray |
| $R^3$ | IF $D_{i,j}$ is High AND $D'_{i,j}$ is Low | $\tilde{C}_B$ is Gray |
| $R^4$ | IF $D_{i,j}$ is High AND $D'_{i,j}$ is High | $\tilde{C}_B$ is Black |

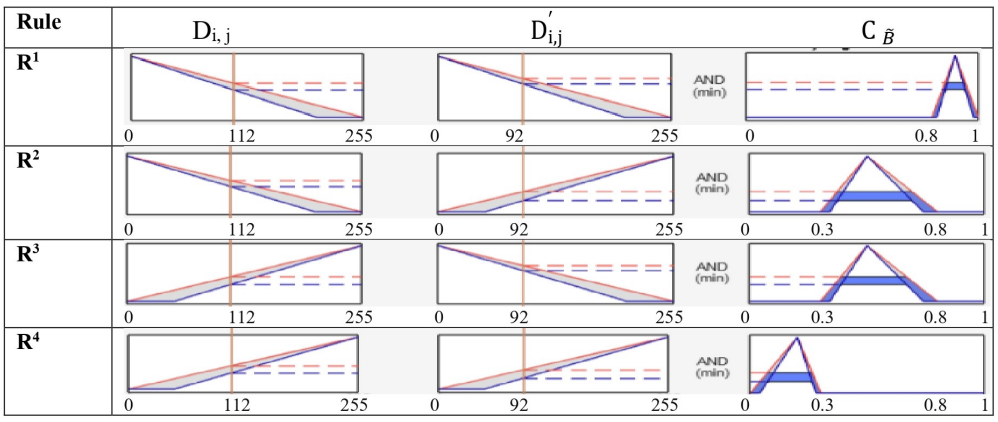



Figure 3. Implication of fuzzy rules

### *3.1.4. Type-reduction*

The aggregated output of the Interval Type-2 Fuzzy Logic System (IT2FLS) is an interval Type-2 fuzzy set and therefore cannot be directly represented by a single crisp value. A type-reduction operation is first performed to

Table 3. Fuzzy Rule Evaluation

| Rule | Expression | Evaluation | Result |
|---|---|---|---|
| $R^1$ (UMF) | $\bar{\mu}_{white}(x) = \min(\bar{\mu}_{low}(112), \bar{\mu}_{low}(92))$ | $\min(0.5607, 0.6392)$ | 0.5607 |
| $R^2(UMF)$ | $\bar{\mu}_{Gray}(x) = \min(\bar{\mu}_{low}(112), \bar{\mu}_{high}(92))$ | $\min(0.5607, 0.3607)$ | 0.3607 |
| $R^3(UMF)$ | $\bar{\mu}_{Gray}(x) = \min(\bar{\mu}_{high}(112), \bar{\mu}_{low}(92))$ | $\min(0.4392, 0.6392)$ | 0.4392 |
| $R^4(UMF)$ | $\bar{\mu}_{Black}(x) = \min(\bar{\mu}_{high}(112), \bar{\mu}_{high}(92))$ | $\min(0.4392, 0.3607)$ | 0.3607 |
| $R^1$(LMF) | $\underline{\mu}_{white}(x) = \min(\underline{\mu}_{low}(112), \underline{\mu}_{low}(92))$ | $\min(0.4536, 0.5512)$ | 0.4536 |
| $R^2$(LMF) | $\underline{\mu}_{Gray}(x) = \min(\underline{\mu}_{low}(112), \underline{\mu}_{high}(92))$ | $\min(0.4536, 0.2048)$ | 0.2048 |
| $R^3$(LMF) | $\underline{\mu}_{Gray}(x) = \min(\underline{\mu}_{high}(112), \underline{\mu}_{low}(92))$ | $\min(0.3024, 0.5512)$ | 0.3024 |
| $R^4$(LMF) | $\underline{\mu}_{Black}(x) = \min(\underline{\mu}_{high}(112), \underline{\mu}_{high}(92))$ | $\min(0.3024, 0.2048)$ | 0.2048 |

transform the aggregated IT2 fuzzy output into an interval-valued Type-1 set. In the proposed *mshEdgeGrayFT2* algorithm, the Enhanced Karnik–Mendel (EKM) algorithm is employed for this purpose. For the output variable, the upper and lower membership functions are defined as

$$\begin{aligned} &\text{UMF:} \quad \text{Black} = [0, 0.2, 0.3], \quad \text{Gray} = [0.3, 0.6, 0.8], \quad \text{White} = [0.8, 0.9, 1], \\ &\text{LMF:} \quad \text{Black} = [0.038, 0.2, 0.2777], \quad \text{Gray} = [0.338, 0.5, 0.738], \quad \text{White} = [0.82, 0.9, 0.983]. \end{aligned}$$

