Mühendislik Bilimleri ve Tasarım Dergisi
8(2), 357 – 365, 2020
e-ISSN: 1308-6693

Journal of Engineering Sciences and Design
DOI: 10.21923/jesd.537183

Araştırma Makalesi | Research Article

A NEW IMPERCEPTIBLE STEGANOGRAPHY METHOD FOR GRAYSCALE IMAGES

Hüseyin Bilal MACİT[1*], Arif KOYUN[2]

[1] Mehmet Akif Ersoy University, Bucak Z.T. School of Applied Technology and Business, Department of IT Systems and Technologies, Burdur, Turkey
[2] Suleyman Demirel University, Engineering Faculty, Department of Computer Engineering, Isparta, Türkiye

Keywords

Steganography, LSB, Data Hiding, Image Steganography, Block Based Steganography, SSIM.

Abstract

A standard Least Significant Bit (LSB) algorithm changes average 50% of the least significant bits of an image. This causes approximately a 0.2% distortion on a grayscale image. Therefore, the application of LSB on non-sharp grayscale images causes too much noise that can be easily detected by the human visual system. In addition; when the image is scanned pixel by pixel, hidden message can be easily read by unauthorized persons. A robust and imperceptible block-based LSB steganography method is developed in this study. First, cover image is divided into blocks, then median geometric mean and arithmetic mean of each block is calculated to select blocks those contain less detail of the image. Method has an advantage of calculating data load capacity of image before embedding process. Proposed method also makes it difficult for unauthorized people to detect hidden messages. Performance of the method is measured by mathematical methods such as MSE, PSNR and SSIM. The method achieved structured similarity score over 0.9999% on four test images, indicating a 20-fold better result than the standard LSB algorithm.

GRİ TONLAMALI GÖRÜNTÜLER İÇİN YENİ BİR ALGILANAMAZ STEGANOGRAFİ YÖNTEMİ

Anahtar Kelimeler

Steganografi, EÖB, Veri Gizleme, Görüntü Steganografi, Blok Tabanlı Steganografi, SSIM.

Öz

Standart bir En Önemsiz Bit (EÖB) algoritması, bir görüntünün en az önemli bitlerinin ortalama %50' sini değiştirir. Bu durum gri tonlamalı bir görüntüde yaklaşık olarak %0,2 bozulmaya neden olur. Bu nedenle, keskin olmayan gri tonlamalı görüntülerde EÖB uygulaması, insan görsel sistemi tarafından kolayca tespit edilebilecek çok fazla gürültüye neden olur. Ek olarak; görüntü piksel piksel tarandığında, gizli mesaj yetkisiz kişilerce kolayca okunabilir. Bu çalışmada sağlam ve algılanamayan blok tabanlı EÖB steganografi yöntemi geliştirilmiştir. İlk önce kapak resmi bloklara bölünür, sonra görüntünün daha az ayrıntılarını içeren blokları tespit etmek için her bir bloğun medyan geometrik ortalaması ve aritmetik ortalaması hesaplanır. Yöntemin, veri gizleme işleminden önce görüntünün veri yükü kapasitesini hesaplayabilme avantajı vardır. Geliştirilen yöntem ayrıca yetkisiz kişilerin gizli mesajları tespit etmelerini zorlaştırır. Yöntemin performansı MSE, PSNR ve SSIM gibi matematiksel yöntemlerle ölçülmüştür. Yöntem, dört test görüntüsünde %0,9999' un üzerinde yapısal benzerlik skoru elde etmiştir ve standart EÖB algoritmasından 20 kat daha iyi sonuç alındığı gösterilmiştir.

Alıntı / Cite

Macit, H.B., Koyun, A., (2020). A New Imperceptible Steganography Method for Grayscale Images, Journal of Engineering Sciences and Design, 8(2), 357-365.

