Soft Computing (2019) 23:9685–9699
https://doi.org/10.1007/s00500-018-3535-9

Selecting optimal blocks for image watermarking using entropy and distinct discrete firefly algorithm

Elham Moeinaddini[1]

Published online: 22 September 2018

Abstract

Image watermarking is the most promising method for preserving image copyright and its ownership identification. Watermark should have two contradictory properties of transparency and robustness. Location of embedding watermark in the image plays an important role in balancing these two properties. In this paper, a novel robust image watermarking scheme is proposed which uses combination of entropy and distinct discrete firefly algorithm (DDFA) for selecting suitable blocks to balance transparency and robustness. As image blocks numbers are distinct and discrete values, DDFA which is the modified version of firefly algorithm for optimizing distinct and discrete values is proposed and used to select optimal blocks. Hadamard transform applies on each selected block and watermark bits are embedded in Hadamard coefficients using average neighboring coefficients. Using Hadamard transform domain caused more robustness against signal processing attacks. The proposed method has been investigated by various standard metrics and experimental results showed its high robustness and imperceptibility; furthermore, there is a significant balance between these two properties.

Keywords Optimal block selection · Distinct discrete firefly algorithm · Entropy · Robustness · Transparency

1 Introduction

Development of Internet and image editing tools has provided the ability to access, copy and edit digital images. Since copied images have the same quality as the original image, protecting their copyright has become an important challenge. Digital watermarking is a proper method for protecting copyright of digital images. Imperceptibility (transparency) and robustness (resistance) are two basic properties for evaluating watermarking algorithms. These variables are contradicting and increasing one of them reduces the other one. One of the important factors which affect transparency and robustness is the texture of image areas which are selected for embedding watermark. Embedding in smooth areas of the image makes the watermark perceptible while its robustness is increased. However, embedding watermark in areas with sharp edges increases transparency and reduces robustness (Maity and Kundu 2010). Different methods are employed for selecting suitable blocks of the image for embedding watermark. One of the most applicable methods is using visual properties of human's visual system. To this end, concepts like variance, entropy and standard deviation are used to describe texture of blocks. Using these concepts, depending on the selected blocks might increase transparency or robustness of the watermark but does not optimize them (Ali et al. 2015; Kazemivash and Moghaddam 2017; Maity and Kundu 2010; Raihan and Gogoi 2017). Since the goal is to find the suitable blocks such that transparency and robustness are balanced and these blocks are determined dynamically for each image, this issue can be considered as an optimization problem and meta-heuristic methods can be used to solve it. However, recently researchers have employed population-based and swarm intelligence methods (FA, PSO, ACO and ABC) to find thresholds and scaling factors in watermarking techniques and have improved results significantly but so far these methods have not been used to select the proper blocks (Ali et al. 2015; Dong et al. 2015; Ishtiaq et al. 2010; Loukhaoukha et al. 2011; Mishra et al. 2014; Vahedi et al. 2012).

In this paper, a new scheme is proposed for optimal selection of blocks using a combination of entropy and firefly algorithm. Since the block numbers are distinct and discrete

Communicated by V. Loia.

✉ Elham Moeinaddini
el_moin@ujiroft.ac.ir; el_moin@yahoo.com

[1] Department of Electrical Engineering, University of Jiroft, PO Box 364, Jiroft, Iran

values, firefly algorithm should have been modified to generate distinct and discrete values. DDFA is an improved version of firefly algorithm which is proposed in this paper to generate such values.

Another factor which can increase the efficiency of a watermarking scheme is selecting suitable transform domain for embedding the watermark. Analytical and simulation methods show that Hadamard transform provides good transparency while it is robust against most attacks and processes (Maity and Kundu 2011). In embedding process of proposed scheme after selecting proper blocks watermark bits are embedded in Hadamard coefficients using average neighboring coefficients and selected blocks location saves as secret key for extraction.

Hadamard transform as frequency domain and a combination of entropy and DDFA for selecting blocks improve and balance the watermark transparency and robustness. The proposed scheme is novel in case of introducing DDFA for selecting optimal blocks and using new method for embedding watermark in Hadamard coefficients. Furthermore, this scheme achieves good imperceptibility and robustness compared to other existing schemes.

Rest of this paper is organized as follows: Sect. 2 reviews related studies. Section 3 presents employed concepts and algorithms. The proposed DDFA and watermarking method are described in Sect. 4. Section 5 presents experiments and results. Section 6 compares the results of the proposed scheme with existing schemes, and finally the paper is concluded in Sect. 7.

2 Related works

Most important part of the proposed scheme is meta-heuristic selection of blocks of the cover image for embedding watermark. This section describes some of the block-based watermarking schemes and their block selection methods.

Many researchers have used entropy for selecting blocks. In these schemes, visual entropy and edge entropy for each block are calculated using relations defined by Shannon and blocks are selected based on this information (Shannon 1948). Method proposed by Maity et al. divides image blocks based on their entropy to low, medium and high informative blocks (Maity and Kundu 2010). Watermark is once embedded in low and once again embedded in medium informative blocks. Images which are watermarked using this method do not have good transparency and in addition robustness of this method is low. Another method which has used entropy is proposed by Lai (2011). In this method, entropy is calculated for each block and blocks with lower entropy are selected for embedding. Although this scheme offers a good transparency, its robustness against different attacks is not good. Franklin et al. (2011) proposed an informed method for watermarking images using visual entropy. In this method, visual entropy and threshold for each block are calculated and blocks which their entropy is greater than threshold are selected. Robustness of this method is low but its transparency is acceptable. Ali et al. (2015) proposed a blind method for watermarking images in which proper blocks are selected using entropy. In this method, first redistributed invariant wavelet transform is applied to the cover image and its low-frequency sub-band is divided into non-overlapping blocks. Then, blocks with lower entropy are selected for embedding the watermark. SVD is applied to selected blocks, and watermark bits are embedded in first column of left singular matrix using different threshold values. Artificial bee colony algorithm is used for optimizing threshold values. This method has acceptable robustness and transparency but low payload.

In some watermarking schemes, statistical parameters like variance and standard deviation are used for selecting proper blocks. Kazemivash and Ebrahimi Moghaddam (2016) have used standard deviation for selecting proper blocks. In this scheme, after applying lifting wavelet transform to the cover image, low-frequency part of the image is divided into 3×3 blocks. Blocks with higher standard deviation are selected and some of them are ignored to balance transparency and robustness. Neglected blocks are selected by the firefly algorithm. Robustness of this method is good but its transparency is not acceptable. In this scheme, FA is used for finding blocks which should be eliminated and it is not directly used for selecting blocks. In another method proposed by the same authors, standard deviation is again used for selecting blocks (Kazemivash and Moghaddam 2017). In this method, like previous method, after applying lifting wavelet transform to the cover image, its low-frequency section is divided into 3×3 blocks but unlike previous method, blocks with smaller standard deviation are used for embedding the watermark. Transparency of this method is not good but its robustness is high. In the method proposed by Raihan, variance is used for selecting blocks (Raihan and Gogoi 2017). Walsh–Hadamard transform is applied to 8×8 blocks which have higher variance and two watermark bits are embedded in each block at DC coefficients. Although transparency of this method is good, its robustness has not been investigated.

In the scheme proposed by Aung et al. in order to increase transparency, blocks with low amplitude are used for embedding the watermark (Aung et al. 2011). The robustness of this method is high, but its transparency is not so good. The first best search has been employed by Sarker et al. to select a sequence of blocks for watermarking (Sarker and Khan 2013). In this method, after dividing the cover image into 8×8 blocks and applying Hadamard transform, BFS is used to find longest sequence of blocks with largest coefficients (except DC coefficient). Blocking and Hadamard transform are applied to the watermark image too and coefficients of these blocks are multiplied by scaling factor and then replaced by selected blocks of the cover image. Since watermark is embedded in all coefficients except DC coefficient, this method is vulnerable against different noises.

