RESEARCH ARTICLE OPEN ACCESS

High Payload Image Steganography Method Using Fuzzy Logic and Edge Detection

Hala Salih Yusuf [1], Hani Hagras [2]
[1] (Computer Science, Sudan University of Science and Technology/College of Graduate Studies, Khartoum, Sudan)
[2] (The Computational Intelligence Centre, University of Essex /School of Computer Science and Electronic Engineering, U.K)

ABSTRACT
Steganography is one of the information hiding Techniques that hide a message inside another message without drawing any suspicion. In recent years, different methods, which combined steganography and edge detection, have been proposed.
This paper presents a novel Image Steganography method using Least Significant Bit (LSB) and fuzzy logic. We used gradient type-1 fuzzy logic edge detection technique to increase edge pixels, to embed more secret data into the edge pixels than the non-edge pixels, based on the (LSB) substitution technique. Many experiments were conducted to measure the performance of the proposed method by comparing both the original image and stego image, using metrics, like Peak Signal to Noise Ratio (PSNR) and human visual system (HVS), on BSD300 dataset color images. When we compared our results with the previous schemes, the results showed that our proposed scheme provides higher embedding capacity, as well as better stego image quality than previous schemes.
Keywords — Image Steganography, LSB, type-1fuzzy logic, edge detection.

I. INTRODUCTION

In recent decades, due to the increasing development in both Internet and computer technologies, information security is regarded as one of the most important factors of information technology and communication. For that reason, we need to take measures which protect the secret information. In general, secret information may be protected using one of two ways, either cryptography, or steganography. The cryptography method codes the secret message so as not to be understood, while the steganography methods hide the existence of the secret message [1].

Steganography is the art and science of concealing the information in ways that prevent the detection of secret messages. It is derived from the Greek word that means "covered writing." It includes a different array of secret communication methods that hide the existence of the secret message [1],[2].

Steganography can be classified into four types: text steganography, image steganography, audio/video steganography and protocol steganography***3***]. The image steganography method is one of the most effective ways to protect your privacy.The process of hidding a secret message within an image has been widely used, because of the weaknesses of the human visual system (HVS), as well as due to the lowering of the cost of image storage and communication[4].

Image Steganography techniques can be divided into two groups: the Transform domain technique group and the spatial domain group. The Transform domain technique embeds information in the frequency domain of previously transformed image, whereas the spatial domain technique directly embeds information in the intensity of the pixels [1][4].

Regarding the terminology related to image steganography, we note that the original image without the embedded secret message is termed as cover or carrier image, while the image resulting from embedding the secret message is termed as stego image. The secret message can be like a text, image, audio or video. After applying the steganography method, the produced output stego image should look like the cover image.

There are interrelationships between the requirements of steganography: capacity, robustness and imperceptibility (Fig.1 shows this relation). Robustness refers to the amount of alteration that the stego image can resist without an attacker enabled to discover hidden information. Capacity indicates the amount of information that can be embedded in the cover image without damaging the integrity of the cover image.The most important requirement of a steganography system is the imperceptibility, as the strength of steganography system relies on its ability to be unnoticed by the human senses (visually or acoustically). How to balance these three requirements in the fields of information hiding is an interesting issue [2].

The security of any steganography technique depends on the chosen of pixels for embedding. Pixels in textured and noisy areas are better choices for embedding, because they are hard to model. Pixels in edges area can be seen as noisy pixels, because their intensities are either lower or higher than their neighbor pixels, due to a sudden change in the coefficient gradient. Due to these sharp changes in the visual and statistical properties, edges are hard to model in comparison with pixels of smoother area [2]

Our proposed scheme is a kind of the spatial domain technique where the secret message is embedded in edges area using LSB substitution.

This paper is organized as follows; Section II presents an

overview on the related work. Section III shows the classic LSB steganography. Section IV provides overviews of Fuzzy Logic Systems. Section V shows the details of edge detector. Section VI describes the proposed method details. Section VII presents experimental results. Finally, conclusions in Section VIII.

