Multimed Tools Appl (2017) 76:20499–20524
DOI 10.1007/s11042-016-3962-5

A robust digital image watermarking technique using lifting wavelet transform and firefly algorithm

Behnam Kazemivash [1] · **Mohsen Ebrahimi Moghaddam** [1]

Received: 20 December 2015 / Revised: 30 August 2016 / Accepted: 14 September 2016 /
Published online: 28 September 2016

Abstract In this paper, an optimized and robust digital image watermarking technique based on lifting wavelet transform (LWT) and firefly algorithm is proposed. LWT is newer and faster generation of former wavelet transforms and firefly algorithm is an efficient optimizing algorithms. In current technique, base image decomposed by LWT into 4 sub bands then the first sub band separated into non overlapping blocks. After that blocks are sorted in order of descending based on standard derivation of each block. Selecting suitable blocks for special embedding process seems to be an optimization problem due to existence of a trade-off between imperceptibility and robustness. Firefly algorithm used to solve this trade-off while selecting primary blocks causes high robustness and low imperceptibility and vice versa. For improving security, Arnold transform applied to watermark and achieved scrambled image bits used as condition for embedding process. The proposed technique evaluated by variety of attacks like additive noise, average filter, median filter, sharpening filter and some other geometric and non-geometric attacks and experimental results showed its good imperceptibility and high robustness.

Keywords Digital image watermarking · Lifting wavelet transform · Firefly algorithm · Optimization · Arnold transform

1 Introduction

In the last few years, with outspread of cyber technologies, multimedia copy rights become unavoidable domain in business dealings [1, 23, 34]. Therefore, watermarking techniques which are the methods of embedding data into image, audio, and video to prevent illegal use of

✉ Behnam Kazemivash
be.kazemivash@gmail.com

1 Faculty of Computer Science and Engineering, Shahid Beheshti University:G.C, Tehran, Iran

copy files [1, 30] got great attention. It is clear-cut that security and efficiency have the main role in evaluating a watermarking method. On the whole, watermarking techniques are discussed in two different domains:

1 - Spatial Domain which concentrates on applying algorithm to pixels directly. It is easy in implementation but decrease of efficiency caused by complex calculus must be considered [8, 32].
2 - Transform Domain that provides high flexibility and efficiency with low overhead of calculus. However, it must be noticed that locating watermark values into low-frequency ratio may cause huge alternation in base image which is not acceptable. On the other side, embedding that into high frequency ratio steers to a weak point because applying some filters(Attacks) such as low pass filter causes watermark looseness [8, 32]. So as a factor in watermarking process, using transform domain could have lots of effects on results. Many of recent papers, have focused on wavelet based transforms, because it is fast and efficient regarding to other techniques. One of the known wavelet based transforms is DWT (Discrete Wavelet Transform) which is suitable transform domain for watermarking because preferred sub-band can be chosen which causes slight changes in watermarked image [1, 17, 19, 35]. Agoy et al. [1] introduced a novel combination of DWT via chirp z-transform and SVD for image watermarking which shows medium imperceptibility and robustness. As another work, Amiri et al. [3] proposed new visual cryptography based watermarking scheme using DWT and SIFT which shows good imperceptibility and high values for robustness. Also Hu et al. [16] introduced novel scheme based on combination of DWT-DCT for blind image watermarking which shows good capabilities according to metrics. As another wavelet based transforms, the newer version of DWT called LWT became more famous because of fastness and efficiency via other wavelet based transforms [23]. Mehta et al. [24] proposed novel image watermarking scheme based on LWT and GA-LSVR which shows high values in both metrics. As another research, Mehta et al. [25] worked on combination of LWT, QR decomposition and LSVR for image watermarking and results shows high robustness against different kinds of attacks. As another approach, using lossless data hiding technique for reversible watermarking became popular. An et al. [4] proposed a watermarking method using Enhanced Pixel-Wise Masking and Clustering which showed good robustness. Also An et al. [6] presented Content-adaptive reliable method for robust lossless data embedding. In another paper, An et al. [5] developed a statistical quantity histogram Shifting and clustering-based method which shows same good results due to experiments. Furthermore, Gao et al. [12] proposed a novel method for improving reversibility in lossless data hiding with two technique of block skipping scheme and modifying embedding level by a parameter model. Also, Gao et al. [13] developed a framework based on generalized statistical quantity histogram that show same good results.

Up to above discussion, watermarking technique metrics include: imperceptibility that concerns low difference between original and watermarked image and good robustness that points to successful watermark detection even after applying some attacks to watermarked image. These metrics seems to be Consolidation inevitable since you need to use large amount of data to improve robustness that conflicts with imperceptibility [8, 10, 24]. It seems that we need such a trade-off between these two parameters that is reminiscent of optimization problems. Thus as an optimization technique such as genetic algorithms (GA), particle swarm optimization (PSO), differential evolution (DE) may be employed in watermarking applications. There are some strategies numerated to face this kind of issue but in this paper we focus on Firefly algorithm. The main role of optimization phase takes place in choosing suitable blocks for embedding watermark.

Choosing the best blocks in transform domain for embedding process derived to ignore specific percentage of blocks that this percentage named as Ignoring Factor in following. In this case, the Firefly algorithm applied to optimize Ignoring Factor (IF) for embedding the watermark into base medium via lifting wavelet transform.

In this paper, we scrutinize novel method and combination of Lifting Wavelet Transform and Firefly optimization algorithm for digital image watermarking which produces significant results in robustness. Embedding and extracting phase implemented in LWT domain for specific characteristics stated before and firefly algorithm used for optimizing block ignoring factor which balances trade of between imperceptibility and robustness. Remain parts structured as follows. In Section 2, Overview of Lifting Wavelet Transform, Firefly Algorithm, Arnold Transform and evaluation metrics are discussed. The usage of proposed algorithm is introduced in Section 3. Then, experimental results and discussion are presented in Section 4 followed by conclusion and future work in Section 5.

2 Overview of LWT, firefly algorithm, Arnold transformation and evaluation metrics

2.1 Lifting wavelet transform

In recent decade, LWT mentioned by Sweldens [9], becomes a powerful scheme for various applications in the field of image processing such as watermarking [21], image compression [11], pattern recognition [39]. LWT overcomes the defects and limitations of former wavelet transform [23]. On the whole, this strategy contains three fundamental section:

Splitting: This points to dividing primary signal $Z(x)$ into even and odd non-overlapping items of $Z_e(x)$ and $Z_o(x)$.

$$Z_e(x) = Z(2x), Z_o(x) = Z(2x+1) \tag{1}$$

Prediction: After applying correlation between even and odd item, the one can be predictable by other one. It can be considered as high-pass filter operation. For prediction of $Z_o(x)$ we brought $Z_e(x)$ item by:

$$h(x) = Z_o(x) - P(Z_e(x)) \tag{2}$$

Where $h(x)$ shows subtraction between primary signal and predicted rate and $P(.)$ is prediction operator.

Updating: It can be considered as low-pass filter operation which depict coarse figure of primary signal. For updating even items, we hired *h(x)* and operator of updating *U(.)* as follow:

$$l(x) = Z_e(x) + U(h(x)) \tag{3}$$

2.2 Firefly algorithm

Lately, meta-heuristic algorithms has been the focus of academic communities. This kind of algorithms are suitable for conquering optimization problems. FA is one of the meta-heuristic algorithms which proposed by Yang in 2007 [36] for solving optimization problems specially those of multimodal and nonlinear ones. Firefly algorithm inspired from instinctive behavior pattern of fireflies on communicating to each other or finding the mates by bioluminescence process of illustrating natural light [18, 27, 33]. Simplicity, easy implementation, flexibility are strength points of this algorithm [2]. There are some rules we consider for ease of perception [38]:

1. Fireflies attract to each other regardless of sexuality.
2. Attraction depends on brightness. Thus, it can be said, the less bright firefly move to brighter one (In max condition). We'll see random moves in the case of no light or no brighter firefly can be found.
3. Firefly light intensity is specified by optimized objective function value.

