Borneo International Journal eISSN 2636-9826; Vol. 6 (3); 2023; 119- 127
Published by *Majmuah Enterprise*
www.majmuah.com

Comparative Analysis of Methods for Digital Steganography in Images

ALAA JABBAR QASIM ALMALIKI, OSMAN GHAZALI and ROSHIDI DIN
School of Computing, Universiti Utara Malaysia (UUM), 06010 Sintok, Kedah Darul Aman, MALAYSIA
Email : alaa_jabbar@ahsgs.uum.edu.my, osman@uum.edu.my , roshidi@uum.edu.my
| Tel: *+60184020823,+60124422747, +60175981306*|

Received: September 22, 2023
Accepted: September 25, 2023
Online Published: September 26, 2023

Abstract
Sensitive information being shared on the internet is growing. Because of this, it is increasingly necessary to take security measures whilst this information travels in the network. Digital steganography allows one to send sensitive information in a hidden manner. Although there is a plethora of techniques for such a goal, finding an appropriate one is not always simple. This paper implements and compares spatial-domain digital steganography techniques in both RGB and grayscale images. A frequency-domain heuristic for reducing the visual impact of digital steganography in grayscale images is presented. As another result of this work, a dataset is also available in the Kaggle platform with 18 GB of images, containing secret messages using the techniques under study.
Keywords: Digital Stenography, Information Security, Digital Image Processing

1. Introduction

The advent of the internet has engendered several technological advances in academic and corporate environments, with possibly even more drastic repercussions in other areas of modern society. In parallel with these advances, there is a growing concern about possible privacy violations, which motivate the development of effective methods of protecting digital information. As promising methods belonging to this technological spectrum, the digital steganography techniques, on which this article focuses, should be highlighted. Steganography is a technique that consists of hiding a message within another message, so that only the sender and recipient of the message, who are aware of the existence of the hidden information, have access to such information. The word łesteganografiaż derives from the Greek steganós (which means hidden, hidden) and grafia, which means writing (Din, Qasim, Abdullah, Elias, & Technology, 2019; Qasim & Alyousuf, 2021; Qasim, Din, Alyousuf, & Informatics, 2020). In its digital aspect, steganography focuses on information stored in digital media (Alaa Jabbar & Farah Qasim Ahmed, 2021). The steganography problem can be interpreted as a variant of the prisoner problem, in which two individuals aim to initiate communication through a medium without arousing suspicion of possible intruders (Din, Qasim, & Informatics, 2019; QASSIM & SUDHAKAR, 2015; Roshidi Din, 2018).

It is worth confusing steganography with cryptography, because while the latter is intended for encoding a message, the former aims to camouflage a message within other information, usually of an audio-visual character. In the event of such insertion of information is discovered by an interceptor, the interceptor may easily rescue the original image, if the technique used by the steganography is known to him. It should be noted that, in this context, the interceptor does not need to know any security keys. specific decryption(Alwan, Farhan, Mahdi, & Engineering, 2020). Embedding the message in the different steganography methods should reduce the resulting distortion, which could be formulated as a coding problem source with a fidelity criterion (Sayood, 2017). if convenient ensure greater security to the information transmitted or stored, Encryption techniques can be simultaneously combined with those of steganography. The use of digital media to communicate and store information has gradually increased over time, as has the number of attacks aimed at stealing such information (Manjula & Danti, 2015). With this scenario, it becomes increasingly necessary to guarantee the protection and integrity of data, and the development of encryption and steganography technologies is increasing in order to prevent unauthorized people from having access to them (Khudher, 2021; Vijay, Jayareka, Kirubasri, & Vijay, 2021). It is important to point out that digital steganography techniques can be used in malicious attacks to prevent the leakage of sensitive data from being noticed by anti-malware tools. In these attacks, sensitive data is hidden by the attacker in information that would normally travel over the network. In this way, such data becomes invisible to security systems. Among the main contributions of this article, it is worth mentioning:

(1) Proposes a modification of the SSB-4-based technique.
(2) Comparison between different digital steganography techniques
present in the literature, namely:
•LSB (Least Significant Bit);
•LSB Gray Scale.
•SSB-4 (System of Steganography Using Bit 4);
•SSB-N (System of Steganography Using Bit N);
• Steganography based on DCT (Discrete Cosine Transform);
• Steganography based on FFT (Fast Fourier Transform).
(3) Comparison of the LSB technique in both the RGB and grayscale domains.

