The Journal of Systems and Software 86 (2013) 1465–1481

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

New steganography algorithm to conceal a large amount of secret message using hybrid adaptive neural networks with modified adaptive genetic algorithm☆

Nameer N. El-Emam[a,*], Rasheed Abdul Shaheed AL-Zubidy[b]

[a] Department of Computer Science, Philadelphia University, Jordan
[b] Department of Computer Information System, Philadelphia University, Jordan

ARTICLE INFO

Article history:
Received 6 May 2012
Received in revised form
18 November 2012
Accepted 4 December 2012
Available online 31 December 2012

Keywords:
Adaptive image segmentation
Concealing with high capacity
Concealing with high security
Neural networks
Genetic algorithm
Steganography

ABSTRACT

In this paper, we propose a new steganography algorithm using non-uniform adaptive image segmentation (NUAIS) with an intelligent computing technique to conceal efficiently a large amount of confidential messages (Smsg) into color images. Whereas, the number of secret bits to be replaced is non uniform from byte to another byte; it based on byte characteristics, which are extracted by using 16 byte levels (BL) with variance distribution of the Neighboring Eight Bytes (NEB) around the current byte. Four security layers are introduced to increase resistance against statistical and visual attacks. These layers are designed to make an excellent imperceptible concealing Smsg with lower distortion of a color plane and high protection of Smsg. The proposed intelligent technique using the hybrid adaptive neural networks and modified adaptive genetic algorithm employing uniform adaptive relaxation (ANN_AGAUAR) is working as the fourth security layer to improve the quality of the stego image (I_s). The results are discussed and compared with the previous steganography algorithms; it demonstrates that the proposed algorithm's effectiveness can be concealed efficiently the number of secret bits reached to four bits per byte with better visual quality.

1. Introduction

Steganography algorithm is kind of camouflage that allows to embed the encrypted data into cover media (e.g. Text, image, audio, video, etc.) to generate stego-media. The secret message (Smsg) cannot be seen in the stego-media while it is transmitted on public communication channels in the common types of computer networks. The goal of steganography algorithm is not only to conceal a large amount of secret messages but also to transmit these imperceptible messages; this is done by different techniques and used by many researchers (Filler et al., 2011; Fridrich, 2009; Pevný et al., 2010).

Wang and Moulin (2008) constructed perfectly secure steganographic by embedding the message into the cover-text. The resulting Stego-image (I_S) has exactly the same probability distribution as the cover-text. EL-Emam (2007) proposed an efficient algorithm to hide a large amount of data into the color bitmap image and to work against statistical and visual attacks. Munuera (2007) shown some relations between steganographic algorithms and error-correcting codes, these relations are used to construct good steganographic protocols and deduces their properties from those of the corresponding codes. Sajedi and Jamzad (2010) introduced boosted steganography scheme (BSS) that has a preprocessing stage before applying steganography methods. The goal of BSS is increasing the undetectable of I_S. Qian and Zhang (2012) proposed lossless data hiding method to embedding secret data into JPEG bitstream by Huffman's code mapping. Qu et al. (2010) proposed a novel quantum steganography protocol based on quantum secure direct communication using entanglement swapping of Bell's states, the protocol builds up hidden channel within the improved ping-pong protocol to transmit secret messages. Lee and Chen (2010) proposed a novel data hiding scheme that uses a simple modulus function to address four performance criterion (the embedding capacity, the visual quality of the I_S, the security, and the complexity of the data-embedding algorithm). Lee et al. (2010) presents an adaptive reversible data scheme based on the prediction of difference expansion; this scheme gains from embedding capacity by taking full advantage of the large quantities of smaller

☆ Controversy corner. It is the intention of the Journal of Systems and Software to publish, from time to time, articles cut from a different cloth. This is one such article. The goal of CONTROVERSY CORNER is both to present information and to stimulate thought and discussion. Topics chosen for this coverage are not just traditional formal discussions of research work; they also contain ideas at the fringes of the field's "conventional wisdom". These articles will succeed only to the extent that they stimulate not just thought, but action. If you have a strong reaction to the article that follows, either positive or negative, send it along to your editor, at card@software.org. We will publish the best of the responses as CONTROVERSY REVISITED.

* Corresponding author. Tel.: +962 7 77489308; fax: +962 2 6374444.
E-mail address: nemam@philadelphia.edu.jo (N.N. El-Emam).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.12.006

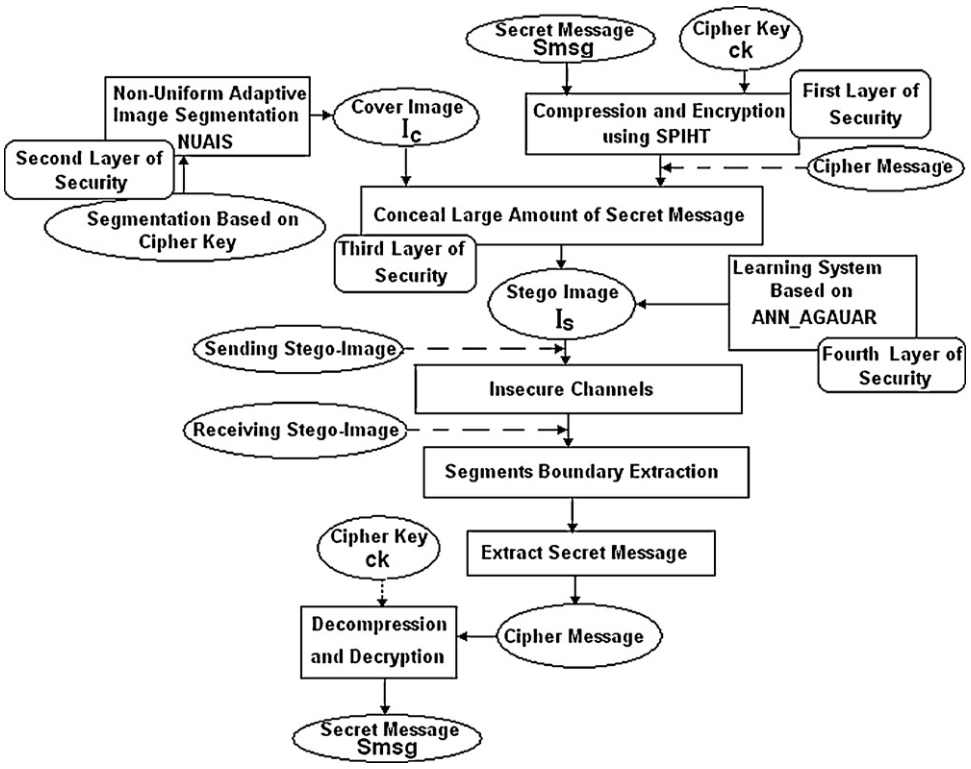

Fig. 1. Steganography with four security layers.

difference values where secret data can be concealed, therefore, several advantages of were gained (1) the location map is no more required, (2) the embedding capacity can be adjusted depending on the practical applications, and (3) the high embedding capacity with minimal visual distortion can be achieved. Wu et al. (2011) proposed a novel secret image sharing scheme by applying optimal pixel adjustment process to enhance the image quality under different payload capacity and various authentication bit's conditions. Phadikar and Maity (2011) proposes a joint data-hiding and data modulation scheme to serve the purpose of quality access control of image(s) using quantization index modulation (QIM).

In the recent years, some researches in the data concealing were using an intelligent algorithm based on soft computing. Such algorithms are used to achieve robust, low cost, optimal and adaptive solutions in data concealing problems. Fuzzy Logic (FL), Rough Sets (RS), Adaptive Neural Networks (ANN), Genetic Algorithms (GA) Support Vector Machine (SVM), Ant Colony, and Practical Swarm Optimizer (PSO) etc. are the various components of soft computing, and each one offers specific attributes (El-Emam and Abdul-Shaheed, 2008). A data concealing scheme using a well-known GA-AMBTC based on genetic algorithm, block truncation code and modification direction techniques were proposed by Chang et al. (2009) to embed secret data into compression codes of color images. Wu and Shih (2006) presents an efficient concept of developing a robust steganographic system by artificially counterfeiting statistic features instead of the traditional strategy of avoiding the change of statistic features. This approach is based on genetic algorithm by adjusting gray values of a cover-image (I_c) while creating the desired statistic features to generate the I_s that can break the inspection of steganalytic systems. Arsalan et al. (2012) developed an intelligent reversible watermarking approach for medical images by using GA to make an optimal tradeoff between imperceptibility and payload through effective selection of threshold. Particle Swarm Optimization algorithm (PSO) was introduced by Li and Wang (2007) to improve the quality of I_s; this is done by deriving an optimal substitution matrix for transforming the secret messages. Zhang et al. (2008) proposed a new method of information-embedding capacity bounds analysis that is based on the neural network theories of attractors and basin of attraction. The development of blind detection algorithms for digital image steganography is reviewed by Luo et al. (2009); this approach is based on image multi-domain features merging and BP (Back-Propagation) neural network. Luo et al. (2010) applied LSB matching revisited image steganography and propose an edge adaptive scheme which can select the embedding regions according to the size of secret message and the difference between two consecutive pixels in the I_c.

This paper proposes a new algorithm of data concealing using hybrid adaptive neural networks with an adaptive genetic algorithm based on a new version of adaptive relaxation named uniform adaptive relaxation ANN_AGAUAR. With this algorithm, a large amount of data can be concealed into a color bitmap image with four layers of security.

The rest of the paper is structured as follows: In Section 2, the proposed steganography with four layers of security is discussed. Phases of the proposed steganography algorithm based on ANN_AGAUAR are appearing in Section 3. In Section 4 the intelligent technique based on adaptive neural networks and adaptive genetic algorithm with uniform adaptive relaxation is discussed. Experimental results are discussed in Section 5. Finally, Section 6 summarizes the main conclusions of the proposed algorithm.

2. Requirements of the new steganography algorithm

A new steganography algorithm using four layers of security has been constructed and developed. The first three layers were proposed in the previous works (EL-Emam, 2007) and in this work, we developed the first three layers and add a fourth layer based on ANN and an adaptive genetic algorithm to acquire high security and to provide unbreakable security wall (see Fig. 1).

Before discussing the proposed steganography algorithm, it is important to show the following requirements:

Fig. 2. Types of image segmentation.

2.1. Compression and encryption requirement

The lossless compression and encryption mechanism for secret messages (Smsg) is applied in the first layer of security using the Set Partitioning In Hierarchical Trees (SPIHT) and Advanced Encryption Standard (AES) algorithms respectively (Chen, 2008; Ali et al., 2011).

2.2. Image segmentation requirement

New non-uniform adaptive image segmentation (NUAIS) is constructed in this work see Section 3.2 to build a second layer of security for the proposed steganography algorithm, and it's based on random segmentation of the I_c into a number of non-uniform segments according to the cipher key (ck) Fig. 2. It appears that non-uniform segmentation is more secure than uniform segmentation to carry the input information due to hard to detect the segments' edges by stegoanalysis.

2.3. Compression of I_s requirement

Lossless image compression based on SPIHT algorithm is implemented on I_s to avoid sending enormous file size..

2.4. Bytes classification requirement

The four categories are proposed in this work for byte classification, it's based on the number of bits per byte (Nbpb), which are used to embed Smsg.

The criterion of byte classification is based on the variance σ^2 of the Neighboring Eight Bytes (NEB) of the specific byte. This measure uses a new concept called the byte level (BL) for each byte in the image (see Definition 3.1). This approach is powerful in the sense of reducing the noise of a I_s and support the third layer of security as shown in Fig. 1.

2.5. Intelligent technique requirements

Hybrid adaptive neural networks and adaptive genetic algorithm employing uniform adaptive relaxation (ANN_AGAUAR) is proposed to work against visual and statistical attacks. This approach is used effectively to support a fourth layer of security see Fig. 1.