In firefly algorithm, we face two essential factors: light intensity alteration and formulizing attraction rate. The attraction rate β is obtained by light intensity *I*. The alternation of attraction rate between firefly *i* and *j* in distance of d_{ij}, occurs on decrease of light intensity caused by distance. Following equation denotes calculus of intensity for a firefly in the simplest form [8].

$$I(d) = I_0 e^{-\gamma d} \tag{4}$$

Where γ is the light absorption coefficient and I_0 is the base intensity. Since attraction rate of a firefly depends on received intensity in adjacent fireflies, the attraction rate β can be specified by

$$\beta(d) = \beta_0 e^{-\gamma d^2} \tag{5}$$

Where β_0 is attraction rate at $d = 0$. We can use Cartesian or Euclidean distance for calculating the distance between a pair of fireflies at position of x_i and x_j as follows:

$$d_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{dim} \left(x_{i,k} - x_{j,k}\right)^2} \tag{6}$$

Where *dim* shows spatial coordinate dimensions. The fireflies movement are consequence of interactions in such a way that low brighter firefly attracted by high brighter one in maximization condition and vice versa in minimization condition. This process continues until the best result achieved. This process formulated by following equation:

$$x_i = x_i + \beta_0 e^{-\gamma d_{ij}^2}\left(x_j - x_i\right) + \alpha\varepsilon_i \quad (7)$$

Where the current location of firefly is shown in the primary term and second one indicates attraction while third one depicted randomization by ε as the vector of random values drawn from a Gaussian distribution and α as randomization parameter. In this case, we assume $\gamma = 1$ for faster convergence of algorithm to optimal result, $\beta_0 = 0.1$ and α in desired range of [0, 1]. It should be noticed that β_0 ranges between [0, 1] where lower bound eventuate to random, distributed and non-cooperative search and on the other side, the upper bound tantamount to cooperation on local domain search. γ represents attractiveness variation which theoretically, ranges in [0, ∞] but practically it must be selected according to the optimization issue. In general, choosing zero value for that shows constant attractiveness and letting it to infinity causes near to zero values for attractiveness which means random search. On the whole, values of these parameters are very significant with regard to impact on algorithm behaviors [7, 28, 37]. The pseudo code for firefly algorithm in finding Min condition is shown in Fig. 1.

2.3 Arnold transformation

For security reasons, applying image scrambling methods are recommended and Arnold transform is one of the powerful and popular one proposed by V.I. Arnold. For a square image, common 2D form of that specialized by equation:

Fig 1 Pseudo code of firefly algorithm

Firefly algorithm for Min Condition

```
Objective function: f(x) , x = (x_1, x_2, x_3,…, x_q)^T
Defining light absorption coefficient γ
Produce primary population of fireflies x_i , i = 1,2,…,n
for i ← 1 to n
| Calculate light intensity I_i from f (x_i ) at x_i
end;
while (t < Max Generation)
| for i ← 1 to n
| | for j ← 1 to i
| | | If (I_i < I_j) then
| | | | Move firefly j toward i
| | | end if
| | | Vary attraction with distance d by exp(-γd^2 )
| | | Evaluate new results and update light intensity
| | end for (j)
| end for (i)
| Rank the fireflies and find the current best
end while
```

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} mod\ N \tag{8}$$

Where N indicates the height of processing image, a and b positive integer values and x_k, y_k are scrambled image peculiarities in accordance with *kth* iteration of x_{k-1} and $y_{k-1.}$ According to special property of Arnold transform, after $T_{n\ (}$image period) iteration position of pixel changes to primary position as depicted in Fig. 2. Mentioned figure shows an instance of applying Arnold transform to image with size of 32 × 32. It is clear cut that for k iteration scrambled image, we need T_n - k iteration to achieve original image.

2.4 Evaluation metrics

As stated in Section 1, we have two aimed metrics to evaluate our technique. These two metrics are imperceptibility which is measured by peak signal to noise ratio (PSNR) specified by equation:

$$PSNR = 10\log_{10}\left(\frac{I_{max}^2}{MSE}\right) \tag{9}$$

Where I_{max} is maximum intensity level and MSE is mean square error specified by equation:

$$MSE = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(I(x,y) - I'(x,y)\right)^2}{M \times N} \tag{10}$$

Where I and I' denotes intensity of pixel(x, y) before and after watermarking and robustness which is measured by normalized correlation (NC) specified by Eq. (11) and Bit error rate (BER) specified by Eq. (12).

$$NC\left(W, W^*\right) = \frac{\sum_{i=1}^{N1}\sum_{j=1}^{N2}\left(W(i,j) \times W'(i,j)\right)}{\sum_{i=1}^{N1}\sum_{j=1}^{N2}\left(W(i,j)\right)^2} \tag{11}$$

$$BER = \frac{\sum_{i=1}^{N1}\sum_{j=1}^{N2}\left(W(i,j) \otimes W'(i,j)\right)}{N1 \times N2} \tag{12}$$

Where *W(i, j)* and *W'(i, j)* represent embedded watermark and extracted watermarked.

Also, for checking visibility and quality of watermarked image there is a subjective method which works with human visual system (HVS) but Sheikh et al. [29] proposed new objective metric named Visual Information Fidelity (VIF) that shows better efficiency and accuracy.

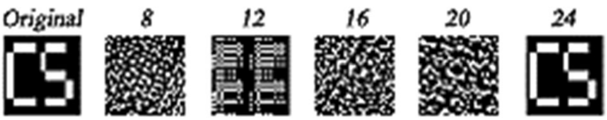

Fig 2 Arnold transform process (*numbers upper images* refer to iteration count)

3 Proposed watermarking technique

In current section, we explain proposed watermarking technique in which include Firefly algorithm implementation details for optimizing ignoring factor, embedding and extracting process of watermarking.

3.1 Watermark embedding process

Step1: Watermark safety via Arnold Transform

Watermark image is converted to binary image which size is $N_w{}^2$ as a square image. We robust safety of our watermark by applying Arnold transform using k iteration (here k = 16) and this process results to scrambled image. After that, scrambled image formed as a row matrix with length of $L_{swi.}$ Matrix elements are binary values which must be used in embedding process. General form of this matrix is shown in following equations:

$$SWI = [be_1, be_2, \ \dots, \ be_{L_{swi}}], \ be_i \in \{0, 1\} \tag{13}$$

Step2: Applying LWT into base image

Apply one level LWT with db2 filter to base image with size of $M \times N$ for decomposing it into four sub-bands *LL, LH, HL* and *HH*. The size of each sub-band can be calculated by following equations:

$$SS = M_L \times N_L \tag{14}$$

$$M_L = M \Big/ 2^k, \ N_L = N \Big/ 2^k \tag{15}$$

Where k refers to decomposition level (Here, $k = 1$).

After that, rip *LL* matrix coefficients of LWT transform into 3 × 3 non-overlapping blocks as shown in Fig. 3.

Step3: Choosing suitable blocks

Standard deviation (SD) of each block must be calculated and sorted in descending order. If we start embedding from first block with most deviation according to embedding technique, results in high robustness but low imperceptibility and vice versa. So in the following, some percentage of primary blocks (achieved by Firefly Algorithm) ignored for balancing trade-off between our two metrics. After applying Ignoring Factor, select (L_{swi}) number of first blocks for embedding process. For extracting phase, position of these block are needed, so we save index of used blocks in a key_vector array. It should be passed to extracting function as an input parameter.

Step4: Watermark embedding

Afterward, for each block we use coefficients of that block to find min/max values. Then we use min and max values to embed watermark data by instruction depicted in Fig. 4. According to value of each scrambled watermark bit, we change central coefficient to min or max value. In the following, we reconstruct LL sub-band by merging changed blocks and applying inverse LWT to get watermarked Image.

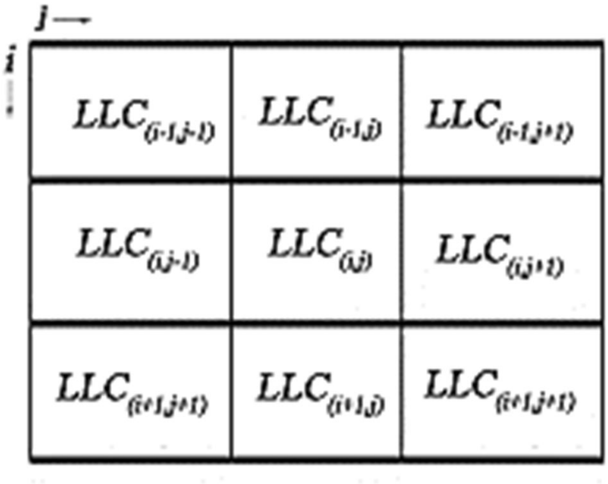

Fig 3 General form of 3 × 3 non-overlapping blocks

3.2 Watermark extracting process

Step1: Applying LWT into base image

Apply one level LWT and make 3 × 3 non-overlapping blocks as mentioned in step 2 from Section 3.1.

Step2: Extracting scrambled watermark data from watermarked image

Select blocks due to key_vector values, then use instruction shown in Fig. 5 to extract hidden values of scrambled watermark. We find max and min coefficient for each block. After that we compare destination of these two values with central coefficient. Lower value of destination shows tendency to min or max. Then we use it as a condition to get scrambled watermark. Main reason for this comparison refers to probability of distortion occurrence for coefficients by attacks.

Step3: Applying Arnold Transform

Reshape resulted row matrix into two dimensional form and apply $(T_n - k)$ time Arnold transform to get watermark.

3.3 Implementation of firefly algorithm to achieve ignoring factor

Step 1: Define *n* fireflies which each one contains decimal random values.

Step 2: Perform watermark embedding process for each firefly.

