Journal of Engineering and Sustainable Development
www.jeasd.org
Vol. 21, No.01, January 2017
ISSN 2520-0917

SELECTED LEAST SIGNIFICANT BIT APPROACH FOR HIDING INFORMATION INSIDE COLOR IMAGE STEGANOGRAPHY BY USING MAGIC SQUARE

Sahar Mahdie Klim

Misan University-Engineering college, Misan, Iraq

Abstruct : The main goal of steganography is to hide the existence of any secret data from the eye of third party, so the resultant stego-image must appear normal and not suspicious after the impeding process. Stego-image must have an acceptable quality in comparison with its size. Moreover, its size has to be on the rate of the usual Internet images, because it is required to be sent by this medium. This work focuses on these problems, using a novel Selected Least Significant Bit method. Standard color images were used to hide the data. The proposed approach has been evaluated and the results had been analyzed. The obtained results showed clear enhancements on the impeding space considering previous related works.

Keywords: *Steganography, Image domain, Least Significant Bit, Selected Least Significant Bit.*

طريقة مبتكرة لإخفاء المعلومات داخل صور ملونة باستخدام البت الأقل أهمية الأخير

الخلاصة: الهدف الرئيسي من تقنية "الستيجانوجرافي" هي إخفاء وجود أية بيانات سرية عن أعين أي طرف ثالث، لذا الصورة الناتجة عن عملية الإخفاء يجب أن تكون صورة طبيعية، وليست صورة غريبة. فيجب أن يكون لديها نفس حجم الصور المرسلة عبر الإنترنت، بالإضافة إلى النوعية المعتادة في هذه البيئات. هذا العمل يركز على هذه المشاكل، باستخدام طريقة البت الأقل أهمية المحدد. صور معيارية ملونة تم استخدامها لإخفاء بيانات. الطريقة المقترحة تم تقييمها والنتائج تم تحليلها. وقد أثبتت فعاليتها وجودتها بزيادة حجم السعة التخزينية في الصورة. مقارنة بأعمال سابقة.

1. Introduction

Maintaining privacy in personal communications is something everyone desires, so the security of information and data is one of the most important factors of communication technology. Cryptography is the field of technologies for hiding information. It tries to hide, encrypt, the information in such a way that a third party who has access to the hidden, encrypted, data cannot reconstruct or decrypt, the original information.

* sahar_mahdi@uomisan.edu.iq

Steganography is the art and science of invisible communication [1][2]. This is accomplished through hiding information in other information in such a way that no one apart from the intended recipient knows of the message existence, thus hiding the existence of the communicated information [1]. The word steganography is derived from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" [3] defining it as "covered writing". In image steganography, the information is hidden exclusively in images.

The idea of hiding information has a long history. In Histories the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son-in-law in Greece. He shaved the head of one of his most trusted slaves and tattooed the message onto the slave's scalp. When the slave's hair grew back, the slave was dispatched with the hidden message [4].

The categories of steganography classified depending on the cover format, which hides the secret data. The best file formats that are more suitable for steganography are those with a high degree of redundancy. Redundancy can be defined as data or bits that could be omitted without loss of meaning or function; repetition or superfluity of information. Figure 1 shows the four main categories of file formats that can be used for steganography as a cover for hiding any appropriate digital data as a secret.

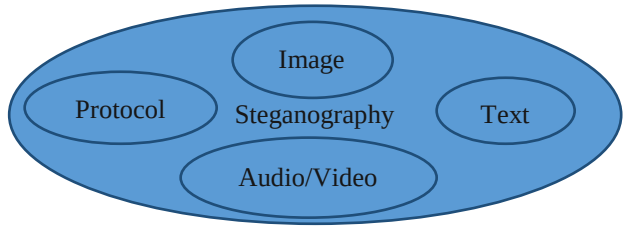


Figure 1: Categories of steganography [1][2]

Digital images are the most popular cover objects for steganography, because of their proliferation especially on the Internet, and the large amount of redundant bits present in the digital representation.In the domain of digital images many different image file formats exist, most of them for specific applications. For these different image file formats, different steganography algorithms exist.

