**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**
**IMPACT FACTOR: 6.199**

***IJCSMC, Vol. 8, Issue. 6, June 2019, pg.106 – 123***

# Using Color Image as a Stego-Media to Hide Short Secret Messages

**Rushdi Abu Zneit; Jamil Al-Azzeh; Ziad Alqadi; Belal Ayyoub; Ahmad Sharadqh**

Department of Computer Engineering, Al Balqa'a Applied University, Amman, Jordan

***Abstract: Steganography is the process of hiding secret messages in color covering images, steganography hides the existence of the message within the color image so that it cannot be noticed by human eyes. Steganography is an important process which can be used in various life applications, this paper will introduce some methods used for data steganography, these method will be implemented and experimental results will be obtained. A novel method of data steganography will be introduced, tested and implemented, it will be shown how this method will enhance efficiency, capacity, security and covering image quality.***

***Keywords: Steganography, stego-key, LSB, LSB2, PVD, PSNR, MSE, hiding time, extraction time***

## *1- Introduction*

### *1-1 Digital color image*

True color image (RGB color image) [1], [2], [3] is one of the most important digital data type used in data communication [4]. Digital color image can be reperesented by 3D matrix, the first dimension is reserved for the red color, the second for the green color, and the third for the blue color [5], [6]. Now most used color images have a high resolution, which leads to a huge size, thus color images can be used as suitable media to hold text messages, providing a high secure environment to hold and cover secret data [7].

Color image can be treated using the mixed 3 colors, or we can treate each color iindividualy using the 2D matrix reserved for the color [8] [9]. Each color value ranges from 0 to 255, and if the color image is equalized then these values will be coverd in the image histogram, figure 1 shows a color image and the colors histogram, while figure 2 shows the distibuted color values for each color channel:

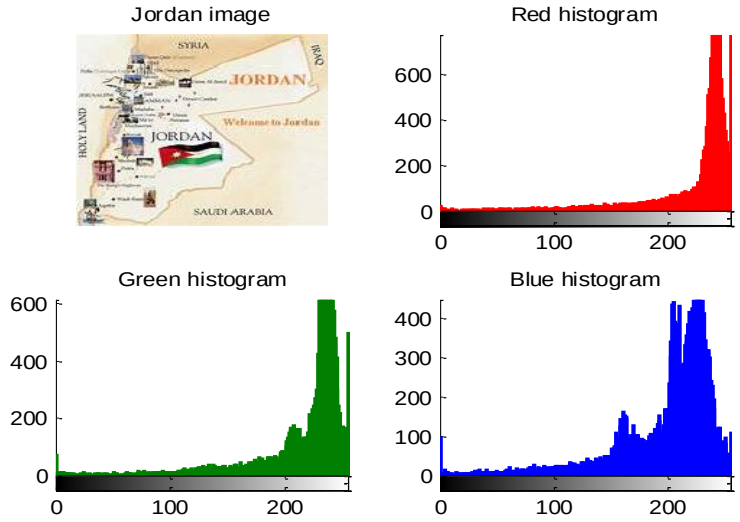



Figure 1: Color image and colors histogram

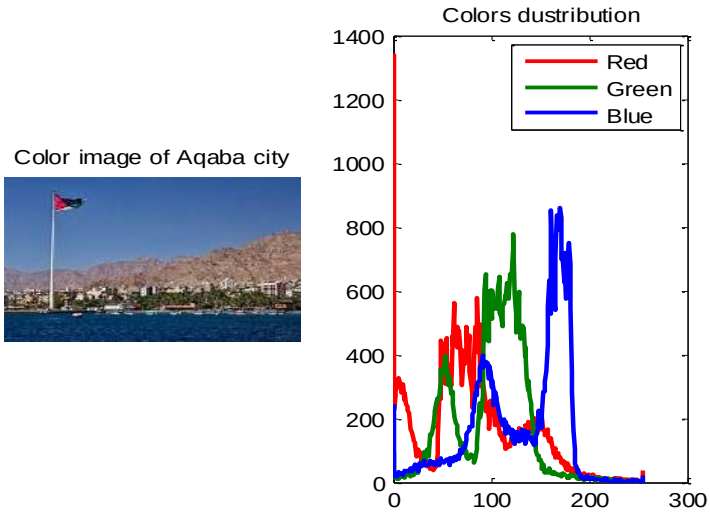



Figure 2: Colors distribution

In this paper research we will focus on pixels values to be used to hold the ASCII values of the secret message, for example, if want to hide the letter "A", this can be possible if at least one pixel exists in the image and the color value is equal to 65, this pixel can be used for any repetion of this letter in the message, so here we do not need to update the pixel value, but we have to remember the pixel position in the image(row, column, and color channel). Histogram equalization is here needed in order to be certain that all the color values in the image are covered.

### *1-2 Data steganography*

Data steganography is the process of ebbeding data in a seemingly innocouous covering media, such as digital color image, steganography hides the existences of the secrete message so that it can not be noticed by the human eyes[10], [11].

Steganography is an important process and it is now used in various vital applications such as[12], [13]:

- Protecting private data to be unseen by unautherised person.

- Protecting confidential data to be unread by third party.
- Generating a tag for any image to define the image originality or to identify image authority.

Hiding impotant data into a covering media requires the following elements[14], [15] as shown in figure 3:

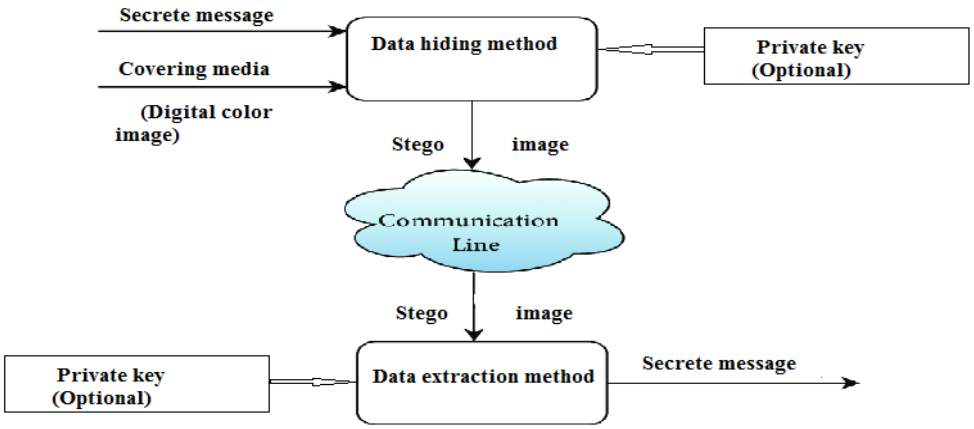



Figure 3: Elements of steganography

- The covering media, that will hold the secrete message.
- The secret message.
- The stego-function and its inverse.
- Stego-key to improve the security level of the stego-function.