For each pixel $B_{i,j}$, the EKM procedure determines the left and right endpoints of the type-reduced centroid interval. The resulting type-reduced set is expressed as

$$C^{TR}_{B_{i,j}} = \left[ C^{L}_{B_{i,j}}, C^{R}_{B_{i,j}} \right], \tag{8}$$

where $C^{L}_{B_{i,j}}$ and $C^{R}_{B_{i,j}}$ denote the left and right endpoints, respectively, of the type-reduced centroid interval for the pixel located at $(i, j)$. These endpoints incorporate the uncertainty represented by the lower and upper membership functions of the IT2 fuzzy output.

For the considered numerical example, the type-reduction stage produces the two endpoint values

$$\begin{aligned} C^{L}_{B_{i,j}} &= \frac{\int_X \bar{\mu}_{\tilde{A}_i}(x)\, x\, dx}{\int_X \bar{\mu}_{\tilde{A}_i}(x)\, dx}, \qquad C^{R}_{B_{i,j}} = \frac{\int_X \underline{\mu}_{\tilde{A}_i}(x)\, x\, dx}{\int_X \underline{\mu}_{\tilde{A}_i}(x)\, dx} \\ C^{L}_{B_{i,j}} &= 0.5350, \qquad C^{R}_{B_{i,j}} = 0.5630. \end{aligned} \tag{9}$$

### *3.1.5. Defuzzification*

After type-reduction, the interval $C^{TR}_{B_{i,j}}$ is converted into a single crisp value. The final crisp fuzzy output corresponding to pixel $B_{i,j}$ is obtained by taking the arithmetic mean of the left and right endpoints:

$$C_{B_{i,j}} = \frac{C^{L}_{B_{i,j}} + C^{R}_{B_{i,j}}}{2}. \tag{10}$$

For the above example,

$$C_{B_{i,j}} = \frac{0.5350 + 0.5630}{2} = 0.5490. \tag{11}$$

Hence, the final crisp fuzzy output for the current pixel is

$$\boxed{C_{B_{i,j}} = 0.5490}.$$

The value $C_{B_{i,j}}$ represents the fuzzy suitability of pixel $B_{i,j}$ for edge-based data embedding. According to the fuzzy-rule convention adopted in the proposed method, a smaller value indicates a stronger edge-related embedding preference.

The primary fuzzy-edge pixels are identified using the threshold condition

$$C_{B_{i,j}} \leq Th, \qquad Th = 0.8. \tag{12}$$

For the considered example,

$$0.5490 \leq 0.8.$$

Therefore, pixel $B_{i,j}$ satisfies the fuzzy-edge criterion and belongs to the primary fuzzy-edge set. In addition to identifying the primary fuzzy-edge pixels, the crisp fuzzy outputs are used to establish the priority of candidate pixels for data embedding. The candidate pixels are arranged in ascending order of their fuzzy suitability values as

$$C_\beta(p_1) \leq C_\beta(p_2) \leq \cdots \leq C_\beta(p_N), \tag{13}$$

where $p_1, p_2, \ldots, p_N$ denote the candidate pixel locations. Pixels with smaller values of $C_\beta$ are assigned higher embedding priority according to the adopted fuzzy-rule convention. The final crisp value falls in the interval $[0.8]$, marking the current pixels as edge pixels. The procedure is repeated for all pixels to check whether they are edge pixels.

For a given embedding requirement, the required number of highest-priority pixels is selected from the ordered candidate set. The selected embedding set can therefore be represented as

$$\mathcal{E}_s = \{p_1, p_2, \ldots, p_{N_s}\}, \tag{14}$$

where $N_s$ denotes the number of pixels selected for data embedding.