Fig 4 Proposed watermark embedding instruction

Watermark embedding instruction

```
If (scrambled watermark bit == 1)
  LLC(i,j) = Max (Block Coefficients)
else
  LLC(i,j) = Min (Block Coefficients)
end if
```

Fig 5 Proposed watermark extracting instruction

Watermark extraction instruction

```
for each 3×3 block do
    Dis_1 = | Max (Block Coefficients) - LLC (i, j) |
    Dis_2 = | Min (Block Coefficients) - LLC (i, j) |
    if (Dis_1<Dis_2)
        Watermark Bit_i = 1
    else
        Watermark Bit_i = 0
    end if
end for
```

Step 3: Obtain $M+1$ different watermarked image include one attack free and M attacked ones and compute *PSNR* and *BER* for attack free watermarked image.

Step 4: Produce M watermark from attacked watermarked image using extraction process and compute *NC* and *BER* for each one.

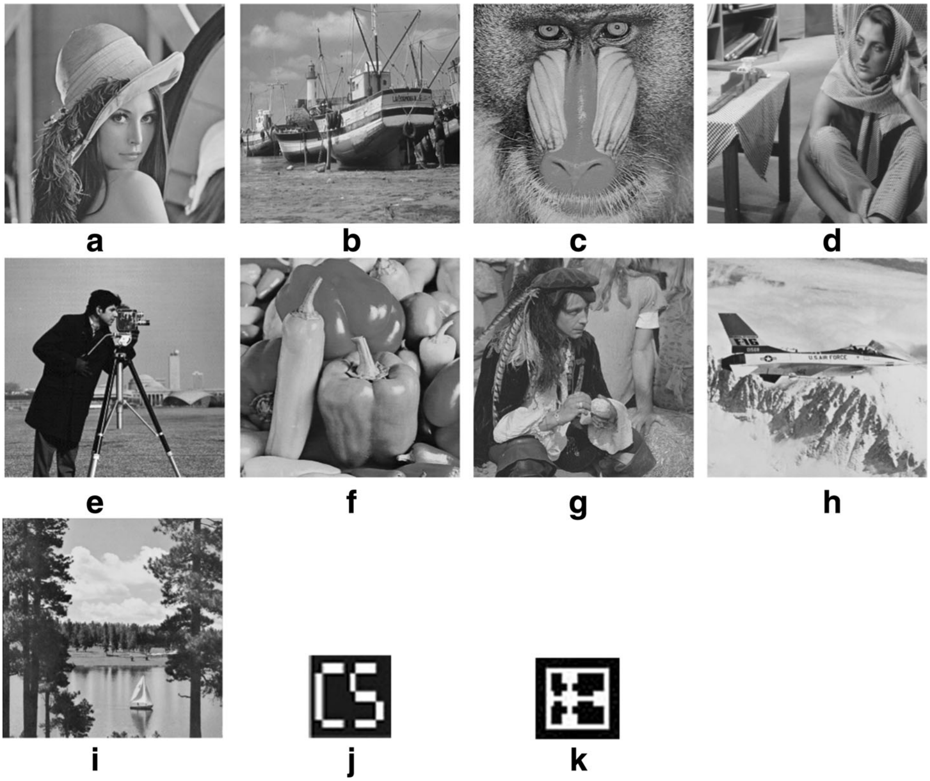

Fig 6 Standard grayscale images of (**a**) "Lena", (**b**) "Boat", (**c**) "Mandrill", (**d**) "Barbara", (**e**) "Cameraman", (**f**) "Pepper", (**g**) "Pirate", (**h**) "Jet Plane" and (**i**) "Lake" with size of 512 × 512 and (**j**) with (**k**) are binary images as 32 × 32 watermarks

Table 1 Experimental values of PSNR, BER and NC for watermarked imag in condition of using WM1 and WM2

Image/Metrics	PSNR (dB)		BER		NC	
	WM1	WM2	WM1	WM2	WM1	WM2
Lena	**37.9815**	37.3697	**0.0035**	0.0036	**0.9977**	0.9973
Boat	**37.7568**	37.5716	**0.0059**	**0.0059**	**0.9975**	0.9974
Mandrill	**33.2136**	33.1430	**0.0067**	0.0068	0.9916	**0.9917**
Cameraman	38.3383	**39.1184**	0.0054	**0.0051**	0.9988	**0.9990**
Peppers	37.1583	**37.7683**	0.0034	**0.0031**	0.9978	**0.9981**
Pirate	**37.9239**	37.8487	**0.0035**	0.0036	**0.9978**	0.9977
Barbara	36.2540	**36.7197**	0.0038	**0.0036**	0.9974	**0.9977**
Jet plane	**37.7658**	36.9398	**0.0026**	0.0030	**0.9975**	0.9969
Lake	35.5835	**35.7566**	0.0034	**0.0032**	0.9979	**0.9980**

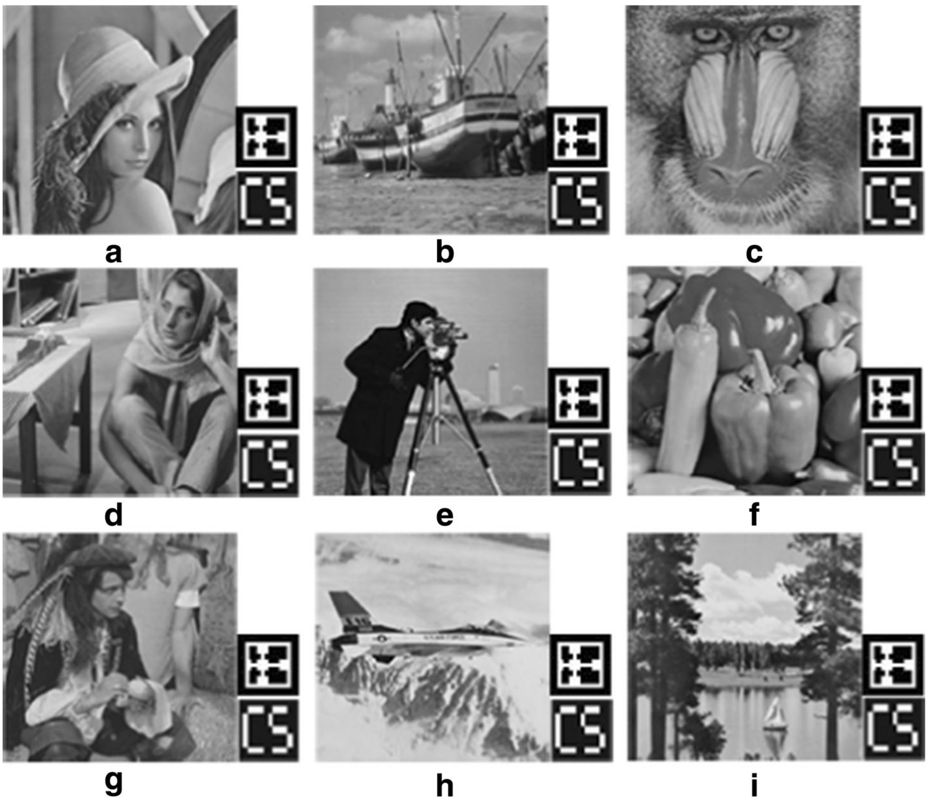

Fig 7 Watermarked images of (**a**) "Lena", (**b**) "Boat", (**c**) "Mandrill", (**d**) "Barbara", (**e**) "Cameraman", (**f**) "Pepper", (**g**) "Pirate", (**h**) "Jet Plane" and (**i**) "Lake" with extracted watermarks (WM1 and WM2) in attack free phase

Table 2 Experimental values of NC and BER of proposed method for extracted watermarks from images of Lena, Boat, Mandrill

Attack/Images		Lena		Boat		Mandrill	
		NC	BER	NC	BER	NC	BER
Attack Free	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Salt & Pepper noise (0.01)	WM1	**0.9609**	**0.0156**	0.9310	0.0273	0.9730	0.0107
	WM2	0.9549	0.0195	**0.9395**	**0.0264**	**0.9841**	**0.0068**
Salt & Pepper noise (0.02)	WM1	0.8776	0.0498	0.8727	**0.0518**	0.9386	0.0244
	WM2	**0.9050**	**0.0420**	**0.8761**	0.0547	**0.9532**	**0.0205**
Salt & Pepper noise (0.03)	WM1	0.8454	**0.0635**	0.8122	0.0762	0.9081	0.0371
	WM2	**0.8473**	0.0664	**0.8550**	**0.0635**	**0.9234**	**0.0332**
Poison noise	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Speckle noise (0.001)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Speckle noise (0.005)	WM1	0.9975	**0.0010**	**1**	**0**	**1**	**0**
	WM2	**0.9977**	**0.0010**	**1**	**0**	**1**	**0**
Speckle noise (0.009)	WM1	0.9825	0.0068	0.9724	**0.0107**	**1**	**0**
	WM2	**0.9886**	**0.0049**	**0.9750**	**0.0107**	**1**	**0**
Gaussian noise (0.001)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gaussian noise (0.005)	WM1	0.9195	0.0322	0.9109	**0.0352**	0.9975	0.0010
	WM2	**0.9360**	**0.0283**	**0.9178**	0.0361	**1**	**0**
Gaussian noise (0.009)	WM1	0.8097	0.0791	0.8011	**0.0801**	0.9730	0.0107
	WM2	**0.8552**	**0.0654**	**0.8060**	0.0840	**0.9819**	**0.0078**
Cropping (Top-Left)	WM1	**0.9850**	**0.0059**	**1**	**0**	**0.9650**	**0.0137**
	WM2	0.9773	0.0098	0.9977	0.0010	0.9615	0.0166
Cropping (Center)	WM1	0.9625	0.0146	0.9575	0.0166	0.9950	0.0020
	WM2	**0.9728**	**0.0117**	**0.9638**	**0.0156**	**0.9977**	**0.0010**
Cropping (Bottom-Right)	WM1	**0.9825**	**0.0068**	0.9499	**0.0195**	**0.9975**	**0.0010**
	WM2	0.9706	0.0127	**0.9501**	0.0215	0.9932	0.0029
Scaling (512-1024-512)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Scaling (512-256-512)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Rotation (90)	WM1	**0.1084**	0.5205	0.1050	0.5039	0.1087	0.4873
	WM2	0.1023	**0.5049**	**0.1055**	**0.4883**	**0.1182**	**0.4854**
Rotation (180)	WM1	0.1015	0.5156	**0.1125**	0.4902	0.0984	**0.4795**
	WM2	**0.1141**	**0.4873**	0.1112	**0.4834**	**0.1024**	0.4844
Gamma correction (0.9)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gamma correction (0.6)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
JPG compression (Q = 80)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
JPG compression (Q = 75)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**