On the whole, based on the related works, given the strength of schemes proposed by Franklin et al. (2011), Ali et al. (2015) and Kazemivash and Ebrahimi Moghaddam (2016) motivate us to use entropy and a swarm intelligence method for selecting appropriate blocks for watermarking. So a powerful optimization algorithm named firefly algorithm is selected and improved for selecting blocks. It is worthwhile to say no watermarking scheme has been used combination of entropy and firefly algorithm for selecting optimum blocks till now.

3 Reviewing concepts and employed algorithms

In this section, concepts and employed algorithms are described briefly.

3.1 Hadamard transform

2D Hadamard transform is an orthogonal non-sinusoidal transform which reduces bandwidth of the signal and consequently reduces space required for storage in addition, and it is a fast transform. 2D Hadamard transform operates based on Hadamard matrix. Hadamard matrix is a $2^n \times 2^n$ matrix including + 1 and − 1 where its columns and rows are orthogonal. Due to orthogonality of rows and columns, Hadamard matrix has special properties. One of these significant properties is shown in the following equation:

$$H_n = H_n^{\mathrm{T}} = H_n^{-1} = H_n^* \tag{1}$$

where H_n is Hadamard matrix, H_n^{T} is transpose of Hadamard matrix, H_n^{-1} is inverse and H_n^* is complex conjugate of Hadamard matrix. For $N = 2$, Hadamard matrix H_1 is called core matrix which is defined as follows:

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2}$$

Hadamard matrix of degree n is constructed using Hadamard matrix of degree $n - 1$ and Kronecker product as follows:

$$H_n = H_{n-1} \otimes H_1 \tag{3}$$

or

$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} \tag{4}$$

And finally, 2D Hadamard transform for an image is computed using the following equation:

$$V = \frac{H_n U H_n}{N} \tag{5}$$

in this equation, U is the original image and V is the transformed image. Since H_n has N orthogonal rows, so $H_n \times H_n = N \times I$, ($I$ is the identity matrix) and

$$H_n \times H_n = N \times H_n \times H_n^{-1} \quad \text{so} \quad H_n^{-1} = H_n/N; \tag{6}$$

therefore, inverse Hadamard transform is calculated as follows:

$$U = H_n^{-1} V H_n^* = \frac{H_n V H_n}{N} \tag{7}$$

In the proposed scheme, Hadamard transform is applied to 8 × 8 blocks using Hadamard matrix of degree 3. This matrix is represented in the following.

$$\mathrm{H}_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \tag{8}$$

3.2 Entropy-based block selection

Entropy is an appropriate measure of spatial correlation of neighboring pixels. Shannon's definition of entropy to calculate the visual entropy of an image is given by (Shannon 1948):

$$E = -\sum_{i=1}^{n} p_i \log p_i \tag{9}$$

where p_i is the probability of occurrence of the event i, $0 \le p_i \le 1$ and $\sum_{i=1}^{n} p_i = 1$. In this equation, the signal is considered as a sequence of symbols. Entropy depends on the relative occurrence of the symbols, irrespective of the positions of occurrence. Obviously, it is necessary to have some information about image features to reach an imperceptible and robust watermark. Pal and Pal (1989) defined average edge information as an exponential form to calculate the entropy that can capture two-dimensional spatial correlation of images better than Shannon's entropy. The edge entropy is defined as:

$$E = \sum_{i=1}^{n} p_i e^{1-p_i} \tag{10}$$

where $1 - p_i$ is the uncertainty of the pixel value i. This definition of entropy gives more information about pixel dispersion and edges of an image. For each block, visual and edge entropy are calculated using Eqs. (9) and (10), respectively, and then summed together. In this study, double of the number of required blocks are selected from lower entropy blocks, and optimal blocks for embedding watermark are selected from these blocks using DDFA.

3.3 Firefly algorithm

Firefly algorithm is a meta-heuristic method, inspired by behavior of firefly which offers good optimal results in most problems. In FA, three following laws are considered as the basis: (1) fireflies are considered to be mono-sexual and each firefly is attracted to other fireflies. (2) Attraction of each firefly depends on its brightness; fireflies with less light are attracted to fireflies with more light. (3) Brightness of each firefly is determined either randomly or by the objective function. Here, performance of FA is described:

Distance between two fireflies i and j which are at location x_i and x_j is calculated using Cartesian distance:

$$r_{ij} = \sqrt{\sum_{k=1}^{D} \left(x_{i,k} - x_{j,k}\right)^2} \tag{11}$$

In this equation, D indicates number of dimensions. Attractiveness of each firefly is calculated using Eq. (12):

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{12}$$

At $r = 0$, attractiveness is β_0 and γ is light absorption. Movement of firefly i toward a brighter firefly j is calculated using Eq. (13):

$$x_i^{\text{new}} = x_i^{\text{old}} + \beta_0 e^{-\gamma r_{ij}^2} \left(x_i^{\text{old}} - x_j\right) + \alpha \left(\text{rand} - \frac{1}{2}\right) \tag{13}$$

Second term of this equation indicates attractiveness of a firefly toward another firefly and randomness of firefly's movement is calculated through parameter α which is the randomness parameter. rand generates random values. X_i^{new} is new location of the ith firefly and $\mathrm{x}_{\mathrm{i}}^{\mathrm{old}}$ is previous location of the firefly. Parameter γ attractiveness and convergence speed. Before implementation, values of α, β and γ should be determined. Pseudo-code of firefly algorithm is shown in Fig. 1.

```
/* Firefly Algorithm*/
Define objective function f(x) where $x = (x_1, x_2, \ldots, x_D)$
Generate initial population of fireflies $x_i$ $(i = 1,2, \ldots N)$
Define $\alpha$, $\beta_0$, maximum generation number and $\gamma$
Begin
while (t<MaxGeneration)
    For each firefly i=1 to N
        For each firefly j=1 to N
            if f(x_i) < f(x_j)
                $r_{ij} = \sqrt{\sum_{k=1}^{D}(x_{i,k} - x_{j,k})^2}$
                $\beta(r) = \beta_0 e^{-\gamma r^2}$
                $x_i^{new} = x_i^{old} + \beta_0 e^{-\gamma r_{ij}^2} \left(x_i^{old} - x_j\right) + \alpha\ (rand - \frac{1}{2})$
            End if
            Attractiveness varies with distance r via $\beta_0 e^{-\gamma r^2}$
            Evaluate new solutions and update objective function
        End for i
    End for j
Rank the fireflies and find the current best
End while
Post process results and visualization
End
```

Fig. 1 Pseudo-code of Firefly Algorithm

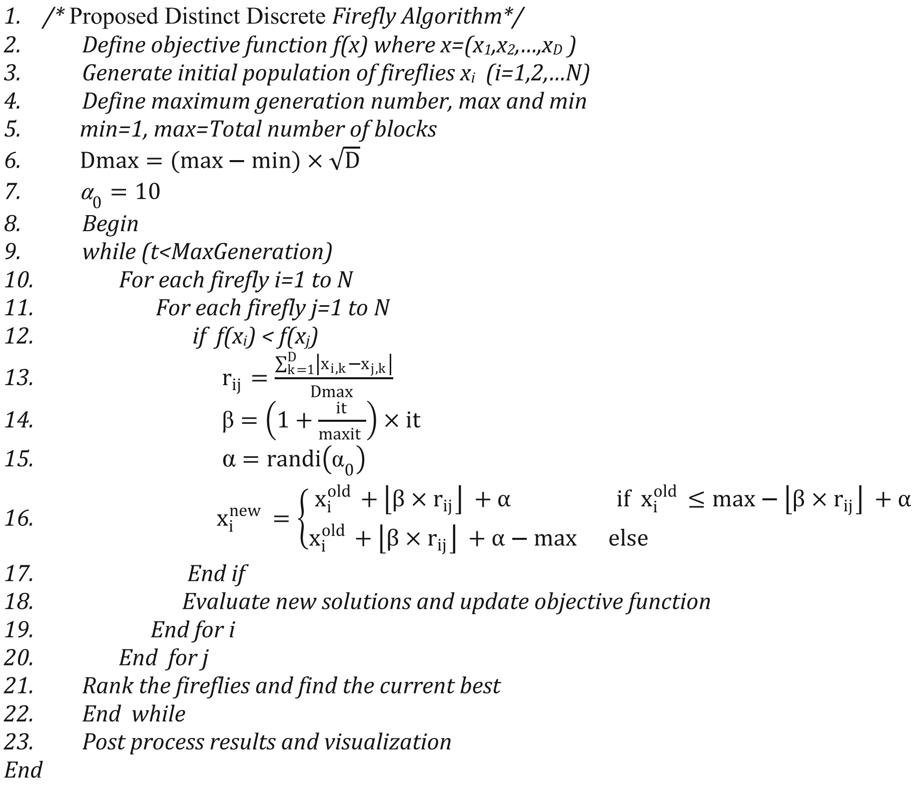
4 The proposed method

In this section, first the proposed distinct discrete firefly algorithm (DDFA) and its objective function are described and then watermark embedding and extracting processes are explained.

