Received August 15, 2019, accepted September 5, 2019, date of publication September 13, 2019,
date of current version September 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941440

An Optimized Steganography Hiding Capacity and Imperceptibly Using Genetic Algorithms

RANYIAH WAZIRALI[1], WALEED ALASMARY[2], (Member, IEEE), MOHAMED M. E. A. MAHMOUD[3], (Member, IEEE), AND AHMAD ALHINDI[1], (Member, IEEE)

[1]Department of Computer Science, Umm Al-Qura University, Makkah 21955, Saudi Arabia
[2]Department of Computer Engineering, Umm Al-Qura University, Makkah 21955, Saudi Arabia
[3]Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN 38505, USA

Corresponding author: Waleed Alasmary (wsasmary@uqu.edu.sa)

This work was supported in part by the Postdoctoral Fellowship Initiative from the Ministry of Education that is granted to the PI from Umm Al-Qura University, and in part by the U.S. National Science Foundation under Grant 1618549.

ABSTRACT In stenography, embedding data within an image has a trade-off between image quality and embedding capacity. Specifically, the more data are concealed within a carrier image, the further distortion the image suffers, causing a decline in the resultant stego image quality. Embedding high capacity of data into an image while preserving the quality of the carrier image can be seen as an optimization problem. In this paper, we propose a novel spatial steganography scheme using genetic algorithms (GAs). Our scheme utilizes new operations to increase least significant bits (LSB) matching between the carrier and the stego image which results in increased embedding capacity and reduced distortion. These operations are optimized pixel scanning in vertical and horizontal directions, circular shifting, flipping secret bits and secret data transposing. We formulate a general GA-based steganography model to search for the optimum solutions. Finally, we use LSB substitution for data embedding. We conduct extensive experimental testing of the proposed scheme and compare it to the state-of-art steganography schemes. The proposed scheme outperforms the relevant GA-based steganography methodologies.

INDEX TERMS Data hiding, embedding capacity, evolutionary optimization, genetic algorithms, least significant bits, LSB matching, steganography.

I. INTRODUCTION

Steganography is the science of hiding data within other data [1]. These data could be text, image, audio or video. The main aim of steganography is to conceal a secret data into a host media for confidentiality. Therefore, steganography provides secure communication through multimedia [2], [3]. Moreover, steganography can be used as a tool for multimedia encapsulation, in which one media is inserted in another media. Hence, multimedia encapsulation provides secure transfer of its metadata from one device to another [4]. Figure 1 illustrates the main concept of steganography. As shown in the figure, steganography hides secret data, resulting in the bottom stego image, in a way that is not visible by the human eyes [1].

There are many schemes used for steganography. The schemes can be classified into two major categories, including frequency domain and spatial domain. The first category of schemes is the one that performs computations on the image in the frequency domain. This type of

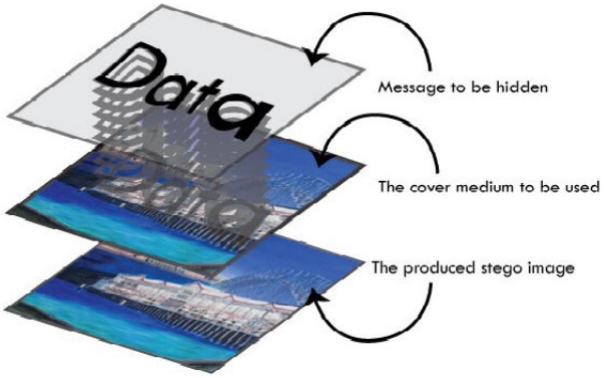


FIGURE 1. An illustration of the main concept of steganography.

The associate editor coordinating the review of this manuscript and approving it for publication was Mehedi Masud.

schemes deploys the image throughout a variety of transforms such as discrete cosine transform (DCT) [5]–[8], discrete wavelet transform (DWT) [9], or discrete fourier transform (DFT) [5], [10]–[13].

The second category of steganography schemes is the spatial domain steganography, in which the image pixels are directly processed in order to embed secret data into the carrier image pixels [14]–[19]. Among the most widely used spatial domain hiding schemes is the least significant bits substitution (LSB) [20]–[22]. LSB features low computational complexity and high embedding capacity. However, LSB-based hiding schemes have certain limitations such as visible degradation of the stego image when embedding large secret data. In this paper, we focus on improving the imperceptibility of LSB approach while guaranteeing high embedding capacity. The vast majority of spatial steganography approaches focus on the embedding procedure without considering the proper position of embedded data. Hence, a few data can be embedded securely.

The success of steganography schemes relies on the delicacy of balancing imperceptibility and embedding capacity. It is always desired to embed the largest possible secret data without degrading visual quality of the stego image. However, the proper selection of the pixels position for embedding data is normally achieved with either scarifying quality and increasing hiding capacity, or preserving quality and scarifying larger hiding capacity. In order to minimize the trade-off, different schemes of data hiding are proposed in literature, and they mainly attempt to search for the best positions to hide data in an imperceptible way [23]–[25]. Moreover, proper selection of pixels based on the LSB matching should ensure minimum distortion to the image. LSB matching is a process of finding more similar LSB values between the carrier image pixels and the stego image pixels. To achieve maximum LSB matching, extensive search process should be applied. Hence, steganography can be viewed as a search and an optimization problem.

In this paper, to achieve an effective level of data embedding, we consider steganography as an optimization problem. We use general genetic algorithms (GAs) to search for the optimum positions to hide LSBs pixels. In our proposed scheme, GA is used to achieve the optimized embedding of the secret data while achieving high embedding capacity and guaranteeing the quality of the stego image.

Our proposed study answers the following questions:

- Can we achieve an effective level of data embedding by considering steganography as an optimization problem?
- Can GA assists to search for proper positions to hide the secret image pixels (in case the secret data is an image) based on maximum LSB matching, which leads to high embedding capacity while preserving the carrier image quality?

In another words, this paper is an attempt to discover an optimized scheme for hiding secret data in a carrier image to increase the quality and the payload capacity. The procedure of mapping the secret data is accomplished by finding the proper positions in the carrier image using various operations. These operations include, scan the carrier image pixels, applying circular shifting, flipping and transposing to hide the secret image with maximum LSB matching.

The rest of the paper is organized as follows. Section II explains the related works including the literature reviews of GA-based steganography. Then, Section III describes the proposed GA-based steganography scheme. After that, the experiment setup and the evaluation results for all of our experiments are presented in Section IV. Finally we draw conclusions in Section V.

II. BACKGROUND AND LITERATURE REVIEW

Section A briefly explains LSB scheme. In Section B, we present a brief overview of GAs. In order to highlight the differences between our work and previous GA-based steganography research, we state the differences between our proposed scheme and other GA-based steganography schemes in Section C.

A. LEAST SIGNIFICANT BITS SUBSTITUTION

LSB substitution has been widely used for embedding secure data into carrier images due to its simple implementation. However, with the increase of hiding capacity, the quality of stego image degrades significantly. LSB is a substitution steganography scheme that substitute the least significant bits from the carrier by the secret bits. The bites are found in the least points of interest. Therefore, this kind of substitution does not leave any remarkable changes.

The fact of the least possible weight is kept by the bits at the rightmost side of the image which is really essential to consider. Figure 2 shows the effect of replacing k LSB bits of the carrier image (a) and by a secret data (b), and the resulting stego image is shown in (c). The extracted secret data is shown in Figure 2(d). The image degradations due to using different number of LSB bits are shown in Figures 2(c) and (d). We can conclude that, as more LSB bits are used, the worse the stego image becomes, and the more the stego recovered secret data is, and vice versa.

