A novel RGB image steganography algorithm using type-1 fuzzy logic

Navita Dhaka, Meenakshi Hooda, Vinita Yadav, Sumeet Gill
Department of Mathematics, Maharshi Dayanand University Rohtak, Haryana, India

Article Info

Article history:

Received Jun 26, 2024
Revised Sep 12, 2024
Accepted Sep 29, 2024

Keywords:

Edge detection
Fuzzy type-1
LSB method
PSNR
RGB image

ABSTRACT

Steganography aims to conceal secret data within images without affecting image quality. Traditional methods often struggle with balancing simplicity, effectiveness and payload capacity while maintaining imperceptibility. Proposed algorithm: the paper proposed a novel steganographic mshEdgeRGB_T1 algorithm that combines Mamdani fuzzy type-1 logic with the least significant bit (LSB) method. The LSB method is chosen for its simplicity and effectiveness in hiding messages. The mshEdgeRGB_T1 algorithm focuses on embedding secret messages in edge pixels, detecting more edge pixels compared to other methods, thus increasing payload capacity. Findings: the algorithm's performance is evaluated using metrics such as peak signal-to-noise ratio (PSNR), mean squared error (MSE) and histogram analysis to measure the similarity between the cover and Stego images, quantifying the level of imperceptibility. Experimental analysis demonstrates that the mshEdgeRGB_T1 algorithm offers improved payload capacity, enhanced security and reduced imperceptibility compared to many existing methods. Conclusion: the proposed mshEdgeRGB_T1 algorithm effectively balances simplicity, payload capacity and image quality, making it a better use for image steganography.

Corresponding Author:

Meenakshi Hooda
Department of Mathematics, Maharshi Dayanand University
124001, Rohtak, Haryana, India
Email: meenakshi.maths@mdurohtak.ac.in

1. INTRODUCTION

In the modern world, due to the increasing developments in communication and internet technology, transmitting confidential information securely presents a significant challenge. Before transmission over public channels, confidential information is well-defined and formatted [1]. Well-formed refers to the data being converted from one form to another so that only the intended recipient can understand it [2]. There are mainly three techniques used for this purpose: cryptography, steganography and watermarking, which ensure the security, quality and capacity of the transmitted message.

Cryptography and steganography are two major branches of information security [3]. Compared to cryptography, steganography is considered a more secure method. The term steganography originates from the combination of two Greek words: Steganos which means to covered and graphie which means to writing [4]. Cryptography can change the version of a message, such as converting a text message into binary form, which can be decrypted by the intended recipient only [5]. Steganography is the art and science of hiding data in such a way that only the sender and receiver can extract the information. Steganography uses audio, image, text and video files mainly, which are referred to as Stego objects and are used to embed and transfer secret data [6].

Edge detection techniques are used to extract information about the image, such as shape, area and colour intensity, which play an important role in steganography to select areas for embedding the secret message [5]. Edge detection is an essential tool in the field of digital image processing [7]. The human eye is susceptible to finding areas for embedding secret messages, such as smooth-edge portions, changes in color intensity and changes in image brightness [8].

Image steganography techniques primarily have three basic requirements: payload capacity, imperceptibility and similarity index. The payload capacity refers to the number of secret message bits that can be hidden in the cover image without causing noticeable distortion. Imperceptibility is measured by evaluation parameters such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE). The similarity index is represented by the histogram of the cover and Stego images, which shows the frequency of the pixels in ten sites. The proposed algorithm has been evaluated for all metrics mentioned above.

Neamah *et al.* [1] proposed the least significant bit (LSB) steganography method to hide secret data in colour images. An encryption key and the XNOR gate are employed to encrypt the secret information. Wu and Tsai [2] focused on grayscale images and introduced a new reverse steganography method in image steganography, which yields a better similarity index and higher payload capacity. Yusuf and Hagras [5] proposed a new steganography technique by combining a fuzzy system and the Canny operator. Melin *et al.* [8] suggested no filter is needed to find edges in an image. A new edge detection technique was proposed, which relied on gradient values and fuzzy set theory. Nadernejad *et al.* [9] proposed a novel method for detecting the edges of any colored object, which operates on Euclidean distance. Comparative analysis is conducted between the proposed edge detection methods and the Canny operator, mainly aimed at overcoming difficulties in multiflash edge detection. Yuvaraja and Sabeenian [10] used the DWT technique to normalize colour images, both horizontally and vertically, as well as diagonally. The fuzzy type1 is utilized for hiding information. Tang *et al.* [11] introduced a chaotic method for embedding secret messages. A new fuzzy inference system (FIS) was designed to work with LSB and embedding bit optimization, aiming to improve payload capacity and imperceptibility. Tang *et al.* [12] focused on the LSB steganography method, comparing PSNR and MSE values for 1-LSB and 4-LSB in color images. Al-Faydi *et al.* [13] proposed a new steganography method based on LSB and MSB techniques, utilizing the LZW technique for compressing sensitive data in digital images. Muhammad *et al.* [14] presented a new steganography method referred to as SKA-LSB, where SKA stands for Stego key algorithms and LSB stands for LSB. These 2 algorithms employed two types of LSB, namely two-level LSB and Multi-level LSB, to achieve better similarity between the original image and the Stego image.