Yazar Kimliği / Author ID (ORCID Number)

H.B. Macit, 0000-0002-5325-5416
A. Koyun, 0000-0001-6701-363X

Makale Süreci / Article Process	
Başvuru Tarihi / Submission Date	08.03.2019
Revizyon Tarihi / Revision Date	18.12.2019
Kabul Tarihi / Accepted Date	02.06.2020
Yayım Tarihi / Published Date	25.06.2020

[*] İlgili yazar / Corresponding author: hbmacit@mehmetakif.edu.tr, +90-248-491-2450

1. Introduction

Data hiding methods, which date back to ancient times, appear in different forms in the 21st century digital environment (Yalman and Ertürk, 2009). The concept of 'What you see is what you get' is now not completely true with the growing digital World even an ordinary data could be perceived as a suspicious data (Cheddad et al., 2010). One of the methods of data hiding is steganography which is hiding a message in a cover media so as not to be noticed. In simple terms, steganography means "storing one data into another".

The name Steganography was put by Johannes Trithemus who lived between years 1462 and 1516. The word "Steganographia" is composed of Greek words "στεγανό-ς" which means secret and "γραφ-ειν" which means writing (Por and Delina, 2008). Steganography shows similaritiy with cryptography. Both deal with the security of the data. Steganography and cryptology create an effective data-hiding method when used together. However, the most important difference between steganography and cryptography is the doubt factor (Brainos, 2004). All data in cryptography are encrypted and clear. But in steganography, all ordinary data passing through the communication line may be suspicious.

The most common use of steganography has been in the military field. One of the best known examples is a message carried by a Nazi spy in World War II. The message is: "Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils". When the second letters of the words in this insignificant like message are written side by side, the secret message appears: "Pershing sails from NY June 1" (Johnson and Jajodia, 1998). Here, a hidden text is sent by hiding in another text which is called the cover media. Cover media and hidden messages can be a text, picture, sound, video or another data packet (Chandramouli and Memon, 2001).

Large distortions can be made to a digital media to hide other information into it. However, the major changes in the original media will be noticeable by the human visual system (HVS) or the human auditory system (HAS). HAS is more sensitive to detect small changes in objects than the HVS. For example; HAS is able to detect small changes in audio frequencies below 2 kHz. However, this does not apply to the HVS. Thus, the secure data hiding method used to protect an audio data may not be applicable to an image data. It is inevitable to use different methods for different types of media (Chore and Tiwari, 2017).

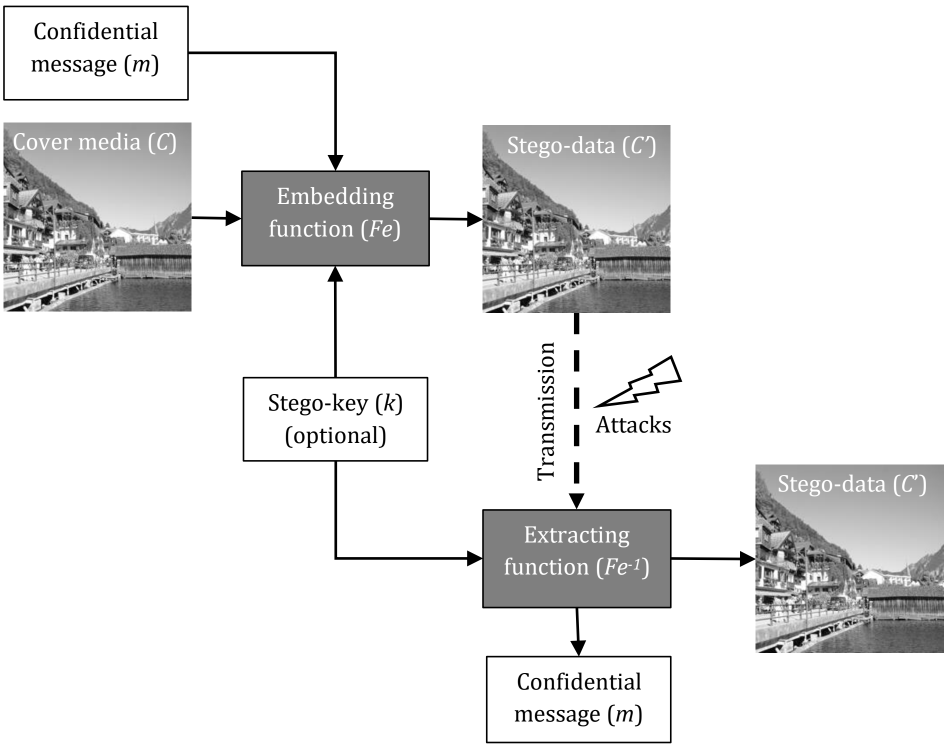


Figure 1. A standart steganographic operation model

A typical steganographic operation is performed in figure 1. As it is seen in figure 1, hidden message m is embedded in the cover media C using an embedding function Fe in a steganography method. Here, the stego-key k may represent a crypto-key or it may be a set of information that stores how the message is hidden or going to hide.