4.1 Proposed distinct discrete firefly algorithm (DDFA)

Firefly algorithm was proposed for solving problems with continuous values while these values might be repetitious; therefore, they cannot be employed for selecting proper blocks which are distinct and discrete values. Thus, the original algorithm has been modified so that it can be used for solving this problem. In the proposed algorithm, each firefly introduces a possible sequence of blocks for embedding the watermark. Dimension of each firefly (D) is equal to number of required blocks and their values are distinct integer numbers between min and max; (min is 1 and max is double of the number of required blocks). In order to initialize fireflies instead of random initialization, different permutations of blocks between min and max are generated and first D values of each permutation which have distinct values are used as initial fireflies. Using different permutations of blocks, it can be guaranteed that all dimensions are distinct and non-

Fig. 2 Pseudo-code of proposed distinct discrete firefly algorithm (DDFA)



```
/* Proposed Distinct Discrete Firefly Algorithm*/
    Define objective function f(x) where x=($x_1$,$x_2$,...,$x_D$ )
    Generate initial population of fireflies $x_i$ (i=1,2,...N)
    Define maximum generation number, max and min
    min=1, max=Total number of blocks
    $\text{Dmax} = (\max - \min) \times \sqrt{\text{D}}$
    $\alpha_0 = 10$
    Begin
    while (t<MaxGeneration)
        For each firefly i=1 to N
            For each firefly j=1 to N
                if f(x_i) < f(x_j)
                    $r_{ij} = \frac{\sum_{k=1}^{D}|x_{i,k} - x_{j,k}|}{\text{Dmax}}$
                    $\beta = \left(1 + \frac{\text{it}}{\text{maxit}}\right) \times \text{it}$
                    $\alpha = \text{randi}(\alpha_0)$
                    $x_i^{new} = \begin{cases} x_i^{old} + \lfloor \beta \times r_{ij} \rfloor + \alpha & \text{if } x_i^{old} \le \max - \lfloor \beta \times r_{ij} \rfloor + \alpha \\ x_i^{old} + \lfloor \beta \times r_{ij} \rfloor + \alpha - \max & \text{else} \end{cases}$
                End if
                Evaluate new solutions and update objective function
            End for i
        End for j
    Rank the fireflies and find the current best
    End while
    Post process results and visualization
End
```

repetitive. β is the amount of displacement of each firefly toward the brighter firefly. In this algorithm, the following equation is used for calculating β at each round:

$$\beta = \left(1 + \frac{\text{it}}{\text{maxit}}\right) \times \text{it} \tag{14}$$

it is the current iteration and maxit is maximum iteration numbers. As iterations of the program increases, value of β increases, thus in initial rounds, attraction is less so that search is performed in all spaces of the problem. But in the final rounds, attraction is increased so that solutions converge toward the optimal solution faster.

α determines random displacement of each firefly and is calculated as follows:

$$\alpha = \text{randi}(\alpha_0) \tag{15}$$

randi generates random integer numbers less than α_0 and value of α_0 is considered to be 10 in this work.

Distance between two fireflies i and j is calculated using the following equation:

$$r_{ij} = \frac{\sum_{k=1}^{D}|x_{i,k} - x_{j,k}|}{D\max} \tag{16}$$

In this equation, D is dimension of fireflies and Dmax is calculated as follows:

$$D\max = (\max - \min) \times \sqrt{D} \tag{17}$$

As mentioned before, min is 1 and max is double of the number of required blocks which are selected from lower entropy blocks. Finally, displacement of firefly i toward brighter firefly j is calculated using the following equation:

$$x_i^{\text{new}} = \begin{cases} x_i^{\text{old}} + \beta \times r_{ij} + \alpha & \text{if} x_i^{\text{old}} \le \max - \beta \times r_{ij} + \alpha \\ x_i^{\text{old}} + \beta \times r_{ij} + \alpha - \max & \text{else} \end{cases} \tag{18}$$

In this equation, $\lfloor\ \rfloor$ is the floor function. According to this equation, $\beta \times r_{ij} + \alpha$ which is an integer value is added to all dimension of ith firefly. If by adding this value, some of the dimensions become greater than max, those dimensions are transferred to the beginning of the interval and will not exceed max value. Since dimensions of each firefly is initialized as distinct and discrete values, and while movement a constant integer value is added to all dimensions; it can be guaranteed that dimensions of the new firefly are distinct and discrete. Pseudo-code of DDFA is shown in Fig. 2.

4.2 Objective function and evaluation metrics

Objective function of DDFA is a minimization function and is designed such that fireflies with lower objective function are more attractive. A combination of transparency and robustness is considered as objective function and is calculated using the following equation:

$$f = \frac{\gamma}{\mathrm{PSNR}} + \frac{1}{\mathrm{SSIM}} + \frac{1}{\frac{1}{\mathrm{NA}} \sum_{i=1}^{\mathrm{NA}} \mathrm{NC}_i(w, w\prime)} \tag{19}$$

In this equation, NA is the number of attacks applied to the watermarked image and γ is used for balancing effect of PSNR. Normal correlation (NC) is a measure of similarity between watermark w and extracted watermark w' which evaluates quality of the extracted watermark. NC is calculated under different attacks and their average is used in the objective function. Following equation calculates NC:

$$\mathrm{NC} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} w(i, j).w'(i, j)}{\sum_{i=1}^{m} \sum_{j=1}^{n} (w(i, j))^2} \tag{20}$$

m and n are dimensions of the watermark image. Value of NC is between 0 and 1. For two images if NC is 1, it means that images are completely similar (Lewis 1995).

Quality of the watermarked image is evaluated through PSNR and SSIM. These measures are calculated using the watermarked image while no attack is applied to it. PSNR is a statistical measure of similarity between original image and the manipulated image which is measured in dB. RMSE and PSNR for cover image I and watermarked image $I\prime$ are represented in Eqs. (21) and (22), respectively.

$$\mathrm{RMSE} = \sqrt{\frac{1}{MN} \sum_{M,N} [I(i, j) - I\prime(i, j)]^2} \tag{21}$$

$$\mathrm{PSNR} = 10\log 10\,(R^2/\mathrm{RMSE}) \tag{22}$$

In this equation, M and N are dimensions of the cover image and R is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255.