B. GENETIC ALGORITHMS

1) OVERVIEW

GAs are a class of heuristic optimization methods that is based on the concept of species evolution [26]. It is based on the concept of reproduction of generations, selection based on fitness parameters/functions, and replacements of the weak offspring. Therefore, GAs are population-based algorithms. That means, instead of beginning with a single solution, a GA starts with a population of solutions, which provide the basis on which the system evolves. Then, every two individuals from this population marry, (i.e., undergo crossover and mutation), and the new individual, i.e., the offspring, exhibits features from each of the parents. Eventually, the offspring becomes a parent. By repeating this process numerous times over different generations, and given certain conditions,

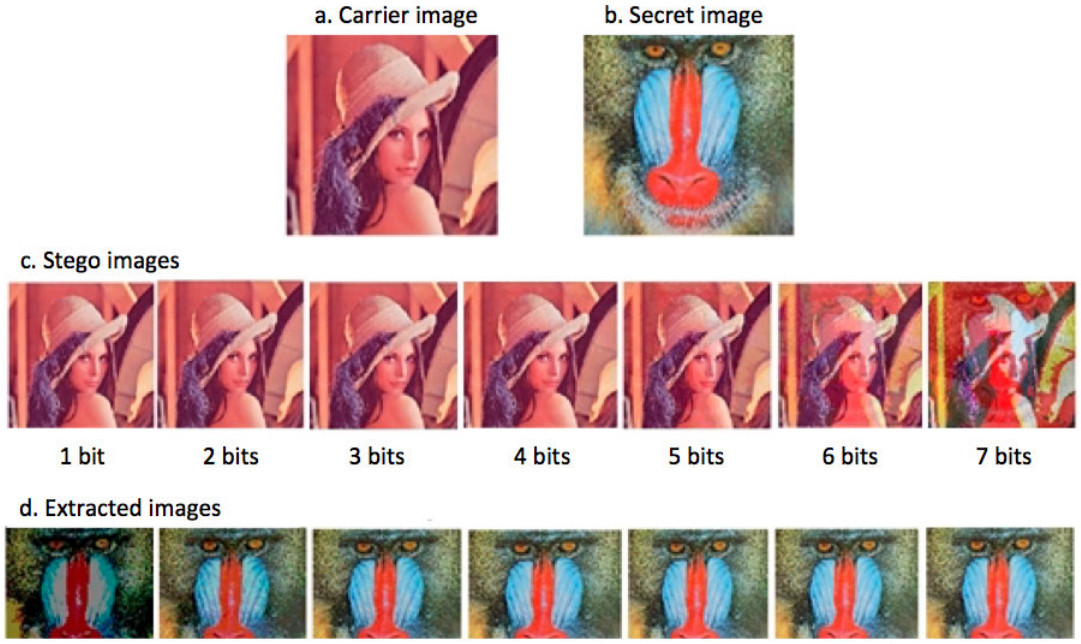


FIGURE 2. LSB substitution degrades the stego image quality as the data to be hidden increases and the number of bits used for embedding increases. a. carrier image, b. secret data, in this case it is an image, c. stego image with different embedding capacity (i.e., different used LSB bits, the more bits are used, the higher is the capacity), and d. the extracted secret images corresponding to different stego images.

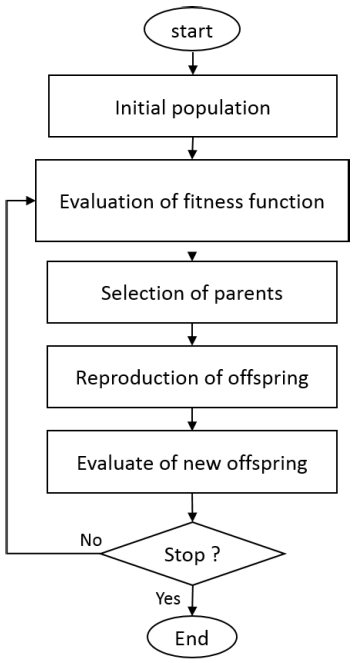


FIGURE 3. A flowchart of a general genetic algorithm.

a GA reaches a globally optimal solution of the optimization problem by reaching an optimal offspring that optimizes the fitness function. The general flowchart of the GA is provided in Figure 3.

GAs incorporate a number of general components, referred to as operators [27], [28]. In order to define a particular GA, these components must be specified. The most important components are listed below:

- Representation
- Evaluation of individual
- Population
- Selection mechanism
- Reproduction operators
- Replacement

Alongside the aforementioned components that are required to define a particular GA, two additional steps are needed, namely, the initialization of the population and stopping criteria.

a: EVALUATION OF INDIVIDUAL

This step helps a GA to apply the role of natural selection in evolution process that occurs in nature. Each individual within a population has an evaluated value to identify itself, i.e., in terms of quality or fitness, among others. We can think of evaluations as scores. The higher individual score is, the fit its descendants becomes to survive an evolution. Thus, some individuals are evaluated with high values and others with low values. Typically, an evaluation function or procedure is required to evaluate individuals and thus the function works as a means of measuring the quality of each individual. The evaluation function is commonly called the fitness function in the GA community.

b: REPRODUCTION (CROSSOVER AND MUTATION)

This operator acts directly on the individuals and is often used after the selection process. The reproduction consists of the following:

- **Crossover**: It mimics biological reproduction. Two individuals (parents) mate and elements from each are combined, producing a new individual with features of both. The crossover occurs with a predefined probability P_c of typical value from 0.6 to 0.9 [29]. A random number R_c is generated in the range [0, 1], and the crossover is applied if $R_c \leq P_C$. Otherwise, the parents proceed without crossover and produce offspring that are exactly similar to them.
- **Mutation**: It is inspired by the mutation of an organisms DNA in natural evolution. Thus, it makes random changes (swaps) to each element of an individual. Mutation happens with a predefined probability P_m of typical value from 0.01 to 0.001 [30].

Note that crossover alone may not generate a new individual. Thus, GA should use mutation in conjunction with crossover.

c: REPLACEMENT

the surviving or replacement mechanism is a crucial component of the evolutionary cycle. Fitter individuals should be selected and would be able to survive to the next generation. This operation follows the reproduction operation and is applied to the offspring. This process decides whether an offspring is replaced or not based on its fitness score.

d: INITIALIZATION AND STOPPING CRITERIA

Initialization and stopping criteria are required to start and end the evolutionary cycle in GAs. On one hand, the objective of initialization is to generate an initial population of individuals. While it tends to be kept simple in most cases, a random method based on a specific problem can also be used. On the other hand, the GA continues the evolutionary cycle until a stopping condition is met. Different stopping conditions exist. The GA can stop after a maximum number of generations, after the optimal solution has been found, or once a population possesses certain characteristics [27].

C. RELATED WORKS ON EVOLUTIONARY STEGANOGRAPHY

GAs have not been widely used in steganography, however, there are a few studies that used GAs to enhance steganogrpahy performance. A review of some of their applications are presented here.

In [31], the authors developed a spatial GA-based steganography scheme. The carrier image is divided into blocks and GA is used to find the most proper start point to hide the secret bits in each block while ensuring minimum stego image quality degradation.

GAs with linear congruential generator have been used to hide secret data [32]. Here, the secret data is embedded in carrier image by finding appropriate positions to embed two bits of the data in each pixel. These coefficients are embedded in the remaining sections of the carrier image using LSB substitution steganography. This scheme provides a secure data hiding however, the hiding capacity is low as it hides only two bits in each pixel.

GAs are also used in securing data in steganographic schemes. Khamrui and Mandal propose a steganographic model based on the concept of GA in the frequency domain [6]. In this scheme, four frequency components are generated through the process of DCT that is applied on a 2×2 sub-mask of the carrier image. Due to the small size of the masks and the low embedding capacity of DCT, this scheme can hide a small size of secret data. By hiding large data, the distortion of the stego become noticeable by human eyes.

Score matrix is a method developed to improve the traditional LSB with the choice of whether to add one or subtract one from the carrier image pixel [33]. In [34], the authors adapt the score matrix strategy and optimize it with GAs. The main reason for using these schemes is to find a near optimal of pairwise LSB scheme based on GA. This scheme uses dual score matrix T_i. When a pixel from the carrier image is similar to another one from the secret image, the score is referred to as T_1, and when the pixels are different the score is referred to as T_2. The objective here is to make T_1 greater than T_2. This method provides a high stego image capacity due to the selection criteria of the chromosomes and the score matrix method.

