ORIGINAL ARTICLE

A novel optimized block-based approach for image watermarking

Harun Akbulut · Muhammet Emin Sahin · Oğuzhan Akay

Received: 25 April 2025 / Accepted: 10 April 2026
© The Author(s) 2026

Abstract
In this study, a novel block-based image watermarking method is proposed to enhance the quality and robustness of watermarked images. Traditional image watermarking techniques often treat the host image as a whole, leading to information loss due to the homogeneous fusion ratio applied across the entire image. To address this issue, the proposed method divides the host and watermark images into non-overlapping blocks of fixed sizes. Each block is then processed individually using artificial intelligence optimization algorithms, including Genetic Algorithm (GA), Differential Evolution Algorithm (DE), Artificial Bee Colony (ABC), and Sand Cat Swarm Optimization Algorithm (SCSO), to determine the optimal fusion coefficients. The block-based approach minimizes the contrast loss and enhances the watermark's visibility and robustness. The application of a smoothing operator further mitigates the saw-tooth effect, commonly observed in block-based methods, leading to improved visual quality. Furthermore, the proposed method's performance is compared against traditional watermarking techniques across various quality metrics. The findings unequivocally indicate that the proposed approach demonstrates a superior performance relative to conventional methodologies. In the comparative analysis of optimization algorithms, the SCSO exhibited a markedly enhanced performance over its counterparts, while DE algorithm also outperformed ABC and GA. These findings suggest that the proposed method provides a significant advancement over traditional techniques in the domain of image watermarking.

Keywords Image watermarking · Genetic algorithm · Differential evolution algorithm · Artificial bee colony · Block-based method · Quality metrics

1 Introduction

Nowadays, many images found on the internet are not watermarked, making it very easy for anyone to download and modify these images, which could result in the original creator losing their rights to their image. In particular, the expansion of high-speed internet and computer networks has enabled the discovery of new avenues in scientific research and development, business and entertainment. The cost-effectiveness of commercial software has significantly increased the trade of quality works of art in the form of digital images and video series streamed over the Internet. Sending hard copies through the mail may soon be a thing of the past. The growth is also due to concern about the use of digital format data. With the spread and increasing popularity of the Internet, physical

Published online: 09 July 2026

watermarking concepts have been extended to the protection and verification of digital data. Previously, data manipulation in analog environments was quite laborious, but with digital tools such changes and edits have become much easier. Therefore, stronger measures are needed to protect digital resources. This is a process done through special information or marks hidden within digital data. These embedded messages can be used for a variety of purposes, such as verifying the authenticity of digital content, identifying copyright ownership, or enabling tracking of content. This technique helps prevent unauthorized use and modification of digital content, while also providing protection to copyright owners. Digital watermarking uses the concept of embedded messages for copyright or ownership protection purposes [1, 2].