Table 2 (continued)

Attack/Images		Lena		Boat		Mandrill	
		NC	BER	NC	BER	NC	BER
JPG compression (Q = 50)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gaussian filter (3 × 3)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gaussian filter (5 × 5)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Average filter (3 × 3)	WM1	0.9727	0.0107	0.9365	0.0254	0.9925	0.0029
	WM2	**0.9773**	**0.0098**	**0.9639**	**0.0156**	**0.9977**	**0.0010**
Median filter (3 × 3)	WM1	**0.9013**	**0.0391**	0.8025	0.0811	**0.9161**	**0.0342**
	WM2	0.8756	0.0537	**0.8534**	**0.0645**	0.8960	0.0459
Histogram Equalization	WM1	**1**	**0**	**0.9975**	**0.0010**	0.9975	0.0010
	WM2	**1**	**0**	0.9955	0.0020	**1**	**0**
Sharpening filter	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Wiener filter (3 × 3)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**

Step 5: Compute objective function depicted in Eq. 16 for current firefly.

$$\mathrm{OF} = \left[\omega \times \left(BER + \frac{1}{PSNR}\right)\right] + \left[\left(\frac{1}{M} \times \sum_{k=1}^{M} \frac{1}{NC(W, W')_k}\right) + \left(\frac{1}{M} \times \sum_{k=1}^{M} BER_k\right)\right] \quad (16)$$

Where first bracket is for attack free phase and second one for attacked phase. Our main aim is minimizing objective function so, as is obvious in formula, lower value of BER along with higher values of PSNR and NC (Bigger values in denominator makes two fraction smaller) makes it possible for us. Second bracket contains average values of BER and 1/NC for attack phase. Also ω (Here ω = 10) used for balancing effect of PSNR, BER and NC for two phase.

Step 6: Move fireflies as for explained in Section 2.2.

Step 7: Iterate step 2 to 6 until max generation reached.

4 Experimental results and comparison

4.1 Experiment description

Our experiments implemented using Intel(R) Core™2 Due 2.53 GHz, windows7 with 4GB RAM in platform of MATLAB R2014b(8.4.0). The performance of our technique evaluated by utilizing standard grayscale images of “Lena”, “Boat”, “Mandrill”, “Barbara”, “Cameraman”, “Pepper”,

Table 3 Experimental values of NC and BER of proposed method for extracted watermarks from images of Cameraman, Peppers, Pirate

Attack/Images		Cameraman		Peppers		Pirate	
		NC	BER	NC	BER	NC	BER
Attack Free	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Salt & Pepper noise (0.01)	WM1	0.9310	0.0273	0.9510	**0.0195**	0.9427	**0.0225**
	WM2	**0.9423**	**0.0254**	**0.9524**	0.0205	**0.9430**	0.0244
Salt & Pepper noise (0.02)	WM1	0.8663	0.0547	0.8971	**0.0410**	0.8868	**0.0469**
	WM2	**0.8804**	**0.0518**	**0.9008**	0.0430	**0.8875**	0.0498
Salt & Pepper noise (0.03)	WM1	**0.8274**	**0.0713**	**0.8584**	**0.0596**	0.8178	**0.0752**
	WM2	0.8261	0.0762	0.8489	0.0684	**0.8245**	0.0781
Poison noise	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Speckle noise (0.001)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Speckle noise (0.005)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	0.9977	0.0010	0.9977	0.0010	**1**	**0**
Speckle noise (0.009)	WM1	**0.9800**	**0.0078**	**0.9850**	**0.0059**	0.9900	0.0039
	WM2	0.9682	0.0137	0.9750	0.0107	**0.9932**	**0.0029**
Gaussian noise (0.001)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	0.9955	0.0020	**1**	**0**
Gaussian noise (0.005)	WM1	0.8917	0.0439	0.9142	0.0342	0.9274	0.0293
	WM2	**0.9204**	**0.0342**	**0.9262**	**0.0322**	**0.9460**	**0.0234**
Gaussian noise (0.009)	WM1	**0.7904**	**0.0869**	**0.8384**	**0.0664**	0.7651	0.0996
	WM2	0.7556	0.1104	0.8235	0.0781	**0.8602**	**0.0605**
Cropping (Top-Left)	WM1	0.9975	0.0010	0.9525	0.0186	**0.9700**	**0.0117**
	WM2	**0.9977**	**0.0010**	**0.9796**	**0.0088**	0.9615	0.0166
Cropping (Center)	WM1	**0.9825**	**0.0068**	**0.9600**	**0.0156**	**0.9825**	**0.0068**
	WM2	0.9796	0.0088	0.9593	0.0176	0.9773	0.0098
Cropping (Bottom-Right)	WM1	**0.8974**	**0.0400**	0.9575	**0.0166**	**0.9175**	**0.0322**
	WM2	0.8697	0.0566	**0.9593**	0.0176	0.9054	0.0410
Scaling (512-1024-512)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Scaling (512-256-512)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Rotation (90)	WM1	0.0913	0.4844	**0.1263**	0.5078	0.1003	**0.4932**
	WM2	**0.1058**	**0.4775**	0.1056	**0.4932**	**0.1067**	0.5166
Rotation (180)	WM1	**0.1103**	**0.4951**	0.1165	0.5186	0.1229	0.5166
	WM2	0.0982	0.5078	**0.1176**	**0.4805**	**0.1230**	**0.5039**
Gamma correction (0.9)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gamma correction (0.6)	WM1	0.9975	0.0010	**1**	**0**	**1**	**0**
	WM2	**0.9977**	**0.0010**	**1**	**0**	**1**	**0**
JPG compression (Q = 80)	WM1	**1**	**0**	**1**	**0**	**1**	**0**

Table 3 (continued)

Attack/Images		Cameraman		Peppers		Pirate	
		NC	BER	NC	BER	NC	BER
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
JPG compression (Q = 75)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
JPG compression (Q = 50)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gaussian filter (3 × 3)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gaussian filter (5 × 5)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Average filter (3 × 3)	WM1	**0.9386**	**0.0244**	0.9122	0.0361	0.9631	0.0146
	WM2	0.9220	0.0342	**0.9310**	**0.0303**	**0.9753**	**0.0107**
Median filter (3 × 3)	WM1	0.8626	0.0547	0.8234	0.0713	0.8657	0.0547
	WM2	**0.9032**	**0.0420**	**0.8795**	**0.0527**	**0.8855**	**0.0508**
Histogram Equalization	WM1	**1**	**0**	0.9975	0.0010	**1**	**0**
	WM2	0.9977	0.0010	**0.9977**	**0.0010**	**1**	**0**
Sharpening filter	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Wiener filter (3 × 3)	WM1	0.9950	0.0020	0.9975	0.0010	**1**	**0**
	WM2	0.9932	0.0029	**1**	**0**	**1**	**0**

"Pirate", "Jet Plane" and "Lake" with size of 512 × 512 and two binary image of CS as WM1 and test logo as WM2 with size of 32 × 32 which are depicted in Fig. 6a–k.

For optimizing process performed by FA we use 10 fireflies with max generation of 50 and $\gamma = 1.0$, $\beta_0 = 1.0$ and α = 0.1. Convergence of this algorithm with given parameters studied on "Lena" image with watermark WM2 and gained results showed in Fig. 12. On the whole, all images investigated from two aspect of attack and attack free. For the aspect of attack, we applied 8 kind of image processing functions like Center Cropping (size of 100 × 100), JPEG compression (Q = 70), Average filter, Gaussian filter, Sharpening filter, Gaussian noise, Speckle noise and Salt and Pepper noise as attack process. According to above data, for attack free phase acceptable imperceptibility and high robustness achieved. Good value of PSNR shows good quality of watermarked image so that it is hard to recognize base image from watermarked one. For extracted watermark, significant values of *NC* and *BER* indicates accuracy of our technique in extracting watermark. Table 1 shows gained results while Fig. 7 indicates watermarked image and extracted watermark for attack free phase.