Any method or technique used for data steganography can be characterized by the following factors:

a) Capacity: the maximum amount of data (in bytes) which can be hidden in the covering color image without destroying the image(the changes in the image must not be noticed by human eyes).[16], [17].

b) Image quality: Here the covering image quality must be closed to the original image quality [18], this factor can be measured by peak signal to noise ratio (PSNR) [19] or mean square error (MSE) [20] between the original and the covering images, the latger PSNR value is a good quality, and the smaller MSE is also a good quality, so a good method of data steganography must maximize PSNR value and minimize MSE value.

c) Effeciency: here we can use 2 parameters to measure the method efficiency: the hiding time and the extraction times, these time must be small enough [21].

d) Security: under security we mean the possibility of secret message hacking or attacking [22], [23], [24], [25] . There are several atacks possible when using any method of data steganography such as:

- Known carrier attack: here the original ,covering images and the stego-method are known and it is very easy to hack the message( no security).
- Steganography Only attack: In this type of attacks, only stego-image is available for analysis.
- Key attack: in this type of hacking only the key is available for analysis.
-

### *1-3 Related works*

Many authers used least significient bit (LSB) of data steganography as a based to deign a stego-method [10], [11]. Some authers added a key to LSB method to enhnace the method security [12], some of the authers modified the LSB method to LSB2 method to increase the hiding capacity [17], [24], some of them used the pixel value differencing (PVD) to increase the hiding capacity [25] , here we will provide a short review of these methods.

- ***LSB method***

The LSB method of data steganography reserves 8 pixels from the covering image to hold one character from the secret message, each binary bit from message character is to be inserted in the least sinificient bit of the associated pixel of the covering image as shown in figure 4.

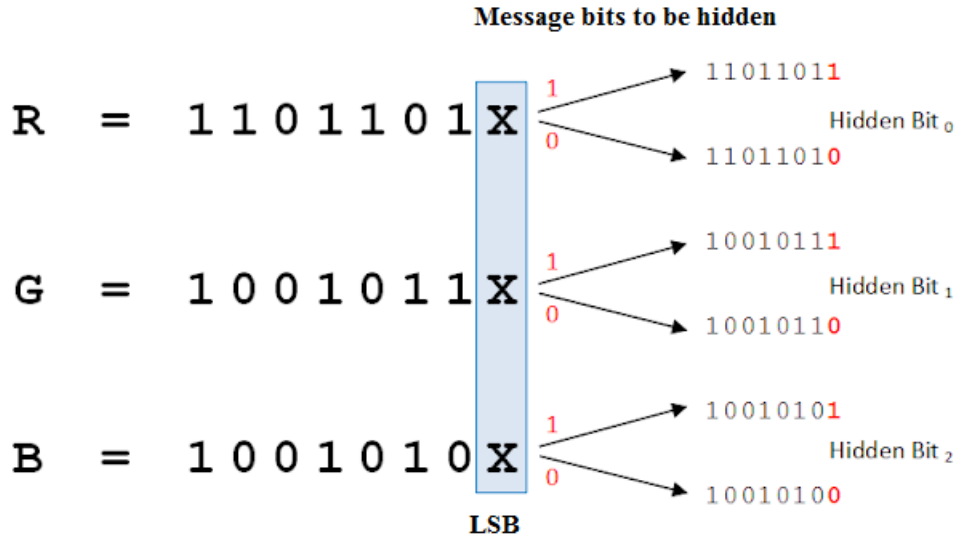



Figure 4: LSB operations

The capacity of LSB method is good enough and it is equal the covering image size divided by 8,

The quality of the covering image is high, LSB method provides a high value for PSNR and low value for MSE, because it addes minor changes to the covering image and the holding byte of the covering image may remaine the same or added by 1 or subtracted by one, which means that the resulting holding pixel will be closed to the origina; omage pixel.

LSB method is very simple to implement but it dosnt provide any kind of security because the covering image is known and available for analysis, also the stego-function is also known.

To add some kind of security to LSB method it possible to add a stego-key which will point to the starting position where to start secret message hiding.

- ***LSB2 method***

LSB2 method of data steganography likes LSB method, but it reserves 4 pixels from the covering image to hold one character from the secret message, by this LSB2 method increases the capasity twise and here the maximum capacity will be equal the holding image size divided by 4. Figure 5 shows how this method operates.

The quality of the covering image will be less or closed to quality in LSB, and LSB2 method may adds a minor changes in the holding image, these changes will be within the range -3 to +3.

LSB2 method also dosnt provide any kind of security because the covering image is known and available for analysis, also the stego-function is also known.

To add some kind of security to LSB method it possible to add a stego-key which will point to the starting position where to start secret message hiding.

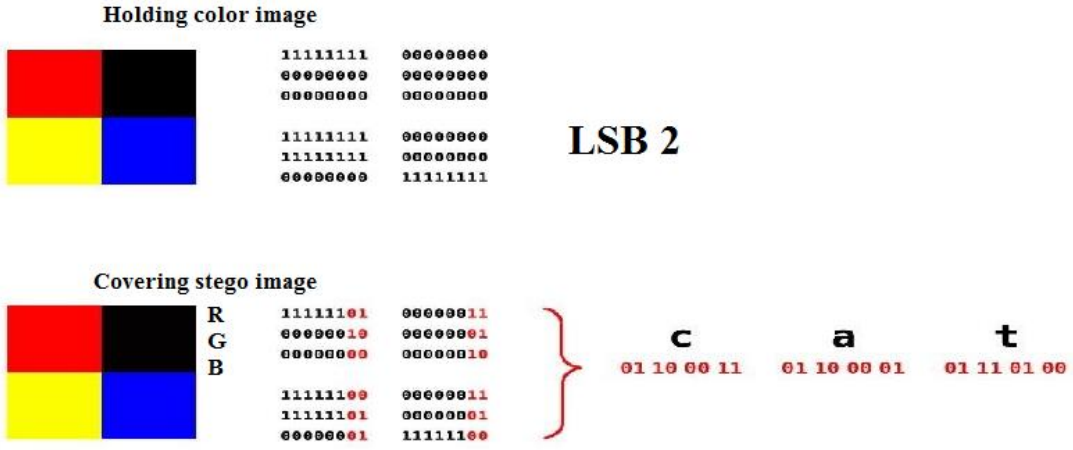



Figure 5: LSB2 operations

- ***PVD method***

Pixel value differencing (PVD) method uses 2 pixels from the covering image to hide a set of bits from the secret message to be hidden in image [25], [26]. This method of data hiding starts with an initialization phase, in which we divide the pixels range value (0:255) into non-overlapping partitions (these partitions are called lookup table, each partition has a ***lower*** and **upper** values, and each partition is associated with number of bits from the message to be hidden, this number is calculated by using formula (1):

$$t=\lfloor \log_2 (upper_i - lower_i + 1) \rfloor \qquad (1)$$

Two consecutive pixels in the $i$th partition are denoted as $P_i$ and $P_{i+1}$, respectively. The difference value, $d_i$, between two consecutive pixels is calculated by $d_i = |P_i - P_{i+1}|$. The absolute value of $d_i$ denotes the variation present in each partition. A small value of $d_i$ suggests the presence of a smooth region, whereas a larger value indicates the presence of the edge region. The $d_i$ value can be quantized into several regions as shown in examples 1 and 2.The obtained bit sequence is converted into decimal value, $t_d$. The new difference value ($d'i$) is obtained by $d'i = td + lower_i$.

