COLOR IMAGES STEGANOGRAPHY BY IMPLEMENTING SIMPLE LOGICAL OPERATIONS USING ASELECTED STEGANOGRAPHIC FACTOR

Adnan Manasreh[1], Mohammad S. Khrisat[2], Hatim Ghazi Zaini[3] and Ziad A. Alqadi[2]
[1]Department of Electrical Engineering, Applied Science Private University, Amman, Jordan
[2]Department of Computer Engineering, Faculty of Engineering Technology, AL-Balqa Applied University, Amman, Jordan
[3]Computer and Information Technology College, Taif University, Taif, Kingdom of Saudi Arabia
E-Mail: adnan_m@asu.edu.jo

ABSTRACT

The use and circulation of digital images has spread recently, and the digital image can be personal or confidential, which requires preventing intruders from understanding it and therefore it must be protected. Steganography is one of the popular techniques used to protect secret images. In this paper research a simple method of data steganography will be presented, tested and implemented. The obtained results will be compared with LSB method to show how the proposed method will increase the efficiency and the capacity of data steganography keeping good value for the quality parameters MSE and PSNR.

Keywords: covering image, stego image, steganography, SF, MSE, PSNR, speed up.

1. INTRODUCTIO

Digital color images are one of the most widespread, widely used types of digital data, and they are currently used in many important vital applications [1], [2], [3]. The digital image has a huge amount of integer data, and with this, the process of manipulating it is very easy, because the color digital image is represented by a three-dimensional matrix.

The digital image may be confidential or of a special nature, or it may be carrying confidential information that requires protection from intruders or from any third party not authorized to view the image [4], [5]. [6].

To protect digital images, various data steganography methods are used, and here data steganography means hiding the image massage into a covering image to produce a stego image as shown in figure 1, the stego image can processed using the same method of data steganography to produce the extracted method as shown in Figure-2 [10], [11].

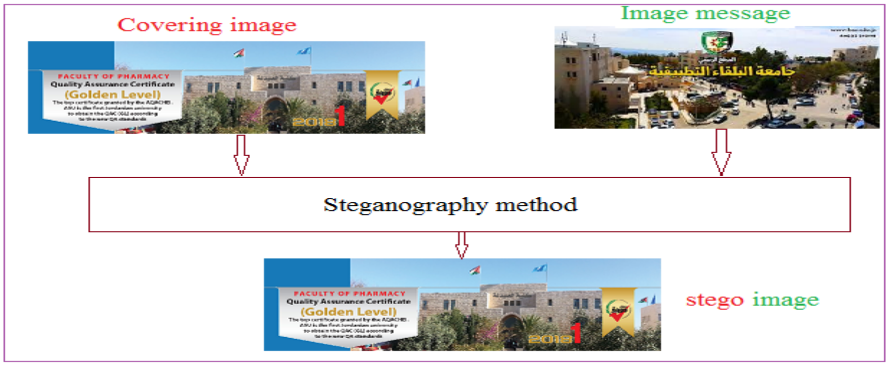


Figure-1. Hiding image message.

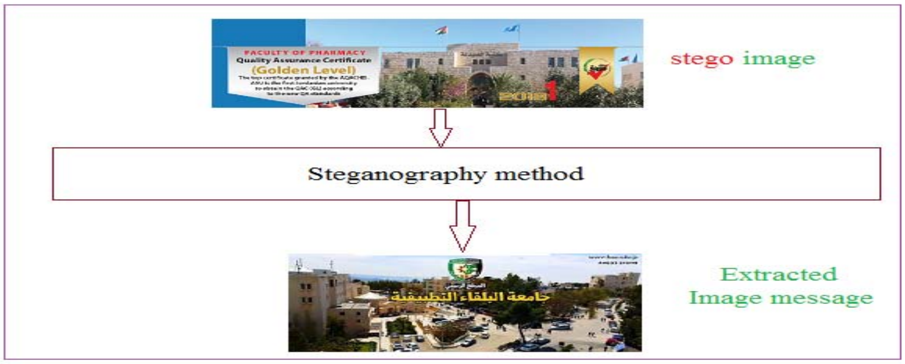


Figure-2. Extracting image message.

