

# Modified Multi-Level Steganography to Enhance Data Security

Shadi Elshare, Nameer N. EL-Emam

Department of Computer Science, Philadelphia University, Jordan

**Abstract:** Data-hiding using steganography algorithm becomes an important technique to prevent unauthorized users to have access to a secret data. In this paper, steganography algorithm has been constructed to hide a secret data in a gray and a color images, this algorithm is named deep hiding/extraction algorithm (DHEA) to modify multi-level steganography (MLS). The suggested hiding algorithm is based on modified least significant bit (MDLSB) to scatter data in a cover-image and it utilizes a number of levels; where each level perform hiding data on a gray image except the last level that applies a color image to keep secret data. Furthermore, proper randomization approach with two layers is implemented; the first layer uses random pixels selection for hiding a secret data at each level, while the second layer implements at the last level to move randomly from segment to the others. In addition, the proposed hiding algorithm implements an effective lossless image compression using DEFLATE algorithm to make it possible to hide data into a next level. Dynamic encryption algorithm based on Advanced Encryption Standard (AES) is applied at each level by changing cipher keys (Ck) from level to the next, this approach has been applied to increase the security and working against attackers. Soft computing using a meta-heuristic approach based on artificial bee colony (ABC) algorithm has been introduced to achieve smoothing on pixels of stego-image, this approach is effective to reduce the noise caused by a hidden large amount of data and to increase a stego-image quality on the last level. The experimental result demonstrates the effectiveness of the proposed algorithm with bee colony DHA-ABC to show high-performing to hide a large amount of data up to four bits per pixel (bpp) with high security in terms of hard extraction of a secret message and noise reduction of the stego-image. Moreover, using deep hiding with unlimited levels is promising to confuse attackers and to compress a deep sequence of images into one image.

**Keywords:** Steganography, Multi-level steganography, Bee Colony Algorithm, least Significant Bit, Image Smoothing, Segmentation Image

## 1. Introduction

Data security is one of the most important trends; it increases every day due to the massive spread of internet services and the huge demands and dependability on the internet for exchanging valuable data [18]. In addition, data exchanging like money transfers to user bank account and services authentication, have also become the most security demanding sectors over the Internet, they need to protect services from attackers [15]. Therefore, data hiding techniques using steganography algorithms are taking place, since it covers the existence of data transmission. Recently, many new tools have been developed by steganalysis to extract successfully valuable secret information from stego-file. Therefore many researchers

have been developed steganography algorithms to make hard to extract secret data by steganalysis and making the human visual system (HVS) difficult to find a slight difference that happens on a cover data such as audio, image, and video after hiding process. These steganography algorithms have been developed using the following approaches:

- I. Modified LSB (MDLSB).
- II. Hiding secret data randomly instead of sequentially.
- III. Using soft computing to make smoothing on pixels' values.
- IV. Apply multi-level steganography (MLS) instead of single-level steganography (SLS), where multi-level steganography (MLS) is promising to provide more strength compare to single-level steganography (SLS) technique [24].

In the proposed steganography algorithm, the above approaches have been developed to construct high secure hiding algorithm

There are two important factors that every successful and powerful steganography system should take into consideration; these factors are the hiding efficiency and the hiding payload. The first factor developed by researchers to satisfy the efficiency of steganography scheme by making a visual quality of stego-file matching with the cover-file. When viewers notice any distortion that will raise the probability of attacker's suspicion, then steganalysis tools will detect the secret data easily. The second factor developed by researchers to hide high payload means a large capacity of secret information to be concealed in cover-file. These two factors, hiding efficiency and hiding payload, conflict with each other. When the data hiding efficiency goes up, the data hiding payload goes down. The users' requirements and the type of steganography scheme are playing basic role to change these two factors [20].

The rest of the paper is structured as follows: In Section 2, related works and the requirements of deep hiding/extracting algorithm is appeared in section 3. Phases of the proposed steganography algorithm based on deep hiding with bee colony algorithm (DHA-ABC) are appearing in Section 4. In Section 5 the deep extraction algorithm (DEA) has been discussed. Experimental results are discussed in Section 6. Finally, Section 7 summarizes the main conclusions of the proposed algorithm.

## 2. Related Work

Steganography techniques are mostly applied to digital file formats and it is popular for concealing information in image and audio files. Latika, Gulati, Y. (2015) [16]] explained types of data media used by steganography techniques, these types are divided to four main categories; text, image, audio, and protocol, where the first three

involves concealing information in the respective file format, while the protocol is more advanced and involves using the communication platform or protocol to hide the information.

Jain, R., Kumar, N. (2012) [12] proposed a technique based on substitution approach using the least significant bit (LSB), as known, the LSB has wide range used applications with the ability to hide information in 8-bit and 24-bit images, which is working on the concept that information is hidden in the least significant bit of a pixel byte, thus making it practically impossible for the human eye to notice that there is a difference between the new image which has the secret data and old one. In their proposed algorithm, they adopt large data hiding mechanism also they used lossless compression technique applied on the secret message to reduce its size, then they test each pixel in the cover file for the density level and based on it they hide a different number of secret bits in each pixel. On the other hand, the algorithm still weak to handle by statistical analysis, since it uses sequential hiding mechanism, also the data are being hidden without any type of encryption which makes it very easy to be extracted.

Other researchers reduce the noise on the cover-image by hiding one bit per pixel [4] and maintained that LSB achieves its objective by altering the coloring scheme of an image, thus making it difficult for the human eye to discern the content and found that LSB does not distort the image. Yang, C., et al. (2008) [25] modified LSB technique known as adaptive data hiding, whereby it works on the principle of hiding information within the edges of an image. This technique has been applied to calculate two-pixel values; one from the smooth edge (where the color changes between neighbors in a fixed manner) and the other from the sharp edge (where the color changes between neighbors in great and noticeable manner) of the image. The difference in the pixel value has been implemented to determine the pixel that would be substituted with the information to be concealed.

Babita, A., Kaur, M. (2009) [3] suggested other LSB technique and algorithms that utilize the RGB coloring scheme. In particular, the method uses the median of the respective code values of the four color schemes to determine the criteria of hiding technique of a secret data in a cover-image. Authors describe another LSB algorithm that utilizes the texture format of an image. In this case, the simple texture generates 3 LSB channels while the complex texture generates 4 LSB channels.

Many researchers have been adopted new methods of substitution technique to overcome the problem of sequence hiding in LSB which makes data easy for extracting secret data, these methods are based on randomizing the distribution of hidden data over the cover-image or splitting the hidden message itself to blocks to be hidden in different locations inside the cover-image. Moreover, researchers proposed Matrix Embedding with Repeat.

Bawaneh, M., et al. (2018), [5] proposes a new secure technique called flash video (FLV) file steganography that keeps the frame video quality and is difficult to detect. The technique can hide any type of secret message inside a given FLV file. The secret message is divided into packets of the same length, reordered packet, and encrypted bytes before being hidden at the end of a selected video tag. A simulated annealing (SA) approach to select tags for steganography is presented to reduce or avoid the challenge of steganalysis.

Accumulate (ME-RA) based steganography to operate on the principle of hiding bits using randomly chosen blocks of the message. This method is reliant on the performance of the computer software being used to produce quality concealed images [22].

There are many mechanisms involve dividing and partitioning the cover-image into segments [13], this approach proposed to substitute data hiding method to encrypt the message in non-adjacent and random pixel locations using the edges in the image and then smooth its areas. After encryption, the data is hidden randomly by selecting edges pixels in 1-3-4 LSBs of RGB channels. This algorithm is efficient to select a suitable location of pixels in color image to reduce the noise after the hiding process by working on the edges. However, this algorithm is not applying soft computing to perform smoothing on the pixel after the hiding process, moreover, it does not apply MLS.

Karim, S., et al. (2011) [14] using a secret key to locate and set the positions within the cover-image to hide the secret message, also they use with the secret key the red channel of the cover-image for each cover pixel and use the LSB value of the red color which interacted with the current bit of the secret key using the operation exclusive OR (XOR). If the result of the XOR operation is 0, then the hidden bit will set in the LSB of the blue color in the current pixel. Otherwise the hidden bit will set in the LSB of the green color and so on. So, it is difficult to retrieve the hidden data in absence of the secret key. However this technique cannot embed large amount of data and using single-level is not enough to make high secure system.

Akhtar, N., et al. (2013) [2] applied RC4 algorithm (Rivest Cipher 4) with a stego-key to randomize the embedding of secret data to generate random order over the entire cover-image to hide the secret data into the pixels according to the random order. They also have introduced a new technique called bit-inversion to improve the stego-image quality. The technique works by splitting the pixels into four sections according to the third and second bits values. The first section is of all the pixels that have the third and second bits with 00 values. The second section is of all pixels with third and second bits of 01 values, and so on, the third is of 10 values and the fourth is of 11 values. Finally, for each section, the count of changed and unchanged pixels is found, and if the number of changed pixels is greater than the unchanged pixels, then the LSB of the section is inverted.

Medeni, O., Souidi, M. (2010) [19] discussed another technique based on pixel value differencing (PVD), which works by combining various blocks that connect data pixels for embedding. Argue that the majority of the popular steganographic techniques are designed to eliminate distortion, and thus they have some similarity with the models in the distortion category. In this context, the popular technique utilizes an error encoding technique, whereby the majority of the processing operation involves the functionality to extract code from cover material. They implement their works by segmenting the grayscale image into blocks, and for each block, they hide the secret message inside the edges of the image depending on the four left bits value of the pixel for reducing the noise. However, using the edges only will reduce the hiding capacity in a massive way, also the algorithm applies only on grayscale images and does not support colored images.

Gaikwad, V. B., Wagh, V. G. (2010) [10] mentioned that steganography is achieved high security if it is able to restore the cover-image to its original content. This technique uses tunneling to recover blurred images obtained from the Point Spread Function (PSF). Despite the proliferation of a high number of steganography techniques, there exist some common limitations that harbor their efficiency and performance levels. The least significant bit, in particular, relies on a fixed bit operational model in an embedded grayscale. Also, the algorithm depends on a third randomly generated key that needs to be extracted from the image which hidden over the final image and can be detected easily.

El-Emam, N., Qaddoum, K. (2015) [8] presented a steganography security technique using irregular segmentations of an image, and a hybrid adaptive neural network with modifying on Ant colony optimization. This technique is applied to strengthen the data hiding security and raise the payload capacity. The algorithm holds four levels of safety, the first level uses a set partition in hierarchical trees (SIPHT), and advanced encryption standard (AES) mechanisms respectively to compress and encrypt the secret message. The second safety level is applying irregular image segmentation on the cover-image, depending on the adaptive reallocation segments' edges (ARSE). ARSE finds the numerical solution of the proposed partial differential equation (PDE) when applying an adaptive finite-element method (AFEM). The third safety level contrasts a learning system by using a hybrid adaptive neural network with a modified ant colony optimizer (ANN\_MACO). This system generates input patterns to be byte features to modify the cover-image, using a support vector machine (SVM). The proposed method proves high-security robustness, large capacity amount reaches to 6 bits per pixels in color cover-images, and precise data recovering. However, using the hybrid adaptive neural network with a modified ant colony optimizer to smooth the selected pixel using single-level is not enough to produce high secure system.