The modified pixel values are computed based on the condition shown in formula 2:

$$\textbf{Holding 2 bytes} \quad (P_i', P_{i+1}')$$

$$= \begin{cases} \left(P_i + \left\lceil \frac{m}{2} \right\rceil, P_{i+1} - \left\lfloor \frac{m}{2} \right\rfloor\right), \text{ if } P_i \geq P_{i+1} \text{ and } d_i' > d_i & \textbf{Rule 1} \\ \left(P_i - \left\lfloor \frac{m}{2} \right\rfloor, P_{i+1} + \left\lceil \frac{m}{2} \right\rceil\right), \text{ if } P_i < P_{i+1} \text{ and } d_i' > d_i & \textbf{Rule 2} \\ \left(P_i - \left\lceil \frac{m}{2} \right\rceil, P_{i+1} + \left\lfloor \frac{m}{2} \right\rfloor\right), \text{ if } P_i \geq P_{i+1} \text{ and } d_i' \leq d_i & \textbf{Rule 3} \\ \left(P_i + \left\lceil \frac{m}{2} \right\rceil, P_{i+1} - \left\lfloor \frac{m}{2} \right\rfloor\right), \text{ if } P_i < P_{I+1} \text{ and } d_i' \leq d_i & \textbf{Rule 4} \end{cases} \qquad (2)$$

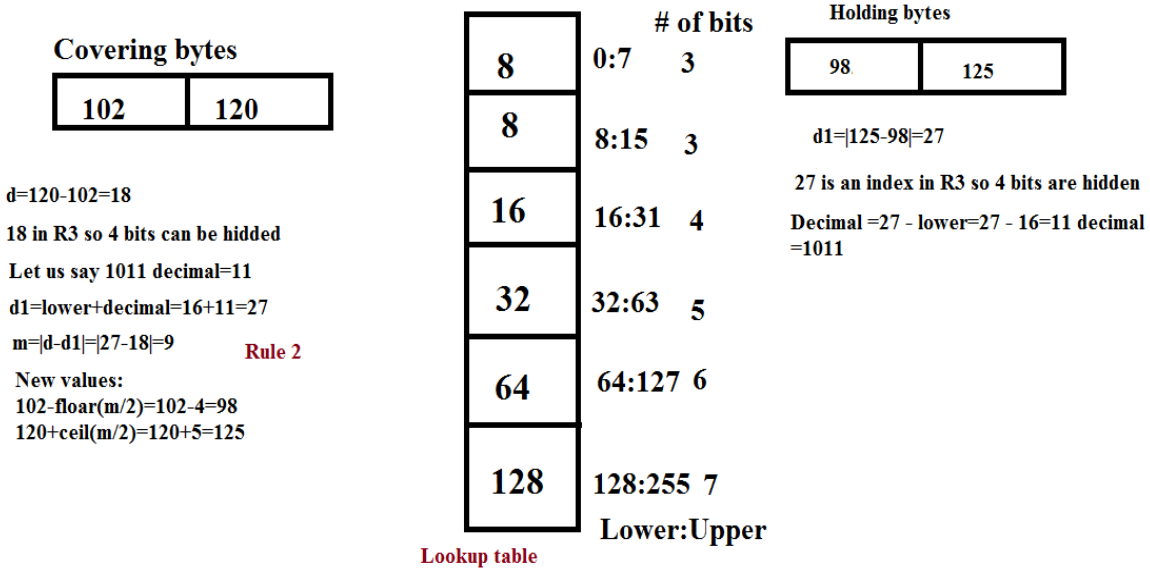
Where $m = |d'i - di|$.

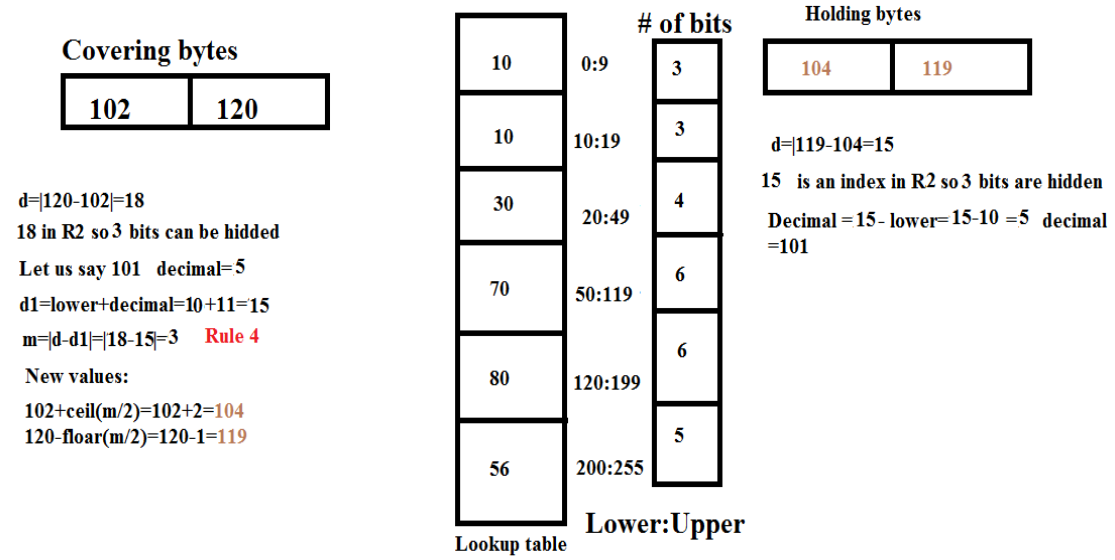
The capacity of this method is depend on the average bits in lookup table and it is always bigger than 2 bits from a message into one byte from the covering image.

Example 3 and 4 show how this method works:

***Example 1:***




***Example 2:***




PVD method provides a high PSNR values keeping the quality of the covering image high, in this method a lockup table is unkwown which leads to some kind of security, but the stego-image is known and it will be under analysis by the attackers.

### *2- The proposed method*

The proposed method of data hiding uses a secret message, covering image( if needed: optional) as an inputs for the data hiding process and it can be implemeted applying the following steps:

1. Get the holding covering image.
2. Get the secret message.
3. Retrive the message length(L).
4. Initialize the stego-key with L rows and 3 columns.
5. Get the ASCII code of the message.
6. For each character in the secret message do the following:
    a) Find the first appearance of the character value in the image.
    b) Get the row, column, and color chanel of the found pixel.
    c) Add one row to stego-key
    d) End do
7. Save the stego-key.