3. The proposed steganography scheme

3.1. Phases of steganography algorithm

The proposed steganography algorithm has two phases (data concealing at the sender side and data extracting at the receiver side). These phases are constructed and implemented to reduce the chances of statistical detection and provide robustness against a variety of image manipulation attacks. After concealing data, I_s is produced, which does not have any distortion artifacts. Moreover, the new steganography algorithm should not sacrifice a concealing capacity in order to decrease the perceptible of data concealing.

The first phase is used to conceal a Smsg into an I_c according to the following general steps:

Step 1: Encrypt ck;
Step 2: Perform compressions on the Smsg to increase the amount of data concealing.
Step 3: Perform encryption on the compressed Smsg.
Step 4: Compute non uniform adaptive segmentation of the I_c (according to the ck, see Section 3.2.
Step 5: Scan all bytes of each segment and check byte characteristics to fix the Nbpb that should be used to conceal a part of a Smsg at the right nibble of the byte.
Step 6: Perform concealing process of the Smsg into an I_c.

The second phase is used to extract data from an I_s at the receiver side in conformity with the following steps:

Step 1: Find the edge of non uniform adaptive segments according to the ck.
Step 2: Extract bytes' characteristics for each segment to show Nbpb that hold the secret data.
Step 3: Extract Smsg.
Step 4: Decompress and decrypt the Smsg by using the ck.

3.2. NUAIS algorithm

New algorithm of image segmentations is proposed in this work; it bases on non-uniform adaptive image segmentation of the I_c. This approach is introduced as second security layer to make hard to detect non uniform segment edges by stego-analysis and to work against statistical analysis of Pair of Values (PoVs) by avoiding exchanged during sequential concealing. The proposed algorithm of NUAIS is presented in the following steps:

Step 1: Let S be a length of ck see Eq. (1).

$$S = \left|ck\right| \tag{1}$$

Step 2: Calculate the size of each segment for both vertical and horizontal directions (Ver, Hor) by using the proposal formulas see Eqs. (13) and (14):

$$\mathrm{Ver}_i = \left\lceil \frac{\mathrm{Dec}(ck_i) * h_{I_c}}{\sum_{m=1}^{S} \mathrm{Dec}\,(ck_m)} \right\rceil \quad \forall i = 1, \ldots, S \tag{2}$$

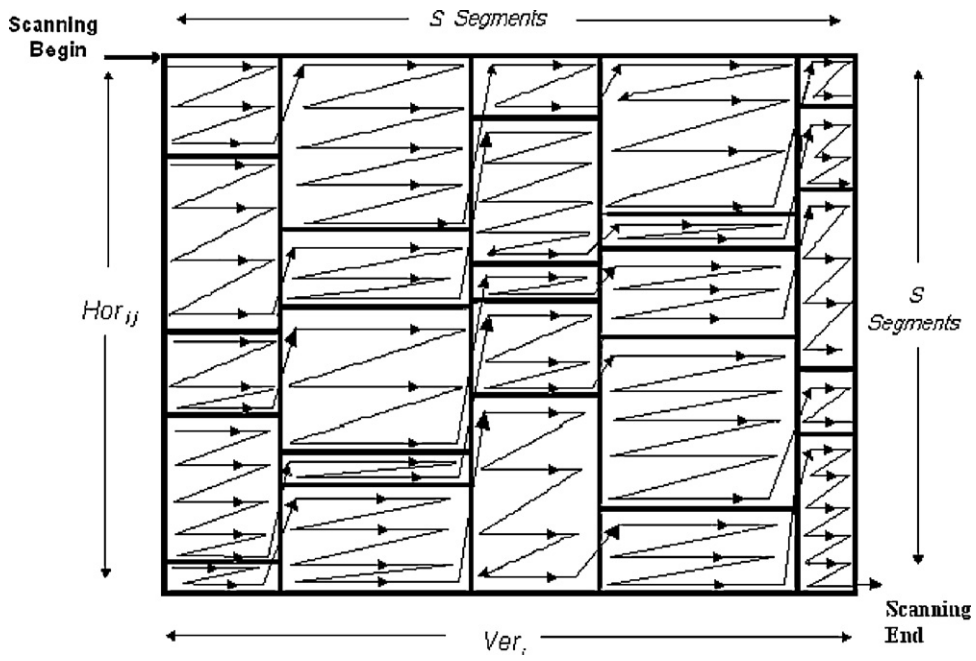

Fig. 3. Scanning pixels on the adaptive image's segments.

$$\mathrm{Hor}_{ij} = \left\lceil \frac{\mathrm{Dec}(ck_k) * w_{I_c}}{\sum_{m=1}^{S} \mathrm{Dec}(ck_m)} \right\rceil ,$$

$$\forall i = 1, \ldots, S, \quad \forall j = 1, \ldots, S, \quad k = (i+j) \bmod S + 1 \tag{3}$$

where $\mathrm{Dec}(ck_i)$ represents the decimal value of the ith character of the ck The h_{I_c} and w_{I_c} are the height and the width of the I_c respectively.

Step 3: Using Ver_i, Hor_{ij} values to find the edge of non-uniform image segmentation on the I_c with the size $(S \times S)$ segments.

Step 4: Apply row wise scanning for both non-uniform segments at the I_c and all pixels at each segment, this process is used to specify byte for concealing secret bits. See Fig. 3.

3.3. The proposed steganography algorithm

As mentioned in Section 2, due to aiming at achieving high concealing efficiency, we propose 16 byte levels (BL) with variance distribution of the NEB around the current byte to study the characteristics of each byte; this process is used to determine the Nbpb.

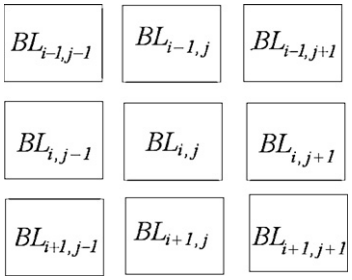

Fig. 4. Byte levels $\mathrm{BL}_{i,j}$ of $NEB_{i,j}$ for the current byte $b^c_{i,j}$.

Definition 3.1. The Byte Level BL factor is an integer value uses to categorize bytes into a number of levels. This factor is working beside other factors to achieve the size of the concealing bits for each byte at the I_c. The value of BL is calculated by using Eq. (4):

$$\mathrm{BL}^c_{i,j} = \left\lfloor \frac{b^c_{i,j}}{16} \right\rfloor + 1 \tag{4}$$

where $b^c_{i,j}$ represents the byte value in a decimal notation at the (i,j) location of the I_c for the color c, where $c \in \{R, G, B\}$

Definition 3.2. Byte characteristics denoted by $Nbpb_{ij}$ is based on the number of secret bits to be replaced for the specific byte $b^c_{i,j}$ at each segment in the color c. The value of $Nbpb_{ij}$ can be derived in Eq. (5) by using variance $\sigma^2_{i,j,c}$ of $NEB_{i,j}$. See Fig. 4.

$$Nbpb^c_{i,j} = \begin{cases} 4 & \text{if } \left(\left(\mathrm{BL}^c_{i,j} = 1\right) \vee \left(\mathrm{fr}(\mathrm{BL}^c_{i,j}) = \underset{\mathrm{BL}^c_{i,j} \cup NEB_{i,j}}{arg\,\min} \left(\mathrm{fr}\left(\mathrm{BL}^c_{i,j}\right)\right) \right) \right) \\ \left\lceil \dfrac{\sigma^2_{i,j,c}\left(\mathrm{BL}^c_{i,j}\right)}{\underset{\mathrm{BL}^c_{i,j} \cup NEB_{i,j}}{arg\,\max} \left(\sigma^2_{i,j,c}\left(\mathrm{BL}^c_{i,j}\right)\right)} \times 3 \right\rceil + 1 & \text{if } \left(\left(\underset{\mathrm{BL}^c_{i,j} \cup NEB_{i,j}}{arg\,\max} \left(\sigma^2_{i,j,c}\left(\mathrm{BL}^c_{i,j}\right)\right) > 0 \right) \wedge \left(\mathrm{fr}(\mathrm{BL}^c_{i,j}) \neq \underset{\mathrm{BL}^c_{i,j} \cup NEB_{i,j}}{arg\,\min} \left(\mathrm{fr}\left(\mathrm{BL}^c_{i,j}\right)\right) \right) \right) \\ 1 & \text{if } \left(\underset{\mathrm{BL}^c_{i,j} \cup NEB_{i,j}}{arg\,\max} \left(\sigma^2_{i,j,c}\left(\mathrm{BL}^c_{i,j}\right)\right) \approx 0 \right) \end{cases} \tag{5}$$

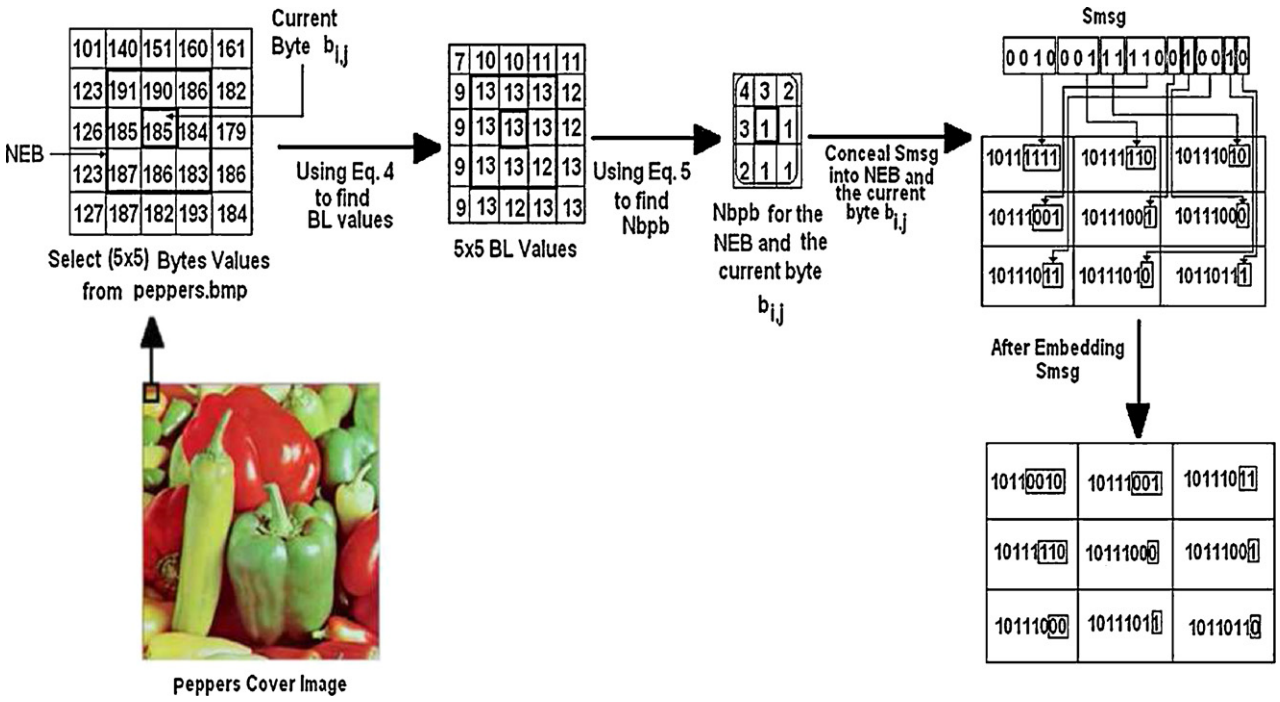

Fig. 5. Calculate $Nbpb_{ij}^{c}$ for the current byte and NEB and then show how to embed the Smsg.

where fr($BL_{i,j}$) is the frequency of $BL_{i,j}$ in the I_c, and the variance $\sigma_{i,j,c}^2$ of the current byte and the $NEB_{i,j}$ see Eq. (6).

$$\sigma_{i,j,c}^2 = \frac{1}{9}\sum_{m=i-1}^{i+1}\sum_{n=j-1}^{j+1}\left(BL_{m,n}^{c} - \mu_{i,j}^{c}\right)^2 \tag{6}$$

and the mean μ of NEB is defined in Eq. (7):

$$\mu_{i,j}^{c} = \frac{1}{9}\sum_{m=i-1}^{i+1}\sum_{n=j-1}^{j+1} BL_{m,n}^{c} \tag{7}$$

It appears that the concealing capacity is controlled by the integer variable named Nbpb for each byte in a I_c, where the value of $Nbpb \in [1,4]$ and it is based on the ratio between the variance of the current $BL_{i,j}$ and the maximum variance of NEB, see Eq. (5).

