FPGA Implementation of Robust Image Steganography Technique based on Least Significant Bit (LSB) in Spatial Domain

E. A. Elshazly, Safey A. S. Abdelwahab
Nuclear Research Center, Atomic Energy Authority, Cairo, Egypt.

R. M. Fikry, S. M. Elaraby
Nuclear Research Center, Atomic Energy Authority, Cairo, Egypt.

O. Zahran, M. El-Kordy
Faculty of Electronic Engineering, Menoufia University, Menouf, Egypt.

ABSTRACT

There are many different data hiding techniques, the Least Significant Bit (LSB) based steganography algorithm is considered as one of the most popular algorithms in the spatial domain. In this paper, the proposed algorithm embeds data in each component of color image, where the signature of the transmitter and the length of the secret text are hidden in Red component, while the binary bit stream of the secret text is hidden in Green and Blue components of the color image. After embedding, the three components are re-combined to form a stego-image. The stego-image is passing through a communication channel and a noise may be added to it. At the receiver, the hidden text can be extracted from the noisy stego-image without any knowledge of the original image after applying a filtration in the pre-processing stage. The embedding and extracting processes in the proposed algorithms are performed using MATLAB and implemented on a field programmable gate array (FPGA) using Xilinx system generator (XSG) based on Hardware/Software Co-simulation. The implementation of the proposed algorithms on FPGA has the advantages of using an embedded multipliers and large memory. The Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) are used to check and measure the statistical distortion between the cover image and stego-image, while the Normalized Cross Correlation (NCC) is used to evaluate the degree of closeness between them. The experimental results are showing the efficiency of the proposed algorithms as well as proving that embedding larger size of data with better results of MSE and PSNR.

General Terms

Data hiding.

Keywords

Image Steganography, LSB, MSE, PSNR, NCC, MATLAB, FPGA and XSG.

1. INTRODUCTION

There is a large number of data hiding techniques that can be used to hide a secret text within a digital image. Image steganography is one of the most widely acceptable data hiding techniques. The word steganography is derived from the Greek words “steganos” which means “covered” and “graphie” which means “writing” that defining it as “covered writing”. Steganography is concerned with transmitting a secret message while hiding its existence [1]. It is the art and science of embedding a secret text in a cover image without leaving a remarkable track on the cover image. The secret text is hidden in the cover image and the resulting image is called a "stego-image". The stego-image is passing through a communication channel. At the receiving, a pre-processing stage is applied to the received stego-image, then the secret text can be extracted from it [2].

There are two general approaches to categorize the systems of steganography techniques. According to the type of the cover object, the steganography techniques are categorized into image steganography, video steganography, network steganography, text steganography and audio steganography [3]. According to the embedding techniques, the steganography techniques are categorized into six main types which are those embedding in the spatial domain, such as Least Significant Bit (LSB) based approaches and Pixel Value Differencing (PVD) based approaches, those embedding in the transform domain such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), spread spectrum techniques, distortion techniques, masking and filtering techniques, and cover generation techniques [4].

This paper focuses on a LSB based image steganography algorithm in the spatial domain as it considered as one of the famous approaches in the spatial domain that used for embedding the existence of the secret text in a cover image in which the LSBs of a cover image are changed according to the bit stream of the secret text to be hidden. Figure (1) shows an overview of image steganography algorithm [5].

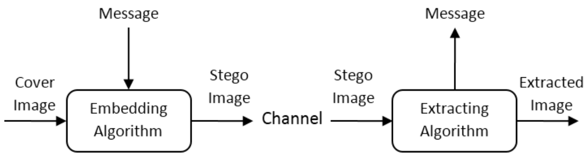


Fig 1: An overview of image steganography algorithm.

The LSB based image steganography algorithm consists of two phases; the embedding phase and the extracting phase. During the embedding phase, the algorithm reads the cover image and embed the secret text in it in such a way that it could not be detected by a human vision, the resulting image (stego-image) is obtained and travels through a communication channel. During the extracting phase, the hidden text is extracted from the LSBs of the stego-image. In this paper, the embedding and extracting phases of the proposed algorithms are performed using MATLAB and implemented using Xilinx System Generator (XSG) based on Hardware/Software Co-simulation [6].

In the embedding phase, the Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) are used to check and measure the statistical distortion amount between cover image and stego-image. In addition to the Normalized Cross

Correlation (NCC) is used to evaluate the degree of closeness between cover and stego-images, while in the extracting phase, the MSE and PSNR are used to and measure the distortion amount between original text to be hidden and the extracted text [7].

The remaining of this paper is organized as follows; Section 2 introduces a Xilinx system generator, Section 3 presents the design flow for image steganography with Xilinx System Generator, Section 4 gives a review of the LSB algorithm, Section 5 presents the proposed LSB based image steganography algorithm, Section 6 discusses the Xilinx system generator models for the proposed LSB based image steganography algorithms, Section 7 presents the FPGA hardware co-simulation, Section 8 gives the experimental results and discussion and finally Section 9 presents the conclusions.

2. XILINX SYSTEM GENERATOR

The XSG is an Integrated Design Environment (IDE) for FPGAs within the Integrated Synthesis Environment (ISE) development suite, which often utilizes Simulink as a development environment and is introduced in the form of model based design. It has an integrated design flow to move directly to the actual bitstream file (*.bit) from Simulink design environment which is essential for programming the actual FPGA [6]. One of the most important features of XSG will be owned the abstraction arithmetic which is dealing with representation in fixed point with a precision arbitrary, including quantization and overflow. XSG may just perform simulations as a fixed-point double precision type and automatically generates VHDL code as well as a draft of the ISE model being developed. It also generates hierarchical VHDL/Verilog synthesis, floor plan and mapping hardware, together with generating a user constraint file (UCF), simulation and test bench and test vectors among other things.

The XSG was created primarily to deal with complex DSP applications, but it also deals with implementation of many applications. The Xilinx blocks operate with Boolean values or in a discrete-time and a fixed point formats, for a better approach to hardware implementation, while the Simulink works with numbers of double precision floating point. The connection between the Simulink Blocks and Xilinx Blocks are gateway blocks, for conversion between the double precision floating point and a fixed point formats. The design entry is the first step for FPGA design flow. It is performed either using a Hardware Description Language (HDL) or a Schematic based approach in conventional FPGA design [8]. The FPGA based hardware implementation of LSB based image steganography is proposed and implemented on FPGA board using a combination of Simulink blocks and Xilinx Blockset. One of the advantages of using the XSG for hardware implementation is that, the Xilinx Blockset provides close integration with MATLAB Simulink which helps in co-simulating the FPGA module with pixel vector provided by MATLAB Simulink Blocks [9].

