Image Steganography using LSB and LSB+Huffman Code

Wa'el Ibrahim A. Al-Mazaydeh
Al-Balqa' Applied University
Aqaba University College
Department of Applied Science
Aqaba-Jordan

ABSTRACT
Steganography is an important area of research in recent years involving a various number of applications. It is the science of embedding information into cover of the media such as text, image, audio, and video. This paper uses two techniques for Steganography (text into image): Least Significant Bit (LSB) and Least Significant Bit with Huffman code (LSB+HUFF). It uses the zigzag scanning for the two methods to increase the security, and compares the results using Peak Signal to Noise Ratio (PSNR). All images used here is a gray scale images to implement the study; what implemented on gray scale image can be applied on colored image.

General Terms
Cryptography, Steganography, Security, and Compression.

Keywords
Least Significant Bit (LSB), ASCII code, Peak Signal to Noise Ratio (PSNR), zigzag scanning

1. INTRODUCTION
Security is a process, not an absolute or a goal; you can have a lot of technique to protect yourself but you can't to reach the state that you are safe completely (100%). The main aim of the security is to find the most technique to protect the assets (The organizational resource that is being protected. An asset can be logical, such as a Web site, information, or data; or an asset can be physical, such as a person, computer system, or other tangible object [5]) from the attacks or hackers and allow the authorized user to use your recourses and protect them from the illegitimate users.

Digital communication has become an essential part of transfer data from place to another; these days a lot of applications are depend on the internet to transferring data to be secret. There are two technique are available to achieve this goal: the first method is cryptography, and the second method is Steganography.

Image Steganography is the art of information hidden into cover image. It is the process of hiding secret message within another message. The word Steganography in Greek means "Covered Writing". The information hiding process in a Steganography with different techniques includes identifying cover mediums redundant bits. The embedding process creates a stego medium by replacing the Least Significant Bits with data from the secret message [1].

The different between Steganography and cryptography: the cryptography is used to change the text into unreadable text that an intruder cannot understand the text; while the Steganography technique is used to hide the text into cover media (such as image or video), which it difficult for an observer to see it.

There are three essential elements related to the art of Steganography: capacity, security, and robustness. Capacity is a size of the information that can be hidden in the cover medium (secret message), security is a process that used to prevent an intruder to detect any changes on the stego media, and robustness is the amount of modification the stego media can resists before an attacker destroys the secret message [2].

This study is implemented on MATLAB program because it deals with images in an easy, good and accurate, addition it has many of functions that save time and effort for the programmer to deal images and matrices.

2. RELATED WORK
This study depends on the Least Significant Bit technique because it is the most technique used in the Steganography. It has a many of method: Huffman code to compression the secret message before sending it to the receiver, zigzag scanning to select the pixels that will secret message is hidden within, and PSNR to compare the stego image with the original image and see the amount of change between the two images.

2.1 Era of Steganography
The Egyptians were the first of the art encryption used before 4000 years ago. Pharaohs had using the Hieroglyphic writing on buildings and temples to describe their life at tat time. Hieroglyphic writing uses characters in the form of pictures. While hieroglyphics are not thought of as a form of secret writing in the modern world, some hieroglyphics were stylized in such a way that only those who knew what to look for could read them properly. This meager distinction can be considered one of the first instances of Steganography [9].

The Greek historian Herodotus about 440 BC records two events in steganography.The first event that King Darius of Susa shaved the head of one of his prisoners and wrote a secret message on his scalp. When the prisoner's hair grew back, he was sent to the Kings son in law Aristogoras in Miletus undetected. The second event which allege that a soldier named Demeratus needed to send a message to Sparta that Xerxes prepared to invade Greece. The secret message was text written on wax-covered tablets. Demeratus removed the wax from the tablet, wrote the secret message on the underlying wood, recovered the tablet with wax to make it appear as a blank tablet and then sent the Secret message without being detected [9].

Romans used invisible inks, which were based on natural substances such as fruit juices and milk. The secret message was read by heating the hidden text, thus revealing its contents [9].