El-Emam, N., Al-Zubidy, R. (2013) [6] proposed a new steganography with a modified adaptive genetic algorithm using hybrid adaptive neural networks to conceal a large amount of secret message technique in color images by using non-uniform adaptive image segmentation with an intelligent computing segmentation, this technique is used to conceal secret message randomly instead of sequentially. In addition, four security levels implemented to increase the resistance against statistical and visual attacks and decrease distortion of a color channel and high protection of secret message. Furthermore, the stego-images produced by this algorithm are not detected by the state-of-the-art steganalysers, but this algorithm using single-level steganography.

El-Emam, N. (2015) [7] proposed a new data-hiding algorithm based on adaptive neural networks with modified particle swarm optimization. This algorithm used for embedding a large amount of secret messaging in a color image this algorithm employs an impressive image segmentation algorithm that uses two levels of adaptive non-uniform segmentation to embed data randomly instead of sequentially. For each byte and for each color in a cover-image, a non-uniform number of bits to be replaced by secret bits is imposed, depending on the byte characteristic's assessment using a weighting factor that is created from a

cipher key to damp the difference in the surrounding twelve high bytes for the current byte, also a machine learning has been proposed to adjust a pixel's value based on an adaptive neural network with modified particle swarm optimization and all of this with five protection levels for the security that work against statistical and visual attacks. The experimental results of this algorithm show that the difference values between the cover pixels and their corresponding stego-pixels are small, and attackers would not be expected regard PE less of whether the hidden data were embedded, also the five levels of protection are very significant in obtaining good performance, but this algorithm using single-level steganography.

Al-Shatanawi, M., El-Emam, N. (2015) [1] proposed a new steganography algorithm for hiding a large amount of secret data. Basically, the algorithm depending on different size image segmentation (DSIS) and modified a least significant bit (MLSB). DSIS applied random embedding secret message. Each byte has a non-uniform number of bits to be replaced, by contrasting an effective hypothesis based on byte characteristics. The results show that the algorithm is efficient, high imperceptible, high capacity payload reaching four bits per byte, but this algorithm had been applied single-level steganography.

El-Emam, N., Al-Diabat, M. (2015) [9] presented hiding data algorithm in color images by constructing a three phases intelligent technique and preserves high retrieving data. The first phase is applied before hiding the secret message and the others after hiding. The first phase basically estimates the number of bits to be hidden, using an adaptive neural network with the genetic algorithm. The second phase is checking up the proposed steganography algorithm performance; by using the available cover-image to create rich images models. The third phase improves the stego-image and defends attacks, based on the concurrent approach. This algorithm is providing different capacities, since estimating the number of hidden bits, but this algorithm using single-level steganography.

Other steganography techniques combine the randomization mechanisms with multi-level hiding and smoothing to generate most security capabilities. Sikarwar, S. (2012)[24] proposed MLS approach based on an integrated and synchronized protocol, combining effort of secure channel algorithm, dynamic key cryptography, and multilevel steganography; since provides more strengthen compare to simple steganography. Multilevel steganography and dynamic cryptography derive the idea about an integrated synchronized protocol for secure information transmission. Secure channel for sharing initial information is the success of any dynamic cryptography. This approach provides architecture for a secure channel, but it is working on a few numbers of levels and using uniform hiding algorithm from level to next one.

The traditional Multi-Level Steganography (MLS) algorithm is defenseless; due to use a static encryption key with a limited number of levels and hiding secret data sequentially. Therefore, in this paper, a modified MLS has been proposed defined deep hiding/extracting algorithms (DHEA) which is based on dynamic encryption keys with the unlimited number of levels and hiding secret data randomly using two layers. Furthermore, the most commonly used steganography algorithm based on least significant bit (LSB), this algorithm is easy to retrieve a secret data, therefore, a modified LSB (MDLSB) has been proposed to

confuse attackers, this algorithm is applied a non-uniform number of bit to be hidden. Finally, hide a large amount of a secret data using traditional hiding technique make the imperceptible level of stego-image is very high, therefore, we introduce image smoothing based on Artificial Bee Colony (ABC) to produce high imperceptible stego-image.

### 3. Requirements Of Deep Hiding / Extracting Algorithm

It is essential to define the main specifications of the proposed new steganography algorithm based on deep hiding with bee colony algorithm (DHA-ABC) as follows.

#### 3.1 Multi-levels steganography (MLS) technique

The new approach of MLS has been proposed to implement deep hiding (DHD) by using a recursive loop to hide secret data upon unlimited levels. The formal definition of hiding and extracting techniques are defined in the following definitions

##### Definition 1:

Let the recursive function defines in the map  $(DHD : (HD)^n \rightarrow Is)$  be a deep hiding function (DHD) based on MLS, where HD is hiding function map  $(HD: Ic \times CESm \times Ck \times Nbpp \rightarrow Is')$ ,  $Ic$  is a cover-image,  $CESm$  is a compressed encrypted secret message,  $Ck$  is a cipher key and  $Nbpp$  is a number of bit per pixel.

##### Definition 2:

Let the function map  $(EX: Is \times Ck \times Nbpp \times loc_s^n \rightarrow ECSm)$  is the extraction function (EX), where its domain bases on three parameters ( $Is$ ,  $Ck$ , and  $Nbpp$ ), while the function rang is  $ECSm$ .

#### 3.2 Image segmentation

Image segmentation is the process of splitting images into multiple segments, so the image to become more clear so it will be easier to analyze and understand by showing the borders in the image exactly.

##### Definition 3:

Let image segmentation define in the map  $(C_{non-uniform} : Ic \times Ck \rightarrow Seg)$ , where  $C_{non-uniform}$  represents the segmentation function based on non-uniform segments, and  $Seg$  is the set of non-uniform segments constructed from  $Ic$ . Eq. (1) illustrates the relation between  $Seg$  and  $Ic$ .

$$Ic = \bigcup_{\forall i} Seg_i \quad (1)$$

where  $Seg = \{Seg_1, Seg_2, \dots, Seg_n\}$ .

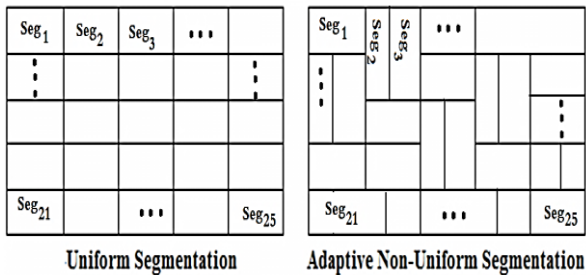


Figure 1. Types of image segmentation

#### 3.3 Image smoothing

Image smoothing is a mechanism for reducing the noise in the stego-image caused by the hidden data.

##### Definition 4:

Let image smoothing define in the map  $(BCA: Is \rightarrow \hat{Is})$ , where  $|\hat{Is} - Ic| < |Is - Ic|$ ,  $Is'$  is a stego-image after smoothing process.

#### 3.4 Image compression and decompression

Data compression bases on lossless approach using (Deflate algorithm), this algorithm has been applied to avoid the problem of lossy approach, where Deflate algorithm is associated file format that uses a combination of the LZ77 algorithm and Huffman coding that is convenient to use. The formal definition of image compression and decompression are defined in the following:

##### Definition 5:

Data compression function defines in the map

$$(Comp: Is \times \hat{U}_m \times \hat{U}_s \times \check{I}_m \rightarrow CIs) \&$$

$(Comp: Sm \times \hat{U}_m \times \hat{U}_s \times \check{I}_m \rightarrow CSm)$ , where  $(Comp)$  is a compression technique,  $\hat{U}_m$  is unimportant image

information,  $\hat{U}_s$  is a list of unimportant sets,  $\check{I}_m$  is a list of important image information and  $CIs$  represents a stego-image after compression [1].

##### Definition 6:

Data decompression function defines in the map  $(DComp: CIs \times \hat{U}_m \times \hat{U}_s \times \check{I}_m \rightarrow Is):$ , where  $(DComp)$  is a decompression technique.

#### 3.5 Image encryption and decryption

The Advanced Encryption Standard tool (AES) has been chosen to meet the objective that needs a very strong encryption mechanism without the complexity of sharing multi-keys. The formal definition of image encryption and decryption are defined in the following:

##### Definition 7:

Image encryption is defined in the map  $(Eny: CIs \times LCIs \rightarrow ECIs) \& (Eny: CSm \times LCSm \rightarrow ECSm)$ , where  $CIs$  is a compressed stego-image,  $CSm$  is a compressed secret message,  $LCIs$  is a length of  $CIs$ ,  $LCSm$  is a length of  $CSm$ ,  $ECIs$  represents a compressed stego-image after encryption and  $ECSm$  represents a compressed secret message after encryption.

##### Definition 8:

Image decryption is defined in the map  $(DEny: ECIs \times LCIs \rightarrow CIs)$ .

#### 3.6 Randomized pixel selection

The proposed approach scatter the secret message over the cover-image in non-uniform way based on secret keys  $C$  and  $D$ .

##### Definition 9:

Let the next selected pixel position  $(RND: Seg \times C \times D \rightarrow P_{i,j})$ , where  $RND$  represent the Randomization function,  $Seg$  is the selected segment to select randomized pixels inside,  $C$ ,  $D$  is the two parameters that are generated from the secret key entered by the sender, and  $P_{i,j}$  is the selected pixel location.

#### 4. Deep Hiding with Artificial Bee Colony Algorithm (DHA-ABC)

The proposed steganography algorithm is constructed to hide a large collective of a secret message with rigid security while working against visual and statistical attacks. Deep hiding based on Multi-Levels Steganography (MLS) has been proposed as given in Figure 2.

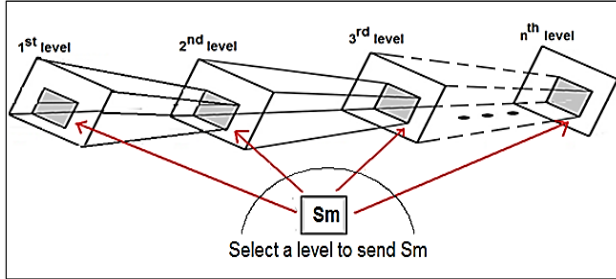


Figure 2. Deep hiding based on MLS

For each level, a sequence of procedures is implemented to hide secret data in cover-image, where the first procedure is based on data compression of (Sm) to produce (CSm) or data compression of (Is) to produce (CIs). The second procedure is attentive on encryption of (CSm) to produce (ECSm) or encryption of (CIs) to produce (ECIs). The third procedure is the hiding technique to embed randomly (ECSm) in (Ic). The deep hiding technique is desirable to use (Ck) and sequence of pixels' locations to hide data randomly for each level except the last one that requires an extra process to confirm the security by adding image segmentation procedure to scattering secret data into segments. Hiding technique needs to prepare an (Ic) that holds a compress and an encrypted secret data. The output of ( $i^{th}$ ) level is (Is) that represents a secret data of ( $i+1$ )<sup>th</sup> level. The dynamic transfer mode has been adapted from level to next one by varying the method of calculating a (Ck<sub>i</sub>) and applying different data hiding technique among levels. Finally, to improve the security, ABC has been applied at the last level to make smoothing on (Is) and to confirm the imperceptibility, see Figure 3.