The extraction process can be implemented applying the following steps:

1. Get the holding image.
2. Load the stego-key.
3. For each row in the stego-key retrieve the pixel value using the information in this row (pixel row, pixel column and pixel color).
4. Concatenate each retrieved pixel value to the message.
5. Change the message from decimal to character.

This method has the following important features:

1. The capacity of the method is unlimited and it can exceed the holding image size.
2. The quality of the holding image is the same as for the original image (PSNR=infinite, MSE=0).
3. The surety level of the proposed method is high because we use here a stego-key and the holding image can be kept away, the sender and the receiver must agree on the image to be used as a holding image without transferring it.
4. The same holding image can be used for retrieving different short messages with different stego-keys.
5. The hiding stage can be considered as a stage of generating stego-key.
6. The efficiency parameters will be shown later in the experimental part.

### *3- Implementation and experimental results*

The following 3 short messages shown in figure 6 were used in our experiment:


Figure 6: Messages examples

The images shown in figure 7 were used in this experiment:

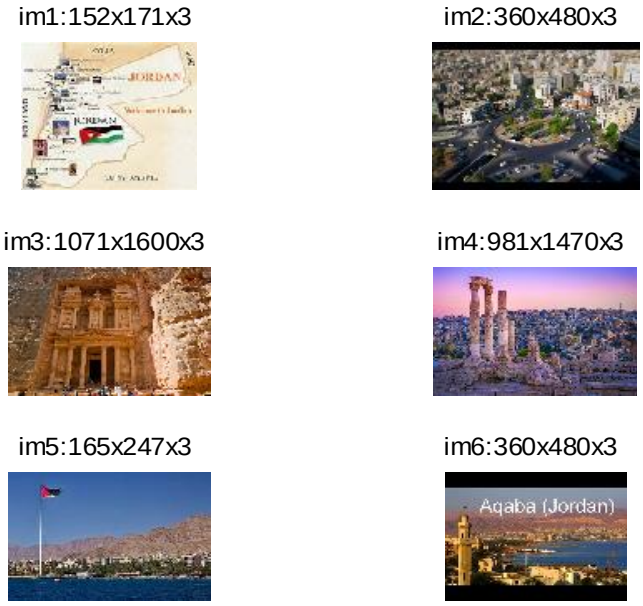



Figure 7: Used color images.

As we said, the original image and the covering images using the proposed method will be the same, figures 8 and 9 shows the original image and the image holding 3 messages:

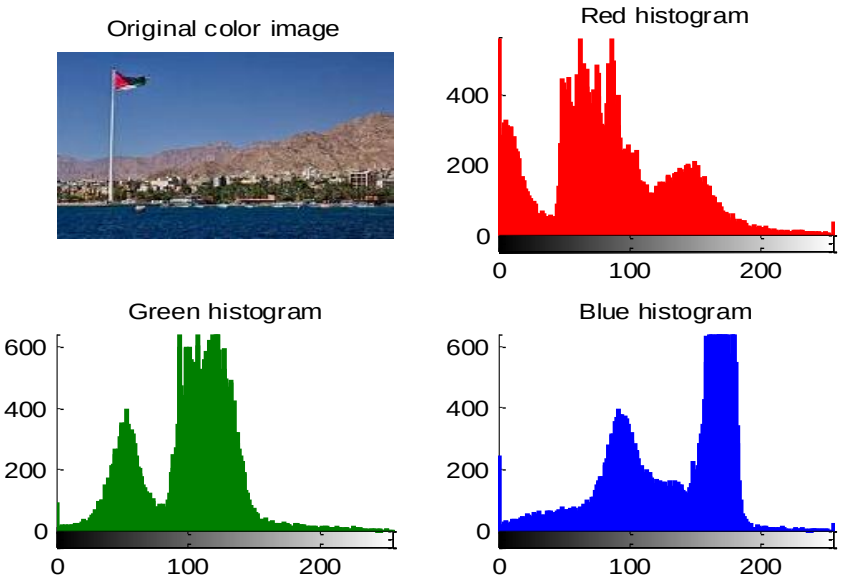



Figure 8: Original image

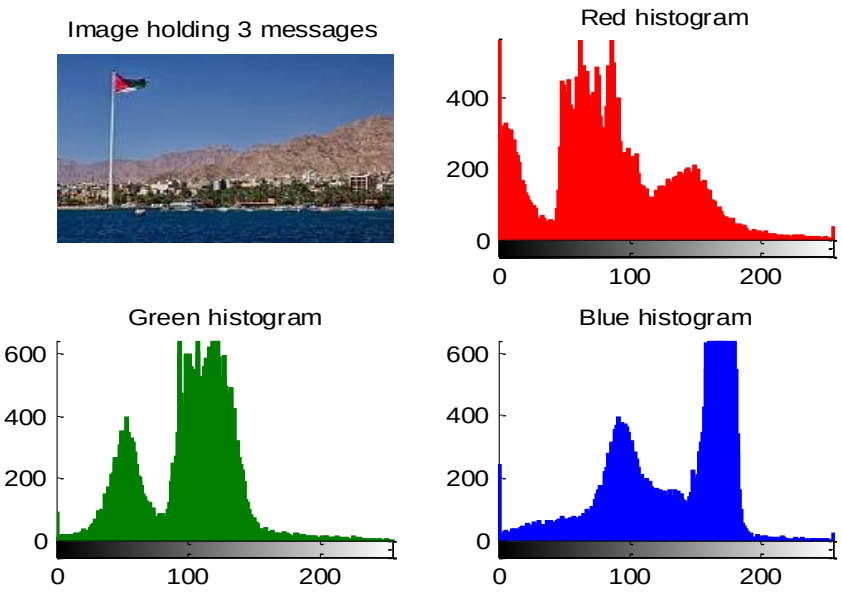



Figure 9: Holding image

Here the sender of the message and the receiver will agree on the stego-key, thus there is no need to transfer the image, only different stego-keys can be transferred, figures 10, 11 and 12 show the generated stego-key for each message:

| | | |
|---|---|---|
| 141 | 3 | 1 |
| 159 | 1 | 1 |
| 196 | 1 | 1 |
| 224 | 1 | 1 |
| 270 | 1 | 1 |
| 308 | 1 | 1 |
| 150 | 1 | 1 |
| 141 | 1 | 1 |
| 196 | 1 | 1 |
| 224 | 1 | 1 |
| 159 | 1 | 1 |