Recently, a new image steganography technique has been proposed for embedding secret data into digital images. This technique is based on a fuzzy edge detector and the LSB method. The LSB and MSB methods for embedding secret messages have been discussed by Al-Faydi *et al.* [13]. Motivated by the work done in this field, the present research proposed mshEdgeRGB_T1 algorithm based on fuzzy sets and the LSB method, which satisfies all three basic requirements of image steganography. To improve security, edge pixels are used to hide the secret data. A fuzzy type-1 system is employed to find more edge pixels than traditional edge operators to increase embedding capacity. Lastly, for the similarity index, the LSB method is used to hide the secret message in the cover image, resulting in a Stego image identical to the cover image.

The paper is organized as follows: in section 2, we propose the mshEdgeRGB_T1 algorithm to embed and extract the secret data from RGB images based on fuzzy type-1 and the LSB method. In section 3 covers the LSB method and the FIS. In section 4, we analyzed the similarity and quality between the cover and Stego images. Also shows a comparative analysis between the total number of edge pixels obtained from mshEdgeRGB_T1 algorithms and traditional operators like Prewitt, Sobel, Canny, and LoG operators. The last section 5 concludes the paper.

2. PROPOSED ALGORITHMS

In this section, we proposed mshEdgeRGB_T1 embedding and extracting algorithms that combine the simplicity of the LSB method with the robustness of a type-1 Mamdani FIS to embed and extract secret messages in RGB images. Algorithm 1 represents the embedding process which starts with reading the cover image and extracting the pixel intensity values for the RED, GREEN, and BLUE channels separately. A mask is applied to each channel and the differences between neighboring pixels in both horizontal and vertical directions are calculated. A FIS is then created and initialized with input and output variables, and fuzzy rules are defined to capture the relationships between inputs and outputs. The image data undergoes fuzzification and defuzzification to obtain fuzzyEdgeOut, which helps identify edge pixels. The secret message is then embedded within these edge pixels using the LSB and XOR methods, resulting in the Stego image.

Algorithm 1. mshEdgeRGB_T1 embedding algorithm

```
Input   RGB Cover image, secret message.
Output  RGB Stego image.
Step 1  Read Cover image.
Step 2  Extract pixel intensity values of Cover image for all three channels separately
        (RED, GREEN, and BLUE).
Step 3  Mask on cover image for each channel.
Step 4  Find the difference of two neighbouring pixels in the horizontal and vertical
        directions for each channel separately.
Step 5  Create FIS and initialize its input and output variables.
Step 6  Set the rules for FIS.
Step 7  Apply fuzzification to the image data and defuzzification to the intermediate
        results to obtain fuzzyEdgeOut.
Step 8  Use the LSB and XOR method to hide secret message within edge pixels of Cover image
        based on fuzzyEdgeOut and obtain Stego image.
```

Algorithm 2 represents the mshEdgeRGB_T1 extracting algorithm, in the extraction process the Stego image is read and pixel intensity values for the three channels are extracted. The differences between neighboring pixels are calculated again and the FIS is worked with the same input and output variables and applies the same fuzzy rules. The data undergoes fuzzification and defuzzification to regenerate fuzzyEdgeOut. The secret message is finally extracted from the edge pixels of the Stego image using the modulus method based on fuzzyEdgeOut. This approach ensures a high level of security and imperceptibility, making it difficult for the human eye to detect differences between the cover and Stego images.

Algorithm 2. mshEdgeRGB_T1 extracting algorithm

```
Input   RGB Stego image.
Output  RGB Cover image, secret message.
Step 1  Read Stego image.
Step 2  Extract pixel intensity values of Stego image for all three channels separately
        (RED, GREEN, and BLUE).
Step 3  Find difference of two neighbouring pixels in the horizontal and vertical directions
        for each channel.
Step 4  Create a FIS and initialize its input and output variables.
Step 5  Set the rules for FIS.
Step 6  Apply fuzzification to the image data and defuzzification to the intermediate
        results to obtain fuzzyEdgeOut.
Step 7  Use the modulus method to extract the secret message from the edge pixels of Stego
        image using fuzzyEdgeOut.
```

3. METHOD

This section describes the specific procedures and techniques used in this work to address the research questions and hypotheses raised in section 1. The mshEdgeRGB_T1 algorithm for RGB image steganography which are described in section 2 based on type-1 fuzzy logic system and LSB method.

3.1. Materials

In this paper, we worked on a standard PC with 10 GB RAM using MATLAB (version R2015a) for experimental analysis of the different sizes of images sourced from Google Photos.

3.2. Method and technique

3.2.1. LSB method

Image steganography involves two major domains: transform and the spatial domain. In spatial domain, the LSB technique is a simple method for hiding secret messages. This technique involves embedding the confidential message within an image's least significant pixel bits [15]. Following LSB algorithms make it easy to hide secret data and the implementation process becomes straightforward. In this method, message bits are hidden in every pixel's LSB value of the cover image. For example, 1 LSB procedure is illustrated as follows: cover image pixel=01010010 00110101 01000101 10100110, message bits=1010, Stego image pixels=01010011 00110100 01000101 10100110. Detecting the difference between cover and Stego image using the RGBdetectfuzzyLSB algorithm can pose a challenge to the human eye. Researchers might utilize a greater number of LSB bits per pixel to augment payload capacity and bolster security [16].