Example 3.1. Assume we select window with the size (5×5) from the Peppers color image and we need to show step by step how to find BL and how to used it to find a number of secret bits to be concealed and then show the concealing process of the input Smsg into NEB and $b_{i,j}^{c}$. See Fig. 5.

Definition 3.3. Concealing function Conceal($Smsg_{i,j}$, $b_{i,j}^{c}$, $Nbpb_{i,j}^{c}$) is used to embed a Smsg into the current byte at the I_c. This function requires three parameters such as, $Smsg_{i,j}$, $b_{i,j}^{c}$,and $Nbpb_{ij}^{c}$, where $Smsg_{i,j}$ is a part of Smsg and the following condition should be satisfied $\left|smsg_{i,j}\right| = Nbpb_{i,j}^{c}$

3.4. The computation time of concealing and extraction algorithms

Computation time looks up to the amount of time needed to embed or to extract a Smsg with length L bits into/from RGB image. In this work, it can be observed that for an image of size $(n \times n)$ pixels with 24 bits for each pixels and the number of segments of in color image is equal to $(3 \times S \times S)$ segments. The time complexity of computing using the proposed algorithm to conceal (extraction as well) Smsg into/from an color image is shown in Tables 1 and 2.

It can be noted that the worst case of the complexity order for concealing or extracting algorithms equal to $O(n^2) + O(S^2) \rightarrow O(n^2)$ where $n > S$; while the best case of the complexity order for each algorithm is equal to $O(L)$ where $L = |Smsg|$.

An excellent time analysis is measured by using the following aspects:

Aspect 1: Most steganography schemes are time consuming, because they scan the host image more than once, while, in the proposed algorithm, the concealing process into the current byte is implemented immediately after scanning process, and the scanning image is performed once.

Aspect 2: The trained ANN model is needed on the concealing side only and using a new modification of the genetic algorithm abbreviated by AGAUAR to reduce the number of iterations in the training process, see Section 4.1.

4. Intelligent technique

The proposed intelligent technique based on hybrid adaptive neural networks ANN and adaptive genetic algorithm AGAUAR represents the fourth security layer. This layer is introduced to support and enhanced steganography algorithm and produce an excellent imperceptible of I_s by working effectively against statistical and visual attacks.

4.1. Proposed ANN_AGAUAR architecture

The proposed intelligent technique ANN_AGAUAR includes $(n–p–n)$ Perceptron layers' architecture. It has n neurons in input layer, p neurons in the hidden layer and n neurons in the output layer with full connection (Fig. 6).

The solid arrow in Fig. 6 shows two kinds of transitions; one of them is many-to-one while the other is one to many transitions among Perceptron layers, whereas dotted arrow refers to one-to-one transition, and the dashed arrow shows the sending

Table 1
Concealing algorithm with the account of its time analysis.

Concealing algorithm	Time analysis
Apply NUAIS Algorithm to find edges of each segment.	$\sum_{j=1}^{S}\sum_{i=1}^{S} C1_{ij} \rightarrow O(S^2)$ *where S is the length of ck*
Foreach (Color $c \in \{R, G, B\}$) { //the time analysis $\rightarrow \sum_{c=1}^{3}$	$O\left(\sum_{c=1}^{3}\sum_{s=1}^{S^2}\sum_{b=1}^{\lfloor (n/S)^2 \rfloor}(C2+C3+C4+C5+C6+C7+C8)\right)$
Foreach (Segment $S \in I_c$) { //the time analysis $\rightarrow \sum_{s=1}^{S^2}$	$\rightarrow O\left(3 \times S^2 \times \left(\lfloor \frac{n}{S} \rfloor\right)^2\right)$
Foreach(Byte $b_{i,j}^c \in S$) { //the time analysis $\rightarrow \sum_{b=1}^{\lfloor (n/S)^2 \rfloor}$	$\rightarrow O(n^2)$
Compute $(BL_{i,j}^c, \mu_{i,j}^c, \sigma_{i,j,c}^2, fr(BL_{i,j}^c), \min arg(fr(BL_{i,j}^c))$; //using Eqs. (4)–(7) and the time analysis $\rightarrow C2+C3+C4+C5+C6$	
Compute $Nbpb_{i,j}^c$; //using Eq. (5) and the time analysis $\rightarrow C7$	
Conceal $(Smsg_{i,j}, b_{i,j}^c, Nbpb_{i,j}^c)$; //the time analysis $\rightarrow$ C8 and the worst case $\rightarrow 4$	
}	
}	
}	

Where $C_i \forall i = 1, \ldots, 8$ are the amount of time to complete the corresponding operations.

Table 2
Extracting algorithm with the account of its time analysis.

Extraction algorithm	Time analysis
Apply NUAIS Algorithm to find edges of each segment.	$\sum_{j=1}^{S}\sum_{i=1}^{S} C1_{ij} \rightarrow O(S^2)$ *where S is the length of ck*
Foreach (Color $c \in \{R, G, B\}$) { //the time analysis $\rightarrow \sum_{c=1}^{3}$	$O\left(\sum_{c=1}^{3}\sum_{s=1}^{S^2}\sum_{b=1}^{\lfloor (n/S)^2 \rfloor}(C2+C3+C4+C5+C6+C7+C8)\right)$
Foreach (Segment $S \in I_c$) { //the time analysis $\rightarrow \sum_{s=1}^{S^2}$	$\rightarrow O\left(3 \times S^2 \times \left(\lfloor \frac{n}{S} \rfloor\right)^2\right)$
Foreach(Byte $b_{i,j}^c \in S$) { //the time analysis $\rightarrow \sum_{b=1}^{\lfloor (n/S)^2 \rfloor}$	$\rightarrow O(n^2)$
Compute $(BL_{i,j}^c, \mu_{i,j}^c, \sigma_{i,j,c}^2, fr(BL_{i,j}^c), \min arg(fr(BL_{i,j}^c))$ //using Eqs. (4)–(7) and the time analysis $\rightarrow C2+C3+C4+C5+C6$	
Compute $Nbpb_{i,j}^c$; //using Eq. (5) and the time analysis $\rightarrow C7$	
Extract $(Smsg_{i,j}, b_{i,j}^c, Nbpb_{i,j}^c)$ //the time analysis $\rightarrow C8$ and the worst case $\rightarrow 4$	
}	
}	
}	

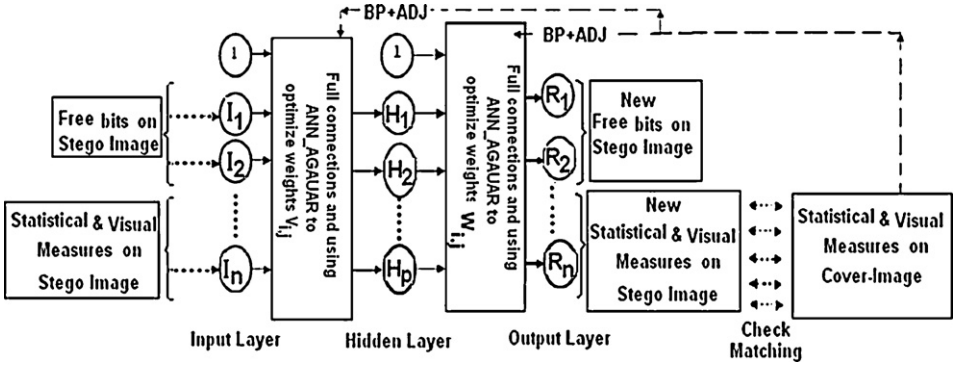

Fig. 6. Three layers of ANN_AGAUAR architecture.

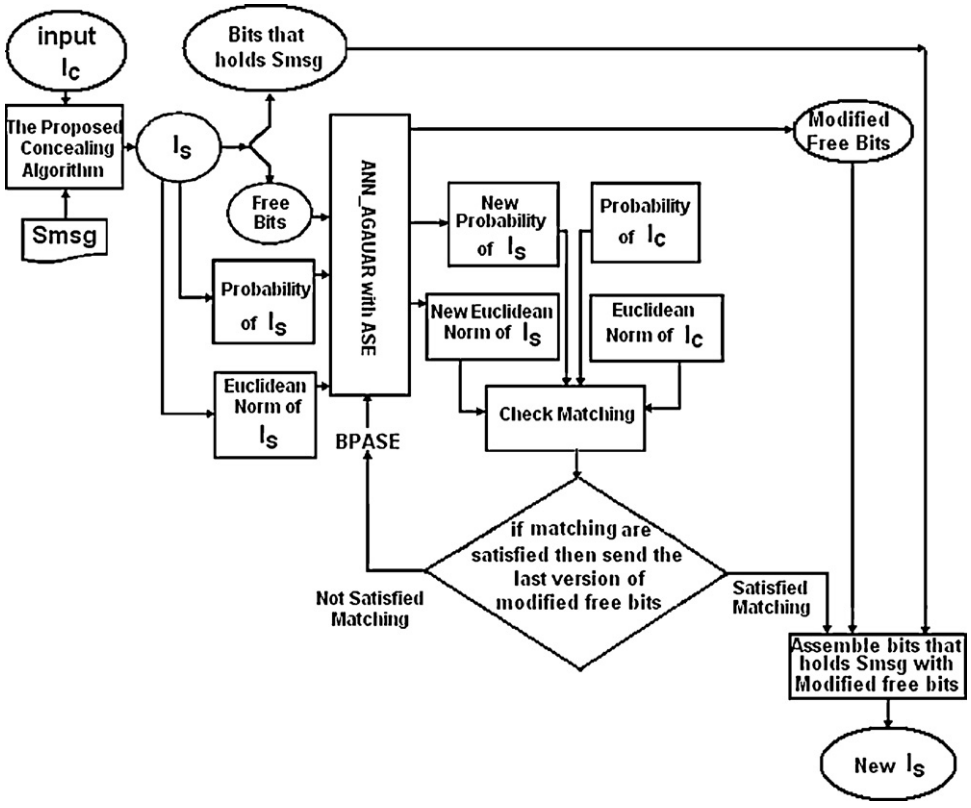

Fig. 7. ANN_AGAUAR training system.

action to adjust a process. Back-propagation algorithm with hybrid ANN_AGAUAR algorithm is applied through three stages: the feed forward of the input training pattern, the back-propagation of the associated error, and the adjustment of the weights. In addition, the adaptive smoothing error ASE is introduced effectively to speed up training processes (El-Emam and Abdul-Shaheed, 2008). Extra difficulties are added to work against statistical and visual attacks see in Fig. 7. The proposed training algorithm is shown in the following steps:

Step 1: Input: I_c and Smsg.
Step 2: Implement the proposed steganography algorithm to conceal a Smsg and produce I_s.
Step 3: Find statistical measures (Chi-square χ^2 and probabilities Pr) and visual measure (Euclidian) for each I_s and I_c.
Step 4: Extract all bits with their byte location, which are not used by the proposed steganography algorithm and saved into a temporary buffer that calls a free bits' buffer.
Step 5: Use hybrid ANN_AGAUAR Eqs. (8)–(29) based on back propagation algorithm and adaptive smoothing error BPASE with adaptive genetic algorithm using uniform adaptive relaxation (Sections 4.2–4.3), where the input layer contains free bits' buffer, statistical measure and visual measure for the specific I_s while the output layer produces a new free bits' buffer and new statistical and visual measures for the updated I_s, see Fig. 7.
Step 6: Check matching between I_s and I_c with respect to statistical and visual measures. If matching is satisfied, then the new I_s is constructed by adding new free bits' buffer to stego bits' buffer, else adjust (ADJ) weights value V_{ij}, W_{jk} and afterwards go to **Step 5**, see Fig. 6.