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain [4]. Image – also known as spatial – domain techniques embed messages in the intensity of the pixels directly, while for transform – also known as frequency – domain, images are first transformed and then the message is embedded in the image [1].

Image domain techniques apply bit insertion. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format.

Transform domain techniques and methods hide messages in more significant areas of the cover image making it more robust. Many transform domain methods are independent of the image format, those methods is suitable for lossy and lossless compression.

Least Significant Bit (LSB) insertion is a common, simple approach to embed information in a cover image on the image domain [5]. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue (RGB) color components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data. For example a grid for 3 pixels of a 24-bit image can be as follows,[5]:

(00101101 00011100 11011100)
(10100110 11000100 00001100)
(11010010 10101101 01100011)

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

(00101101 00011101 11011100)
(10100110 11000101 00001100)
(11010010 10101100 01100011)

Least Significant bit (LSB) is the simplest form of bit insertion, it could be improved to Least Significant bits (LSBs) to increase the capacity, and a several methods can be used to improve the security.

Image domain methods or Transform domain methods have different strong and weak points. Image domain methods are simple approach than transform domain methods, and image domain least significant bit (LSB) technique in (RBG) colored images like PNG images and BMP images is most suitable for applications where the focus is on the amount of information to be transmitted and not on the secrecy of that information [1].

The following table compares least significant bit (LSB) insertion in RBG colored image and in GIF files and JPEG compression steganography: [1].

Table 1: Comparison of image steganography algorithms [1].

	LSB in RGB	LSB in GIF	JPEG Compression
Invisibility	High*	Medium*	High
Payload Capacity	High	Medium	Medium
Robustness Against Statistcal Attacks	Low	Low	Medium
Robustness Against Image Manipulation	Low	Low	Medium
Independent Format	Low	Low	Low
Unsuspecious Files	Low	Low	High

*Depends on Cover Image Used

On LSB principle, there are many methods and techniques developed to increase the capacity and the quality of the cover image and to increase the security of the hidden data. This data could be anything, text or pixels of a secret image etc.
When working especially with colored images as a secret message, which contains huge data, and in the same time it contains redundant data, it will be a good idea to reduce the unwanted or the not useful data in a way that doesn't affect the purpose that is needed from sending the secret image.

This paper will work with color images in order to propose a contributed Selected Least Significant Bit (SLSB) method for hiding information inside images. The methodology and the experimental results are shown in this work.
Besides this section, the second section will review some related works, in order to reach contributed and unique design, the third section explains the proposed methodology, followed with the experiments and the obtained results, which will be analyzed in the fourth section. Finally yet importantly, the concluded points will be discussed in the fifth section.

2. Related Works

Steganographic techniques are classified based on their mode of operations and thus fall into one of these categories; substitution or replacement, transformation domain, statistical, spread spectrum, and distortion [6]. Just as the name suggests, substitution techniques involve replacing a certain part of the digital file with the required piece of information that need to be hidden. Transformation domain, on the other hand, encompasses a process whereby the information that requires concealing is hidden in a frequency space with the file component. Statistical steganographic techniques involve changing the statistical elements of the digital file using various statistical algorithms. Communication using spread spectrum is also another popular steganography technique that hides and recovers a message of substantial length within digital imagery while maintaining the original image size and dynamic range, while distortion involves altering the signal carrying the information and later making a comparison with the original medium content [6].

Xiaolong Li et. Al. [7] proposed the Difference Expansion (DE) technique, which is a lossless technique to assist in imbedding the data inside cover images with high-capacity and high-visual quality ways. This method was based on exploring the redundancy in digital images in order to benefit from the redundant information in increasing the capacity in addition to reducing the distortion.
By calculating the neighboring pixels values difference in order to gain the Difference Expansion (DE), then used the SLSB method to select the embedding area, and imbedded the original values of this area and the payload at the same time. This method imbedded the original values in order to have exact recovery of the image and lower the image distortion. In order to solve the large amount of embedded data (payload and

original values), this work used lossless compression technique before the imbedding process.

Using standard images, he got 34dB Peak Signal to Noise Ratio(PSNR) as an average value for 11 test with different payload sizes, between 39566b and 516794b. This technique showed better PSNR for lower payload size.