### *3.1.6. Embedding Procedure*

After determining the fuzzy suitability of the candidate pixels, the selected pixels are used for secret-data embedding according to their priority. For an embedding level $m$, the secret bitstream is divided into groups of $m$ bits. Each group is converted into its decimal equivalent $b_k$, satisfying

$$0 \leq b_k < 2^m. \tag{15}$$

For a selected cover pixel $I(p_k)$, the stego pixel is obtained as

$$S(p_k) = \left\lfloor \frac{I(p_k)}{2^m} \right\rfloor 2^m + b_k, \tag{16}$$

where $S(p_k)$ denotes the resulting stego pixel. If $N_s$ pixels are selected and $m$ bits are embedded in each selected pixel, the total embedding payload is

$$P = N_s \times m, \tag{17}$$

where $P$ denotes the number of embedded secret bits. During the embedding phase of the ***mshEdgeGrayFT2*** algorithm, the LSB technique embeds the secret message, converts the decimal value of the pixels into its binary representation, then replaces the last m bits with the secret data bits.

Consider the secret message

$$M = (01000111100)_2.$$

For illustration, let $m = 2$, and suppose the first two secret bits to be embedded in the selected pixel are

$$(01)_2.$$

Assume that the pixel $Z_{2,2} = 67$ has been selected according to its fuzzy suitability ranking. Its binary representation is

$$67 = (01000011)_2.$$

For $m = 2$, the two least significant bits are cleared:

$$\left\lfloor \frac{67}{2^2} \right\rfloor 2^2 = 64 = (01000000)_2.$$

The secret-bit group $(01)_2$ has the decimal value

$$b_k = 1.$$

Using the embedding (16), we obtain

$$S_{2,2} = \left\lfloor \frac{67}{4} \right\rfloor 4 + 1 = 64 + 1 = 65.$$

Hence,

$$(01000011)_2 \longrightarrow (01000001)_2.$$

The two LSBs of the selected cover pixel are therefore changed from $11$ to $01$, producing the stego-pixel value $65$.

### *3.2. Extracting procedure*

The extracting procedure of the*mshEdgeGrayFT2* algorithm is explained in Figure 4. At the receiver, the extraction process follows the same pixel-selection procedure used during embedding. First, the stego image is masked by clearing the embedded bit positions:

$$S_{i,j}^{(m)} = \left\lfloor \frac{S_{i,j}}{2^m} \right\rfloor 2^m, \tag{18}$$

where $S_{i,j}$ denotes the stego pixel and $m$ represents the embedding level. The horizontal and vertical intensity differences are then calculated from the masked stego image and supplied to the same IT2FLS used during embedding. The resulting fuzzy suitability values are used to reconstruct the pixel-priority order, and the required pixels are selected accordingly.

For each selected stego pixel, the embedded secret value is recovered as

$$E_{i,j} = S_{i,j} \bmod 2^m, \tag{19}$$

where $E_{i,j}$ denotes the recovered secret value. Its $m$-bit binary representation gives the corresponding secret bits. The recovered bits are concatenated according to the reconstructed pixel-priority order to obtain the complete secret bitstream.
For example, consider a selected stego pixel with value $S_{i,j} = 65$ and $m = 2$. The embedded value is recovered as

$$E_{i,j} = 65 \bmod 2^2 = 1. \tag{20}$$

Since $m = 2$, the recovered value is represented using two binary bits:

$$1 = (01)_2.$$

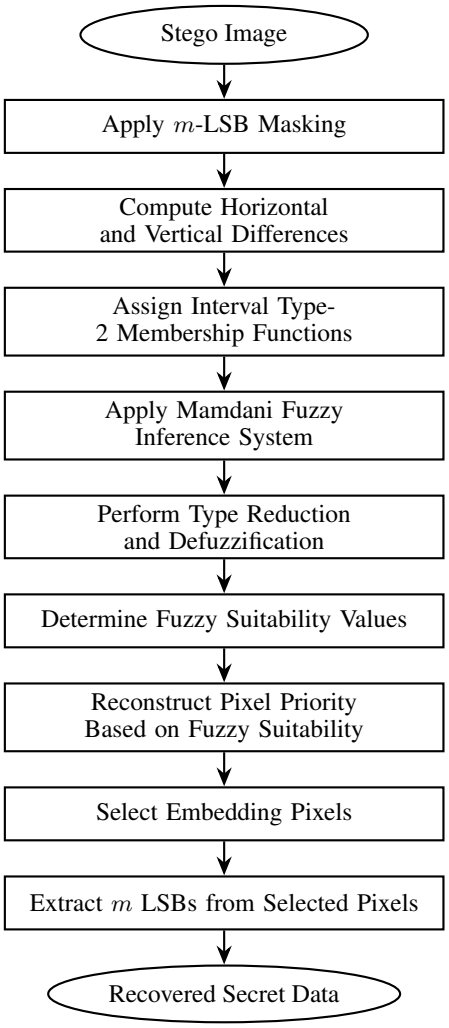



Figure 4. Extraction procedure of the proposed *mshEdgeGrayFT2* algorithm.