Figure 10: Stego-key for message 1

| | | |
|---|---|---|
| 300 | 1 | 1 |
| 196 | 1 | 1 |
| 154 | 1 | 1 |
| 48 | 1 | 1 |
| 150 | 1 | 1 |
| 152 | 1 | 1 |
| 55 | 1 | 1 |
| 270 | 1 | 1 |
| 153 | 1 | 1 |
| 155 | 1 | 1 |
| 270 | 1 | 1 |
| 160 | 1 | 1 |
| 163 | 1 | 1 |
| 154 | 2 | 1 |
| 159 | 1 | 1 |
| 163 | 1 | 1 |
| 160 | 1 | 1 |
| 160 | 1 | 1 |
| 137 | 1 | 1 |
| 159 | 1 | 1 |
| 163 | 1 | 1 |
| 154 | 2 | 1 |
| 270 | 1 | 1 |
| 152 | 1 | 1 |
| 160 | 1 | 1 |
| 154 | 1 | 1 |
| 196 | 2 | 1 |
| 163 | 1 | 1 |
| 153 | 1 | 1 |
| 150 | 1 | 1 |
| 153 | 1 | 1 |
| 154 | 2 | 1 |
| 55 | 1 | 1 |

Figure 11: Stego-key for message 2

```
300     1     1
196     1     1
154     1     1
 48     1     1
150     1     1
152     1     1
 55     1     1
270     1     1
153     1     1
155     1     1
270     1     1
160     1     1
163     1     1
154     2     1
159     1     1
163     1     1

160     1     1
160     1     1
137     1     1
159     1     1
163     1     1
154     2     1
270     1     1
152     1     1
160     1     1
154     1     1
196     2     1
163     1     1
153     1     1
150     1     1
153     1     1
154     2     1
 55     1     1
```

Figure 12: Stego-key for message 3

### *3-1 Parameters calculation for the proposed method*

The proposed method was programmed using matlab, the program was implemented using the selected 3 messages and the selected color images, table 1 shows the obtained parameters using message 1, table 2 shows the obtained parameters using message 2, while table 3 shows the obtained parameters using message 3, figures 13 shows the original and the holding images.

Table 1: Calculated parameters for message 1

| Image | Hiding time(stego-key generation) | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.002000 | 0.000001 | infinite | 0 |
| 2 | 0.008000 | 0.000001 | infinite | 0 |
| 3 | 0.093000 | 0.000001 | infinite | 0 |
| 4 | 0.078000 | 0.000001 | infinite | 0 |
| 5 | 0.003000 | 0.000001 | infinite | 0 |
| 6 | 0.009000 | 0.000001 | infinite | 0 |
| **Average time** | **0.0322** | **0.000001** | | |

Table 2: Calculated parameters for message 2

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.003000 | 0.000001 | infinite | 0 |
| 2 | 0.019000 | 0.000001 | infinite | 0 |
| 3 | 0.237000 | 0.000001 | infinite | 0 |
| 4 | 0.205000 | 0.000001 | infinite | 0 |
| 5 | 0.006000 | 0.000001 | infinite | 0 |
| 6 | 0.019000 | 0.000001 | infinite | 0 |
| **Average time** | **0.0815** | **0.000001** | | |

Table 3: Calculated parameters for message 3

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.004000 | 0.000001 | infinite | 0 |
| 2 | 0.020000 | 0.000001 | infinite | 0 |
| 3 | 0.231000 | 0.000001 | infinite | 0 |
| 4 | 0.189000 | 0.000001 | infinite | 0 |
| 5 | 0.006000 | 0.000001 | infinite | 0 |
| 6 | 0.020000 | 0.000001 | infinite | 0 |
| **Average time** | **0.0783** | **0.000001** | | |

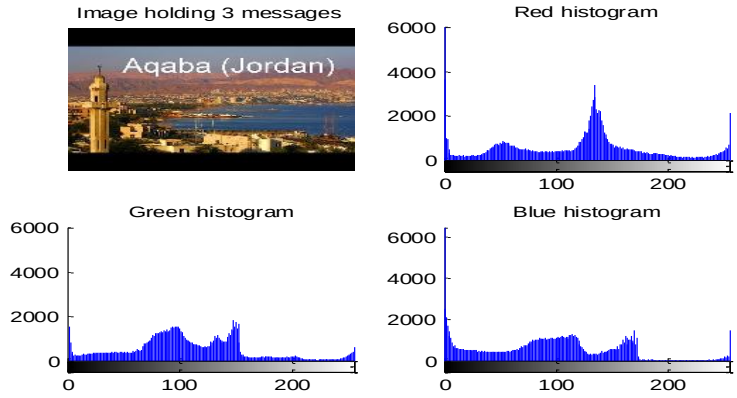



Figure 13: Original image (holding is the same) (proposed method)

### *3-2 Parameters calculation for the LSB method*

LSB method was programmed using matlab, the program was implemented using the selected 3 messages and the selected color images, table 4 shows the obtained parameters using message 1, table 5 shows the obtained parameters using message 2, while table 6 shows the obtained parameters using message 3, figures 14 and 15 show the original and the holding images

Table 4: Calculated parameters for message 1(LSB method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.044000 | 0.035000 | 169.2615 | 0.0029 |
| 2 | 0.052000 | 0.036000 | 199.9253 | 0.00013503 |
| 3 | 0.209000 | 0.071000 | 210.8421 | 0.000045324 |
| 4 | 0.189000 | 0.067000 | 209.0314 | 0.000054320 |
| 5 | 0.044000 | 0.034000 | 179.9245 | 0.0009783 |
| 6 | 0.365000 | 0.359000 | 199.9253 | 0.00013503 |
| **Average time** | **0.1505** | **0.1003** | | |

Table 5: Calculated parameters for message 2(LSB method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.044000 | 0.033000 | 155.6676 | 0.0113 |
| 2 | 0.053000 | 0.036000 | 186.3523 | 0.00052469 |
| 3 | 0.209000 | 0.070000 | 197.0006 | 0.00018091 |
| 4 | 0.188000 | 0.069000 | 197.0215 | 0.00018053 |
| 5 | 0.046000 | 0.034000 | 162.8621 | 0.0055 |
| 6 | 0.053000 | 0.037000 | 186.3523 | 0.00052469 |
| **Average time** | **0.0988** | **0.0465** | | |

Table 6: Calculated parameters for message 3(LSB method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.044000 | 0.034000 | 155.6676 | 0.0113 |
| 2 | 0.053000 | 0.039000 | 186.3523 | 0.00052469 |
| 3 | 0.210000 | 0.066000 | 197.0006 | 0.00018091 |
| 4 | 0.188000 | 0.070000 | 197.0215 | 0.00018053 |
| 5 | 0.045000 | 0.034000 | 162.8621 | 0.0055 |
| 6 | 0.055000 | 0.036000 | 186.3523 | 0.00052469 |
| **Average time** | **0.0992** | **0.0465** | | |