Hao Luo et. al. [8] proposed a median-based technique to enhance the imbedding capacity and marked image's quality. Their method scanned the image looking for high-correlation corresponding block pixels, in order to obtain a difference histogram. Considering the integer median of the selected blocks, multi-level shifting of the histogram is used for the embedding phase.

They grouped the measured blocks into four categories based on the related method; the goal of this process is to keep the calculated median during the impeding phase. In the receiver side, those median pixels are extracted at the beginning then the payload is retrieved. Moreover, the cover image will be restored to its originality with very low distortion using the inverse histogram multi-level shifting technique.

Their technique resulted in average PSNR of 48dB. Using different standard images, with three block-partitioning sizes of (2x2, 3x3, 4x4) showing that the PSNR has better values for lower block partitioning size.

On the other hand, Chang et al concentrated in their work [9] on Side Matching Vector Quantization (SMVQ), to make it easy for the receiver, which needs only two steps to extract the payload and restore the cover image. They supposed that original compression code should be restored at the time of payload extraction, in order to be used by the receiver. They obtained relatively low PSNR with average value of 28db using standard images as cover for their payload.

They enhanced their method to obtain better results in their work [10], which was also based on Vector Quantization (VQ) image compression technique. They adopted the VQ compressed code, that was previously categorized into three clusters, to fulfill data recovery and secret concealment, using techniques of trio extension and frequency clustering,

This way allowed them to expand the payload imbedding capacity in the same standard image. By doing so, they enhance the PSNR to average value of 32dB. Which is still low value considering the previously mentioned works.

Tsing et al in their work [11] proposed a technique to enhance the restoration quality of both the payload and the cover image. Using a prediction error expansion technique, their proposed approach determined the predictive values of the imbedding areas in the image. Then the imbedding process contains exploiting the difference expansion between each pixel and its predictive value. Using standard images, they reached average value 45dB for PSNR.

3. Proposed Method

In this section, we propose a steganography spatial domain algorithm. We will describe our research methodology for hiding text inside an image. There are some terms used in this research: cover image, payload and stego image. The cover image is the original image used to hide secret information or messages and can be in any format and dimensions, the payload is the text we want to conceal and in our case will be converted into a stream of bits, and stego image is the image resulted from combining the cover image and the payload. Our methodology divided into two phases; the first phase will discuss the proposed algorithm used to hide the text in an image, while the second discusses how to retrieve the text from stego image.

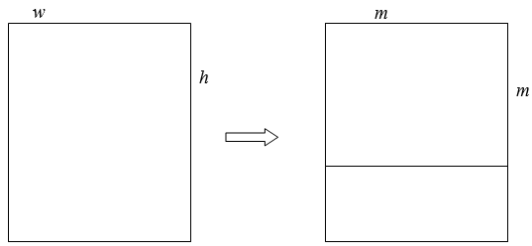
3.1 *A Proposed Steganography Algorithm for Hiding Secret Text in an Image*

This section will clarify the proposed algorithm; our proposed algorithm is an enhancement for SLSB. The proposed algorithm used a cover image to hide secret information or text, it works on the $m \times m$ part of the cover image, where m is an odd number. The enhancement algorithm has two strength points; the secret text is scattered among the image using magic square order, which increased the complexity of the algorithm, and the text bits are modified by their Xor with the corresponding SLSB value. Therefore, the first step is to define the value of m. Initially select a square part of the cover image, if we have a $w \times h$ cover image; where w: width and h:height, then m value will be set regards to the equation (1) that determine the value of m:

$$m=min(w,h) \quad (1)$$

$if\ m\ is\ an\ even\ number\ then$

$m = m - 1$

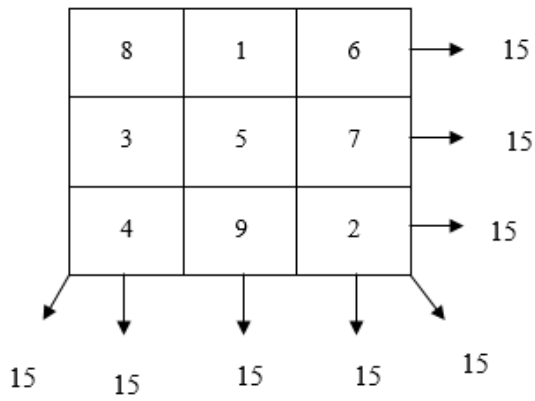


$m = min(w, h) = w$

The aim of defining the $m \times m$ part of the image for steganography processing is to apply magic square on it, then use its arrangement number for the sequence of pixels for steganography use.