### *3.3. Dataset and Evaluation Metrics*

The performance of the proposed ***mshEdgeGrayFT2*** method was evaluated using five standard grayscale images, namely Lena, Baboon, Jet, Barbara, and Sailboat, each with a resolution of $512 \times 512$ pixels, as illustrated in Figure 5. The quality and imperceptibility of the generated stego images were evaluated using three widely used image-quality metrics: peak signal-to-noise ratio (PSNR), mean squared error (MSE), and structural similarity index (SSIM) [29].

The MSE measures the average squared difference between the cover and stego images and is defined as

$$MSE = \frac{1}{PQ} \sum_{p=1}^{P} \sum_{q=1}^{Q} [I_1(p,q) - I_2(p,q)]^2 , \tag{21}$$

where $P$ and $Q$ denote the number of rows and columns of the image, respectively, while $I_1(p,q)$ and $I_2(p,q)$ represent the pixel intensities of the cover and stego images.
PSNR quantifies the distortion introduced during data embedding and is calculated as



(a) (b) (c) (d) (e)

Figure 5. Cover images used for experimental evaluation.

$$PSNR = 10\log_{10}\left(\frac{I_{\max}^2}{MSE}\right), \tag{22}$$

where $I_{\max} = 255$ for an 8-bit grayscale image. A higher PSNR value indicates lower distortion between the cover and stego images.

SSIM measures the structural similarity between the cover and stego images and is expressed as

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{23}$$

where $\mu_x$ and $\mu_y$ represent the mean intensities of the cover and stego images, $\sigma_x^2$ and $\sigma_y^2$ denote their respective variances, $\sigma_{xy}$ represents their covariance, and $C_1$ and $C_2$ are stabilization constants.

### *3.4. Results and Discussion*

The proposed *mshEdgeGrayFT2* method was initially evaluated at four embedding levels, $m = 1, 2, 3$, and $4$. These levels correspond to payloads of 49,152, 98,304, 147,456, and 196,608 bits, respectively. Table 4 presents the PSNR, MSE, and SSIM values obtained for the five test images.

As the embedding level increases, the payload capacity increases because a larger number of secret bits are embedded in each selected pixel. This increase in capacity is accompanied by a gradual decrease in PSNR and an increase in MSE because additional least significant bit positions are modified. At $m = 1$, the PSNR values are approximately 58.40 dB for all five test images, while the SSIM values remain close to 1, indicating minimal structural distortion. At $m = 2$, the PSNR values remain above 51 dB, whereas at $m = 3$, they range from approximately 44.72 to 45.11 dB. At the maximum embedding level, $m = 4$, corresponding to a payload of 196,608 bits, PSNR values range from 38.2775 to 38.6480 dB, while SSIM values range from 0.954404 to 0.980696. These results demonstrate the expected trade-off between embedding capacity and stego-image quality. A bit error rate (BER) of zero was obtained for all tested images and embedding levels. Furthermore, no selection mismatch was observed between the embedding and extraction processes, indicating consistent reconstruction of the fuzzy-ranked pixel locations during data extraction.

#### *3.4.1. Steganalysis*

In addition to image-quality evaluation, the statistical detectability of the proposed *mshEdgeGrayFT2* method was examined using Chi-square and RS analyses. These experiments were performed on Lena, Baboon, and Jet images of size $256 \times 256$ pixels at four embedding levels, $m = 1, 2, 3$, and $4$, corresponding to payload rates of 0.75, 1.50, 2.25, and 3.00 bpp, respectively. The results are presented in Table 5.