Borneo International Journal eISSN 2636-9826; Vol. 6 (3); 2023; 119- 127
Published by *Majmuah Enterprise*
www.majmuah.com

Steganography techniques are commonly classified into spatial domain or frequency domain. The techniques in the domain spatial changes directly change the least significant bits of the pixel values that make up the matrix representation of a image (Alyousuf, Din, Qasim, & Informatics, 2020). Among the techniques presented in this work, the following are contained in the spatial domain: LSB, grayscale LSB, SSB-4 and SSB-N. The techniques contained in the frequency domain convert the cover image to the frequency domain by applying a transform before storing the secret message (Liao, Wen, & Zhang, 2011). Among the transforms that can be applied, we can mention the discrete cosine transform, discrete Fourier transform, wavelet transform, among others (Sidhik, Sudheer, & Pillai, 2015). The techniques cited in this work that are in this field are DCT-based steganography and FFT-based steganography. A more detailed presentation on each of these techniques is given in Section 2. Computational experiments were carried out to compare the implemented techniques, as well as to evaluate the proposed modifications. The results obtained show that after the steganography process the image containing the secret message has little visual distortion. Furthermore, for the DCT and FFT techniques, a low error rate is presented in the secret message after its retrieval. It is important to emphasize that for the other techniques mentioned in this work, under ideal conditions and as long as the cover image does not undergo compression or editing processes, no errors are observed in the recovery of secret messages. This article is organized as follows. Section 2 presents the theoretical framework of the digital steganography techniques in focus, as well as the proposed methodology, as well as the changes made to the DCT and FFT techniques.

2. Theoretical Approach and Methodology

A comparison between six digital steganography techniques will be made in this article. These techniques are based on the algorithms of: (i) Least Significant Bit (LSB), (ii) Least Significant Bit in Grayscale Images (Least Significant Bit Grayscale - LSB Grayscale), (iii) Steganography System Using Bit 4 (System of Steganography Using Bit 4 - SSB-4), (iv) System of Steganography Using Random Bit -SSB-n, (v) Discrete Cosine Transform - DCT, and (vi) Fast Fourier Transform (FFT). Such techniques were implemented using the Python language. Two datasets extracted from the Kaggle platform were used, a dataset of images (Landscape Pictures) 1, which was used as a public image, also known as a cover image, to store the secret messages, and a dataset of articles from Wikipedia (English Wikipedia Articles). 2017-08-20 SQLite) 2 that was used to compose secret messages. The compared techniques will be described below.

2.1 Less Significant Bit

The most common approaches for performing message insertion in images using noise are based on the LSB (Least Significant Bit) technique, which is a non-adaptive steganography method in the spatial domain, which mobilizes the least significant bits to store the data that intends to protect. Figure 1 illustrates the use of the LSB technique through the representation of each pixel in the image. It can be seen that the bits signalled in Gray are representing the message being stored, while the others refer to the information of the original image. Thus, it is possible to select a least significant bit in each byte of the image as a place where the message will be safely hidden, without causing visually perceptible changes in it (Dhawan, 2011; Mulla, Gunjikar, & Naik, 2013). In this article, several tests were performed, and it was empirically observed that changes in the last two bits of the image provide the best relationship between minimizing visual distortion and maximizing space for message storage (Kermani & Jamzad, 2005). Therefore, this article uses the two least significant bits of a set of eight bits that represent the values of each channel (red, green and blue) of each pixel in the image, as represented in Figure 1.

	Red Channel (R)								**green Channel (G)**								**blue Channel (B)**							
1º Pixel	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1	0	1	0
2º Pixel	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1
3º Pixel	1	0	1	1	0	0	0	1	0	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
nº Pixel	0	1	1	1	0	0	1	0	1		1	1		1	1	0	0	0	1	1	1	0	1	0

Figure 1: Representation of each image pixel using the LSB technique.