SSIM is another parameter for measuring quality of an image and is designed such that if two images are more similar, this measure is closer to 1 (Zhang et al. 2011).

$$\mathrm{SSIM}(I, I\prime) = l(I, I\prime).c(I, I\prime).s(I, I\prime) \tag{23}$$

$$\begin{cases} l(I, I\prime) = \frac{2\mu_I \mu_{I\prime} + c_1}{\mu_I^2 + \mu_{I\prime}^2 + c_1} \\ c(I, I\prime) = \frac{2\sigma_I \sigma_{I\prime} + c_2}{\sigma_I^2 + \sigma_{I\prime}^2 + c_2} \\ s(I, I\prime) = \frac{\sigma_{II\prime} + c_3}{\sigma_I \sigma_{I\prime} + c_3} \end{cases} \tag{24}$$

$l(I, I\prime)$, $c(I, I\prime)$ and $s(I, I\prime)$ are similarity of images I and $I\prime$ in terms of luminance, contrast and correlation coefficient, respectively. μ_I and $\mu_{I\prime}$ are mean luminance and σ_I and $\sigma_{I\prime}$ are standard deviation of images I and $I\prime$. $\sigma_{II\prime}$ is covariance between two images which computes using Eq. (25) and c_1, c_2 and c_3 are three positive constants for stabilizing division with small denominator and are calculated using the following equation:

$$\sigma_{II\prime} = \frac{1}{N-1} \sum_{i=1}^{N} (I_i - \mu_I)(I\prime_i - \mu_{I\prime}) \tag{25}$$

$$c_1 = (k_1 L)^2 \tag{26}$$

$$c_2 = (k_2 L)^2 \tag{27}$$

$$c_3 = c_2/2 \tag{28}$$

in which $k_1 = 0.01$, $k_2 = 0.03$ and $L = 255$. k_1 and k_2 values are somewhat arbitrary, but the performance of the SSIM is fairly insensitive to variations of these values (Wang et al. 2004).

4.3 Watermark embedding process

In order to embed a binary image of dimension $n \times n$ in the cover image of dimension $M \times N$, following procedure is performed.

Step 1 Cover image I is divided into 8×8 non-overlapping blocks.

Step 2 According to Sect. 3.2, entropy is calculated for each block and double numbers of required blocks are selected from blocks with lower entropy. 4 bits are embedded in each block. Thus, for embedding a watermark of dimension $n \times n$, $\frac{n^2}{4}$ blocks are required, so $\frac{n^2}{2}$ blocks are selected from lower entropy blocks.

Step 3 Using DDFA Most suitable blocks are selected for embedding watermark. The coordinates (x, y) of the selected blocks are stored in two matrices as secret key for further considerations.

In each block, four watermark bits are embedded in four Hadamard coefficients. Coefficients used in each block are shown in Fig. 3 and represented as $c_{3,2}$, $c_{3,6}$, $c_{7,2}$ and $c_{7,6}$. Steps 4–6 are repeated for each block until all bits are embedded:

Step 4 Hadamard transform is applied to the current selected block and Hadamard coefficients are obtained.

For each of the coefficients $c_{3,2}$, $c_{3,6}$, $c_{7,2}$ and $c_{7,6}$ step 5 is repeated:

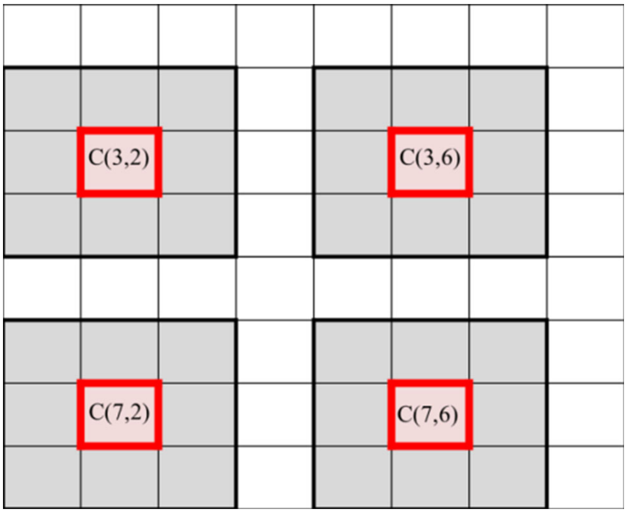


Fig. 3 Position of insertion four watermark bit ($c_{3,2}$, $c_{3,6}$, $c_{7,2}$ and $c_{7,6}$) and their neighboring area (gray color) in a 8 × 8 block

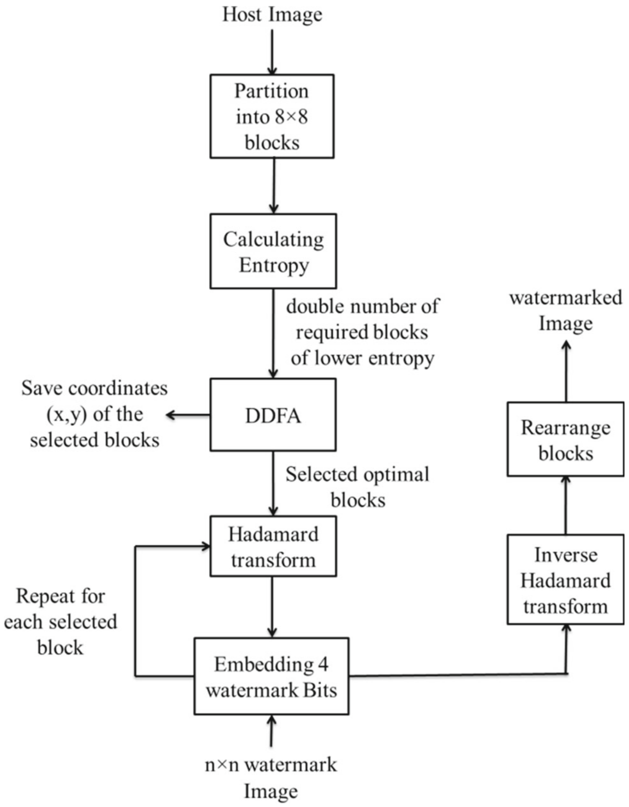

Host Image
Partition into 8×8 blocks
Calculating Entropy
double number of required blocks of lower entropy
DDFA
Save coordinates (x,y) of the selected blocks
Selected optimal blocks
Hadamard transform
Repeat for each selected block
Embedding 4 watermark Bits
n×n watermark Image
Inverse Hadamard transform
Rearrange blocks
watermarked Image

Fig. 4 Watermark embedding process

Step 5 $\mathrm{avg}_{i,j}$ is average of neighboring coefficients for each $c_{i,j}$ and $c'_{i,j}$ is calculated considering current bit of watermark $w(k)$, where $1 \le k \le n^2$:

$$\mathrm{avg}_{i,j} = \Big(c_{i-1,j-1} + c_{i-1,j} + c_{i-1,j+1} + c_{i,j-1} + c_{i,j+1} + c_{i+1,j-1} + c_{i+1,j} + c_{i+1,j+1}\Big)/8 \tag{29}$$

$$c'_{i,j} = \begin{cases} \mathrm{avg}_{i,j} + \mathrm{Th} & \text{if} \quad w(k) = 1 \\ \mathrm{avg}_{i,j} - \mathrm{Th} & \text{if} \quad w(\mathrm{k}) = 0 \end{cases} \tag{30}$$

In this equation, Th is a constant threshold value, $i = 3,\ 7$ and $j = 2,\ 6$. Then, $c_{i,j}$ is replaced by $c'_{i,j}$.

Step 6 Inverse Hadamard transform is applied to the current block.

Step 7 8×8 blocks are put together to get watermarked Image I_{w} .

Embedding processes are depicted in Fig. 4.

4.4 Watermark extraction

In proposed scheme, the extraction process is done blindly and does not require the cover image. Only watermarked image I'_{w} (which might be affected by different attacks) and secret key which determines the coordinates $(x,\ y)$ of the embedded blocks are required. Watermark extraction steps are as follows:

Step 1 Watermarked image I'_{w} is divided into 8×8 equal and non-overlapping blocks.

Step 2 Using the secret key generated at embedding stage which is transferred via a secure channel, indexes of embedded blocks is obtained. For each block, steps 3 and 4 are repeated until all watermark bits are detected:

Step 3 Hadamard transform is applied to the current block and Hadamard coefficients are obtained.

Step 4 For each coefficient $c''_{3,2}$, $c''_{3,6}$, $c''_{7,2}$ and $c''_{7,6}$; average of neighboring coefficients $\mathrm{avg}'_{i,j}$ is calculated using Eq. 29 and watermark bits are detected using the following equation:

$$w'(k) = \begin{cases} 1 & \text{if} \quad \mathrm{avg}'_{i,j} \le c''_{i,j} \\ 0 & \text{if}\ \mathrm{avg}'_{i,j} > c''_{i,j} \end{cases} \tag{31}$$

In this equation,$1 \le k \le n^2$ for a watermark of dimension $n \times n$, $i = 3,\ 7$ and$j = 2,\ 6$.