The Chi-square results indicate that the statistical response varies with both the characteristics of the cover image and the embedding level. For Lena at $m = 1$, the stego image produces a $p$-value of 0.2439. For Baboon, the stego-image $p$-values are 0.03494, 0.01394, 0.07804, and 0.01028 for $m = 1$ to $m = 4$, respectively. In comparison,

Table 4. Results obtained using the proposed method ***mshEdgeGrayFT2***

| **Standard Images** | $m$ | **Payload (bits)** | **PSNR (dB)** | **MSE** | **SSIM** |
|---|---|---|---|---|---|
| Lena | 1 | 49,152 | 58.3993 | 0.094006 | 0.999790 |
| | 2 | 98,304 | 51.3404 | 0.477577 | 0.998856 |
| | 3 | 147,456 | 44.7195 | 2.193451 | 0.991116 |
| | 4 | 196,608 | 38.3885 | 9.423855 | 0.960844 |
| Baboon | 1 | 49,152 | 58.4224 | 0.093506 | 0.999890 |
| | 2 | 98,304 | 51.3963 | 0.471466 | 0.999440 |
| | 3 | 147,456 | 45.0987 | 2.010067 | 0.997545 |
| | 4 | 196,608 | 38.6480 | 8.877361 | 0.980696 |
| Jet | 1 | 49,152 | 58.3952 | 0.094093 | 0.999787 |
| | 2 | 98,304 | 51.3405 | 0.477558 | 0.998894 |
| | 3 | 147,456 | 44.8399 | 2.133499 | 0.992004 |
| | 4 | 196,608 | 38.3536 | 9.499973 | 0.957582 |
| Barbara | 1 | 49,152 | 58.4067 | 0.093845 | 0.999783 |
| | 2 | 98,304 | 51.3306 | 0.478653 | 0.998748 |
| | 3 | 147,456 | 44.7941 | 2.156086 | 0.991547 |
| | 4 | 196,608 | 38.2775 | 9.667831 | 0.954404 |
| Sailboat | 1 | 49,152 | 58.4125 | 0.093719 | 0.999882 |
| | 2 | 98,304 | 51.4403 | 0.466709 | 0.999387 |
| | 3 | 147,456 | 45.1130 | 2.003460 | 0.996670 |
| | 4 | 196,608 | 38.6048 | 8.966125 | 0.974719 |

Table 5. Chi-square and RS steganalysis results of the proposed *mshEdgeGrayFT2* method at different embedding levels.

| **Image** | $m$ | **Cover** $R/S$ | **Stego** $R/S$ | **Cover** $\chi^2$ | **Cover** $p$**-value** | **Stego** $\chi^2$ | **Stego** $p$**-value** |
|---|---|---|---|---|---|---|---|
| Lena | 1 | 0.6086/0.3914 | 0.5399/0.4601 | 199.63 | $2.73 \times 10^{-7}$ | 118.86 | 0.2439 |
| | 2 | 0.6086/0.3914 | 0.5340/0.4660 | 199.63 | $2.73 \times 10^{-7}$ | 178.83 | $3.68 \times 10^{-5}$ |
| | 3 | 0.6086/0.3914 | 0.5402/0.4598 | 199.63 | $2.73 \times 10^{-7}$ | 227.25 | $5.27 \times 10^{-10}$ |
| | 4 | 0.6086/0.3914 | 0.5413/0.4587 | 199.63 | $2.73 \times 10^{-7}$ | 190.95 | $3.65 \times 10^{-6}$ |
| Baboon | 1 | 0.5834/0.4166 | 0.5302/0.4698 | 133.15 | 0.00876 | 123.72 | 0.03494 |
| | 2 | 0.5834/0.4166 | 0.5295/0.4705 | 133.15 | 0.00876 | 130.14 | 0.01394 |
| | 3 | 0.5834/0.4166 | 0.5234/0.4766 | 133.15 | 0.00876 | 117.37 | 0.07804 |
| | 4 | 0.5834/0.4166 | 0.5256/0.4744 | 133.15 | 0.00876 | 132.13 | 0.01028 |
| Jet | 1 | 0.6625/0.3375 | 0.5759/0.4241 | 631.86 | $8.78 \times 10^{-76}$ | 497.72 | $5.08 \times 10^{-52}$ |
| | 2 | 0.6625/0.3375 | 0.5655/0.4345 | 631.86 | $8.78 \times 10^{-76}$ | 571.31 | $1.56 \times 10^{-64}$ |
| | 3 | 0.6625/0.3375 | 0.5510/0.4490 | 631.86 | $8.78 \times 10^{-76}$ | 529.04 | $4.24 \times 10^{-57}$ |
| | 4 | 0.6625/0.3375 | 0.5563/0.4437 | 631.86 | $8.78 \times 10^{-76}$ | 313.82 | $2.94 \times 10^{-21}$ |

Jet exhibits very small $p$-values for both the cover and stego images. Therefore, the Chi-square results should be interpreted relative to the statistical characteristics of the corresponding cover image rather than considering the stego-image $p$-value independently. The RS analysis also shows changes in the proportions of regular and singular groups after data embedding. For Lena, the cover-image values $R = 0.6086$ and $S = 0.3914$ change to $R = 0.5399$ and $S = 0.4601$ at $m = 1$. Similarly, the corresponding values change from $0.5834/0.4166$ to $0.5302/0.4698$ for Baboon and from $0.6625/0.3375$ to $0.5759/0.4241$ for Jet. In all three cases, the regular and singular proportions become closer after embedding, indicating a measurable statistical effect of LSB modification.