Every steganography method does not have to use the Stego-key. However, if stego-key is used in the embedding phase, the extracting function Fe^{-1} cannot extract the secret message m without this key.

The performance of a steganographic system can be evaluated on various features. The most important of these is imperceptibility. Imperceptibility expresses how well the hidden data is camouflaged. Another measure is the steganographic data load. Data load refers to quantity of data which can be embedded in the cover data. As the hidden data size decreases and the cover data size increases, the success of the steganographic method will also increase. Studies on steganography have focused on the development of algorithms that generally embed larger sized hidden data within smaller sized cover data successfully. Furthermore, as the number of data transferred in a transmission medium increases, the possibility of stego data detection will decrease in the same proportion (Eggers et al., 2002). Other performance criteria is robustness which refers to the resistance of stego-data to attacks by unauthorized persons (Şatır, 2013). These three features are expressed in a triangle which is called the magic triangle shown in figure 2. As it is seen, if a method gets closer to one feature, it gets farther to other features. Hence, recommendation of the method must be clearly preferred.

If the detectability of confidential data is zero, it can be said that this steganography method is perfectly safe. Any image steganography method should be optimized to achieve the minimum detectability possible, taking into account the data load. (Yu et al., 2005).

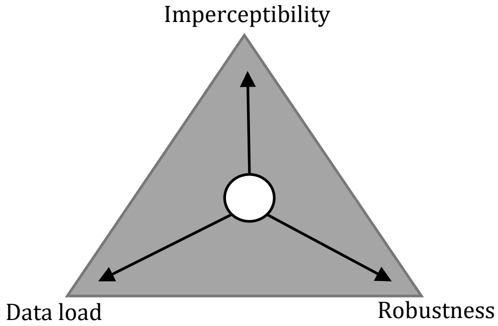


Figure 2. Magic triangle of data hiding (Hamza, 2008).

An image file is actually a matrix of pixels. Each pixel can be expressed mathematically (Hamid et al., 2012). Image steganography methods are examined in two categories as spatial domain and transform domain methods. A spatial domain method embeds the confidential message in the pixel space. The transform domain methods embed the confidential message in the frequency domain. This study proposes a new image steganography method using the LSB technique, which is the most common method of spatial domain methods. The proposed method is designed for grayscale images and tested for performance.

2. Methodology

The standard LSB algorithm is a form of image processing, like signal processing. The input data of this study is an image and the output data is the hidden message generated from the image data. The image is treated as a matrix of pixels. Each element of the matrix refers to a pixel of the image.

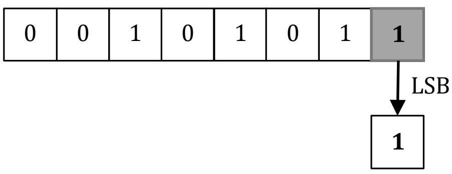


Figure 3. LSB value of an example pixel which is 43 in decimal (Luo et al., 2010).

In a grayscale image, each pixel is encoded with 8 bits which is equal to 1 byte as it is shown in figure 3. Pixels encoded with 1 byte each have a decimal value of 0 to 255. LSB algorithm changes the LSB bit of next pixel with next bit of the confidential message. This operation makes a change in the pixel value with a small ratio.