A magic square is a 2 dimensional array, in which the sum of each row, column or diagonal are equal. Magic square has the same number of columns and rows; it is

possible to construct a magic square for any number except for 2. Constructing a magic square for odd number is easy, so we will use it in our methodology. The following are example of 3×3 magic square:



The method to construct an odd magic square is too simple, starting by filling number 1 in the middle column, in the first row. Then goes diagonal up and right, if it leave the square, then it is wrapped around to the last row of the first column, else if it encounters a filled square then it goes step down, and continue to fill the magic square.

Now, the next step after finding the sequence of pixels, is filtering them by using a threshold value, the pixels whose obtained the threshold, can be used to contain the payload. We suggest multiple threshold values range from (248 down to 16), the largest number of threshold value that can fit the payload depend on its length will be chosen. Then the threshold value will depend on three factors; payload length, the image size, and the SLSB used (1 bit, 2 bit, or 3 bit). The following example will clarify the threshold selection value:

Let an image have a dimension of 1021 × 1021, and then we have 1042441 pixels. Moreover, have a payload of 100-character length. Each character represented by 8 bits, if we use SLSB with 2 bits, then each character requires 4 pixels to represent it in the image, which mean that at least 400 pixels of the cover image should obtain the threshold value. Here we will pick the largest threshold value that have at least 400 pixels obtain the threshold value.

The choice of the color to hide the information is determined by an operation known as pair analysis that determines the color with the greatest diversity ratio[12], in order to achieve an enhanced image quality through applying filter algorithms as we mention it in the literature review chapter.

To compute the diversity ratio for the colors, we calculate the standard deviation for each colors (red, green, and blue), the color has the highest standard deviation will be chosen for hiding payload in it, and the second highest standard deviation value will chose for the XOR process. Standard deviation calculated according to equation (2).

$$\sigma = \frac{\sqrt{\sum_{i=1}^{n}(x_i-\mu)^2}}{n} \quad (2)$$

The threshold value will be checked for the same color choosing by pair analysis. After choosing the threshold value, the pixels sequence are chosen depending on magic square. The first three bytes will contain the threshold value, the length of the payload, and the color. Then the fourth byte contains the first character of the payload and so on. Only the pixels that obtained the threshold will use for hiding the payload, the pixel value will change by XOR the payload bits with the least significant bit of the second greater diversity ratio color by pair analysis.

The resulting image will be the stego image. The following example shows how to hide a message ("a") inside a 3×3 square magic, after performing pair analysis the result shows that the blue color has the highest diversity ratio, and the red color has the second highest diversity ratio:

The "a" representation in binary is (01100001).

8	1	6
3	5	7
4	9	2

→ (RGB)

(10010101 00101110 10101100)

10010101 stands for red values, 00101110 stands for green values, and 10101100 stands for blue value, according to pair analysis results, the blue color will used for hiding the payload, while the red color will used for XOR. The threshold value should be first checked for the blue value (depends on pair analysis results of diversity ratio), let suggest that the threshold value for blue is (128), then pixel number two obtained the threshold condition, and can be used for hiding the text in it. We take the first two bits of"a" (01), and XOR it with the two least significant bits of red color for the same pixel (01) as follows:

XOR:

01
01
00

Then the value will be stored in pixel number 2 will be (10010101 00101110 10101100). Here we note that the pixel value not changed. Then we continued the same steps, until the whole payload are inserted. Figure 2 shows the structure for the algorithm for hiding text in the image.