To safeguard copyright, digital image watermarking, which is widely employed, can be broadly categorized into two types: visible and invisible watermarking. In invisible image watermarking, copyright-related information is imperceptibly embedded within the image, eluding human detection. Conversely, in visible image watermarking, such information is embedded in a manner that is discernible to human perception. This study introduces a block-based visible image watermarking method developed on the foundational principle of image fusion. Image fusion is defined as the process of synthesizing a single, enhanced image that encapsulates a greater degree of information from two or more source images. Numerous image fusion techniques have emerged in recent years, broadly categorized into spatial domain-based and transform domain-based methods [3, 4]. Spatial domain-based approaches involve applying information measures directly on images to assess the information level of pixels or regions. Subsequently, pixels or regions with the highest information level at each position are chosen to construct a fully improved image. Pixel-based methods operate directly on pixel values, earning them the designation of single-scale fusion [5]. These methods analyse pixels within an image to identify regions or segments that contribute to the final fused image. Pixel processing is a common feature across various image fusion algorithms, regardless of the fusion domain. Some examples of such algorithms are cited here [6, 7]. Pixel-level or low-level fusion methods produce fused images with diminished contrast due to the equal treatment of all pixels [8]. To address this issue, block-based or region-based image fusion algorithms have been suggested in [8–10]. Spatial domain techniques identify containing a greater amount of information regions within the input image through measures of quality metrics. These regions can be translated into a transform domain using various transforms such as Fourier, wavelet, or DCT. Subsequently, pixel-based methods are employed to fuse images within the transform domain. Consequently, methods relying on transforms are termed multi-scale fusion approaches. These techniques utilize transforms like pyramid transform [11], wavelet transform [12], discrete cosine transform [13], contourlet transform [14], curvelet transform [15] [18], and shearlet transform [16]. Incorporating an aspect of deep learning into the fusion process leads to the emergence of learning-based fusion methods [17]. These approaches utilize deep neural networks such as Deep Belief Network (DBN), Convolutional Neural Network (CNN) and etc. [18–21]. In addition to neural networks, Genetic Algorithm (GA) is employed in learning-based image fusion methods. As a bio-inspired algorithm, GA iteratively mutates and recombines the most suitable individuals within the current population to update it at each iteration. When addressing image fusion, methods based on region selection process input image pairs by dividing them into patches or blocks. Kong et al. [22] introduced a genetic algorithm for multi-focus image fusion. Determining the block size is crucial to prevent artifacts in the fused image. GA can be utilized to obtain the optimal block size. In the literature, GA is utilized to optimize the fusion rule, which determines the contribution of each source image to the final fused image. Additionally, the combination of GA and cellular neural networks is applied in the image fusion process [23, 24]. Aslantas and Kurban proposed a multi-focus algorithm that utilizes the differential evolution algorithm to optimize the block size [25]. Banharnsakun et al. introduced a multi-focus image fusion method employing the Artificial Bee Colony (ABC) algorithm [26]. Siddiqui et al. employed Particle Swarm Optimization (PSO) for block-based pixel-level multi-focus image fusion [27]. Akbulut investigated the optimization of block size in multi-focus image fusion using 16 meta-heuristic algorithms [28]. Experiments on Lytro dataset image pairs demonstrate that the Artificial Bee Colony (ABC) algorithm achieves the best performance based on both objective quality metrics and subjective evaluations.

Advancements in computer and internet technologies have made accessing information incredibly easy, consequently exacerbating the issue of copyright infringement. Additionally, artificial intelligence optimization algorithms have been increasingly utilized for solving multidimensional, nonlinear problems, yielding superior results compared to traditional methods. In this study, we propose a block-based visible image watermarking method using artificial intelligence optimization algorithms such as GA, DE and ABC which outperforms existing watermarking techniques in literature for copyright protection. In existing visible image watermarking approaches, the host image is generally treated as a whole, and a single global fusion coefficient is employed for watermark embedding. This approach can lead to problems such as low watermark visibility in some regions and visual quality degradation in others. In this study, we propose a novel strategy in which images are divided into blocks, and the optimal fusion coefficient for each block is individually determined using metaheuristic optimization algorithms. This strategy allows the method to adapt to the heterogeneous structure of the image, aiming to simultaneously optimize both visual quality and watermark visibility. The aim is to enhance the quality of the resulting watermarked image, making it more useful for both human perception and computer vision applications. This study addresses this goal by offering a solution to this problem of significant importance, such as copyright infringement. Block diagram of study is given in Fig. 1. The contributions of our study to the literature can be summarized as follows:

- A novel block-based approach for visible digital image watermarking has been introduced for the first time in the literature. Unlike traditional methods that utilize a single fusing coefficient for embedding both the host and watermark images, our approach divides the host and watermark images into equal blocks. The optimal fusing coefficient is then computed for each block individually, resulting in more effective watermarking.