4.2. Apply ANN approach

The ANN has been trained using three layers back-propagation, the input neurons $I_i(\forall i = 1, \ldots, n)$ are received signals from I_s that contains the following buffers:

- Free bits which are not used by the proposed concealing algorithm with the size v neurons.
- The cumulative probabilities based on the chi-square statistic with the size $(n - v + 1)/2$ neurons.
- And Euclidean norm with the size $(n - v + 1)/2$ neurons.

and then broadcasts them to each of the hidden neurons $H_j(\forall j = 1, \ldots, p)$ through weights V_{ij} by using the activation function $\Phi(\cdot)$. See Eq. (8).

$$H_j = \Phi\left(\sum_{i=0}^{n} I_i V_{ij}\right) = \frac{1-e^{-\sum_{i=0}^{n} I_i V_{ij}}}{1+e^{-\sum_{i=0}^{n} I_i V_{ij}}}, I_0 = 1, \quad \forall j = 1, \ldots, p \tag{8}$$

Each hidden neuron H_j computes its activation function $\Phi(h)$ and sends its signal H_j to each of the output neurons $R_k(\forall k = 1, \ldots, n)$ through weights W_{jk}, where h is the activation function parameter of the hidden layer.

Each output neuron R_k computes its activation function $\Phi(r)$ to form the response of the neural network as in Eq. (9), where r is the activation function parameter of the output layer.

$$R_k = \Phi\left(\sum_{j=0}^{p} H_j W_{jk}\right) = \frac{1-e^{-\sum_{j=0}^{p} H_j W_{jk}}}{1+e^{-\sum_{j=0}^{p} H_j W_{jk}}}, H_0 = 1, \forall k = 1, \ldots, n \tag{9}$$

Activation functions $\Phi(\xi)$ that are used in the training system is bipolar sigmoid as in Eq. (10) with the range $[1, -1]$ and the first derivative of this function is defined in Eq. (11).

$$\Phi(\xi) = \frac{1-e^{-\xi}}{1+e^{-\xi}} \tag{10}$$

where ξ is the activation function parameter and the first-order derivative of activation functions $\Phi(\xi)$ is:

$$\frac{d(\Phi(\xi))}{d\xi} = \frac{1-\Phi^2(\xi)}{2} \tag{11}$$

During the training, the set of output neurons represents the following buffers from new I_s:

- New free bits' buffer.
- New cumulative probabilities buffer.
- And the new Euclidian norm buffer.

The new value of probabilities and Euclidian norm of I_s for the output layer represented by R_k should be compared with the target value t_k of probabilities and Euclidian norm of I_c see in Eq. (12).

$$\delta_k = (t_k - R_k)\frac{d\left(\Phi\left(\sum_{j=0}^{p} H_j W_{jk}\right)\right)}{dr} = (t_k - R_k)\left(\frac{1-\Phi^2\left(\sum_{j=0}^{p} H_j W_{jk}\right)}{2}\right)$$
$$= (t_k - R_k)\left(\frac{1-\left(\frac{1-e^{-\sum_{j=0}^{p} H_j W_{jk}}}{1+e^{-\sum_{j=0}^{p} H_j W_{jk}}}\right)^2}{2}\right), \forall k = v, \ldots, n \tag{12}$$

where $\delta_\kappa(\forall k = 1, \ldots, n)$ is the distribution error of the neurons values at the output layer and v is the size of free bits' buffer and $n-v+1$ is the size of two buffers (the cumulative probabilities based on the Chi-square statistic buffer and Euclidean norm buffer) for I_s.

$$\Delta W_{jk}^{new} = \beta \alpha_{jk}^{new} \delta_k H_j + (1-\beta).\Delta W_{jk}^{old} \tag{13}$$

where β is damping parameter in the interval [0, 1] and the adjustment of the weight ΔW_{jk}^{new} Eq. (13) is based on the factor δ_k and the activation of the hidden neuron H_j.

In the same manner, the factor $\delta_j(\forall j = 1, \ldots, p)$ has been computed for each hidden neuron H_j see Eq. (14).

$$\delta_j = \sum_{k=1}^{m} \delta_k W_{jk}\left(\frac{d\left(\Phi\left(\sum_{i=0}^{n} I_i V_{ij}\right)\right)}{dh}\right) = \sum_{k=1}^{m} \delta_k W_{jk}\left(\frac{1-\left(\Phi\left(\sum_{i=0}^{n} I_i V_{ij}\right)\right)^2}{2}\right)$$
$$= \sum_{k=1}^{m} \delta_k W_{jk}\left(\frac{1-\left(\frac{1-e^{-\sum_{i=0}^{n} I_i V_{ij}}}{1+e^{-\sum_{i=0}^{n} I_i V_{ij}}}\right)^2}{2}\right), \forall j = 1, \ldots, p \tag{14}$$

The adjustment to the weight ΔV_{ij}^{new} from the input neuron I_i to the hidden neuron H_j is based on the factor δ_j and the activation of the input neuron, see Eq. (15).

$$\Delta V_{ij}^{new} = \beta \alpha_{ij}^{new} \delta_j I_i + (1-\beta)\Delta V_{ij}^{old} \tag{15}$$

where the value of β at Eqs. (13) and (15) equal to 0.1.

Update the value of weight functions using Eqs. (16) and (17) and then modifying the numerical values of neural network connection weights by using a new optimization approach named Adaptive Genetic Algorithm with Uniform Adaptive Relaxation procedure AGAUAR(.), see Section 4.3.

$$W_{jk}^{new} = \text{AGAUAR}(W_{jk}^{old} + \Delta W_{jk}) \tag{16}$$

$$V_{ij}^{new} = \text{AGAUAR}(V_{ij}^{old} + \Delta V_{ij}) \tag{17}$$

Adaptive learning rate (El-Emam and Abdul-Shaheed, 2008) is applied to improve the speed of training by changing the rate of learning α during a training process, see Eq. (18).

$$\alpha_{jk}^{new} = \begin{cases} \alpha_{jk}^{old} + \lambda & \text{if} \quad \Delta W_{jk}^{new}\Delta W_{jk}^{old} > 0 \\ (1-\varepsilon)\alpha_{jk}^{old} & \text{if} \quad \Delta W_{jk}^{new}\Delta W_{jk}^{old} < 0 \\ \alpha_{jk}^{old} & \text{otherwise} \end{cases} \tag{18}$$

The appropriate values of λ and ε parameters have been predicted, these values are equal to 0.02 and 0.9 respectively. The training processes in the proposed algorithm are repeated many

times and update the old values of weights, which are represented by two-dimensional arrays V and W. The repetition of training is reached when the following condition is satisfied Eq. (19):

$$\max_k \left| t_k - R_k \right|^2 < 10^{-6}, \quad k = s, \ldots, n \tag{19}$$

Finally, we introduce an effective technique named Adaptive Smoothing Error (ASE) (El-Emam and Abdul-Shaheed, 2008; El-Emam and Al-Rabeh, 2011) to speed up training processes.

4.3. New adaptive genetic algorithm with uniform adaptive relaxation AGAUAR approach

The genetic algorithm is an optimization algorithm based on Darwinian models of natural selection and evolution (Wu and Shih, 2006), this algorithm has been used to solve a wide variety of problems. They have confirmed to be a distinguished helpfulness in solving optimization problems. The nonlinear adaptation applies nonlinear estimates to reshape the probability distribution of the trial parameters. The adaptive genetic algorithm had significantly faster convergence than non-adaptive genetic algorithms. Basic idea behind AGAUAR is that an initial population of candidate states of size n is chosen at random, and each state is evaluated according to the function of the optimization. The proposed AGAUAR algorithm with two levels showing below is the new version of the GA_MAR algorithm published in El-Emam and Al-Rabeh (2011).

First Level

Step 1: input the size of the population for ANN equal to n;
Step 2: foreach (Iteration q)
Step 2-2: foreach (Pattern P)
Step 2-2-1: Create randomly an initial population of individual Weights V_i, W_i $\forall i = 1, \ldots, n$
Step 2-2-2: Apply training on the pattern P through the following steps.
Step 2-2-2-1: Each weight V_i is characterized by its chromosome S_i, which is created by encoding weight value as bit strings of length L, see Eq. (20).

$$S_i = \text{bin}(V_i) = (S_{i1}, S_{i2}, \ldots, S_{iL}), \quad \forall i = 1, \ldots, n. \qquad |S_i| = L = 32. \tag{20}$$

Step 2-2-2-2: Evaluate the fitness function $f(S_i)$ Eq. (21) for each chromosome S_i, where the population at tth generation is designated by a sequence of bits $\{S_{i1}(t), S_{i2}(t), \ldots, S_{in}(t)\}$.

$$f(S_i) = 1 + \frac{\text{Cos}(S_i)}{1 + 0.01\, S_i^2} \forall\, i = 1, \ldots, n. \tag{21}$$

Step 2-2-2-3: Evaluate the uniform adaptive fitness function $f_{UAFF}(Si)$ Eq. (22).

$$f_{UAFF}(S_i) = i + (n - r) f(S_i) \quad \forall i = 1, \ldots, n, \exists r = \text{Rand}(1, \ldots, n) \tag{22}$$

where Rand(.) is random function.
Step 2-2-2-4: Find the probability Pr for each S_i see Eq. (23):

$$Pr(S_i) = \frac{f_{UAFF}(S_i)}{\sum_{j=1}^{n} f(S_j)}; \forall i \in 1, \ldots, n. \tag{23}$$

Step 2-2-2-5: Select randomly two chromosomes Si Eq. (23):

$$S_k, S_j \ni k = \text{Rand}(1, \ldots, n) \wedge j = \text{Rand}(1, \ldots, n) \wedge k \neq j$$

and the following condition should be satisfied:

$$\Pr(S_z) < 0.6 \ \forall z \in \{k, j\}$$

Step 2-2-2-6: Applying genetic operators (single point crossover and point mutation) to generate an offspring population $\{S_{z_1}(t+1), S_{z_2}(t+1), \ldots, S_{z_n}(t+1)\}$ $\forall z \in \{k, j\}$ according to the following steps:
Step 2-2-2-6-1: Applying single point crossover (Op_{Xor}) Eq. (24) on two individuals chromosomes S_k and S_j that are selected in the **Step 2-2-2-5**. A crossover position between 1 and L is chosen at random, and the two parents exchange portions of their binary representations.

$$\{S'_k(t), S'_j(t)\} = Op_{Xor}(S_k(t), S_j(t)) \tag{24}$$

Step 2-2-2-6-2: Create new offspring population by applying mutation ($Op_{\neg}$) Eq. (25) on the output of the crossover phases $\{S'_k(t), S'_j(t)\}$ by selecting randomly one bit from each chromosome, and then convert into its negative value.

$$\{S_k(t+1), S_j(t+1)\} = Op_{\neg}(S'_{k,h}(t), S'_{j,h}(t)) \tag{25}$$

where $h \in [1, L]$ is the bit index in the chromosome.
Step 2-2-2-7: Replace new weights in the population.
Step 2-2-2-8: Using uniform adaptive relaxation (the new version of MAR (El-Emam and Al-Rabeh, 2011) to find new values of chromosomes $\{S_k^{\text{new}}(t+1), \quad S_j^{\text{new}}(t+1)\}$, Eqs. (26) and (27).

$$S_z^{\text{new}}(t+1) = \begin{cases} \rho' S_z(t+1) + (1-\rho'') S'_z(t) & \left| f(S_z(t+1)) - f(S'_z(t)) \right| > \varepsilon. \\ \rho' S_z(t+1) + (1-\rho'') S'_z(t) & \left| f(S_z(t+1)) - f(S'_z(t)) \right| < \varepsilon. \end{cases}, \quad \forall z \in \{k, j\} \tag{26}$$

where $\varepsilon = 10^{-3}$ and

$$\rho' = 0.3\rho, \quad \rho'' = 0.3 + 0.3\rho, \quad \text{and} \quad \rho \in \text{Rand}(0 \sim 1) \tag{27}$$

Step 2-2-2-9: Repeat **Step 2-2-2-1** to **Step 2-2-2-8** $\forall i = 0, 1, 2, \ldots$ until the convergence criterion is satisfied Eq. (28).

$$Pr(S_i) \geq 0.6 \quad \forall i = 1, \ldots, n. \tag{28}$$

Step 2-2-2-10: Apply **Step 2-2-2-1** to **Step 2-2-2-8** on the weight W_i
Step 2-2-3: Update and save the new output of the weights see Eqs. (16) and (17) with the smoothing processes see Eqs. (13) and (15).