The selected method of steganography must satisfy the following conditions [12], [13], [14]:

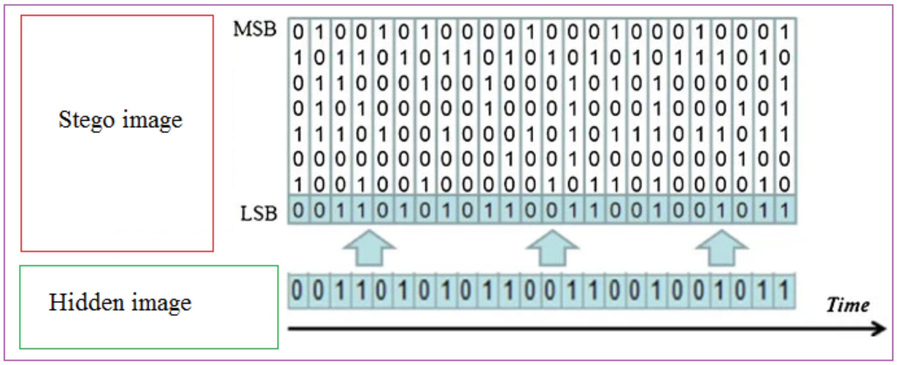
- The mean square error (MSE) (see equations 1 and 2) between the covering and the stego images must be very low, or the value of the peak signal to noise ratio (PSNR) must high, this indicates that the distortions in the image are not observed with the naked eye [15], [16].
- The mean square error (MSE) (see equations 1 and 2) between the message and the extracted images must be very low, or the value of the peak signal to noise ratio (PSNR) must high, this indicates that the extracted image is identical to image massage [17], [18].
- The hiding and extracting times must be very low to increase the used method efficiency [19], [20], [21].

MSE between messages S and R, n: message length

$$MSE_{SR} = \frac{1}{N} \sum_{j=0}^{n-1} [S(j) - R(j)]^2 \,, N = n \qquad (1)$$

$$PSNR_{SR} = 10 * \log_{10} \frac{(MAX_j)^2}{MSE_{SR}} \qquad (2)$$

Several methods are used to hide confidential data, and most of these methods depend on the least significant bit (LSB) method of data steganography. LSB reserved 8 bytes from the covering image to hide one byte from the image message, so the capacity of this method is limited to covering image size divided by 8 (see Figure-3)



MSB 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0
0 1 1 0 0 0 1 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 1
0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 1
1 1 1 0 1 0 0 1 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1
0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0
1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 1 0
LSB 0 0 1 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 0 0 1 0 1 1

0 0 1 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 0 0 1 0 1 1

Figure-3. LSB method data hiding.

LSB method adds a minor changes to the stego images, and each byte in the stego image will be incremented by 1 or decrementing by 1 or remaining the sane depending on the LSB of message byte to be hidden (see Figure-4),

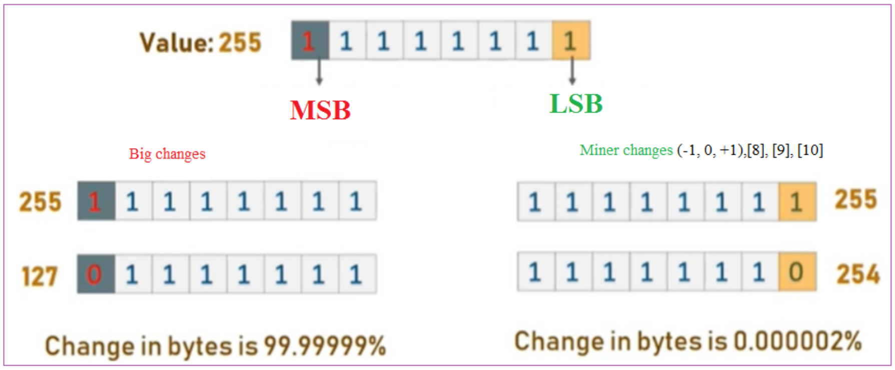


Figure-4. Changes in the stego bit.