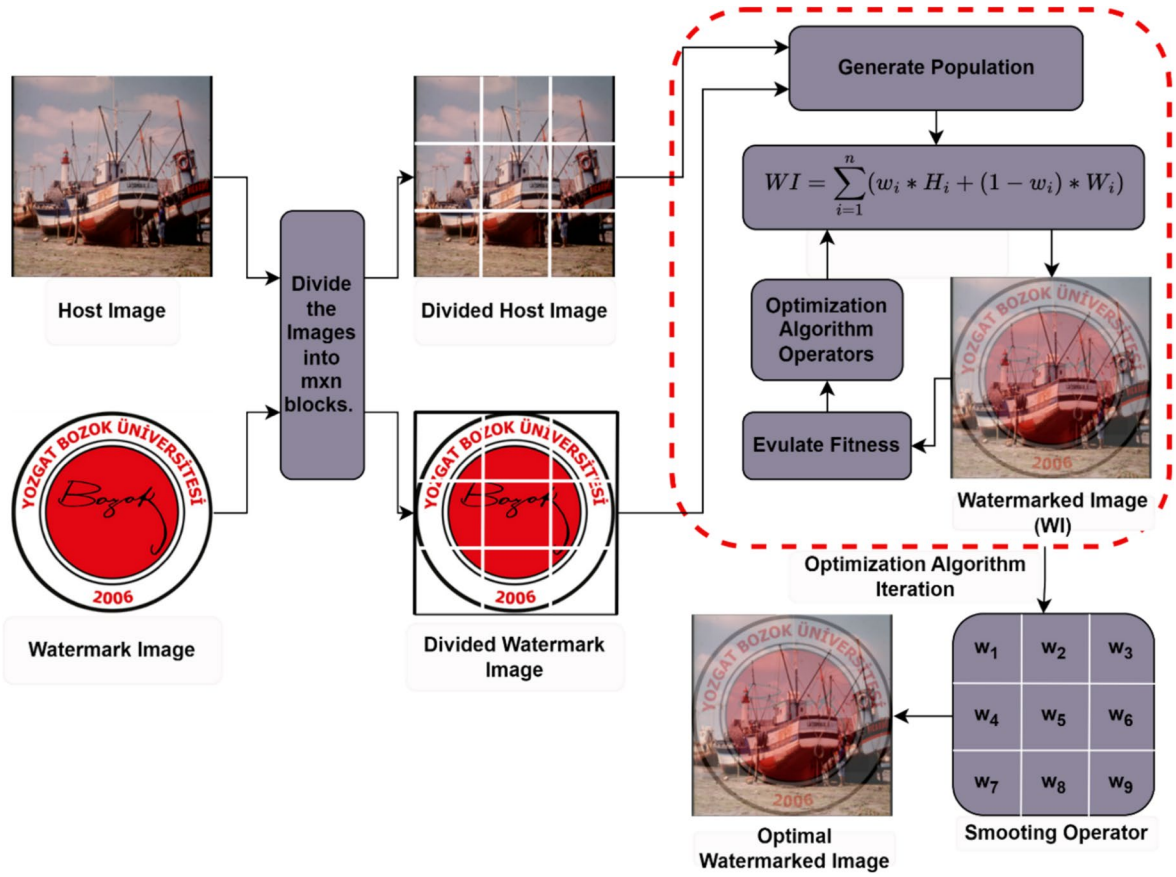

Fig. 1 Block diagram of study

- The optimal fusing for each block has been calculated using optimization algorithms GA, DE and ABC, a widely utilized artificial intelligence optimization algorithms in the literature.
- To evaluate the performance of the proposed method, experiments have been conducted on commonly used images in the field of image processing, ensuring the assessment of the method's performance across different images.
- The quality of the obtained watermarked images has been assessed using commonly employed quality metrics in the literature, including entropy (QM_E), edge-based fused quality index (QM_Q), sum modified laplacian (QM_S), variance (QM_{Var}), and Chen-Blum metric (QM_{CB}). Detailed numerical results have been provided.
- The proposed method has been evaluated against the traditional method using both objective and subjective evaluation techniques, demonstrating that the proposed method consistently outperforms the traditional approach.
- By introducing a novel block-based visible image watermarking method and conducting comprehensive evaluations, study contributes significantly to the existing literature in the field of digital image watermarking.