For the phase of attack, *NC* and *BER* for each attack extracted watermark were computed and results shown in Tables 2, 3, and 4. The description of each attack used in firefly implementation and evaluating current technique are explained as:

> *Filtering Attack*: Mean filter and Gaussian filter with mask size of 3 × 3 along with Unsharp Masking as sharpening filter applied to watermarked image and *BER* and *NC* of extracted watermark used in objective function of FA.

Table 4 Experimental values of NC and BER of proposed method for extracted watermarks from images of Barbara, Jet Plane, Lake

Attack/Images		Barbara		Jet Plane		Lake	
		NC	BER	NC	BER	NC	BER
Attack Free	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Salt & Pepper noise (0.01)	WM1	**0.9556**	**0.0176**	0.9195	0.0322	**0.9677**	**0.0127**
	WM2	0.9466	0.0234	**0.9395**	**0.0264**	0.9639	0.0156
Salt & Pepper noise (0.02)	WM1	0.8975	**0.0410**	0.8557	0.0605	**0.9179**	**0.0332**
	WM2	**0.9037**	0.0420	**0.8847**	**0.0518**	0.9110	0.0400
Salt & Pepper noise (0.03)	WM1	**0.8820**	**0.0479**	0.8300	0.0723	0.8733	0.0518
	WM2	0.8729	0.0566	**0.8701**	**0.0576**	**0.9106**	**0.0391**
Poison noise	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Speckle noise (0.001)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Speckle noise (0.005)	WM1	**1**	**0**	**0.9850**	**0.0059**	**1**	**0**
	WM2	**1**	**0**	0.9841	0.0068	**1**	**0**
Speckle noise (0.009)	WM1	**0.9975**	**0.0010**	0.9345	0.0254	0.9950	0.0020
	WM2	0.9955	0.0020	**0.9522**	**0.0205**	**1**	**0**
Gaussian noise (0.001)	WM1	**1**	**0**	0.9975	0.0010	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gaussian noise (0.005)	WM1	**0.9629**	**0.0146**	0.9024	0.0391	0.9826	0.0068
	WM2	0.9591	0.0176	**0.9371**	**0.0273**	**0.9841**	**0.0068**
Gaussian noise (0.009)	WM1	**0.8961**	**0.0420**	0.7681	0.0967	**0.9168**	**0.0332**
	WM2	0.8843	0.0508	**0.8260**	**0.0771**	0.9033	0.0430
Cropping (Top-Left)	WM1	0.9700	0.0117	**0.9775**	**0.0088**	0.9500	**0.0195**
	WM2	**0.9796**	**0.0088**	0.9638	0.0156	**0.9503**	0.0215
Cropping (Center)	WM1	0.9975	0.0010	0.9600	**0.0156**	**0.9600**	**0.0156**
	WM2	**0.9977**	**0.0010**	**0.9615**	0.0166	0.9548	0.0195
Cropping (Bottom-Right)	WM1	**0.9750**	**0.0098**	0.9400	0.0234	**1**	**0**
	WM2	0.9615	0.0166	**0.9503**	**0.0215**	0.9932	0.0029
Scaling (512-1024-512)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Scaling (512-256-512)	WM1	0.9900	0.0039	**1**	**0**	**1**	**0**
	WM2	**0.9955**	**0.0020**	**1**	**0**	**1**	**0**
Rotation (90)	WM1	0.0898	**0.4756**	0.1125	**0.5068**	0.1044	0.5029
	WM2	**0.1225**	0.5117	**0.1262**	0.5322	**0.1218**	**0.4736**
Rotation (180)	WM1	0.1073	**0.4883**	**0.1085**	0.5234	0.0928	**0.4473**
	WM2	**0.1116**	0.5078	0.1016	**0.4756**	**0.1128**	0.4883
Gamma correction (0.9)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gamma correction (0.6)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
JPG compression (Q = 80)	WM1	**1**	**0**	**1**	**0**	**1**	**0**

Table 4 (continued)

Attack/Images		Barbara		Jet Plane		Lake	
		NC	BER	NC	BER	NC	BER
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
JPG compression (Q = 75)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
JPG compression (Q = 50)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gaussian filter (3 × 3)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Gaussian filter (5 × 5)	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Average filter (3 × 3)	WM1	**0.9801**	**0.0078**	0.8592	0.0596	0.9676	0.0127
	WM2	0.9752	0.0107	**0.8972**	**0.0459**	**0.9750**	**0.0107**
Median filter (3 × 3)	WM1	**0.8669**	**0.0547**	0.8011	0.0811	0.8669	0.0547
	WM2	0.8648	0.0596	**0.8205**	**0.0791**	**0.8859**	**0.0508**
Histogram Equalization	WM1	0.9975	0.0010	0.9876	0.0049	0.9975	0.0010
	WM2	**1**	**0**	**0.9955**	**0.0020**	**1**	**0**
Sharpening filter	WM1	**1**	**0**	**1**	**0**	**1**	**0**
	WM2	**1**	**0**	**1**	**0**	**1**	**0**
Wiener filter (3 × 3)	WM1	**1**	**0**	0.9850	0.0059	0.9975	0.0010
	WM2	0.9977	0.0010	**0.9909**	**0.0039**	**1**	**0**

Noise Attack: Salt and Pepper noise with density of 0.01, Gaussian noise and Speckle noise with variance of 0.05 added to image. All results with combination of other densities and variances depicted in Tables 2, 3, and 4.

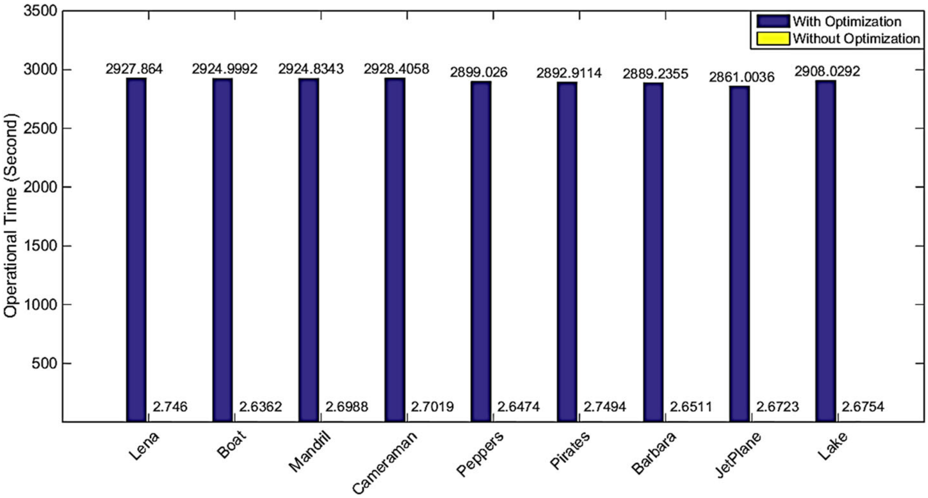

Fig 8 Operational time graph showing two phase: optimization based and without optimization

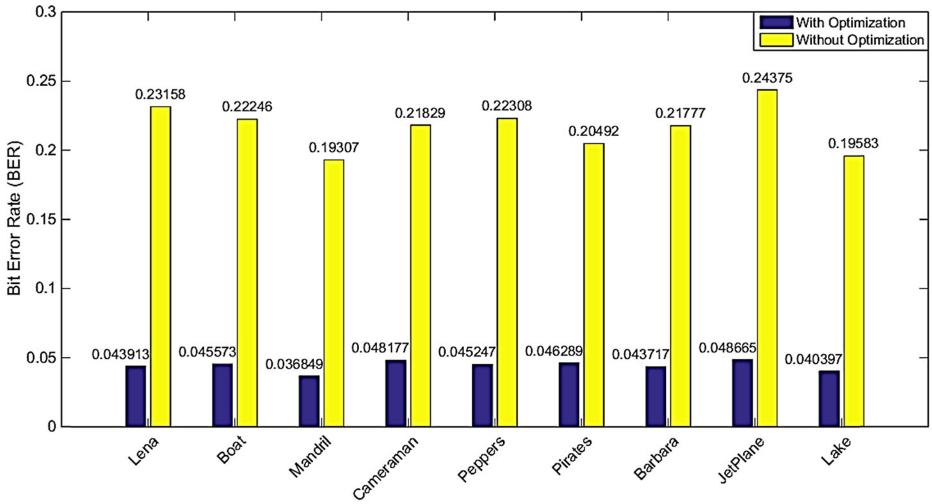

Fig 9 BER graph showing average BER of attacks in two phase: optimization based and without optimization

Cropping Attack: Watermarked image cropped by center like a square with size of 100 × 100 and filling it by zero. Achieved results with various positions shown in Tables 2, 3, and 4.