3.2.2. Fuzzy inference system

Lofti Zadeh first introduced the concept of fuzzy sets in the twentieth century. Fuzzy set theory is a mathematical model that addresses problems related to uncertainty and imprecision in data [17]. In such sets, linguistic variables represent input and output values and fuzzy sets define membership functions that determine the degree of membership [0-1] for each input value. In this paper, we use two input variables to determine a single crisp output value [18]. This research work applies four fuzzy if-then rules, where the antecedent and consequent parts contain fuzzy propositions, to capture expert knowledge or empirical relationships between inputs and outputs [19]. Three types of membership functions are used to represent the membership value of any fuzzy set: triangular, trapezoidal, and gaussian membership functions.

Figure 1 depicts the input and output variables initialized for FIS for edge detection. Input 1 represents the difference between horizontal pixels, labeled as differencePixelsColumnWise and Input 2 represents the difference between pixels vertically, labeled as differencePixelsRowWise. Two membership functions low and high are associated with each input variable using triangular member functions, which define the degree of membership of different input values. On the other hand, three membership functions black, gray, and white are associated with the fuzzy output variable fuzzyEdgeOut using triangular member function.

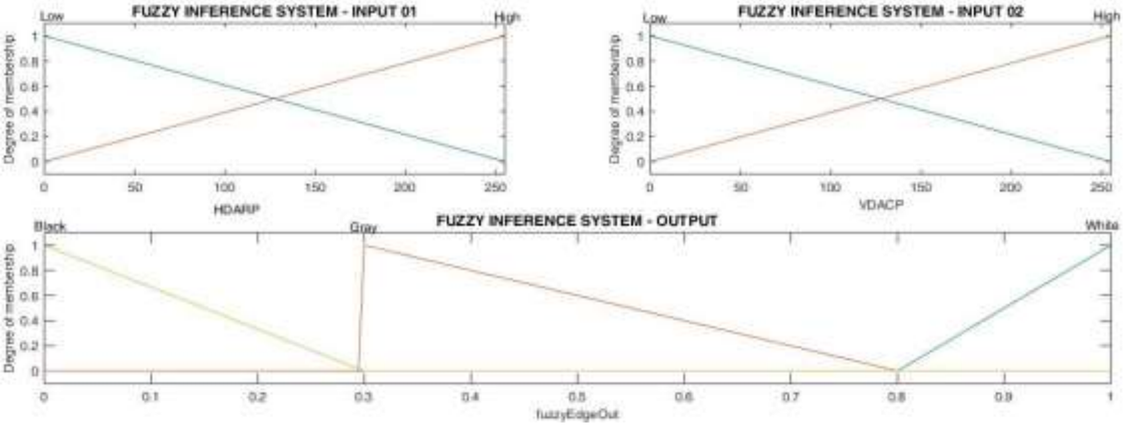


Figure 1. Input and output variables for FIS

3.3. Procedure

In this step, we give an example that explains the proposed algorithms step by step. Example of mshEdgeRGB_T1 algorithm is:

Step 1: let us take a sample block of size 6×6 from the RGB cover image (Lena) of size 256×256 pixels [20]. Separate different color intensities red, green, and blue in three different matrices. D_r where $1 \leq r \leq 3$. All pixel intensities fall in the range [0-255].

Figure 2 shows the intensity values for all three channels red(R), green(G), and blue(B) of sample block. $D_{r,x,y}$ represents the intensity values of x^{th} row and y^{th} column of r^{th} channel, where $1 \leq x, y \leq 256$. Let us take the intensity values given below from the RED channel as an example, $D_{1,2,3} = 174$, $D_{1,2,4} = 146$, $D_{1,3,3} = 134$.

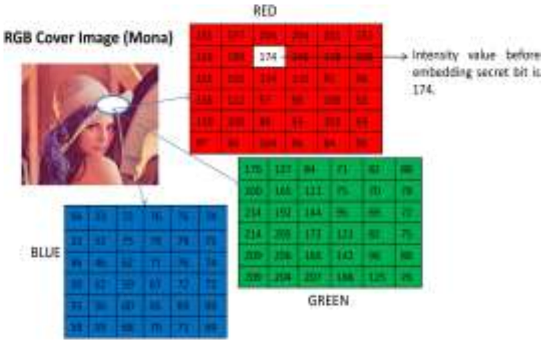


Figure 2. Sample block of Lena cover image

Step 2: mask last four bits of all pixels for all three channels using (1).

$$D'_{r,x,y} = \text{int16}\left(\frac{D_{r,x,y}}{4}\right) \times 4 \quad (1)$$

Now, the masked image pixels are $D'_{1,2,3}$=172, $D_{1,2,4}$=144, $D_{1,3,3}$=132
Step 3: calculate horizontal difference between adjacent row pixels (HDARP) of masked image as,

$$D''_{r,x,y-1} = \text{mod } [D'_{r,x,y-1} - D'_{r,x,y}] \quad (2)$$

the HDARP values which are obtained by using (2), on masked image pixels are as,

$$D''_{1,2,3} = \text{mod } [D'_{1,2,4} - D'_{1,2,3}] = \text{mod } (144 - 172) = 28$$