Step 5 $w'(k)$s is located beside each other and creates the detected watermark image w'.

Extraction processes are depicted in Fig. 5.

5 Experimental results

This section evaluates performance of the proposed watermarking scheme. To this end, 10 standard grayscale images of size 512×512 are used as test images and two binary images of size 64×64 are used as watermark. All test images are adopted from USC-SIPI database(USC-SIPI 2016) and first watermark is logo of University of Jiroft. Figure 6a–j

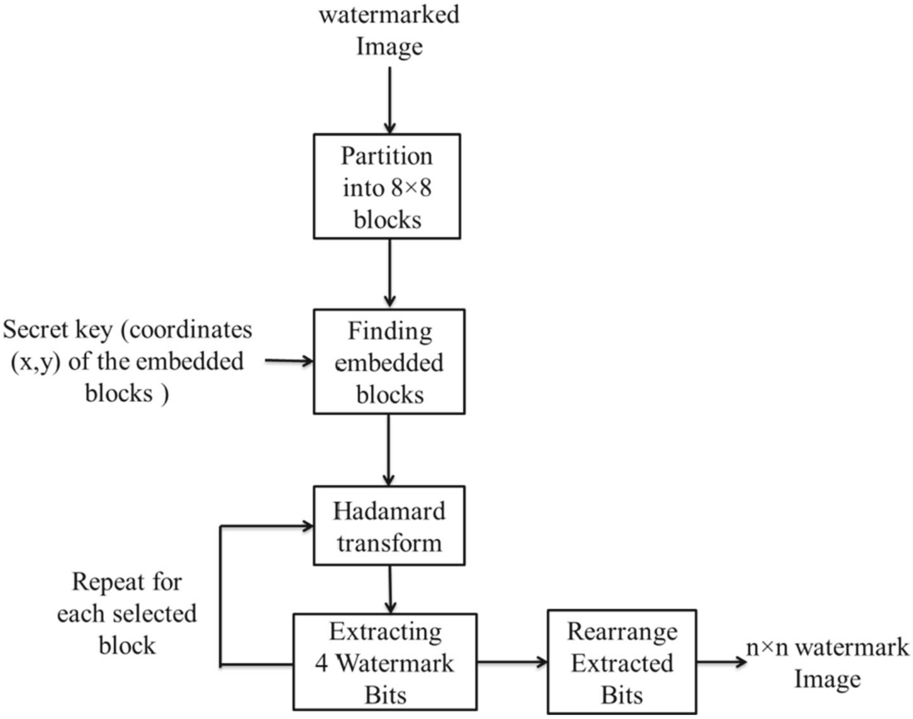

watermarked Image
Partition into 8×8 blocks
Secret key (coordinates (x,y) of the embedded blocks)
Finding embedded blocks
Hadamard transform
Repeat for each selected block
Extracting 4 Watermark Bits
Rearrange Extracted Bits
n×n watermark Image

Fig. 5 Watermark extraction process

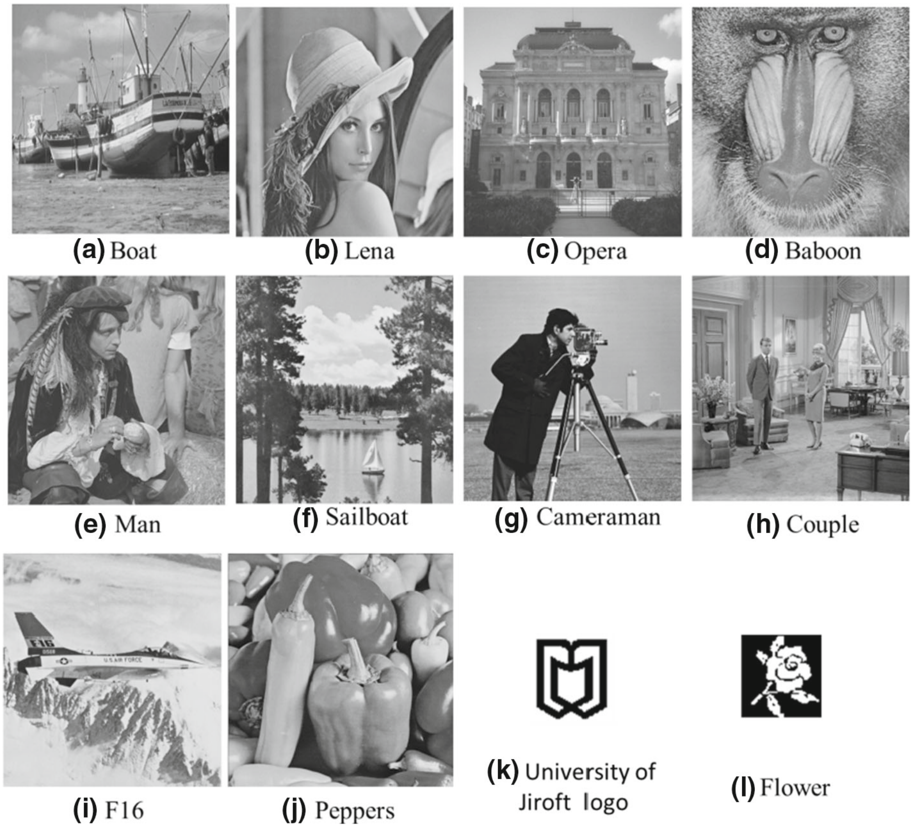


Fig. 6 Standard grayscale test images (**a**–**j**), watermark images (**k**–**l**)

shows test images and (k −l) watermark images. In all executions of DDFA, $\alpha_0 = 10$, number of initial fireflies is 10 and maximum number of iteration is 100. Length of each firefly is one-fourth of watermark bits, that is $\frac{n^2}{4}$, for $n \times n$ watermark. Firefly dimensions are initialized using distinct block indexes between 1 and $\frac{n^2}{2}$ which is double of the number of required blocks with lower entropy. Entropy block selection and DDFA are described in Sects. 3.2 and 4.1, respectively. Also, parameter γ which is used in objective function (Eq. 19) is set to 10. DDFA has been executed 30 times for each image and best results have been selected for experimental analysis. In this paper, a constant threshold value is used for all blocks. Threshold value for each image is selected experimentally; such that mean NC be more than 0.8, the PSNR be greater than 45 and the SSIM be more than 0.99.

In order to evaluate imperceptibility of the proposed scheme, test images are watermarked using two watermark images Fig. 6k, l. PSNR and SSIM are calculated using Eqs. 22 and 23, respectively, while no attack has been applied on the images. Table 1 shows the obtained results. High values of PSNR and SSIM show good quality of the watermarked image.

Quality of the extracted watermark is evaluated using NC. In order to evaluate robustness of the proposed method, 25 attacks are applied to the watermarked image and after extracting the watermark, NC values are calculated. Tables 2 and 3 represent values of NC obtained for 10 test images using two watermarks under various attacks. Considering the obtained results, robustness of the proposed method against most attacks is good which is confirmed by high values of NC. Minimum value of NC is associated with mean filter and since embedding watermark is performed using average neighboring coefficients, robustness against this filter is less than other attacks. Additionally, values of NC without applying attacks are equal to 1 in all cases, which confirm the accuracy of proposed watermark extraction process.