JPEG Compression: Watermarked image compressed with quality factor of 70 % for FA phase and other QF hired to evaluate proposed method. All outcomes brought in Tables 2, 3, and 4.

There are some other attacks like median filter, scaling, rotation and Gamma correction applied to watermarked image for boosting evaluation and results shown in Tables 2, 3, and 4.

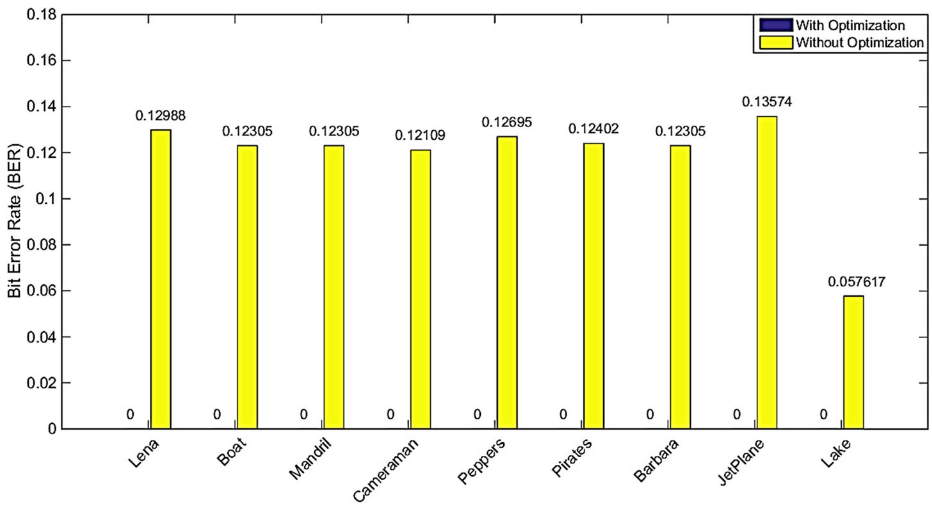

Fig 10 Graph showing BER of attack free phase in two scheme: optimization based and without optimization

Table 5 PSNR of watermarked image via NC and BER values of extracted watermark in attack free phase with average and standard deviation of NC and BER values of extracted watermark in attacked phase on 30 images of Pasadena-Houses-2000 dataset

Image	PSNR(dB)	Attack Free Phase		Attack Phase			
		NC	BER	Average NCs	Standard Deviation of NCs	Average BERs	Standard Deviation of BERs
	38.2561	1	0	0.9071	0.2282	0.0470	0.1233
	39.3486	1	0	0.8922	0.2344	0.0537	0.1265
	38.1799	1	0	0.9010	0.2289	0.0496	0.1223
	37.2022	1	0	0.9085	0.2298	0.0470	0.1261
	33.9351	1	0	0.9230	0.2274	0.0403	0.1244
	35.9447	1	0	0.9163	0.2274	0.0429	0.1227
	37.9812	1	0	0.9053	0.2303	0.0489	0.1281
	37.1024	1	0	0.8917	0.2239	0.0545	0.1225
	34.4320	1	0	0.9095	0.2283	0.0471	0.1266
	36.9445	1	0	0.9136	0.2258	0.0455	0.1272
	32.9763	1	0	0.9239	0.2260	0.0406	0.1262
	33.1625	1	0	0.9246	0.2258	0.0393	0.1223
	33.9325	1	0	0.9182	0.2313	0.0428	0.1271
	37.2329	1	0	0.9117	0.2242	0.0449	0.1213
	37.2124	1	0	0.9112	0.2255	0.0468	0.1280
	35.1527	1	0	0.9203	0.2265	0.0402	0.1188

Table 5 (continued)

	35.6091	1	0	0.9111	0.2284	0.0461	0.1261
	33.8661	1	0	0.9205	0.2293	0.0401	0.1204
	32.8616	1	0	0.9224	0.2311	0.0381	0.1165
	37.4727	1	0	0.9029	0.2289	0.0496	0.1255
	35.6887	1	0	0.9181	0.2301	0.0425	0.1252
	36.5983	1	0	0.9126	0.2256	0.0465	0.1289
	36.6024	1	0	0.9080	0.2243	0.0483	0.1278
	34.0981	1	0	0.9214	0.2324	0.0360	0.1078
	33.2033	1	0	0.9275	0.2289	0.0385	0.1256
	29.3282	1	0	0.9303	0.2318	0.0364	0.1237
	35.7498	1	0	0.9170	0.2268	0.0414	0.1180
	36.5906	1	0	0.9084	0.2279	0.0454	0.1188
	37.2811	1	0	0.9021	0.2311	0.0497	0.1253
	36.6035	1	0	0.9083	0.2313	0.0465	0.1245

4.1.1 Study of operational time

For studying operational status of our scheme, we concentrate on that part which has main share in whole time which is optimization code, so proposed scheme was checked in two phase: First phase include Firefly optimization algorithm to get ignoring factor and second phase checked that without optimization by applying random value as ignoring factor. Comparison of operational time, Average BER of attacks and BER for attack free phase showed in Figs. 8, 9 and 10.

4.1.2 Performance evaluation of proposed technique on a dataset

As an extra study, behavior of our technique investigated on 30 images of Pasadena-Houses-2000 dataset [15]. To have fair comparison, all images resized to 512 × 512 and grayscale mode. Experimental results gathered in Table 5 and average values of BER for each attack on all sample images depicted in Figs. 11 and 12.

4.2 Comparison and discussion

In case of investigating operational time, according to our experiment between two schemes of using Firefly optimization algorithm and using random value instead of that, FF shows long runtime (More than 2500 s) for achieving ignoring factor rather than using random number but in case of investigating extracted watermark, significant values of BER and NC indicates undeniable robustness of first scheme. As shown in Fig. 9, upper bound of average BER in optimized phase is less than 0.05 that shows good robustness rather than second scheme with lower bound of 0.19. Interesting point is zero values of BER for optimized scheme in attack free phase against unacceptable values (More than 0.12 except for Lena which is 0.06) of that for second scheme which shows efficiency of firefly algorithm in proposed scheme. So it seems to be worthwhile to spend that much time to achieve this amount of robustness.

Due to main results, our technique shows good robustness about different kinds of additive noise like Salt & Pepper, Poison, Speckle, Gaussian noise with different variances. Low value of BER and on the other side high value of NC shows superiority of our technique. In case of geometric attacks like scaling and rotation inconsistent results achieved. For scaling in both side, down scaling and upper scaling, significant values for BER and NC achieved but rotation demonstrate as a weak point. Main reason for that clearly goes back to block indexes passed from embedding phase. After rotation, position of each block will be changed. So retrieved blocks differs from those of embedding phase. For filtering attacks like Gaussian filter, Average filter, Median filter, Sharpening and wiener filter considerable values obtained. Our scheme was checked with some other kind of attacks like JPEG compression, Gama