### *3.4.2. Comparison with Existing Methods*

To provide a consistent comparison with existing data-hiding methods, an additional experiment was conducted using $256 \times 256$ grayscale cover images with a fixed secret payload of 65,536 bits (1 bpp). Unlike the preceding experiment, in which the payload increased with $m$, the total payload in this comparison was maintained at 65,536 bits for all embedding levels, $m = 1, 2, 3,$ and $4$.

Accordingly, the number of selected pixels was adjusted with respect to the embedding level: 65,536 pixels for $m = 1$, 32,768 pixels for $m = 2$, approximately 21,846 pixels for $m = 3$, and 16,384 pixels for $m = 4$. The IT2FLS-based suitability measure was used to prioritize candidate pixels for embedding. The resulting PSNR and MSE values were compared with those obtained using OPAP [21], Thien [30], IP [31], and ARIP [32]. The comparative results are presented in Table 6.

Table 6. Comparison of PSNR and MSE for different methods at a fixed payload of 65,536 bits for different embedding levels ($m$)

| **Cover Image** | **Method** | $m = 1$ | | $m = 2$ | | $m = 3$ | | $m = 4$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | PSNR | MSE | PSNR | MSE | PSNR | MSE | PSNR | MSE |
| Lena | OPAP [21] | 51.1251 | 0.5018 | 46.3663 | 1.5012 | 40.7327 | 5.4930 | 34.7967 | 21.5482 |
| | Thien [30] | 51.1239 | 0.5020 | 46.3301 | 1.5138 | 40.7235 | 5.5046 | 34.7837 | 21.6129 |
| | IP [31] | 51.6171 | 0.4481 | 46.3868 | 1.4942 | 40.7395 | 5.4845 | 34.9113 | 20.9872 |
| | ARIP [32] | 51.7499 | 0.4346 | 46.6559 | 1.4044 | 40.9556 | 5.2182 | 35.1746 | 19.7523 |
| | ***mshEdgeGrayFT2*** | 51.1500 | 0.4990 | 47.0890 | 1.2710 | 42.4690 | 3.6830 | 37.3620 | 11.9380 |
| Baboon | OPAP [21] | 51.1446 | 0.4996 | 46.3876 | 1.4939 | 40.7468 | 5.4752 | 34.8125 | 21.4698 |
| | Thien [30] | 51.1463 | 0.4994 | 46.3841 | 1.4951 | 40.7183 | 5.5112 | 34.8178 | 21.4437 |
| | IP [31] | 51.5622 | 0.4538 | 46.4376 | 1.4768 | 40.7651 | 5.4522 | 34.8005 | 21.5295 |
| | ARIP [32] | 51.6414 | 0.4456 | 46.6910 | 1.3931 | 40.9816 | 5.1871 | 35.0359 | 20.3934 |
| | ***mshEdgeGrayFT2*** | 53.9960 | 0.2591 | 49.9520 | 0.6575 | 45.1550 | 1.9840 | 39.8210 | 6.7756 |
| Jet | OPAP [21] | 51.1350 | 0.5007 | 46.3601 | 1.5034 | 40.8081 | 5.3985 | 34.9874 | 20.6223 |
| | Thien [30] | 51.1333 | 0.5009 | 46.3701 | 1.4999 | 40.8053 | 5.4019 | 34.9465 | 20.8175 |
| | IP [31] | 51.1376 | 0.5004 | 46.3899 | 1.4931 | 40.9283 | 5.2511 | 35.0635 | 20.2644 |
| | ARIP [32] | 51.6501 | 0.4447 | 46.7088 | 1.3874 | 41.0139 | 5.1486 | 35.5117 | 18.2774 |
| | ***mshEdgeGrayFT2*** | 51.1380 | 0.5004 | 47.0300 | 1.2885 | 42.4700 | 3.6823 | 37.4110 | 11.8020 |