In order to investigate the effect of DDFA algorithm on optimization of robustness and transparency, proposed method run using three different cases of block selection:

Case 1 blocks are selected using DDFA from lower entropy blocks. This method is called DDFA block selection

Case 2 lower entropy blocks are selected for embedding. This method is called entropy block selection

Case 3 random blocks are selected for embedding. This method is called random block selection

Comparing PSNR and SSIM for these cases of selecting blocks for 10 cover images and watermark *k* of Fig. 6 is shown in Fig. 7a and b. Minimum values are obtained when blocks are selected randomly but using entropy and DDFA has improved the results. When DDFA is used, values of PSNR and SSIM are higher compared to two other cases which indicate that transparency of watermark using DDFA is increased. Figure 7c shows comparison of mean NC for different cases. Robustness of the watermark against attacks when blocks are selected randomly is low. In most cases, results of DDFA are better than entropy with a small difference. In fact, using blocks with low entropy generates good

Table 1 Experimental values of PSNR and SSIM for 10 watermarked test images

Watermark images	Test images									
	Boat	Lena	Opera	Baboon	Man	Sailboat	Cameraman	Couple	F16	Peppers
PSNR	0.9943	0.9942	0.9945	0.9964	0.9962	0.9941	0.9932	0.9954	0.9935	0.9941
SSIM	47.877	47.984	48.344	46.410	47.031	47.573	48.465	47.767	47.608	47.261
PSNR	0.993	0.9937	0.993	0.9944	0.9937	0.9937	0.9924	0.9941	0.9928	0.9936
SSIM	47.877	47.984	48.416	46.598	47.702	47.607	48.493	47.931	47.664	47.285

Table 2 Experimental values of NC obtained by taking over the Boat, Lena, Opera, Baboon and Man test images corresponding to the 25 distortion attacks

Image	Boat		Lena		Opera		Baboon		Man	
Attack description										
No attack	1	1	1	1	1	1	1	1	1	1
Median filter (3 × 3)	0.7650	0.7367	0.7763	0.7719	0.7838	0.7506	0.7738	0.7548	0.7672	0.7223
Gaussian filter (3 × 3)	0.9983	1	0.9885	0.9932	0.9966	0.9983	0.9871	0.9937	0.9956	0.9712
Gaussian filter (5 × 5)	0.9983	1	0.9885	0.9932	0.9966	0.9983	0.9868	0.9934	0.9954	0.9712
Average filter (3 × 3)	0.8458	0.8319	0.8159	0.8000	0.8010	0.8052	0.7497	0.7404	0.8683	0.7555
Sharpening filter	0.9993	0.9980	0.9961	0.9934	0.9932	0.9910	0.9890	0.9817	0.9988	0.9751
Histogram equalization	0.9971	1	0.9995	1	0.9890	0.9939	0.9995	0.9995	0.9985	0.9980
Gamma correction (0.2)	1	1	1	1	1	1	1	1	1	0.9983
Gamma correction (0.4)	1	1	1	1	1	1	1	1	1	0.9990
Gaussian noise (0.001)	0.9038	0.8997	0.8916	0.894	0.8906	0.895	0.9031	0.8918	0.8931	0.8987
Gaussian noise (0.005)	0.8715	0.8737	0.8747	0.8754	0.8639	0.8539	0.8708	0.8625	0.8749	0.8678
Gaussian noise (0.009)	0.8190	0.8134	0.808	0.8175	0.8197	0.8102	0.8148	0.831	0.819	0.7980
Salt and pepper noise (0.01)	0.9465	0.9436	0.9507	0.9453	0.9495	0.9507	0.9517	0.9463	0.9431	0.9487
Salt and pepper noise (0.02)	0.9019	0.8967	0.8879	0.8975	0.9133	0.9026	0.9058	0.8999	0.8962	0.9009
Salt and pepper noise (0.03)	0.8486	0.8428	0.8481	0.8538	0.8547	0.8547	0.8564	0.8501	0.8579	0.8518
Cropping 1/4 (center)	0.8730	0.9294	0.9658	0.9802	0.9766	0.9895	0.8112	0.8455	0.8450	1
Cropping 1/4 (top-left)	0.8278	0.8611	0.8408	0.929	0.8747	0.8848	0.894	0.9456	0.9163	0.9292
Cropping 1/4 (bottom–right)	0.8818	0.927	0.8096	0.9014	0.9150	0.9541	0.8083	0.9043	0.8682	0.9158
Scaling 512_1024_512	1	1	1	1	1	1	1	0.9995	1	0.9771
Scaling 512_256_512	0.8892	0.8628	0.9387	0.9121	0.9260	0.9063	0.8444	0.8131	0.8857	0.8073
Rotation (anticlockwise 5°)	0.8518	0.9097	0.9128	0.9485	0.8480	0.8618	0.8535	0.9207	0.8840	0.8298
Rotation (anticlockwise 45°)	1	1	1	1	1	1	1	1	1	1
Rotation (anticlockwise 90°)	0.9656	0.9753	0.9741	0.9792	0.9351	0.9519	0.9700	0.9819	0.9780	0.9695
JPEG compression (70)	0.8950	0.8704	0.8987	0.8752	0.9004	0.8660	0.8469	0.8242	0.8860	0.8501
JPEG compression (80)	0.9539	0.9341	0.958	0.9402	0.9609	0.9351	0.9370	0.9153	0.9556	0.9097
JPEG compression (90)	0.9919	0.9856	0.981	0.9766	0.9795	0.9768	0.9810	0.9741	0.9836	0.9800
Average	0.9240	0.9266	0.9271	0.9337	0.9295	0.9281	0.9128	0.9180	0.9273	0.9163

Table 3 Experimental values of NC obtained by taking over the Sailboat, Cameraman, Couple, F16 and Peppers test images corresponding to the 25 distortion attacks

Image	Sailboat		Cameraman		Couple		F16		Peppers	
Attack description										
No attack	1	1	1	1	1	1	1	1	1	1
Median filter (3 × 3)	0.7860	0.76902	0.74203	0.720099	0.7680	0.7504	0.7619	0.7521	0.7824	0.7736
Gaussian filter (3 × 3)	0.9944	0.99606	1	1	0.9875	0.9915	0.9834	0.9922	0.9905	0.9917
Gaussian filter (5 × 5)	0.9944	0.9963	1	1	0.9875	0.9915	0.9834	0.9922	0.9905	0.9917
Average filter (3 × 3)	0.8715	0.8598	0.8608	0.8459	0.8415	0.8119	0.8181	0.8164	0.8844	0.8781
Sharpening filter	0.9956	0.9944	1	1	0.9875	0.9846	0.9893	0.9805	0.9919	0.9902
Histogram equalization	1	1	1	1	0.9968	0.9985	1	1	1	1
Gamma correction (0.2)	1	1	1	1	1	1	1	1	1	1
Gamma correction (0.4)	1	1	1	1	1	1	1	1	1	1
Gaussian noise (0.001)	0.9021	0.8901	0.8945	0.8958	0.8940	0.8875	0.8904	0.8938	0.9041	0.8994
Gaussian noise (0.005)	0.8571	0.8668	0.8617	0.8705	0.8646	0.8729	0.8617	0.8617	0.8659	0.8686
Gaussian noise (0.009)	0.8178	0.8249	0.8200	0.8041	0.8107	0.818	0.8202	0.8112	0.809	0.819
Salt and pepper noise (0.01)	0.9546	0.9512	0.9495	0.9482	0.9448	0.9446	0.9561	0.9543	0.9497	0.9431
Salt and pepper noise (0.02)	0.9023	0.8997	0.9075	0.9114	0.8979	0.8855	0.908	0.8975	0.8982	0.9004
Salt & pepper noise (0.03)	0.8652	0.8645	0.8657	0.8733	0.8447	0.8545	0.8708	0.8584	0.8679	0.8538
Cropping 1/4 (center)	0.8976	0.9021	0.8499	0.9128	0.8743	0.9309	0.9143	0.9543	0.8682	0.9255
Cropping 1/4 (top-left)	0.8232	0.9075	0.8942	0.8835	0.8206	0.9114	0.9055	0.9509	0.8311	0.9111
Cropping 1/4 (bottom–right)	0.8594	0.9277	0.9727	0.9846	0.8591	0.9285	0.8342	0.9065	0.9248	0.9636
Scaling 512_1024_512	0.9998	1	1	1	0.9998	0.9998	1	1	1	1
Scaling 512_256_512	0.9016	0.8804	0.9465	0.9219	0.8501	0.8193	0.9319	0.9128	0.9163	0.9016
Rotation (anticlockwise 5°)	0.8994	0.9451	0.9346	0.9614	0.8484	0.9131	0.8513	0.9138	0.9006	0.9424
Rotation (anticlockwise 45°)	1	1	1	1	1	1	1	1	1	1
Rotation (anticlockwise 90°)	0.9854	0.989	0.9805	0.9895	0.9575	0.9729	0.9812	0.9824	0.9875	0.9922
JPEG compression (70)	0.8979	0.8667	0.8965	0.8833	0.8804	0.8486	0.9087	0.8809	0.9019	0.8755
JPEG compression (80)	0.9553	0.9321	0.9622	0.9429	0.9546	0.9314	0.9604	0.9375	0.9568	0.9387
JPEG compression (90)	0.9861	0.9851	0.9712	0.9619	0.9883	0.9829	0.9775	0.9668	0.9844	0.9807
Average	0.9287	0.9326	0.9350	0.9354	0.9176	0.9242	0.9272	0.9313	0.9310	0.9361