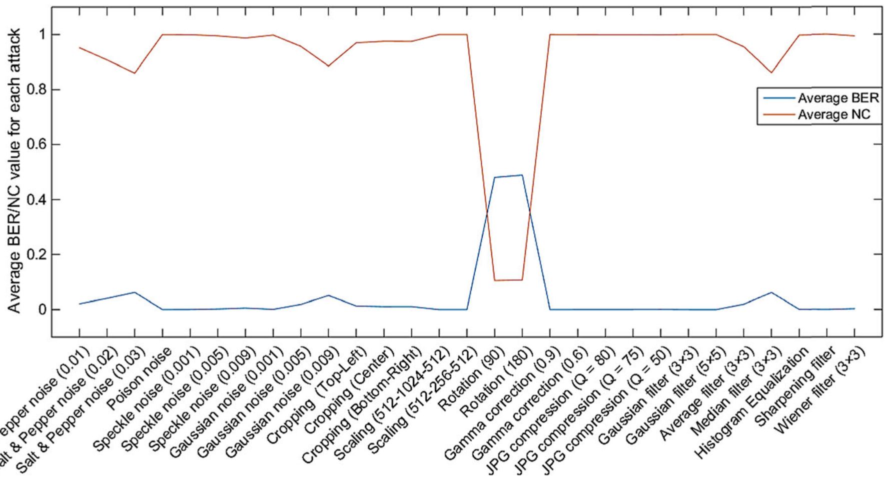

Fig 11 Graph showing average BER/NC of extracted watermarks on each attack

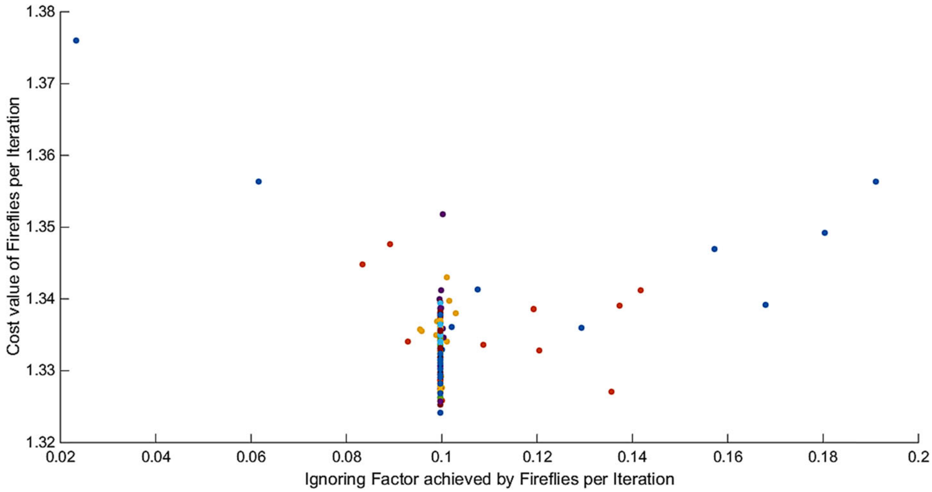

Fig 12 Convergence behavior graph of fireflies in 50 generation

correction, histogram equalization that led to similar results. On evaluating imperceptibility values bounded between 35 and 39 dB which shows slight distortion in watermarked image but high values of VIF range (more than 0.97) represents good visibility of watermarked image. Visibility of proposed scheme checked on both primary and dataset images with WM2 watermark shown in Fig. 13. In following, an extra study on 30 images done that shows similar results to primary ones, Achieved PSNR of watermarked image and BER, NC values of extracted watermark in attack free phase are in same range of primary results which shows superiority of our scheme. According to Table 5 and Fig. 11, average value of NC and BER of each sample image shows robustness of technique against various kinds of attacks.

In following, our method compared with methods based on discrete wavelet transform and singular value decomposition [20], wavelet coefficient quantization based watermarking [22],

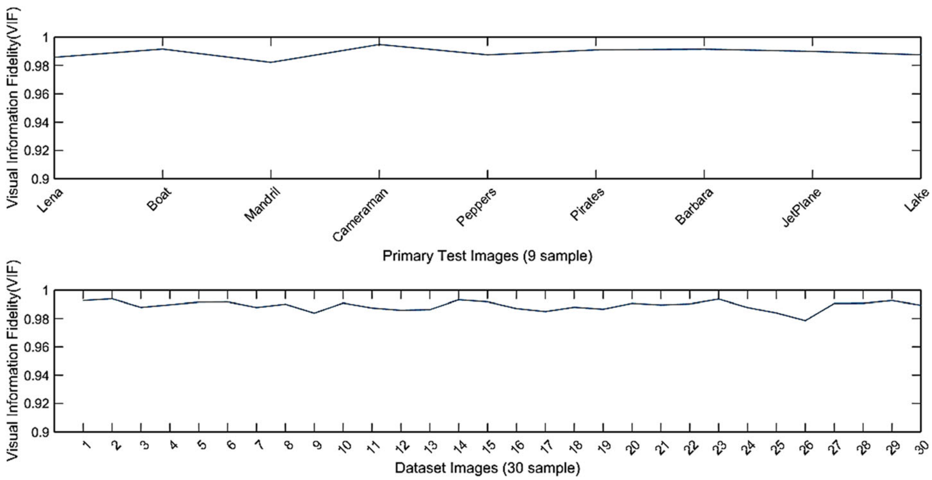

Fig 13 Visibility status of primary images and dataset images based on VIF

Table 6 Comparison of BER values for extracted watermark with method [14, 20, 22, 26, 31] against various attacks on sample image of "Lena"

Attack/Methods	Proposed Method	Lin et al. [22]	Lai et al. [20]	Miller et al. [26]	Soniwal et al. [31]	Hajiramezan et al. [14]
Attack Free	**0**	0	0	0	0.0060	0
Salt & Pepper noise (0.01)	0.0195	0.1700	0.0378	**0**	0.1282	0.0039
Salt & Pepper noise (0.02)	0.0420	0.3200	0.0425	**0.0300**	0.1210	0.0068
Salt & Pepper noise (0.03)	0.0664	0.3100	**0.0473**	0.0800	0.1486	0.0127
Poison noise	**0**	0	0	0	0.0060	0
Speckle noise (0.001)	**0**	0	0.0134	0	0.0526	0
Speckle noise (0.005)	0.0010	0.0500	0.0380	**0**	0.0833	0.0020
Speckle noise (0.009)	0.0049	0.0600	0.0383	**0**	0.1112	0.0225
Gaussian noise (0.001)	**0**	0.0100	0.0268	0	0.0628	0.0010
Gaussian noise (0.005)	0.0283	0.2400	0.0469	**0**	0.1018	0.0840
Gaussian noise (0.009)	0.0654	0.3700	0.0489	**0.0200**	0.1373	0.1523
Cropping (Top-Left)	0.0098	**0**	0.0029	0.0200	0.2451	0.0078
Cropping (Center)	**0.0117**	0.0300	0.0570	0.1200	0.2298	0.0098
Cropping (Bottom-Right)	**0.0127**	0.0300	0.0602	0.0500	0.1614	0.0146
Scaling (512-1024-512)	**0**	0	0	0	0.0060	0
Scaling (512-256-512)	**0**	0	0.5854	0	0.1044	0.0713
Rotation (90)	0.5049	0.4200	0.0568	0.4900	**0.0060**	0.5039
Rotation (180)	0.4873	0.5400	**0**	0.5100	0.0060	0.4912
Gamma correction (0.9)	**0**	0	0.0189	0	0.6205	0
Gamma correction (0.6)	**0**	0	0.1295	0	0.4325	0.6699
JPG compression (Q = 80)	**0**	0	0.0205	0	0.0285	0
JPG compression (Q = 75)	**0**	0	0.0191	0	0.0189	0
JPG compression (Q = 50)	**0**	0	0.0285	0	0.0208	0.0430
Gaussian filter (3 × 3)	**0**	0	0.0088	0	0.0108	0
Gaussian filter (5 × 5)	**0**	0	0.0088	0	0.0108	0
Average filter (3 × 3)	**0.0098**	0.0700	0.0733	0.0200	0.1525	0.1758
Median filter (3 × 3)	0.0537	**0**	0.1303	0	0.1742	0.3711

Table 6 (continued)

Attack/Methods	Proposed Method	Lin et al. [22]	Lai et al. [20]	Miller et al. [26]	Soniwal et al. [31]	Hajiramezan et al. [14]
Histogram Equalization	**0**	0.0600	0.0250	0	0.2205	0.1572
Sharpening filter	**0**	0	0.0223	0	0.0547	0
Wiener filter (3 × 3)	**0**	0	0.0643	0	0.1608	0.0684

dirty paper trellis codes [26], SWT-SVD based watermarking scheme [31] and non-blind watermarking method based on Integer Wavelet Transform [14]. Considering to provide comprehensive comparison to methods, compared papers implemented with grayscale sample image of Lena (Size is [512,512]) on the same platform as ours. As a point, intensity of sample image normalized before operation. Lai's method [20], concentrate on combination DWT and SVD. We used strength parameter value of 0.01 as scaling factor in implementation. On the other side, Lin's method [22], focus on difference of wavelet coefficient quantization. We used blocks with size of 7 as mentioned in paper with thresholds of $T = 0.04$ and $y = 0.02$. For implementing Miller's method [26], which is based on dirty paper trellis codes, we used trellis with 64 state and 64 arc per state as mentioned in paper. In other work, Soniwal's method [31] concentrates on combination of Stationary Wavelet Transform (SWT) with Singular Value Decomposition (SVD). We used $k = 0.03$ as scaling factor along with mentioned parameters in paper. The last paper, we compare with is Hajiramezan's non-blind watermarking scheme [14] which focused on Integer Wavelet Transform (IWT) and we implemented that by applying $M = 0.1$ as strength factor. Details of comparison gathered in Table 6. As stated our method shows better robustness against various attacks rather than other methods except in rotation which discussed before. In following, quality of watermarked image and payload of each technique gathered in Table 7. Situation of these two items shows acceptable quality of proposed scheme. Notice that bold values shows progress in robustness of our technique.

5 Conclusion

In this paper, a novel digital image watermarking technique using combination of Lifting Wavelet Transform and Firefly Algorithm presented. According to standard deviation, non-overlapping blocks sorted and optimization algorithm used for specifying best block for starting embedding process. Our experimental results compared with known other schemes

Table 7 Comparison of PSNR and Payload of proposed method with [14, 20, 22, 26, 31] on sample image of "Lena"

	Proposed Method	Lin et al. [22]	Lai et al. [20]	Miller et al. [26]	Soniwal et al. [31]	Hajiramezan et al. [14]
PSNR (dB)	37.9815	51.1431	**64.6413**	49.7941	36.1175	42.9665
Payload	7281	585	131072	384	**262144**	65536

for various image processing attacks like adding noise, smooth filtering, sharpening, gamma correction, cropping and scaling depicted high robustness and acceptable imperceptibility.