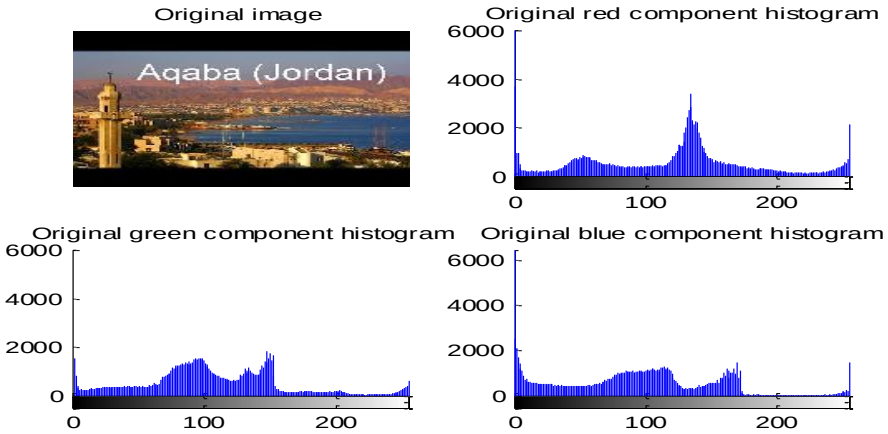



Figure 14: Original image (LSB method)

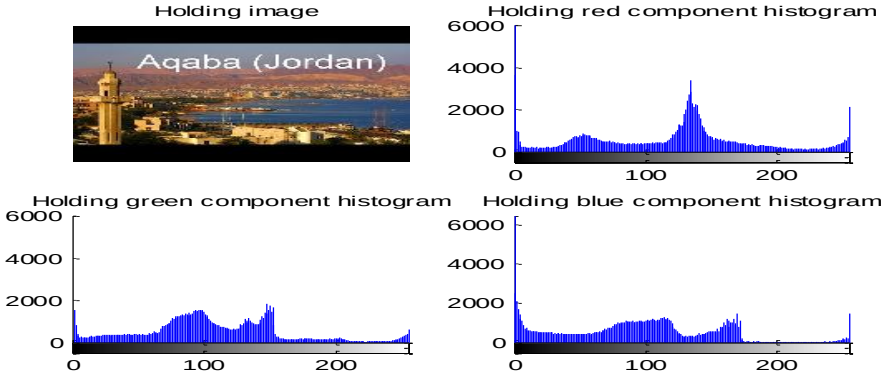



Figure 15: Holding image (LSB method)

### *3-3 Parameters calculation for the LSB2 method*

LSB2 method was programmed using matlab, the program was implemented using the selected 3 messages and the selected color images, table 7 shows the obtained parameters using message 1, table 8 shows the obtained parameters using message 2, while table 9 shows the obtained parameters using message 3, figures 16 and 17 show the original and the holding images

Table 7: Calculated parameters for message 1(LSB2 method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.000020 | 0.000020 | 180.7002 | 0.00092336 |
| 2 | 0.000020 | 0.000020 | 196.6632 | 0.00018711 |
| 3 | 0.002000 | 0.000020 | 217.9905 | 0.000022176 |
| 4 | 0.002000 | 0.001000 | 217.3776 | 0.000023577 |
| 5 | 0.000020 | 0.000020 | 183.7763 | 0.00067885 |
| 6 | 0.000020 | 0.000020 | 198.8432 | 0.00015046 |
| **Average time** | **0.00068** | **0.00018333** | | |

Table 8: Calculated parameters for message 2(LSB2 method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.000020 | 0.000020 | 166.4624 | 0.0038 |
| 2 | 0.001000 | 0.000020 | 185.3060 | 0.00058256 |
| 3 | 0.002000 | 0.000020 | 209.5552 | 0.000051548 |
| 4 | 0.002000 | 0.000020 | 206.9285 | 0.000067033 |
| 5 | 0.001000 | 0.000020 | 171.9805 | 0.0022 |
| 6 | 0.000020 | 0.000020 | 186.2426 | 0.00053048 |
| **Average time** | **0.0010** | **0.00002** | | |

Table 9: Calculated parameters for message 3(LSB2 method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 0.000020 | 0.000020 | 164.7175 | 0.0046 |
| 2 | 0.001000 | 0.000020 | 184.1208 | 0.00065586 |
| 3 | 0.003000 | 0.000020 | 206.9752 | 0.00066721 |
| 4 | 0.002000 | 0.000020 | 205.3378 | 0.00078591 |
| 5 | 0.000020 | 0.000020 | 170.4073 | 0.0026 |
| 6 | 0.001000 | 0.000020 | 184.3589 | 0.00064043 |
| **Average time** | **0.0012** | **0.000020** | | |

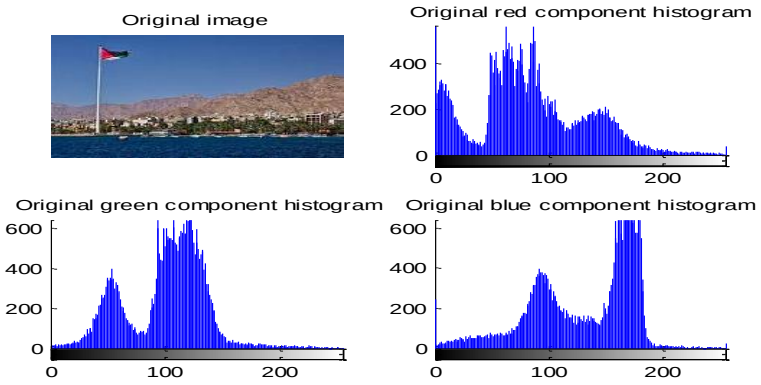



Figure 16: Original image (LSB2 method)

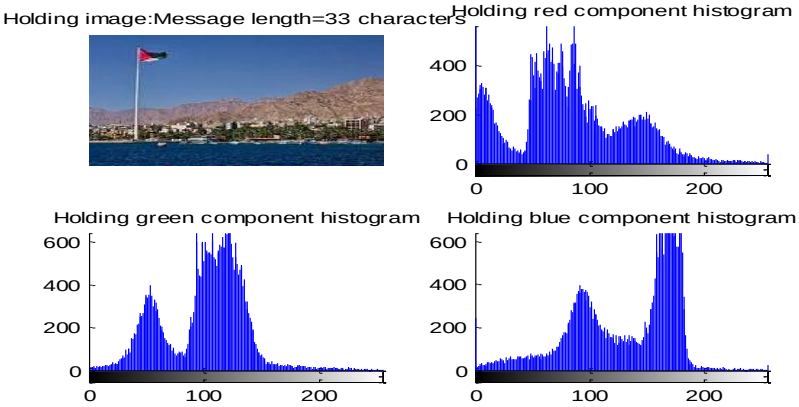



Figure 17: Holding image (LSB method)

### *3-4 Parameters calculation for the PVD method*

PVD method was programmed using matlab, the program was implemented using the selected 3 messages and the selected color images, table 10 shows the obtained parameters using message 1, table 11 shows the obtained parameters using message 2, while table 12 shows the obtained parameters using message 3, figures 18 and 19 show the original and the holding images

