Multi-Edge Concept used for Image Steganography

Author:

Rasber Dh. Rashid[1]
Ladeh S. Abdulrahman[2]
Taban F. Majeed[3]

SCIENTIFIC RESEARCH

ABSTRACT

Digital Steganography means hiding sensitive data inside a cover object in a way that is invisible to un-authorized persons. Many proposed steganography techniques in spatial domain may achieve high invisibility requirement but sacrifice the good robustness against attacks. In some cases, we need to take in account not just the invisibility but also we need to think about other requirement which is the robustness of recovering the embedded secret messages. In this paper we propose a new steganoraphic scheme that aims to achieve the robustness even the stego image attacked by steganalyzers. Furthermore, we proposed a scheme which is more robust against JPEG compression attack compared with other traditional steganography schemes.

[1] Software Engineering Department, Faculty of Engineering, Koya University, Iraq , rasber.rashid@koyauniversity.org .
[2] Software Engineering Department, Faculty of Engineering, Koya University, Iraq , ladeh.sardar@koyauniversity.org .
[3] Software Engineering Department, Faculty of Engineering, Koya University, Iraq, taban.majeed@koyauniversity.org .

INTRODUCCTION

Steganographic systems usually use one of the digital multimedia objects such as images, audio or video as a cover to embed secrete message in it used for communication. The high redundancies include in digital images with respect to human perceptibility make them usable morethan other cover objects when trying to embed message imperceptibly [1]. Embedding a secret message into the cover image create what is called a stego-image. The stego-image should be free from any detectable artifacts that resulted from message embedding. Such artifacts, if present, would hint to a third party (Attacker) that a secret message is present; an event that would bring the whole steganographic tool into failure [2] [3]. For any steganographic system, there are three main requirements that need to take in account: invisibility, capacity, and robustness [4]. Among them, the robustness is the main challenge objective [5] [6]. There are several forms of attacks on hidden messages: detecting, extracting, and disabling or destroying hidden information. The main aim of this work is to create and design an embedding system that possesses a higher security profile and is more resistant to several types of steganalysis and attacks. To achieve this, we first looked at finding the best locations for embedding in the cover image which is less affected by attacks. The results obtained, so far, are promising in terms of defending against JPEG compression attacks when compared to previous schemes.

LSB based Steganography schemes
In this section, we will analyze some techniques that are widely used and available in the spatial domain by discussing the strength and weakness of each technique. This will enable us to design and implement the new method which is more robust for some types of attacks especially for JPEG compression attack. Techniques used here for analysis are based on Least Significant Bit replacement, gray images are used as a cover, and the secrete message that needs to be embedded is face gray images.

Least Significant Bit embedding
Replacing the Least Significant Bits (LSB) of the image pixels with the secrete message bits is one of the simplest way of embeding data [7] [8]. By doing this the image pixel values will change by 1 only or pixel value remain unchanged in case of matching betwen secrete bit and the LSB of the pixel is happen. For gray images, there are 256 intinsity values, changing the values by 1 will make small change and human eye cannot detect the changes [9] [10]. All or part of the image can be used for embedding purpose. Although, embedding in all pixel results as a high capacity but it is not secure as an attacker can simply repeat the process to quickly recover the hidden information [11]. The capacity even can be increased to twise or three times by using two or three bits for embedding. In this case, the quality of the image may be affected and the technique may lose its main requirement which is invisibility [12] [13] see figure 1. To increase the robustness of embedding, the pixels that are used for embedding can be selected randomly over the hole or part of the image, this is so called Random LSB(RLSB). The noise introduced by RLSB is randomly placed and often causes the resulting stego-image to look speckled [14] [15].

Figure 1. Affect of using more than one-bit plane for embedding.

Edge Least Significant Bit (ELSB) Embedding
Edge in gray image can be defined as a significant dissimilarity used as a boundary between two regions in an image fragment [13]. Embedding in such regions, will conclude new scheme of embedding known as edge-based steganography [16] [17]. The idea behind using edges for embedding is that the edges can carry more screte messages than smooth área [16]. Attackers has less suspicions of the presence of message bits in edges therefore embedding in edge area is of more strength than the traditional LSB. There are several types of edge operators like Sobel, Prewitt, Laplacian and Canny operators. Among them, canny edge detector is considered the most rigorously defined operator and is widely used in steganography schemes. The wide usage of the canny edge detector can be attributed to three criteria of good detection, good localization, and single response to an edge [13].

The main issue of using edge for embedding is that a pixel which is detected as an edge position in original cover before embedding may not be detected as an edge position in stego-cover after embedding [14]. Another drawback of using edges for embedding is the capacity if compared with the SLSB and RLSB because number of edges are limit inside any image.