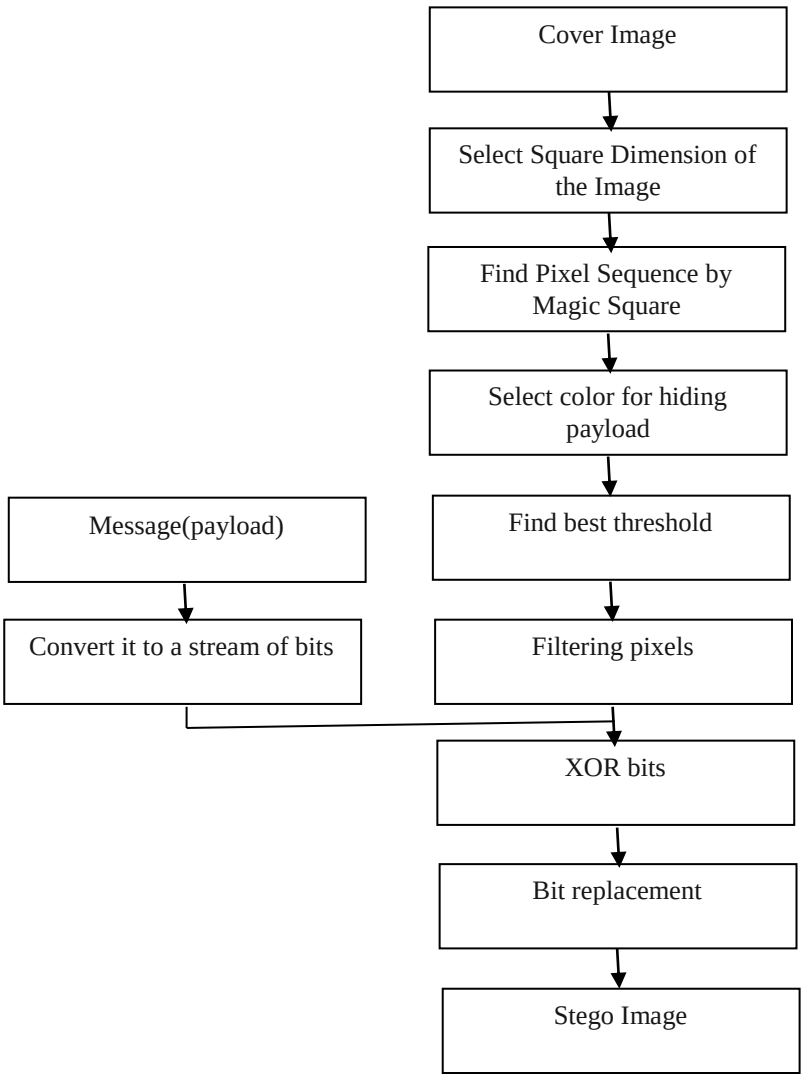
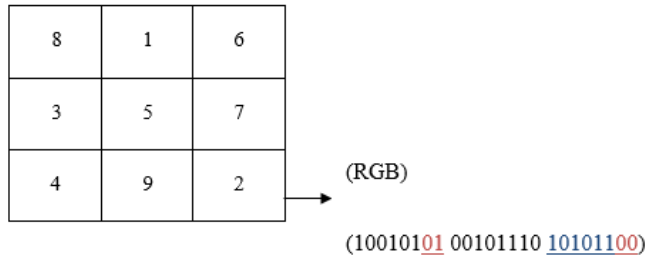


Figure 2: Block Diagrm of Stegonagraphy algorithm.

3.2 Retrieving Payload From Stego Image

In this section, we discuss how to retrieve secret messages (payload) hidden in stego image. As the first step in hiding the payload into a cover image, we find the stego image dimensions by taking the square dimensions of the image, where it is an odd number. Then, the sequence of pixels will be picked regarding to the magic square. The first three bytes will retrieve the payload length, color, the threshold value. Then each pixel in the magic square sequence will be checked if it obtained the threshold value. We notice that the modification of bits will not affect the threshold value. If the pixel obtained the threshold, then we will reverse the process in hiding the payload by XOR 2 bits of red and blue color. The following shows how we retrieve the payload bit from the previous example:



From pixel number 2 that represents (10010101 00101110 10101100), we will XOR the least significant two bits of the red color, with the least significant two bits of blue color, the result should retrieve the payload bits.