Second Level

Step 1: Apply ASE through the following steps:
Step 1-1: Foreach (Pattern P), find the sum of neurons' errors.
Step 1-2: Select P that has the maximum sum of neurons' errors Eq. (29).

$$P' = \max_{\forall P} \left| \sum_{\forall \text{neurons}} (\text{Error}) \right| \tag{29}$$

Step 1-3: Go to **step 2–2-1** at the first level of the algorithm and set P equal to P'.

Example 4.1. Assume we have nine surrounding bytes (3×3) in a cover image I_c (the current byte $b_{i,j}$ and $NEB_{i,j}$) shown in Fig. 8. We should explain step by step how to conceal Smsg and then how to implement the intelligent technique ANN-AGAUAR to improve the values of bytes and produce stego image I_s with high imperceptible ratio. Assume that the Nbpb is calculated by the proposed algorithm for each byte, and it is equal to 2 bits of each byte.

It appears that the first step of the proposed algorithm is used to conceal the secret message into I_c while the second step using an intelligent technique ANN_AGAUAR to reduce PVD (Pixels Value Difference) between I_c and I_s and then to satisfy the pixel uniformity to work against statistical and visual attacks.

5. Experimental results and discussion

In the proposed scheme, new steganography algorithms are working efficiently to conceal a large amount of Smsg into a cover image with payload capacity reached to 50% of the image size, moreover ANN_AGAUAR has been introduced effectively to work against statistical and visual attacks and to modify the stego image to become imperceptible to the human eye.

In this study, more than 500 color images are used to achieve training on the proposed system ANN_AGAUAR and to perform comparisons between the proposed scheme and previous works.

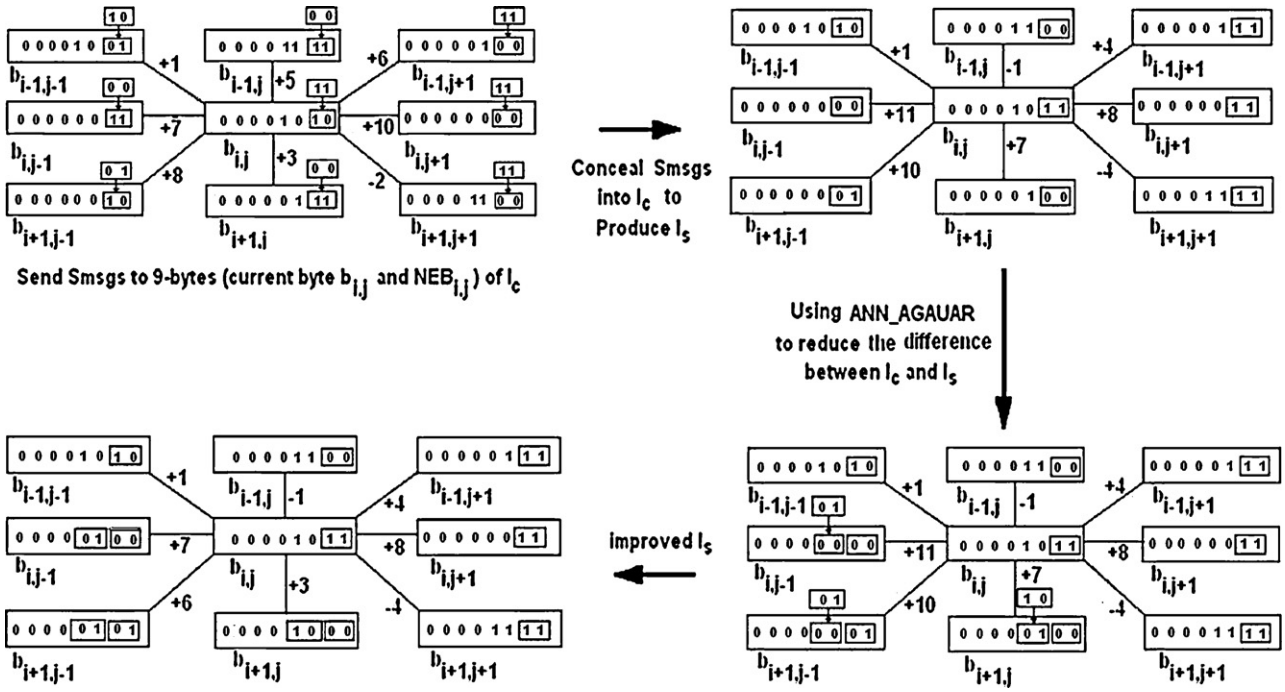

Fig. 8. Steps to conceal Smsg into I_c and to improve I_s.

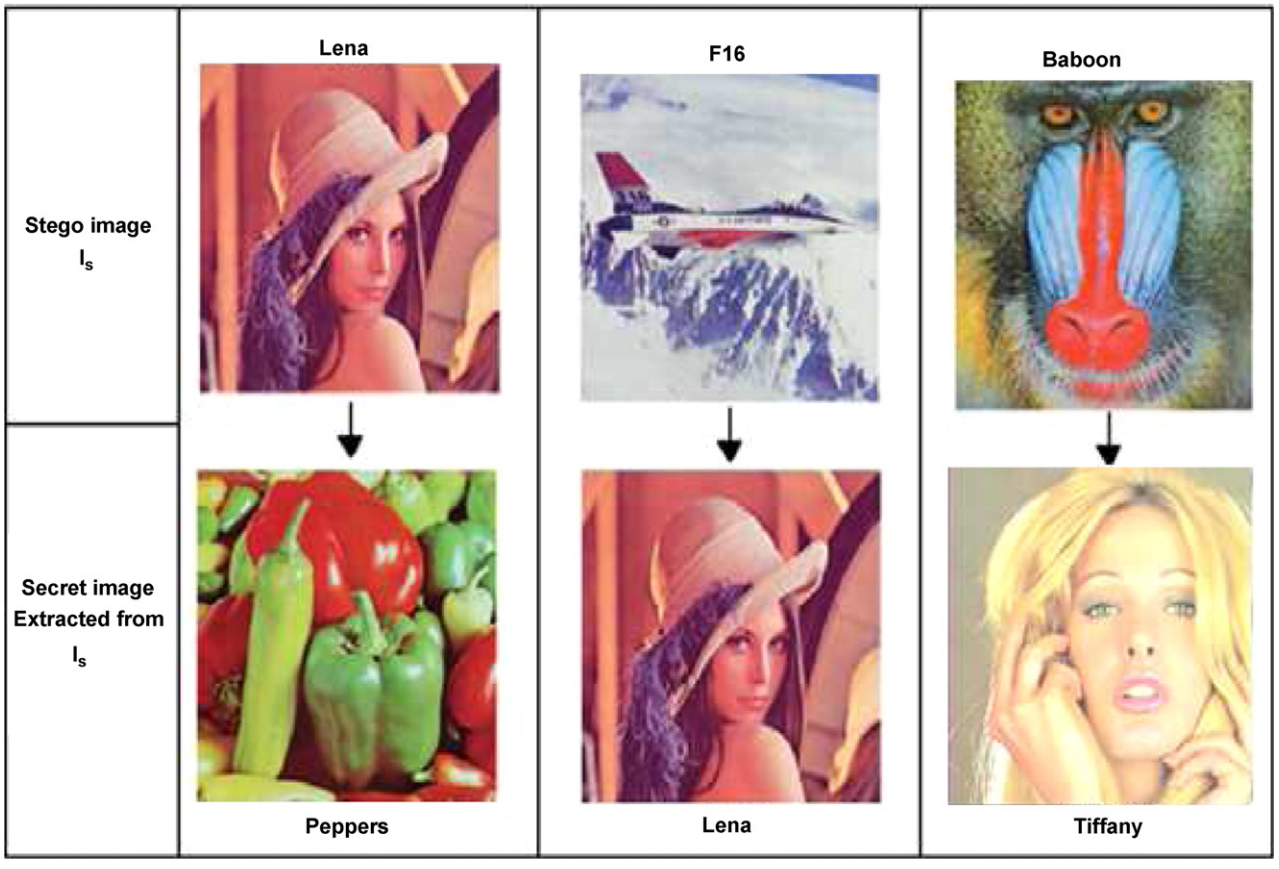

Fig. 9. Stego-images I_s and their corresponding extracted secret images.

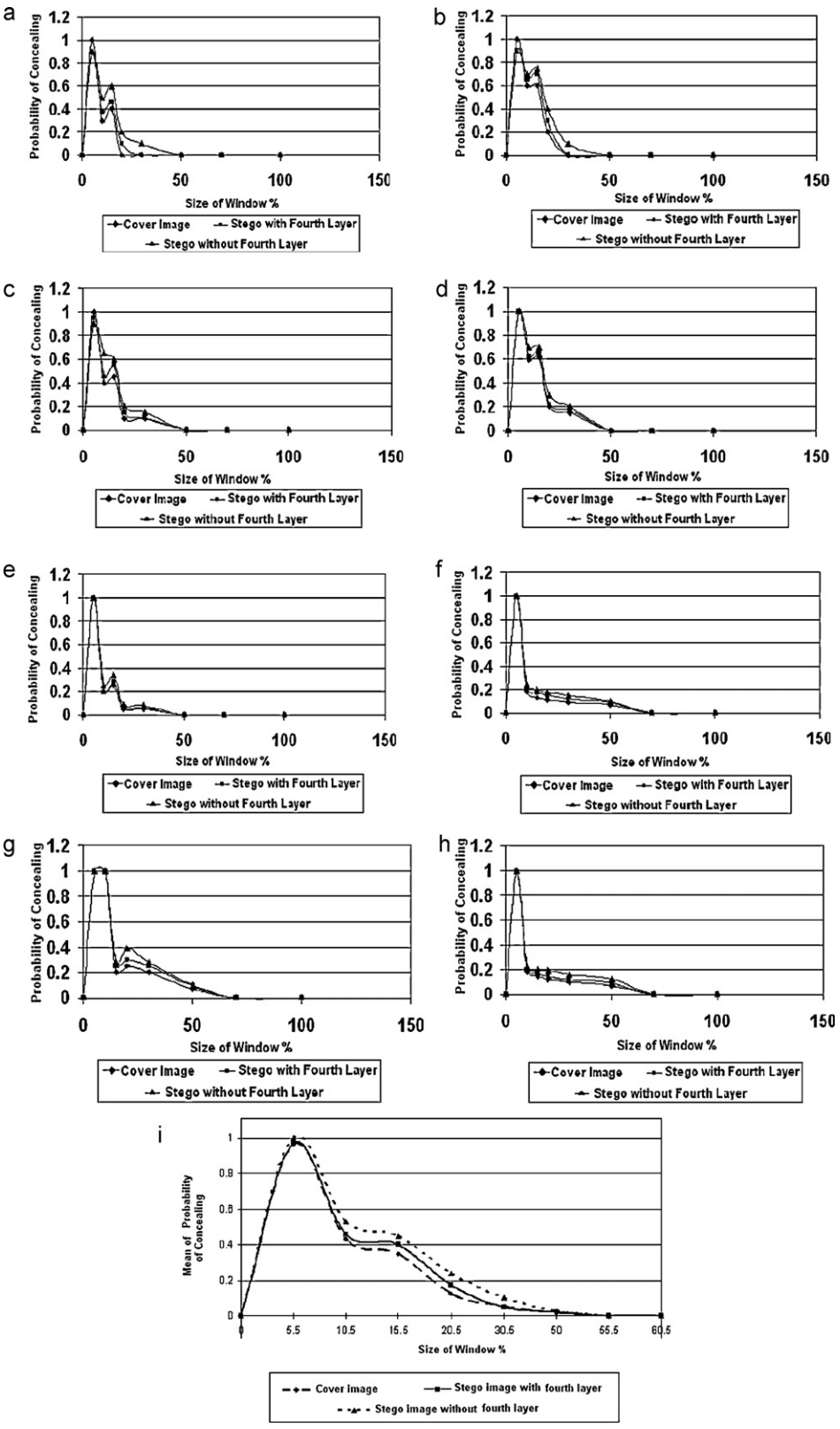

Fig. 10. (a) Probability of concealing a Smsg of length 25% of the Baboon image size. (b) Probability of concealing a Smsg of length 25% of the Peppers image size. (c) Probability of concealing a Smsg of length 49.42% of the Baboon image size. (d) Probability of concealing a Smsg of length 48.56% of the Peppers image size. (e) Probability of concealing a Smsg of length 25% of the Lena image size. (f) Probability of concealing a Smsg of length 25% of the F16 image size. (g) Probability of concealing a Smsg of length 47.78% Of the Lena image size. (h) Probability of concealing a Smsg of length 46.13% of the F16 image size. (i) The average of probabilities of concealing for 300 color images.