The message is encoded in this work through encoding ASCII The least significant bits of the image are replaced by the bits of the message that you want to protect. The Algorithm 1

presents the pseudocode of the LSB technique. The algorithm starts by receiving the cover image (ic) and the secret message (msg). Then the bits of the cover image are extracted using the function $getBits$ (ic (line 1). The amount of) bits of the cover image is obtained through the function $getSizes$ (ic (line 2). From the number of bits in the cover image, the maximum size that the secret message can have been calculated, so that it is possible to hide it inside the cover image. This maximum size is obtained as shown in line 3, as we will only use a quarter of the bits to store the secret message. If it is possible to store the secret message inside the cover image (line 4), a loop that performs this procedure is executed

Borneo International Journal eISSN 2636-9826; Vol. 6 (3); 2023; 119- 127
Published by *Majmuah Enterprise*
www.majmuah.com

(lines 7 - 11). This loop iterates over the vector containing the secret message bits ($bitsMsg$), extracting two bits per iteration. At each iteration, these two bits are stored in the last two bits of a byte of the cover image (lines 8 - 9). After storing the bits, 8 is added to the variable $contador\ Bits$, in order to allow processing of the next byte (line 10). This variable is an index to the vector $ibits$. After the loop ends, the variable im receives the image containing the secret message (line 12). the algorithm ends by returning the original image properly modified by inserting the hidden message (line 13). Algorithm 1 is executed in O(n) time, where n represents the amount of secret message bits.

Algorithm 1: LSB (i_c, msg)
1: $i_{bits} \leftarrow getbits(i_c)$
2: $size_{i_{bits}} \leftarrow getsize(i_{bits})$
3: $tamMaxMsg \leftarrow \lfloor size_{i_{bits}}/4 \rfloor$
4: **If** $getsize(msg) \leq tamMaxMsg$ **then**
5: $bitMsg[\ \ ] \leftarrow$ getAsciiBit(msg)
6: $contadorBits \leftarrow 7$
7: **For** each $\{bit_1, bit_2\} \in bitsMsg$ **do**
8: $i_{bits}[contadorBits] \leftarrow bit_1$
9: $i_{bits}[contadorBits + 1] \leftarrow bit_2$
10: $contadorBits += 8$
11: **End for**
12: $i_m \leftarrow$ bitsToImage (i_{bits})
13: **Return** i_m
14: **End if**

2.2 Less Significant Bit in Grayscale

	Grayscale Channel							
1º Pixel	1	0	1	1	0	1	1	1
2º Pixel	0	1	1	0	1	0	1	0
3º Pixel	1	0	1	1	0	0	0	1
...	...	...	...	...	...	...	...	...
nº Pixel	0	1	1	1	0	0	1	0

Figure 2: Representation of each image pixel using the LSB technique in Gray Scale.

The grayscale LSB technique carries out the entire process mentioned above. Noted in Section 2.1, but using grayscale images. In this work we convert the test dataset images to grey scale. The use of grayscale images has with the objective of reducing both the visual distortions in the post-processed image, as well as improving the values obtained in the metrics of evaluation used, (i) Mean Squared Error (MSE) and (ii) Peak Signal to Noise Ratio (PSNR). This technique is presented in Figure 2, in which each line represents a pixel of the image matrix, traversing the respective image matrix from left to right, line by line. In this figure, the bits in white color represent the bits of each pixel that remain unchanged at the end of the procedure and in gray they are the bits that are changed to store the secret message are represented. In this article, the comparison is performed between the image in original grayscale and the grayscale image that contains the secret message. The conversion from RGB to grayscale was performed following the ITU-R Recommendation BT.709.

	Grayscale Channel							
1º Pixel	1	0	1	1	0	1	1	1
2º Pixel	0	1	1	0	1	0	1	0
3º Pixel	1	0	1	1	0	0	0	1
...	...	...	...	...	...	...	...	...
nº Pixel	0	1	1	1	0	0	1	0

Figure 3: Representation of each image pixel using the SSB-4 technique.