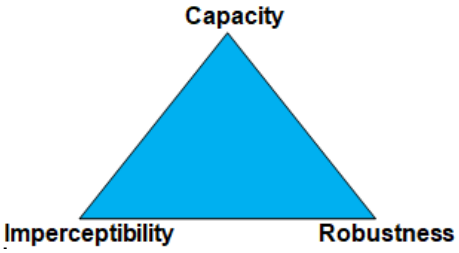


Fig. 1 Magic triangle model of Steganography

II. RELATED WORK

In [*5*] Chen et al. presented a high embedding capacity steganography scheme with a hybrid edge detector. Their scheme was constructed by a combination of the fuzzy edge detector and the Canny edge detector, using the grayscale image to create the hybrid edge image. The embedding operation consists of dividing the edge image into a set of blocks. Each block contains n pixels. They used the first pixel to store the status of the rest pixels. The status of each pixel is classified as '1' or '0' if the pixel is an edge pixel, or non-edge pixel, respectively. Depending on this classification, the x secret bits are embedded into the edge pixel, and y secret bits are embedded into the non-edge pixels, using the LSB substitution technique. For example, take a block B = [P1, P2, P3, P4] pixels, with n = 4, x=1 and y=3. The binary values of these pixels are {[10101010], [10000000], [11111100], [00001111]} respectively, with the secret message S = '0110101'. Assume P2 and P4 are edge pixels. Depending on this, the status of P2, P3 and P4 is '101'. Replace three LSB in pixel P1 with '101' to store the status of the rest pixels into the block. The new value of pixel P1 is [10101**101**]. Then, P2 and P4 are used to embed three secret message bits (y = 3), while P3 is used to embed one secret message bits (x = 1). The new values of pixels P2, P3and P4 after embedding process are [10000**011**], [1111110**0**] and [00001**101**], respectively. In the extraction operation, the inverse process is executed to retrieve the secret message from the stego image according to the status of each pixel, stored in the first pixel of each block. If the pixel status is an edge pixel, extract three LSB, or extract one LSB, if it is a non-edge pixel.

Tseng and Leng [*6*] proposed a block-based scheme using a hybrid fuzzy edge detector, which extended [*5*] to achieve minimal distortion. the proposed scheme has one parameter x, for which the number of embedding secret message bits of non-edge pixels are represented, instead of using the two parameters x and y, as was done in Chen et al.'s scheme. The number of edge pixel embedded should be greater than x , so it uses two bits to present four cases of [x, x+1], [x, x+2], [x, x+3], and [x, x+4] .The second element represents the number of secret bits that should be embedded in edge pixels. In order to achieve the minimal distortion, one of the four cases is chosen by calculating minimum mean square error (MSE) for each 4×4 pixels block.

In *[7]* J. Bai et al. proposed a scheme based on the LSB methodology, combined with the edge detector, which uses the principle that edge areas can tolerate a larger number of embedded bits more than smooth areas according to HVS. In their scheme, they use the cover image to generate Most-Significant-Bit (MSB) image by clearing the last 5 LSBs of each pixel in the original image for edge detection. The 5 LSBs are employed for embedding the secret data while 3 MSBs of all pixels remain unchanged. They categorize the pixels of the cover image into two categories, which are non-edge pixels and edge pixels, respectively. Each cover pixel in the first category contains 'x' secret message bits, and the second category contains 'y' secret message bits, using LSBs substitution. For these two categories, pixels are embedded by the k-LSB substitution, where the value k equals either x or y, which is decided by the edge information. The secret key K is shared between the sender side and the receiver side. For example, suppose the block is = 4 pixels, that is, P1 = [10011011], P2 = [01111110], P3 = [01011000], P4 = [10011100], x = 2, y = 4, then the secret bit S = ' 101001111110 '. Based on the edge information of these four pixels, we know that P1 and P4 are edge pixels and P2 and P3 are non-edge pixels. P1 and P4 pixels will include 4 bits of a secret message, while P2 and P3 will include 2 bits of a secret message. These four pixels will switch to P1 '= [1001 **1010**], p2' = [011111**01**], P3 '= [010110 **11**] and P4' = [1001 **1110**]. In the extraction phase, the receiver retrieves the two parameters x and y from the last four pixels of the image. And also, the edge information is determined the same as in the embedding phase.

III. THE CLASSIC LSB STEGANOGRAPHY

The most clears method to hide information within cover image is the Least Significant Bit (LSB) insertion technique, which provides great performance in terms of high payload and low computational complexity [*4*] [*7*]. In this method, secret message bits are embedded in the least significant bit of each pixel in the image, which means that the value of the lowest bits of pixels in the cover image is employed to indicate the secret message. On the receiving side, the recipient extracts the secret message by reading the lowest bits of pixels in the stego image [*7*].

It is so difficult to notice the difference between the cover image and the stego image to the human eye. For increasing the embedding capacity, two or more LSBs in each pixel can be used to embed secret messages. However, there is a trade-off between invisibility of the message and the embedding payload [*3*][*7*].

IV. FUZZY LOGIC SYSTEMS

The concept of Fuzzy Logic (FL) was initially introduced by Lotfi Zadeh, the founding father of the entire field, in the 1960s [*8*][*9*]. Basically, Fuzzy Logic attempt to mimic human-like style of thinking by means of the use of fuzzy set theory [*10*], [*11*], [*12*].

Fuzzy Logic Systems (FLSs) provide white box models which could be easily understood and analyzed by the ordinary user[*13*]. FLSs employ fuzzy sets as shown in Fig. 2 where a fuzzy set is characterized by a fuzzy Membership Function (MF), i.e. the membership grade for each element is a crisp number in [0,1] [*9*],[*14*], [*15*].