Table 10: Calculated parameters for message 1(PVD method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 1.754300 | 3.369000 | 162.8621 | 0.0055 |
| 2 | 2.240000 | 5.635000 | 166.1397 | 0.0040 |
| 3 | 20.118000 | 0.640000 | 189.8196 | 0.00037095 |
| 4 | 16.644000 | 0.591000 | 183.6810 | 0.00068536 |
| 5 | 1.8702000 | 3.488000 | 164.7175 | 0.0046 |
| 6 | 2.207000 | 4.399000 | 162.3700 | 0.0058 |
| **Average time** | **7.4722** | **3.0203** | | |

Table 11: Calculated parameters for message 2(PVD method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 1.764300 | 3.769000 | 166.4624 | 0.0038 |
| 2 | 2.250000 | 5.552000 | 160.5540 | 0.0069 |
| 3 | 20.096000 | 0.735000 | 181.3010 | 0.00086951 |
| 4 | 16.654000 | 0.684000 | 176.2367 | 0.0014 |
| 5 | 1.8402000 | 3.988000 | 155.6676 | 0.0113 |
| 6 | 2.120000 | 4.430000 | 154.7448 | 0.0124 |
| **Average time** | **7.4541** | **3.1930** | | |

Table 11: Calculated parameters for message 3(PVD method)

| Image | Hiding time(stego-key generation | Extraction time(seconds) | PSNR | MSE |
|---|---|---|---|---|
| 1 | 1.764300 | 3.769000 | 166.4624 | 0.0038 |
| 2 | 2.223000 | 5.565000 | 153.1944 | 0.0145 |
| 3 | 20.186000 | 1.042000 | 175.2287 | 0.0016 |
| 4 | 16.605000 | 0.972000 | 170.8782 | 0.0025 |
| 5 | 1.8402000 | 3.988000 | 155.6676 | 0.0113 |
| 6 | 2.114000 | 4.399000 | 149.3048 | 0.0213 |
| **Average time** | **7.4554** | **3.2892** | | |

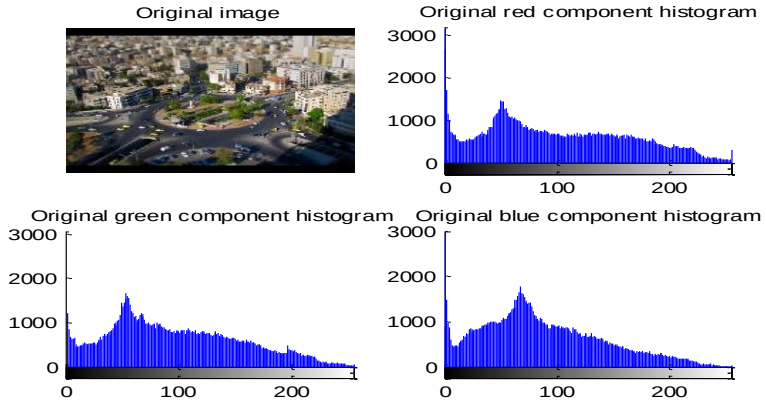



Figure 18: Original image (PVD method)

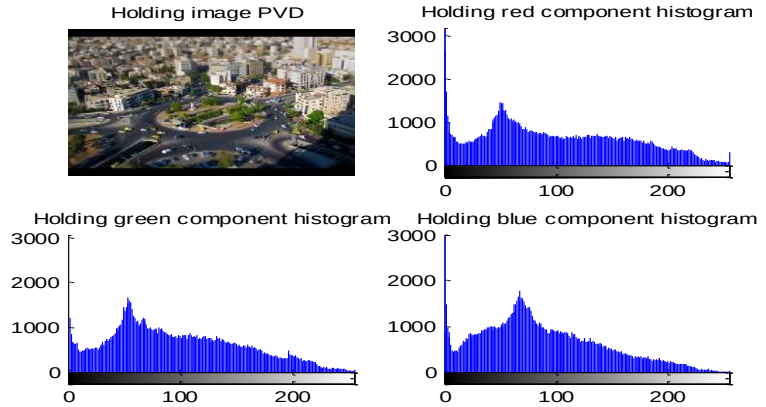



Figure 19: Holding image (PVD method)

From the obtained results we can raise the following facts:

- The proposed method does not affect the original covering image, and the quality of the holding image is the same as that for the original image, this method always gives an infinite value of PSNR and MSE with zero value.
- The efficiency parameters for this method are closed to the parameters obtained by other methods, and some time they are better.
- The proposed method is secure because it uses a stego-key, and there is no need to send the covering image, here the sender and receiver must agree on the image to be used for data hiding.
- The covering image may be used to extract any message using the stego-key generated for this message.
- The capacity of the proposed method is unlimited; here we can extract a message with size bigger than the holding image size.

***Conclusion***

Different methods of data steganography were implemented. A new method of data steganography was proposed, tested and implemented. It was shown the proposed method has some achievements over the existing studied method, the proposed method enhances the stego-process efficiency, capacity, security and image quality.