3. DESIGN FLOW FOR LSB IMAGE STEGANOGRAPHY WITH XSG

Two software tools are required to perform and implement the task of LSB based image steganography; which are MATLAB Version R2008a or higher to perform the task and the Xilinx ISE 11.1 or higher for implementation. The system generator token must be configured to MATLAB to add Xilinx Blockset to the MATLAB Simulink environment which can be directly utilized for building the models for LSB based image steganography using library provided by Xilinx Blockset. The image pixels are provided to the Xilinx models in Xilinx fixed point format. These models are simulated in MATLAB Simulink environment with suitable simulation time and simulation mode and tested. Once the results are obtained, the System Generator is configured for suitable FPGA board. FPGA board Spartan-3E starter kit is used here and the model is implemented for JTAG hardware co-simulation. The system generator parameters are set and generated. On compilation, the netlist is generated and a draft for the model and programming file in Verilog HDL are formed which can be accessed using Xilinx ISE. The module is checked for behavioral syntax, synthesized and implemented on FPGA. Bitstream compilation is done to create an FPGA bit file which is suitable for FPGA input [10]. Figure (2) shows the design flow for XSG.

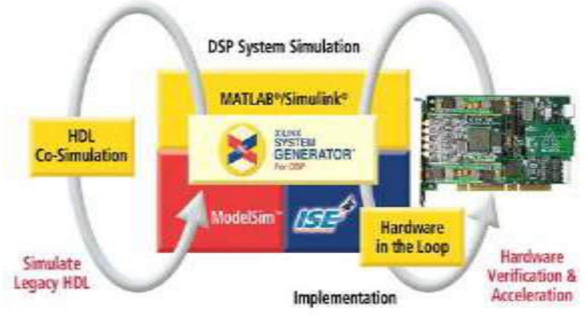


Fig 2: Design flow of XSG system.

4. A REVIEW OF LSB ALGORITHMS

The Least Significant Bit (LSB) is the most well-known algorithm to hide a secret text in an image. The LSB embedding algorithm replaces the LSBs of the pixels of the cover image by the bit stream of the secret text to be hidden. The stego-image is almost similar to the cover image as the changes in the LSBs of image pixels do not introduce too much differences in the image. If the LSB of the pixel value $I_C(i, j)$ is equal to the message bit m to be embedded, $I_C(i, j)$ remain unchanged; if not, the LSB of $I_C(i, j)$ is set to m. The stego-image is obtained according to the following formula [11, 12].

$$I_S(i,j)=\begin{cases} I_C(i,j)-1 & \mathrm{LSB}(\mathrm{I_C(i,j)})=1 \ \mathrm{and}\ \mathrm{m}=0\\ I_C(i,j) & \mathrm{LSB}(\mathrm{I_C(i,j)})=\mathrm{m}\\ I_C(i,j)+1 & \mathrm{LSB}(\mathrm{I_C(i,j)})=0\ \mathrm{and}\ \mathrm{m}=1\end{cases} \tag{1}$$

where LSB ($I_C(i, j)$) stands for the LSB of $I_C(i, j)$ and m is the next bit to be embedded. In the case of 24-bit images, three bits can be embedded in each pixel by changing a bit in each color component. Given three adjacent pixels (9 bytes) with the RGB encoding as follows [13].

10010101 00001101 11001001
10010110 00001111 11001011
10011111 00010000 11001011

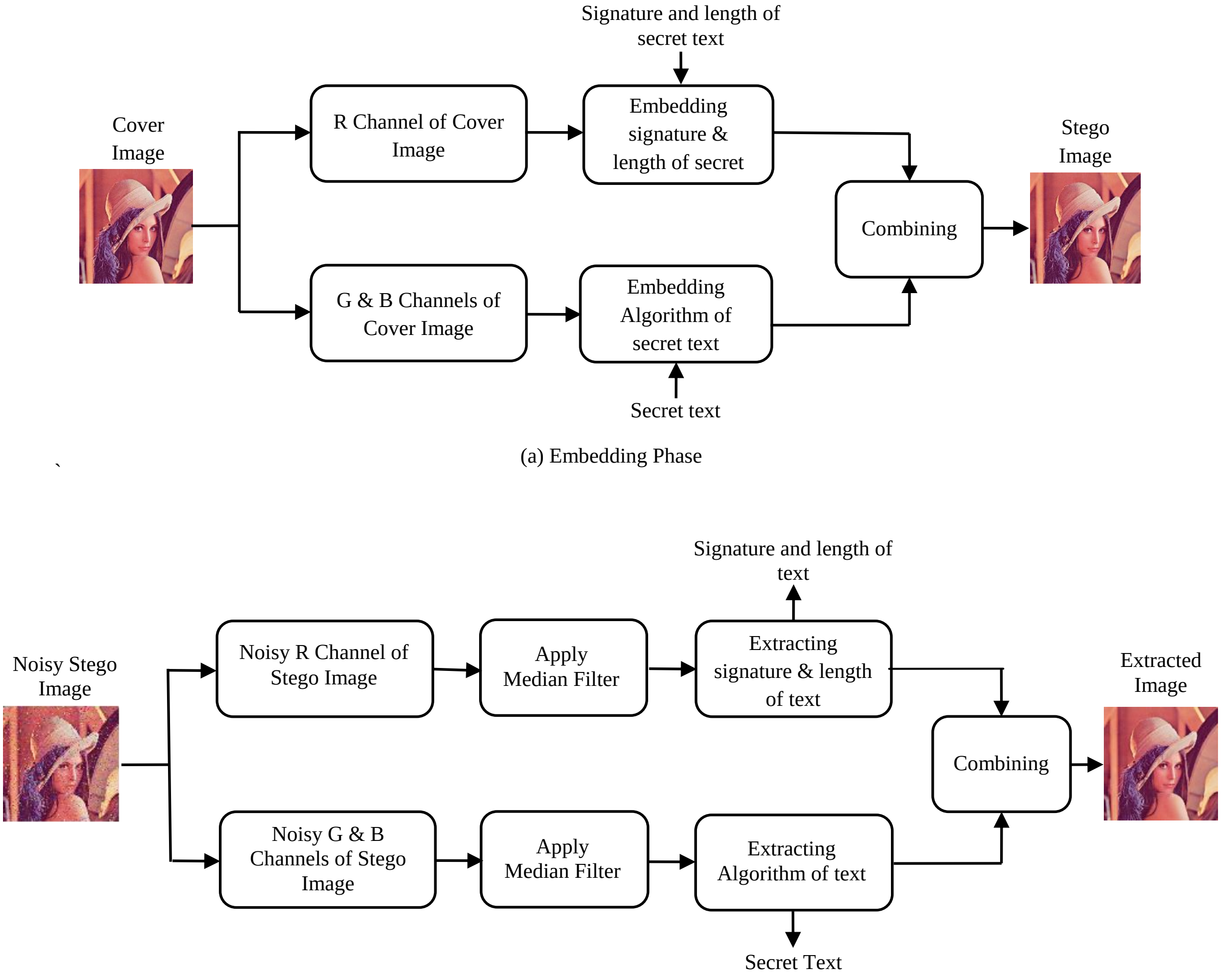
To embed the number 400, with a binary representation (110 010 000), into the LSBs of the above part of the image (9 bytes), we get the following new 9 bytes:

1001010**1** 0000110**1** 1100100**0**
1001011**0** 0000111**1** 1100101**0**
1001111**0** 0001000**0** 1100101**0**

The number 400 is hidden into the grid, only 4 bits need to be changed according to the message to be embedded. This means that, only 50% of pixels of an image are needed to be modified to embed a secret message in the case of full capacity (the number of bits in the secret message equals to the number of pixels in the cover image). Using the human eye, it will be very hard to notice the difference between the cover image and the stego-image [14, 15].

5. THE PROPOSED LSB BASED IMAGE STEGANOGRAPHY ALGORITHM

In the proposed technique, all the three color components have been used for data embedding after decomposing color components to obtain three separated MxN matrices, one for each color component, where the original image size is MxN. After that, LSB method is applied for data hiding in each matrix separately, but in a sequential manner [16, 17, 18]. The proposed LSB algorithm applied on color images has two phases; an embedding phase and an extracting phase. In the embedding phase, the cover image is decomposed into Red, Green and Blue (RGB) channels, then the signature of the sender and the length of the secret text to be hidden are embedded in the LSBs of the pixels of Red channel, while the binary bit stream of the secret text is embedded in the LSB of each pixel in the Green and Blue channels of the color image. After embedding process, the color channels are concatenated to produce a stego-image, which is passing through a communication channel, and a noise may be added to it. The type of noise taken into consideration is "impulsive noise", as it adds a dark pixels in bright regions and bright pixels in dark regions. In the extracting phase, the noisy stego-image is decomposed into Red, Green and Blue channels, then a median filter is applied to each channel, the hidden message can be extracted without any knowledge of the cover image. The signature of the transmitter and the length of the secret text are extracted from the LSBs of the pixels of Red channel, while the binary bit stream of the secret text is extracted from the LSB of each pixel in the Green and Blue channels of the color image. After extracting process, the color channels are concatenated to produce the extracted image. The embedding and extracting processes are performed using a MATLAB software and implemented using FPGA. The proposed LSB image steganography algorithm applied on color images is shown in Figure (3).


Signature and length of secret text
Cover Image
R Channel of Cover Image
Embedding signature & length of secret
Stego Image
Combining
G & B Channels of Cover Image
Embedding Algorithm of secret text
Secret text
Signature and length of text
Noisy Stego Image
Noisy R Channel of Stego Image
Apply Median Filter
Extracting signature & length of text
Extracted Image
Combining
Noisy G & B Channels of Stego Image
Apply Median Filter
Extracting Algorithm of text
Secret Text

(a) Embedding Phase

(b) Extracting Phase

Fig 3: The proposed LSB based image steganography algorithm.

In the proposed algorithms, 512x512 color images have been used as cover images. So, a secret text up to 98,304 characters can be hidden. The secret text is embedded in the LSBs of the cover image to protect the message against the external influences and to increase the robustness of the system. To minimize the difference between the pixels in the cover image and the new pixels in the stego-image, the following embedding and extracting algorithms are proposed.

Steps of the proposed embedding algorithm:

Inputs: Cover image and secret text.

Outputs: Stego image (image containing a hidden secret text).

Step 1: Read the cover color image.

Step 2: Decompose the cover image into R, G and B channels.

Step 3: Read the signature and the secret message and estimate the length of the secret text.

Step 4: Convert the R, G and B channels, the signature, the length of secret text and the secret text from decimal to binary.

Step 5: Replace the LSBs of the first 16 pixels of the R channel of the cover image by the bit stream of the signature, then replace the LSBs of the 17th to 48th pixels of the R channel of the cover image by the bit stream of the length of the secret text to be hidden.

Step 6: Replace the LSBs of the G and B channels of the cover image by the binary bit stream of the secret text to be hidden.

Step 7: Steps 5 and 6 will continue until the signature, the length of the text and the complete text are completely hidden into the image.

Step 8: Compose again the R channel (containing the embedded signature and length of secret text) and the G and B channels (containing the secret text).

Step 9: Write the stego-image.

Step 10: Calculate the MSE, PSNR and the NCC of the stego-images.

Steps of the proposed extracting algorithm:

Inputs: Noisy Stego image.

Outputs: Secret text.

Step 1: Read the noisy stego-image.

Step 2: Decompose the noisy stego-image into R, G and B channels.

Step 3: Apply the Median filter to each channel

Step 4: Extract the signature and the length of secret text from the R channel.

Step 5: Extract the secret text from the G and B channels.

Step 6: Steps 4 and 5 will continue until the signature, the length of the text and the secret text are completely extracted.

Step 7: Compose again the R channel (after extracting signature and length of secret text), and the G and B channels (after extracting the secret message).

Step 8: The extracted image is now formed.

Step 9: The secret text is now extracted.