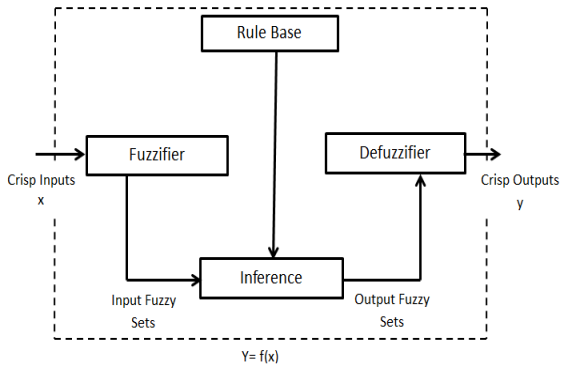


Fig. 2 Fuzzy Logic System[*16*]

A. Definition of Fuzzy Sets

The definition of a fuzzy set F, in a universe of discourse U, is characterized by a membership function μF(x),which associates with values in the interval [0, 1] [*8*].

We can represent the fuzzy set F in U as a set of ordered pairs of a generic element x and its grade of membership function [*8*]:

$$F = \{(x, \mu_F(x)) \mid x \in U\} \qquad (1)$$

When U is continuous (e.g., the real numbers), F is commonly written as:

$$F = \int_U \mu_F(x) / x \qquad (2)$$

In this equation, the integral sign does not denote integration; it denotes the collection of all points x∈U, with the associated membership function $\mu_F(x)$ [*8*].

When U is discrete, F is commonly written as:

$$F = \sum_U \mu_F(x) / x \qquad (3)$$

In this equation, the summation sign does not denote arithmetic addition; it denotes the collection of all points $x \in U$, with the associated membership function $\mu_F(x)$ [*8*].

B. Type-1 Fuzzy Logic Systems

A Fuzzy Logic System (FLS) (shown in Fig. 2) transforms crisp input into crisp output. It consists of four components which are fuzzifier, rules, inference engine, and defuzzifier. When the rules have been established, a FLS can be viewed as a transforming from inputs to outputs and this mapping quantitatively can be expressed as y=f(x)*17*]. Rules may be obtained by experts or consultants, or can be acquired from numerical data. In both cases, the rules are expressed as a group of IF-THEN statements. The fuzzifier transforms crisp numbers into fuzzy sets. It is needed in order to activate, in terms of linguistic variables, the rules, which have fuzzy sets associated with them [*16*]. The FLS inference engine transforms fuzzy sets into other fuzzy sets, and control the way of combining the rules. Like as we humans use many various styles of inferential procedures to help us understand things or to make decisions, there are many different fuzzy logic inferential procedures*18*].

Type-1 fuzzy sets handle the uncertainties related to the FLS inputs and outputs, using precise and crisp membership functions that the user thinks capture the uncertainties [*19*], [*20*], [*21*], [*22*]. When the type-1 membership functions have been selected, all the uncertainty disappears, due to type-1 membership functions are completely precise [*23*],[*24*], [*25*].

V. EDGE DETECTOR

Edge detection a process applied to digital image processing, particularly in the areas of feature extraction, to refer to algorithms and tools that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are usually systematic into a set of curved line segments termed edges.[*26*]. There are many (classical) standard edge detection algorithms such as Sobel, Prewitt, Roberts, Laplacian and Canny operators [*5*],*7*], [*9*].

In the last years, new edge detection methods have been developed using fuzzy techniques, because of the flexibility in dealing with the vagueness and uncertainty in digital images.

A. Gradient Edge Detector

There are some methods to perform the edge detection process; most of them are based on image gradient magnitude, which are calculated with the first derivative of an image. In this paper, the edge detection method is performed by calculating the image gradients with the Euclidean distance; which is the most used and the most important method. This operation includes calculating four image gradients to indicate the edge direction based on a 3×3 matrix (Di, for i =1 . . . 4) (Fig. 3 illustrates that). Each matrix position (Di), of Fig. 3, is represented in Fig. 4, where f indicates the image, x-axis the rows and y-axis the columns [*27*].

According to these positions, the Euclidean distance is applied to calculate the gradients Di using the eq (4). Gradient magnitude, or The edges E, can be calculated with the eq(5) [*27*],[*28*].

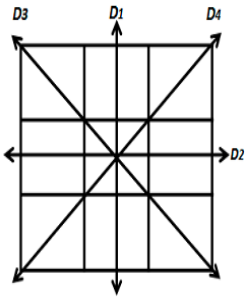


Fig. 3 3 × 3 Matrix indicating direction of the four gradients Di [*27*]

$P_1 = f(x-1, y-1)$	$P_2 = f(x-1, y)$	$P_3 = f(x-1, y+1)$
$P_4 = f(x, y-1)$	$P_5 = f(x, y)$	$P_6 = f(x, y+1)$
$P_7 = f(x+1, y-1)$	$P_8 = f(x+1, y)$	$P_9 = f(x+1, y+1)$