2.2 ASCII Code

American Standard Code for Information Interchange ASCII is the most common format for characters in the computer systems. In an ASCII code, each alphabetic, numeric, or special character is represented with a 7 bits binary number (a string of seven 0s or 1s). Table 1 bellow shows a few characters and its representations in decimal and binary ASCII code.

Table 1: ASCII Codes for some characters

Character	ASCII Code (decimal)	ASCII Code (binary)
a	97	1100001
B	98	1100010
A	65	1000001
B	66	1000010
+	43	101011
(	40	101000
8	56	111000
9	57	111001

2.3 Least Significant Bit (LSB)

LSB is common technique in encrypting and decrypting the secret information. LSB method is based on substituting the redundant bits that are least important with the bits of the secret message. If we have 8 bytes of data and you want to hide the number "213" which is represented in ASCII code as (11010101). Figure 1 shows LSB process.

We Will Hide 213 which are represented as (11010101) in ASCII code by using one bit substitute:

Byte 1	Byte 2	Byte 3	Byte 4
1000010**0** 1	1000011**0** 1	1000100**1** 0	1000110**1** 1
1000010**1**	1000011**1**	1000100**0**	1000110**1**

Byte 5	Byte 6	Byte 7	Byte 8
0111100**1** 0	0110010**1** 1	0100101**0** 0	0010011**0** 1
0111100**0**	0110010**1**	0100101**0**	0010011**1**

Fig 1: Least Significant Bit (LSB).

2.4 Huffman Code

It was developed by David A. Huffman while he was a Ph.D. student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes". Huffman code is an algorithm to compression based on the frequency of occurrence of a symbol in the file that is being compressed. For example to compression the message (ABEACADABEA) we need to construct a Huffman tree which is the button-up approach according of the following steps:

- Count the frequency of each character in the message as a list as shown in the table 2.
- Sort the list by frequency and make the two lowest elements into leaves, creating a parent node with a frequency that is the sum of the two lower element's frequencies [10].
- The two elements are removed from the list and the new parent node is inserted into the list by frequency. So now the list, sorted by frequency [10]
- You then repeat the loop, combining the two lowest elements.
- You repeat until there is only one element left in the list.

Table 2: the frequencies and probabilities of the text (ABEACADABEA)

Symbol	frequency	Probability
A	5	5 / 11 = 0.45
B	2	2 / 11 = 0.18
C	1	1 / 11 = 0.09
D	1	1 / 11 = 0.09
E	2	2 / 11 = 0.18

To generate a Huffman code you traverse the tree to the value you want, outputting a 0 every time you take a left hand branch and a 1 every time you take a right hand branch [10]. Figure 2 illustrate the Huffman tree for the text (ABEACADABEA).

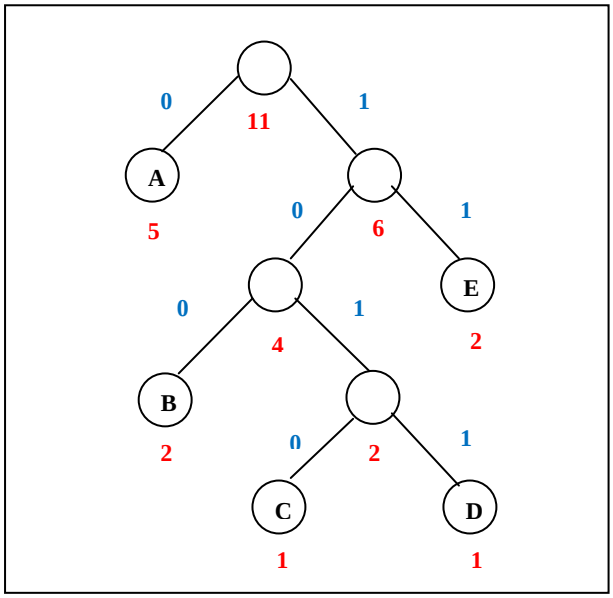


Fig 2: Huffman tree to (ABEACADABEA)

According the above Huffman tree we obtain the following code word in table 3.