6. THE XSG MODELS

The entire implementation of the proposed LSB based image steganography algorithm using Simulink and Xilinx blocks is done through three phases [6, 8].

1. The pre-processing phase.
2. The LSB based Image steganography algorithm.
3. The post-processing phase.

The design flow of the hardware implementation of a LSB based image steganography algorithm using XSG is presented in Figure (4). The image pre-processing and image post-processing units are designed using Simulink blocksets, while LSB based image steganography algorithm is implemented using XSG blocksets [6, 8, 9].

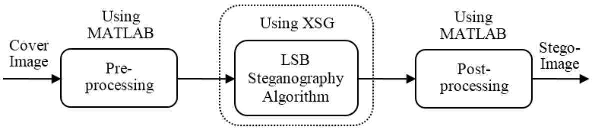


Fig 4: Design flow of hardware implementation of LSB based image steganography.

6.1 The pre-processing phase

The image pre-processing is done using MATLAB Simulink that provides an input to FPGA as a specific vector array which is suitable for FPGA bitstream compilation using system generator, as the image is converted into a single array of pixel (Convert 2-D to 1-D).

6.2 The LSB based image steganography algorithm

Four different cases of LSB applied on color images are proposed (LSB-1, LSB-2, LSB-3 and LSB-4), this means embedding the bits of a secret text in the 1st LSB, the 2nd LSB, the 3rd LSB, and the 4th LSB of each pixel respectively. The model based design using Xilinx blocksets for the proposed LSB-1 algorithm applied on color images is shown in Figure (5). The proposed LSB-2 algorithm applied on color images is shown in Figure (6). The proposed LSB-3 and LSB-4 are the same as LSB-2 after changing their parameters according to the 3rd and the 4th bit of each pixel respectively.

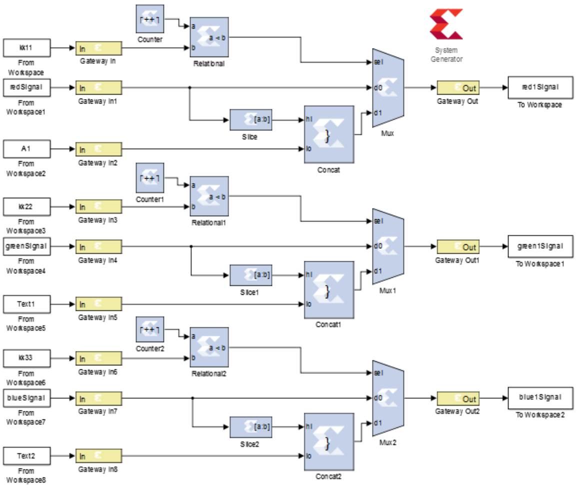


Fig 5: FPGA implementation of the proposed LSB-1 image steganography algorithm.

6.3 The post-processing phase

The image post-processing blocks which are used to convert the output of image back to floating point type, as the 1-D image signal is converted to 2-D image matrix (Convert 1-D to 2-D).

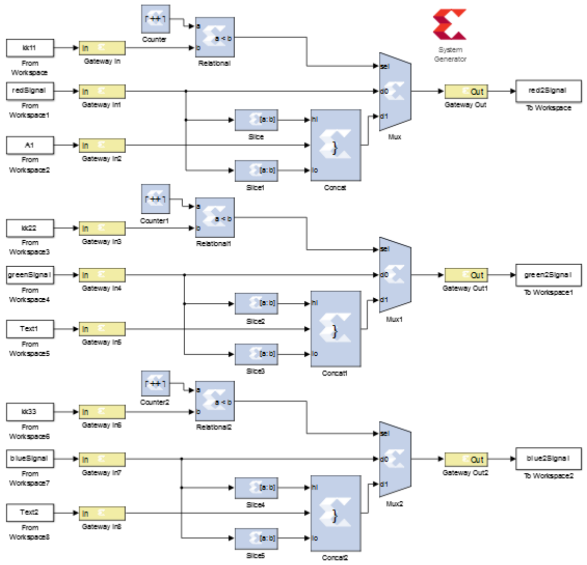


Fig 6: FPGA Implementation of the proposed LSB-2 image steganography algorithm.

7. THE FPGA H/W CO-SIMULATION

The Embedding/Extraction is hardware generated through the system generator, target board Spartan 3E starter kit and JTAG cable. A model can be co-simulated provided it to meet the requirements of the underlying hardware board. Now, it can be verified that; the results using hardware and the results using software are the same [9, 10]. The hardware co-simulation for LSB-1 based image steganography algorithm applied on color images is shown in Figure (7).

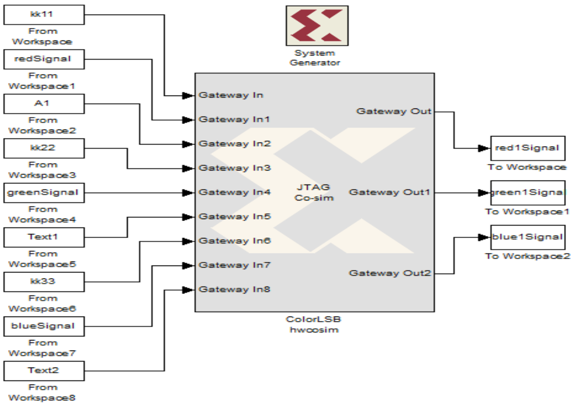


Fig 7: The hardware Co-Simulation for LSB-1 based image steganography algorithm.

8. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed LSB-based steganography algorithms applied on color images have been done on various images, and a comparative analysis of the proposed algorithms is investigated. The performance evaluation parameters in all experiments are the MSE, PSNR and NCC. In the proposed algorithms 512x512 color images have been used as cover images. The secret text up to 98,304 characters can be hidden. The proposed LSB algorithms are performed using MATLAB software and implemented using FPGA.

The MSE and the PSNR are the two error metrics that used to evaluate the efficiency of the above algorithms. The MSE represents the cumulative squared error between the cover image and the stego-image, whereas the PSNR represents a measure of the peak error [19, 20, 21].