Step 4: calculate the vertical difference between adjacent column pixels (VDACP) of masked images as,

$$D'''_{r,x-1,y} = \text{mod } [D'_{r,x,y} - D'_{r,x-1,y}] \quad (3)$$

from (3), the obtained VDACP values are,

$$D'''_{1,2,3} = \text{mod } [D'_{1,3,3} - D'_{1,2,3}] = \text{mod } (132 - 172) = 40$$

Step 5: now, we need to identify the edge pixels using FIS. Here, we have used Mamdani FIS for type-1 fuzzy systems. HDARP and VDACP provide the crisp inputs to the system and each input is associated with two triangular-shaped membership functions low (L) and high (H). The vector parameter related to the triangular membership function (trimf) is initialized in the form [p, q, r], where p and r define the extent of the membership function and q defines its peak. This function returns fuzzy membership value for y in fuzzy set z using (4).

$$\mu_Z(y) = \begin{cases} 0, & y \leq p \\ \frac{y-p}{y-q}, & p \leq y \leq q \\ \frac{r-y}{r-q}, & q \leq y \leq r \\ 0, & y \geq r \end{cases} \quad (4)$$

Low (L) triangular membership function is initialized with vector parameters [0 0 255], while high (H) triangular membership function is initialized with vector parameter [0 255 255]. On evaluating trimf function, we get the degree of membership for HDARP and VDACP input values, with which they are associated with the low (L) and high (H) fuzzy sets. The degree of membership lies in the range [0 - 1]. Here,

μ_{Low} (HDARP): membership degree of HDARP values within the fuzzy set labeled as low.
µHigh (HDARP): membership degree of HDARP values within fuzzy set labeled as high.
µLow (VDACP): membership degree of VDACP values within fuzzy set labeled as low.
µHigh (VDACP): membership degree of VDACP values within fuzzy set labeled as high.

By using (4), we get,

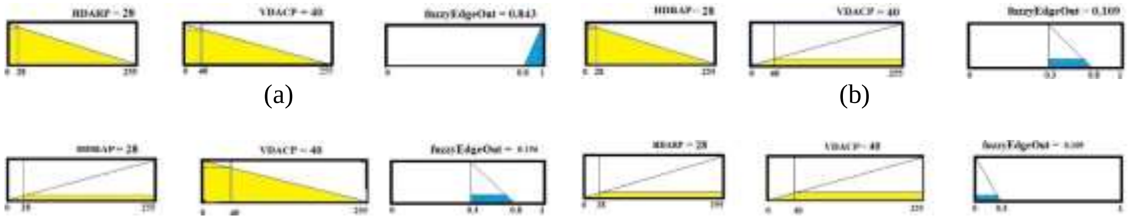
$$\mu\text{Low } (D''_{1,2,3}) = \frac{255 - D''_{1,2,3}}{255} = \frac{255-28}{255} = 0.890$$
$$\mu\text{High } (D''_{1,2,3}) = \frac{0 - D''_{1,2,3}}{255} = \frac{0-28}{255} = 0.109$$
$$\mu\text{Low } (D'''_{1,2,3}) = \frac{255 - D'''_{1,2,3}}{255} = \frac{255-40}{255} = 0.843$$
$$\mu\text{High } (D'''_{1,2,3}) = \frac{0 - D'''_{1,2,3}}{255} = \frac{0-40}{255} = 0.156$$

After evaluation for given values, we observe that $D''_{1,2,3}$ belongs to fuzzy set R_L with 0.890 degrees of membership. Similarly, other results can be analyzed by observing their degree of membership in different fuzzy sets. Similarly, the output fuzzyEdgeOut is associated with three trimf functions black, gray, and white. The vector parameters initialized for black, gray, and white triangular membership function are [0 0 0.3], [0.3 0.3 0.8], and [0.8 1 1] respectively.

Step 6: define fuzzy rules to categorize the fuzzy inputs into different fuzzy outputs. Here, we have defined four fuzzy rules.

Figure 3 demonstrates the fuzzy logic rules applied for edge detection based on (HDARP) and (VDACP) values. Each subfigure illustrates a specific rule's output.

- Figure 3(a) Rule 1: if HDARP is low and VDACP is low, then fuzzyEdgeOut is white.
- Figure 3(b) Rule 2: if HDARP is low and VDACP is high, then fuzzyEdgeOut is gray.
- Figure 3(c) Rule 3: if HDARP is high and VDACP is low, then fuzzyEdgeOut is gray.
- Figure 3(d) Rule 4: if HDARPP is high and VDACP is high, then fuzzyEdgeOut is black.



(a)

(b)

(c)

(d)

Figure 3. Fuzzy logic rules for edge detection output; (a) Rule 1- fuzzyEdgeOut is white, (b) Rule 2- fuzzyEdgeOut is gray, (c) Rule 3- fuzzyEdgeOut is gray, and (d) Rule 4- fuzzyEdgeOut is black

Fuzzy rules are evaluated in Figures 3(a) to 3(d) and their implications are performed through fuzzy set operations. These operations employ the maximum for OR and minimum for AND operators. Here, we have combined the individual results using minimum operation.