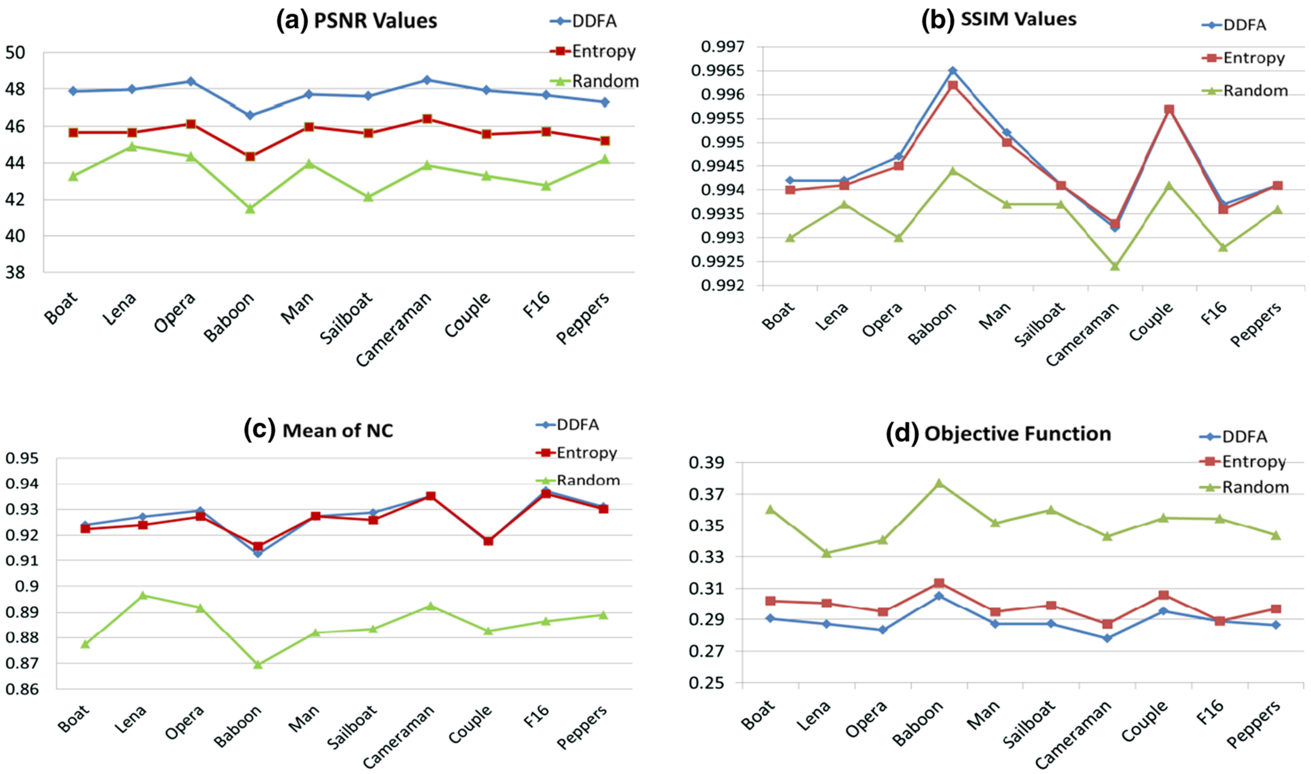


Fig. 7 **a** PSNR, **b** SSIM, **c** mean of NC and **d** objective function values of watermarked test images for three cases of block selection (DDFA, entropy and random)

Table 4 Summarization of the schemes

Attribute	Proposed scheme	Ali et al. (2015)	Lai (2011)	Kazemivash and Moghaddam (2016)	Sarker and Khan (2013)
Block Selection Strategy	Entropy + DDFA	Entropy	Entropy	SD + FA	BFS
Transform Domain	HT	RIDWT + SVD	DCT + SVD	LWT	HT
Host image Size	512 × 512	512 × 512	512 × 512	512 × 512	512 × 512
Watermark Size	64 × 64	32 × 32	32 × 32	32 × 32	64 × 64
Capacity (b/p)	0.0156	0.0039	0.0039	0.0039	0.0156

values of NC and optimizing using DDFA does not affect robustness significantly but it increases transparency by preserving robustness. In Sect. 4.2, it was mentioned that the objective function employed in DDFA is a combination of PSNR and SSIM values before applying attacks and mean NC after applying attacks and it is a minimization function. Value of the objective function for three aforementioned cases is calculated and results are shown in Fig. 7d. When DDFA is used for selecting blocks, objective function has minimum value which verifies that transparency and robustness are increased. Random block selection gives maximum values of the objective function. In general, results show that DDFA block selection is more efficient and effective than two other cases.

6 Comparison and discussion

In the following, results obtained from the proposed scheme using DDFA are compared with results of schemes proposed by Ali et al. (2015), Kazemivash and Moghaddam (2016), Lai (2011), Sarker and Khan (2013). These schemes are briefly described in Sect. 2, and their summarization is given in Table 4. In the schemes proposed by Ali et al. and Lai, entropy is used for selecting blocks (Ali et al. 2015; Lai 2011). Kazemivash and Ebrahimi have used a combination of standard deviation and FA for selecting the blocks and best-first search has also been used by Sarker et al. for block selection (Kazemivash and Moghaddam 2016; Sarker and Khan 2013). For experimental analysis, all cover images are considered to be grayscale of size 512 × 512 and watermark

Table 5 Comparison of NC for various attacks and PSNR, SSIM and Objective function values, with existing methods (Ali et al. 2015; Lai 2011; Kazemivash and Moghaddam 2016; Sarker and Khan 2013) for Lena test image

	Proposed method	Ali et al. (2015)	Lai (2011)	Kazemivash and Moghaddam (2016)	Sarker and Khan (2013)
No attack	1	1	1	1	1
Median filter (3 × 3)	0.7763	0.5522	0.9434	0.8613	0.9854
Gaussian filter (3 × 3)	0.9885	1	0.9895	1	0.991
Gaussian filter (5 × 5)	0.9885	1	0.9893	1	0.9929
Average filter (3 × 3)	0.8159	0.4927	0.8997	0.8727	0.8641
Sharpening filter	0.9961	1	0.969	1	0.9662
Histogram equalization	0.9995	1	0.9822	1	0.999
Gamma correction (0.2)	1	1	0.9768	1	1
Gamma correction (0.4)	1	1	0.9829	1	1
Gaussian noise (0.001)	0.8916	0.999	0.9302	1	0.9759
Gaussian noise (0.005)	0.8747	0.9644	0.7803	0.9013	0.9033
Gaussian noise (0.009)	0.808	0.927	0.7146	0.7955	0.8528
Salt and pepper noise (0.01)	0.9507	0.9775	0.8945	0.9541	0.9805
Salt and pepper noise (0.02)	0.8879	0.9458	0.8193	0.8506	0.9064
Salt and pepper noise (0.03)	0.8481	0.9253	0.7629	0.8325	0.8586
Cropping 1/4 (center)	0.9658	0.8413	0.8489	0.95	0.9794
Cropping 1/4 (top-left)	0.8408	0.9985	0.8513	0.9714	0.96
Cropping 1/4 (bottom–right)	0.8096	0.5081	0.8552	0.9756	0.9519
Scaling 512_1024_512	1	0.9995	0.9985	1	0.9995
Scaling 512_256_512	0.9387	0.824	0.9534	1	0.9054
Rotation (anticlockwise 5°)	0.9128	0.9009	0.8711	0.8134	0.8125
Rotation (anticlockwise 45°)	1	1	1	0.9021	0.9059
Rotation (anticlockwise 90°)	0.9741	0.9932	0.9683	0.7084	0.7113
JPEG compression (70)	0.8987	0.9915	0.999	1	0.9909
JPEG compression (80)	0.958	0.9915	0.999	1	1
JPEG compression (90)	0.981	0.9915	0.999	1	1
Mean(NC) (for all attacks)	0.9271	0.9129	0.9191	**0.9355**	0.9147
PSNR (Without any attack)	**47.9843**	37.214	38.2359	34.9465	38.95
SSIM (Without any attack)	**0.9942**	0.9319	0.962	0.9259	0.9552
Objective function	**0.2987**	0.4205	0.3773	0.4222	0.3595