For Lena at $m = 1$, the proposed method achieves a PSNR of 51.1500 dB and an MSE of 0.4990. Although this result is slightly better than those obtained using OPAP and Thien, IP and ARIP achieve PSNR values of 51.6171 and 51.7499 dB, respectively. From $m = 2$ onward, however, the proposed method produces higher PSNR and lower MSE than all four comparison methods. At $m = 4$, the proposed method achieves a PSNR of 37.3620 dB with an MSE of 11.9380, compared with 35.1746 dB and 19.7523, respectively, for ARIP. For Baboon, the proposed method provides higher PSNR and lower MSE than all four compared methods at each tested embedding level. The difference becomes more pronounced at higher values of $m$. At $m = 4$, the proposed method achieves a PSNR of 39.8210 dB and an MSE of 6.7756, whereas ARIP achieves 35.0359 dB and 20.3934, respectively. For Jet at $m = 1$, the PSNR of the proposed method is 51.1380 dB, which is comparable to those obtained using OPAP, Thien, and IP but lower than the 51.6501 dB obtained using ARIP. At $m = 2$, $m = 3$, and $m = 4$, however, the proposed method achieves higher PSNR and lower MSE than all four comparison methods. At $m = 4$, the proposed method

obtains a PSNR of 37.4110 dB and an MSE of 11.8020, compared with 35.5117 dB and 18.2774 for ARIP. The comparative results indicate that the effectiveness of the IT2FLS-guided pixel-selection strategy depends on both image characteristics and embedding level. At $m = 1$, some existing methods provide comparable or slightly better distortion performance for Lena and Jet. However, the proposed method demonstrates increasingly favorable PSNR and MSE values at higher embedding levels. These observations suggest that fuzzy-guided prioritization of suitable pixels is particularly beneficial when multiple LSB positions are used while maintaining a fixed secret payload.

## 4. Conclusion

This paper presented ***mshEdgeGrayFT2***, an image steganography method that combines Interval Type-2 Fuzzy Logic with LSB embedding for fuzzy-guided pixel selection. Experimental results demonstrated a trade-off between payload capacity and image quality, with payloads up to 196,608 bits while maintaining high structural similarity. Zero BER and no pixel-selection mismatch confirmed accurate data extraction. The fixed-payload comparison showed favorable PSNR and MSE performance at higher embedding levels, while Chi-square and RS analyses indicated that statistical detectability depends on image characteristics and embedding level. Overall, the results demonstrate the potential of IT2FLS-guided pixel selection for balancing embedding capacity, imperceptibility, and reliable data recovery.

## Conflicts of interest

The authors have no conflict of interest in any aspect.