- Rule 1: min ($\mu_{\text{Low}}D''_{1,2,3}$), $\mu_{\text{Low}}D'''_{1,2,3}$) = min (0.890,0.843) = 0.843.
- Rule 2: min ($\mu_{\text{Low}}D''_{1,2,3}$, $\mu_{\text{High}}D'''_{1,2,3}$) = min (0.890,0.0.156) = 0.156.
- Rule 3: min ($\mu_{\text{High}}D''_{1,2,3}$, $\mu_{\text{Low}}D'''_{1,2,3}$) = min (0.109,0.843) = 0.109.
- Rule 4: min ($\mu_{\text{High}}D''_{1,2,3}$, $\mu_{\text{High}}D'''_{1,2,3}$) = min (0.109,0.156) = 0.109.

In a FIS, decisions are made by evaluating all rules, the rule outputs must be combined in same manner. Aggregation refers to the merging of fuzzy sets representing each rule's output into a singular fuzzy set. As a result of the aggregation process, a single fuzzy set is produced for each output variable. Figure 4 illustrates the outcome of this aggregate process, showcasing the fusion of all four fuzzy sets derived from the evaluation of four fuzzy rules.

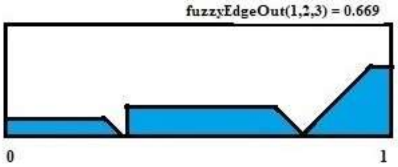


Figure 4. Aggregation of fuzzy outputs

The defuzzification process takes the aggregate fuzzy output set as input, producing a single crisp value as the output. We have used center of sum (COS) method for defuzzification's, which returns the center of the area under the aggregated fuzzy set. According to COS which is defined in (5), the crisp value will be changed.

$$x* = \frac{\sum_{i=1}^{256} A_i.\bar{X}_i}{\sum_{i=1}^{256} A_i} \tag{5}$$

Here, A_i Is the area of the region bounded by the fuzzy set got after evaluating i$^{\text{th}}$ fuzzy rule and X_i is the geometric center of that area. After defuzzification, the final crisp value that we got is 0.669 which is within 9 the defined threshold values [0 0.8], so, the current pixel is marked as an edge pixel. The same steps are repeated for all pixels of every channel to mark edge pixels.

Step 7: for embedding process let us take the secret message in binary form 01000100101 and hide one of its msgbit (0) in the identified edge pixel using (6).

$$Sr,i,j = Gr,i,j + XOR\ (mod(Gr,i,j,2),msgbit) \quad (6)$$

Where, $G_{r,i,j}$ is marked edge pixel, msgbit is the secret bit to embed and $S_{r,i,j}$ is corresponding pixel of Stego image. In, our example, let us take $G_{1,2,3}$ edge pixels to embed secret message bit. Then, using (6).

$S_{1,2,3}$ = $G_{1,2,3}$ + XOR (mod ($G_{1,2,3}$, 2), 0)

$S_{1,2,3}$ = 174 + XOR (mod (174, 2), 1) = 174 + 1 = 175.

Finally, the receiver gets a Stego image containing the secret data, Figure 5 shows the sample block of RGB Stego image.

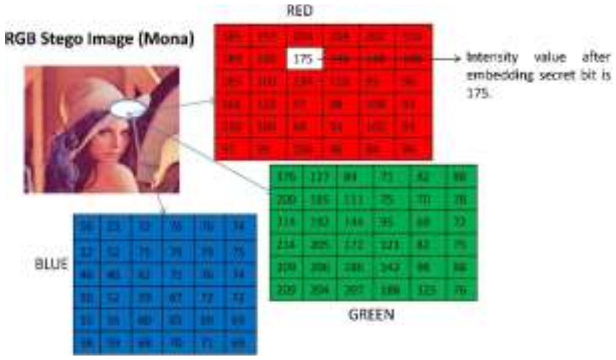


Figure 5. Sample block of Lena Stego image

Extracting process: on the receiving side, the receiver extracts the secret message from the Stego image using mshEdgeRGBT_1 extracting algorithm. Steps 1 to step 6, to find edge pixels are similar to those in the embedding algorithm. To extract the secret message bit, use the modulus function on the Stego image pixel $S_{r,i,j}$.

SecretMessageBit = mod ($S_{r,i,j}$, 2) (7)

Using (7), SecretMessageBit = mod (175,2)=1.
Similarly, extract all secret message bits, concatenate them to get the secret message in binary form, convert them into ASCII format and finally, get the secret message in readable form.

4. EXPERIMENTAL RESULTS

In the present section, mshEdgeRGB_T1 algorithm has been tested on five color images (Mona, Rose, Parrot, Lena, and Baboon) [21]-[24] using MATLAB software to demonstrate its results. In this paper, both quantitative and qualitative parameters have been used to evaluate mshEdgeRGB_T1 algorithm. The mean square error (MSE) and peak signal to noise ratio (PSNR) were used to compare the cover and Stego images quantitatively, while histograms were utilized for qualitative analysis to show the similarity between cover and Stego images.

4.1. Mean square error

MSE parameter defines the pixel-to-pixel difference between the Cover and Stego image. It can be calculated as:

$$\text{MSE} = \frac{\sum_{n=1}^{P}\sum_{m=1}^{Q}(U(n,m)-V(n,m))^2}{P\times Q}$$

where U (n, m) and V (n, m) are the corresponding pixels value of Cover and Stego image at position (n, m). P and Q are the dimensions of the images. Lower the MSE value, the higher the quality of Stego image [25].