Nickfarjam and Azimifar propose a scheme for image steganography that uses Particle Swarm Optimization (PSO) and LSB replacement [35]. The process depends on concealing Most Significant Bits (MSBs) of pixels in the secret data (e.g, image) in the LSBs of the carrier image. Four feature functions with four corresponding coefficients are defined to rank the pixels. The basis on which both the features and coefficients are defined is MSBs of the carrier image. Unlike the schemes in [33] and [34], the proposed scheme in [36] is based on image morphing and the main idea is to hide a secret image into a morphed image. The first thing required to produce the natural morphed image is choosing a proper feature point set (FPS). As it is a tedious work to perform this process manually for a large number of possible FPSs, interactive genetic algorithm (IGA) is adopted in the study.

Ghasemi *et al.* propose a steganography scheme using session based stego-key and GA [37]. This scheme is similar to other schemes that use encrypted secret data. This scheme uses transportation operator P to obtain chromosome which is in 16 bits. The aim of the GA here is to determine the best P. The transportation operator P is obtained by dividing the image into 4×1 image blocks and each has 16 bits. This scheme provides a high level of security, with a medium hiding capacity due to the random selection of blocks.

A steganography scheme based on Optimal Pixel Adjustment Process (OPAP) and GA is developed in [38]. This scheme modifies secret bits for attaining more compatibility with the carrier image. Another study uses discrete ripplet transformation (DRT) with adaptive genetic algorithm (AGA) for OPAP to support effective LSB message embedding which led to improved imperceptibility and hiding capacity [39]. A recent study proposes the use of GA in DCT

transform domain to select the best mapping coefficients to improve the performance of data hiding in the frequency domain [40]. GA is also used to choose suitable positions in carrier image using DCT domain in [41]. The secret image is regularized and then embedded in the lower DCT blocks energy of the selected regions. However, only a few data can be embedded due to DCT capacity limitations.

Miri and Faez use mother wavelength matrix to estimate the best frequency space to hide the secret image [42]. GA used to extract proper parameters for the mother wavelength transform by setting the parameters as values for genes to form chromosome in genetic algorithm. As the frequency domain hiding capacity is low, this scheme slightly improves the hiding capacity of DWT.

Another research work has developed a steganography scheme that works on JPEG images [43]. This scheme has two parts. In the first part, an adaptive version of LSB steganography is used which can preserve the stego image statistics. For the second part, to minimize visual degradation of the stego image, there is a need to consider the order of shuffling bits based on chaos, and the parameters of this process is selected by the genetic algorithm. Due to the extreme sensibility of initial conditions and the outspreading of orbits over the entire space, this approach is used for hiding data when the high security is required [43]. However, the quality of the stego image of this scheme is low when hiding large capacity. Another similar study uses the same concept that converts the secret message and the adaptive LSBs to strings and then uses GA for best possible mapping [44]. However, the random selection of bits leads to visible distortion when hiding large message.

Unlike the previous works, our proposed scheme is based on an optimized selection process of the most proper pixels to embed the secret image. This selection uses pixel scanning, pixel shifting in x and y directions, and pixel flipping. The proposed scheme transposes the secret image whenever needed. Followed by finding higher LSB matching by performing an exclusive *OR* (*XOR*) operation for the secret and carrier images representing pixels (in bits format). The bits for *XOR* is either 1 or 0 which is determined by a gene in the chromosome. This type of operations ensures proper selection the pixels which ensures minimum degradation on the stego image quality. To the best of our knowledge, we are the first to propose such an embedding procedure for a steganography scheme.

The work in [34] determines the most proper matching between the stego and carrier images by the use of dual state scoring matrix. This scheme searches for a pixel on the carrier image that matches the secret image pixels and replaces it. Score matrix M is calculated by scanning the secret and the carrier images which takes long time. Then, the scheme uses GA to find a near optimal adjustment list. Our proposed scheme does not need to previously define any score matrix to find the most proper selection of bits. In [6], the embedding is performed in four steps, transformation into frequency, hiding secret data, re-convert into spatial domain using inverse DCT and then applying GA. The use of GA in this scheme does not come in a very late process which does not optimize the results. In [37], the authors use GA and OPAP to obtain an optimal matching between the carrier image and stego image. However, unlike our scheme, the selection of the embedding pixels is based on of $4x4$ random blocks. This random selection results in high distortion. Our scheme is not based on logistic map like in [43] where they modify the carrier image during the embedding by shuffling message bits. GA and chaos are used for shuffling to find the proper shuffle based on logistic map which is the simplest way of chaotic maps. This logistic map was defined randomly which leads to inaccurate results.

III. THE PROPOSED SCHEME

A. OVERVIEW OF THE PROPOSED SCHEME

The proposed scheme takes the carrier and secret images as inputs and performs multiple operations on the pixel level to produce the stego image. In our scheme, we find the best positions to hide the secret image as a coherent unit. Therefore, the secret image is used as a mask that moves as a whole unit around the carrier image while applying different operations to increase the LSB matching between the carrier image pixels and the secret image pixels. The aim of the matching process is to find the most similar pixels which cause reduced distortion. The proposed scheme consists of six major operations, as follows:

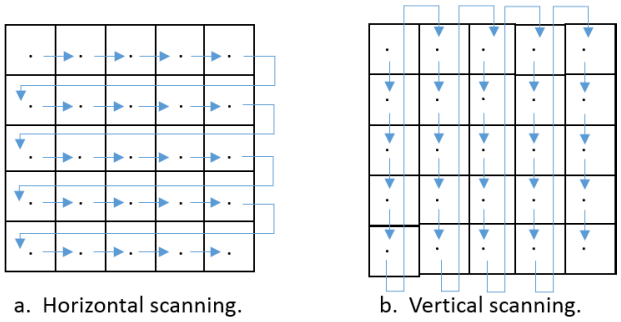
1) **Scanning**: It is used to scan the carrier image in horizontal and vertical directions.
2) **Shifting**: Pixels are shifted in a circular manner so the new position of each pixel is moved to other position.
3) **Flipping**: Secret bits are flipped in opposite direction. The flipping process results in a mirror version of secret bits.
4) **Transposing**: This operation interchanges the row and the column pixels.
5) **LSB matching checking**: It is a process of checking the maximum possible similarities between the carrier's LSB and the secret bits values. The aim is to get more matching bits which leads to minimum distortion.
6) **Secret data embedding**: Upon achieving high LSB matching, we embed the secret image into the LSB of the carrier image pixels coordinates.

The details of the above mentioned pixel level operations are provided in the following section.

B. PROPOSED SCHEME OPERATIONS

1) PIXEL SCANNING

A carrier image is scanned to find the best positions to embed secret image. We propose two directions of image scanning which are (a) horizontal and (b) vertical image pixels scanning [31]. The two schemes can be described in Figure 4. From the figure, the pixel scanning of an image can be a row by row or column by column process.



a. Horizontal scanning. b. Vertical scanning.

FIGURE 4. The horizontal and vertical scanning.

2) PIXEL SHIFTING

Pixels are shifted in a circular manner during which no pixels are lost. However, pixels that are presumed lost in non-circular shifting are fed back to the image here, hence it is named circular pixel shifting. Generally, given a pixel position in an image of (x, y), a shift of p pixels in the x-axis and q pixels in y-axis, results in the pixel new position of $(x + p, y + q)$. The carrier image is considered to be in a coherent form. That means a shift of one pixel affect all of its neighboring pixels in the circle. An illustrative example is given in Figure 5. Figure 5a shows the original positions of the pixels, whereas Figure 5b shows a circular shifting of the same pixels with $p = 2$ and $q = 3$. Figure 5 simply demonstrates the idea of circular shifting without losing pixels. For example, pixel $(1, 5)$ is moved 3 locations circularly in the x-direction and 2 locations in the y-direction, and now its new position is $(3, 3)$.