Fig. 4 Matrix position[*27*]

$$D_1 = \sqrt{(p_5 - p_2)^2 + (p_5 - p_8)^2} \qquad (4)$$
$$D_2 = \sqrt{(p_5 - p_4)^2 + (p_5 - p_6)^2}$$
$$D_3 = \sqrt{(p_5 - p_1)^2 + (p_5 - p_9)^2}$$

$$D_4 = \sqrt{(p_5 - p_3)^2 + (p_5 - p_7)^2}$$
$$Edge = D_1 + D_2 + D_3 + D_4 \quad (5)$$

B. Edge Detection Process Based On the Gradient Approach and Type-1 Fuzzy Logic System

The fuzzy logic methodology for edge detection, using gradient magnitude consists of using Eq (4) To get the gradients in the four directions (D1, D2, D3, D4) and use them as inputs to a fuzzy inference system (FIS), instead the Eq. (5). Several kinds of membership functions exist to represent T1FIS, such as triangular, Trapezoidal, Gaussian, etc. T1FIS Gaussian membership function is illustrated in Fig. 5.

The T1FIS in this paper is a singleton Mamdani type, which was designed with four inputs (D1, D2, D3 and D4), and one output. The inputs and outputs are fuzzified using Gaussian membership functions with uncertain mean; each input has three linguistic values (low, medium and high) to determine the grade to which the evaluated gradient corresponds, to be the output edge. Each output has two linguistic values (edge and background) to produce the gradient magnitude edge detection.

For each D input, the Gaussian membership functions were obtained with (9) – (11), and the centers of each function were obtained with (6) – (8), as shown in Fig. 6. For the outputs E (the edges), we can obtain these membership functions directly with (12) – (13) [***28***].

$$low = min(D_i) \quad (6)$$
$$high = max(D_i) \quad (7)$$
$$medium = low + (high - low)/2 \quad (8)$$
$$\sigma = high/8$$

$$\mu(low) = e^{\frac{-(x - low)^2}{2(\sigma)^2}} \quad (9)$$
$$\mu(high) = e^{\frac{-(x - high)^2}{2(\sigma)^2}} \quad (10)$$
$$\mu(medium) = e^{\frac{-(x - medium)^2}{2(\sigma)^2}} \quad (11)$$
$$\mu(background) = e^{\frac{-(x - black)^2}{2(\sigma)^2}} \quad (12)$$
$$\mu(edge) = e^{\frac{-(x - white)^2}{2(\sigma)^2}} \quad (13)$$
$$where\ white = 255, \sigma = white/8$$

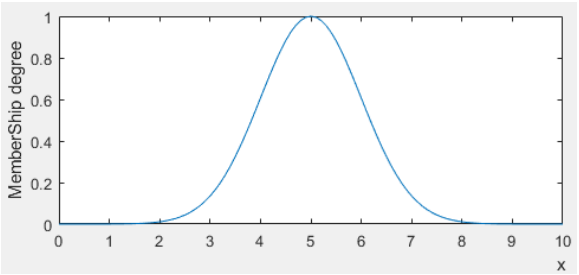


Fig. 5 Type-1 membership function

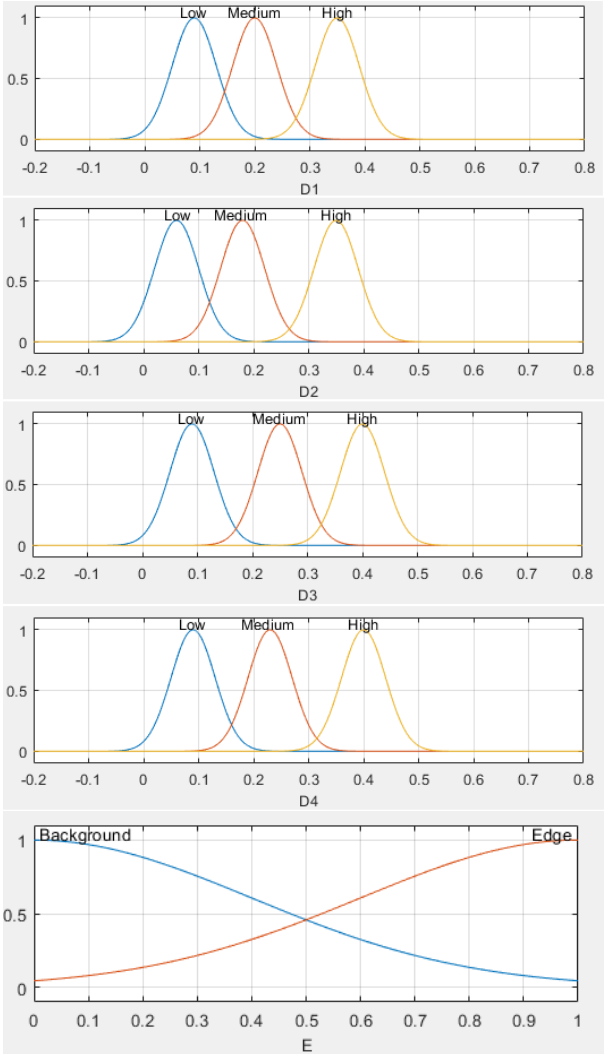


Fig. 6 T1FIS membership function for the inputs (D1, D2, D3 and D4) and the output E

The important part of a fuzzy system is the fuzzy rules, for this proposed method the fuzzy rules consider various combinations of the gradients inputs Di to produce the gradient magnitude output. Fuzzy rules presented in Table I.