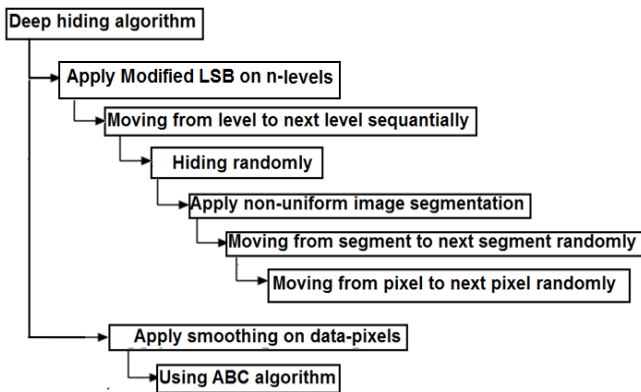


Figure 3: Structure of deep hiding algorithm.

The deep hiding architecture shown in Figure 4 illustrates steps of process to produce (Is) upon n-levels.

##### 4.1. Using modified LSB (MDLSB)

Random pixels location have been calculated to scatter secret data randomly, this calculation is used the current pixel location to find the next pixel location, this technique is needed two extra parameters, they are generated randomly from the input secret key entered from the sender (loc1, loc2) as illustrated in chapter two, see definition 9. On the

other hand, a color image has been used at the last level to hide stego-image  $Is_{n-1}$  of the previous level upon three color components using MDLSB algorithm. In the last level, non-uniform image segmentation has been implemented to hide secret data randomly using two layers of randomization, the first layer is moving from segment to next segment randomly, while the second layer moves from pixel to next pixel randomly at each segment. The proposed MDLSB algorithm is applying non-uniform numbers of bits to be hiding per byte, where the number of bits that will be replaced at each component will be determined in the algorithm 2.

##### 4.2 Non-uniform image segmentations:

A proposed image segmentation algorithm is based on the definition 3 has been proposed in this paper and implemented in the last level. This algorithm is based on adaptive non-uniform segmentation of a cover-image (Ic). The following steps are applied to display the proposed image segmentation. The segmentation algorithm depends on a security key (SK), the algorithm starts by dividing the cover-image into 4 non-uniform vertical slots and (n) non-uniform horizontal slot, as shown in Figure 5.

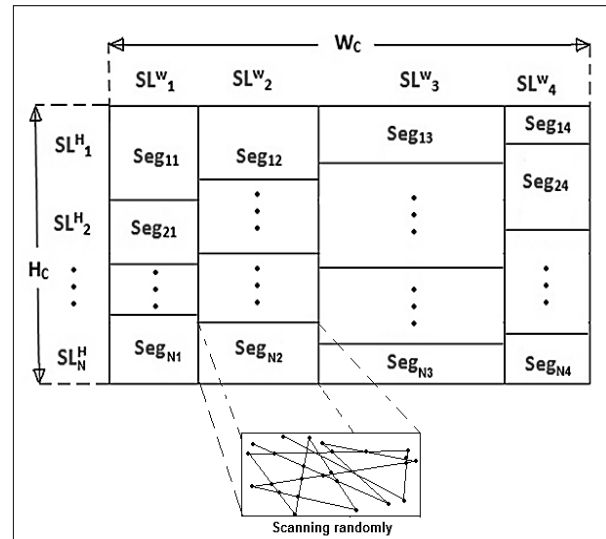


Figure 5. Segmentation method with pixels scanning randomly

The proposed segmentation algorithm steps are illustrated below:

**Algorithm1:** Segment Seg(Ic,Ck) // Segmentation

{  
/\*

Let  $W_c$  be an image width;

Let  $SK_{bea}$  be a key using in segmentation;

Let  $SL^w_i$  be a slot width where  $\forall i=1, \dots, 4$ ;

Let  $SL^h_j$  be a slot high where  $\forall j=1, \dots, N$ ;

Let  $H_c$  be an image high;

Let  $Seg_{LS}$  be a segment at the location (L, S) where  $\forall L = 1 \dots n$  and  $\forall S = 1 \dots 4$ ;

\*/

**Step1:** Generate  $SK_i^c$ ;

$SK_i^c = Rand(.); // \forall i=1, \dots, 4, c = \{R, G, B\}$

**Step2:** For each color component in Ic; //  $\forall c = \{R, G, B\}$

{

**Step2.1:** Compute  $SL^w_1$ ;

$SL^w_1 = (W_c/SK_1) + (W_c/100 \times SK_2);$

(2)

**Step2.2:** Compute  $SL^w_2$ ;



$$SL^{W_2} = (W_C - SL^{W_1} / SK_1) + (W_C / 100 \times SK_3); \quad (3)$$

**Step2.3: Compute**  $SL^{W_3}$ ;

$$SL^{W_3} = (W_C - SL^{W_2} - SL^{W_1} / SK_1) + (W_C / 100 \times SK_4); \quad (4)$$

**Step2.4: Compute**  $SL^{W_4}$ ;

$$SL^{W_4} = W_C - SL^{W_3} - SL^{W_2} - SL^{W_1}; \quad (5)$$

**Step2.5: For each** slot  $SL^{W_j}$  in  $W_C$  { //  $\forall j = 1 \dots 4$

**Step2.5.1: For each** slot  $SL^{H_i}$  in  $H_C$  //  $\forall i = 1, \dots, N-1$

{  
Compute  $Seg_{i,j}$  such that

$$Seg_{i,j} = (H_C - Seg_{i-1,j} / SK_5) + (H_C / 100 \times SK_N) + N; \quad (6)$$

} // for each slot  $SL^{H_i}$

} // for each slot  $SL^{W_j}$

**Step 2.6: Compute**  $Seg_{N,j}$ ;

$$Seg_{N,j} = H_C - \sum_{s=1}^4 \sum_{k=1}^{N-1} Seg_{ks} \quad (7)$$

} // for each color component

} // Algorithm1

### 4.3 Calculate a number of bits per pixel (Nbpp)

The proposed hiding algorithm HD(.) introduce the modified LSB algorithm (MDLSB) at the level (n) by implementing a non-uniform hiding mechanism over each segment. Moreover, the algorithm split each segment into three color components (Red, Green, and Blue). Each color component and each segment, the Standard Deviation (STD) metric has been calculated. After that, all three STD value for the three color components inside each segment are sorted from biggest to lower, and then the algorithm assigns 2 bits for the highest STD value color component and 1 bit for the two other color components. Using this technique the algorithm makes sure that payload will be 4 bits per pixel over the last level (Colored cover-image).

Hiding algorithm uses a randomization function to hide a number of secret bits per pixels at each segment; this approach is proposed to damp the noise of the stego-image. In addition, the proposed DHA-ABC algorithm applies a smoothing function based on Artificial Bee Colony (ABC) algorithm which alters the unused bits in the pixel to round the value of the whole pixel to be close to  $(I_c)$ .

The proposed (DHA) has been described in the following algorithm steps:

**Algorithm2:** Image DHD(Sm, Ic, Ck) // Deep hiding

{  
/\*

**Let** n be number of levels.

**Let** Sm be secret message.

**Let**  $I_c^i$  be cover-image at the  $i^{th}$  level.

**Let**  $I_s^i$  be stego-image at the  $i^{th}$  level.

**Let** CSm be compressed secret message.

**Let** ECSm be encrypted and compressed secret message.

**Let** CIs be compressed stego-image.

**Let** ECIs be encrypted and compressed stego-image.

**Let**  $Seg_{(R,G,B)}$  be a segment of Image.

**Let**  $\sigma^{Seg_{LS}}$  be standard deviation of the segment at the location (L,S).

**Let**  $\sigma_{i,j}^{9-Neighbors}$  be standard deviation of 9-neighbors, see Figure (6).

**Let** NP be number of pixels at a specific segment.

**Let** NPx be number of pixels at the cover image  $I_c$ .

**Let**  $I_c^i$  be  $i^{th}$  pixels at a cover image  $I_c$ .

**Let**  $\mu$  be the average of pixels at the segment  $Seg_{LS}$ .

**Let**  $\mu^{Ic}$  be the average of pixels at the cover image  $I_c$ .

**Let** Nbpp be a number of bits per pixel.

**Let** NS be a number of segments.

**Let**  $\hat{I}_s^n$  be stego-image at the  $n^{th}$  level after smoothing.

**Let** LCIs be a length of CIs.

\*/

**Step1: Input** n;

**Step2: Find** CSm such that:

$$CSm = \text{Comp}(Sm, \hat{U}_m, \hat{U}_s, \hat{I}_m); \quad // \text{ See (Definition5)}$$

**Step3: Find** ECSm such that:

$$ECSm = \text{Eny}(CSm, LCSm); \quad // \text{ See (Definition7)}$$

**Step4: Find** Set of locations  $Loc^1_s$ ;

**Step5: Set** Nbpp=4;

**Step6: Find**  $Is_1$  such that:

$$Is_1 = \text{HD}(Ic_1, ECSm, Ck_1, Nbpp, Loc^1_s); \quad // \text{ See (Definition1)}$$

**Step7: For** (inti=2; i<=n-1 ;i++) {

**Step7.1: Input**  $Ic^i$ ;

**Step7.2: Compute**  $CIs^{i-1}$  such that

$$CIs^{i-1} = \text{Comp}(Is^{i-1}, \hat{U}_m, \hat{U}_s, \hat{I}_m); \quad // \text{ See (Definition5)}$$

**Step7.3: Compute**  $ECIs^{n-1}$  such that:

$$ECIs^{n-1} = \text{Eny}(CIs^{i-1}, LCIs^{i-1}); \quad // \text{ See (Definition7)}$$

**Step7.4: Find** Set of location  $Loc^i_s$ ;

**Step7.5: Find**  $Is^i$  such that:

$$Is^i = \text{HD}(Ic_i, ECIs^i, Ck_i, Nbpp, Loc^i_s); \quad // \text{ See (Definition1)}$$

} // end for i

// Modified LSB (MDLSB)

**Step8: Let** i = n;

**Step9: Compute**  $\sigma^{Ic}$

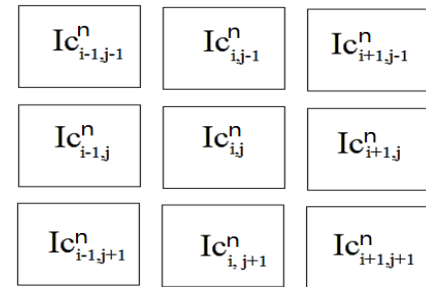
$$\sigma^{Ic} = \left( \sqrt{\frac{1}{NP} \sum_{i=1}^{NPx} (Ic_i - \mu^{Ic})^2} \right) \quad (8)$$

**Step10: Input** N, M ; // N, M be a number of rows & number of columns of  $Ic$ , where  $NPx = M \times N$ .

**Step11: Call**  $Seg(Ic^n, Ck)$  // See algorithm1

**Step12: For each** Segment  $Seg_{LS}$  in  $Ic^n$ ; //  $\forall L = 1, \dots, 4$   
and  $\forall S = 1, \dots, N$ , where  $NS = 4 \times N$