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

(a) The original pixels position without any shifting.

(4,3)	(4,4)	(4,5)	(4,1)	(4,2)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(1,3)	(1,4)	(1,5)	(1,1)	(1,2)
(2,3)	(2,4)	(2,5)	(2,1)	(2,2)
(3,3)	(3,4)	(3,5)	(3,1)	(3,2)

(b) The circular shifting of the image block in 3 pixels in x direction and 2 pixels in y direction, i.e., pixel (x, y) is shifted to $(x+3, y+2)$.

FIGURE 5. An example of the circular shifting process.

3) PIXELS TRANSPOSITION

Image transposition is similar to matrix transposition. This operation interchanges the rows and columns of image pixels.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

a. Original pixels.

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

b. Transposed pixels.

FIGURE 6. Transposition process.

In transposition, a row is transposed into a column and a column is transposed into a row. In other words, for each pixel (i.e., element in a matrix) with coordinates (i.e., indices) (x, y), its location after transposition will be (y, x). Figure 6 shows an example of an image and the resulting pixels order after the transposition process.

4) FLIPPING SECRET BITS

The bits of the secret pixels are flipped in opposite direction. The flipping process results in a mirrored version of secret bits. In some cases where the matching between the pixels is low, a secret image pixels are flipped into the opposite direction as a chance to increase the similarities. In flipping a pixel, the bits position of the pixel changed from the rightmost to the leftmost. The first (i.e., left most bit) bit becomes in the last bit position (i.e., right most bit), the second bit becomes the in the 7^{th} bit position, the third bit becomes int the 6^{th} bit position, and so on, until the 7^{th} bit becomes in the second bit position, and the last bit (i.e., right most bit) becomes in the first bit position (i.e., left most position). Figure 7 provides a graphical illustration of the flipping process. The top part in the figure shows the original order of the bits and the bottom part indicates the new value of the pixels after the flipping process. For example, assume there is a pixel with a binary representation of 01110010. The flipped version of the pixel would be represented by the binary bits of 01001110.

5) LSB MATCHING CHECKING

LSB matching engages a binary function to check the number of LSB values that are matched between the carrier image and the produced stego image. The aim is to get more similar matching between the carrier image LSB and the produced stego image. The best LSB matching between the carrier and the secret images are computed by applying Exclusive *OR* (*XOR*) logical operation between the two images pixels in binary bits. Best LSB pixels matching is achieved when the value of the carrier image pixels and that of the secret image pixels do not change significantly. We check the similarity of the LSB of carrier image bits with the secret image bits by applying *XOR* operation on the secret bits. *XOR* is a logical operation that returns 0 when inputs are the same otherwise it returns 1. Therefore, the aim is to achieve more zeros than ones to ensure minimum changes. After each of the

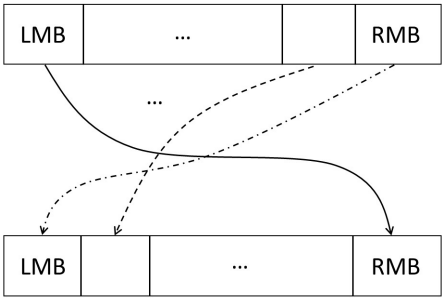


FIGURE 7. An example of the flipping process.

above operations, LSB matching checks the similarities and compares it with fitness function.

6) DATA EMBEDDING

Upon achieving high LSB matching between the secret image bits and the least significant bits of the carrier image, we embed the secret image into the LSB of the carrier image pixels coordinates.

C. ALGORITHM COMPLEXITY

The possible number of combinations can be summarized with respect to the operations as given in Table 1. There are 255 possible solutions on shifting pixels in x directions and 255 on shifting pixels in y directions. Moreover, the directions of embedding are 2 possible solutions as we can embed in vertical or horizontal. The possible solutions of flipping the secret image are 2 as it is possible to flip or not. Finally, there are 2 possible solutions of transposing the secret image as it is possible to transpose the secret image or not. Therefore, there are $255 \times 255 \times 2 \times 2 \times 2$ possible combinations to hide the secret data. Therefore, we employ a genetic algorithm to search for the solution in a near optimum way. The following section explains the used of GA approach.

TABLE 1. Different image operations with all possible combinations.

Operations	Possible number of solutions
Shifting pixel in x direction	255
Shifting image pixel in y direction	255
Direction of embedding	2
Flipping secret bits	2
Transposing secret image	2

D. GA APPROACH

We use a GA to search for a solution by designing a chromosome with genes referring to each of the previously mentioned image processing operations. Initially, the population size has a certain number of chromosomes (i.e., possible solutions). Then, we perform a crossover to produce offspring (i.e., new chromosomes) for the next generations. Each chromosome is made of 5 genes and can be extended to form a chromosome of 255 bits as shown in Table 2. The genes have different representations for the pixel features. Gene A represents the direction of the carrier image, gene B indicates that the shifting operation is in x direction, gene C indicates that the shifting operation in the y direction, gene D represents the flipping mutation on the secret image, gene E indicates the transposing status. The possible states of genes A, D and E are illustrated in Table 3. These genes are set to one if the operations are applied on the secret data pixels; otherwise the genes are set to zero indicating no change to secret data pixels. Genes B and C are converted from binary representation to decimal values for their respective coordinates.

TABLE 2. Five possible genes from 25 bit chromosome.

Genes name	Genes description	Number of bits	Number of possible values
A	Direction of scanning image	1 bit	2
B	Shifting bit in x direction	8 bit	256
C	Shifting bit in y direction	8 bit	256
D	Flipping secret bits	1 bit	2
E	Transpose secret bits	1 bit	2

TABLE 3. The possible state of gene A and gene B.

Gene name	Value	Description
A	1	Vertical pixel scanning
	0	Horizontal pixel scanning
D	1	Secret bits are flipped
	0	No change in secret bit
E	1	Transpose the secret image
	0	No change on the secret image

E. EMBEDDING AND EXTRACTION PROCEDURES

1) SECRET DATA EMBEDDING PROCEDURE

The proposed scheme aims to find the best positions to hide the secret image using the operations of scanning carrier image, shifting secret image, and flipping and transposing secret image. Due to the large search space, we use GAs. The stopping criteria is when the scan reaches the end of the carrier image. To ensure effectiveness of the embedding procedure, the carrier image is ensured to be larger than secret image. The flowchart of secret image embedding procedure is shown in Figure 8. The following steps explain the embedding procedure:

1) The embedding starts by preparing the carrier and secret images, and converting them to binary.