TABLE I THREE FUZZY RULES FOR EDGE DETECTION [***28***]

Fuzzy Rules
1. If (D1 is HIGH), or (D2 is HIGH), or (D3 is HIGH), or (D4 is HIGH), then (E is Edge.)
2. If (D1 is MEDIUM), or (D2 is MEDIUM), or (D3 is MEDIUM), or (D4 is MEDIUM), then (E is Edge.)
3. If (D1 is LOW), (D2 is LOW), (D3 is LOW) and (D4 is LOW), then (E is Background.)

The first rule tests the four directions (D1, D2, D3, and D4) if it is high this means an edge. The second rule tests the four directions (D1, D2, D3, and D4) if it is medium this means also an edge. The third rule is only to confirm the first two, because if the four directions are low this mean there is no edge in this pixel[***28***].

VI. PROPOSED METHOD

In this section, a new LSB steganography method will be proposed, which uses hybrid edge detection to make disclosure of an existing secret message a hard operation. The proposed

method is developed into two sides, the sender's side that treats with the embedding process, and the receiver's side that treats with extraction processes.

A. Preprocessing Operation

Some processes are done prior to initiating the embedding operation. On the sender's side, the cover image will be converted into grayscale images, pass a copy of this grayscale image to the canny edge detector and another copy to the proposed type-1fuzzy edge detector algorithm. After that, the two edges will combine together to have a new hybrid edge image. The hybrid edge image, cover image and the secret message will be the inputs of (embedding algorithm) the LSB substitution algorithm.

On the receiver's side, only the stego image will be the input of extraction algorithm to get the secret message back again.

B. Embedding Operation

The embedding operation is responsible for hiding the secret message into the cover image file, using the proposed LSB method that uses the spatial domain of RGB color image. On the sender's side, the secret message will be embedded into the cover image file, and obtained the stego image file as output.

The operation of embedding a secret message in a cover image depends on the proposed hybrid edge image. The number of bits that should embed in each pixel is determined by the category of the pixel in the cover image. We utilize two parameters: x=9 and y=3. If the pixel is an edge pixel, the number of secret bits to be embedded will be x bits, and if the pixel is non-edge pixel, the number of secret bits to be embedded will be y bits.

The first bit in the red color of the cover image will be 0 or 1 to indicate that it is a non-edge or edge pixel respectively. Pixels are embedded by the M-LSB substitution, where the value M equals either x or y, which is decided by the edge information.

Assume that 4 pixels [P1, P2, P3, and P4], are read from the cover image I. According to the edge information of these four pixels, we know that both the third and the fourth pixels are edge pixels, and that the first and second pixels are non-edge pixels. Consider that the secret bit stream s='110010100100111010011110'. The 9 LSBs of the third and the fourth pixels are replaced with the corresponding secret bits, and the first bit of red will be 1 (because it is an edge pixel). Similarly, the 3 secret bits are embedded into the first and second pixels by the LSB substitution method, and the first bit of red will be 0, because it is a non-edge pixel. The values of the pixels before and after the embedding operation will be shown in fig. 7:

```
P1 (11011100 11101011 11101110)
P2 (11010110 11101011 11101100)
P3 (10100001 10000001 01110100)
P4 (10100101 01111011 01100011)
```

```
P1 (11011101 11101010 11101100)
P2 (11011001 11101111 11101010)
P3 (10100000 10000001 01110101)
P4 (10100110 01111011 01100010)
```

Fig. 7. Example of the Proposed Embedding Operation

C. Extracting Operation

In the extraction operation, the receiver extracts the first bit in red color if its 0 or 1 to determine the parameter value of m. If the first bit of red color is 0, this means the value of m will be 3 bits (1bit in red, 1bit in green and1bit in blue color.), and if it is 1, this means the value of m will be 9 bits (3bits in red, 3bits in green and 3bits in blue color). Thus, the secret data will be accurately extracted.

VII. EXPERIMENTAL RESULTS

In this section, experimental results are presented to demonstrate the performance of our proposed method. The embedding capacity, also called payload, is measured by the maximum number of embedded bits per pixel (bpp). Its formula is defined as follows:

$$bpp = \frac{Maximal\ Embedding\ bits}{H \times W}, \qquad (14)$$

where H and W, respectively, are the height and width of the original cover image.