{



**Figure 6.** Nine neighbors pixels (3 × 3 Window)

**Step12.1: For each** Pixels (i,j) in  $Seg_{LS}$

{

**Step12.1.1: Find**  $\sigma^{Seg_{LS}}$ , and  $\sigma_{i,j}^{9-Neighbors}$ , see Eqs.(9-10)

**Step12.1.2: Find** set of locations at segment  $Seg_{Loc}^n_s$ ;

$$\sigma^{Seg_{LS}} = \left( \sqrt{\frac{1}{NP} \sum_{v_i, v_j}^{NP} (Ic_{i,j}^n - \mu)^2} \right)_{Seg_{LS}} \quad (9)$$

$$\sigma_{i,j}^{9-Neighbors} = \left( \sqrt{\frac{1}{9} \sum_{v_i, v_j}^9 (Ic_{i,j}^n - \mu^{9-Neighbors})^2} \right)_{9-Neighbors} \quad (10)$$

**Step12.1.3 Compute** Nbpp such that:

$$Nbpp = \begin{cases} 3 & \text{if } \left| \sigma_{i,j}^{9\text{-Neighbors}} - \sigma_{SegLS}^{SegLS} \right| > \sigma^{Ic} \ \& \ \sigma^{Ic} > 10^{-5} \\ 2 & \text{if } \left| \sigma_{i,j}^{9\text{-Neighbors}} - \sigma_{SegLS}^{SegLS} \right| < \sigma^{Ic} \ \& \ \sigma^{Ic} > 10^{-5} \\ 1 & \text{if } \left| \sigma_{i,j}^{9\text{-Neighbors}} - \sigma_{SegLS}^{SegLS} \right| < 10^{-5} \end{cases} \quad (11)$$

**Step12.1.4 Find  $I_{S_n}$  and  $\hat{I}_S^n$** 

$I_{S_n} = HD (Seg_{LS}, ECI_{S_n}, Ck^n, Nbpp, Loc^n_s);$   
*// See (Definition1)*

$\hat{I}_S^n = ABC(I_{S_n}, I_c^n);$  *// Call bee colony to perform smoothing on stego-image.*

**Step12.1.5: Find  $CI_{S_n}$  such that:**

$\hat{CI}_{S_n} = \text{Comp}(\hat{I}_S^n, \hat{U}_m, \hat{U}_s, \hat{I}_m);$  *// See (Definition5)*

**Step12.1.6: Find  $ECI_{S_n}$  such that:**

$ECI_{S_n} = \text{Eny}(CI_{S_n}, LCI_{S_n});$  *// See (Definition7)*

} *// end for each pixel*  
 } *// end for each segment*

} *// end Algorithm2*

The proposed (HD) has been described in the following algorithm steps:

**Algorithm3:** Image **HD**( $I_c, S_m, Ck, Nbpp, Loc_s$ ); *// Data hiding*

**Step1: For each** bit Value in  $S_m$  *// where |bitValue| = Nbpp*  
 {

**Step1.1: Find** BitValue from  $S_m$  such that

BitValue = Extract ( $S_m, Nbpp$ ); *// extract Nbpp from  $S_m$*

**Step1.2:** Replace  $I_c^{Loc_s}$  by Hbits;

} *// For each bit Value.*

} *// Algorithm3.*

The proposed artificial bee colony algorithm (ABC) has been applied to perform smoothing on stego-image, Figure 6 shows the steps of ABC algorithm:

**Algorithm4:** Image ABC ( $I_s, I_c$ )

{  
 /\*

**Let**  $P_{COVER}$  be the selected pixel from cover-image segment.

**Let**  $P_{STEGO}$  be the selected pixel from stego-image segment.

**Let**  $C$  be 8 bits binary value of the selected color component from  $P_{COVER}$ .

**Let**  $S$  be 8 bits binary value of the selected color component from  $P_{STEGO}$ .

**Let**  $R$  be the number of replaced bits inside  $S$ .

**Let**  $L_{COVER}$  be a list for saving the cover patterns values of  $C$ .

**Let**  $L_{STEGO}$  be a list for saving the stego patterns values of  $S$ .

**Let**  $L_{PATTERNS}$  be a list for saving the best smoothing pattern for  $C$ .

**Let**  $IMG_C$  be a segment from  $I_c$ .

**Let**  $IMG_S$  be a segment from  $I_s$ .

**Let**  $NS$  be a number of segments in  $I_c$  or  $I_s$ .

**Let**  $STD(.)$  be standard deviation function.

**Let**  $IMG_S\_width$  be image segment's width.

**Let**  $IMG_S\_Height$  be image segment's height.

\*/

**Step1: For** ( $inti=1; i \leq NS; i++$ )

{

**Step1.1: Define**  $IMG_C[i]$ ; *//  $i^{th}$  segment from  $I_c$ .*

**Step1.2. Define**  $IMG_S[i]$ ; *//  $i^{th}$  segment from  $I_s$ .*

**Step1.3. Call** **Optimized\_BEE\_Learning**( $IMG_C[i], IMG_S[i]$ );

} *// End*

} *// End Algorithm 4*

**FunctionVoid**

**Optimized\_BEE\_Learning**(Cover\_image\_segment  $IMG_C$ , Stego\_image\_Segment  $IMG_S$ )

{

**Step1: Find**  $S_R = STD (IMG_{CR}, \text{mean});$

**Step2: Find**  $S_G = STD (IMG_{CG}, \text{mean});$

**Step3: Find**  $S_B = STD (IMG_{CB}, \text{mean});$

**Step4: Find**  $\text{max} = \text{MAX} (S_R, S_G, S_B);$

**Step5: If** Color component is max then  $R$  equal 2 **else**  $R$  equal 1;

**Step6: For** ( $inti = 1; i \leq IMG_S\_width; i++$ )

**Step6.1: For** ( $int j=1; j \leq IMG_S\_height; j++$ ) {

**Step6.1.1: Call** **Smooth\_Single\_Component** ( $IMG_{CR} [i,j], IMG_{SR} [i,j], R$ );

**Step6.1.2: Call** **Smooth\_Single\_Component**( $IMG_{CR} [i,j], IMG_{SR} [i,j], R$ );

**Step6.1.3: Call** **Smooth\_Single\_Component**( $IMG_{CR} [i,j], IMG_{SR} [i,j], R$ );

} *// end for j;*

} *// end for i;*

} *// End FunctionOptimized\_BEE\_Learning*

**Function Void** **Smooth\_Single\_Component**( $C, S, R$ ) *// one color component function*

{

**Step1: If**  $|C - S| > 1$  then

{

**Step1.1: Set**  $n = 8 - R$ ;

**Step1.2: Set**  $W_b = 2^n$ ;

**Step1.3: for** ( $inti = 1; i \leq W_b; i++$ )

{

**Step1.3.1: Generate** Worker Bee ( $b_i$ )

**Step1.3.2: Set**  $\text{Bin}(b_i) = \text{LSB}(S, R) + \text{Bin}(i)$  *// generate all probabilities of smoothing patterns . Bin(.) is the binary value;*

**Step1.3.3: Set**  $F(\text{Bee}) = |C - \text{Bin}(b_i)|$ ; *// Calculate fitness function  $F(\text{Bee})$*

} *// end for i*

**Step1.4: Set**  $\text{Min} = \text{Bin}(255)$ ;

**Step1.5: For** ( $int j = 1; j \leq W_b; j++$ )

{

**Step1.5.1: if** ( $\text{Bin}(b_j) < \text{Min}$ )

Set  $\text{Min} = \text{Bin}(b_j)$ ; *// find the best smoothing pattern*

} *// end for j*

**Step1.6: Add**  $C$  to  $L_{COVER}$ ;

**Step1.7: Add**  $S$  to  $L_{STEGO}$ ;

**Step1.8: Add**  $\text{Min}$  to  $L_{PATTERNS}$ ;

} *// end if  $|C - S|$*

**Else** {

**Step1.1: Add**  $C$  to  $L_{COVER}$ ;

**Step1.2: Add**  $S$  to  $L_{STEGO}$ ;

**Step1.3: Add**  $S$  to  $L_{PATTERNS}$ ;

```

    } // end else
} // End of Smooth_Single_Component(.) function

```

Figure 7 explains the implementation part of hiding and smoothing techniques. It appears that we have two steps in the following example, the first step concerns to hide the secret message to cover-image using modified LSB (MDLSB) while the second step represent the smoothing process using ABC algorithm to enhance data security by reducing a noise.

The result of example shows the effect of smoothing using ABC to enhance the security, it appears that the difference between  $I_{c_{pixel}}$  and  $I_{s_{pixel}}$  equal 66255 while the difference between  $I_{c_{pixel}}$  and  $I_{s_{pixel}^{Smoothing}}$  after perform smoothing equal 12863; it means that the noise is damped.

The proposed (Randomization Key) has been described in the following algorithm steps:

**Algorithm5:Char Rand\_Sk() // Cipher key Generation**

```

{
    Step 1: Let Sk be input key of 10 digit
    Step 2: Let n be the numb of DHD levels
    Step 3: Let  $C^i$  be 2 digit integer value where i is the level value
    Step 4: For (inti =1; i<= n; i++)
    {
        Step4.1:  $C^i$  = Sub integer from SK[i, (i+1) mod 10 ]
        Step4.2:  $Sk_i = C^i$ 
    } // end for i
} // endAlgorithm5

```

The proposed (Location Parameters) has been described in the following algorithm steps:

**Algorithm6:Location Rand\_Loc() // Find location**

```

{
    Step1: Input the initial value of one seed Loci with i=0.
    Step2: Input prime number for the incremental constant C0.
    Step3: Input the multiplier constant C such that C-1 must be prime number.
    Step4: Input the value of Modulus m.
    Step5: Calculate the value of Loci+1 by using the following formula:
         $Loc_{i+1} = (C \times Loc_i + C_0) \bmod m$  (12)
    Step6: If  $(Loc_{i+1} \neq 0)$  and  $(Loc_{i+1} \text{ not occur before } i)$  then goto Step7.
        Else goto Step 8.
    Step7:  $i=i+1$  and then goto Step5.
} // endAlgorithm6

```

The time analysis and time complexity measures are studied on the proposed algorithm DHA-ABC. The overall (DHD) algorithm complexity calculation can be defined as:

$$\left. \begin{aligned} \text{Time (DHA - ABC)} &= O(6 + 4n - 4 + 4 + 20n) \\ &= O(6 + 4n + 20n) \\ &\approx O(n), \text{ where } n > N \end{aligned} \right\} \quad (13)$$

## 5. Deep Extracting Algorithm (DEA)

The stego-image is received and the required keys (Sk, Loc, Ck) are used to extract the next image, then decrypt and decompress and obtain the next stego-image. These steps are applied until the required confidential message is reached, see Figure 8.