Our future work will concentrate on using other kind of optimization algorithm and transform domain. Also working on such a features that is more resistance of geometrical transform to use in embedding process.

References

1. Agoyi M, Çelebi E, Anbarjafari G (2015) A watermarking algorithm based on chirp z-transform, discrete wavelet transform, and singular value decomposition. SIViP 9:735–745. doi:10.1007/s11760-014-0624-9
2. Ali ES (2015) Speed control of DC series motor supplied by photovoltaic system via firefly algorithm. Neural Comput & Applic 26:1321–1322. doi:10.1007/s00521-014-1796-5
3. Amiri T, Moghadam ME (2015) A new visual cryptography based watermarking scheme using DWT and SIFT for multiple cover images. Multimedia Tools Appl. doi:10.1007/s11042-015-2770-7
4. An L, Gao X, Li X, Tao D, Deng C, Li J (2012) Robust reversible watermarking via clustering and enhanced pixel-wise masking. IEEE Trans Image Process. doi:10.1109/TIP.2012.2191564
5. An L, Gao X, Yuan Y, Tao D (2012) Robust lossless data hiding using clustering and statistical quantity histogram. Neurocomputing 77(1):1–11. doi:10.1016/j.neucom.2011.06.012
6. Ana L, Gao X, Yuanc Y, Taod D, Denga C, Jia F (2012) Content-adaptive reliable robust lossless data embedding. Neurocomputing 79:1–11. doi:10.1016/j.neucom.2011.08.019
7. Arora S, Singh S (2013) The firefly optimization algorithm: convergence analysis and parameter selection. Int J Comput Appl. doi:10.5120/11826-7528
8. Chen YH, Huang HC (2015) Coevolutionary genetic watermarking for owner identification. Neural Comput & Applic 26:291–298. doi:10.1007/s00521-014-1615-z
9. Daubeches I, Sweldens W (1998) Factoring wavelet transform into lifting steps. J Fourier Anal Appl 4(3): 247–269. doi:10.1007/BF02476026
10. Elshazly EH, Faragallah OS, Abbas AM, Ashour MA, El-Rabaie EM, Kazemian H et al (2014) Robust and secure fractional wavelet image watermarking. SIViP. doi:10.1007/s11760-014-0684-x
11. Fan W, Chen J, Zhen J (2005) SPIHT algorithm based on fast lifting wavelet transform in image compression. In: Hao Y et al. (eds) CIS 2005, Part II, LNAI 3802, pp 838–844. doi:10.1007/11596981_122.
12. Gao X, An L, Li X, Tao D (2009) Reversibility improved lossless data hiding. Signal Process 89(10):2053–2065. doi:10.1016/j.sigpro.2009.04.015
13. Gao X, An L, Yuan Y, Tao D, Li X (2011) Lossless data embedding using generalized statistical quantity histogram. IEEE Trans Circuits Syst Video Technol 21(8):1061–1070. doi:10.1109/TCSVT.2011.2130410
14. Hajiramezan S, Hasanzadeh M, Mirghadri A (2015) A new robust and non-blind digital image watermarking scheme based on integer wavelet transform. Knowledge-Based Engineering and Innovation (KBEI), 2015 2nd International Conference on. doi:10.1109/KBEI.2015.7436044
15. Helle C, Perona P Pasadena Houses 2000. © California Institute of Technology. Available: http://vision.caltech.edu/archive.html.
16. Hu HT, Hsu LY (2016) Collective blind image watermarking in DWT-DCT domain with adaptive embedding strength governed by quality metrics. Multimedia Tools Appl. doi:10.1007/s11042-016-3332-3
17. Huang HN, Chen ST, Lin MS, Kung WM, Hsu CY (2015) Optimization-based embedding for wavelet-domain audio watermarking. J Signal Proc Syst 80:197–208. doi:10.1007/s11265-013-0863-y
18. Kaur R, Rattan M (2015) Optimization of the return loss of differentially fed microstrip patch antenna using ANN and firefly algorithm. Wirel Pers Commun 80:1547–1556. doi:10.1007/s11277-014-2099-y
19. Khan M, Shah T (2015) A copyright protection using watermarking scheme based on nonlinear permutation and its quality metrics. Neural Comput & Applic 26:845–855. doi:10.1007/s00521-014-1747-1
20. Lai CC, Tsai CC (2010) Digital image watermarking using discrete wavelet transform and singular value decomposition. IEEE Trans Instrum Meas. doi:10.1109/TIM.2010.2066770
21. Lei B, Soon IY, Zhou F, Li Z, Lei H (2012) A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition. Signal Proc 92(9):1985–2001. doi:10.1016/j.sigpro.2011.12.021
22. Lin WH, Horng SJ, Kao TW, Fan P, Lee CL, Pan Y (2008) An efficient watermarking method based on significant difference of wavelet coefficient quantization. IEEE Trans Multimedia 10(5):746–757. doi:10.1109/TMM.2008.922795

23. Mehta R, Rajpal N, Vishwakarma VP (2015) A robust and efficient image watermarking scheme based on Lagrangian SVR and lifting wavelet transform. Int J Mach Learn Cybern. doi:10.1007/s13042-015-0331-z
24. Mehta R, Rajpal N, Vishwakarma VP (2015) Robust image watermarking scheme in lifting wavelet domain using GA-LSVR hybridization. Int J Mach Learn Cybern. doi:10.1007/s13042-015-0329-6
25. Mehta R, Rajpal N, Vishwakarma VP (2015) LWT- QR decomposition based robust and efficient image watermarking scheme using Lagrangian SVR. Multimedia Tools Appl. doi:10.1007/s11042-015-3084-5
26. Miller ML, Doerr GJ, Cox IJ (2002) Dirty-paper trellis codes for watermarking. Image Processing. Proceedings. 2002 International Conference on. doi:10.1109/ICIP.2002.1039904
27. Rahebi J, Hardalaç F (2015) A new approach to optic disc detection in human retinal images using the firefly algorithm. Med Biol Eng Comput. doi:10.1007/s11517-015-1330-7
28. Raja NSM, Manic KS, Rajinikanth V (2013) Firefly algorithm with various randomization parameters: an analysis (Eds): SEMCCO 2013, Part I, LNCS 8297, pp. 110–121. doi:10.1007/978-3-319-03753-0_11
29. Sheikh HR, Bovik AC (2006) Image information and visual quality. IEEE Trans Image Process. doi:10.1109/TIP.2005.859378
30. Soliman MM, Hassanien AE, Onsi HM (2015) An adaptive watermarking approach based on weighted quantum particle swarm optimization. Neural Comput & Applic. doi:10.1007/s00521-015-1868-1
31. Soniwal K, Taparia R, Parmar G (2016) Imperceptible hybrid image watermarking using SWT-SVD and their performance analysis in compressed domain. Int J Electron Electr Comput Syst 5
32. Su Q, Wang G, Jia S, Zhang X, Liu Q, Liu X (2015) Embedding color image watermark in color image based on two-level DCT. SIViP 9:991–1007. doi:10.1007/s11760-013-0534-2
33. Thafasal Ijyas VP, Sameer SM (2014) Firefly algorithm for joint estimation of frequency offsets and channel in OFDMA uplink. Wirel Pers Commun 79:565–580. doi:10.1007/s11277-014-1873-1
34. Walia E, Suneja A (2014) A robust watermark authentication technique based on Weber's descriptor. SIViP 8:859–872. doi:10.1007/s11760-012-0312-6
35. Wang J, Lian S, Wang J (2015) Hybrid additive multi-watermarking and decoding. Multimedia Systems. doi:10.1007/s00530-013-0338-9
36. Yang XS (2008) Nature-inspired metaheuristic algorithms. Luniver Press, Frome, p 808, ISBN: 1-905986-10-6
37. Yang XS (2009) Firefly algorithms for multimodal optimization. In: Stochastic algorithms: Foundations and applications, SAGA, Lecture notes in computer sciences, 5792:169–178. doi:10.1007/978-3-642-04944-6_14
38. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimization. Int J Bio-inspired Comput 2(2):78–84. doi:10.1504/IJBIC.2010.032124
39. Zhang SQ, He N, Zang HY (2006) Research of the lifting wavelet arithmetic and applied in rotary mechanic fault diagnosis. J Phys Conf Ser 48:696–700. doi:10.1088/1742-6596/48/1/131

Mr. Behnam Kazemivash is MSc. Student of computer software engineering in Islamic Azad University, Damavand branch. His main research area include: Image Processing and Pattern Recognition, Computer Vision. He works as Research Assistant in image processing lab of Shahid Beheshti University.

Dr. Mohsen Ebrahimi Moghaddam is an associate professor in electrical and computer engineering department, Shahhid Beheshti university of Iran since 2006. He has got his Phd, Msc, and Bsc from Sharif University in Iran. His research interests are image processing and pattern recognition specially using artificial intelligence techniques such as image security, watermarking, blurring. He is the image processing lab chief in his department and one of major works in this lab is in watermarking. There are several publication in this scope by him and his students.