LSB method provides good values for MSE and PSNR but it requires more time for hiding and extracting. LSB requires also covering image resizing if the capacity is less than the image message size multiplied by 8.

LSB method can be easily implemented applying the following steps:

- Get the sizes of the covering and message images.
- Find the capacity of the covering image, resize the image if needed.
- Covert the covering and the message images to binary.
- For each byte in the message image reserve 8 bytes from the covering image, adjust the LSB of them according the message byte bits values ,

Figures-5 and 6 show the result outputs of implementing LSB using images examples

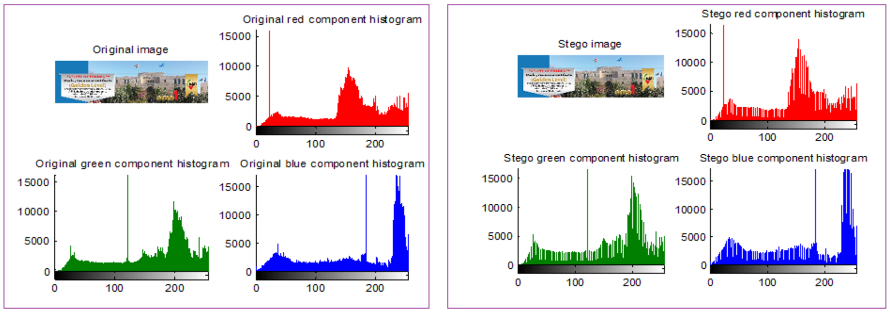


Figure-5. Covering and stego images using LSM method.

Figure-6. Message and extracted images using LSB method.

2. THE PROPOSED METHOD

This method is introduced to enhance the efficiency of data steganography by reducing the hiding and extraction times and keeping good values for the quality parameters MSE and PSNR; the method is based on using a steganographic factor. Steganographic factor (SF) is an integer value within the range 1 to 7, this value is needed to perform bit shifting to the left number of times depending on the results of subtracting FS from 8, the value of 0 was excluded because the extracted image will be black, and the value 8 was also excluded because the stego and extracted images will equal the image message to be hidden.

The proposed method can be implemented applying the following steps:

Hiding phase:

Step 1: Get the sizes of the covering and the message images; select the value of SF (between 1 and 7).

Step 2: Resize the covering image to match the image message size.

Step 3: Calculate f1 as a bit complement of:

```
f1=bitcmp(2^SF-1,8)
```

Step 4: Calculate f2 by anding the covering image x with f1:

```
f2=bitand(x,f1)
```

Step 5: Calculate f3 by bit shifting of the image massage y using the value of the selected SF:

```
f3=bitshift(y,SF-8)
```

Step 6: Get the stego image by Oring f2 and f3:

```
stego=uint8(bitor(f2,f3))
```

Extraction phase:

This phase can be implemented applying the following steps:

Step 1: Get the stego image and SF value.

Step 2: Calculate f5 by bit shifting the stego image:

```
f5=bitshift(Stego,8-SF)
```

Step 3: Get the extracted image by ANDing 255 with f5:

```
Extracted=uint8(bitand(255,f5))
```

This method was implemented and figure 7 shows an example output

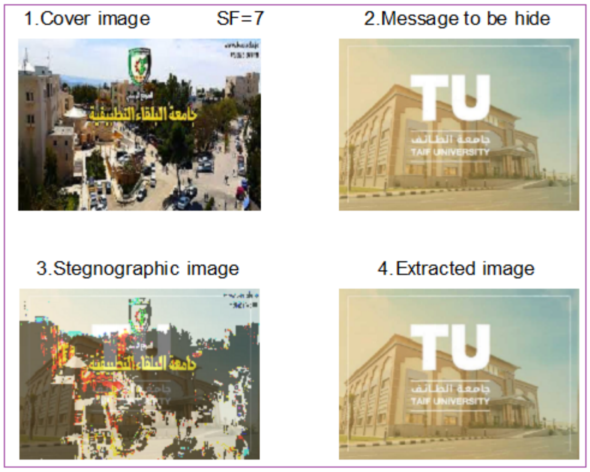


Figure-7. Proposed method implementation output example.