In some of the digital steganography techniques present in the literature, the least significant bits of an image are used to store the secret message. However, in order to make it difficult to detect digital steganography through the analysis of the last two, best bits of each pixel of the image, the technique (SSB-4) is based in storing the secret message in the fourth bit of the image of cover (Bender, Gruhl, Morimoto, & Lu, 1996). In this technique, the cover image is divided into n equal parts, with one pixel of each part destined for the storage of the secret message. The value of n is calculated through Equation (1):

$$n = \lfloor getSize(msg) \rfloor \quad (1)$$

where ic represents the cover image and msg represents the message secret. In this technique, after inserting the secret message in the fourth bit of the selected pixels, a normalization of bits 1, 2, 3, and/or 5 is performed so that the difference between the original value of that pixel and the value after modification is minimal. With this, it is expected that the visual distortion generated by this technique is minimal. This technique is presented in Figure 3, in which each line represents a pixel of the image matrix being traversed from left to right, line by line. While the bits represented in white are preserved during the procedure, the bits marked in gray are changed to incorporate the secret message. In turn, the bits represented in yellow are modified by the normalization routine, aiming to reduce the difference between the pixel that contains the secret message and the original pixel, as well as reducing the distortion generated in the image that will contain the secret message.

2.4 Random Bit (SSB-N) in Grayscale

	Grayscale Channel							
1º Pixel	1	0	1	1	0	1	1	1
2º Pixel	0	1	1	0	1	0	1	0
3º Pixel	1	0	1	1	0	0	0	1
…	…	…	…	…	…	…	…	…
nº Pixel	0	1	1	1	0	0	1	0

Figure 4: Representation of each image pixel using the SSB-N technique.

In order to make both the detection and the extraction of the secret message stored in an image to which it has been applied to digital steganography, we propose a new approach based on SSB-4 technique. The proposed approach stores each bit referring to the secret message in a randomly chosen bit between the first and fourth bits of each pixel of the cover image. To reduce the discrepancy between the pixel of the modified image and the respective pixel of the cover image, alterations of the other bits located between the first and fifth bits are allowed. As the choice of the bit to be changed will be random, the aim is to generate a distortion in the resulting image that is less severe than that engendered by the SSB-4 technique, since in some cases the bit to be modified will be less significant than the fourth bit. It is important to emphasize that as the choice of the bit that will contain the secret message in each pixel will be random, it will be necessary to know the sequence of indices used to retrieve the secret message. This provides additional protection for the message being stored in the cover image. As we defined earlier a maximum size for the cover image, this sequence index can be reused across multiple images, simplifying the secret message retrieval process. In order to further reduce the distortion generated by storing the secret message in the cover image using the SSB-N technique, the cover image is divided into n equal parts, so that the first pixel of each part is used to store the secret message. The value of n is calculated through Equation (2), and the size of n in bits is calculated through Equation (3):

$$n = \lfloor getSize(msg) \rfloor \quad (2)$$

$$getSize(msg) = \left\lfloor \frac{getSize(i_C)/8}{getSize(msg)} \right\rfloor \quad (3)$$

where ic represents the cover image, msg represents the secret message, and the function $getSize$ returns the size in bits. The cover image size is divided by 8, as only one bit of each pixel is used to store the hidden message. With this, it becomes necessary to know the size of the secret message for its recovery, providing more security to this technique. This technique is presented in Figure 4, in which each line represents a pixel of the image matrix being traversed from left to right, line by line. The colors of each bit follow the same rules used in the presentation of Figure 3. These rules are described in Section 2.3.

2.5 Discrete Cosine Transform

DCT-based steganography techniques are based on the property that the images have some redundancy. Of that shape, for each color component, the technique uses the transform of discrete cosine to convert successive blocks of 8 × 8 pixels into 64 coefficients of a DCT. In this way, the least significant bits of the DCT coefficients can be used as bits. redundant to hide a message. As the changes performed in the image are concentrated in the frequency domain, and not in the spatial domain, techniques based on DCT do not leave perceptible traces for visual analysis. In the implementation of this work, we chose to hide the secret message in the DC component of the DCT, which is represented by the coefficient located in the upper left corner of the DCT coefficient matrix. This coefficient was chosen in order to reduce the distortion generated when storing the message in the cover image, given that it represents the average color value of an 8x8 region of the image. This choice was made based on empirical tests and existing data in the literature. The algorithm developed for