If I_C and I_S are the cover image and stego-image respectively, then the MSE and PSNR can be calculated using the following equations [20, 21]:

$$MSE = \frac{1}{MxN}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[I_S(i,j) - I_C(i,j)]^2 \qquad (2)$$

$$MSE_{RGB} = \left[\frac{MSE_R + MSE_G + MSE_B}{3}\right] \qquad (3)$$

$$PSNR = 10\log_{10}\left(\frac{C^2{}_{max}}{MSE_{RGB}}\right) \qquad (4)$$

Where M and N are the number of rows and columns in the cover images respectively and C_{max} holds the maximum value in the original images. The NCC is one of the best known methods that evaluate the degree of closeness between the cover image and the stego-image. The NCC can be calculated using the following equation [22]:

$$NCC = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} I_C(i,j) \text{ x } I_S(i,j)}{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} I_C{}^2(i,j)} \qquad (5)$$

Table (1) summarizes the values of the MSE, PSNR and NCC for the proposed LSB algorithms applied on color images, while Table (2) summarizes a comparison between PSNR of the previous LSB algorithms and the proposed LSB algorithms. Table (3) represents a comparison between the capacity of the previous LSB algorithms and the proposed LSB algorithms. It is clear that it is better to hide data in the first LSB because the PSNR values are more than 55 dB, the values of MSE are less than 0.2 and the NCC values are 0.9999. Figure (8) shows the results of the proposed LSB steganography algorithms applied on color images, the cover images and stego-images. Figure (9) shows the stego, noisy stego, restored and extracted images for embedding and extracting techniques applied on color images at 5% percentage error, while Figure (10) shows the effect of the percentage of impulsive error on the MSE and PSNR of the extracted text compared with the original text. Figure (11) shows the effect of different percentage of impulsive error on the PSNR values of stego, noisy stego, restored and extracted images for the proposed embedding and extracting techniques, while Figure (12) shows the effect of changing the length of secret text on the PSNR for the image steganography algorithms, and it is clear that LSB-1 image steganography algorithm provides the best results. The MSE and PSNR are used to check and measure the statistical distortion amount between cover image and stego-image, while NCC is used to show the amount of deflection in the stego-image with respect to the cover image after insertion of the secret text. A large PSNR and NCC values indicate that, the visual quality of the stego-image is good which means that, a small amount of distortion has occurred.

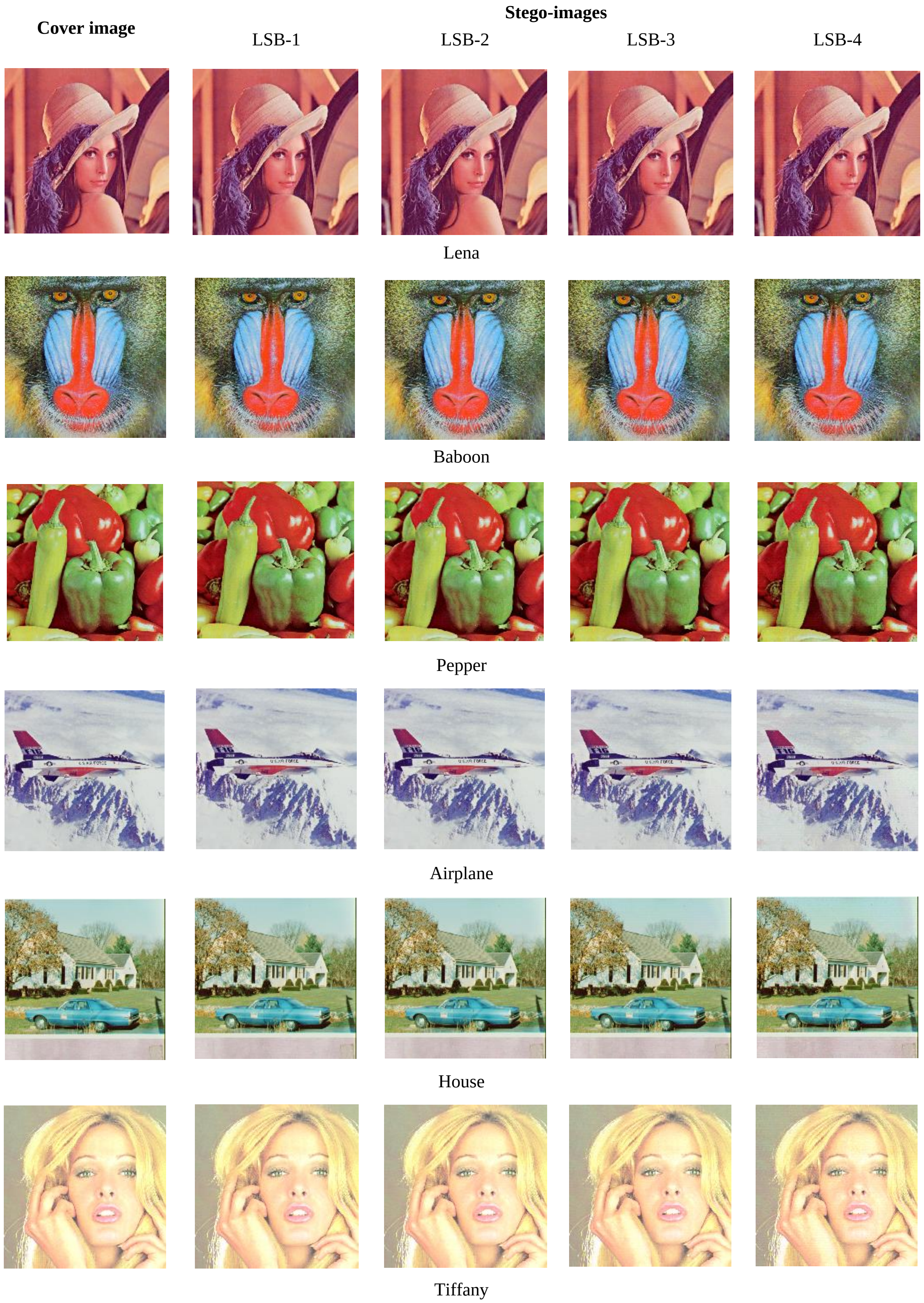


Fig 8: Cover images and their corresponding stego-images for LSB-1, LSB-2, LSB-3, and LSB-4 embedding techniques.

Table 1. MSE, PSNR and NCC values for LSB embedding techniques.