Table 3: The code word of the text (ABEACADABEA) using Huffman tree

Symbol	Code word
A	0
B	100
C	1010

D	1011
E	11

After completion Huffman tree to the text (ABEACADABEA) we obtain (23 bit) code word:

01001101010010110100110

While the message in ASCII code is represented as 77 bits (11 characters × 7 bits), so Huffman code saves more than 25% in the size of the message.

2.5 Zigzag Scanning

This study uses zigzag scanning to increase the security in the process of hiding the secret message into the image. Zigzag scanning selects the pixels that will be hidden secret message inside; figure 3 shows the zigzag scanning process.

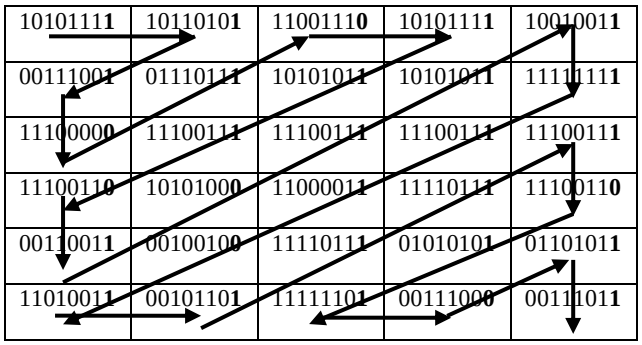


Fig 3: Zigzag Scanning

2.6 PSNR

Peak Signal to Noise Ratio (PSNR) (equation 1) is measured on a logarithmic scale. It depends on the mean squared error (MSE) between an original image and stego image , relative to $(2^n - 1)^2$ (the square of the highest-possible signal value in the image, where n is the number of bits per image sample) [7].

$$PSNR_{db} = 10\log_{10}\frac{(2^n-1)^2}{MSE} \quad (1)$$

"PSNR can be calculated easily and quickly and is therefore a very popular quality measure, widely used to compare the 'quality' of compressed and decompressed video images" [7].

3. METHODOLOGY

This study shows two Methods to hide text in gray scale image: Steganography using Least Significant Bit (called LSB) and Steganography using Least Significant Bit and Huffman Code (called LSB+HUFF).

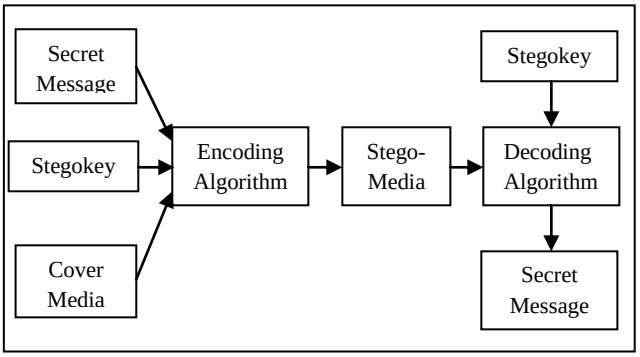
3.1 Steganography Technique

The following figure 4 shows the Steganography technique:

Steganography technique consists of the following phases as shown in Figure 3.

- Secrete Message: the information that you want to embed inside the cover media.
- Stegokey: the key used in the steganography process.
- Cover Media: the medium used in Steganography process (such as: image, video, audio, etc).
- Encoding Algorithm: the method used in Steganography process.
- Stego-Media: the medium resulting from the adding the secrete message into cover media using stegokey and encoding algorithm.
- Decoding Algorithm: the method used to extract the secrete message from Stego-media using Stegokey.

Fig 4: Steganography technique



3.2 Steganography using LSB

This method uses only the least significant bit (only one bit in each pixel). The size of data (Secrete Message) that you can imbed into the image is computed as the following:

$$S = M \times N - 40 \quad (2)$$

Where S is the size of the secret message, M is the number of rows in the image, N is the number of columns in the image, and (40) are illustrated in figure 4; where (1 to 20 the length of the secret message, 21 to 40 the length of the secret message too (for more security). Figure 5 shows that.