Borneo International Journal eISSN 2636-9826; Vol. 6 (3); 2023; 119- 127
Published by *Majmuah Enterprise*
www.majmuah.com

this technique is very similar to the algorithm of the LSB technique, and its main difference occurs before calling the function $getBits$ (ic (row 1), moment) where the image is converted to YCbCr format and applied the discrete cosine transform in the image, as well as after the return procedure (line 13) the inverse operation is performed. The 2D discrete cosine transform is shown in Equations (4), (5) and (6):

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2Y+1)v\pi}{2M}\right) \quad (4)$$

$$C_u = \begin{cases} \frac{1}{\sqrt{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1,2,\dots,N-1 \end{cases} \quad (5)$$

$$C_v = \begin{cases} \frac{1}{\sqrt{M}} & v = 0 \\ \sqrt{\frac{2}{M}} & u = 1,2,\dots,M-1 \end{cases} \quad (6)$$

where $F\ (u, v)$ represents the value of position (u, v) in the image after the discrete cosine transform, $f\ (x, y)$ represents the value of the position (x, y) in the image before the discrete cosine transform. Note that Cu and Cv are constants defined in Equations (5) and (6), where N and M represent the image dimensions, width, and height, respectively. The digital images used in this work are stored in matrix format, where an RGB image has three arrays, one to represent each channel. As each channel has two dimensions (width and height), the 2D discrete cosine transform is used.

2.6 Fast Fourier Transform

The fast Fourier transform based steganography technique is very similar to the discrete cosine transform based technique. However, in the calculation of the 2D fast Fourier transform we use:

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\,\mathrm{EXP}\left(-j\frac{2\pi(u\pi+vy)}{n}\right) \quad (7)$$

where $F\ (u, v)$ represents the value of position (u, v) in the image after the fast Fourier transform, $f\ (x, y)$ represents the value of position (x, y) in the image before the fast Fourier transform. As before, (N, M) represent the dimensions of the image, width and height, respectively. As in the DCT technique, the secret message is stored in the least significant bits of the element located in the upper left corner of the matrix resulting from the application of the transform in each 8×8 block of the image. Therefore, the algorithm developed for this technique is very similar to the pseudocode presented in Algorithm 1, for the LSB technique, its main difference being that before calling the function $getBits$ (i_c(line 1) the image is)converted to the YCbCr format and the fast Fourier transform is applied to the image, and after the procedure returns (line 13) the inverse operation is performed. In this article, the fast Fourier transform based steganography technique was implemented in order to perform an empirical comparison of the results of the computational experiments of this technique and the technique based on the cosine transform. discreet. Both techniques belong to the set of techniques in the frequency domain, while the other techniques presented operate in the spatial domain.

2.7 Comparison of the Techniques Presented

Table 2 presents a comparison of the six digital steganography techniques presented in this article. The first two techniques provide two bits per pixel for the secret message, therefore allow larger messages to be hidden in compared to the other techniques. It is important to point out that the SSB-4 and SSB-N techniques perform a normalization in the value of each pixel in order to reduce the distortion generated in the cover image and make it difficult to detect digital steganography. To recover secret message using the SSB-N technique it is necessary to have knowledge of the sequence of indices that were used to hide the message, as this sequence is generated randomly.

Table 1: Comparative results of the techniques under study

	LSB RGB	LSB Grayscale	SSB-4	SSB-N	DCT	FFT
Domain	Spatial	Spatial	Spatial	Spatial	Frequency	Frequency
Bits per pixel used for secret message	2	2	1	1	1	1
Uses normalization to reduce the distortion generated			✓	✓		
It is necessary to know some additional data to reveal the message				✓		

Borneo International Journal eISSN 2636-9826; Vol. 6 (3); 2023; 119- 127
Published by *Majmuah Enterprise*
www.majmuah.com

Use transforms to hide the message					✓	✓

2.8 Evaluation and Comparison Metrics

In this work, they were used as a quality metric of the images the values of MSE, PSNR and SSIM to measure the difference between the original image and the resulting image, which contains the hidden message. The value of MSE can be obtained through the Equation (8).

$$MSE = \frac{1}{MN} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left(f(x,y) - f^{\tilde{}}(x,y) \right)^2 \quad (8)$$

where (N, M) represents the dimensions of the image, $f\ (x, y)$ represents the value at position (x, y) in the original image, and $f^{\tilde{}}\ (x, y)$ represents the value at position (x, y) in the image with the message hidden.