# References

[1] Ziad A. Alqadi, Majed O. Al-Dwairi, Amjad A. Abu Jazar and Rushdi Abu Zneit, Optimized True-RGB color Image Processing, World Applied Sciences Journal 8 (10): 1175-1182, ISSN 1818-4952, 2010.
[2] A. A. Moustafa, Z. A. Alqadi, Color Image Reconstruction Using A New R'G'I Model, journal of Computer Science, Vol.5, No. 4, pp. 250-254, 2009.
[3] Jamil Al Azzeh, Hussein Alhatamleh, Ziad A. Alqadi, Mohammad Khalil Abuzalata, Creating a Color Map to be used to Convert a Gray Image to Color Image; International Journal of Computer Applications , November 2016,Volume 153,Issue 2.
[4] Jamil Al-Azzeh, Ziad Alqadi, Mohammed Abuzalata, Performance Analysis of Artificial Neural Networks used for Color Image Recognition and Retrieving, International Journal of Computer Science and Mobile Computing,2019,Volume 8 Issue 2.
[5] Jamil AL-Azzeh, Bilal Zahran, Ziad Alqadi, Belal Ayyoub and Mazen Abu-Zaher: A Novel Zero-Error Method to Create a Secret Tag for an Image; Journal of Theoretical and Applied Information Technology 15th July 2018.
[6] Jamil AL-Azzeh, Bilal Zahran and Ziad Alqadi: Salt and Pepper Noise: Effects and Removal, International Journal on Informatics Visualization July 2018, Volume 2 Issue 4.
[7]Musbah J. Aqel , Ziad A. Alqadi, Ibraheim M. El Emary ,Analysis of Stream Cipher Security Algorithm, Journal of Information and Computing Science Vol. 2, No. 4, 2007, pp. 288-298.
[8] Belal Ayyoub, Ashraf Abu-Ein, Ziad Alqadi, Suggested Method to Create Color Image Features Victor, Journal of Engineering and Applied Sciences, 2019, Volume14, Issue7.
[9] K Matrouk, A Al-Hasanat, H Alasha'ary, Z. Al-Qadi,H Al-Shalabi, Speech fingerprint to identify isolated word person, World Applied Sciences Journal, Vol. 31, No. 10, pp. 1767-1771, 2014.
[10] Mohammed Abuzalata, Ziad Alqadi; Jamil Al-Azzeh; Qazem Jaber, Modified Inverse LSB Method for Highly Secure Message Hiding, *IJCSMC, Vol. 8, Issue. 2, February 2019, pg.93 – 103*
[11] Mutaz Rasmi Abu Sara Rashad J. Rasras, Ziad A. AlQadi, Engineering, A Methodology Based on Steganography and Cryptography to Protect Highly Secure Messages Technology & Applied Science Research, Vol.9 Issue 1, Pages 3681-3684, 2019.
[12] Ziad Alqadi, Bilal Zahran, Qazem Jaber, Belal Ayyoub, Jamil Al-Azzeh, Ahmad Sharadqh, proposed Implementation Method to Improve LSB Efficiency, International Journal of Computer Science and Mobile Computing, Vol.8 Issue.3, March-2019, pg. 306-319.
[13] Deepak Garg, Gourav Sharma, Applications of Steganography in Information Hiding, international Journal of Advanced Research in Education & Technology (IJARET) 12 Vol. 3, Issue 1 (Jan. - Mar. 2016).
[14]J. Al-Azzeh, B. Zahran, Z. Alqadi, B. Ayyoub, M. Abu-Zaher, "A Novel zero-error method to create a secret tag for an image", Journal of Theoretical and Applied Information Technology, Vol . 96. No. 13, pp. 4081-4091, 2018.
[15] Prof. Ziad A.A. Alqadi, Prof. Mohammed K. Abu Zalata, Ghazi M. Qaryouti, Comparative Analysis of Color Image Steganography, JCSMC, Vol.5, Issue. 11, November 2016, pg.37–43.
[16] M. Jose, "Hiding Image in Image Using LSB Insertion Method with Improved Security and Quality", International Journal of Science and Research, Vol. 3, No. 9, pp. 2281-2284, 2014.
[17] Ziad Alqadi; Bilal Zahran; Qazem Jaber; Belal Ayyoub; Jamil Al-Azzeh, Enhancing the Capacity of LSB Method by Introducing LSB2Z Method; International Journal of Computer Science and Mobile Computing,2019,Volume 8 Issue 3.
[18] Ahmad Sharadqh, Belal Ayyoub, Ziad Alqadi, Jamil Al-azzeh; Experimental investigation of method used to remove salt and pepper noise from digital color image, International Journal of Research in Advanced Engineering and Technology,2019.Volume 5 Issue 1.
[19] Jihad Nader, Ziad Alqadi, Bilal Zahran, Analysis of Color Image Filtering Methods, International Journal of Computer Applications (IJCA), Volume 174, issue 8, 2017, pp:12-17.
[20] Jamil AL-Azzeh, Bilal Zahran and Ziad Alqadi: Salt and Pepper Noise: Effects and Removal, International Journal on Informatics Visualization July 2018, Volume 2 Issue 4.
[21] Prof. Ziad A.A. Alqadi, Prof. Mohammed K. Abu Zalata, Ghazi M. Qaryouti, Comparative Analysis of Color Image Steganography, JCSMC, Vol.5, Issue. 11, November 2016, pg.37–43.
[22] M. Jose, Hiding Image in Image Using LSB Insertion Method with Improved Security and Quality, International Journal of Science and Research, Vol. 3, No. 9, pp. 2281-2284, 2014.
[23] R. M. Patel, D. J. Shah, Conceal gram :Digital image in image using LSB insertion method, International Journal of Electronics and Communication Engineering & Technology, Vol. 4, No.1, pp. 230-2035, 2013.

[24] N. Akhtar, P. Johri, S. Khan, Enhancing the security and quality of LSB based image steganography, 5th International Conference on Computational Intelligence and Communication Networks, Mathura, India.
[25] Wu D-C, Tsai W-H. 2003A steganographic method for images by pixel value differencing. Pattern Recognition. Lett. 24, 1613–1626.
[26] Zhou X, Gong W, Fu W, Jin L. 2016An improved method for LSB based color image steganography combined with cryptography. In *2016 IEEE/ACIS 15th Int. Conf. on Computer and Information Science (ICIS), Okayama, Japan*, pp. 1–4 .
[27] Belal Ayyoub, Ahmad Sharadqh, Ziad Alqadi, Jamil Al-azzeh, Simulink based RNN models to solve LPM, International Journal of Research in Advanced Engineering and Technology. Volume 5; Issue 1; January 2019; Page No. 49-55.
[28] Jamil Al-Azzeh , Rashad Rasras , Ziad Alqadi , Belal Ayyoub4 , Ahmad Sharadqh Adaptation of matlab K-means clustering function to create Color Image Features, International Journal of Research in Advanced Engineering and Technology, Volume 5; Issue 2; April 2019; Page No. 10-18.
[29] Jamil Al Azzeh, Alqadi Ziad, Jabber Qazem, Statistical Analysis of Methods Used to Enhanced Color Image Histogram, XX International Scientific and Technical Conference 2016.
[30] Rushdi Abu Zneit, Amjad Abu Jazar, Belal Ayyoub, Automatic Color Images Classification Algorithm, IJCSI International Journal of
Computer Sciences Issues, (2012), Vol. 9, Issue 2, No. 1, March 2012, 305-310, ISSN (online): 1694-0814.
[31] Abdelwadood Mesleh, Jamil Al-Azzeh, Rushdi Abu Zneit, Ashraf Abu-Ein, Detection of Eyes Using FCM, International review on computers and software,(2012),vol. 7 n.4,Papers Part A.
[32] Rushdi Saleem Abu Zneit, On-line Handwriting Signature Verification based on Using Extreme Points Extraction, ETASR, Vol 6, No. 4, 2016, 1084-1088, ISSN(e/p): 1792-8036, 2241-4487.{ ETASR indexed in ESCI (Thomson Reuters/Web of Science) was added in late 2015 to the Thomson Reuters Master Journal List through the Emerging Sources Citation Index.
[33] Dr. Rushdi S. Abu Zneit, Dr. Ziad AlQadi, Dr. Mohammad Abu Zalata, A Methodology to Create a Fingerprint for RGB Color Image, IJCSMC, vol. 6, Issue 1, January 2017, pg. 205-2012.
[34] Dr. Rushdi S. Abu Zneit, Dr. Ziad AlQadi, Dr. Mohammad Abu Zalata, Procedural Analysis of RGB Color Image Objects, IJCSMC, vol. 6, Issue 1, January 2017, pg. 197-204.
[35] Dr. Ziad AlQadi, Dr. Rushdi S. Abu Zneit, Dr. Mohammad Abu Zalata, Extracting Individual Objects from RGB color Image IJCSMC, vol. 6, Issue 1, January 2017, pg. 190-196