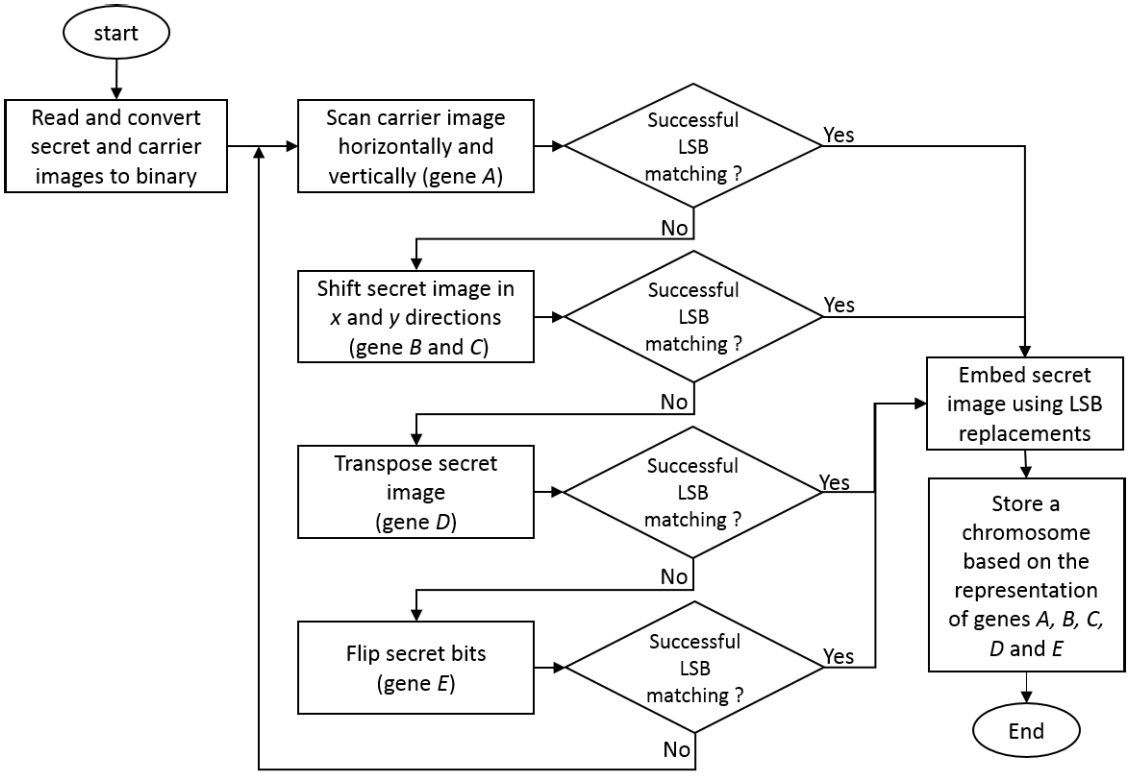


FIGURE 8. The embedding procedure flowchart.

2) Pixel scanning is determined in x and y directions of the carrier image and related direction is stored in gene A.
3) Shifting is performed on the secret image pixels in a circular manner and related coordinates are stored in genes B and C.
4) Secret image bits are flipped and related flipping state is stored in gene D.
5) The secret image is transposed and related transpose state is stored in gene E.
6) Exclusive *OR* operation is performed on secret bits and the carrier LSB bits after each operation. If the fitness value is higher than the previous solution, then we consider this change in the stego image pixel positions as a new solution.
7) If the LSB matching does not attain the required LSB matches between the secret and carrier images pixels given the used fitness function, the above operations are repeated by changing scanning directions by one vertically and horizontally, shifting, flipping and transposing, the secret bits.
8) Upon achieving the fitness function, LSB embedding is executed.
9) A chromosome based on the representation of genes A, B, C, D and E are stored to save the position and the operations that are performed on the secret image.
10) Convert binary secret bit to gray-scale stego image.
11) Calculate the PSNR between the carrier and the stego image (PSNR is considered as the fitness function).
12) These steps are repetitive until the GA produced 200 generations. The fitness function value of the new generation is at least equal to the one of the previous generation which ensures an optimized matching between the carrier and the stego images.

2) SECRET DATA EXTRACTION PROCEDURE

The extraction procedure is the reverse process of the embedding. The flowchart of the secret image extraction from stego image is shown in Figure 9. The following steps explain the extraction procedure:

1) The secret image and the related chromosome are read. The chromosome along with the stego image are needed as inputs to extract the genes that store the operations which generated the the stego image.
2) The chromosome is split into multiple genes. Each gene decides the state of operation required to perform at a particular step.
3) After deciding the states of operations, scanning the stego image in horizontal or vertical directions based on gene A is determined.
4) Pixel shifting is performed in x and y directions based on the values of genes B and C of the stego image.
5) The extracted bits are flipped or transposed based on genes D and E.

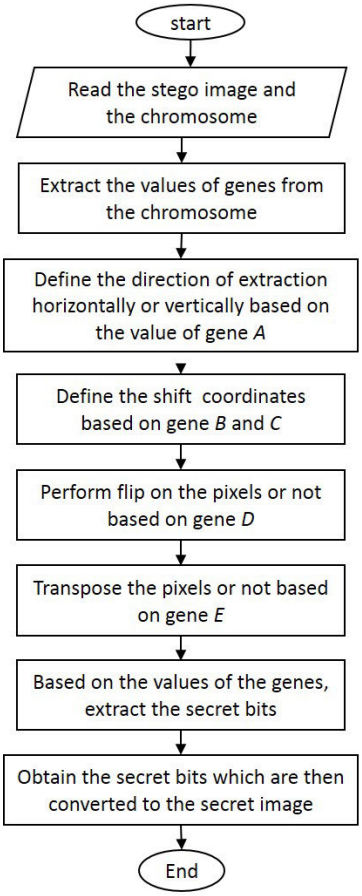


FIGURE 9. The extraction procedure flowchart.

6) Finally, the secret bits are obtained which are then converted to the secret image.

IV. EXPERIMENTAL EVALUATION

The proposed scheme was implemented in Matlab. All the experiments were conducted on a *PC* with *3.2 GHz 6-core 8th-gen i7* processor, and 8 *GB RAM*. All statistical results are averaged over 50 independent runs.

A. PERFORMANCE METRICS

The performance metric that is used in our experimental is Peak signal-to-noise ratio (PSNR). In our paper, we use PSNR to measures the distortion level of each stego scheme as well as testing the quality of stego-images. PSNR basically calculates the changes in pixel intensity between the carrier image and the stego image. Higher value of PSNR indicates that less distortion has occurred on the stego image and consequently the image has better quality. The PSNR metric is defined as

$$PSNR = 10 \times log_{10} \frac{255^2}{MSE},$$

where MSE refers to mean squared error. The PSNR is measured by Decibel (*dB*) which is the measurement unit that expresses the ratio. MSE is a standard statistical measure for the difference between two images. The MSE is defined as

$$MSE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} (x_{ij} - y_{ij})^2,$$

where

- W = width of image in pixel,
- H = height of image in pixel,
- x_{ij} = image pixel at a point (i, j) for carrier image X,
- y_{ij} = image pixel at a point (i, j) for stego image Y.

A small value of MSE indicates that the average difference between images is small and vice versa. The MSE value of two identical images is zero.

B. EXPERIMENTAL SETUP

In our experiments, we consider a set of three carrier images which are Pepper, Baboon and Jet as shown in Figure 10. Moreover, we use the secret image of the Lena with 9 different sizes for evaluating the performance of the proposed scheme. The secret image sizes are 8,192 bits, 12,000 bits, 20,000 bits, 32,768 bits, 65,536 bits, 131,072 bits, 156,800 bits, 180,000 and 524,288 bits. The maximum number of generations is set to 200. Table 4 presents parameters used for the GA part including the population size, crossover rate, mutation rate, and chromosome representation. In the GA part of our implementation, the representation of individual is binary string, and initial population of individuals are randomly selected. For the selection scheme, proportional selection, (i.e., roulette wheels) is chosen. Parents are chosen through a one-point crossover process of the genes with bit mutation.

C. EXPERIMENTAL RESULTS

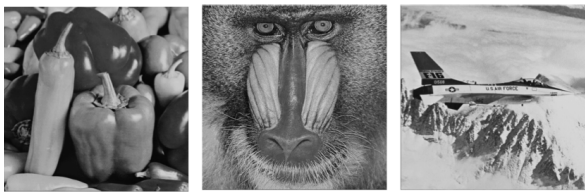
This section discusses the experimental results that evaluate the performance of the proposed GA-based steganography scheme. The effectiveness of the scheme is evaluated using the PSNR. The experimental procedure includes mathematical, visual and statistical test.

To compare the effectiveness of proposed scheme, the scheme has been compared with the LSB scheme by directly inserting secret bit to the LSB in the image pixels as shown in Table 5. This comparison is necessary to assess how the proposed scheme is capable of improving PSNR outcome at different sizes of the secret image.

TABLE 4. GA parameters.

Parameter	Value
Population size	200
Cross over rate	0.7
Mutation rate	0.1

With the parameters presented in Table 4, the improvement during the 200 generations has been shown in Figure 11.



a. Pepper b. Baboon c. Jet

FIGURE 10. The three carrier images used in the experiments.