XOR:

01
00
01

We got the first two bits of the payload, then we select the next pixel regarding of the magic square sequence, and check the threshold value for it, these steps are repeated until we exceed the payload length specified in the image. Figure 3 shows the structure for retrieving text from the stego image.

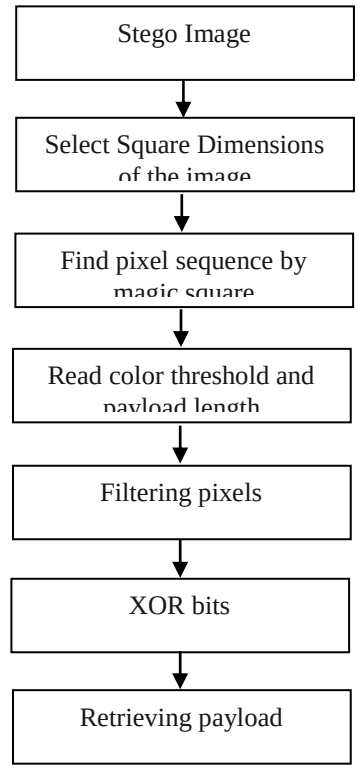


Figure 3: block diagram for retrieving payload.

4. Experimental Results

This section will show the experiments and the obtained results after applying the proposed approach on standard RGB images.

The proposed approach was applied on the below standard images [13], as shown below, there are no difference between the original images and the stego images after applying the proposed method and carrying the payload. The Peak Signal to Noise Ratio (PSNR) for the images are shown in the below table.

Figure 4: Lena - original cover image size 256*256

Figure 5: Lena - Stego cover image size 256*256

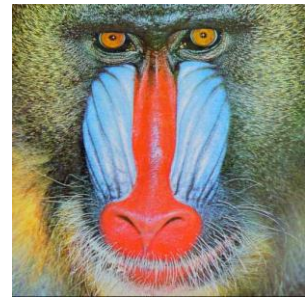


Figure 6: Baboon - Original Cover image size 256*256

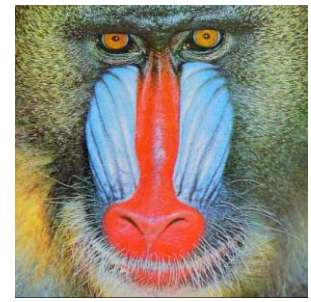


Figure 7: Baboon - Stego cover image size 256*256

Figure 8: Airplane (F16) - original cover image size 256*256

Figure 9: Airplane (F16) - Stego cover image size 256*256

Figure 10: Pepper - original cover image size 256*256

Figure 11: Pepper - Stego cover image size 256*256

Figure 12: House - original cover image size 256*256

Figure 13: House - Stego cover image size 256*256

Figure 4 shows the cover original Lena image with size 256 *256 that used to hide the message. Figure 5 shows the cover stego Lena image after carrying the payload.
Moreover, Figure 6 shows the cover original Baboon image with size 256 *256 that used to hide the message. Figure 7 shows the cover stego Baboon image after carrying the same payload.

On the other hand, Figure 8 shows the cover original Airplane image with size 256 *256 that used to hide the message. Figure 9 shows the cover stego Airplane image after carrying the same payload.

Moreover, Figure 10 shows the cover original Pepper image with size 256 *256 that used to hide the message. Figure 11 shows the cover stego Pepper image after carrying the same payload.

Finally, Figure 12 shows the cover original House image with size 256 *256 that used to hide the message. Figure 13 shows the cover stego House image after carrying the same payload.

As shown in figures (above), the stego images are similar as cover with no difference, however, the stego contain the secret data inside them.
After applying the algorithm, a comparison of the PSNR between the original cover and stego images will be shown in the table below.

Table 2: Cover-Stego PSNR

Cover image	Stego image	PSNR
Lena	LenaS	40.7756
Baboon	BaboonS	40.7194
Airplane	AirplaneS	40.7888
Pepper	PepperS	40.6278
House	HouseS	40.7293

Moreover, the Mean Square Error, between the cover and stego images are also measured and shown in the table below.

Table 3: Cover- Stego MSE

Cover image	stego image	MSE
Lena	LenaS	10.72
Baboon	BaboonS	9.27
Airplane	AirplaneS	10.81
Pepper	PepperS	9.11
House	HouseS	9.88

On the Other hand, to get more evaluations, the results of correlation between the cover and stego images are measured, and shown in the below table.