Any edge detection method uses a threshold to decide which pixel is edge and which is not. Based on value of threshold the numbers of edges are increased or decreased. When the threshold is high then fewer numbers of edges are detected and when threshold is low then higher numbers of edges are detected, see figure 2.

In the literature there are several techniques deal with the problem of the edge positions which they may change before and after embedding. One of the solutions is proposed in [13], the algorithm is done by blocking the image into non-overlapping raster block which the first pixel in each block contain the information about the rest pixels in the block, explain that the next pixels are edge or not. The proposed algorithm is also embedding less bits of secret message in the non edge pixels while embed large number of bits in the edge pixels. FilterFirst is another solution for this problem that was proposed in [12], authors claim that because the embedding only change x least significant bits, then remaining y most significant bits can be used for the filter and finding edges. This can guarantee the same pixels that were used for embedding are used for extraction because the y bits that were used for filter remain the same and the receiver can do the same filtering to get the same pixel positions that were used in the embedding process. The weakness of FilterFirst is that it is not a secure algorithm. An attacker can repeat the filtering process and retrieve the hidden information with very little effort [18].
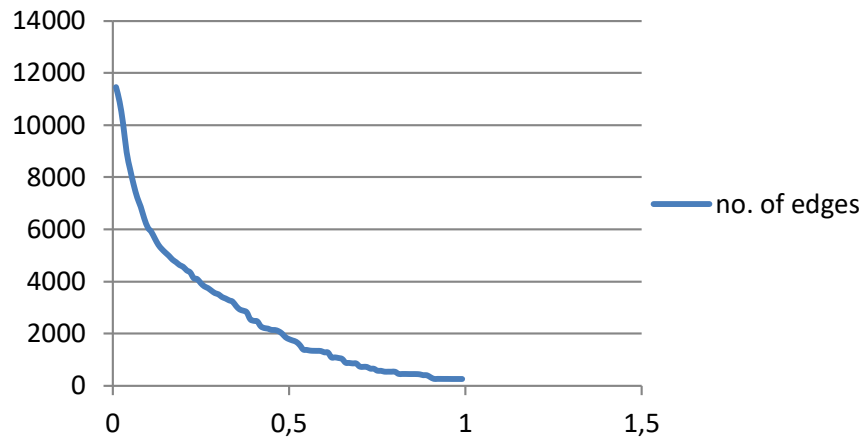
Relation Between Threshold and No. of Edges

Figure 2. Relation between number of edge and the threshold.

Proposed Scheme

Most of the existing methods that were described above, there are some principles and concepts that were not taken into account when dealing with embedding subject. Here we explain some of them which we used as a base for our work.

- Some information is more important than the others which they need to be embedded, for this reason we need more secure locations to embed the more important information. For example, we are interested in hiding face image, this face image is an 8-bit gray image. The first 4-bits of each pixel (4 MSBs) are more important than the other 4-bits (4 LSBs), then we need to embed the 4 MSBs in more secure locations than the other 4 LSBs.
- When we are using edges location for embedding we need to take into account that some edges are more significant than others, which they remain as an edge location after doing some image processing attacks like JPEG compression attacks. In our case we benefit from this property of the edge by embedding the most important information in most significant edge locations and the other less important information in the least significant edges.

- When the embedding capacity is low we can use only sharper edge regions for embedding and keep the other edges as they are, and when the capacity of embedding is increased then more edges can be selected for embedding, although this increasing will affect the invisibility but still in acceptance rate.
- By using traditional edge embedding techniques, we embed the secret message sequentially in the edge locations; however, when we apply two-, four-, and eight edge detection algorithm on the same image with relevant threshold then we guarantee that the secret message is not embedded in the edges sequentially.

In our proposed method, 8-bit gray image was used as a cover image and the face image as a secret message which is also 8-bit gray image. Like any other steganography systems, our proposed method consists of two parts, embedding and extraction.

Embedding process

Our embedding method using 8 different edge locations is shown as a flow chart in figure 3, and we explain the flow chart by the following steps:

Step 1: Read the 8-bit gray secret image (Face) and

convert the secret image to the 8 bit planes. Figure 5 shows part of the image planes.

Step 2: Read cover image and by using canny edge detector calculate the different (8, 4, or 2) edge locations of the original cover image by using different (8, 4, or 2) threshold parameters. Selecting the threshold is dependent on the number of edge positions that are needed for embeding.

Step 3: Embed each plane to the corresponding edge locations in the original cover image by replacing the Least Significant Bit(s) of the cover image:

A- In 8 different edge method, each plane is embedded to the corresponding edge location by replacing only 1 bit of the cover.

B- In 4 different edge method, every 2 plane are embedded to the corresponding edge location by replacing 2 bits of the cover.

C- In 2 different edge method, every 4 plane is embedded to the corresponding edge location by replacing 4 bits of the cover.