Embedding Techniques	Evaluation Parameters	Color Images (512x512)					
		Lena	Baboon	Pepper	Airplane	House	Tiffany
LSB-1	MSE	0.1838	0.1814	0.1703	0.1795	0.1792	0.1870
	PSNR	55.4871	55.5434	55.1830	54.8429	55.5970	55.4113
	NCC	0.9999	0.9999	0.9999	0.9999	0.9999	0.9998
LSB-2	MSE	0.7181	0.7174	0.6766	0.7160	0.7216	0.7399
	PSNR	49.5687	49.5734	49.1919	48.8352	49.5481	49.4393
	NCC	0.9996	0.9998	0.9998	0.9996	0.9998	0.9992
LSB-3	MSE	2.8868	2.8800	2.6902	2.8650	2.8595	2.9812
	PSNR	43.5267	43.5368	43.1972	42.8130	43.5680	43.3870
	NCC	0.9984	0.9991	0.9991	0.9982	0.9991	0.9967
LSB-4	MSE	11.7428	11.5235	10.6799	11.6664	10.7655	12.0517
	PSNR	37.4331	37.5150	37.2093	36.7149	37.8105	37.3203
	NCC	0.9936	0.9963	0.9964	0.9930	0.9964	0.9868

Table 2. PSNR comparison of the previous LSB and the proposed LSB embedding techniques [23].

Color Images (512x512)	LSB Technique	1-3-4 Technique	Proposed Technique			
			LSB-1	LSB-2	LSB-3	LSB-4
Lena	41.0053	47.5897	55.4871	49.5687	43.5267	37.4331
Baboon	33.9879	36.3637	55.5434	49.5734	43.5368	37.5150
Pepper	42.3743	45.9238	55.1830	49.1919	43.1972	37.2093
Airplane (Jet)	39.3827	42.5050	54.8429	48.8352	42.8130	36.7149

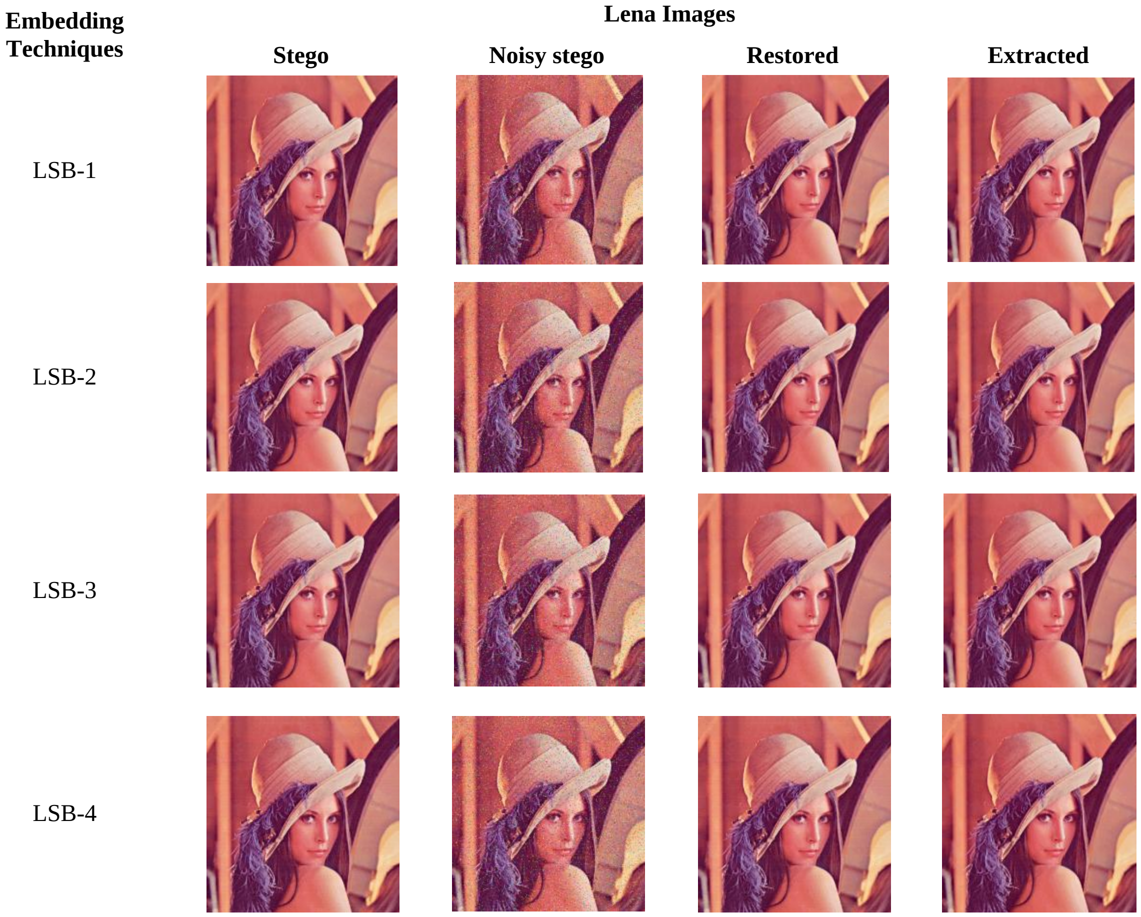
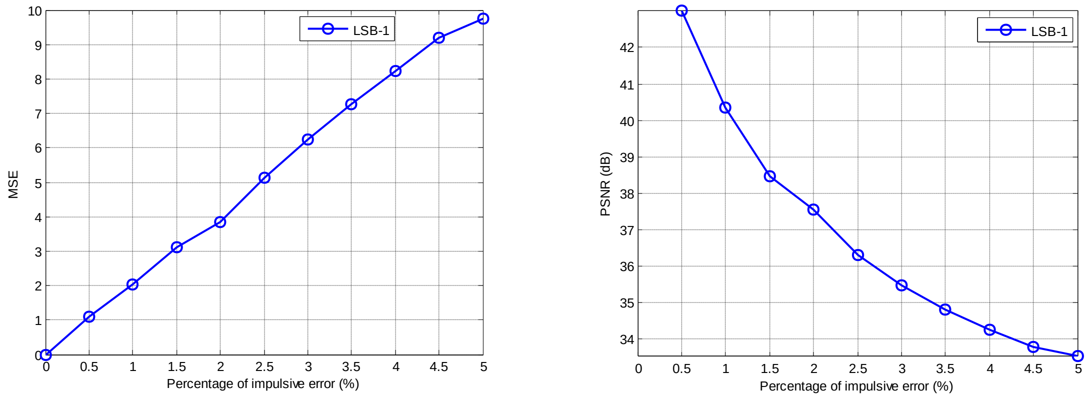


Fig 9: Stego, Noisy stego, Restored and Extracted images for LSB embedding and extracting techniques at 5% percentage of impulsive error.