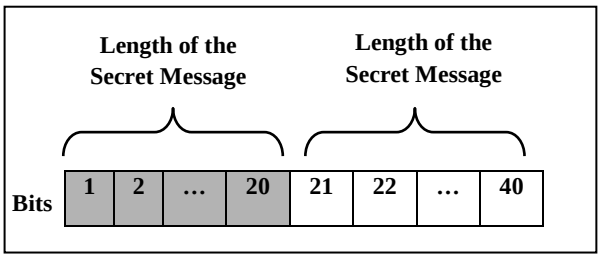


Fig 5: The length of the secrete message before Steganography process

3.3 Steganography using LSB+HUFF

This method relies on the compression of the secret message by using the method of Huffman code so as to reduce the size of the message data. LSB+Huff technique is better than LSB because the size of the secret message is less.

The size of secret message that can be concealed in this way is computed as the following:

$$S = M \times N - 80 \quad (3)$$

Where S is the size of the secret message, M is the number of rows in the image, N is the number of columns in the image, and (80) is illustrated in figure 5; where (1 to 20 are the length of the secret message, 21 to 40 are also the length of the secret message (for more security), 41 to 60 the length of the secrete message after Huffman code, and 61 to 80 are also the length of the secrete message after Huffman code (for more security)). Figure 6 shows that.

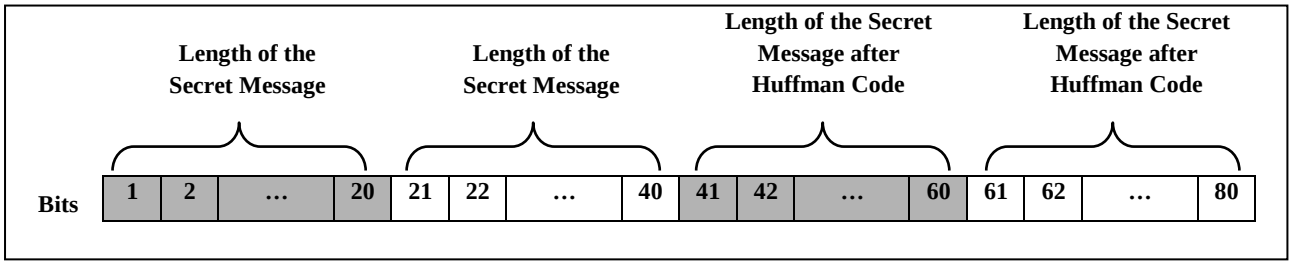


Fig 6: the length of the secret message and the length of the secrete message after Huffman code before Steganography

3.4 Encoding and Decoding of Image Steganography

This study is explained in the figure 7. It has the following steps:

First, Encoding Process

- Selecting an original image for example with the extension (.jpg, .jpeg, and .bmp) and converting it to (grayscale image).
- Converting the original image (Matrix of pixels) to column vector of pixels (Bytes) by using zigzag scanning.
- Transforming each elements of column vector to its binary representation to getting a matrix of bits.
- Identifying the secrete message as shown in figure 4 or figure 5 according to the Steganography method. Then, Compression the secrete message by applying the Huffman code (encode) to decrease the size of data.
- Substituting each bit from the secrete message to the least significant bit (LSB) of the matrix of bit.
- Finally, getting on stego-media.