Let C is a grayscale cover media composed of C_1xC_2 pixels. C is represented as in equation (1).

$$C = \{x_{ij} | 0 \leq i < C_1, 0 \leq j < C_2, x_{ij} \in \{0,1,2,\ldots,255\}\} \qquad (1)$$

m is the confidential message composed of n bits in equation (2).

$$m = \{m_i | 0 \leq i < n\} \; m_i \in \{0,1\} \quad (2)$$

Suppose that n bits confidential message m is going to embed in k LSB bits of C. First, m must be denoted as m' which is composed of k bits as in equation (3) and equation (4).

$$m' = \{m'_i | 0 \leq i < n'\} \quad (3)$$

$$m'_i \in \{0,1,2 \ldots, 2^k - 1\} \; n' < m_c x m_c \quad (4)$$

Mapping n bits of confidential message $m=\{m_i\}$ and hidden message $m'=\{m'_i\}$ is shown in equation (5).

$$m'_i = \sum_{j=0}^{k-1} m_{i.k+j} . 2^{k-1-j} \quad (5)$$

n' pixels $\{x_1, x_2, \ldots, x_n\}$ are selected from cover image C. k LSB x_i of this sequence swaps with m'_i. Swapped x_i is expressed as x'_i deriven from m'_i as in equation (6).

$$x'_i = x_i - x_i mod 2^k + m'_i \quad (6)$$

To extract hidden message, the sequence $\{x'_1, x'_2, \ldots, x'_n\}$ of stego image in embedding process is followed. Equation (7) calculates an m' sequence which is created by k LSB bits.

$$m'_i = x'_i mod 2^k \quad (7)$$

Here, theoretically, the worst imperceptibility is obtained during embedding when all LSB series $\{x_1, x_2, \ldots, x_n\}$ are modified. Assuming that the entire series have not changed, the best imperceptibility is achieved. However, this possibility is not calculated in the standard LSB algorithm before embedding process.

3. Proposed Method

The LSB algorithm attempts to hide the message by scanning the entire image at once. The major difference of the proposed method is; it pre-calculates the distortion in the cover data to minimize the detectability of existence of the hidden message. C is a grayscale cover image composed of $C_1 x C_2$ pixels and is calculated with equation (8).

$$C = \{x_{ij} | 0 \leq i < C_1, 0 \leq j < C_2 \quad x_{ij} \in \{0,1,2, \ldots, 255\}\} \quad (8)$$

For the embedding process, proposed method divides cover image C into 32x32 equal sized blocks $B(i,j)$ as it is shown in figure 4. The pseudo code of this process is as follows:

```
for i=1→C1–((C1/32)–1) step C1/32
        for j=1 →C2–((C2/32)–1) step C2/32
                B(i,j)=C(i:i+((C1/32)–1),j:j+((C2/32)–1))
```

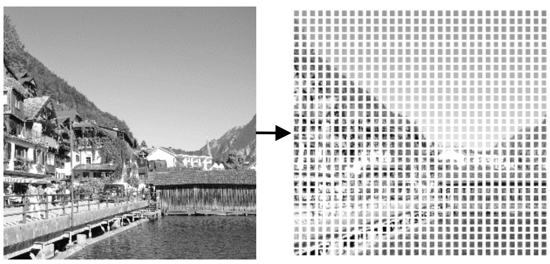


Figure 4. Cover image divided into 32x32 blocks

Method generates a copy of each original image blocks $B(i,j)$ to new blocks $B'(i,j)$ to calculate optimum data load before hiding the message. Then the mean of each block of the original image is stored in a O matrix which is in size of 32x32 cells as in equation (9).

$$O_{(i,j)} = \sum_{i=1}^{32} \sum_{j=1}^{32} B_{(i,j)} \quad (9)$$

For effective steganography that cannot be detected by HVS, blocks containing least image detail should be selected. For this purpose, blocks *O'(i, j)* are generated from *O(i, j)* blocks using geometric median filtering as in equation (10).

$$O'_{(i,j)} = \frac{1}{\mu - \varphi} \sum r_{(i,j)} \qquad (10)$$

In equation (10), *r(i,j)* represents remaining pixels after substracting the lowest and highest *φ/2* pixel value each step.

$$\varphi = \frac{\mu - 1}{2}$$
$$0 \leq \varphi \leq \mu - 1 \qquad (11)$$

Equation (11) calculates median mean of the block. Now; generated *O(i,j)* and *O'(i,j)* points out each block of cover image *C*. To select the blocks carrying minimum detail of image; the absolute value of the difference between the matrixes *O* and *O'* is calculated and stored into the *F* matrix in equation (12).

$$F_{(i,j)} = |O_{(i,j)} - O'_{(i,j)}| \qquad (12)$$

The *F* matrix briefly shows the difference of the median mean and arithmetic mean of each block. Equation (13) calculates the arithmetic mean *δ* of the *F* matrix which is the average difference of the median mean and arithmetic mean of the cover image *C* and generated image *C'*.

$$\delta = \frac{\sum_{i=1}^{32} \sum_{j=1}^{32} F_{(i,j)}}{32x32} \qquad (13)$$

δ shows the difference of *O* and *O'* blocks and it also shows the blocks which are selected to embed the message to gain imperceptibility. Change of LSBs in selected blocks can't be easily captured by the HVS beacuse these blocks contain minimum details of the cover image as shown in the example in figure 5.