The PSNR value is calculated using

$$PSNR = 10 \log_{10} \left(\frac{MAX_i^2}{MES} \right) \quad (9)$$

where MSE represents the value calculated by Equation (8) and MAX_i^2 represents the largest value for a pixel present in the squared image, usually this value is 255 2 = 65025. The value of SSIM (Structural Similarity) is calculated through Equation (10):

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (10)$$

where μ_x and μ_y represent the average value of x and y, respectively. σ_x^2 and σ_y^2 are the variance of x and y, in that order, and σ_{xy} represents the covariance between x and y. The coefficients c_1 and c_2 are defined by $c_1=(k_1L)^2$ and $c_2=(k_2L)^2$, where $L= 2^{\text{bits per pixel}} - 1$, $k1= 0.01$ and $k2= 0.03$.

3. Computational Experiments

The previously presented steganography techniques are evaluated through some computational experiments. This section compares these techniques and discusses the results obtained. The tests were performed on a machine with the following configuration: Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz 1.80 GHz, 4.00 GB RAM, 750 Gigabyte HD.

3.1 Dataset

3 An 18 GB dataset of images was generated with messages hidden secrets using the implemented techniques. It fits emphasize that such dataset is available for use public.

3.2 Results

Figure 5 presents the images resulting from the techniques applied in RGB images and Figure 6 shows the resulting images application of the techniques under study to scaled images of Grey. The images presented in Figures 5(a), 5(b) and 5(c) are the cover images (RGB). Figures 5(d)-5(l) show the images results (with the messages hidden) after applying each steganography technique, for images in RGB, to the respective cover images. In general, it can be seen that the result obtained has little visual distortion, and for the LSB is visually imperceptible, regardless of the cover image used. The images presented in Figures 6(a), 6(b) and 6(c) are the cover images for the grayscale tests. Watching the results presented, through the images, for the techniques in grayscale (Figures 6(d)-6(j), 6(e)-6(k) and 6(f)-6(l)), we can conclude that the comparison between the cover images and the images results (with the message hidden) were satisfactory, being impossible to visually perceive the difference between the images of cover and those resulting from the applied techniques. Table 2 presents the values of the MSE, PSNR and SSIM quality measures for the techniques studied in this work, referring to the cover images in comparison with the respective post-processed image. An article taken from the Wikipedia dataset was stored in each of them. In the images being compared in the table, the same article that was chosen randomly was stored. Ideally, to compare the results, it is necessary that all images maintain the same ratio between the size of the cover image and the size of the secret message stored. In order to maintain this proportion, the secret message was repeated n times until the storage space available for each technique in each cover image was completely filled. It is important to note that in the LSB technique, using RGB images,

(a) Original Baboon Image.

(b) Original Image - Pepper.

(c) Original Landscape Image.

Borneo International Journal eISSN 2636-9826; Vol. 6 (3); 2023; 119- 127
Published by *Majmuah Enterprise*
www.majmuah.com

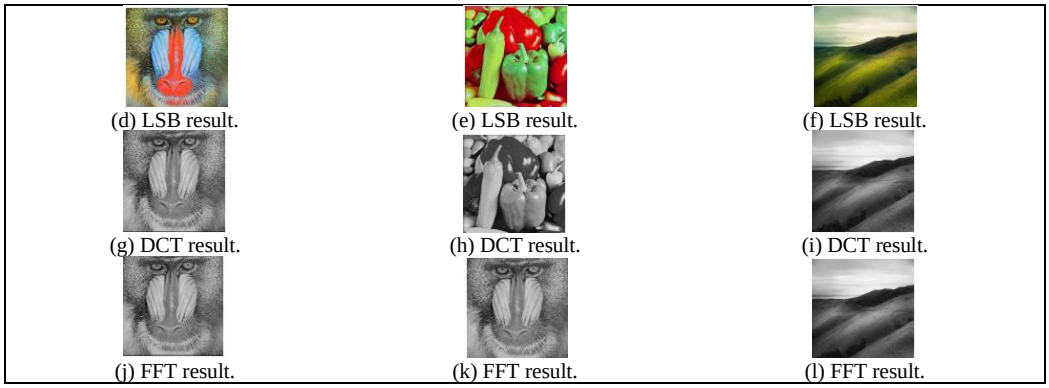

Figure 5: Result of the application of steganography techniques digital image in three RGB images.