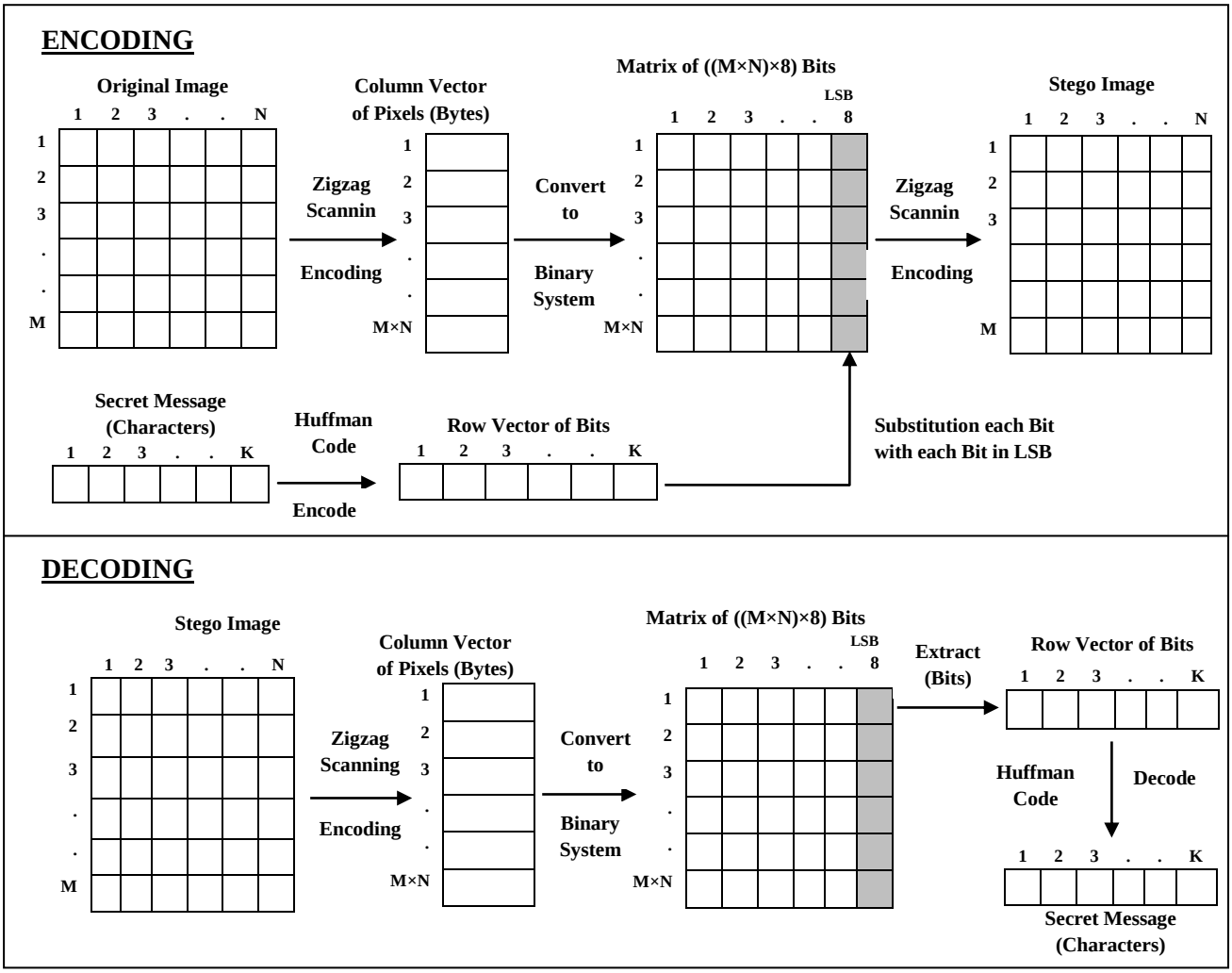


Fig 7: The Encoding and Decoding of this study.

Second, Decoding Process

- Selecting the stego-media.
- Converting the stego-image (Matrix of pixels) to column vector of pixels (byes) by using zigzag scanning.

- Transforming each element of vector to its binary representation to getting the matrix of bits.
- Extracting the specific bits from the Least Significant Bit (LSB) of the Matrix of bits according to the used method for Steganography. which is weather (LSB) or (LSB+Huff).

Getting the secrete message (characters) by applying the Huffman code (decode).