4.2. Peak signal to noise ratio

Another parameter PSNR is used to measure the quality of Stego image. It works inversely proportional to the MSE [26]. It is calculated as:

$$\text{PSNR} = \frac{10log_{10}(MAX)^2}{MSE}$$

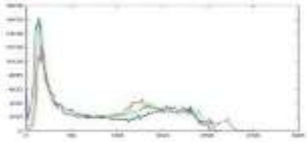
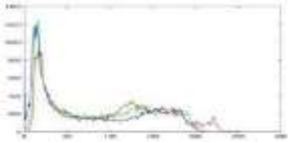
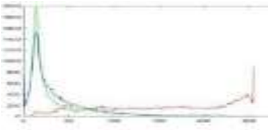
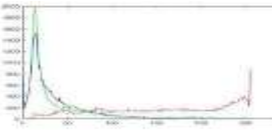
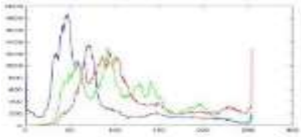
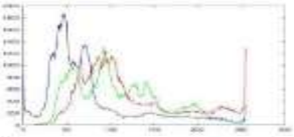
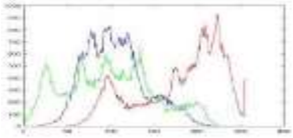
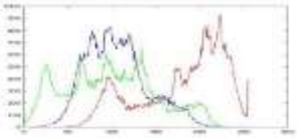
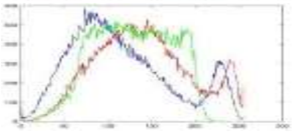
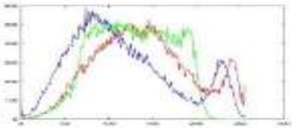
The experimental result indicates that a lower MSE value corresponds to a higher quality Stego image, as shown in Table 1. For example, when the message length is 1496 bits, the PSNR value for the Mona image is 71.332, which decreases to 68.796 when the message length is increased to 2,592 bits. Similarly, the MSE value increases from 0.0048 to 0.0086 for the same image as the message length increases.

Table 1. Evaluation parameter with different message lengths

Image name	Message length (bits)=1496		Message length (bits)=2592	
	PSNR value	MSE value	PSNR value	MSE value
Mona	71.332	0.0048	68.796	0.0086
Rose	69.933	0.0066	67.714	0.011
Parrot	73.584	0.0028	71.314	0.0048
Lena	72.298	0.0038	69.867	0.0067
Baboon	72.364	0.0038	69.963	0.0066

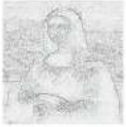
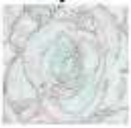
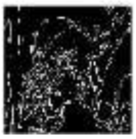
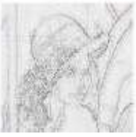
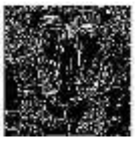
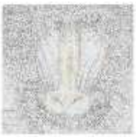
While embedding the secret message in a cover image, the pixel intensities must be least affected so that the changes remain undetectable to human eyes. The experimental analysis shows that the distribution of pixels over all three channels red, green, and blue in Stego images remains almost similar to that of cover images which is shown in Table 2, which makes the proposed algorithms mshEdgeRGB_T1 more trustworthy and robust.

Table 2. Distribution of pixel intensities for R, G, and B channels

Image name	Cover image	Cover image (Distribution of pixel intensities)	Stego image	Stego image (Distribution of pixel intensities)
Mona				
Rose				
Parrot				
Lena				
Baboon				

In another experiment, researchers compared the mshEdgeRGB_T1 algorithm with other edge detectors like Prewitt, Sobel, Canny, and LOG to analyse identified edge pixels which are shown in Table 3 and Figure 6. As depicted in Table 3, for Mona image, where Prewitt detected 2326 edge pixels, Canny detected 7,426 edge pixels, proposed mshEdgeRGB_T1 algorithm detected 12149811 edge pixels, which are sixty times more than Prewitt and twenty times more than Canny edge detector. So, it increased the embedding capacity image many times as compared to other operators. Figure 6 shows compare the edge pixel counts produced by different edge detection methods. In Figure 6, the images are displayed on the X-axis and the methods on the Y-axis, with each image color-coded for clarity. It effectively illustrates how each method performs across different images, emphasizing the differences in edge detection results and the overall effectiveness of the MshEdgeRGB_T1 algorithm.