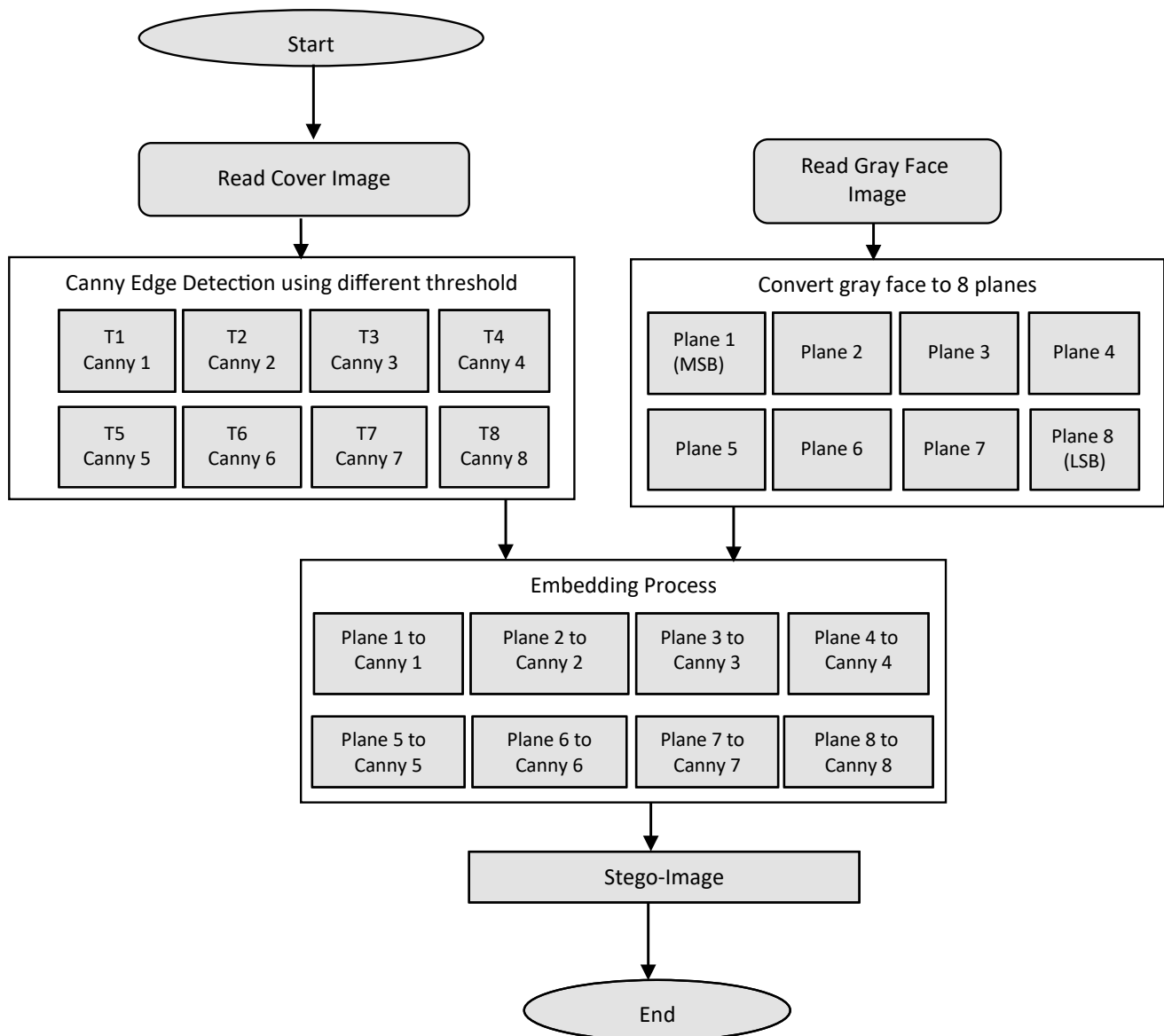**Step 4:** The output well be the Stego-Image.

Figure 3. Embedding process of proposed method

Extraction process

At the receiver side, the extraction is needed. For extraction only the threshold (8, 4, or 2) values are needed. The flow chart below (figure 4) shows the extraction method and the process is explained by the following steps:

Step 1: Read the stego-image.

Step2: Using Canny edge detector find the same (8, 4, or 2) edge locations of the stego-cover image by using same (8, 4, or 2) threshold parameters that were used in the embedding process.

Step 3: Extract each plane from the corresponding edge locations in the stego-image by reading the Least Significant Bit(s) of the stego-image:

A- In 8 different edge method each plane is extracted from the corresponding edge locations by reading only 1 bit of the stego-image.

B- In 4 different edge method each plane is extracted from the corresponding edge locations by reading 2 bits of the stego-cover.