secret message is stored in all existing channels in the images. In the LSB, SSB-4, and Pseudo Random Permutation of Bit Index techniques using grayscale images, the secret message is stored in your single channel. Therefore, the intensity of each channel in these techniques does not generate a change significant in terms of quality. For DCT and FFT techniques the quality measure is sensitive to the intensity of the chosen channel to store the message.

Table 2: Results of the Evaluation Metrics.

	Baboon			Pepper			Landscape		
Technique	MSE	PSNR (dB)	SSIM	MSE	PSNR (dB)	SSIM	MSE	PSNR (dB)	SSIM
LSB	4.1721	41.9272	0.9972	4.1235	41.9780	0.9878	6.2399	40.1790	0.9824
LSB Grayscale	0.0091	68.3163	0.9999	0.0090	68.0764	0.9999	2.3906	44.1388	0.9806
SSB-4 Gray Scale	0.0219	64.5051	0.9999	0.0244	63.7639	0.9999	12.0203	37.3316	0.9122
Grayscale Pseudo-Random Permutation	0.2501	53.9414	0.9999	0.1052	57.4182	0.9998	3.9873	42.0898	0.9539
DCT	5.9931	40.3542	0.9987	5.8939	40.4267	0.9961	5.9536	40.3829	0.9950
FFT	30.5323	33.28	0.9863	58.1736	30.4835	0.9270	103.9995	27.9604	0.8814

The best values for quality measures were obtained with grayscale LSB, grayscale SSB-4 and pseudo random permutation of the bit index, due to the fact that both mainly modify the least significant bits of the value representing the intensity of each pixel in the image. The other techniques based on gray scale presented reasonable results in the metrics obtained from the experiments carried out.

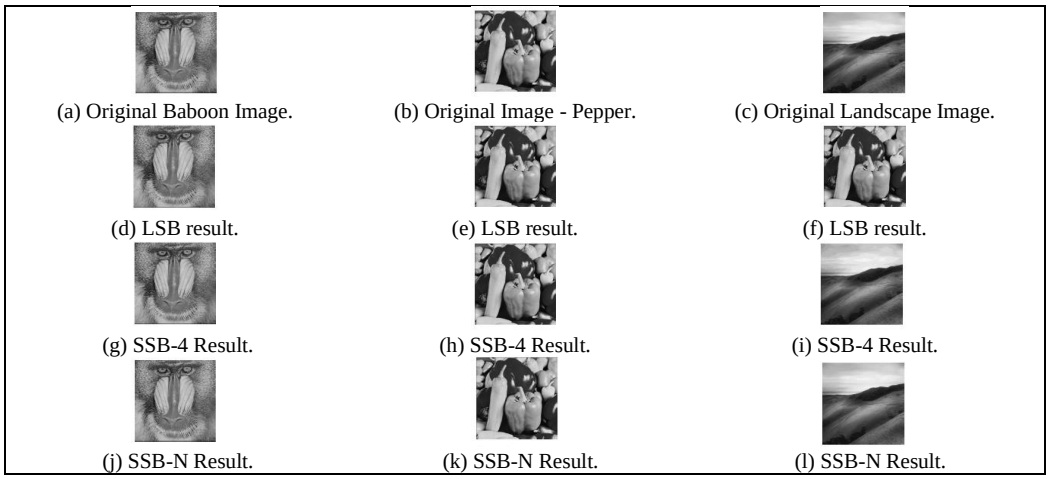

Figure 6: Result of the application of steganography techniques digital image in three grayscale images.