4. EXPERIMENTAL RESULTS

4.1 The Implementation

A program had been created which is called (Wa'el-Steganography) as shown in figure 8 to implement this research. Petra image was used (as an example) with size (404×446×3 pixels) and its extension is (jpeg). The following steps explain the technique:

- Convert the Petra image to gray scale.
- The size of data (secrete message) that can be embedded in this image by using LSB method is:

  404 × 446 – 40 = 180144.
- The size of data (secrete message) that can be embedded in this image by using LSB+HUFF method is:

  404 × 446 – 80 = 180104 bits.
- Select the Secrete Message (as an example) is the Abstract of this paper. The count of the bits of this abstract is:
- 642 characters × 7 bits = 4494 bits.
- Choose the Steganography method; weather (LSB) or (LSB+HUFF).
- Compare the results between (LSB) method and (LSB+HUFF) method using the PSNR.

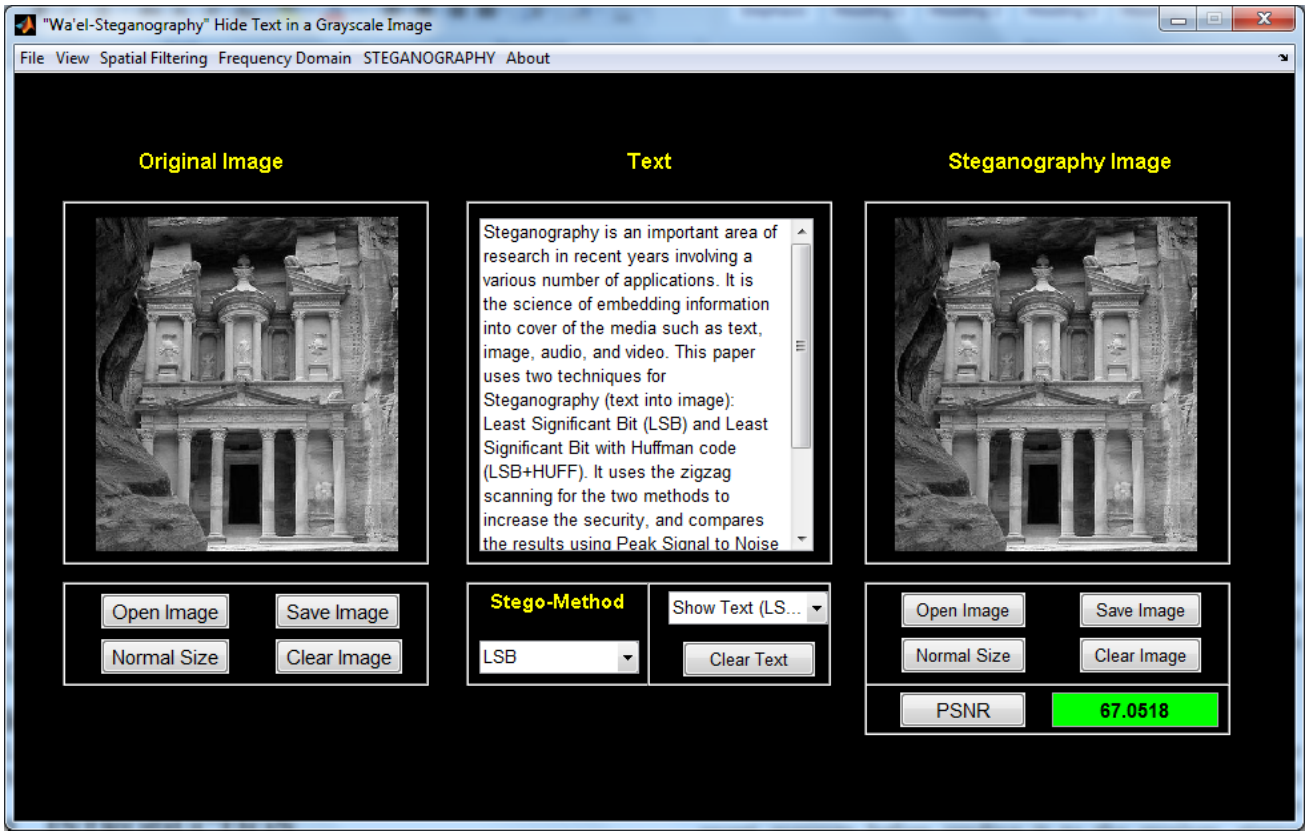


Fig 8: (Wa'el-Steganography) program

4.2 The Results

Table 4 clears the results after the implementation; for all results the PSNR of LSB+HUFF method is more than the PSNR of LSB method. So the LSB+HUFF method is better than the LSB method in Steganography process.