C- In 2 different edge method each plane is extracted from the corresponding edge locations by reading 4 bits of the stego-cover.

Step 4: Collect all the planes that were recovered from step 3 to get the recovered face image.
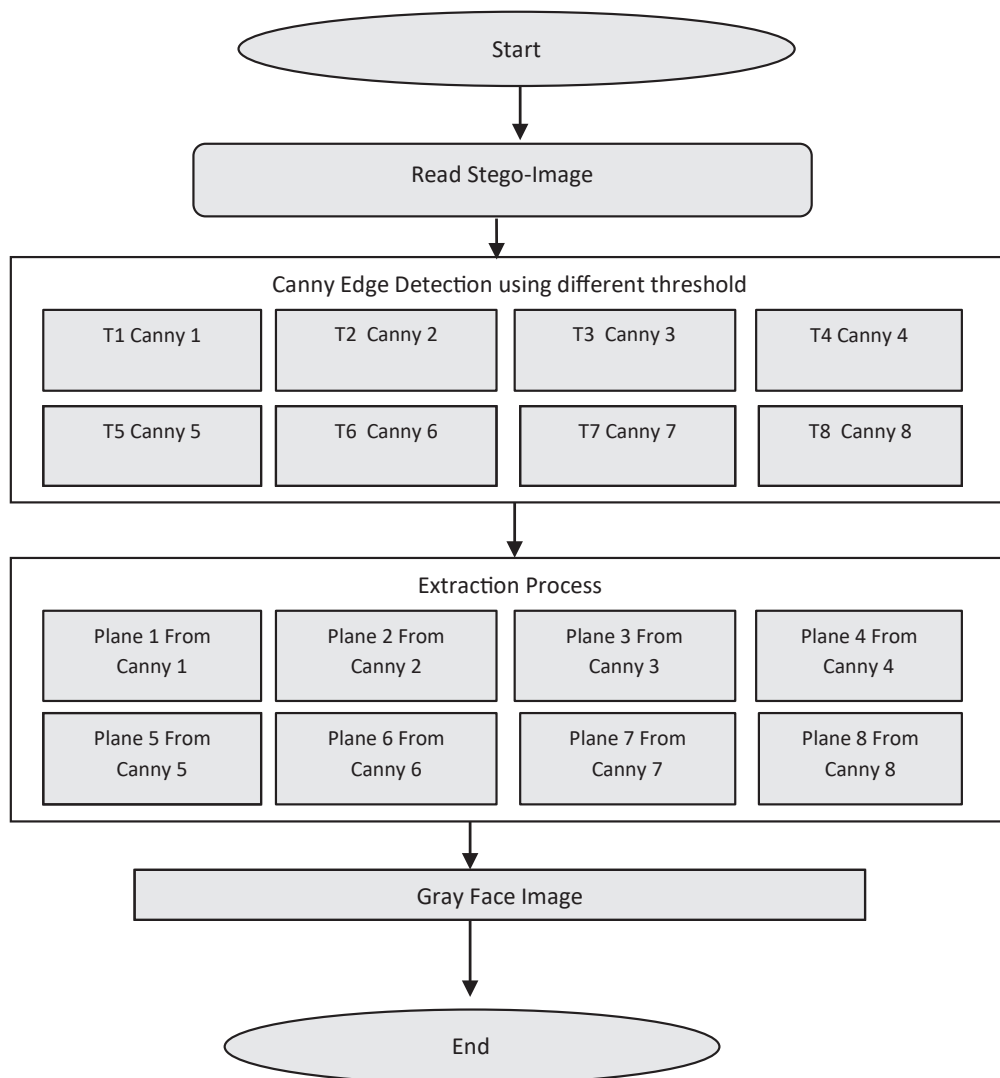