3. NUMERICAL EXAMPLES

Let us take the covering byte x=15 and message byte y120, for various values of SF (n), table 1 shows the results of calculation (case 1)

Table-1. Calculation results (case 1).

n	f1	f2	f3	f4	S	f5	f6	E
0	**255**	**15**	**0**	**15**	**15**	**0**	**0**	**0 excluded**
1	254	14	0	14	14	0	0	**0**
2	252	12	1	13	13	64	64	64
3	248	8	3	11	11	96	96	96
4	240	0	7	7	7	112	112	112
5	224	0	15	15	15	120	120	120 good
6	192	0	30	30	30	120	120	120 good
7	128	0	60	60	60	120	120	120 good
8	0	0	120	120	120	120	120	120 excluded

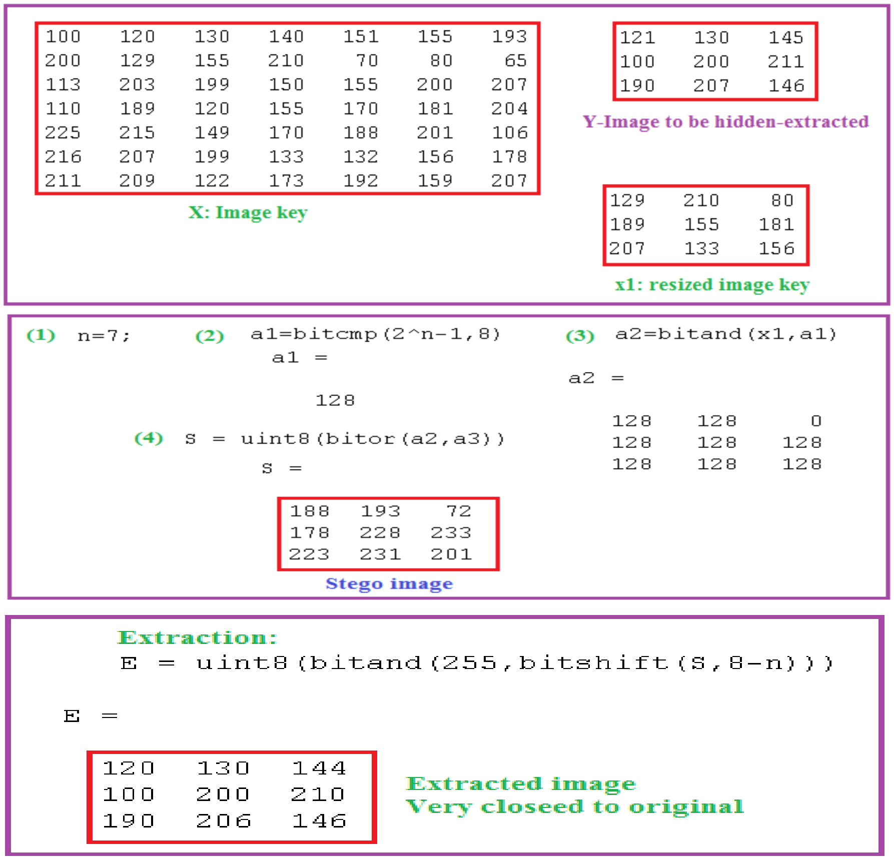
Now lets us select x equal to 151 and y equal to 239, table 2 shows the calculation results (case 2)

Table-2. Calculation results (case 2).

n	f1	f2	f3	f4	S	f5	f6	E
0	255	151	0	151	151	0	0	0 excluded
1	254	150	1	151	151	128	128	128
2	252	148	3	151	151	192	192	192
3	248	144	7	151	151	224	224	224
4	240	144	14	158	158	224	224	224
5	224	128	29	157	157	232	232	232 good
6	192	128	59	187	187	236	236	236 good
7	128	128	119	247	247	238	238	238 good
8	0	0	239	239	239	239	239	239 excluded

From Tables 1 and 2 we can see that when SF=0 the extracted image will be black, thus 0 value must be excluded, also when SF =8 the stego image will equal the hidden image and this value must also excluded, preferable values for SF are 5, 6, and 7 as shown in the previous tables.