Table 4: Cover-Stego Correlation Results

Cover image	Stego image	correlation
Lena	LenaS	0.0028
Baboon	BaboonS	0.0024
aIRPLANE	AirplaneS	0.0029
Pepper	PepperS	0.0022
House	HouseS	0.0025

After applying the algorithm, a comparison of the histogram between the original cover and stego images will be shown in figures to evaluate the proposed methodology regarding the mentioned standard color images.

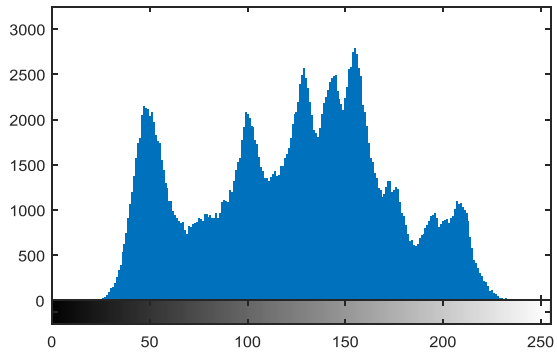


Figure 9: Lena - Cover Image Histogram

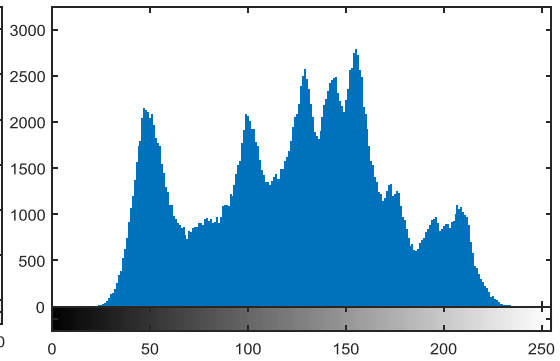


Figure 10: Lena - Stego Image Histogram

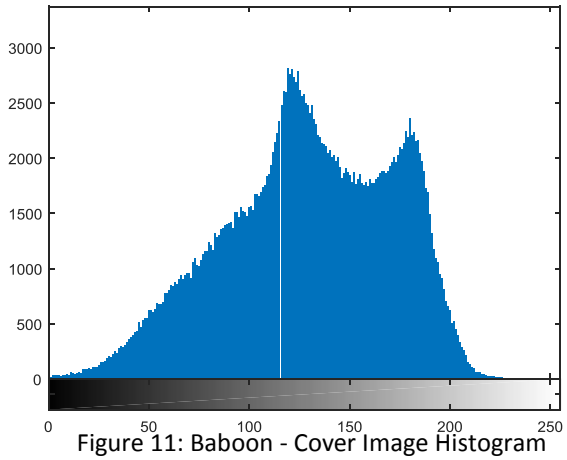


Figure 11: Baboon - Cover Image Histogram

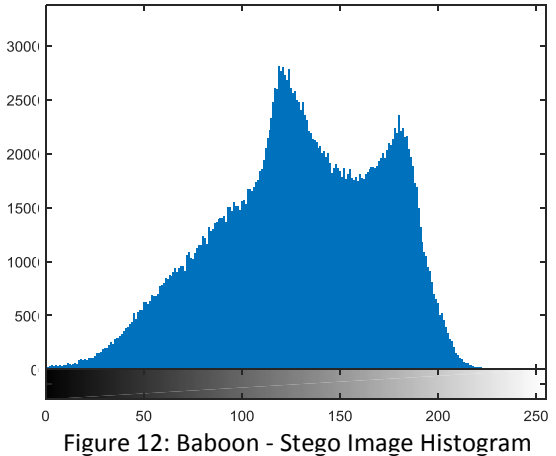


Figure 12: Baboon - Stego Image Histogram

Figure 9, above shows the histogram of the cove Lena image, on the other hand Figure 10, shows the histogram of the Stego image, which shows no difference. However, Figure 11 shows the histogram of the cover Baboon image. Figure 12, shows the histogram of Baboon stego image, which also does not show any difference. For the other pictures we did not show their histograms as they have the same results.