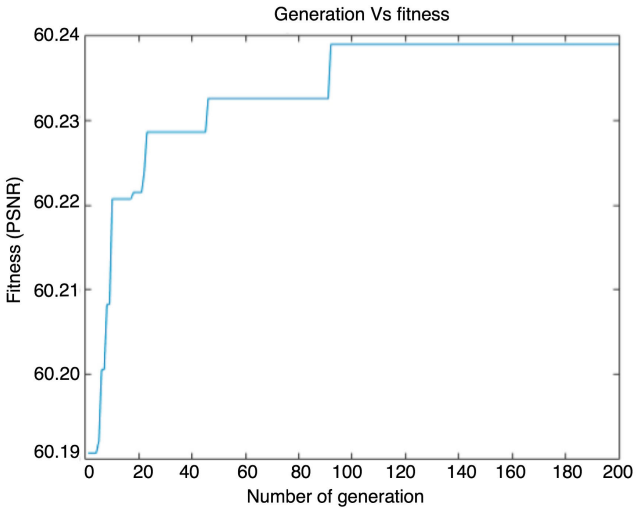


FIGURE 11. The evaluation of the PSNR-metric for 32,768 embedding capacity in GA during 200 generations.

TABLE 5. Comparing the proposed scheme with the state-of-art LSB scheme for different sizes of secret image.

Hiding sizes	LSB	Proposed scheme
8,192 bits	60.2	66.36
32,768 bits	56.14	60.25
80,000 bits	51.15	58.23
131,072 bits	43.64	56.76
156,800 bits	41.61	53.11
180.000 bits	40.86	51.36

The figure demonstrates the improvements in the quality with the increase in the number of generations. Instead of fixing the crossover and mutation rate parameters, The test has been conducted in different crossovers starting from 0.1 to 0.9 with mutation rate of 0.05, 0.1 and 0.2 and the results of hiding 8,192 secret bits performances for all test images are shown in Figure 12. The crossover of 0.7 shows a slightly better performance among other probabilities and the mutation rate of 0.1 provides higher performance.

1) COMPARISON WITH BENCHMARKS

In order to evaluate the performance of the proposed scheme, it has been compared with other adaptive GA-based steganography techniques. It has been compared with the schemes

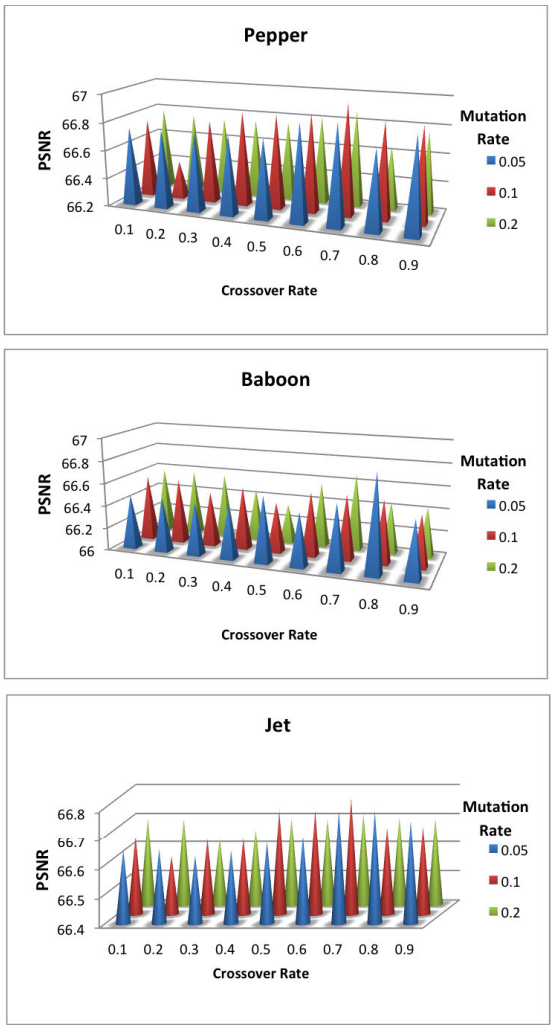


FIGURE 12. PSNR for different crossover and mutation rates for different carrier images.

developed by Yu *et al.* [43], Ghasemi *et al.* [37], Khamrui and Mandal [6], and Soleimanpour-Moghadam and Talebi [34]. The results of the quality analysis have been compared using 180,000 bits hiding capacity for all test images. Table 6 demonstrates the performance of the proposed scheme and compares it with different GA-based steganography schemes. The proposed scheme demonstrates superior higher PSNR values with comparing to the the other schemes. The quality results of our proposed scheme are much higher than [43], [37], and [6]. The PSNR of [34] is high but our proposed scheme provides higher quality results.

2) EMBEDDING CAPACITY EFFECT ON PSNR

steganographic capacity measures the volume of bits acceptable for embedding in a carrier image. Venkatraman *et al.* [45], defined the embedding capacity as the amount of data that can be hidden in the carrier image without any perceptual effect. According to [1], [45], [46], PSNR value of 40 and above is considered acceptable high quality and reflects a closer

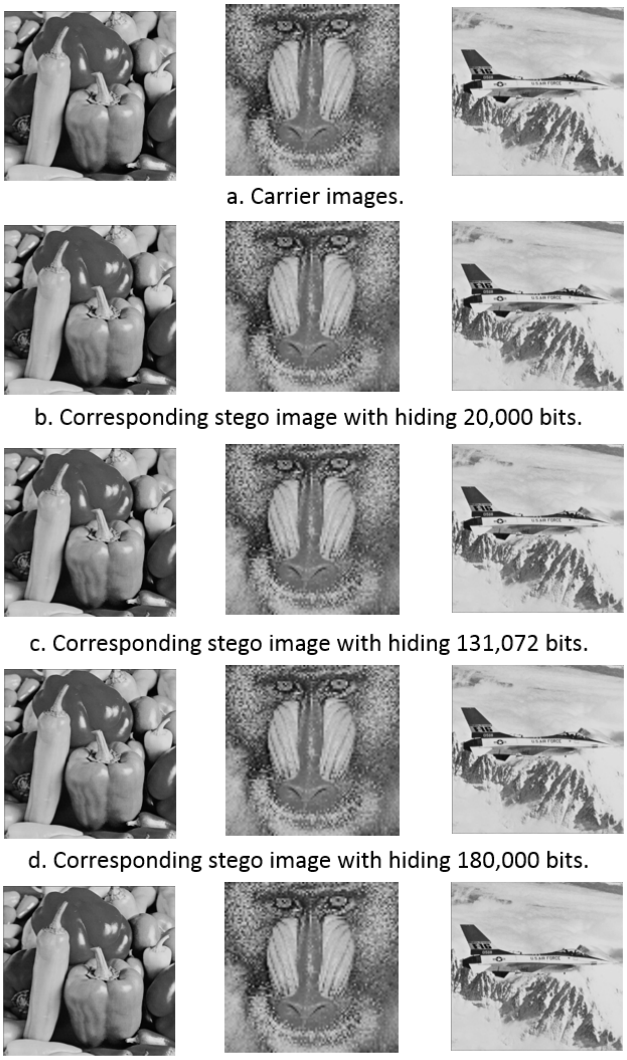


FIGURE 13. Visual analysis of the proposed scheme for different carrier images.

TABLE 6. Comparative performance of the proposed scheme and the different GA-based steganography schemes.

Stego image	Lifang et al. [43]	Ghasemi et al. [37]	Khamrui and Mandal [6]	M. Soleiman-pour et al. [34]	Proposed scheme
Jet	37.65	44.36	46.46	48.54	51.35
Papper	38.35	44.74	46.35	48.57	51.36
Babbon	37.47	44.67	46.25	48.59	51.37

similarity between two images and higher PSNR indicates higher quality. The proposed scheme has been also tested with different hiding capacity in Table 7. It is clearly observed that the proposed scheme provides high quality performance with various hiding capacities. The proposed scheme achieved PSNR of 66.38dB when hiding 8,192 bits, PSNR of 60.8dB when hiding 32,768 bits, and 56.55dB when the hiding 131,072 bits. The proposed scheme attains the minimum PSNR value which is 45.22dB with hiding 524,288 which uses all 4 LSBs. The hiding of this large number of bits indicates a very high hiding capacity. Therefore, the proposed scheme provides high embedding capacity.