Other examples were calculated using various matrices, figures



```
100   120   130   140   151   155   193
200   129   155   210    70    80    65
113   203   199   150   155   200   207
110   189   120   155   170   181   204
225   215   149   170   188   201   106
216   207   199   133   132   156   178
211   209   122   173   192   159   207
```

X: Image key

```
121   130   145
100   200   211
190   207   146
```

Y-Image to be hidden-extracted

```
129   210    80
189   155   181
207   133   156
```

x1: resized image key

```
(1)  n=7;        (2)  a1=bitcmp(2^n-1,8)        (3)  a2=bitand(x1,a1)
                        a1 =
                                                     a2 =
                            128
                                                          128   128     0
               (4)  S = uint8(bitor(a2,a3))               128   128   128
                        S =                               128   128   128

                         188   193    72
                         178   228   233
                         223   231   201
```

Stego image

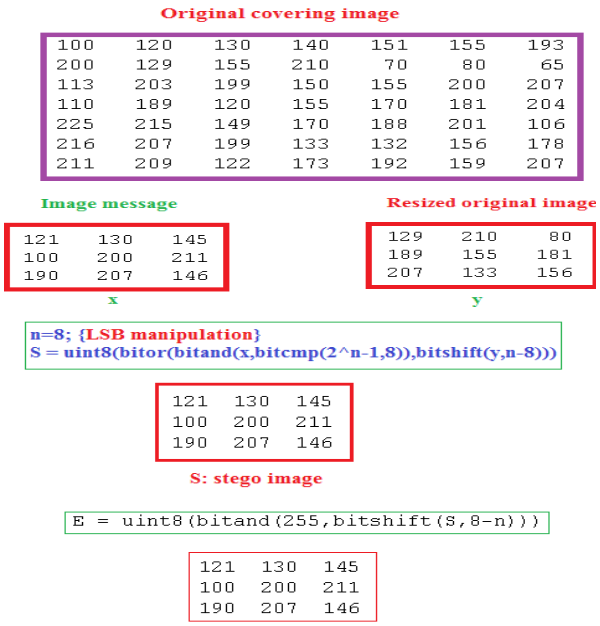
```
Extraction:
E = uint8(bitand(255,bitshift(S,8-n)))

E =

   120   130   144
   100   200   210
   190   206   146
```

Extracted image
Very closeed to original

Figure-8. Stego and extracted images when SF=7.



Original covering image

```
100  120  130  140  151  155  193
200  129  155  210   70   80   65
113  203  199  150  155  200  207
110  189  120  155  170  181  204
225  215  149  170  188  201  106
216  207  199  133  132  156  178
211  209  122  173  192  159  207
```

Image message

```
121  130  145
100  200  211
190  207  146
```

x

Resized original image

```
129  210   80
189  155  181
207  133  156
```

y

```
n=8; {LSB manipulation}
S = uint8(bitor(bitand(x,bitcmp(2^n-1,8)),bitshift(y,n-8)))
```

```
121  130  145
100  200  211
190  207  146
```