Figure 5. Extraction process of proposed method
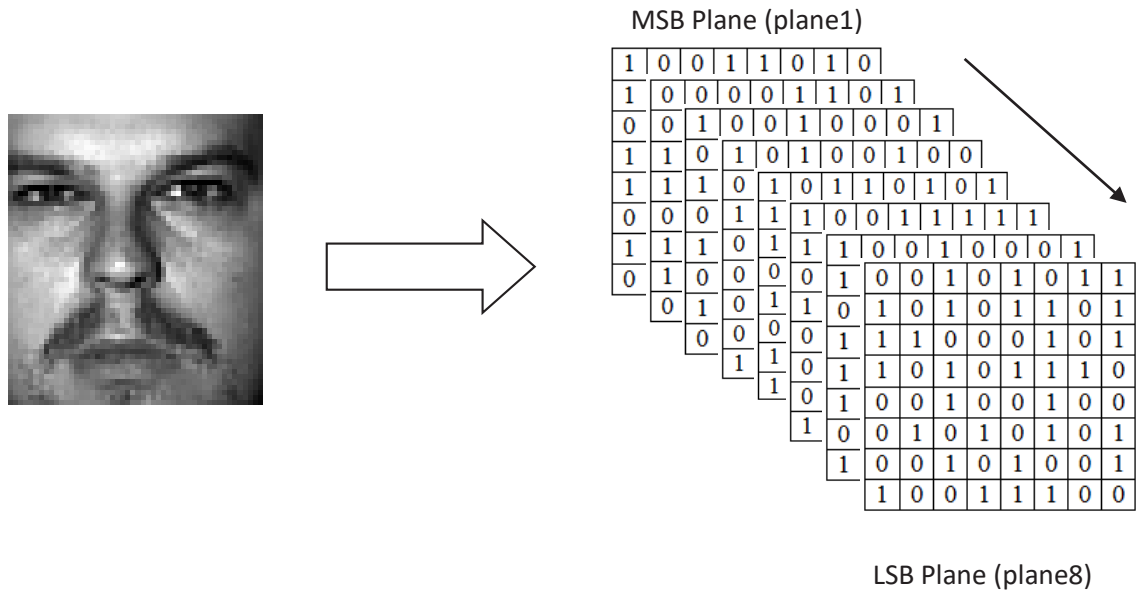
MSB Plane (plane1)

LSB Plane (plane8)

Figure 6. Image Bit planes

Experimental Results

For the purpose of comparision we implemented and tested number of techniques which they are Sequential Least Significant Bits (SLSB), Random Least Significant Bits (RLSB) , Edge Least Significant Bits (ELSB), Filter First Least Significant Bits (FFLSB), and our proposed method in three forms Two Edge Least Significant Bits (2ELSB), Four Edge Least Significant Bits (4ELSB), and Eight Edge Significant Bits (8ELSB). Comparison among all techniques is conducted in the direction of robustness to a specific type of attack which is JPEG compression attack with some degree of the JPEG compression from quality 100 to 90 which can be represented as compression ratios between 2.29 and 5.56 as well. Experiments are conducted on the five different gray images with size (512x512) shown in figure 6 used as a cover and five different face images with size (17x17) that were taken from CroppedYele database as a secret image that is shown in figure 7.
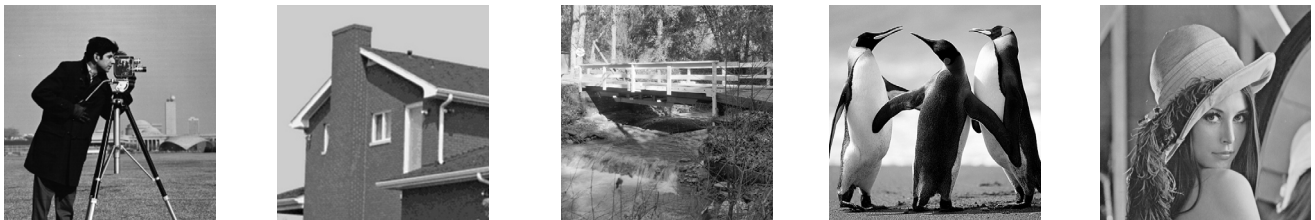

Figure 7. Cover-Images used

Figure 8. Embedded Face-Images

We measure the invisibility by using Peak Signal to Noise Ratio (PSNR) between the cover-image and stego-image before and after JPEG compression attacks to see how JPEG affect the invisibility of each method. Table 1 shows the results. Results show that all the methods were approximately affected to the same extent when the stego-image compressed by JPEG compression. In the case of no attack, the value of PSNR appears to be relatively the same in several cases instead of the case of 4ELSB and 2ELSB. This is because we changed 2 or 4 LSBs of each pixel positions that were used for embedding.

This makes more differences between the cover and the stego-cover but still the values remain in a good range of invisibility.

$$PSNR = X*Y* (max P(x,y))^2 / \sum (P(x,y) - P'(x,y))^2$$

Where X,Y is the width and length of the image, $P(x,y)$ and $P'(x,y)$ represents the pixel with row number x and column number y in the original and stego-object, respectively. PSNR is measured in decibels (db), such that:

$$PSNRindb = 10 *log10 (PSNR)$$

Table 1. PSNR between Cover-Images and Stego-Images

Methods	SLSB	RLSB	ELSB	FFLSB	8ELSB	4ELSB	2ELSB
No Attack	71.7777	71.6829	71.6121	66.7798	71.5389	67.7708	58.4366
JPG 100	59.0263	59.0251	59.0347	59.0342	59.0249	58.6902	55.8154
JPG 99	57.2065	57.2308	57.2389	57.2498	57.2375	56.9848	54.8201
JPG 98	53.6488	53.664	53.6442	53.6519	53.6524	53.5679	52.3882
JPG 97	51.5765	51.5741	51.5722	51.5735	51.5686	51.5212	50.7465
JPG 96	50.2645	50.267	50.2575	50.2651	50.2601	50.2249	49.6689
JPG 95	49.1962	49.1878	49.1885	49.1874	49.1903	49.1689	48.7648
JPG 94	48.4358	48.4354	48.4311	48.4354	48.433	48.4152	48.0936
JPG 93	47.7621	47.7607	47.7581	47.7553	47.7599	47.7452	47.482
JPG 92	47.1145	47.1131	47.1111	47.1104	47.1092	47.0989	46.8737
JPG 91	46.6563	46.6572	46.6561	46.6557	46.6527	46.6444	46.4474
JPG 90	46.1749	46.173	46.1723	46.1714	46.1711	46.1598	46.0053

Then in the receiver side we extracted the logo in attacked stego-image. We measure the differences between original face and recovered face in each method by PSNR, table 2 and figure 8 shows the PSNR between original and recovered face images. From the results we can see that our proposed algorithm in the case of 2ELSB has better results in all cases when compared with other algorithms and is more robust to the JPEG compression. Figure 10 shows the recovered face image after attacks. In this figure, if you look for example to the images in the SLSB and RLSB methods, we can only distinguish a face in

the cases of JPEG with quality 100 and 99 degrees; after these degrees we cannot distinguish anything. In case of ELSB there are some damages even without attacks. In case of using FFLSB, the embedded image is damaged, only in the case of no attack we can decide there is a face image after that we cannot. But if we look to our proposed method especially in the case of 2ELSB the embedded image remains until some level of applying JPEG attacks.

Table 2. PSNR between original and recovered face images

Test	SLSB	RLSB	ELSB	FFLSB	8ELSB	4ELSB	2ELSB
JPG 100	15.0518	15.7413	8.8779	9.5917	14.956	18.0659	19.632
JPG 99	12.3344	11.6058	10.5645	9.2946	11.802	14.3062	18.6257
JPG 98	10.1941	10.0702	8.6803	8.6078	9.2313	12.0031	14.7513
JPG 97	9.4229	9.306	8.5504	8.2976	8.4965	10.0486	14.8496
JPG 96	9.1376	8.9487	8.728	8.3547	8.9293	9.124	13.361
JPG 95	9.24	8.8583	8.4077	8.163	8.5023	9.2231	12.0757
JPG 94	8.5997	8.2273	8.5195	8.2279	8.9061	8.5828	11.8261
JPG 93	8.5638	8.8099	8.2741	8.1628	8.4047	8.3727	11.4375
JPG 92	8.5295	8.3646	8.8054	8.2301	8.4159	8.2259	11.2018
JPG 91	8.8592	8.0349	8.27	8.293	8.1621	8.4464	11.9281
JPG 90	8.4822	8.3195	9.0017	8.8993	8.0083	8.8921	12.2606