TABLE 7. Experimental results of the proposed scheme using various hiding sizes.

Capacity	PSNR		
Number of hidden bits	Jet	Pepper	Baboon
8,192	66.35	66.36	66.46
12,800	65.64	65.54	65.47
20,000	63.54	63.97	63.26
32,768	60.80	60.69	60.44
65,536	59.15	59.08	59.27
131,072	56.55	56.54	56.54
156,800	53.17	53.14	53.16
180,000	51.34	51.36	51.35
524,288	45.23	45.23	45.23

3) VISUAL AND HISTOGRAM ANALYSIS

The visual test has been conducted using the human visual system (HVS) on the original image and embedded image for different carrier image using a secret image. It is observed that the proposed scheme produces a stego image with no visual difference when compared with the original image as shown in Figure 13.

The distribution of pixels was computed in the carrier image and stego image by plotting the histogram of the images as shown in Figure 14. The horizontal axis of the histogram denotes the tonal deviations, while the vertical axis denotes the number of pixels in that specific tone. It is observed that the histograms of the carrier image and stego image provide a negligible difference. The proposed steganography scheme is capable of sustaining statistical attack as secret pixels can never be observed from the histogram.

4) SUMMARY OF THE RESULTS

From these results, we have the following conclusions:

- It is observed that the proposed scheme demonstrates superior to other schemes. The proposed scheme achieves a higher PSNR compared to the traditional LSB and the other GA-based steganography state-of-the-art schemes.
- It is observed that even with the increase of the embedding capacity, the proposed scheme provides high quality stego images. By utilizing small and high embedding capacities of the carrier image pixels, the proposed scheme provides high PSNR values.

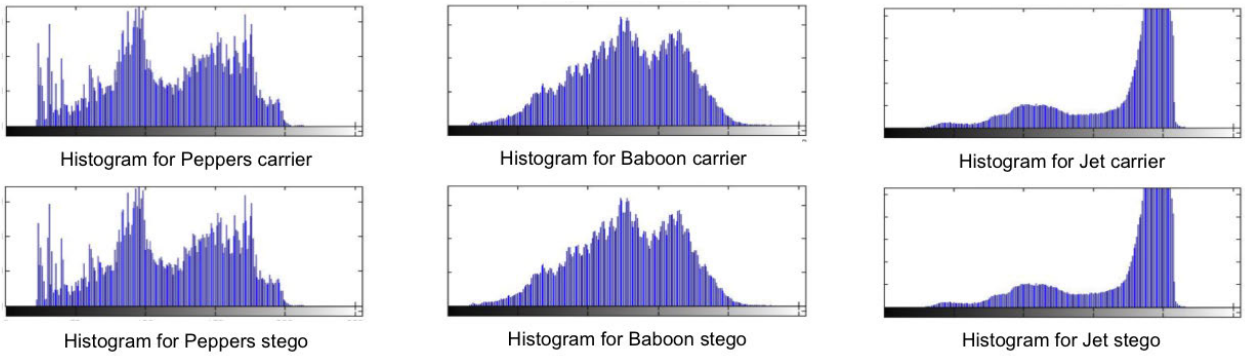


FIGURE 14. Image histograms analysis for three carrier and stego images.

- It is perceived that the degradation of the quality of the stego-images is visually and statically unnoticeable. The distortion on the stego images is invisible to the human visual system. Moreover, the difference between the carrier images and the stego images in terms of the graphical representation of the tonal distribution is unnoticeable as well.

V. CONCLUSION

In this work, a novel way for finding the best LSB matching between the carrier and the stego images has been developed. New concepts of shifting in vertical and horizontal directions, the direction of pixel scanning, secret image transposing, flipping secret bits, and applying *XOR* operation have been used to find the proper bits for hiding data. Due to the large number of search possibilities, a steganography has been considered as an optimization search problem. Therefore, GA is utilized to help in selecting best LSB matching between the carrier and the stego images based on the developed operations. The proposed scheme achieves high embedding capacity and also achieves the desired stego image imperceptibility. Compared to the traditional LSB schemes and multiple GA-based steganography benchmarks, the proposed scheme demonstrates higher PSNR values. Moreover, the proposed scheme is shown to be statistically undetectable, and visually imperceptible by the human eye.

REFERENCES

[1] A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Process.*, vol. 90, no. 3, pp. 727–752, Mar. 2010.

[2] R. Wazirali and Z. Chaczko, "Hyper edge detection with clustering for data hiding," *J. Inf. Hiding Multimedia Signal Process.*, vol. 7, no. 1, pp. 1–10, 2016.

[3] B. E. Carvajal-Gámez, F. J. Gallegos-Funes, A. J. Rosales-Silva, and J. L. López-Bonilla, "Adjust of energy with compactly supported orthogonal wavelet for steganographic algorithms using the scaling function," *Int. J. Phys. Sci.*, vol. 8, no. 4, pp. 157–166, 2013.

[4] N. F. Johnson and S. Katzenbeisser, "A survey of steganographic techniques," in *Information Hiding*. Norwood, MA, USA: Artech House, 2000, pp. 43–78.

[5] H. Noda, M. Niimi, and E. Kawaguchi, "High-performance JPEG steganography using quantization index modulation in DCT domain," *Pattern Recognit. Lett.*, vol. 27, no. 5, pp. 455–461, 2006.

[6] A. Khamrui and J. K. Mandal, "A genetic algorithm based steganography using discrete cosine transformation (GASDCT)," *Procedia Technol.*, vol. 10, no. 1, pp. 105–111, 2013.

[7] S. K. Bandyopadhyay, T. U. Paul, and A. Raychoudhury, "A novel steganographic technique based on 3D-DCT approach," *Comput. Inf. Sci.*, vol. 3, no. 4, p. 229, 2010.

[8] B. Kaur, A. Kaur, and J. Singh, "Steganographic approach for hiding image in DCT domain," *Int. J. Adv. Eng. Technol.*, vol. 1, no. 3, p. 72, 2011.

[9] P.-Y. Chen and H.-J. Lin, "A DWT based approach for image steganography," *Int. J. Appl. Sci. Eng.*, vol. 4, no. 3, pp. 275–290, 2006.

[10] K. Ahrens, S.-F. Chung, and C.-R. Huang, "From lexical semantics to conceptual metaphors: Mapping principle verification with wordnet and sumo," in *Proc. 5th Chin. Lexical Semantics Workshop (CLSW)*, Singapore, 2004, pp. 99–106.

[11] R. Jafari, D. Ziou, and M. M. Rashidi, "Increasing image compression rate using steganography," *Expert Syst. Appl.*, vol. 40, no. 17, pp. 6918–6927, 2013.

[12] X. Wang, Y. Liu, W. Qiu, and D. Zhu, "Immobilization of tetra-tert-butylphthalocyanines on carbon nanotubes: A first step towards the development of new nanomaterials," *J. Mater. Chem.*, vol. 12, no. 6, pp. 1636–1639, 2002.

[13] W.-Y. Chen, "Color image steganography scheme using DFT, SPIHT codec, and modified differential phase-shift keying techniques," *Appl. Math. Comput.*, vol. 196, no. 1, pp. 40–54, 2008.

[14] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. S. Ho, and K.-H. Jung, "Image steganography in spatial domain: A survey," *Signal Process. Image Commun.*, vol. 65, pp. 46–66, Jul. 2018.

[15] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognit.*, vol. 37, no. 3, pp. 469–474, Mar. 2004.

[16] W.-J. Chen, C.-C. Chang, and T. H. N. Le, "High payload steganography mechanism using hybrid edge detector," *Expert Syst. Appl.*, vol. 37, no. 4, pp. 3292–3301, 2010.