S: stego image

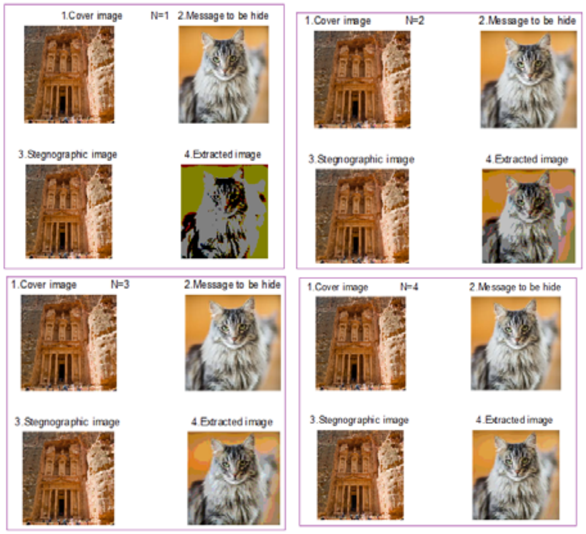
```
E = uint8(bitand(255,bitshift(S,8-n)))
```

```
121  130  145
100  200  211
190  207  146
```

Figure-9. Stego and extracted images when SF=8.

4. IMPLEMENTATION AND EXPERIMENTAL

The proposed method was implemented using various message images, the obtained results were compared with LSB method results. Figure-9 shows and example of hiding the cat image into Petra city image using various values of SF:



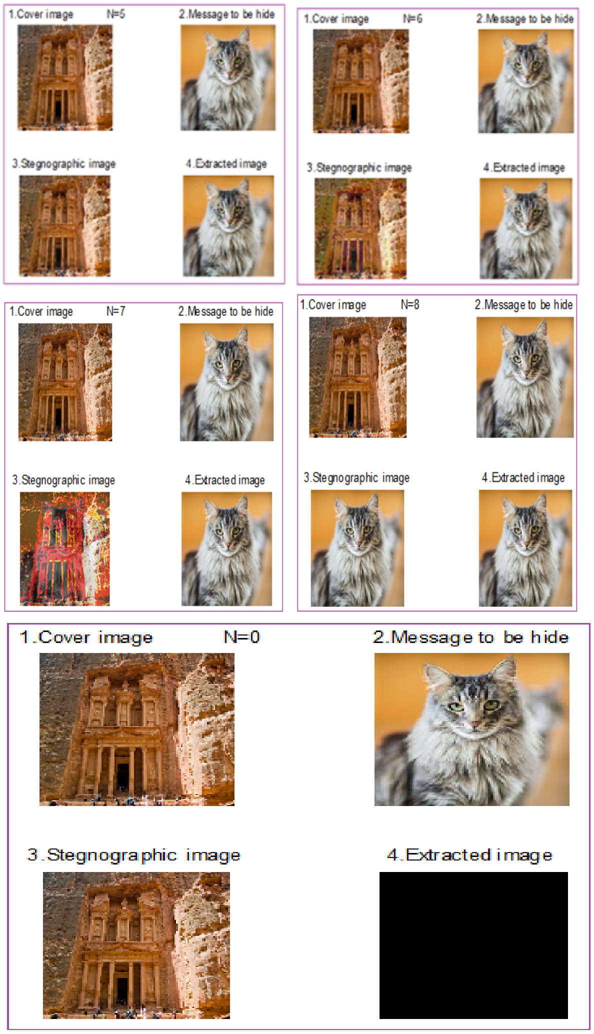


Figure-10. Hiding cat image into Petra city image using various values of SF.

Petra city image was taken as a covering image to hide various message images; Table-3 shows the obtained experimental results using LSB method:

Table-3. Quality factors using LSB method
Covering image size= 5140800 byte, capacity=5140800/8=642600 byte.

Image number	Size(byte)	Between holding and stego images		Between Hidden and extracted images		Resizing holding image
		MSE	PSNR	MSE	PSNR	
1	150849	0.1147	132.4788	0	infinite	no
2	77976	0.0767	136.5002	0	infinite	no
3	518400	0.2074	126.5561	0	infinite	no
4	4326210 (false capacity condition)	0.2557	124.4632	0	infinite	yes
5	122265	0.0904	134.8619	0	infinite	no
6	518400	0.1834	127.7868	0	infinite	no
7	150975	0.1230	131.7797	0	infinite	no
8	150975	0.1119	132.7296	0	infinite	no
9	151353	0.0991	133.9444	0	infinite	no
10	1890000 (false capacity condition)	0.2936	123.0810	0	infinite	yes

The same experiment was performed applying the proposed method; table 4 shows the obtained experimental results

Table-4. Quality factors using the proposed method, Covering image size= 5140800 byte, (SF=7).