The stego image quality is measured by the following two viewpoints. The first one is the peak signal-to-noise ratio (PSNR) measurement, which we used to calculate the difference between the cover and stego images. The second one, we evaluate the quality of the stego image against that of the cover image, as seen by the human visual system (HVS).
The PSNR formula is defined as follows:

$$PSNR = 10.log_{10}\left(\frac{255^2}{MSE}\right)(dB), \qquad (15)$$

where the (MSE) is the mean square error between the stego image and cover image. For a cover image with width W and height H, the MSE formula is defined as follows:

$$\mathbf{MSE} = \sum_{i=1}^{H} \sum_{j=1}^{W} \left(p_{ij} - p_{ij}\right)^2 / (\mathbf{H} \times \mathbf{W}) \quad (16)$$

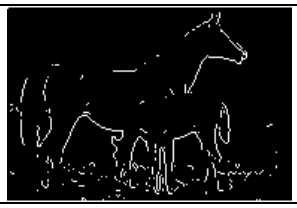
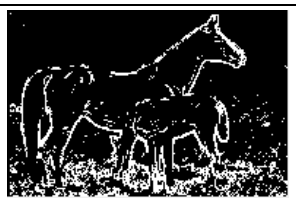
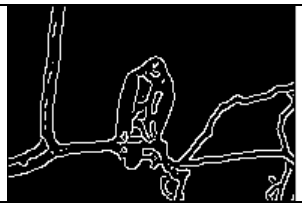
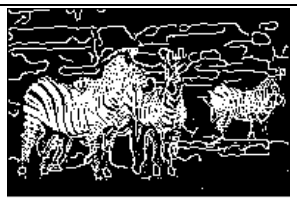
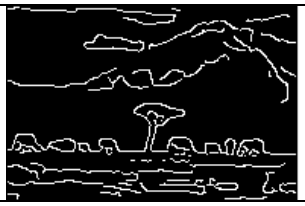
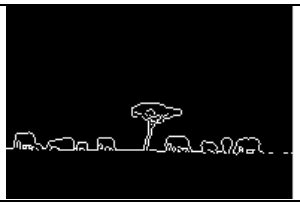
To conduct our experiments, we used, respectively, six 128×192 RGB color images from BSD300 dataset, three from the training images and three from the testing , as shown in Fig. 9. We also used 'lena' image with size 128×128 to compare our results with previous studies.

'Lena' image is shown in Fig. 8.

Fig. 8 'lena' image

		
3096	42049	253027
		
113044	249061	253036

Fig. 9 Six128×192 images from BSD300 dataset

Edge Image		
Canny	**Sobel**	**Proposed method**
		
2008	**439**	**2700**
		
3117	**839**	**5717**
		
1624	**1053**	**3432**
		
2203	**1265**	**3867**
		
3874	**1004**	**6467**
		

1995	**536**	**2600**

Fig. 10 The number of edge pixels detected by Canny, Sobel, and Fuzzy Logic

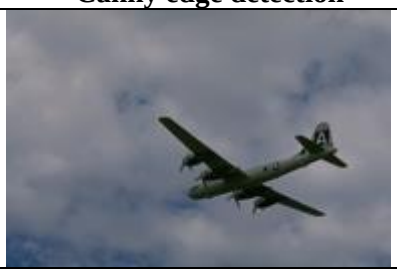
Canny edge detection		
		
Image name = 3096	Image name = 3096	Image name = 3096
PSNR = 48.4630 dB	PSNR = 45.0340 dB	PSNR = 44.5054 dB
payload = 1.3301 bpp	payload = 2.6680 bpp	payload = 3.0010 bpp
Ratio = **32688 bits**	Ratio = 65568 bits	Ratio = 73752 bits
		
Image name = 113044	Image name = 113044	Image name = 113044
PSNR = 48.1963	PSNR = 44.2973	PSNR = 42.7058
payload = 1.3339	payload = 2.6679	payload = 3.6337
Ratio = 32784	Ratio = 65568	Ratio = 89304
Sobel edge detection		
		
Image name = 42049	Image name = 42049	Image name = 42049
PSNR = 50.0422	PSNR = 47.8200	PSNR = 45.4341
payload = 1.3301	payload = 2.00	payload = 3.0010
Ratio = 32688	Ratio = 49152	Ratio = 73752

		
Image name = 249061	Image name = 249061	Image name = 249061
PSNR = 50.1927	PSNR = 46.2740	PSNR = 45.3674
payload = 1.3339	payload = 2.6679	payload = 3.0009
Ratio = 32784	Ratio = 65568	Ratio = 73752
Fuzzy edge detection(our proposed method)		
		
Image name = 253027	Image name = 253027	Image name = 253027
PSNR = 46.4572	PSNR = 42.2049	PSNR = 40.1558
payload = 1.3301	payload = 2.66	payload = 4.5488
Ratio = 32688	Ratio =65568	Ratio = 111792
		
Image name = 253036	Image name = 253036	Image name = 253036
PSNR = 48.1633	PSNR = 44.5028	PSNR = 43.3270
payload = 1.3339	payload = 2.6679	payload = 3.3339
Ratio = 32784	Ratio = 65568	Ratio = 81936