Table 3. Number of edge pixels identified using different methods

Image name	Cover image	Edge detector operator				mshEdgeRGB_T1 algorithm
		Prewitt	Sobel	Canny	Log	
Mona		 2326	 2346	 7427	 5315	 149811
Rose		 2406	 2415	 8429	 5680	 113940
Parrot		 3632	 3581	 14854	 9278	 293973
Lena		 3233	 3223	 10681	 7289	 196469
Baboon		 3522	 3515	 19425	 11956	 200274

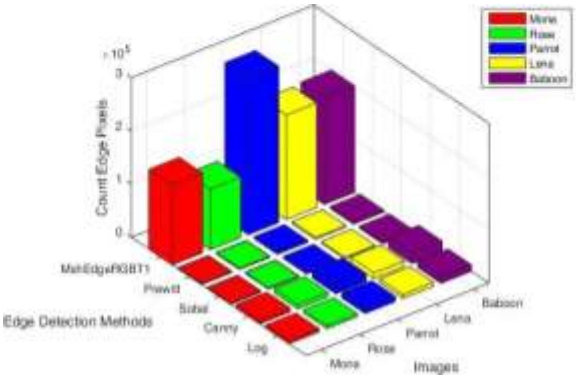


Figure 6. Comparison of edge pixels using different methods

This study aimed to enhance the performance of image steganography through the mshEdgeRGB_T1 algorithm. The results underscore the algorithm's ability to maintain high-quality Stego images, even with increased message lengths, and its superiority in edge detection compared to traditional methods. The importance of this study lies in its potential to significantly increase the embedding capacity of images, making the proposed algorithm highly effective for secure communication. However, some unanswered questions remain, such as the algorithm's performance on a broader range of images and its robustness against various types of attacks.

5. CONCLUSION

In the digital world, sending information through digital media has increased rapidly. Image steganography plays a big role in keeping the data secure while sending information. It is pivotal in ensuring secure and high-quality information exchange. In this study, we explored the mshEdgeRGB_T1 algorithm for image steganography, demonstrating its effectiveness, robustness and flexibility in maintaining image quality while securely embedding data. These attributes make it highly suitable for secure communication applications, such as confidential data transfer and covert operations. The evaluation of the algorithm consistently yields high PSNR and commendably low MSE. Overall, the mshEdgeRGB_T1 algorithm stands as a valuable contribution to the development of secure and efficient data hiding methods, with broad implications for the research community and practical applications. In the future, the work can be extended to implement fuzzy type-2 logic to identify edge pixels and other interesting features of an image. Payload capacity can be further improved by increasing the number of bits embedded per pixel.

ACKNOWLEDGEMENTS

The contributing author recognizes and appreciates the financial aid from the CSIR, under Sanction letter no. 09/382(15462)/2022-EMR-I.

REFERENCES

[1] R. M. Neamah, J. A. Abed, and E. A. Abbood, “Hide text depending on the three channels of pixels in color images using the modified LSB algorithm,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 809–815, Feb. 2020, doi: 10.11591/ijece.v10i1.pp809-815.

[2] 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, Jun. 2003, doi: 10.1016/S0167-8655(02)00402-6.

[3] R. Lukac, K. N. Plataniotis, and A. N. Venetsanopouios, “Color image denoising using evolutionary computation,” *International Journal of Imaging Systems and Technology*, vol. 15, no. 5, pp. 236–251, 2005, doi: 10.1002/ima.20058.

[4] S. Prasad and A. K. Pal, “An RGB colour image steganography scheme using overlapping block-based pixel-value differencing,” *Royal Society Open Science*, vol. 4, no. 4, p. 161066, Apr. 2017, doi: 10.1098/rsos.161066.

[5] H. S. Yusuf and H. Hagras, “High payload image steganography method using fuzzy logic and edge detection,” *International Journal of Computational Science*, vol. 8, no. 4, pp. 123–134, 2020.

[6] D. Dongre and R. Mishra, “A review on edge based image steganography,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 9, pp. 2862–2866, 2014.

[7] R. Devita, I. Fitri, and Yuhandri, “Analysis of gradient magnitude parameter value edge detection on x-ray image of child tuberculosis,” in *AIP Conference Proceedings*, 2024, vol. 3116, no. 1, p. 030001, doi: 10.1063/5.0210241.

[8] P. Melin, O. Mendoza, and O. Castillo, “An improved method for edge detection based on interval type-2 fuzzy logic,” *Expert Systems with Applications*, vol. 37, no. 12, pp. 8527–8535, Dec. 2010, doi: 10.1016/j.eswa.2010.05.023.

[9] E. Nadernejad, S. Sharifzadeh, and H. Hassanpour, “Edge detection techniques: evaluations and comparisons,” *Applied Mathematical Sciences*, vol. 2, no. 31, pp. 1507–1520, 2008, [Online]. Available: http://orbit.dtu.dk/fedora/objects/orbit:39963/datastreams/file_6265007/content.

[10] T. Yuvaraja and R. S. Sabeenian, “Performance analysis of medical image security using steganography based on fuzzy logic,” *Cluster Computing*, vol. 22, no. S2, pp. 3285–3291, Mar. 2019, doi: 10.1007/s10586-018-2096-0.

[11] L. Tang, D. Wu, H. Wang, M. Chen, and J. Xie, “An adaptive fuzzy inference approach for color image steganography,” *Soft Computing*, vol. 25, no. 16, pp. 10987–11004, Aug. 2021, doi: 10.1007/s00500-021-05825-y.

[12] L. Tang, J. Xie, and D. Wu, “An interval type-2 fuzzy edge detection and matrix coding approach for color image adaptive steganography,” *Multimedia Tools and Applications*, vol. 81, no. 27, pp. 39145–39167, 2022, doi: 10.1007/s11042-022-13127-0.

[13] 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, May 2023, doi: 10.1049/ipr2.12773.