Image number	Size(byte)	Between holding and stego images		Between Hidden and extracted images		Resizing holding image
		MSE	PSNR	MSE	PSNR	no
1	150849	1.5329e+003	37.4764	0.5032	117.6938	no
2	77976	3.5719e+003	29.0167	0.5083	117.5929	no
3	518400	2.5322e+003	32.4570	0.4554	118.6910	no
4	4326210	2.3076e+003	33.3854	0.4996	117.7649	no
5	122265	2.0752e+003	34.4472	0.4970	117.8167	no
6	518400	2.7603e+003	31.5943	0.4011	119.9617	no
7	150975	2.5941e+003	32.2154	0.4595	118.6020	no
8	150975	2.0644e+003	34.4995	0.4940	117.8769	no
9	151353	2.6969e+003	31.8266	0.4782	118.2035	no
10	1890000	2.5770e+003	32.2815	0.5207	117.3516	no

The hiding and extraction times were calculated for LSB and the proposed methods, table 5 shows the obtained experimental results:

Table-5. Efficiency parameters.

Image number	LSB		Proposed	
	Hiding time (second)	Extracting time (second)	Hiding time (second)	Extracting time (second)
1	0.4440	0.1690	0.0420	0.0020
2	0.3610	0.0970	0.0460	0.0010
3	0.7760	0.5240	0.0520	0.0090
4	3.5720	0.7360	0.1190	0.0720
5	0.4260	0.1390	0.0610	0.0020
6	0.7680	0.5460	0.0520	0.0080
7	0.4720	0.1680	0.0440	0.0030
8	0.4430	0.1720	0.0610	0.0020
9	0.4510	0.1670	0.0440	0.0030
10	1.7290	0.6780	0.0800	0.0330
Average	0.9442	0.3396	0.0601	0.0135
Speed up	1	1	0.9442/0.0601=15.7105	0.3396/0.0135=25.1556

From the obtained experimental result shown in the previous table we can see the following:

- The proposed method is very efficient and has a considerable speed up comparing with LSB method.
- LSB method must check the capacity factor condition in order to resize the covering image or not, so we have to be care about the capacity, while in the proposed method the capacity condition is ignored.
- The quality parameters for LSB method are better, but they are good and acceptable for the proposed method.

5. CONCLUSIONS

A method of embedding message image into another covering image to produce a stego image was introduced, tested and implemented. The obtained experimental results showed that the proposed method is very efficient keeping good values for the quality parameters MSE and PSNR. The obtained proposed method results were compared with LSB method results and it was shown that the proposed method has a considerable speed up keeping good value for the quality parameters.

ACKNOWLEDGMENT

The researchers are grateful to the Applied Science Private University, Amman, Jordan, for the full financial support grated to this research project.

REFERENCES

[1] Dr. Mohamad Tariq Mohamad Barakat, Dr. Hatim Ghazi Zaini, Prof. Ziad AlQadi, Text File Encryption_Decryption Using Key Quitient and remainder. International Journal of Engineering Technology Research & Management. 5(4): 9-21.

[2] Ziad AlQadi, M. Elsayyed Hussein. 2017. Window Averaging Method to Create a Feature Victor for RGB Color Image. International Journal of Computer Science and Mobile Computing. 6(2).

[3] Bilal Zahran Belal Ayyoub, Jihad Nader, Ziad Al-Qadi. 2019. Suggested Method to Create Color Image Features Victor. Journal of Engineering and Applied Sciences. 14(1): 2203-2207.

[4] Majed O. Al-Dwairi, Ziad A. Alqadi, Amjad A. Abujazar, Rushdi Abu Zneit. 2010. Optimized true-color image processing. World Applied Sciences Journal. 10(8): 1175-1182.