The proposed (DEA) has been described in the following algorithm steps :

**Algorithm7: Image DE( $ECI_s$ )// Extracting**

```

{
    /*
    Let n be number of levels
    Let Sm be secret message
    Let  $I_s$  be stego-image
    Let CSm be compressed secret message
    Let ECSm be encrypt compressed secret message
    Let  $CI_s$  be compressed stego-image
    Let  $ECI_s$  be encrypt compressed stego-image
    */
    Step1: Input n,  $Ck^1, I_s^n$ ;
    Step4: Find Set of locations  $Loc^1_s$ ;
    Step5: SetNbpp = 4;
    Step6: For (inti=n; i>1; i--)
    {
        Step6.1: Extract  $ECI_s^{i-1}$  form  $I_s^n$ ;
        Step6.2: Extract  $CI_s^{i-1}$  such that;
             $CI_s^{i-1} = \text{Deny}(ECI_s^{i-1}, LCIs)$ ; // see (Definition 8)
        Step6.3: Find Set of location  $Loc^i_s$ ;
        Step6.4: Find  $I_s^{i-1}$  at the locations  $Loc^i_s$  such that
             $I_s^{i-1} = \text{DComp}(CI_s^{i-1}, \hat{U}_m, \hat{U}_s, \tilde{I}_m)$ ; // See (Definition 6)
    }
    Step9: End; //For each
    Step10: Let i = 1
    Step11: Extract ECSm form  $I_s^n$ ;
    Step12: Extract CSm;
    CSm = Deny(ECSm, LCSm); // see (Definition 8)
    Step13: Find Sm such that
    Sm = DComp(CSm,  $\hat{U}_m, \hat{U}_s, \tilde{I}_m$ ); // See (Definition 6)
    Step14: End; // Algorithm7

```

The time analysis and time complexity measures are studied on the proposed (DEA). The overall complexity measures is defined as the following:

$$\left. \begin{aligned} \text{Time (DE)} &= O(3 + 5n - 5 + 4) \\ &= O(2 + 5n) \\ &\approx O(n) \end{aligned} \right\} \quad (14)$$

## 6. Results and Discussions

The experimental results have been discussed in this chapter using gray and color images from the (UCID v2) database; the dataset from the database consisting of 6 colored images has been constructed, this dataset includes the most published image in the previous work. The selected dataset covers different parameters that have been shown below:

### 6.1 Size

The dataset covers different sizes of images from very small scales (150×150) to a high-density scale of (1080×1024). This variation was needed to test the efficiency of data hiding and the amount of payload for each scale.

### 6.2 Color intensity

Colors distribution in images is a vital factor in the proposed algorithm since MDLSB is improved (see Algorithms 2 and 3) to calculate a number of bits per pixel (Nbpp) at each segment with least effect over the whole image. Moreover, we calculate a standard deviation of each segment at each color component to specify data hiding capacity. Therefore









the dataset must contain different colors intensity to cover all circumstances of the hiding process.

### 6.3 Resolution

Image resolution factor plays a main role in the proposed algorithm, where the high resolution of the image is important to extract the noise on the stego-image. So, the resolution factor has been included in the test to evaluate the performance of the algorithm. Table 1 display some of bitmap images (.bmp) with the different sizes are used in the evaluation the performance of the proposed algorithm.

**Table 1.** List of images same type (.bmp) used for testing the proposed approach

No	Name	Image	Size
1	People		431×359
2	Tulips		1024×768
3	Koala		1024×768
4	People2		276×183
5	Onion		198×135
6	Lena		512×512

### 6.4 Comparisons with other techniques

In order to emphasize the strength of the proposed algorithm, many tests over the set of images in the dataset have been implemented and compared with the other algorithms under the same condition. Table 2 shows some of the results based on the three metrics, these are the mean square error (MSE), Signal-to-noise ratio (SNR) and Peak signal-to-noise ratio (PSNR) define in Eqs. (15-17), these metrics are measured to compare between the proposed algorithm MDLSB and the selected least significant bit algorithm (SLSB).

$$\text{PSNR} = 10 \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}} \right) \quad (15)$$

$$\text{MSE} = \frac{\text{MSE}_R + \text{MSE}_G + \text{MSE}_B}{3} \quad (16)$$

$$\text{MSE}_c = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (\text{Ic}_{ij}^c - \text{Is}_{ij}^c)^2 \quad ; c \in \{R, G, B\} \quad (17)$$

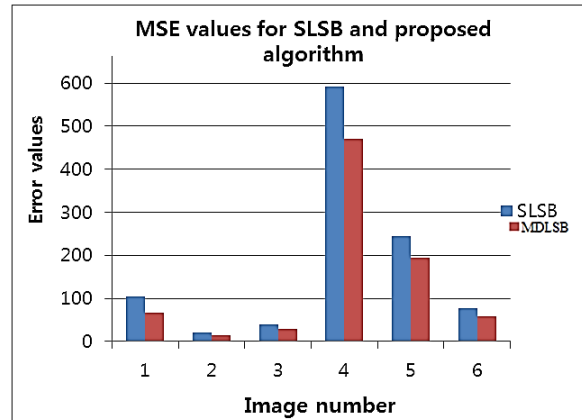
where  $(m \times n)$  is the size of image, and  $\text{Ic}^c$ ,  $\text{Is}^c$  are two bytes of the color  $c$  at the location  $(i, j)$  from  $\text{Ic}$  and  $\text{Is}$  respectively.

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \frac{\mu}{\sigma} \quad (18)$$

where  $\mu$  is a mean of pixels and  $\sigma$  is a standard deviation of pixels.

Results of three metrics appeared that the proposed algorithm for the four stego-images is better than the SLSB algorithm. In addition, Table 2 illustrates the analysis side represented by the histogram to explain the distribution of data over the color intensity and clarifies the enhancement of the proposed model over the standard SLSB. Moreover, the histograms Figs(8-10) show the massive reduction of changes over the original pixels compared to the original one for the proposed model, and it views the strong points of the proposed algorithm.

The experimental results of three metrics (MSE, SNR, and PSNR) show that the proposed method to optimize the stego-image is better than SLSB by approximately 47.25%, 32.33% and 35.35%, respectively for People1 image and is better by approximately 22.53%, 32.34% and 30.18%, respectively for Tulips image and is better by approximately 47.38%, 30.18% and 33.80%, respectively for Lena image and is better by approximately 26.33%, 34.38% and 38.74%, respectively for Onion image. In addition, Figure 9 shows the value of MSE for six images, it appears that the average of MSE using the proposed algorithm with the red legend be 83.68 which considered is accepted error for the image full capacity hiding and it is better than the SLSB algorithm with the blue legend that has the average of MSE be 111.95. Moreover, the maximum error has been shown in the image People2 that includes a large number of colors, while the minimum error appears at the image Tulips that have a little number of colors.



**Figure 9.** MSE comparison

Figure 10 illustrates to measure SNR metric to define the ratio of signal power to the noise power. The proposed algorithm with the red legend is better than the traditional SLSB with the blue legend. Results appear that the proposed algorithm has the maximum SNR among the six images and the average of SNR of the proposed algorithm is 42.5986% dB which is considered accepted results compared with SLSB which has the average of SNR is 24.3336%.

PSNR metric is measured to justify that the system capability in terms of noise reduction. Figure 11 shows the PSNR over the six images, where the results appear that the proposed algorithm has the maximum PSNR among the six images and the average of PSNR of the proposed algorithms 45.6784 dB which is considered accepted results compared to another algorithm in terms of full image capacity usage

and it is better than SLSB which has average of PSNR 29.1059.

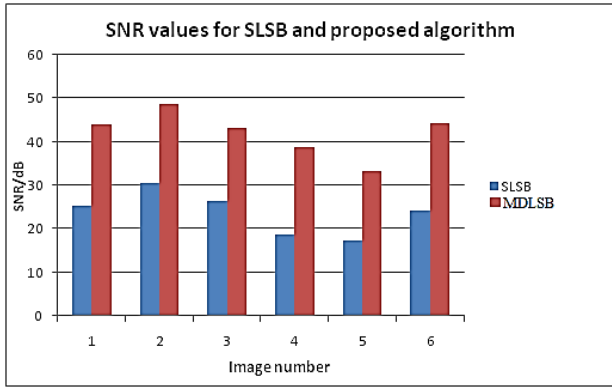


Figure 10. SNR comparison

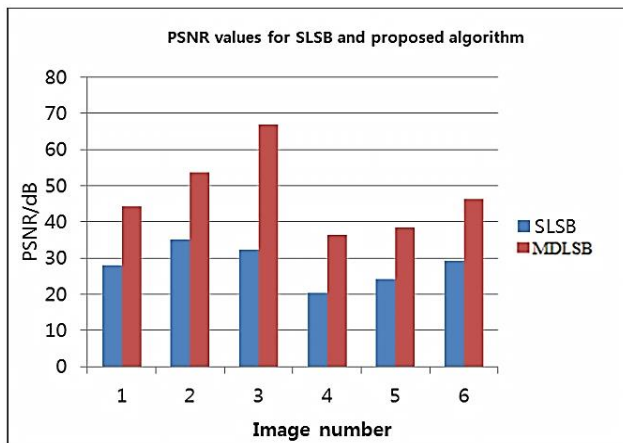


Figure 11. PSNR comparison

### 6.5 Comparison with previous works

The effect of payloads capacity (size of the data hiding in bits) is presented in this section to measure the image distortion using PSNR metric. Table 3 shows the performance of the MDLSB algorithm over the algorithm proposed in (Ou, B., et al., 2015) [21], where this study has been implemented over four test colored images with size of (512×512) are shown in Figure 12.



Figure 12. Four colored images with size of (512×512)

Other comparisons are performed to evaluate the proposed MDLSB algorithm with previous works using four images with five payloads for each image. Table 3 shows the efficiency of the proposed steganography algorithm over the other algorithms.

Furthermore, four testing images displayed in Figure 11 are used to calculate the PSNR metric and compared the proposed hiding algorithm MDLSB with the others algorithm published in (Ou, B., et al., 2015) [21] and (Li, J., et al., 2013) [17]. It appears that the results of the proposed scheme for the minimum payload capacity equal (2×104) bits are better than the other algorithms mentioned above by approximately 3.34%, and 5.74% respectively for Lena image, and better by approximately 4.05%, and 3.54% respectively for Baboon image, and better by approximately 1.89%, and 5.37% respectively for Barbara image, and

better by approximately 3.43%, and 5.39% respectively for Peppers image. Whereas the results of the proposed scheme of the maximum payload capacity equal (15×104) bits for the Lena image, and the proposed algorithm is better than the first algorithm mentioned above by approximately 6.59%, and the maximum payload capacity equal (5.6×104) bits for the Baboon image, the proposed algorithm is better than the other algorithm mentioned above by approximately 6.74%, and 6.55% respectively, and the maximum payload capacity equal (12.5×104) bits for Barbara image, the proposed algorithm is better than the other algorithm mentioned above by approximately 6.75%, and 6.75% respectively, and the maximum payload capacity equal (10.5×104) bits for Peppers image, the proposed algorithm is better than the first algorithm mentioned above by approximately 8.37%.

Table 3. Performance comparison between the proposed algorithm and other embedding algorithms

Images with size (512512)	Payload capacity (bits)×104	PSNR(dB) using Ou, B.,et al., (2015)	PSNR(dB) using Li, J., (2013)	PSNR(dB) using the proposed MDLSB
Lena	2	60.6	59.1	62.7
	6	55.3	53.8	57.3
	8	53.8	52.5	55.6
	12	51.2	50.4	54.8
	15	49.6	N/A	53.1
Baboon	2	56.8	57.1	59.2
	2.8	54.9	55.2	56.1
	3.6	53.3	53.3	54.9
	4.4	51.9	51.9	53.2
	5.6	49.8	49.9	53.4
Barbara	2	62	59.8	63.2
	5	57	55.8	59.6
	7	55.1	54.1	58.5
	10	53	52.5	55.9
	12.5	51.1	51.1	54.8
Peppers	2	59.1	57.9	61.2
	4	55.5	53.7	59.1
	6	53.1	51.8	55.3
	8	51.3	50.2	53.7
	10.5	49.2	N/A	53.7

Moreover, the proposed deep hiding mechanism has been implemented to reduce the probabilities of extraction to (1/n) therefore; this approach is leading to reach higher level of security over other proposed technique using MLS or SLS. Another major enhancement factor of the proposed MDLSB is the non-uniform hiding criteria at each level and at each segment at the last level.