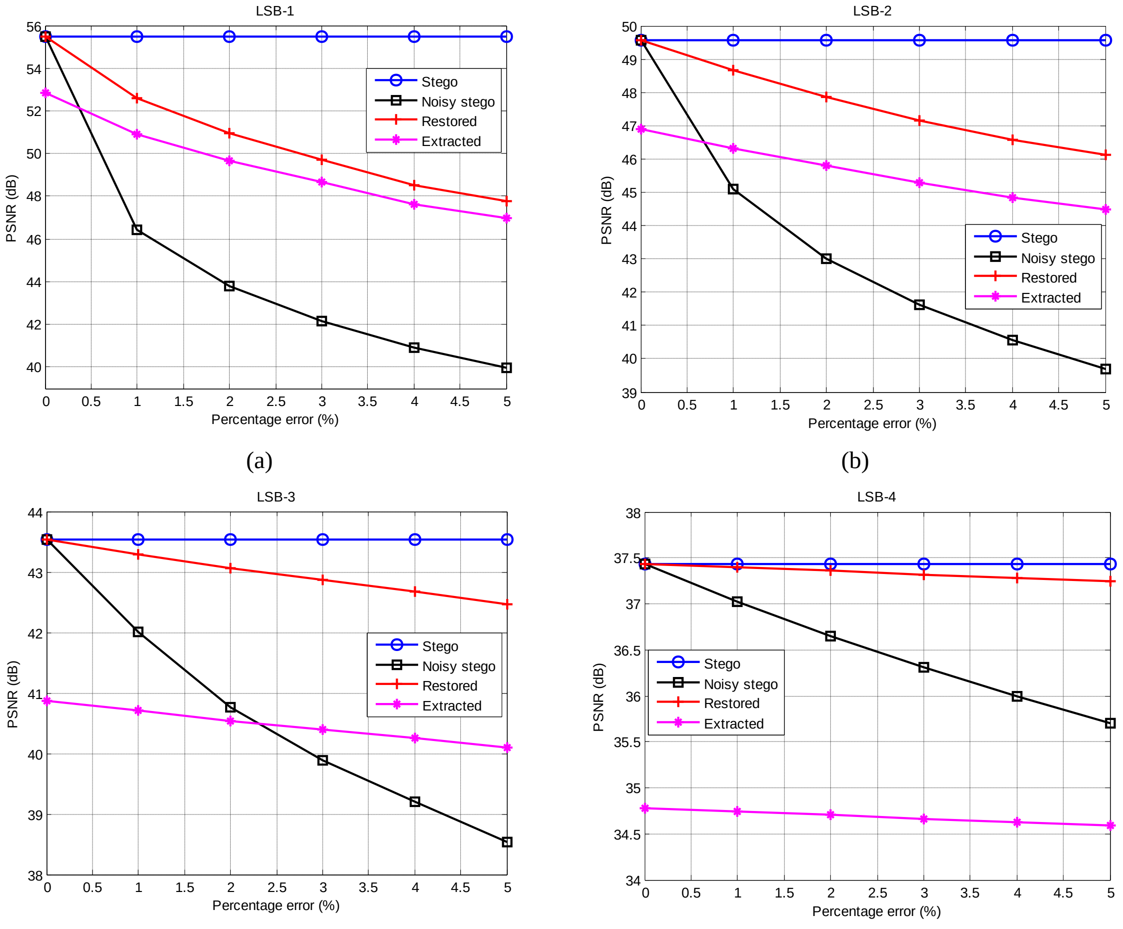
Table 3. PSNR comparison of the previous LSB and the proposed LSB embedding techniques [23].

Color Images (512x512)	LSB Technique	1-3-4 Technique	Proposed Technique			
			LSB-1	LSB-2	LSB-3	LSB-4
Lena	467004	561345	786432			
Baboon	720785	830546				
Pepper	482599	588459				
Airplane (Jet)	463758	505658				



(a) (b)

Fig 10: The effect of percentage of impulsive error on (a) MSE. (b) PSNR.



(a) (b)

(c) (d)

Fig 11: The effect of percentage of impulsive error on the PSNR values of stego, noisy stego, restored and extracted images for the proposed LSB embedding and extracting techniques (a) LSB-1. (b) LSB-2. (c) LSB-3. (d) LSB-4.

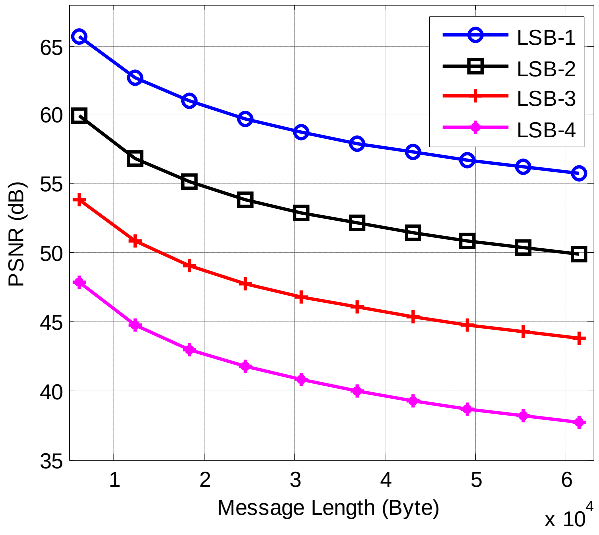


Fig 12: The effect of changing the length of text on the PSNR for LSB image steganography algorithms.