[14] K. Muhammad, J. Ahmad, N. U. Rehman, Z. Jan, and M. Sajjad, “CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method,” *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8597–8626, Mar. 2017, doi: 10.1007/s11042-016-3383-5.

[15] S. Kumar, S. Hiranwal, S. D. Purohit, and M. Prasad, *Proceedings of International Conference on Communication and Computational Technologies : ICCCT 2023*. Singapore: Springer Nature Singapore, 2023.

[16] M. A. Aslam *et al.*, “Image steganography using least significant bit (LSB)-a systematic literature review,” in *Proceedings of 2022 2nd International Conference on Computing and Information Technology, ICCIT 2022*, Jan. 2022, pp. 32–38, doi: 10.1109/ICCIT52419.2022.9711628.

[17] H. Humaira, R. Rasyidah, and I. Rahmayuni, “Designing mamdani fuzzy inference systems for decision support systems,” in *Proceedings of ICAITI 2019 - 2nd International Conference on Applied Information Technology and Innovation: Exploring the Future Technology of Applied Information Technology and Innovation*, Sep. 2019, pp. 111–115, doi: 10.1109/ICAITI48442.2019.8982153.

[18] “MATHWORKS - Mamdani and Sugeno fuzzy inference systems,” 2024. Mamdani and Sugeno Fuzzy inference systems - MATLAB & Simulink - MathWorks Italia (accessed Apr. 20, 2024).

[19] S. Tomasiello, W. Pedrycz, and V. Loia, “Fuzzy inference systems. in: contemporary fuzzy logic,” *Artificial Intelligence*, vol. 1, 2022, doi: 10.1007/978-3-030-98974-35.

[20] E. Zielinska, W. Mazurczyk, and K. Szczypiorski, “The advent of steganography in computing environments,” 2012. http://arxiv.org/abs/1202.5289 (accessed Mar. 10, 2024).

[21] I. Thoughts, “Steganography, Python and image manipulation with Pillow,” 2019. https://cprieto.com/posts/2019/04/steganography-python-and-pillow.html (accessed Mar. 20, 2024).

[22] U. B. Woman, “The Pardo Mona Lisa,” 2012. https://oy-vey-gevalt.blogspot.com/2012/02/prado-mona-lisa.html (accessed Mar. 01, 2024).

[23] R. Lukac, K. N. Plataniotis, and A. N. Venetsanopouios, “Color image denoising using evolutionary computation,” *International Journal of Imaging Systems and Technology*, 2005. https://www.researchgate.net/figure/Test-color-images-a-original-image-Parrots-b-image-Parrots-”corrupted-by-5f ig11228638268.

[24] A. Jangra, “Digital image steganography,” *International Journal of Computer Science and Informatics*, pp. 202–205, Jan. 2012, doi: 10.47893/IJCSI.2012.1038.

[25] A. Ahmed and A. Ahmed, “A secure image steganography using LSB and double XOR operations,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 20, no. 5, p. 139, 2020, [Online]. Available: https://www.researchgate.net/publication/342663405.

[26] S. Rajendran and M. Doraipandian, “Chaotic map based random image steganography using LSB technique,” *International Journal of Network Security*, vol. 19, no. 4, pp. 593–598, 2017, doi: 10.6633/IJNS.201707.19(4).12.

BIOGRAPHIES OF AUTHORS

Navita Dhaka received M.Sc. from the Department of Mathematics, Maharshi Dayanand University, Rohtak. She had passed B.Sc. from M.K.J.K. College, Rohtak. She is currently pursuing Ph.D. Degree in Mathematics from the Department of Mathematics, Maharshi Dayanand University, Rohtak. Her area of research is RGB image steganography and edge detection. She can be contacted at email: navita.rs.maths@mdurohtak.ac.in.

Dr. Meenakshi Hooda did master degree in computer applications and an M.Tech in computer science. Then she completed her M.Phil and Ph.D. in computer science. She has worked with Bharti Telesoft (Comviva Technologies). Okhla, Delhi for approx. 3.5 years as software developer and now working with Maharshi Dayanand University, Rohtak, Haryana as an assistant professor for last 10 years. Her research areas are indexing, sptio-temporal indexing, image processing, fuzzy logic, steganography, and cryptography. She has published 9 research papers. She can be contacted at email: meenakshi.maths@mdurohtak.ac.in.

Vinita Yadav received M.Sc. from the Department of Mathematics, Choudhary Devi Lal University, Sirsa. She had passed B.Sc. from K.L.P College, Rewari. She is currently pursuing Ph.D. Degree in Mathematics from the Department of Mathematics, Maharshi Dayanand University, Rohtak. Her area of research is grayscale image steganography, and edge detection. She can be contacted at email: vinita.rs.maths@mdurohtak.ac.in.

Sumeet Gill has done Ph.D. in Computer Science. He has taught in many reputed technical institutes and has more than 25 years of experience in the field of system security and artificial intelligence. His research papers have been published in different Journals of International/National repute and the proceedings of the national/international conferences. He has delivered invited talks and chaired sessions in various conferences. Presently, he is working with Maharshi Dayanand University, Rohtak, Haryana as professor. He can be contacted at email: drSumeetGill@mdurohtak.ac.in.