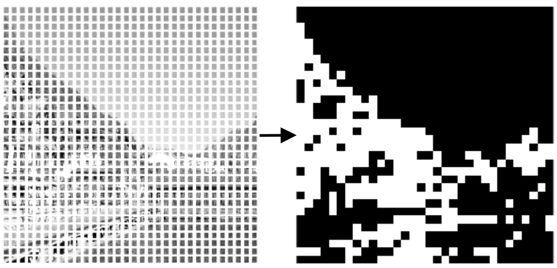


Figure 5. Detection of blocks containing detail below median and arithmetic average

With the UTF-8 character encoding, the number of letters u_c that can be embedded in each selected block can be calculated by dividing the number of pixel count of selected blocks by 8 per character as in equation (14).

$$u_c(F_{(i,j)}) = \{\frac{C_1 x C_2}{32x32x8}, F_{(i,j)} > \delta\} \qquad (14)$$

For example, if the selected block size is 16x16 pixels, 1 bit of each pixel will be used for embedding, so a total of 256 bits (32 bytes) can be embedded in this block. Each character is encoded by 1 byte in a hidden message. So, 32 character message can be embedded in a block sized 16x16 pixels. Once the number of characters u_c has been calculated, the standard LSB algorithm is applied to the selected block as in equation (15).

$$O'_{(i,j)} = \begin{cases} O_{(i,j)} - 1, LSB\left(O_{(i,j)}\right) = 1, m = 0 \\ O_{(i,j)}, \quad LSB\left(O_{(i,j)}\right) = m \\ O_{(i,j)} + 1, LSB\left(O_{(i,j)}\right) = 1, m = 1 \end{cases} \qquad (15)$$

Here; *m* is the next bit of the secret message (Jain et al., 2012). At last step; stego image is created with the composition of message embedded blocks and original blocks as in figure 6.

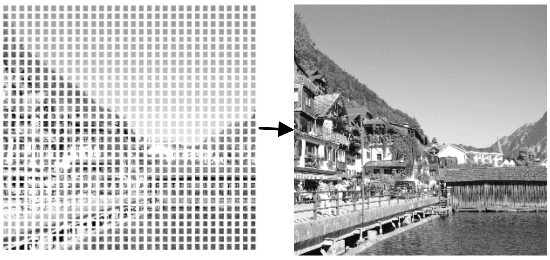


Figure 6. Generated stego image with the composition of 32x32 blocks

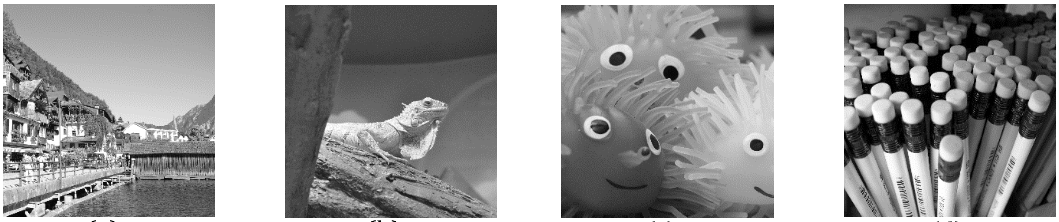
Many data-hiding methods use the inverse of embedding algorithm as the extraction algorithm. The standard LSB algorithm scans all the image pixels and adds the LSB of each pixel to a binary array. The resulting sequence is finally converted to a character set and so hidden message is obtained. However, the proposed method embeds the hidden message in some blocks of the image, not the whole image. Therefore, if a standard LSB extraction algorithm is applied to this image, a meaningless bit sequence is obtained. In order to find out which blocks contain the hidden message bits; it is necessary to repeat the operations in equations (8) to (16). After the amount of letters u_c that can be embedded in each block is calculated, the hidden message in the selected blocks can be extracted as in the equation (18) with the same algorithm.

$$m_{(n)} = \begin{cases} 0, O'_{(i,j)} = 0 \\ 1, O'_{(i,j)} = 1 \end{cases} \quad (16)$$

Here, n shows the index number of binary sequence which composes hidden message m.