5. Comparison with Previous Works

Table 5: Average PSNR Comparison for Stego Cover Images

Method	Lena	Baboon	Airplane	Pepper	House
Thanikaiselvan Et Al [14]	40.1	39.6	-	40.1	-
Thanikaiselvan, Et Al [15]	-	40.0	-	40.0	-
Pria Et Al [16]	41.0	37.0	-	-	-
Proposed Algorithm	40.8	40.7	40.8	40.6	40.7

As shown from the table above, the proposed method, has enhanced results over most of the previous works.

6. Conclusions

This work studied the problem of hiding information inside image steganography; this work proposed a contributed Selected Least Significant Bit (SLSB) method, which was applied on color images. The used images were Lena, Baboon, Airplane, Pepper, and House, which are standard images. The proposed approach enhanced the storage capacity in comparison with related works using this(The proposed method) to enhance the payload capacity of color images the entire image at the same time. Which showed real enhanced results and capacity; moreover, it resulted in no difference between the original image and the stego image with the payload inside. PSNR, MSE, Correlation and Histogram of both cover and stego images for Lena and Baboon standard color images, were measured, which evaluated the proposed contributed approach.

6. References

1. Poornima, R., and R. J. Iswarya. (2013). "An overview of digital image steganography." International Journal of Computer Science and Engineering Survey 4, no. 1: 23.
2. T. Morkel , J.H.P. Eloff , and M.S. Olivier. (2005). ” An Overview of Image Steganography”, Information and Computer Security Architecture (ICSA) Research Group, South Africa Conference (ISSA 2005) Ed. By H.S. Venter et al.
3. Aravindh, S., and S. Karthikeyan. (2013). "Steganography and Steganalysis." Digital Image Processing 5, no. 8: 377-379.

4. Hamid, Nagham, AbidYahya, R. Badlishah Ahmad, and Osamah M. Al-Qershi. (2012). "Image steganography techniques: an overview." International Journal of Computer Science and Security (IJCSS) 6, no. 3: 168-187.
5. Gupta, Shailender, Ankur Goyal, and Bharat Bhushan. (2012). "Information hiding using least significant bit steganography and cryptography." International Journal of Modern Education and Computer Science 4, no. 6: 27.
6. Sumathi, C. P., T. Santanam, and G. Umamaheswari. (2014). "A Study of Various Steganographic Techniques Used for Information Hiding." arXiv preprint arXiv:1401.5561.
7. Li, Xiaolong, Jian Li, Bin Li, and Bin Yang. (2013). "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion."Signal processing 93, no. 1: 198-205.
8. Luo, Hao, et al. (2011). "Reversible data hiding based on block median preservation. "Information Sciences 181.2: 308-328.
9. Qin, Chuan, Chin-Chen Chang, and Yen-Chang Chen. (2013). "Efficient reversible data hiding for VQ-compressed images based on index mapping mechanism." Signal Processing 93, no. 9: 2687-2695.
10. Lee, Jiann-Der, Yaw-Hwang Chiou, and Jing-Ming Guo. (2013). "Lossless data hiding for VQ indices based on neighboring correlation." Information Sciences 221: 419-438.
11. Tseng, Hsien-Wen, and Chi-Pin Hsieh. (2009). "Prediction-based reversible data hiding." Information Sciences 179.14: 2460-2469.
12. Eves, H. W. "Mathematical circles squared: a third collection of mathematical stories and anecdotes", Prindle Weber & Schmidt. (1972).
13. Image Processing and Analysis and JAVA: Available on: http://imagej.nih.gov/ij/images/. Adopted on (2016).
14. Thanikaiselvan, V., et al. (2012). "Horse riding & hiding in image for data guarding."Procedia Engineering 30: 36-44.
15. Thanikaiselvan, V., S. Subashanthini, and Rengarajan Amirtharajan. (2015). "PVD based steganography on scrambled RGB cover images with pixel indicator." Journal of Artificial Intelligence 7.2 (2014): 54.
16. Muhammad, Khan, et al. "A novel image steganographic approach for hiding text in color images using HSI color model." arXiv preprint arXiv:1503.00388.