Extra measure for agreement image property is Mean Structural Similarity (MSSIM) which appears to approximate the observed visual quality of an image as represented in Eq.(19). SIMM metric takes values in the range [0, 1] and the state of image quality is increased if SIMM tends to one[6].

$$\text{SIMM}(I_c, I_s) = \frac{(2\mu_{I_c}\mu_{I_s} + ((2^{24}-1)*0.01)^2)(2\sigma_{I_c, I_s} + ((2^{24}-1)*0.03)^2)}{(\mu_{I_c}^2 + \mu_{I_s}^2 + ((2^{24}-1)*0.01)^2)(\sigma_{I_c}^2 + \sigma_{I_s}^2 + ((2^{24}-1)*0.03)^2)} \quad (19)$$

where  $\mu_{I_c}, \mu_{I_s}$  are the mean of cover and stegoimages respectively,  $\sigma_{I_c, I_s}$  is covariance of cover and stegoimages

and  $\sigma_{I_c}^2$ ,  $\sigma_{I_s}^2$  are the variance of a cover and a stego images respectively.

Table 4 described the comparative the visual quality of the stego-images by using four payload capacities (10%, 30%, 40%, and 50%). The quality of stego-images is estimated by SSIM metric to confirm the enforcement of the proposed algorithm over conventional existing references (Geetha et al., 2011)[11] and (EL-Emam, 2013) [6]. In this research, 400 images have been chosen from (UCID v2) database by size (384x512); all these images are changed to the grayscale images.

**Table 4.** the average values of SSIM for different hiding algorithm

Payload capacity	SSIM using Geetha et al., 2011	SSIM using EL-Emam, 2013	SSIM using the proposed algorithm DHA-ABC
10%	1	0.9999	0.9997
30%	0.9997	0.9998	0.9998
40%	0.9995	0.9996	0.9997
50%	0.9992	0.9995	0.9996

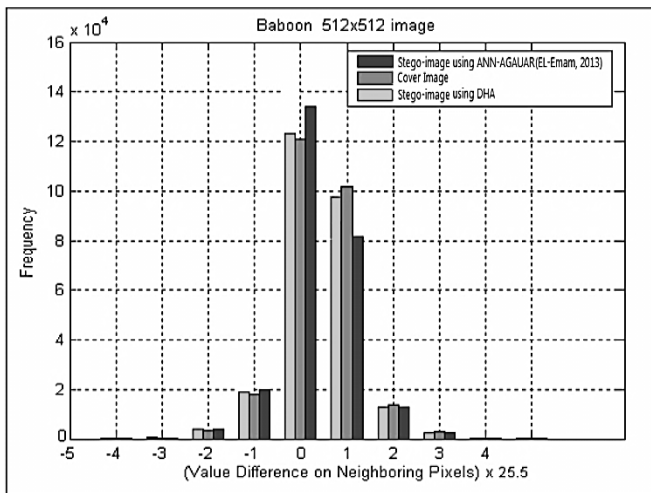
It appears that the proposed algorithm DHA-ABC is working well and better than the other algorithm for the large payload.

### 6.6 Dissimilarity between the adjacent pixels.

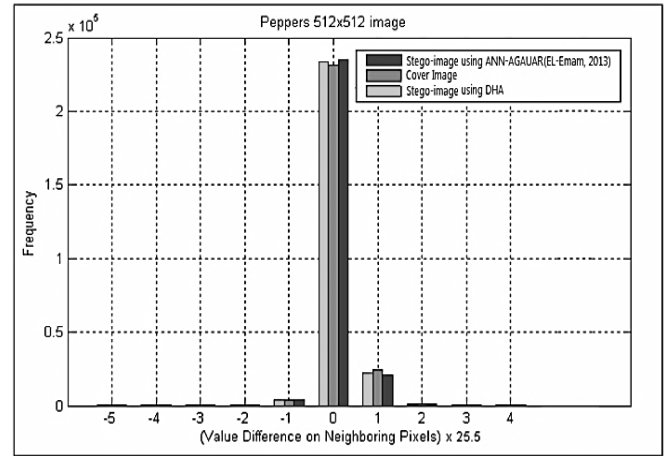
The values  $d_{i,j}^c$  and  $d_{i,j}^s$  are expressed by the difference of the horizontal neighboring pair for cover and stego images respectively. Using Eq.(20) [6] to compute a difference between the adjacent pixels for each image.

$$d_{i,j}^c = P_{i,j}^c - P_{i,j+1}^c, \quad d_{i,j}^s = P_{i,j}^s - P_{i,j+1}^s, \quad \forall i, j (20)$$

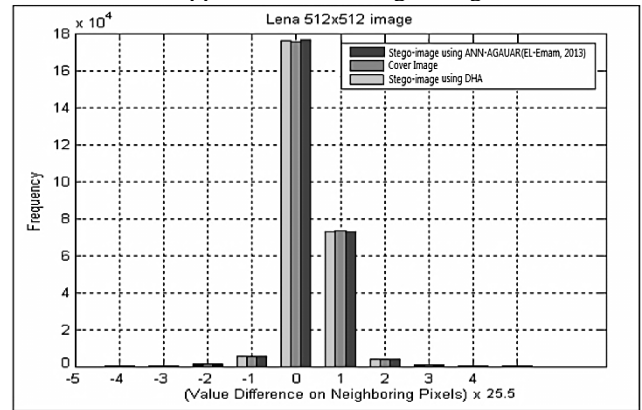
where  $P_{ij}^c$  and  $P_{ij}^s$  are two pixels values at the location (i,j) for both cover and stego images respectively. Results' comparisons of variations of adjacent pixels are shown in Figures (13-15). We perceived that the dissimilarity value between cover and stego-images of the proposed approach is better than previous work published in [6].



**Figure 13.** Value difference on neighboring pixels for Baboon cover and stego-images



**Figure 14.** Value difference on neighboring pixels for Peppers cover and stego-images



**Figure 15.** Value difference on neighboring pixels for Lena cover and stego-images

Results show that lengths of the four images are measured separately; it displays that the least norm is reached when the suggested algorithm using DHA-ABC is executed. Furthermore, we observe that the dissimilarity value using the learning system (ANN-AGAUAR) [6] is certainly greater than the dissimilarity value using DHA-ABC algorithm.

### 6.7 Using Euclidean norm test

Euclidean norm Eq. (21) has been applied to find the distance (d) between cover and stego images. This measure is applied to confirm that we are working against visual attack. Experimental inspections are performed on color images using existing algorithms [6] and the proposed algorithm DHA-ABC, see Figures( 16-18).

$$d = \sqrt{(R_c - R_s)^2 + (G_c - G_s)^2 + (B_c - B_s)^2} \quad (21)$$

Three-color images (RGB format) with the size (512 X 512) are used in this test to show the achievement of the proposed hiding algorithm. Obviously, the smallest distance has arrived when the proposed algorithm is implemented. Moreover, we show that the maximum difference is equal to 70 between the proposed algorithm and the previous [6] at the image Baboon and Peppers with payload percentage equal to 40% and at the image Lena with payload percentage equal to 30%. Whereas the minimum difference is equal to 15 between the proposed algorithm and the previous work at the images Peppers with payload percentage equal to 30%.

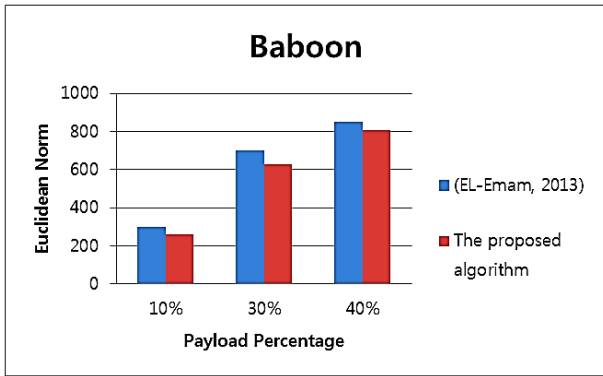


Figure 16. Euclidean norm testing of Baboon color image

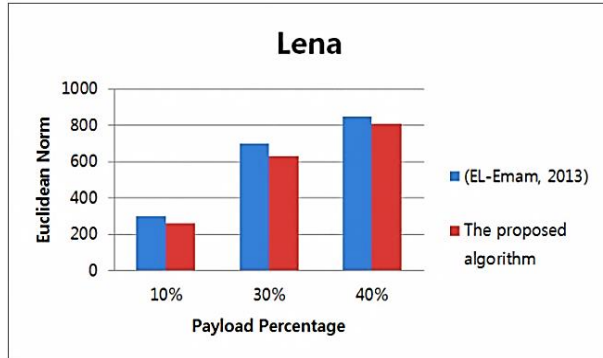


Figure 17. Euclidean norm testing of Lena color image

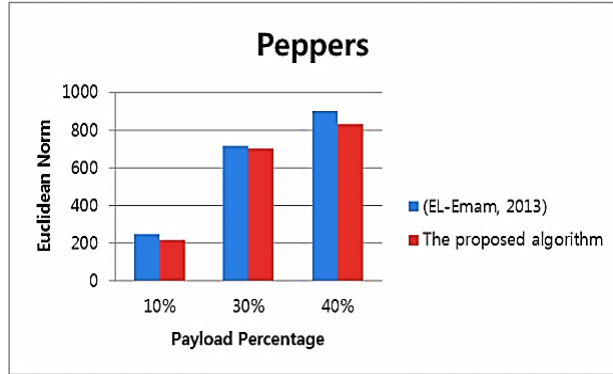


Figure 18. Euclidean norm testing of Peppers color image

### 6.8 The performance of the proposed approach against steganalysers.

The main goal of the proposed steganography algorithm is hiding a secret message in the color image without making doubt that the stego-image contains secret data. In this paper, the stego-image generated by the proposed algorithm DHA-ABC has been tested against WFLoSv attacker using the "Receiver Operating Characteristic" (ROC) curve (Shojaei-Hashemi, A., et al., 2011) [23]. The ROC curve is based on two parameters, the probability of false alarms ( $P_{FA}$ ) and the probability of detections ( $1 - P_{MD}$ ), see Eq. (22). In the ROC scheme  $P_E$ ,  $P_{FA}$  is considered on the horizontal axis, while ( $1 - P_{MD}$ ) is considered on the vertical axis. A steganography technique is said to be absolutely secure with regard to a steganalyser if the following condition is satisfied:

$$|P_{FA} - 1 + P_{MD}| = \varepsilon, \text{ and } \varepsilon \rightarrow 0$$

It means that the area under a curve  $AUC=0.5$ , while the perfect detection of a steganalyser is found when  $\varepsilon \rightarrow 1$ ; see (Li, B et al., 2011).

$$\left. \begin{aligned} P_{FA} &= \frac{N_{Ic}(Is)}{N_{Ic}} \\ P_{MD} &= \frac{N_{Is}(Ic)}{N_{Is}} \\ P_E &= \min \frac{1}{2} (P_{FA} + P_{MD}) \end{aligned} \right\} \quad (22)$$

where

$N_{Ic}(Is)$  is a number of cover-images recognized as stego-images.