4. Results

Figure 7 shows four different grayscale images with 1024x1024 pixels resolution which are used to test proposed method.



(a) (b) (c) (d)

Figure 7. Test images (a) Hallstatt (b) Zoo (c) Toys (d) Pencils

Table 1 shows data load u_c of these test images which is calculated as in equation (14).

Table 1. δ ve u_c of test images

IMAGE	ARITHMETIC MEAN OF F MATRIX δ	DATA LOAD u_c
Hallstatt	0.0055	312
Zoo	0.0032	310
Toys	0.0028	575
Pencils	0.0033	422

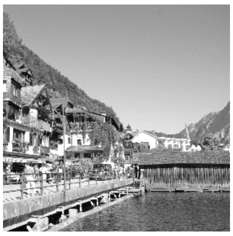
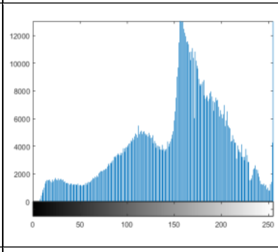
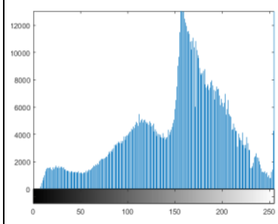
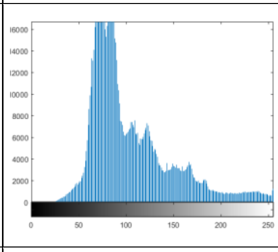
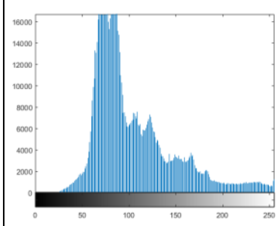
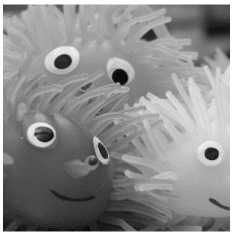
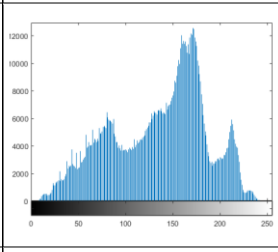
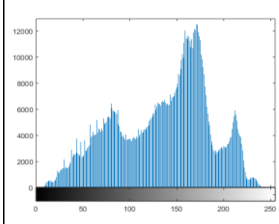
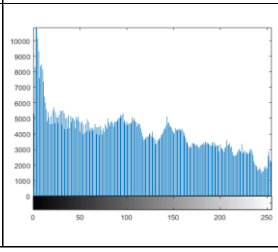
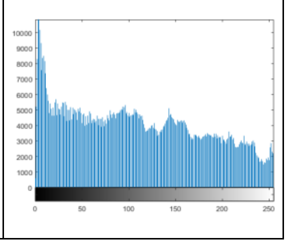
As it is shown in table 1, the data load u_c value obtained from the test images varies between 312 and 575 characters. The method does not allow embedding a message longer then calculated u_c value. So, a 256 characters long message is embedded to these four images to test the method efficiently. Image histograms were taken before and after embedding. Table 2 shows the input and output images of the method with 256 characters embedded message. In the first column of table 2, the blocks selected for embedding message are shown in black color and the blocks with protection are shown in white color.

The performance of a steganography method is measured by the amount of change applied to the cover image. A good steganographic method applies the least possible change in the cover image. An average LSB method is assumed to change half of the last bit of each pixel. In this case, the amount of possible change in the image can be 1/256 for each pixel. It is assumed that half of the LSB pixels of image may change and it means a change about

1/512 for the whole image. The greater the amount of change means the less imperceptibility and likely to be detected with HVS. There are some metrics to determine the amount of change on the image. Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR) and Structured Similarity Index (SSIM) are used to evaluate the performance of the proposed method.