[5] A. A. Moustafa, Z. A. Alqadi/ 2009. Color Image Reconstruction Using A New R'G'I Model. Journal of Computer Science. 5(4): 250-254.

[6] Jamil Al Azzeh, Hussein Alhatamleh, Ziad A. Alqadi, Mohammad Khalil Abuzalata. 2016. Ccreating a Color Map to be used to convert a Gray Image to Color Image. International Journal of Computer Applications. 153(2).

[7] Ashraf Abu-Ein, Ziad A. A. Alqadi, Jihad Nader. 2016. A Technique Of Hiding Secretes Text In Wave File. International Journal of Computer Applications.

[8] Rushdi Abu Zneit, Jamil Al-Azzeh, Ziad Alqadi, Belal Ayyoub, Ahmad Sharadqh. 2019. Using Color Image

as a Stego-Media to Hide Short Secret Messages. IJCSMC. 8(6): 106-123.

[9] Jamil Al-Azzeh, Ziad Alqadi, Belal Ayyoub, Ahmad Sharadqh. 2019. Improving the security of LSB image steganography. JOIV: International Journal on Informatics Visualization. 3(4): 384-387.

[10] Belal Ayyoub Ziad Alqadi, Ahmad Sharadqh, Naseem Asad Ismail Shayeb, Jamil Al-Azzeh. 2019. A highly secure method of secret message encoding. International Journal of Research in Advanced Engineering and Technology. 5(3): 82-87.

[11] Ahmad Sharadqh Ziad Alqadi, Bilal Zahran, Qazem Jaber, Belal Ayyoub, Jamil Al-Azzeh, Proposed Implementation Method to Improve LSB Efficiency. International Journal of Computer Science and Mobile Computing. 8(3): 306-319.

[12] Ziad Alqadi, Bilal Zahran, Qazem Jaber, Belal Ayyoub, Jamil Al-Azzeh. 2019. Enhancing the Capacity of LSB Method by Introducing LSB2Z Method. International Journal of Computer Science and Mobile Computing. 8(3): 76-90.

[13] Ziad Alqadi, Bilal Zahran, Qazem Jaber, Belal Ayyoub, Jamil Al-Azzeh, Ahmad Sharadqh. 2019. Proposed Implementation Method to Improve LSB Efficiency. International Journal of Computer Science and Mobile Computing. 8(3): 306-319.

[14] Dr. Mohammad S. Khrisat Prof. Ziad Alqadi, Simple, Qualities. 2021. Efficient and Secure Method to Encrypt Voice Signal. International Journal of Computer Applications. 183(7): 25-29.

[15] Ziad alqadi Hatim Ghazi Zaini. 2021. Replaced ASCII table to encode-decode secret messages. International Journal of Advanced Research in Computer and Communication Engineering. 10(3): 67-74.

[16] Hatem Zaini Prof. Ziad Alqadi. 2021. Color Image Cryptography Using Huge Random Private Key. Word journal of engineering research and technology. 7(3): 42-52.

[17] Dr. Mohammad S. Khrisat Prof. Ziad Alqadi. 2021. Analysis of Text Files Encryption-Decryption Methods. Ijetrm. 5(3): 48-54.

[18] Prof. Ziad A. Alqadi Anwar Al Abadi. 2021. Using Large Color Image to Encrypt-Decrypt Smaller Ones. Ijetrm. 5(2): 71-81.

[19] Prof. Ziad Alqadil. 2021. Efficient and Highly Secure Method of Message Encryption, IJETRM. 5(2): 58-64.

[20] Prof. Ziad A. Alqadi Anwar Abadi. 2021. Using color image to encrypt-decrypt wave file. IJARCCE. 9(12): 99-106.

[21] Musbah Aqel Ziad A. Alqadi. 2009. Performance analysis of parallel matrix multiplication algorithms used in image processing. World Applied Sciences Journal. 6(1): 45-52.