The best results are represented in bold

Table 6 Mean values of PSNR, SSIM, mean of NC and objective function of the proposed scheme comparing with existing schemes (Ali et al. 2015; Kazemivash and Moghaddam 2016; Lai 2011; Sarker and Khan 2013) for 10 test images

Schemes	PSNR	SSIM	Mean of NC	Objective function
Proposed method	47.7561	0.99456	0.92602	2.2947
Ali et al. (2015)	36.97297	0.94251	0.9101	2.4302
Lai (2011)	38.27468	0.96053	0.9212	2.3879
Kazemivash and Moghaddam (2016)	33.52125	0.92299	0.9277	2.4596
Sarker and Khan (2013)	38.05546	0.95526	0.9318	2.3828

image is shown in Fig. 6k which is a binary image of size 64 × 64. In order to compare robustness of the proposed scheme with other schemes, 25 different attacks are used. Parameters used for simulation are the same as values mentioned in corresponding papers.

Table 5 shows values of NC against various attacks, mean of NC, PSNR, SSIM and Objective function of the proposed scheme compared to four other schemes for Lena test image. Table 6 indicates mean values of PSNR, SSIM, mean of NC and objective function of the proposed scheme comparing with existing schemes for 10 test images (Ali et al. 2015; Kazemivash and Moghaddam 2016; Lai 2011; Sarker and Khan 2013).

High values of PSNR, SSIM and mean of NC indicate high imperceptibility and robustness of watermark using the proposed scheme. Considering the obtained results, mean NC values of the schemes proposed by Sarker et al. and Kazemivash and Ebrahimi are somewhat better than our proposed scheme (Kazemivash and Moghaddam 2016; Sarker and Khan 2013). Objective function of test images is calculated to compare efficiency of the proposed method with other methods in balancing robustness and transparency. Since the objective function is a minimization function, its lower values indicate higher quality. The proposed method generates minimum value of objective function so it can be concluded that, the performance of the proposed scheme in balancing robustness and transparency is better than other schemes.

7 Conclusion

In this paper, a robust watermarking scheme in Hadamard domain is proposed which Selects blocks of the image for embedding watermark through combining entropy and DDFA. In order to embed watermark, cover image is divided into 8 × 8 non-overlapping blocks, then double of the number of required blocks from lower entropy blocks are selected. Among these blocks, required blocks are selected using DDFA algorithm. Each four watermark bits are embedded in one selected blocks of the cover image using mean neighboring coefficients and a constant threshold. Watermark extraction is blind and objective function of DDFA is a combination of transparency and robustness of watermark. Block selection using entropy and DDFA increases transparency and robustness of the watermark. Proposed scheme is novel in case of using entropy and DDFA for selecting optimal blocks. Also, DDFA as an improved version of firefly algorithm which generates distinct and discrete values is proposed in this paper for first time. Robustness of this method is investigated through 25 different attacks, and the results indicate that robustness and transparency of the proposed method are better than other similar schemes. Despite good resistance of watermark against most attacks, its robustness can be improved using multiple threshold values for each block instead of a single value. Improving robustness using this feature and generalizing this method for video can be investigated in future works.

Compliance with ethical standards

Conflict of interest I wish to confirm that there are no known conflicts of interest associated with this article, and there has been no significant financial support for this work that could have influenced its outcome.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent I confirm that I am the only author and there are no other persons who satisfied the criteria for authorship but are not listed. I confirm that I have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing I confirm that I have followed the regulations of my institutions concerning intellectual property.

References

Ali M, Ahn CW, Pant M, Siarry P (2015) An image watermarking scheme in wavelet domain with optimized compensation of singular value decomposition via artificial bee colony. Inf Sci 301:44–60. https://doi.org/10.1016/j.ins.2014.12.042

Aung A, Ng BP, Rahardja S (2011) A robust watermarking scheme using sequency-ordered complex hadamard transform. J Signal Process Syst 64:319–333. https://doi.org/10.1007/s11265-010-0492-7

Dong H, He M, Qiu M (2015) Optimized gray-scale image watermarking algorithm based on DWT-DCT-SVD and chaotic firefly algorithm. In: International conference on cyber-enabled distributed computing and knowledge discovery (CyberC), 17–19 Sept 2015, pp 310–313. https://doi.org/10.1109/cyberc.2015.15

Franklin RV, Manekandan GRS, Santhi V (2011) Entropy based robust watermarking scheme using Hadamard transformation technique. Int J Comput Appl 12:14–21

Ishtiaq M, Sikandar B, Jaffar MA, Khan A (2010) Adaptive watermark strength selection using particle swarm optimization. ICIC Express Lett 4:1–6

Kazemivash B, Moghaddam ME (2016) A robust digital image watermarking technique using lifting wavelet transform and firefly algorithm. Multimed Tools Appl 76:20499–20524

Kazemivash B, Moghaddam ME (2017) A predictive model-based image watermarking scheme using Regression Tree and Firefly algorithm. Soft Comput 22:4083–4098

Lai C-C (2011) An improved SVD-based watermarking scheme using human visual characteristics. Opt Commun 284:938–944

Lewis J (1995) Fast normalized cross-correlation. Vis Interface 1:120–123

Loukhaoukha K, Chouinard J-Y, Taieb MH (2011) Optimal image watermarking algorithm based on LWT-SVD via multi-objective ant colony optimization. J Inf Hiding Multimed Signal Process 2:303–319

Maity SP, Kundu MK (2010) DHT domain digital watermarking with low loss in image informations Aeu. Int J Electron Commun 64:243–257

Maity SP, Kundu MK (2011) Perceptually adaptive spread transform image watermarking scheme using Hadamard transform. Inf Sci 181:450–465

Mishra A, Agarwal C, Sharma A, Bedi P (2014) Optimized gray-scale image watermarking using DWT-SVD and Firefly Algorithm. Expert Syst Appl 41:7858–7867

Pal NR, Pal SK (1989) Object-background segmentation using new definitions of entropy. IEE Proc E Comput Digital Tech 136:284–295

Raihan HK, Gogoi M (2017) A variance based approach (VBA) digital watermarking in frequency domain and comparative analysis using Walsh and Hadamard transform. Int J Comput Appl 163(9)

Sarker MIH, Khan MI (2013) An efficient image watermarking scheme using BFS technique based on hadamard transform. SmartCR 3:298–308

Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423

USC, SIPI (2016) The usc-sipi image database. http://sipi.usc.edu/services/database/database.html

Vahedi E, Zoroofi RA, Shiva M (2012) Toward a new wavelet-based watermarking approach for color images using bio-inspired optimization principles. Digit Signal Proc 22:153–162

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Proc 13:600–612

Zhang L, Zhang L, Mou X, Zhang D (2011) FSIM: a feature similarity index for image quality assessment. IEEE Trans Image Process 20:2378–2386