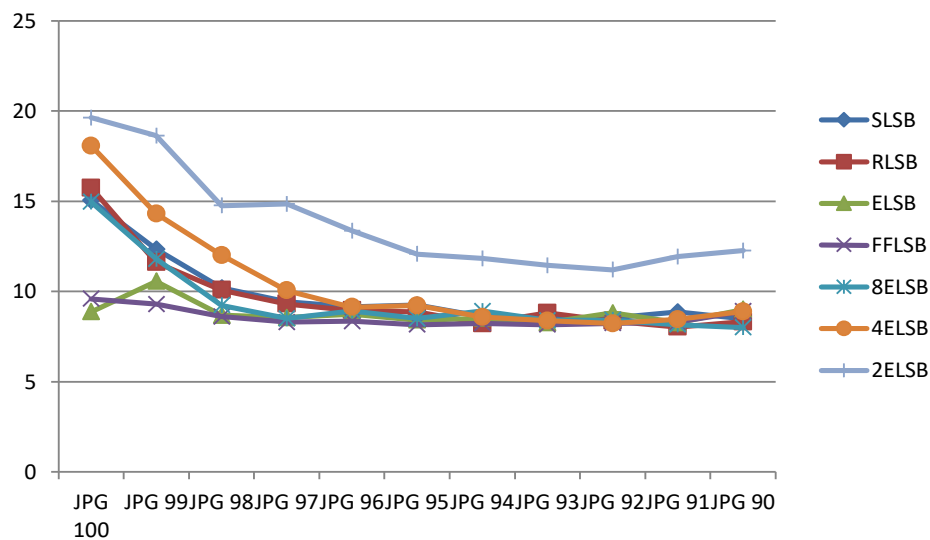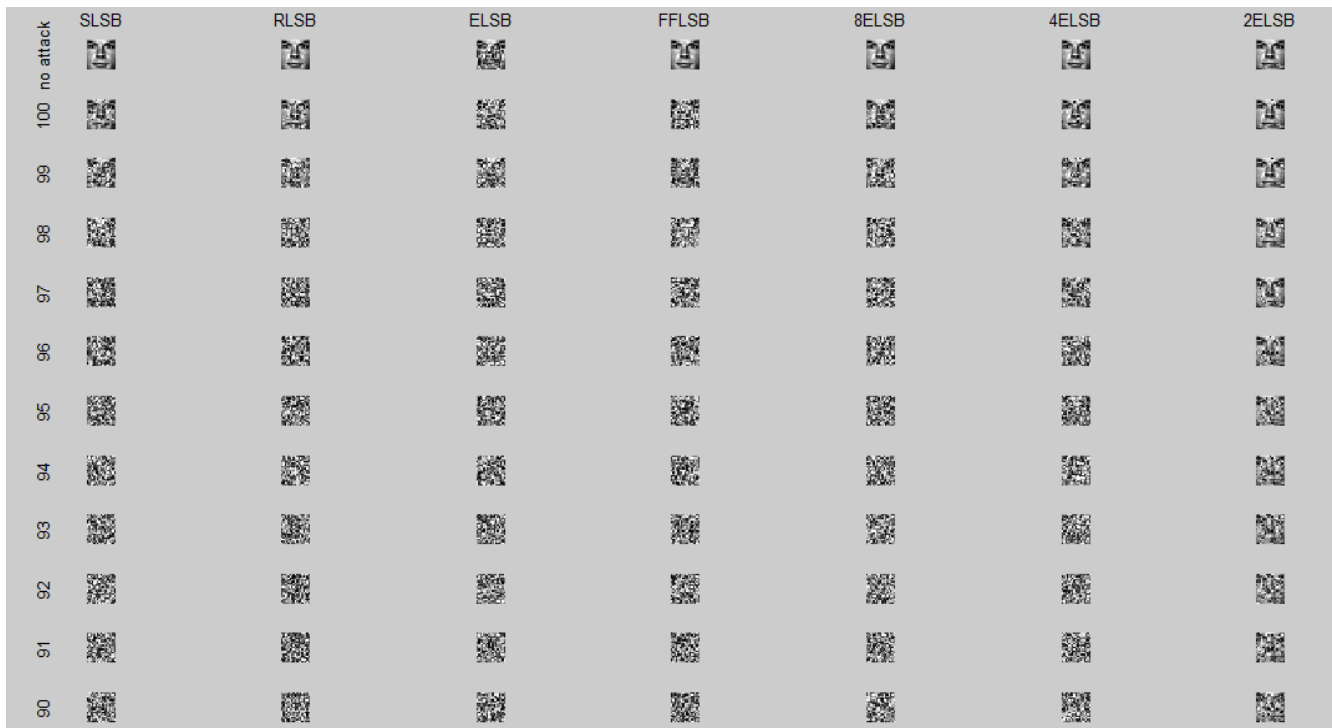**Figure 9.** PSNR between original and recovered face images

Figure 10. Recovered face images after attacks

CONCLUSIONS

Steganography is the process of embedding secret message into the cover image, as a result the process will create what is called a stego-image. One of the main properties of embedding scheme is that the stego-image should be free from any detectable artifacts. In this paper we implemented some methods in image steganogrphy that they are all based on using LSB and compared with our proposed methods. In order to decide which pixel has the high spatial frequency or the one with the low spatial frequency in a digital image, the edge detection algorithm is generally used. The size of the message to be hidden is the guide to the number of edges we need.

In order to find these edges, a Threshold Parameter is used. For a specific number of edges needed the threshold value is different from one image to another. This makes our proposed method more secure because this threshold value is used as a key and shared between the sender and the receiver, also the value of the threshold changes when the cover image is changed. We choose different thresholds for finding the edge locations. We reserve higher thresholds for the bit planes which represent MSBs since they are more important than the others for recovering. The original cover image is not needed to recover the secret embedded face image in the extraction algorithm. Therefore, results show our shemes robustness compared with the previous available embedding schemes.

REFERENCES

1. Agrawal N, Savvides M. Biometric data hiding: A 3 factor authentication approach to verify identity with a single image using steganography, encryption and matching. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops; 2009 Jun 20-25; Miami, FL, USA. IEEE; 2009. p. 85–92. Available from: https://ieeexplore.ieee.org/document/5204308/ https://doi.org/10.1109/CVPR.2009.5204308

2. Fridrich J, Long M. Steganalysis of LSB encoding in color images. In: 2000 IEEE International Conference on Multimedia and Expo ICME2000 Proceedings Latest Advances in the Fast Changing World of Multimedia (Cat No00TH8532);2000 Jul 30 - Aug 2; New York, NY, USA. IEEE; 2000. p. 1279–82. Available from: http://ieeexplore.ieee.org/document/871000/