The DCT technique changes the least significant bits of the coefficients found when calculating the discrete cosine transform of the image. In this work, only the DC coefficient was changed,

preventing data loss during the compression process and minimizing image distortion. The FFT technique is equivalent to the DCT technique, but here the discrete Fourier transform is used. Observing the data in Table 2, it is noted that this technique presents a greater distortion than the DCT technique.

4. Conclusions

This work presented a comparison between the methods LSB, LSB Gray Scale, SSB-4 Gray Scale, SSB-N Gray Scale, DCT and FFT for use in digital steganography through quality measures such as MSE, PSNR and SSIM. The SSB-N method was proposed in this work to ensure more security in the storage of secret messages and to obtain better results for quality measures compared to the SSB-4 method, on which this method is based. In future works, it is intended to propose algorithms to correct errors occurred in the recovery of secret messages using the DCT and FFT techniques, as well as to carry out studies using steganography algorithms associated with algorithms to correct existing errors.

References

Alaa Jabbar, Q., & Farah Qasim Ahmed, A. (2021). History of Image Digital Formats Using in Information Technology. *QALAAI ZANIST JOURNAL, 6*(2), 1098-1112. doi:10.25212/lfu.qzj.6.2.41

Alwan, Z. A., Farhan, H. M., Mahdi, S. Q. J. I. J. o. E., & Engineering, C. (2020). Color image steganography in YCbCr space. *10*(1).

Alyousuf, F. Q. A., Din, R., Qasim, A. J. J. B. o. E. E., & Informatics. (2020). Analysis review on spatial and transform domain technique in digital steganography. *9*(2), 573-581.

Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM systems journal, 35*(3.4), 313-336.

Dhawan, S. (2011). A review of image compression and comparison of its algorithms. *International Journal of Electronics & Communication Technology, IJECT, 2*(1), 22-26.

Din, R., Qasim, A. J., Abdullah, S., Elias, S. J. J. I. J. o. E., & Technology. (2019). Analysis Review on Image Compression Domain. *8*(1.7), 293-296.

Din, R., Qasim, A. J. J. B. o. E. E., & Informatics. (2019). Steganography analysis techniques applied to audio and image files. *8*(4), 1297–1302.

Kermani, Z. Z., & Jamzad, M. (2005). *A robust steganography algorithm based on texture similarity using gabor filter*. Paper presented at the Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.

Khudher, I. M. J. E.-E. J. o. E. T. (2021). LSB Steganography Strengthen Footprint Biometric Template. *1*(9), 109.

Liao, X., Wen, Q.-y., & Zhang, J. (2011). A steganographic method for digital images with four-pixel differencing and modified LSB substitution. *Journal of Visual Communication and Image Representation, 22*(1), 1-8.

Manjula, G., & Danti, A. J. a. p. a. (2015). A novel hash based least significant bit (2-3-3) image steganography in spatial domain.

Mulla, A., Gunjikar, N., & Naik, R. (2013). Comparison of Different Image Compression Techniques. *International Journal of Computer Applications, 70*(28).

Qasim, A. J., & Alyousuf, F. Q. A. J. Q. Z. J. (2021). History of image digital formats using in information technology. *6*(2), 1098-1112.

Qasim, A. J., Din, R., Alyousuf, F. Q. A. J. B. o. E. E., & Informatics. (2020). Review on techniques and file formats of image compression. *9*(2), 602-610.

QASSIM, A. J., & SUDHAKAR, Y. (2015). Information Security with Image through Reversible Room by using Advanced Encryption Standard and Least Significant Bit Algorithm.

Roshidi Din, O. G., Alaa Jabbar Qasim. (2018). Analytical Review on Graphical Formats Used in Image Steganographic Compression. *Indonesian Journal of Electrical Engineering and Computer Science, Vol 12, No 2*, pp. 441~446. doi: 10.11591

Sayood, K. (2017). *Introduction to data compression*: Morgan Kaufmann.

Sidhik, S., Sudheer, S., & Pillai, V. M. (2015). Performance and analysis of high capacity steganography of color images involving wavelet transform. *Optik, 126*(23), 3755-3760.

Vijay, K., Jayareka, K., Kirubasri, G., & Vijay, P. J. A. o. t. R. S. f. C. B. (2021). Enhancing the Security of Data Using Digital Stemage Technique. *25*(6), 9138-9143.