MSE is a measurement of signal fidelity. One of the signal is considered as original and the other one is considered as corrupted or processed. The objective of MSE is to compare two signals by providing a quantitative score that defines the level of error or distortion (Chopra et al.,2012). In the proposed method; *C* is the original image and *C'* is a stego image with *N* pixels.

Table 2. Application and results of proposed method on test images

BLOCKS *O(i,j)*	ORIGINAL IMAGE *C*	STEGO IMAGE *C'*	ORIGINAL HISTOGRAM	STEGO HISTOGRAM
				
				
				
				

Signals *C* and *C'* are supposed as discrete infinite signals. x_i and y_i are *i.* samples of these signals as in equations (17) and (18).

$$C = \{x_i | i = 1,2,3,\dots,N\} \quad (17)$$

$$C' = \{y_i | i = 1,2,3,\dots,N\} \quad (18)$$

In this case, the MSE value between *C* and *C'* is calculated with equation (19).

$$MSE(C,C') = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \quad (19)$$

MSE is transformed to PSNR with equation (20);

$$PSNR = 10log_{10}\frac{L^2}{MSE} \quad (20)$$

L is the dynamic range of the allowed image pixel densities. The grayscale images used in the application use 8-bit color value for each pixel. In this case the L value is calculated in equation (21).

$$L = 2^{bc} - 1 = 2^8 - 1 = 255 \quad (21)$$

Here; *bc* is bit count stored per pixel. Visual distortion in the resulting stego image C' can be measured and analyzed by the corresponding PSNR (Luo et al., 2010). If C and C' are equal images, the PSNR value will be infinite. If they are completely different, PSNR value will be 0.

The distribution of pixels is not completely changed during a digital transformation to natural images. This fact is the basic idea for the definition of SSIM. Original C and modified C' images are separated into brightness, contrast and structure components. These components are defined as in equations (22), (23) and (24).

$$l(C, C') = \left(\frac{2\mu_C\mu_{C'} + k_1}{\mu_C{}^2 + \mu_{C'}{}^2 + k_1}\right) \quad (22)$$

$$c(C, C') = \left(\frac{2\sigma_C\sigma_{C'} + k_2}{\sigma_C{}^2 + \sigma_{C'}{}^2 + k_2}\right) \quad (23)$$

$$s(C, C') = \left(\frac{2\sigma_{CC'} + k_3}{\sigma_C + \sigma_{C'} + k_3}\right) \quad (24)$$

After calculating *l*,*c* and *s*; local SSIM is calculated in equation (25).

$$SSIM(C, C') = l(C, C')^{\alpha}.c(C, C')^{\beta}.s(C, C')^{\gamma} \quad (25)$$

In equation (25); α, β and γ are the three parameters used to adjust the importance of each of the three components. μ_C and $\mu_{C'}$ are sampling mean of C and C' respectively. σ_C and $\sigma_{C'}$ are sample standart deviations of C and C'. $\sigma_{CC'}$ is the cross corelation of C and C' after subtracting their means. k_1, k_2 and k_3 are small positive constants that stabilize each term. Thus, samples, variances or correlations close to zero do not cause numerical instability. All SSIM measurements between the original and the reference image are calculated by taking the average of the local SSIM values which are taken over the whole image (Swanson et al., 1996). The two images match the same if the SSIM result is 1.

The performance metrics obtained after embedding data to four different test images with developed method are shown in table 3.

Table 3. Performance results of proposed method with sample images

IMAGE	MSE	PSNR	SSIM
Hallstatt	0.00097370	78.2465	1.0000
Zoo	0.00095272	78.3411	1.0000
Toys	0.00097370	78.2465	1.0000
Pencils	0.00096130	78.3022	1.0000

MSE is measured to calculate PSNR. But proposed method achieves MSE values below 0.001 which is a successfull value for steganography. The minimum PSNR for unnoticed visual disturbance is 30dB. PSNR with 53 value for 100% embedding (1 bit per 1 pixel) is treated as excellent. Generally for standard algorithms, PSNR value should be 50 and above for 100% embedding. Proposed method achieved PSNR values over 78 with success. SSIM result 1.0000 is related to the SSIM function of the Matlab program which is running with a precision of 1/10000. The developed embedding method has made so little changes in the cover image *C;* thus SSIM is calculated over 0.9999 and Matlab floors the value up to 1. The standard LSB algorithm, as it is mentioned earlier, makes a change of 1/512 in the image, so that the SSIM value is 0.9981 in general. Calculated SSIM value as 1 in all test images shows the efficieny of proposed method. In addition, 32x32 blocks were used in the test method.