3. Fridrich J, Goljan M, Du R. Reliable detection of LSB steganography in color and grayscale images. In: Proceedings of the 2001 workshop on Multimedia and security new challenges - MM&Sec '01 [Internet]; 2001 Oct 5; Ottawa, Ontario, Canada; New York, New York, USA: ACM Press; 2001. p. 27. Available from: http://portal.acm.org/citation.cfm?doid=1232454.1232466 https://doi.org/10.1145/1232454.1232466

4. Rashid RD, Sellahewa H, Jassim SA. Biometric feature embedding using robust steganography technique. In: Proceedings of SPIE - The International Society for Optical Engineering; Baltimore/Maryland/USA; 2013. https://doi.org/10.1117/12.2018910

5. Xu J, Sung AH, Shi P, Liu Q. JPEG compression immune steganography using wavelet transform. In: International Conference on Information Technology: Coding and Computing, 2004 Proceedings ITCC 2004; 2004 Apr 5-7; Las Vegas, NV, USA, USA;2004. p. 704-708 Vol.2. Available from: http://ieeexplore.ieee.org/document/1286737/ https://doi.org/10.1109/ITCC.2004.1286737

6. Rashid RD, Asaad A, Jassim S. Topological data analysis as image steganalysis technique. In: Proceedings of SPIE - The International Society for Optical Engineering; 2018 Apr 15-19; Orlando, Florida, United States; 2018.

7. Johnson NF, Jajodia S. Exploring steganography: Seeing the unseen. Computer (Long Beach Calif) [Internet]. 1998;31(2):26–34. Available from: http://ieeexplore.ieee.org/document/4655281/ https://doi.org/10.1109/MC.1998.4655281

8. Chan C-K, Cheng LM. Hiding data in images by simple LSB substitution. Pattern Recognition. 2004;37(3):459-474. Available from: https://www.sciencedirect.com/science/article/abs/pii/S003132030300284X https://doi.org/10.1016/j.patcog.2003.08.007

9. Morkel T, Eloff J, Olivier M. An overview of image steganography [Internet]. Available from: https://pdfs.semanticscholar.org/bb26/1e7f02f-8597b37a2f71e55c2e2c21aa7575f.pdf

10. Rashid RD, Jassim SA, Sellahewa H. Covert exchange of face biometric data using steganography. In: 2013 5th Computer Science and Electronic Engineering Conference (CEEC);2013 Sep 17-18; Colchester, UK;2013. IEEE; p. 134–9. Available from: http://ieeexplore.ieee.org/document/6659460/ https://doi.org/10.1109/CEEC.2013.6659460

11. Hempstalk K. Hiding Behind Corners: Using Edges in Images for Better Steganography. Available from: https://www.researchgate.net/publication/241605558

12. Bailey K, Curran K. An evaluation of image based steganography methods. Multimed Tools Applications. 2006;30(1):55–88. Available from: http://link.springer.com/10.1007/s11042-006-0008-4 https://doi.org/10.1007/s11042-006-0008-4

13. Chen W-J, Chang C-C, Le THN. High payload steganography mechanism using hybrid edge detector. Expert Syst Appl, 2010;37(4):3292–301. Available from: https://www.sciencedirect.com/science/article/pii/S0957417409008318 https://doi.org/10.1016/j.eswa.2009.09.050

14. Singh KM, Singh LS, Singh AB, Devi KS. Hiding Secret Message in Edges of the Image. In: 2007 International Conference on Information and Communication Technology;2007 Mar 7-9; Dhaka, Bangladesh; IEEE;2007. p. 238–41. Available from: http://ieeexplore.ieee.org/document/4261407/
https://doi.org/10.1109/ICICT.2007.375384

15. Rashid R. Robust Steganographic Techniques for Secure Biometric-based Remote Authentication, 2016. Available from: https://www.researchgate.net/publication/295549501

16. Yang C-H, Weng C-Y, Wang S-J, Sun H-M. Adaptive Data Hiding in Edge Areas of Images With Spatial LSB Domain Systems. IEEE Trans Inf Forensics Secur. 2008;3(3):488–97. Available from: http://ieeexplore.ieee.org/document/4598830/
https://doi.org/10.1109/TIFS.2008.926097

17. H-C, Wu N-I, Tsai C-S, Hwang M-S. Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEE Proceedings - Vision, Image and Signal Processing; 2007;152(5): 611–615. Available from: https://digital-library.theiet.org/content/journals/10.1049/ip-vis_20059022
https://doi.org/10.1049/ip-vis:20059022

18. Unnikrishnan R. Analysis of Modern Steganographic Techniques. 2011. Available from: http://www.bvicam.ac.in/news/INDIACom 2011/9.pdf