Table 3
PSNR(dB) results of I_s on a color plane between Chen's algorithm (Chen, 2008) and the proposed algorithm.

Stego images 512×512	Using SPIHT compression on secret image	PSNR(dB) of I_s on R-plane			PSNR(dB) of I_s on G-plane			PSNR(dB) of I_s on B-plane			PSNR(dB) of I_s (the average of three color planes)		
		Chen (2008)	Proposed algorithm with/without ANN-AGAUAR		Chen (2008)	The proposed algorithm with/without ANN-AGAUAR		Chen (2008)	The proposed algorithm with/without ANN-AGAUAR		Chen (2008)	The proposed algorithm with/without ANN-AGAUAR	
			With out	With		With out	With		With out	With		With out	With
Lena	Peppers	37.97	39.01	47.24	37.87	39.42	47.65	39.78	39.98	46.22	38.54	39.47	47.03
F16	Lena	36.32	37.45	45.48	35.55	37.12	45.35	37.43	39.71	47.94	36.43	38.09	46.25
Baboon	Tiffany	37.39	39.21	47.44	36.38	37.98	46.22	35.85	37.17	45.41	36.54	38.12	46.35

Table 4
PSNR(dB) results of Is on a color plane between of Chang's algorithm (Chang et al., 2008) and the proposed algorithm.

I_s (512×512) includes random secret bits.	Concealing percentage capabilities of RGB color for Chang et al. (2008)	Concealing percentage capabilities of RGB color for the proposed algorithm	PSNR(dB) of I_s on R-plane			PSNR(dB) of I_s on G-plane with maximum payload			PSNR(dB) of I_s on B-plane with maximum payload		
			Chang et al. (2008)	The proposed algorithm with/without ANN-AGAUAR		Chang et al. (2008)	The proposed algorithm with/without ANN-AGAUAR		Chang et al. (2008)	The proposed algorithm with/without ANN-AGAUAR	
				Without	With		Without	With		Without	With
Lena	45.81%	47.78%	34.04	40.21	47.24	31.99	38.68	46.22	33.43	40.68	46.22
F16	45.24%	46.13%	32.23	41.23	47.44	31.02	41.47	47.65	34.78	41.78	47.94
Baboon	44.91%	49.42%	27.3	39.65	45.48	25.98	38.32	45.35	25.92	38.16	45.40
Peppers	45.81%	48.56%	33.65	37.96	46.20	31.46	38.13	46.37	33.26	37.68	45.92
Tiffany	32.98%	46.74%	36.68	41.27	49.51	33.81	38.19	46.43	35.94	38.36	46.60

5.1. Testing image quality

Using Peak Signal to Noise Ratio (PSNR dB) and structural similarity (SSIM) metrics to check image quality after data concealing. PSNR is defined in Eq. (30).

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{\max^2}{\mathrm{MSE}} \tag{30}$$

where max is the maximum pixel value, and MSE represents the average of mean square errors for RGB colors shown in Eq. (31).

$$\mathrm{MSE} = \frac{\mathrm{MSE}_R + \mathrm{MSE}_G + \mathrm{MSE}_B}{3} \tag{31}$$

and the MSE_R, MSE_G, and MSE_B are the mean square of the three colors and are computed by using the following Eq. (32):

$$\mathrm{MSE}_c = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (C_{ij}^c - S_{ij}^c)^2; c \in \{R, G, B\} \tag{32}$$

where $(m \times n)$ is the size of image and C^c, S^c are two bytes of the color c at the location (i,j) from cover and stego images respectively.

In Fig. 9, five testing color images (512×512) have been used, namely "Baboon", "F16", "Lena", "Peppers" and "Tiffany" are shown.

PSNR for each color plane (R, G, B) is calculated individually on three stego images and three secret images extracted from stego images see Fig. 9. The result of the proposed algorithm based on ANN_AGAUAR is compared with Chen's algorithm (Chen, 2008). It appears obvious that the quality of I_s using our scheme is preserved with/without ANN_ AGAUAR, and it obtains superior performance and better than the algorithm in (Chen, 2008) for all colors with an excellent imperceptibility see Table 3.

Moreover, the proposed scheme with ANN_AGAUAR is less degradation around 17% than the same scheme without ANN_AGAUAR and less degradation around 20% than a stego image generated by the Chen's algorithm (Chen, 2008).

Table 4 shows the comparison between the experimental results of the proposed hiding algorithm with/without ANN_AGAUAR and the algorithm in Chang et al. (2008). The comparison is based on PSNR (dB) to demonstrate the visual quality after concealing Smsg, where Smsg is the largest size of the random bit stream generated randomly.

It confirms obvious that the quality of I_s using the proposed algorithm is preserved and better than the algorithm in Chang et al. (2008) for all colors. However, the quality of stego images in the Table 4 generated by Chang's algorithm (Chang et al., 2008) are improved around (29%) and (18%) as the average if the proposed algorithm is accomplished with/without ANN-AGAUAR respectively. In addition, payload capacity of the proposed algorithm is larger than Chang's algorithm.

The SSIM algorithm (Wang et al., 2004) is used to measure the similarity between two identical images. In this work, this metric is introduced using Eq. (33):

$$\mathrm{SIMM}(I_c, I_s) = \frac{((2\mu_{I_c}\mu_{I_s} + (2^{24}-1) \times 0.01)^2)(2\sigma_{I_c I_s} + ((2^{24}-1) \times 0.03)^2)}{(\mu_{I_c}^2 + \mu_{I_s}^2 + ((2^{24}-1) \times 0.01)^2)(\sigma_{I_c}^2 + \sigma_{I_s}^2 + ((2^{24}-1) \times 0.03)^2)} \tag{33}$$

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

Table 5 reported 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 measured by using PSNR (dB) and SSIM metrics to show the performance of the proposed algorithm over typical existing references (Hong et al., 2012; Geetha et al., 2011). In this study, 400 images have been selected by size (384×512); all these images are converted to the grayscale images.

It seems that the proposed algorithm working efficiently, and it is better than the previous works (Hong et al., 2012; Geetha et al., 2011) for the average value of PSNR(dB) and SSIM on various images.

5.2. The amount of data hiding

Experimental results are considered on many color images to check the largest amount of payload capacity and the concealing percentage. Table 6 illustrates the largest amount of the concealing data on five color images using Eq. (5) on each byte. It appears that the proposed steganography algorithm based on ANN_AGAUAR can conceal a large amount of data reach to 12 bpp (bit per pixel) with high image quality. We should emphasize that the "Baboon" image can conceal the largest amount of data while the "F16" image is at the smallest. This variation of image's capabilities to hide bytes depends on the color variation in the small region and the stream content of the secret bits (Smsg).

5.3. Avoiding Chi-square (χ^2) attack

Westfeld and Pfitzmann (2000) introduced a method based on statistical analysis of a Pair of Values (PoVs) that are exchanged during sequential concealing. In this work, we use this kind of testing to check how the expected, and the observed frequencies of stego image pixels are uniformed. Chi-square (χ^2) is used for this purpose and calculated in Eq. (34):

$$\chi_{k-1}^2 = \sum_{i=0}^{k-1} \frac{(O_i - E_i)^2}{E_i} \tag{34}$$

where $(k-1)$ is a degree of freedom and E_i is the expected frequency of ith pair colors see Eq. (35).

$$E_i = \frac{1}{2} \underset{c \in \{R,G,B\}}{fr} \left\{ C_{2i}^c, C_{2i+1}^c \right\}, \forall i = 0, \ldots, k-1 \tag{35}$$

where $\left\{ C_{2i}^c, C_{2i+1}^c \right\}$ is the ith pair of the palette colors $C_0^c, C_1^c, \ldots, C_{255}^c$. Moreover, the frequency observes at the ith color are shown in Eq. (36)

$$O_i = fr(C_i) \forall i = 0, \ldots, k-1 \tag{36}$$

The probability $P_{\chi^2,k-1}$ bases on Chi-square value χ^2 with $k-1$ degree of freedom is calculated using Eq. (37a).

$$P_{\chi^2,k-1} = \left(2^{\frac{k-1}{2}} \Gamma\left(\frac{k-1}{2}\right)\right)^{-1} \int_{\chi^2}^{\infty} (t)^{(k-1/2)-1} e^{-(t/2)} dt \tag{37a}$$

where Gamma Γ is the generalization of the factorial function Eq. (37b)

$$\Gamma(x) \int_0^{\infty} t^{x-1} e^{-t} dt \tag{37b}$$

It appears that the Chi-square attack on I_s with randomly disordered messages produces irregular P values in the beginning and then, as the sample size increases, the P value ultimately drops to zero. Comparisons among four cover images and their corresponding stego images are implemented; it seems that the proposed algorithm with fourth layer (ANN_AGAUAR) produces the same behavior of the probability P with respect to the length of Smsg (25–49%) of the image size for a different kind of window size. In addition, the P value eventually drops to zero at the window sizes 20 or 35 depending on the message size. Therefore, stegoanalysis

Table 5
The average values of PSNR (dB), and SSIM of various stego images generated by different Steganographic algorithms.

Payload capacity	Average values of 8 images (Hong et al., 2012)		Average values of 300 images (Geetha et al., 2011)		Proposed algorithm with ANN-AGAUAR average values of 400 images	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
10%	51.74	NA	50.8	1	64.11	0.9999
30%	44.72	NA	45.5	0.9997	63.50	0.9998
40%	40.83	NA	NA	NA	58.17	0.9996
50%	NA	NA	43.6	0.9992	57.20	0.9995

Table 6
The concealing capacity and the concealing percentage (%) of the proposed algorithm.