REFERENCES


1. S. Dhawan, H.K. Bhuyan, S.K. Pani, V. Ravi, R. Gupta, A. Rana and A. Al Mazroa, *Secure and resilient improved image steganography using hybrid fuzzy neural network with fuzzy logic*, Journal of Safety Science and Resilience, vol. 5, no. 1, pp. 91–101, 2024.
2. A. Ioannidou, S.T. Halkidis and G. Stephanides, *A novel technique for image steganography based on a high payload method and edge detection*, Expert Systems with Applications, vol. 39, no. 14, pp. 11517–11524, 2012.
3. G. Swain and S.K. Lenka, *Classification of image steganography techniques in spatial domain: a study*, Journal of Computer Science & Engineering Technology (IJCSET), vol. 5, no. 3, pp. 219–232, 2014.
4. B. Song, P. Wei, S. Wu, Y. Lin and W. Zhou, *A survey on Deep-Learning-based image steganography*, Expert Systems with Applications, pp. 124390, 2024.
5. C.P. Sumathi, T. Santanam and G. Umamaheswari, *A study of various steganographic techniques used for information hiding*, arXiv preprint arXiv:1401.5561, 2014.
6. H.S. Yusuf and H. Hagras, *A novel payload image steganography using type-2 fuzzy logic and edge detection*, 2021.
7. L.A. Zadeh, *Fuzzy sets*, Information and Control, vol. 8, no. 3, pp. 338–353, 1965.
8. J.A. Goguen, *L-fuzzy sets*, Journal of Mathematical Analysis and Applications, vol. 18, no. 1, pp. 145–174, 1967.
9. T. Fujita, I. M. Batiha, M. Gulistan, B. Batiha, E. L. Al Smadi, and I. Jebril, "Tree Decompositions in Fuzzy Graphs: Foundations and Structural Insights," *Statistics, Optimization & Information Computing*, vol. 16, no. 1, pp. 312–326, 2026. doi: 10.19139/soic-2310-5070-3117.
10. E.H. Mamdani and N. Baaklini, *Prescriptive method for deriving control policy in a fuzzy-logic controller*, Electronics Letters, vol. 11, no. 25–26, pp. 625–626, 1975.
11. J.C. Bezdek, R. Ehrlich and W. Full, *FCM: The fuzzy c-means clustering algorithm*, Computers & Geosciences, vol. 10, no. 2–3, pp. 191–203, 1984.
12. L.A. Zadeh, *The concept of a linguistic variable and its application to approximate reasoning—I*, Information Sciences, vol. 8, no. 3, pp. 199–249, 1975.
13. N.N. Karnik and J.M. Mendel, *Introduction to type-2 fuzzy logic systems*, Proceedings of the 1998 IEEE International Conference on Fuzzy Systems, vol. 2, pp. 915–920, 1998.
14. Q. Liang and J.M. Mendel, *Interval type-2 fuzzy logic systems: theory and design*, IEEE Transactions on Fuzzy Systems, vol. 8, no. 5, pp. 535–550, 2000.
15. J.M. Mendel, R.I. John and F. Liu, *Interval type-2 fuzzy logic systems made simple*, IEEE Transactions on Fuzzy Systems, vol. 14, no. 6, pp. 808–821, 2006.
16. T. Fujita, A. Salem Heilat, and R. Hatamleh, "Upside-Down Logic in Plithogenic Fuzzy Control System: Contradiction-Aware Rule Inversion and De-Plithogenication," *Statistics, Optimization & Information Computing*, vol. 15, no. 1, pp. 877–920, 2026.

17. G.M. Méndez, P.N. Montes Dorantes and A. Mexicano Santoyo, *Interval type-2 fuzzy logic systems optimized by central composite design to create a simplified fuzzy rule base in image processing for quality control application*, The International Journal of Advanced Manufacturing Technology, vol. 102, no. 9, pp. 3757–3766, 2019.
18. V. Yadav, M. Hooda, N. Dhaka and S. Gill, *Comparative analysis of mshEdgeGrayFT2 and Laplacian of Gaussian Method for Edge Detection in Grayscale Images*, Journal of Computational Analysis and Application, vol. 33, no. 2, pp. 501–507, 2024.
19. V. Yadav, M. Hooda, and N. Dhaka, "A Novel Steganography Algorithm based on Interval Type-1 Fuzzy Logic System," *Statistics, Optimization & Information Computing*, vol. 14, no. 3, pp. 1512–1524, 2025.
20. R.Z. 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.
21. 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.
22. R. Demirci, *Similarity relation matrix-based color edge detection*, AEU-International Journal of Electronics and Communications, vol. 61, no. 7, pp. 469–477, 2007.
23. R. Karakış, I. Güler, I. Capraz and E. Bilir, *A novel fuzzy logic-based image steganography method to ensure medical data security*, Computers in Biology and Medicine, vol. 67, pp. 172–183, 2015.
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. R. John, *Type 2 fuzzy sets: an appraisal of theory and applications*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 6, no. 6, pp. 563–576, 1998.
26. J. Mendel and D. Wu, *Perceptual computing: aiding people in making subjective judgments*, John Wiley & Sons, 2010.
27. R. Bhatnagar, M.K. Ghose and V. Bhattacharjee, *Selection of defuzzification method for predicting the early stage software development effort using Mamdani FIS*, International Conference on Computational Intelligence and Information Technology, pp. 375–381, 2011.
28. P.K. Muhuri, Z. Ashraf and Q.M.D. Lohani, *Multiobjective reliability redundancy allocation problem with interval type-2 fuzzy uncertainty*, IEEE Transactions on Fuzzy Systems, vol. 26, no. 3, pp. 1339–1355, 2017.
29. N. Dhaka, M. Hooda, V. Yadav and S. Gill, *A novel RGB Image steganography algorithm using type-1 fuzzy logic*, Indonesian Journal of Electrical Engineering and Computer Science, vol. 37, no. 1, pp. 123–133, 2025.
30. C.C. Thien and J. Chen, *A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function*, Pattern Recognition, vol. 36, no. 12, pp. 2875–2881, 2003.
31. 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.
32. 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.