9. CONCLUSIONS

This paper proposes LSB based image steganography algorithms for embedding data in color images. Four different cases of LSB are considered; which are LSB-1, LSB-2, LSB-3 and LSB-4. The proposed LSB algorithms are performed using MATLAB software and implemented using FPGA. The proposed algorithm is based on embedding the signature of the transmitter and the length of secret text in the Red channel and the secret text itself in the other two channels (Green and Blue). After the embedding process is finished, the three channels are re-combined to form a stego-image. This image is passing through a communication channel, a noise may be added to it. The type of noise taken into consideration is "impulsive noise". Furthermore, the proposed algorithms extract the hidden secret text efficiently without using the cover image. The experimental results prove that the proposed algorithm can embed larger secret text (up to 98,304 characters) with better results of PSNR and NCC, compared to the previous algorithms. The results also show that the PSNR increases as the percentage of impulsive error decreases.

10. REFERENCES

[1] Al-Korbi H. A., Al-Ataby A., Al-Taee M. A., Al-Nuaimy W., 2015. High-Capacity Image Steganography Based on Haar DWT for Hiding Miscellaneous Data. Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). IEEE. 1 – 6, (Nov. 03 – 05) Amman. Jordan.

[2] Thangadurai K. and Sudha Devi G. 2014. An analysis of LSB based image steganography techniques. International Conference on Computer Communication and Informatics (ICCCI). IEEE. 1 – 4, (Jan. 03 – 05). Coimbatore. India.

[3] Kaul N. and Bajaj N. 2013. Audio in Image Steganography based on Wavelet Transform. International Journal of Computer Applications (IJCA). VOL. 79. No. 3. 7 – 10.

[4] Christaline J. A. and Vaishali D. 2011. Image Steganographic Techniques with Improved Embedding Capacity and Robustness. International Conference on Recent Trends in Information Technology (ICRTIT). IEEE. 97 – 101 (June 03 – 05). Chennai. Tamil Nadu.

[5] Singla D. and Juneja M. 2014. An Analysis of Edge Based Image Steganography Techniques in Spatial Domain. Recent Advances in Engineering and Computational Sciences (RAECS). IEEE. 1 – 5. (March 06 – 08). Chandigarh. India.

[6] Christe S. A. Vignesh M. and Kandaswamy A. 2011. An Efficient FPGA Implementation of MRI Image Filtering and Tumour Characterization Using Xilinx System Generator. International Journal of VLSI design & Communication Systems (VLSICS) VOL. 2. No. 4. 95 – 109.

[7] Sun H., Luo H., Tin-Yu Wu and Obaidat M. S. 2015. A PSNR-Controllable Data Hiding Algorithm Based on LSBs Substitution. Global Communications Conference (GLOBECOM). IEEE. 1 – 7. (Dec. 06 – 10). San Diego. CA.

[8] Sujatha C. and Selvathi D. 2014. Hardware Implementation of Image Edge Detection Using Xilinx System Generator. Asian journal of scientific research. 1 – 11.

[9] Warkari D. S. and Kshirsagar U. A. 2015. FPGA Implementation of Point Processing Operation using Hardware Simulation. International Journal of Advanced Research in Computer and Communication Engineering, VOL. 4. Issue 4. 91 – 95.

[10] Dakre K. A. and Pusdekar P. N. 2015. Image Enhancement using Hardware co-simulation for Biomedical Applications. International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC). VOL. 3. Issue 2. 869 – 877.

[11] Yang C. Liu F. Luo X. and Zeng Y. 2013. Pixel Group Trace Model-Based Quantitative Steganalysis for Multiple Least-Significant Bits Steganography. IEEE Transactions on Information Forensics and Security. VOL. 8. No. 1. 216 – 228.

[12] Joshci R. Gagnani L. and Pandey S. 2013. Image Steganography with LSB. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). VOL. 2. Issue 1. 228 – 229.

[13] Deepa S. and Umarani R. 2013. A Study on Digital Image Steganography. International Journal of Advanced Research in Computer Science and Software Engineering. VOL. 3. Issue 1. 54 – 57.

[14] Chen M-C, Agaian S. S. and Chen C. L. P. 2008. Generalized Collage Steganography on Images", International Conference on Systems, Man and Cybernetics (SMC). IEEE. 1043 – 1047. (Oct. 12 – 15). Singapore.

[15] Deshmukh P. U. and Pattewar T. M. 2014. A Novel Approach for Edge Adaptive Steganography on LSB Insertion Technique International Conference on Information Communication and Embedded Systems (ICICES). IEEE. 1 – 5 (Feb. 27 – 28). Chennai. India.

[16] Mandal J. K. and Das D. 2012. Colour Image Steganography Based on Pixel Value Differencing in Spatial Domain. International Journal of Information Sciences and Techniques (IJIST). VOL. 2. No. 4. 83 – 93.

[17] Rawat D. and Bhandari V. 2013. Steganography Technique for Hiding Text Information in Color Image using Improved LSB Method. International Journal of Computer Applications (IJCA). VOL. 67. No. 1. 22 – 25.

[18] Rawat D. and Bhandari V. 2013. A Steganography Technique for Hiding Image in an Image using LSB Method for 24 Bit Color Image. International Journal of Computer Applications (IJCA). VOL. 64. No. 20. 15 – 19.

[19] Yadav M., Yadav A. and Yadav P. 2014. Analysis of Various Image Steganography Techniques. International Journal for Research in Applied Science and Engineering Technology (IJRASET). VOL. 2. Issue 2. 54 – 59.

[20] Mishra M. Routray A. R. and Kumar S. 2012. High Security Image Steganography with Modified Arnold's Cat Map. International Journal of Computer Applications (IJCA). VOL. 37. No. 9. 16 – 20.

[21] Li L. Luo B. Li Q. and Fang X. 2009. A Color Images Steganography Method by Multiple Embedding Strategy Based on Sobel Operator. International Conference on Multimedia Information Networking and Security (MINES '09). IEEE. VOL. 2. 118 – 121. (18 – 20 November). Hubei.

[22] Chawla G. Kamaldeep. Yadav R. and Ravi S. 2012. Analysis of Various Image Steganography Techniques on the Basis of Normalized Cross - Correlation (NCC). International Journal of Advanced and Innovative Research (IJAIR). VOL. 1. Issue 2.

[23] Juneja M. and Sandhu P. S. 2013. An Improved LSB based Steganography Technique for RGB Color Images. Second International Conference on Latest Computational Technologies (ICLCT-2013). 10 – 14, (June 17 – 18). London. UK.