The paper is organized to provide an introduction to digital image watermarking, emphasizing its importance in copyright protection. Section 2 focuses on providing image spatial-domain algorithms, artificial intelligence optimization algorithms, and the application of them in watermarking. The section also outlines quality metrics for evaluating watermarking techniques are discussed in detail. Section 3 presents experimental results, highlighting comparisons between traditional and proposed methods, along with objective and subjective evaluations. In Sect. 4, discussion section underscores the significance of optimization methods, compares the proposed method with existing approaches, and suggests potential future enhancements. Finally, the conclusion summarizes the study's objectives, emphasizes the importance of digital image watermarking, and suggests avenues for future research and improvements.

2 Material and methods

2.1 Image watermarking

Image spatial-domain algorithms deal directly with the pixels of source images and combine relevant and important information from the registered source images using spatial domain fusion techniques to achieve the desired result. The simplest technique is the weighted average of the registered source images. One disadvantage of this approach is that spatial distortion can occur in the fused image due to the use of averaging or weighted averaging rules. This problem can be addressed by frequency domain approaches [29]. Some fusion methods using this approach include maximum selection, minimum selection, average, high-pass filtering, Principal Component Analysis (PCA), and Intensity-Hue-Saturation (IHS) colour space [30].

2.2 Optimization algorithms

Artificial intelligent has emerged as a prominent research area in various fields, garnering significant attention and achieving remarkable successes in recent years [31–33]. These algorithms aim to imbue computer systems with human-like intelligence and learning capabilities, boasting a wide range of applications [34–36]. The use of artificial intelligence algorithms in areas such as copyright protection is steadily increasing. This paper highlights the significance and potential of using optimization algorithms for copyright protection. It discusses the tendency of optimization algorithms to yield superior results compared to traditional methods and emphasizes their role in the development of watermarking techniques. Specifically, it explores how optimization algorithms like differential evolution, genetic algorithms, and artificial bee colony can be employed to enhance visible image watermarking methods. The primary rationale for selecting the GA, DE, and ABC algorithms in this study is to

provide a comprehensive evaluation representing the main classes of well-established metaheuristic approaches in the literature: evolutionary, deterministic, and swarm intelligence-based methods, respectively. GA was chosen due to its widespread acceptance among evolutionary algorithms, strong theoretical foundation, and its effective balance between exploration and exploitation through genetic operators. DE was selected for its deterministic mutation strategy, which requires minimal parameter tuning, offers stable and reproducible search behaviour, and demonstrates strong performance in avoiding local minima. ABC, on the other hand, stands out among swarm intelligence algorithms by effectively reflecting cooperative search dynamics. The combined use of these three fundamental algorithms enables a comprehensive assessment of the proposed block-based watermarking method's algorithm-independent validity and generalizability.

2.2.1 Sand Cat swarm optimization

The Sand Cat Swarm Optimization (SCSO) algorithm is a population-based metaheuristic inspired by the hunting and navigation behaviors of sand cats [34]. The algorithm models the sand cat's advanced auditory perception, particularly under low-visibility conditions, to effectively guide the search process. SCSO incorporates an adaptive mechanism that balances exploration and exploitation during the optimization procedure. The gradual reduction of the control parameter enables a controlled transition from global exploration to local exploitation. The position update mechanism integrates leader-based guidance with stochastic components to reduce the risk of premature convergence. Experimental evaluations demonstrate that SCSO achieves competitive, stable, and effective performance on benchmark functions as well as engineering optimization problems. In recent years, SCSO has attracted considerable attention in the literature, with several studies proposing comprehensive analyses, chaotic variants, and multi-strategy enhanced versions to improve its optimization performance. These developments indicate that SCSO provides a flexible and extensible framework for further advancements in metaheuristic optimization [34–36].