Fig. 11 Experimental results of the proposed scheme with different edge detection techniques using BSD300 dataset image

TABLE II EXPERIMENTAL RESULTS OF THE PROPOSED SCHEME COMPARISON WITH PREVIOUS STUDIES ON 'LENA' IMAGE SIZE OF 128×128

Previous studies						**Proposed scheme**					
Canny		**Sobel**		**Fuzzy**		**Canny**		**Sobel**		**Fuzzy**	
PSNR	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**
48.927	1.111	50.723	1.048	47.003	1.793	49.0110	1.1255	50.2528	1.1223	47.9370	1.2223
39.256	2.222	42.339	2.096	34.554	3.586	45.3841	2.2445	46.9834	2.2510	42.2255	3.515
33.176	3.222	37.397	3.096	26.308	4.586	44.0979	3.0964	45.4577	3.0964	41.5490	4.0776

In this section, the implementation detail of the proposed method will be presented and discussed by applying different edge detectors, which are shown in Fig. 11.The PSNR and HVS have to be used to measure the quality of the stego image in each experiment. Fig. 9 illustrates the comparison between the various types of the edge detectors. From this comparison, we can see that the fuzzy logic edge detector has a larger number of edge pixels.

From Fig. 11, the best PSNR value obtained by the canny edge detector is 48.4630 when the capacity reached to the value 1.3301 bpp, and the worst PSNR obtained is 40.7058, when the capacity reached to the value 3.6337 bpp.

For the Sobel edge detector, the best PSNR obtained is 50.1927 when the capacity reached to the value 1.3339 bpp, and the worst PSNR obtained is 40.7058, when the capacity reached to the value 3.6337 bpp.

The best PSNR obtained by the proposed scheme is 48.1633 when the capacity reached to the value 1.3339 bpp, and the worst PSNR obtained is 40.1558, when the capacity reached to the value 4.5488 bpp.

The results of the experiments indicate that the proposed scheme, has a high number of detecting edge pixels, which means embeds more data without seriously influencing in the quality of the entire image as seen by HVS and PSNR. For that, we can say the proposed scheme satisfies the two requirements of capacity and imperceptibility in the Magic triangle visual model.

In table II demonstrates a comparison between proposed scheme, with the previous scheme in [HYPERLINK \l "Jun17" *7*] in terms of payload and PSNR, on 'Lena' image, with the size of 128×128, using a various edge detector techniques. Through this comparison, we can notice that the PSNR results achieved by our proposed scheme indicate that the stego image quality is better than the previous scheme and that mean is very close to the cover image.

We selected the scheme in *7*] to compare with our proposed scheme, because it outperforms all the previous studies, in terms of payload and PSNR.

The main reason superiority of our proposed scheme is used the RGB image, which gave us a large domain amounting reached to 24bits for each pixel. While these previous schemes used grayscale image, which means that each pixel has only 8bits that can be used for embedded.

This means that the embedding space in our scheme is three times more than the other schemes, and therefore improves image quality.

VIII. CONCLUSIONS

In this paper, we have proposed a new steganography scheme, which combines the gradient type-1 fuzzy edge detector with the canny edge detector to make disclosure of an existing secret message a hard operation. We used the (LSB) substitution technique to embed the secret data into the cover image. Many experiments have been done and when compared our results with the previous schemes, the results showed that our proposed scheme provides high embedding capacity, as well as better stego image quality than previous schemes.

REFERENCE

[1] Amanpreet Kaur, Renu Dhir, and Geeta Sikka, "A New Image Steganography Based On First Component Alteration Technique," *International Journal of Computer Science and Information Security(IJCSIS)*, vol. 6, no. 3, p. 4, 2009.

[2] 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. 3, p. 11, 2016.

[3] T. Morkel , J.H.P. Eloff , and M.S. Olivier , "AN OVERVIEW OF IMAGE STEGANOGRAPHY," *Information and Computer Security Architecture (ICSA) Research Group*, 2005.

[4] Anastasia Ioannidou, Spyros T. Halkidis, and George Stephanides, "A novel technique for image steganography based on a high payload method and edge detection," *Expert Systems with Applications*, vol. 39, p. 14, 2012.

[5] Wen-Jan Chen, Chin-Chen Chang, and T. Hoang Ngan Le, "High payload steganography mechanism using hybrid edge detector," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3292–3301, 2010.

[6] Hsien-Wen Tseng and H ui-Shih Leng, "High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion," *The Institution of Engineering and Technology (IET Image Process)*, vol. 8, no. 11, pp. 647 – 654, 2014.

[7] Junlan Bai, Chin-Chen Chang, Thai-Son Nguyen, Ce Zhu, and Yanjun Liu, "A high payload steganographic algorithm based on edge detection," *Displays*, vol. 46, p. 10, December 2017.

[8] L. A. Zadeh, "Navy Fuzzy Sets *," *Introduction and Control*, vol. 8, pp. 338–353, 1965.