5. Conclusions

It has seen in the tests that increasing the number of blocks increases the data load of the method. However, it increases the complexity and duration of the process. It also increases the robustness of embedded data. While applying the method, all these effects can be considered and the expected imperceptibility, robustness and processing speed can be optimized by changing the number of blocks. The method differs from the methods in the literature with the advantage of calculating the data load capacity of the image before data hiding process. It is also very difficult for unauthorized persons to detect hidden data through the human visual system, because the message is not embedded in the whole picture in order as standart LSB methods. In other words, the method hides data into different pixels of each different image.

Acknowledgement

This work was supported by Scientific Research Fund of the Suleyman Demirel University. Project Number: 4382-D1-15.

Conflict of Interest

No conflict of interest was declared by the authors.

References

Brainos, A., 2004. A Study of steganography and the Art of Hiding Information, Security Writer.
Chandramouli, R., Memon, N., 2001. Analysis of LSB based image steganography techniques, Proceedings of the International Conference on Image Processing, 3(1), 1019 – 1022.
Cheddad, A. Condell, J., Curran, K., McKevitt, P., 2010. Digital Image Steganography: Survey and Analysis of Current Methods, Signal Processing, 90(1), 727–752.
Chopra, D., Gupta, P., Sanjay, G., Gupta, A., 2012. Lsb Based Digital Image Watermarking For Gray Scale Image, IOSR Journal of Computer Engineering, 6(1), 36-41.
Chore, A.M., Tiwari, N., 2017. Survey on Different Methods of Digital Audio Watermarking, Int. Journal of Engineering Research and Application, 7(6), 113-116.
Eggers, J.J., Bauml, R., Girod, B., 2002. A Communications Approach to Image Steganography, Proceedings of SPIE, Security and Watermarking of Multimedia Contents IV San Jose, California, USA.
Hamid, N., Yahya, A., Ahmad, R.B., Al-Qershi, O.M., 2012. Image Steganography Techniques: An Overview, International Journal of Computer Science and Security (IJCSS), 6(1), 3012, 168-187.
Hamza, Y.A., 2008. Blok Kırpma Kodlamasına ve Ayrık Dalgacık Dönüşümüne Dayalı, Dayanıklı Dijital Renkli Resim Damgalama Sistemi, Anadolu University, Institute of Natural and Applied Sciences, Master Thesis, 92p.
Jain, N., Meshram, S., Dubey, S., 2012. Image Steganography Using LSB and Edge Detection Technique, International Journal of Soft Computing and Engineering, 2(3), 217-222.
Johnson, N.F., Jajodia, S., 1998. Exploring Steganography: Seeing the Unseen, IEEE Computer, 26–34.
Luo, W., Huang, F., Huang, J., 2010. Edge Adaptive Image Steganography Based on LSB Matching Revisited, IEEE Transactions on Information Forensics and Security, 5(2).
Por, L.Y., Delina, B., 2008. Information Hiding: A New Approach in Text Steganography, 7th WSEAS Int. Conference on Applied Computer & Applied Computational Science (ACACOS '08), Hangzhou, China.
Swanson, M.D., Zhu, B., Tewfik, A.H., 1996. Transparent robust image watermarking, International Conference on Image Processing, 3(1), 211-214, Laussane, Switzerland.
Şatır, E., (2013). Bilgi Güvenliği İçin Metin Steganografisinde Yeni Bir Yaklaşım, Selçuk University, Institute of Natural and Applied Sciences, Master Thesis, 82p.
Yalman, Y., Ertürk, İ., 2009. Gerçek Zamanlı Video Kayıtlarına Veri Gizleme Uygulaması, XI. Academic Informatics Conference Presentations, Harran University, Şanlıurfa, Turkey, 545-552.
Yu, Y.H., Chang, C.C., Hu, Y.C., 2005. Hiding secret data in images via predictive coding, Pattern Recognition, 38(1), 691–705.