2.2.2 Genetic algorithm (GA)

Charles Darwin formulated the principle of natural selection as the fundamental mechanism of evolution. In 1970, James Holland and his team introduced this methodology in a more formal and actionable manner [37]. Holland demonstrated that genetic algorithms have solid theoretical foundations using schema theory and showed that they can effectively solve a wide range of optimization problems [38]. Genetic algorithms are search algorithms based on natural selection and natural genetic mechanics. This algorithm employ an iterative process to find the best solutions within a population and simulate the selection of the most suitable solutions under environmental pressures. This method has a wide range of applications in computer science and engineering, particularly in solving complex optimization problems.

2.2.3 Differential evolution (DE) algorithm

Differential Evolution (DE) is considered a simple yet competitive stochastic real-parameter evolutionary algorithm (EA). Proposed by Storn and Price, DE is a population-based global optimization algorithm that iteratively applies variation operators (mutation and crossover) and a selection operator similar to typical EAs [39]. DE algorithms are population-based optimization algorithms. A population contains multiple individuals, each represented by a vector. The DE continuously improves individuals in an iterative process, with each individual continuously evolving until ultimately reaching the best solution [40]. DE generally involves three operations: mutation, crossover, and selection. The mutation operation selects several vectors from the population and generates a mutation vector through the mutation operator. The crossover operation applies the crossover operator to the mutation vector and the parent vector to create a trial vector. The selection operation chooses one of the vectors from the trial and parent vectors to survive into the next generation.

2.2.4 Artificial Bee Colony (ABC) algorithm

The Artificial Bee Colony (ABC) algorithm, proposed by Karaboga (2005), simulates the foraging behavior of a bee colony and has been successfully applied to optimize numerical problems [41–43]. A bee colony seeks to maximize the amount of nectar brought into the hive by finding rich food sources. Although there are several task classifications within a real hive, the ABC uses a minimal model that mimics the foraging behavior involving worker bees, onlooker bees, and scout bees. A worker bee finds a food source position by altering the position in its memory, evaluates the nectar amount of each new source, and remembers the better one (greedy selection). Worker bees share information about the quality of the food source they are exploiting in the dance area. Onlooker bees find food sources based on information from worker bees. More profitable sources are more likely to be selected by onlookers. An onlooker bee selects a food source based on this information (probabilistic selection) and makes a modification to the source. A greedy selection is performed to identify a better source. In the natural behavior of real bees, when a food source is no longer worth exploiting, it is abandoned by the bees, and the worker bee of that source becomes a scout, searching the environment randomly or with intrinsic motivation.

2.3 Quality metrics

Evaluating the quality of a fused image generated using an image fusion method is a challenging task. Therefore, various metrics are used to assess the accuracy of the information transferred from the source images without distortion. The metrics employed in this study are listed below.

2.3.1 Variance (QVAR)

VAR is a measure of how much the elements of a set are spread out from the mean of that set. As an image becomes blurred, pixel values tend to approach the average pixel value. Therefore, the variance value of a blurred image is lower than the variance value of the corresponding sharp image. Thus, VAR can be utilized as a quality metric in image fusion applications [44]. For an M × N-sized image block, VAR is calculated as given in Eq. 1:

$$VAR = \frac{1}{MxN} \sum_{i=1}^{M} \sum_{j=1}^{N} (p_{i,j} - \overline{p})^2 \tag{1}$$

Here, $p_{i,j}$ represents the pixel value at the i—th row and j—th column of the block, while the mean pixel value of the block is denoted by $\overline{p}$.