Items	Baboon 512 × 512	Peppers 512 × 512	F16 512 × 512	Lena 512 × 512	Tiffany 512 × 512
Capacity in bits for each Color	R = 1,052,027	R = 1,024,296	R = 979,368	R = 942,056	886,503
	G = 1,071,400	G = 997,284	G = 971,747	G = 1,021,402	1,044,235
	B = 1,068,769	B = 1,033,697	B = 951,277	B = 1,042,653	1,010,245
Total capacity in bits	3,192,196	3,055,277	2,902,392	3,006,111	2,940,983
Percentage of payload capacity	49.42%	48.56%	46.13%	47.78%	46.74%

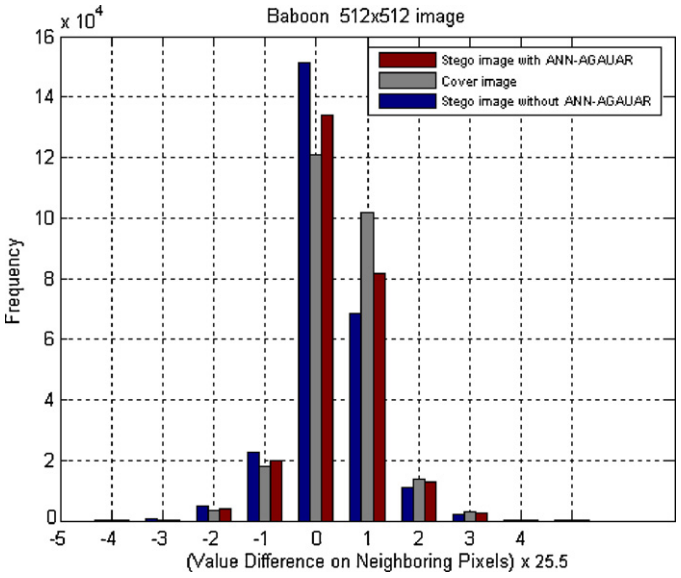

Fig. 11. Value difference on neighboring pixels for both Baboon Cover and Baboon Stego image.

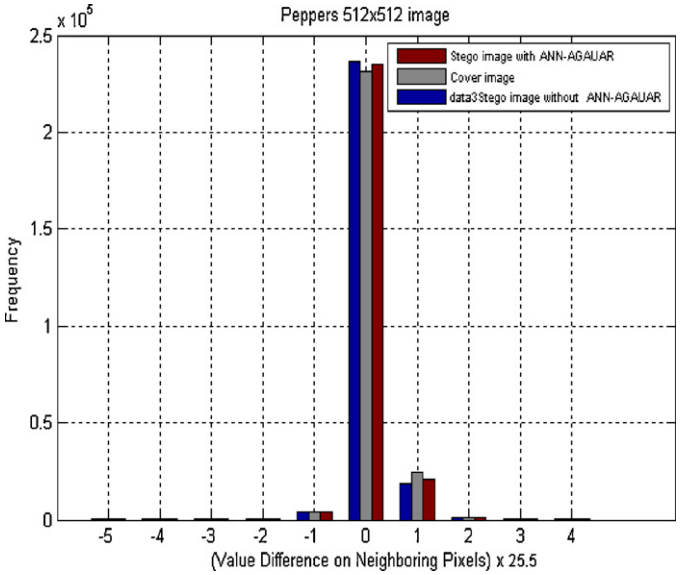

Fig. 12. Value difference on neighboring pixels for both Peppers Cover and Peppers Stego image.

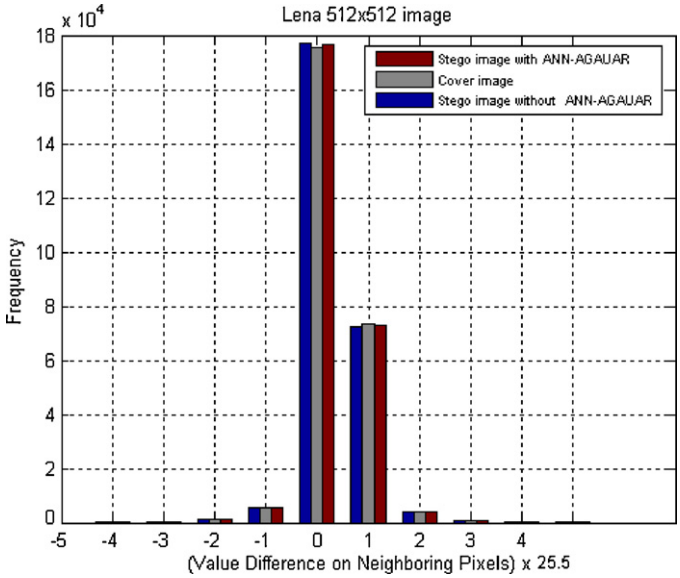

Fig. 13. Value difference on neighboring pixels for both Lena Cover and Lena Stego image.

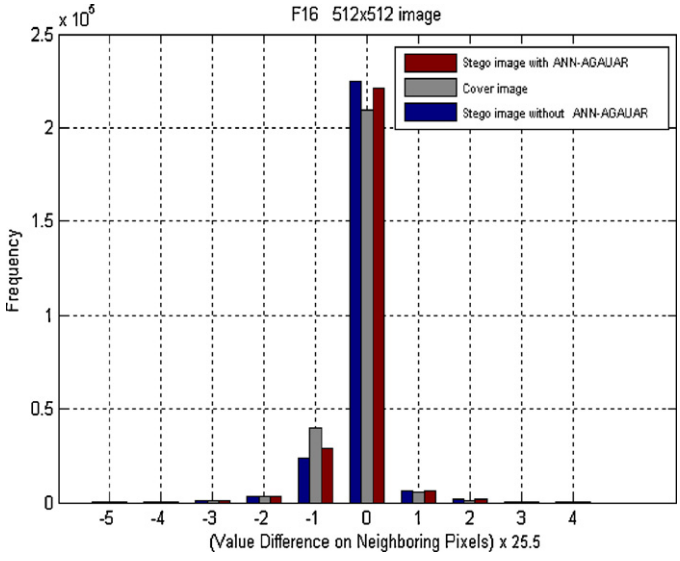

Fig. 14. Value difference on neighboring pixels for both F16 Cover and F16 Stego image.

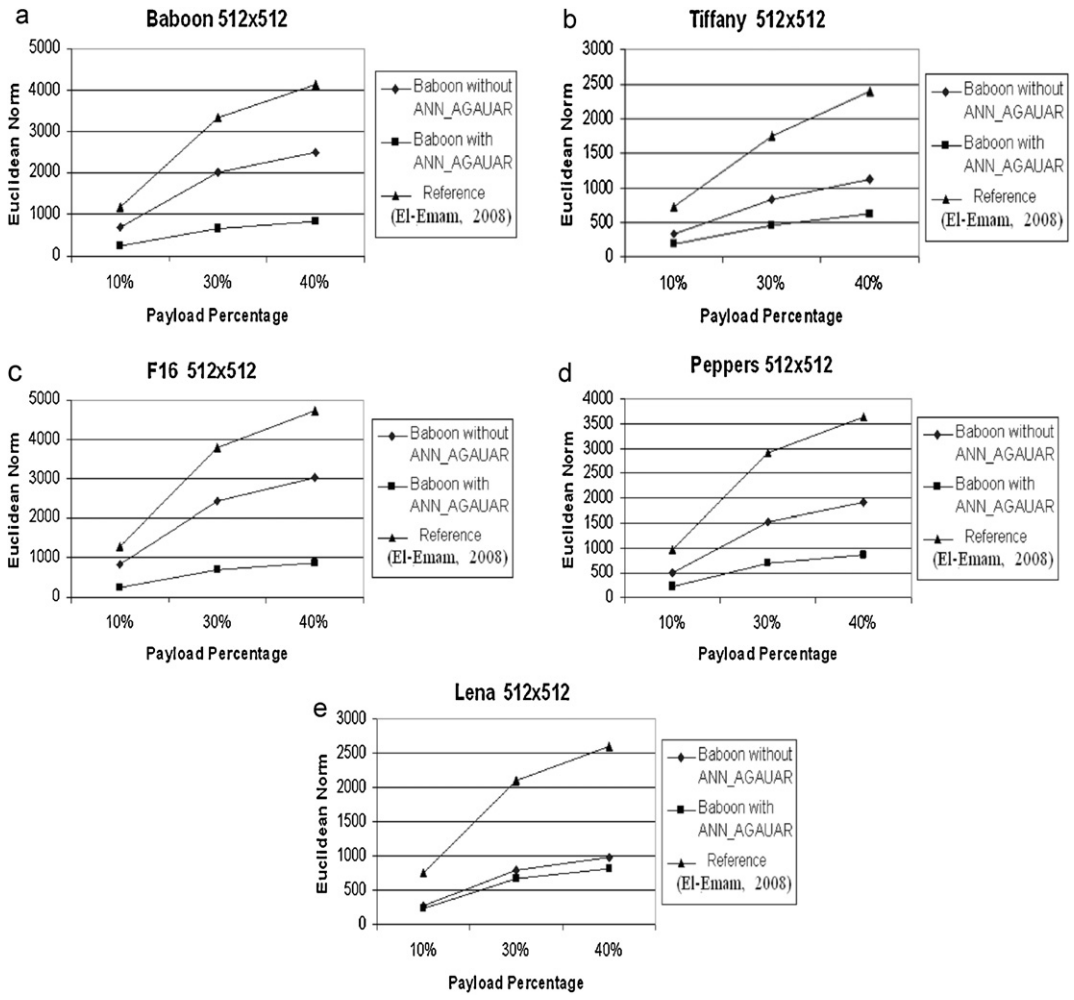

Fig. 15. (a) Euclidean norm testing of Baboon color image. (b) Euclidean norm testing of Tiffany color image. (c) Euclidean norm testing of F16 color image. (d) Euclidean norm testing of Peppers color image. (e) Euclidean norm testing of Lena color image.

cannot be detected Smsg due to the highly matched of P between the stego and cover images see Fig. 10a–h. Finally, Fig. 10i shows the mean of probabilities of concealing for 300 color images. It appears that the average of the probability of the propose algorithm using the fourth layer is an excellent and nearest to the average of the probability of the I_c.

5.4. Difference between neighboring pixels

The difference value of the horizontal neighboring pair for both I_c and I_s are computed using the formula in Eq. (38):

$$d^c_{i,j} = P^c_{i,j} - P^c_{i,j+1}, \quad d^s_{i,j} = P^s_{i,j} - P^s_{i,j+1}, \quad \forall i,j \tag{38}$$

where P^c_{ij}, P^s_{ij} are two pixels values at the location (i,j) for both I_c and I_s respectively. Comparisons of two kinds of differences d^c_{ij} and d^s_{ij} using four images are performed (Figs. 11–14).

Results show that the difference values between I_c and I_s with the fourth layer of security (ANN_AGAUAR) is less than the difference between them without using ANN_AGAUAR.

5.5. Working against visual attack

Two kinds of testing are implemented, the first one bases on the set of the closest colors (one corresponding to the same pixel) using Euclidean norm Eq. (39) to find the distance between the I_c and I_s. Experimental testing of the Euclidean norm has been implemented on two algorithms (our algorithm published in (El-Emam, 2008) and the new algorithm proposed in this work), see Fig. 15a–e.

$$d = \sqrt{(R_c - R_s)^2 + (G_c - G_s)^2 + (B_c - B_s)^2} \tag{39}$$

The distances of five images are calculated individually; it seems that the smallest norm is reached when the proposed algorithm using ANN_AGAUAR is implemented. Moreover, we observed that the greatest difference is equal to 4738 at the image F16 with payload percentage equal to 40% when the earliest work (El-Emam, 2008) is used, while with the proposed algorithm with/without using ANN_AGAUAR, we can reduce the difference about 80% and 37% respectively, whereas the smallest difference is equal to 748

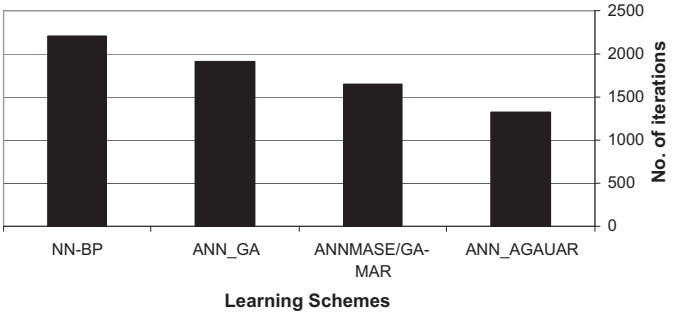

Fig. 16. Speed of training with learning schemes.