$N_{Ic}$  is a total number of cover-images.

$N_{Is}(Ic)$  is the number of stego-images recognized as cover-images.

$N_{Is}$  is the total number of stego-images.

and  $P_{FA}, P_{MD} \in [0,1]$ .

The results validate that the proposed hiding algorithm with DHA-ABC produces high imperceptibility and works against steganalyser for different payloads; see Figure 19. Moreover, the performance of the present steganography can be accomplished by  $P_E$  that having a small area under the curve (AUC).

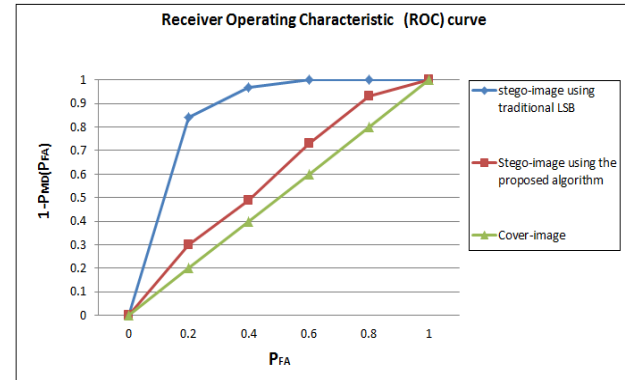


Figure 19. ROC curves of the WFLoSvsteganalyser against the proposed hiding algorithm MDLSB and the traditional LSB with a payload of 40% capacity.

Figure 19 shows the performance of the WFLoSvsteganalyser with a payload of 40% against the proposed steganography and the traditional LSB. It appears that the steganalyser is superior to the traditional LSB, while it has bad detection on the present approach. In addition, the percentage to detect the secret data of the proposed algorithm is not exceeded 15%, whereas the percentage to detect the secret data of the traditional LSB algorithm is approximately 60%.

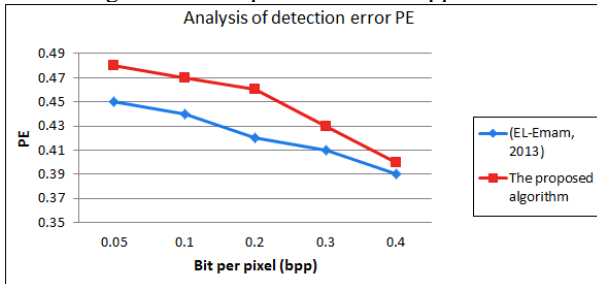
### 6.9 Calculate the detection error $P_E$

Detection error  $P_E$  is represented in Eq.(22) and it is measured as a function of payload. The result of the proposed algorithm has been compared with the previous work [6]. Figure 20 explains the  $P_E$  values belong to the closed interval  $[0, 0.5]$ , where 0.5 refer to the excellent security while zero refers to the excellent detection of the steganography algorithm.

Results illustrate that the average of  $P_E$  for all bpp of the proposed algorithm DHA-ABC is equal to 0.448 and it is better than the previous work [6] by approximately 5.2%



with the excellent security percentage about 89.6%. The result shows that the maximum difference between the values of  $P_E$  for two algorithms is equal 0.04 at the  $\text{bpp}=0.2$ , whereas the minimum difference between the values of  $P_E$  for two algorithms is equal 0.01 at the  $\text{bpp}=0.4$ .



**Figure 20.** Using  $P_E$  as a function of payload to compare proposed algorithm and previous works.

### 6.10 The effect of ABC algorithm

In order to evaluate the effect of the Artificial Bee Colony (ABC) algorithm, four testing color images were implemented using the same MDLSB steganography algorithm with and without using the ABC. The result shown in Table 5 has been proved the capability of the ABC algorithm for enhancing the final result of an MDLSB algorithm and performs smoothing on pixels in terms of distortion values using PSNR.

The experimental results of (PSNR) confirm that the proposed MDLSB with ABC is promising to damp the noise of stego-image, where Table 4 displays the results of PSNR for different payload capacity using four stego-images. The average difference of PSNR around 5dB where the maximum difference equal 6.9 dB that appears in Barbara stego-image for the payload capacity equal  $(4 \times 104)$  bits, while the minimum difference equal 3.3 dB that appears in Baboon stego-image for the payload capacity equal  $(4.4 \times 104)$  bits.

## 7. Conclusions

In this paper, deep hiding/extracting algorithms (DHEA) with modified LSB algorithm (MDLSB) and bee colony algorithm have been proposed to enhance data security. This approach is promising and it confuses attackers to miss understanding how many levels we have and which level include a secret message. The proposed algorithm introduces a non-uniform image segment to obtain a color analysis of each segment and to find payload capacity that should be applied at each pixel to damp the noise produced by hiding a large amount of data. Moreover, randomization of hiding approach in  $n$ th-level has been implemented through two layers; the first layer is applied to move randomly from segment to others, while the second layer is applied to move randomly from pixel to others, where randomization process is introduced to make hard to detect a secret data. The algorithm also uses symmetric encryption to raise the security with a shared key sent on the standalone channel. Artificial bee colony algorithm ABC has been implemented at the  $n$ th-level to perform smoothing on pixels values and to improve the quality of the stego-image to satisfy high imperceptible.

## References

- [1] Al-Shatanawi, O. M., El-Emam, N. N., "A New image steganography algorithm based on MLSB method with random pixels selection," *International Journal of Network Security & Its Applications (IJNSA)*, Vol. 7, No. 2, pp. 21-45, 2015.
- [2] Akhtar, N., Johri, P., Khan, S., "Enhancing the security and quality of LSB based image steganography," *International Conference on Computational Intelligence and Communication Networks, India*, pp. 385-390, 2013.
- [3] Babita, A., Kaur, M., "High capacity filter based steganography," *International Journal of Recent Trends in Engineering*, Vol. 1, No. 1, pp. 672-674, 2009.
- [4] Bawaneh, M., Obeidat A., Al-kofahi, M., "An adaptive FLV steganography approach using simulated annealing," *International Journal of Communication Networks and Information Security (IJCNIS)*, Vol. 10, No. 1, 2018.
- [5] Champakamala, B., Padmini, K., Radhika, D., "Least significant bit algorithm for image steganography," *International Journal of Advanced Computer Technology*, Vol. 3, No. 4, pp. 34-38, 2012.
- [6] El-Emam, N., Al-Zubidy, R., "New steganography algorithm to conceal a large amount of secret message using hybrid adaptive neural networks with modified adaptive genetic algorithm," *Journal of Systems and Software*, Vol. 86, No. 6, pp. 1465-1481, 2013.
- [7] El-Emam, N., "New data-hiding algorithm based on adaptive neural networks with modified particle swarm optimization," *Computers & Security*, Vol. 55, pp. 21-45, 2015.
- [8] El-Emam, N., Qaddoum, K., "Improved steganographic security by applying an irregular image segmentation and hybrid adaptive neural networks with modified ant colony optimization," *International Journal of Network Security & Its Applications*, Vol. 7, No. 5, 2015.
- [9] El-Emam, N., Al-diabat, M., "A novel algorithm for colour image steganography using a new intelligent technique based on three phases," *Applied Soft Computing*, Vol. 37, pp. 830-846, 2015.
- [10] Gaikwad, D., Wagh, S., "Colour image restoration for an effective steganography," *I-manager's Journal on Software Engineering*, Vol. 4, No. 3, pp. 65-71, 2010.
- [11] Geetha, S., Kabilan, V., Chockalingam, S., Kamaraj, N., "Varying radix numeral system based adaptive image steganography," *Information Processing Letters*, Vol. 111, No. 16, pp. 792-797, 2011.
- [12] Jain, R., Kumar, N., "Efficient data hiding scheme using lossless data compression and image steganography," *International Journal of Engineering Science and Technology*, Vol. 4, No. 8, pp. 283-241, 2012.
- [13] Juneja, M., Sandhu, P., "An Improved LSB based steganography technique for RGB color images," *International Journal of Computer and Communication Engineering*, Vol. 2, No. 4, 2013.
- [14] Karim, S. Rahman, M., Hossain, M., "A new approach for LSB based image steganography using secret key," *International conference on computer and information technology (ICCIT2011)*, Bangladesh, pp. 286-291, 2011.
- [15] Khosla, S., Kaur, P., "Secure data hiding technique using video steganography and watermarking," *International Journal of Computer Applications*, Vol. 95, No. 20, 2014.
- [16] Latika, Gulati, Y., "A Comparative study and literature review of image steganography techniques," *International Journal of Science Technology & Engineering*, Vol. 1, No. 10, pp. 238-241, 2015.
- [17] Li, J., Li, X., Yang, B., "Reversible data hiding scheme for color image based on prediction-error expansion and cross-channel correlation," *Signal Processing*, Vol. 93, No. 9, pp. 2748-2758, 2013.



- [18] Liu, T., Tsai, W., "A new Steganographic method for data hiding in Microsoft word documents by a change tracking technique," IEEE Transactions on Information Forensics and Security, Vol. 2, No. 1, pp. 24-30, 2007.
- [19] Medeni, O., Souidi, M., "A generalization of the PVD steganographic method," International Journal of Computer Science and Information Security, Vol. 8, No.2, pp. 156-159, 2010.
- [20] Mstafa, R.; Elleithy, K., "An efficient video steganography algorithm based on BCH Codes," Conference: ASEE American Society for Engineering Education, At: Northeastern University, Boston, MA, USA, pp. 1-10, 2015.
- [21] Ou, B., Li, X., Zhao, Y., Ni, R., "Efficient color image reversible data hiding based on channel-dependent payload partition and adaptive embedding," Signal Processing, Vol. 108, pp. 642-657, 2015.
- [22] Sarkar, A., Madhow, U., Manjunath, B. "Matrix embedding with pseudorandom coefficient selection and error correction for robust and secure steganography," IEEE Transactions on Information Forensics and Security, Vol. 5, No. 2, pp. 225-239, 2010.
- [23] Shojaei-Hashemi, A., Soltanian-Zadeh, H., Ghaemmagham, S., Kamarei, M., "Universal image steganalysis against spatial domain steganography based on energy distribution of singular values," 7th International Conference on Information Technology and Applications (ICITA 2011), Australia, pp. 179-83, 2011.
- [24] Sikarwar, N., "An integrated synchronized protocol for secure information transmission derived from multilevel Steganography and dynamic cryptography," International Journal of Computer Science and Telecommunications, Vol. 3, No. 4, pp. 31-36, 2012.
- [25] Yang, C. H., Weng, C. Y., Wang, S. J., Su, H.M., "Adaptive data hiding in edge areas of images with spatial LSB domain systems," IEEE Transactions on Information Forensics and Security, Vol. 3, No.3, pp. 488-497, 2008.