[17] A. Ioannidou, S. T. Halkidis, and G. Stephanides, "A novel technique for image steganography based on a high payload method and edge detection," *Expert Syst. Appl.*, vol. 39, no. 14, pp. 11517–11524, 2012.

[18] M. Naor and A. Shamir, "Visual cryptography II: Improving the contrast via the cover base," in *Proc. Int. Workshop Secur. Protocols*. Berlin, Germany: Springer, 1996, pp. 197–202.

[19] H. Sajedi and M. Jamzad, "BSS: Boosted steganography scheme with cover image preprocessing," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7703–7710, 2010.

[20] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[21] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognit. Lett.*, vol. 24, pp. 1613–1626, Jun. 2003.

[22] C. H. Yang, C. Y. Weng, S. J. Wang, and H. M. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 488–497, Sep. 2008.

[23] R. A. Wazirali and Z. Chaczko, "EA based heuristic segmentation for efficient data hiding," *Int. J. Comput. Appl.*, vol. 118, no. 5, pp. 1–7, 2015.

[24] R. A. Wazirali and Z. Chaczko, "Perceptual threshold in DWT for optimum embedding rate in data hiding using HVS and GA," in *Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Nov. 2015, pp. 1–8.

[25] R. A. Wazirali and Z. Chaczko, "Data hiding based on intelligent optimized edges for secure multimedia communication," *J. Netw.*, vol. 10, no. 8, pp. 477–485, Oct. 2015.
[26] C. Darwin, *On the Origin of Species*. London, U.K.: Routledge, 2004.
[27] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, vol. 53. Berlin, Germany: Springer, 2003.
[28] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*. Boca Raton, FL, USA: CRC Press, 2018.
[29] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994.
[30] L. D. Davis, *Handbook of Genetic Algorithms*. New York, NY, USA: Van Nostrand Reinhold, 1991.
[31] H. R. Kanan and B. Nazeri, "A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6123–6130, 2014.
[32] P. D. Shah and R. S. Bichkar, "A secure spatial domain image steganography using genetic algorithm and linear congruential generator," in *Proc. Int. Conf. Intell. Comput. Appl.*, S. Dash, S. Das, and B. Panigrahi, Eds. Singapore: Springer, 2018, pp. 119–129.
[33] J. Mielikainen, "LSB matching revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, May 2006.
[34] M. Soleimanpour-Moghadam and S. Talebi, "A novel technique for steganography method based on improved genetic algorithm optimization in spatial domain," *Iranian J. Elect. Electron. Eng.*, vol. 9, no. 2, pp. 67–75, 2013.
[35] A. M. Nickfarjam and Z. Azimifar, "Image steganography based on pixel ranking and particle swarm optimization," in *Proc. 16th CSI Int. Symp. Artif. Intell. Signal Process. (AISP)*, May 2012, pp. 360–363.
[36] Q. Zhao, M. Akatsuka, and C.-H. Hsieh, "Generating facial images for steganography based on IGA and image morphing," in *Proc. IEEE Int. Conf. Syst., Man, Cybrrn. (SMC)*, Oct. 2012, pp. 364–369.
[37] E. Ghasemi, J. Shanbehzadeh, and N. Fassihi, "High capacity image steganography usingWavelet transform and genetic algorithm," in *Proc. World Congr. Eng.*, London, U.K., vol. 2188, Jul. 2012, pp. 495–498.
[38] L.-Y. Tseng, Y.-K. Chan, Y.-A. Ho, and Y.-P. Chu, "Image hiding with an improved genetic algorithm and an optimal pixel adjustment process," in *Proc. 8th Int. Conf. Intell. Syst. Design Appl.*, vol. 3, Nov. 2008, pp. 320–325.
[39] R. F. Mansour and E. M. Abdelrahim, "An evolutionary computing enriched RS attack resilient medical image steganography model for telemedicine applications," *Multidimensional Syst. Signal Process.*, vol. 30, no. 2, pp. 791–814, 2019.
[40] D. Jude Hemanth, J. Anitha, D. E. Popescu, and L. H. Son, "A modified genetic algorithm for performance improvement of transform based image steganography systems," *J. Intell., Fuzzy Syst.*, vol. 35, no. 1, pp. 197–209, 2018.
[41] S. Joshi and K. V. Sonawane, "Selection of image blocks using genetic algorithm and effective embedding with DCT for steganography," in *Proc. 9th Annu. ACM India Conf.*, 2016, pp. 161–166.
[42] A. Miri and K. Faez, "Adaptive image steganography based on transform domain via genetic algorithm," *Optik*, vol. 145, pp. 158–168, Sep. 2017.
[43] L. Yu, Y. Zhao, R. Ni, and T. Li, "Improved adaptive LSB steganography based on chaos and genetic algorithm," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, 2010, Art. no. 876946.
[44] M. Nosrati, A. Hanani, and R. Karimi, "Steganography in image segments using genetic algorithm," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Technol.*, Feb. 2015, pp. 102–107.
[45] S. Venkatraman, A. Abraham, and M. Paprzycki, "Significance of steganography on data security," in *Proc. Int. Conf. Inf. Technol., Coding Comput.*, Apr. 2004, pp. 347–351.
[46] T.-H. Lan and A. H. Tewfik, "A novel high-capacity data-embedding system," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2431–2440, Aug. 2006.

RANYIAH WAZIRALI received the B.Sc. degree in computer science from Taif University (TU), Taif, Saudi Arabia, in 2008, and the M.Sc. degree in information technology and the Ph.D. degree in software engineering from the University of Technology Sydney, Australia, in 2013 and 2016, respectively. She is currently a Postdoctoral Fellow with Umm Al-Qura University, Saudi Arabia. Her current research interests include cyber security, evolutionary algorithms, data security, steganography, and cryptography.

WALEED ALASMARY received the B.Sc. degree (Hons.) in computer engineering from Umm Al-Qura University, Saudi Arabia, in 2005, the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, Canada, in 2015. During his Ph.D. degree, he was a Visiting Research Scholar with Network Research Laboratory, UCLA, in 2014. He was a Fulbright Visiting Scholar with CSAIL Laboratory, MIT, from 2016 to 2017. He subsequently joined the College of Computer and Information Systems, Umm Al-Qura University, as an Assistant Professor of computer engineering. His mobility impact on the IEEE 802.11p article is among the most cited *Ad Hoc Networks* journal articles list. He published articles in prestigious conferences and journals, such as the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. His current research interests include mobile computing, ubiquitous sensing, and intelligent transportation systems.

MOHAMED M. E. A. MAHMOUD received the Ph.D. degree from the University of Waterloo, in 2011. He was a Postdoctoral Fellow with Ryerson University and the Broadband Communications Research Group, University of Waterloo. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN, USA. His current research interests include security and privacy-preserving schemes for smart grid communication network, mobile ad hoc networks, sensor networks, and delay-tolerant networks. He served as a Technical Program Committee Member for several IEEE conferences. He received two Canadian National Awards; NSERC-PDF and MITACS-PDF. He also received the Best Paper Awards from the IEEE International Conference on Communications (ICC'09) and the IEEE Wireless Communications and Networking Conference (WCNC'16). He is an Associate Editor in Springer journal of *Peer-to-Peer Networking and Applications*. He served as a Reviewer for several IEEE journals and conferences.

AHMAD ALHINDI received the B.Sc. degree in computer science from Umm Al-Qura University (UQU), Makkah, Saudi Arabia, in 2006, and the M.Sc. degree in computer science and the Ph.D. degree in computing and electronic systems from the University of Essex, Colchester, U.K., in 2010 and 2015, respectively. He is currently an Assistant Professor in artificial intelligence (AI) with Computer Science Department, UQU. His current research interests include evolutionary multi-objective optimization and machine learning techniques. He is currently involved in AI algorithms, focusing particularly on machine learning and optimization with a willingness to implement them in a context of decision making and solving combinatorial problems in real-world projects.