Table 7
The length of I_c and I_s.

Color Image 512 × 512	Payload percentage (%) using random secret bits	L^c	L^s		
			Without ANN_AGAUAR	With ANN_AGAUAR	El-Emam (2008)
Baboon	10%	225,870	226,330	226,020	233,869
	30%	225,870	230,430	226,330	235,390
	40%	225,870	231,770	226,460	235,897
Tiffany	10%	363,650	364,210	363,760	376,421
	30%	363,650	366,760	364,280	377,479
	40%	363,650	367,880	364,510	377,945
F16	10%	324,020	324,730	324,160	335,486
	30%	324,020	328,090	324,800	336,866
	40%	324,020	329,580	325,080	337,476
Peppers	10%	206,190	206,960	206,340	213,617
	30%	206,190	211,290	207,200	215,407
	40%	206,190	212,390	207,420	215,862
Lena	10%	237,570	238,270	237,710	246,052
	30%	237,570	242,140	238,480	247,652
	40%	237,570	243,200	238,690	248,090

at the image Tiffanty with the payload equal to 10% for the earlier work (El-Emam, 2008) and by using the proposed algorithm with/without using ANN_AGAUAR, we can reduce the difference about 74% and 43% respectively.

The second type of testing is focused on the stimulus length as a measure of brightness Eq. (40), this measure has provided a brightness definition L^C, L^S effective for both I_c and I_s respectively. Table 7 shows the amount of the length L of five color images for both I_c and I_s using the proposed algorithm with/without using ANN_AGAUAR and earlier work (El-Emam, 2008).

$$L^c = \sqrt{R_c^2 + G_c^2 + B_c^2}, \quad L^s = \sqrt{R_s^2 + G_s^2 + B_s^2} \tag{40}$$

The experimental results reported in Table 7 explained that the proposed algorithm with ANN_AGAUAR adjusts image brightness successfully and making it better than the others' techniques. The results with ANN_AGAUAR improve the brightness around 60% and 94% than the proposed algorithm without ANN_AGAUAR and the earlier work (El-Emam, 2008) respectively. Moreover, results are showing that with the payload percentage equal to 40%, the Peppers image has a greatest brightness while Baboon has a minimum brightness. On the other hand, when the payload percentage equal to 10%, the Baboon and the Peppers images have the most brightness while the Tiffany has a least brightness.

5.6. Speed of training of ANN_AGAUAR scheme

In this study, 400 images are selected randomly to build learning scheme ANN_AGAUAR. The speed of training for the proposed scheme is checked based upon the number of iterations to be sure and confirm that the new scheme is working effectively to optimize the values of weights W_{jk}, V_{ij} for the ANN. Fig. 16 illustrates the speed of training with/without using AGAUAR. It appears that training with AGAUAR is 25% quicker than training using GA-MAR (El-Emam and Al-Rabeh, 2011), 45% quicker than the conventional GA and 67% faster than neural networks without GA.

6. Conclusion

This paper proposed new steganography algorithm to conceal large amounts of secret message into color bitmap image using four levels of security. The main contributions of this paper are:

1. Proposed new algorithm for non-uniform image segmentation to conceal Smsg randomly instead of sequentially.
2. Proposed four security layers to work against statistical and visual attacks.
3. Calculate the BL for each 3 × 3 surrounding bytes (the current byte and NEB) with their variances σ^2 to find the maximum length of hiding information.
4. Proposed new intelligent computing technique based on adaptive neural networks with adaptive genetic algorithm using uniform adaptive relaxation ANN_AGAUAR to speed up training process and to achieve a high concealing rate beside an excellent imperceptible data.

Experimental results indicate that the proposed algorithm solves the problem of high concealing capacity. Moreover, results demonstrate that, in contrast with available four layers; this is very valuable and significant obtains a superior concealing rate. According to the property that most image pixels are similar to their surrounding pixels, the difference values between the cover pixels, and their corresponding predictive pixels are small when an intelligent technique is introduced. Finally, the results of our experiments show that the stego images produced by the proposed steganography algorithm are not detected by the state-of-the-art steganalyzers.

Acknowledgments

The authors would like to thank Prof. R. H. Al-Rabeh from Cambridge University for his supported and help. This support is gratefully acknowledged.

References

Ali, L., Aris, I., Hossain, F., Roy, N., 2011. Design of an ultra high speed AES processor for next generation IT security. Computers and Electrical Engineering 37 (6), 1160–1170, Elsevier. doi:10.1016/j.compeleceng.2011.06.003.

Arsalan, M., Malik, S., Khan, A., 2012. Intelligent reversible watermarking in integer wavelet domain for medical images. Journal of Systems and Software 85 (4), 883–894, Elsevier. doi:10.1016/j.jss.2011.11.005.

Chang, C., Lin, C., Fan, Y., 2008. Lossless data hiding for color images based on block truncation coding. Pattern Recognition 41, 2347–2357, Elsevier. doi:10.1016/j.patcog.2007.12.009.

Chang, C., Chen, Y., Lin, C., 2009. A data embedding scheme for color images based on genetic algorithm and absolute moment block truncation coding. Soft Compute 13 (4), 321–331, Springer. doi:10.1007/s00500-008-0332-x.

Chen, W., 2008. Color image steganography scheme using DFT, SPIHT codec, and modified differential phase-shift keying techniques. Applied Mathematics and Computation 196, 40–54, Elsevier. doi:10.1016/j.amc.2007.05.063.

EL-Emam, N., 2007. Hiding a large amount of data with high security using steganography algorithm. Journal of Computer Science 3 (4), 223–232, doi:10.3844/jcssp.2007.223.232.

El-Emam, N., 2008. Embedded a large amount of information using high secure neural based steganography algorithm. International Journal of Information and Communication Engineering 4 (2), 95–106.
El-Emam, N., Abdul-Shaheed, R., 2008. Computing an adaptive mesh in fluid problems using neural network and genetic algorithm with adaptive relaxation. International Journal on Artificial Intelligence Tools 17 (6), 1089–1108, Word Scientific. doi:10.1142/S021821300800431X.
El-Emam, N., Al-Rabeh, R., 2011. An intelligent computing technique for fluid flow problems using hybrid adaptive neural network and genetic algorithm. Applied Soft Computing 11 (4), Elsevier. doi:10.1016/j.asoc.2009.12.009.
Filler, T., Judas, J., Fridrich, J., 2011. Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Transactions on Information Forensics and Security 6 (3), 920–935, doi:10.1109/TIFS.2011.2134094.
Fridrich, J., 2009. Asymptotic behavior of the ZZW embedding construction. IEEE Transactions on Information Forensics and Security 4 (1), 151–153, doi:10.1109/TIFS.2008.2011082.
Geetha, S., Kabilan, V., Chockalingam, S., KamarajVarying, N., 2011. Radix numeral system based adaptive image steganography. Information Processing Letters 111, 792–797, Elsevier. doi:10.1016/j.ipl.2011.05.013.
Hong, W., Chen, T., Luo, C., 2012. Data embedding using pixel value differencing and diamond encoding with multiple-base notational system. The Journal of Systems and Software 85, 1166–1175, Elsevier. doi:10.1016/j.jss.2011.12.045.
Lee, Chen, H., 2010. A novel data hiding scheme based on modulus function. Journal of Systems and Software 83 (5), 832–843, Elsevier. doi:10.1016/j.jss.2009.12.018.
Lee, Chen, H., Tso, H., 2010. Embedding capacity raising in reversible data hiding based on prediction of difference expansion. Journal of Systems and Software 83 (10), 1864–1872, Elsevier. doi:10.1016/j.jss.2010.05.078.
Li, X., Wang, J., 2007. A steganographic method based upon JPEG and particle swarm optimization algorithm. Information Sciences 177, 3099–3109, Elsevier. doi:10.1016/j.ins.2007.02.008.
Luo, X., Wang, D., Hu, W., Liu, F., 2009. Blind detection for image steganography: a system framework and implementation. International Journal of Innovative Computing, Information and Control 5 (2), 433–442.
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), doi:10.1109/TIFS.2010.2041812.
Munuera, C., 2007. Fast communication steganography and error-correcting codes. Signal Processing 87 (6), 1528–1533, doi:10.1016/j.sigpro.2006.12.008.
Pevný, T., Bas, P., Fridrich, J., 2010. Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on Information Forensics and Security 5 (2), 215–224, doi:10.1109/TIFS.2010.2045842.
Phadikar, A., Maity, S., 2011. Data hiding based quality access control of digital images using adaptive QIM and lifting. Signal Processing: Image Communication 26, 646–661, Elsevier. doi:10.1016/j.image.2011.07.008.
Qian, Z., Zhang, X., 2012. Lossless data hiding in JPEG bitstream. Journal of Systems and Software 85 (2), 309–313, Elsevier. doi:10.1016/j.jss.2011.08.015.
Qu, Z., Chen, X., Zhou, X., Niu, X., Yang, Y., 2010. Novel quantum steganography with large payload. Optics Communications 283 (23), 4782–4786, Elsevier. doi:10.1016/j.optcom.2010.06.083.
Sajedi, H., Jamzad, M., 2010. Boosted steganography scheme with cover image preprocessing. Expert Systems with Applications 37 (12), 7703–7710, Elsevier. doi:10.1016/j.eswa.2010.04.071.
Wang, Y., Moulin, P., 2008. Perfectly secure steganography: capacity, error exponents, and code constructions, information theory. IEEE Transactions on Information Theory 54 (6), 2706–2722, doi:10.1109/TIT.2008.921684.
Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: from error measurement to structural similarity. IEEE Transaction on Image Processing 13, 600–612, doi:1057-7149/04$20.00.
Westfeld, A., Pfitzmann, A., 2000. Attacks on Steganographic Systems. 3rd International Workshop, Lecture Notes in Computer Science, vol. 1768. Springer-Verlag, Berlin, Heidelberg, New York.
Wu, Y., Shih, F., 2006. Genetic algorithm based methodology for breaking the steganalytic systems. IEEE Transaction on system, Man, and Cybernetica Part B: Cybernetics 36 (1), 24–31, doi:10.1109/TSMCB.2005.852474.
Wu, C., Kao, S., Hwang, M., 2011. A high quality image sharing with steganography and adaptive authentication scheme. The Journal of Systems and Software 84, 2196–2207, Elsevier. doi:10.1016/j.jss.2011.06.021.
Zhang, F., Pan, Z., Cao, K., Zheng, F., Wa, F., 2008. The upper and lower bounds of the information-hiding capacity of digital images. Information Sciences 178, 2950–2959, Elsevier. doi:10.1016/j.ins.2008.03.011.

Nameer N. EL-Emam received BSc. degree in 1985, MSc. degree in 1989 and Ph.D. degree in 1997, all degrees in Computer Science. He works as an assistant professor in the Computer Science Department at Basra University. In 1998, he joins the department of Computer Science, Philadelphia University, as an assistance professor. Now he is an associated professor at the same university, and he works as a chair of computer science department and the deputy dean of the faculty of Information Technology, Philadelphia University. His research interest in Computer Simulation with intelligent system, Parallel Algorithms, and Soft computing using Neural Network, GA, ACO, and PSO for various kinds of applications like Image Processing, Sound Processing, Fluid Flow, and Computer Security (Seteganography).

Rasheed A. AL-Zubidy has completed his PhD in 1997. He works as an assistant professor as a chairman in the Computer Science Department of Baghdad College for Economic Science, Baghdad, Iraq (1998–2004). Associated professor since 30-12-2001. In 2005 he joins the Department of Computer Science, Philadelphia University, as an associated professor. His research interest in Pattern Recognition with intelligent system, and Soft computing using Neural Network, GA, and PSO for many kinds of applications like Image Processing, Sound Processing, and Computer Security (Information Hiding).