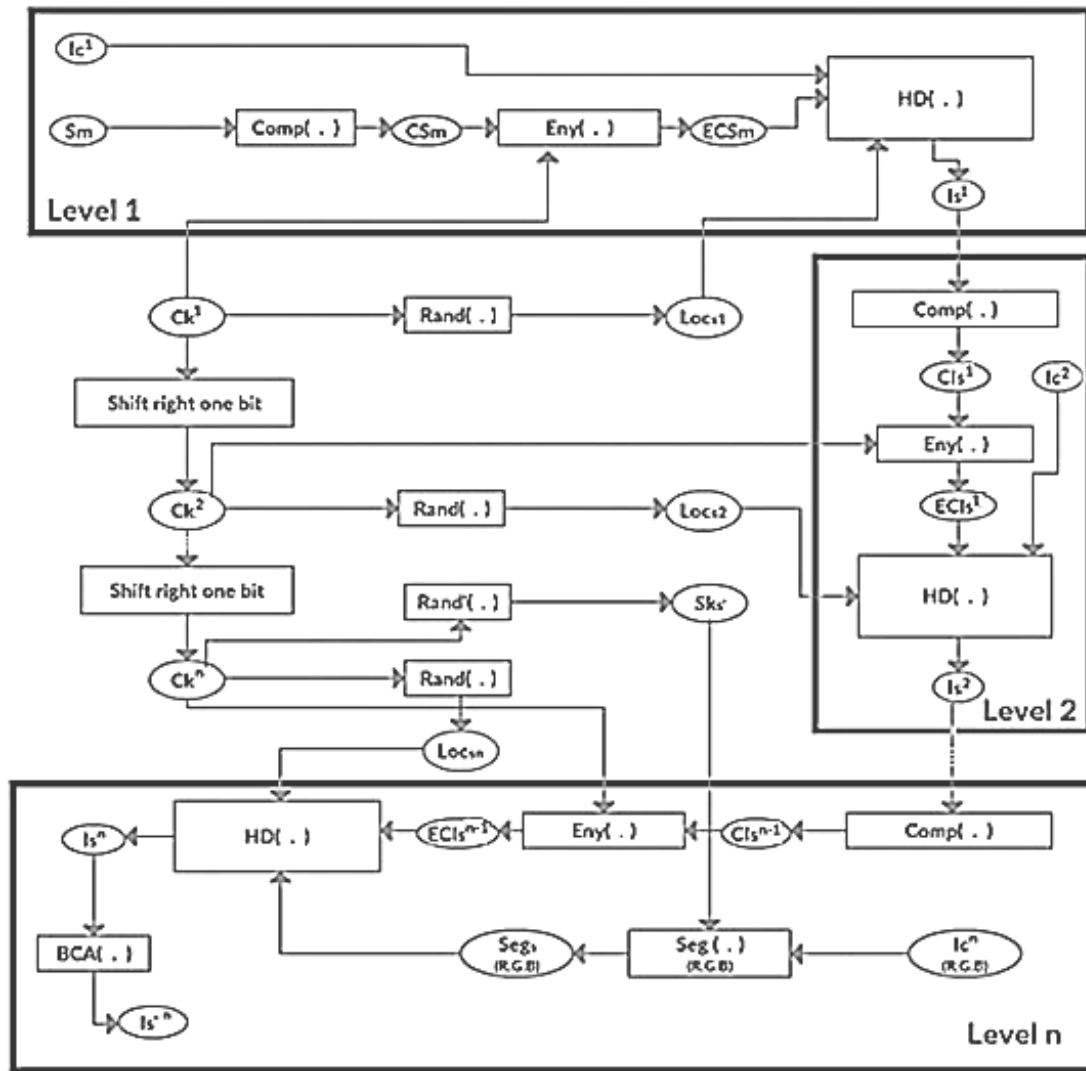


Figure 4. Deep hiding architecture

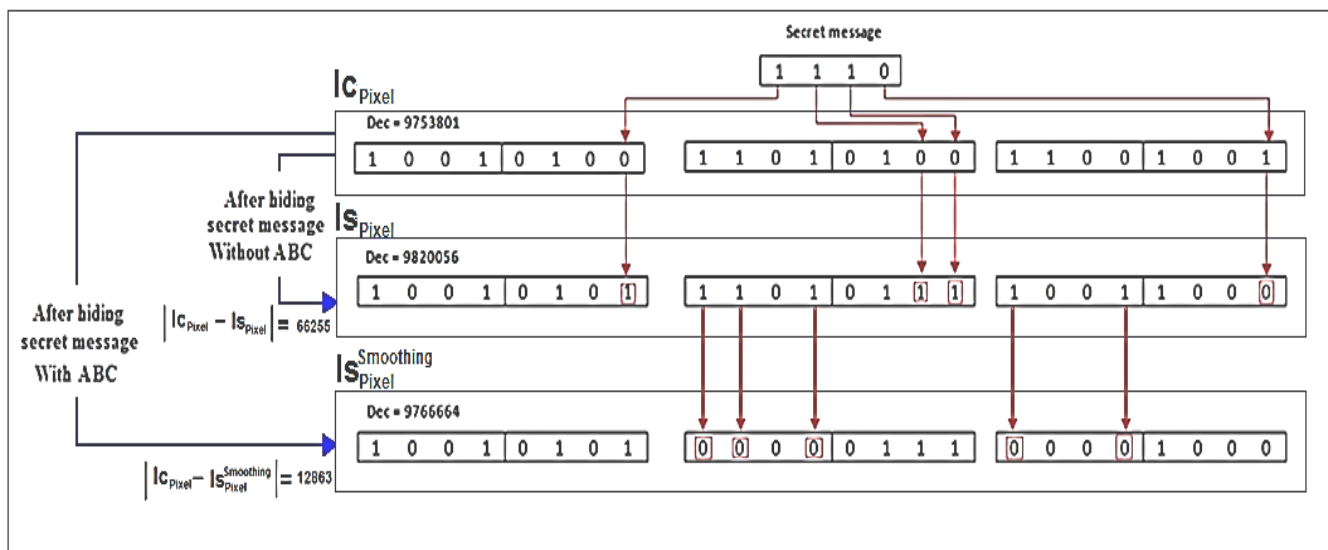


Figure 7. Implementation of ABC algorithm

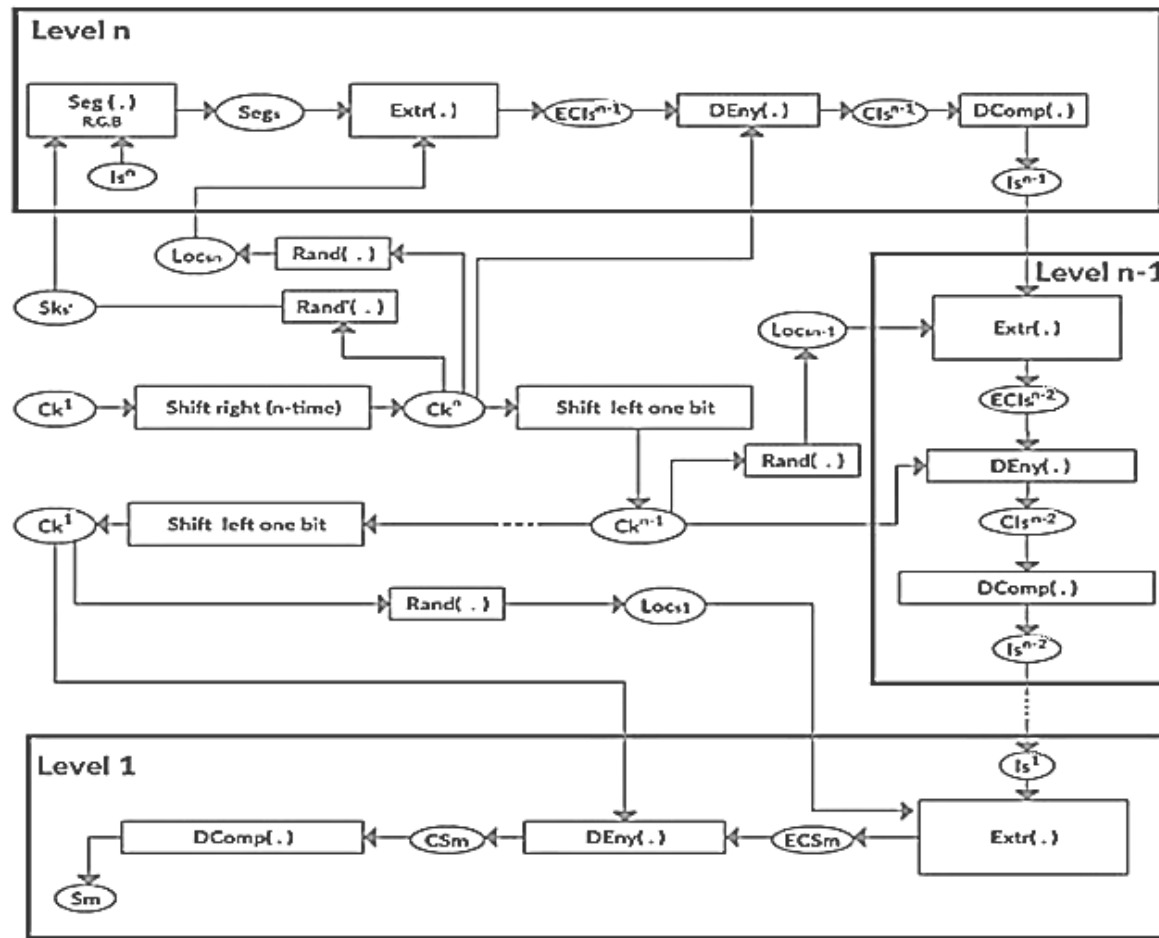


Figure 8. Deep extracting architecture

Table 2. Comparison between proposed algorithm and SLSB

Image name	Traditional SLSB			Proposed algorithm (MDLSB)		
	MSE	SNR	PSNR	MSE	SNR	PSNR
People 1	104.4397	25.3662	27.9421	66.4397	43.9662	44.1421
Tulips	20.5807	30.6276	34.9962	14.3807	48.6376	53.6962
Lena	77.1494	24.1199	29.2575	59.4494	44.4199	46.3975
Onion	245.6408	17.2209	24.2278	194.4408	33.3709	38.4778

**Table 5.** ABC performance algorithm over the proposed MDLSB algorithm

<b>Images 512×512</b>	<b>Payload capacity (bits)×10<sup>4</sup></b>	<b>PSNR(dB) using the proposed MDLSB without ABC</b>	<b>PSNR(dB) using the proposed MDLSB with ABC</b>
<b>Lena</b>	2	58.2	62.7
	6	52.3	57.3
	8	51.1	55.6
	12	49.2	54.8
	15	48.9	53.1
<b>Baboon</b>	2	53.1	59.2
	2.8	51.5	56.1
	3.6	50.2	54.9
	4.4	49.9	53.2
	5.6	48.7	53.4
<b>Barbara</b>	2	58.2	63.2
	5	53.2	59.6
	7	52.8	58.5
	10	51.7	55.9
	12.5	50.6	54.8
<b>Peppers</b>	2	56.4	61.2
	4	52.2	59.1
	6	50.7	55.3
	8	49.2	53.7
	10.5	48.1	53.7