[9] Hala Salih Yusuf and Hani Hagras, "Towards Image Steganography Using Type-2 Fuzzy Logic and Edge Detection ," *In 2018 10th Computer Science and Electronic Engineering (CEEC)*, no. IEEE, pp. 75-78, 2018.

[10] Hellmann M., "Fuzzy Logic Introduction," vol. 1, 1965.

[11] Heba Abdelgader Mohammed and Hani Hagras, "Towards Developing Type 2 Fuzzy Logic Diet Recommendation System for Diabetes," *In 2018 10th Computer Science and Electronic Engineering (CEEC), no. IEEE*, pp. 56-59, 2019.

[12] Hani Hagras, Victor Callaghan, Martin Colley, and Malcolm Carr-West, "A Fuzzy-Genetic Based Embedded-Agent Approach to Learning and Control in Agricultural Autonomous Vehicles," *Proceedings of the1999 IEEE International Conference on Robotics and Automation*, pp. 1005-1010, Detroit, U.S.A, May 1999.

[13] Andrew Starkeya, Hani Hagras, Sid Shakya, and Gilbert Owusu, "A multi-objective genetic type-2 fuzzy logic based system for mobile field workforce area optimization," *Journal of Information Sciences*, vol. 329, pp. 390–411, 2016.

[14] Dario Bernardo, Hani Hagras, and Edward Tsang, "A Genetic Type-2 Fuzzy Logic Based System for the Generation of Summarised Linguistic Predictive Models for Financial Applications," *Soft Computing*, vol. 17, no. 12, pp. 2185-2201, Dec 2013.

[15] Andrew Starkey, Hani Hagras, Sid Shakya, and Gilbert Owusu, "iPatch: A Many-Objective Type-2 Fuzzy Logic System for Field Workforce Optimisation," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 502-514, August 2018.

[16] Victor Callaghan et al., "Programming iSpaces: A Tale of Two Paradigms," in *Intelligent Spaces: The Application of Pervasive ICT*, Chapter 24, pp.389- 421, December 2005. Springer-Verlag, Ed.: (Eds: A. Steventon, S. Wright), December December 2005, ch. 24, pp. 389-421.

[17] Maha Salaheldeen Elbadawy Alhassan and Hani Hagras, "A Congestion Control Approach Based on Weighted Random Early Detection and Type-2 Fuzzy Logic System," *International Journal of Computer Science Trends and Technology (IJCST)* , vol. 8, no. 4, pp. 83-94, Jul-Aug 2020.

[18] Javier Andreu-Perez, Fan Cao, Hani Hagras, and Guang-Zhong Yang, "A self-adaptive online brain–machine interface of a humanoid robot through a general type-2 fuzzy inference system," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 101-116, Februray 2018.

[19] Christopher Lynch, Hani Hagras, and Victor Callaghan, "Embedded Interval Type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines," *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006)*, pp. 1340-1347, July 2006.

[20] Hani Hagras and Christian Wagner, "Towards the Widespread Use of Type-2 Fuzzy Logic Systems in Real World Applications," *IEEE Computational Intelligence Magazine*, pp. 14-24, August 2012.

[21] Michela Antonelli, Dario Bernardo, Hani Hagras, and Francesco Marcelloni, "Multiobjective Evolutionary Optimization of Type-2 Fuzzy Rule-Based Systems for Financial Data Classification," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 2, pp. 249-264, 2016.

[22] Ahmet Sakalli, Tufan Kumbasa, Engin Yesil, and Hani Hagras, "Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the Magnetic Levitation system," in *the 2014 IEEE International Conference on Fuzzy Systems*, Beijing, China, 2014, pp. 1859-1866.

[23] Hani A. Hagras , "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots," *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, vol. 12, no. 4, AUGUST 2004.

[24] J. Mendel , "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions," *Upper Saddle River, NJ: Prentice-Hall*, p. 555pp, 2001.

[25] Maha Salaheldeen Elbadawy Alhassan and Hani Hagras, "Towards Congestion Control Approach Based on Weighted Random Early Detection and Type-2 Fuzzy Logic System," *In 2018 10th Computer Science and Electronic Engineering (CEEC), no. IEEE*, pp. 71-74, 2019.

[26] Nitin Jain, Sachin Meshram, and Shikha Dubey, "Image Steganography Using LSB and Edge – Detection Technique," *International Journal of Soft Computing and Engineering (IJSCE)* , vol. 2, no. 3, p. 6, July 2012.

[27] Claudia I. Gonzalez, Patricia Melin, and Oscar Castillo, "Edge Detection Method Based on General Type-2 Fuzzy Logic Applied to Color Images," *information*, vol. 8, no. 104, August 2017.

[28] Patricia Melin, Olivia Mendoza, and Oscar Castillo, "An improved method for edge detection based on interval type-2 fuzzy logic," *Expert Systems with Applications*, vol. 37, pp. 8527–8535, May 2010.