Table 4: comparison the PSNR between LSB and LSB+HUFF methods

Number of times copied the Abstract to this paper	# bits	Steganography Method	
		LSB (PSNR)	LSB+HUFF (PSNR)
1	4494	67.0518	68.1016
5	22470	60.183	61.8713
10	44040	57.1853	58.9593
15	67410	55.4365	57.2523
20	89880	54.181	56.0208

Figure 9 shows the results (as a diagram) for the same values in table 4.

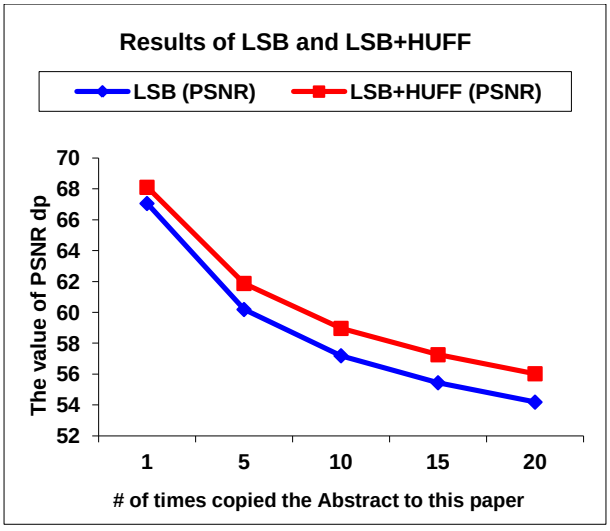


Fig 9: The PSNR of LSB and LSB+HUFF

5. CONCLUSION

This paper explains the Steganography technique in the digital image, and it offers new technique for Steganography using (Least Significant Bit, Zigzag Scanning, and Huffman code). This paper contains two techniques for Steganography: The first is Least Significant bit, and the second is Least Significant bit and Huffman code. By comparing the results between the two techniques it concludes the LSB+HUFF method is better than LSB method to hide text into image. In the future, hope of obtain a new method for compression data with lowest size to decrease the size of secrete message.

6. REFERENCES

[1] Prof.S.V.Kamble and Prof. B.G.Warvante. A Review on Novel Image Steganography Techniques. IOSR Journal of Computer Engineering (IOSR-JCE), ISSN: 2278-0661, ISBN: 2278-8727, PP: 01-04.

[2] Hniels Provos and Peter Honeyman. Hide & Seek An Introduction to Steganography. IEEE Computer Society Pub-2003.

[3] József LENTI. STEGANOGRAPHIC METHODS. PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 44, NO. 3–4, PP. 249–258 (2000).

[4] Joyshree Nath, Sankar Das, Shalabh Agarwal, and Asoke Nath. Advanced Steganographic Approach for Hiding Encrypted Secret Message in LSB, LSB+1, LSB+2 and LSB+3 Bits in Non standard Cover Files. International Journal of Computer Applications (0975 – 8887), Volume 14– No.7, February 2011.

[5] Michael E. Whitman and Herbert J. Mattord. Principles of Information Security. Fourth Edition.

[6] Rafael C. Gonzalez and Richard E. Woods. DIGITAL IMAGE PROCESSING. Third Edition.

[7] Iain E. G. Richardson. H.264 and MPEG-4 Video Compression.

[8] R.Poornima and R.J.Iswarya. AN OVERVIEW OF DIGITAL IMAGE STEGANOGRAPHY. International Journal of Computer Science & Engineering Survey (IJCSES) Vol.4, No.1, February 2013.

[9] Bret Dunbar. Steganographic Techniques and their use in an Open-Systems Environment. SANS Institute Reading Room site, 18 February 2002.

[10] https://www.siggraph.org/education/materials/HyperGraph/video/mpeg/mpegfaq/huffman_tutorial.html. last view 22-7-2014 10:33 am.