2.3.2 Sum Modified Laplacian (QMS)

As the contrast value of an image increases, so does its frequency value. In such cases, a high-pass filter like the Laplacian operator can be used to detect high frequencies. However, considering that the signs of second derivatives may differ in horizontal and vertical directions, it is possible for one sign to nullify the effect of the other. The Modified Sum of Modified Laplacian (QM_S) calculates the sum of absolute values of partial second derivatives to mitigate this effect. This feature distinguishes it from the normal Laplacian operator. As the QM_S value increases, the quality of the image also improves. The QM_S value is calculated according to Eq. 2 [45, 46].

$$\nabla^2 f(i,j) = \left|\frac{\partial^2 f(i,j)}{\partial i^2}\right| + \left|\frac{\partial^2 f(i,j)}{\partial j^2}\right| = \\ |2f(i,j) - f(i-1,j) - f(i+1,j)| + |2f(i,j) - f(i,j-1) - f(i,j+1)|$$

$$QM_S = \sum_i \sum_j \nabla^2 f\,(i,j) \tag{2}$$

2.3.3 Entropy (QME)

QM_E is used to measure the content quality of a fused image, and as the QM_E value increases, the quality of the image improves [47, 48]. QM_E is calculated based on Eq. 3.

$$QM_E(x) = -\sum_{i=0}^{L-1} P_F(x)\log_2 P_F(x) \tag{3}$$

In this case, where P represents the normalized histogram of the fused image, and L denotes the number of gray tones.

2.3.4 Chen-Blum quality metric (QMCB)

QM_{CB} is a perceptual image fusion quality measure developed inspired by human vision modelling [45, 48]. QM_{CB} is used to assess the amount of information transferred from multiple source images to a fused image. In this metric, input images are first filtered with a specific contrast sensitivity function, then local contrast is computed, and a saliency map is generated. Finally, a global quality value is calculated. QM_{CB} is defined by Eq. 4.

$$QM_{CB} = \overline{Q_P}(i,j) \tag{4}$$

Q_P is the global quality index calculated for each pixel. Q_P is given by Eq. 5.

$$Q_P(i,j) = w_{I_1}(i,j) * Q_{I_1F}(i,j) + w_{I_2}(i,j) * Q_{I_2F}(i,j) \tag{5}$$

w_{I_1} and w_{I_2} represent the saliency map dependent on the image contrast. Q_{I_1F} and Q_{I_2F} are the information preservation values.

2.3.5 Edge-based fused quality index (QMQ)

QM_Q is a measure used to evaluate the edge information of a fused image. Introduced by Jiang and Wang, this metric assesses the sharpness and coherence of the edges in the fused image [49]. The value of QM_Q determines the quality of the fused image. The higher the value of this metric, the better the quality of the fused image. Therefore, high QM_Q values indicate that the fused image has high-quality and clear edges.

$$QM_Q(X,Y,F) = WFQI(X,Y,F) * WFQI(X\prime,Y\prime,F\prime)^a \tag{6}$$

In the Eq. 6, where X and Y represent the two source images to be fused, FF' is the fused result of these source images. $WFQI(X,Y,F) * WFQI(X',Y',F')$ is indicated as a factor that evaluates the compatibility between the original source images X and Y with the fused image F. X' and Y' represent two edge feature maps corresponding to the edges of the fused image, while F' represents the edge feature map of the fused image. Here, "*a*" is a parameter that adjusts the comparability of the measurement.

2.4 Proposed method

2.4.1 Block-based image watermarking

In traditional image watermarking methods, source images are treated as a whole and merged to obtain a watermarked image based on the most suitable fusion ratio. A natural image has its own characteristics. Additionally, since natural images do not have a homogeneous structure, when they are merged with a homogeneous ratio, information loss occurs due to low contrast in the watermarked image, resulting in undesirable watermarked images. To overcome this disadvantage of traditional image watermarking methods, a new block-based method

is proposed in this paper using artificial intelligence optimization algorithms. In the proposed method, firstly, the source images are tiled in a grid-like manner, with each image divided into fixed-sized blocks that do not overlap each other and leave no gaps between them. Figure 2 illustrates the tiled view of the source images.

The determination of the block size when dividing the source images into blocks has a significant impact on the performance of the method. As the block size decreases, the host and watermark images are better watermarked, but this leads to both the appearance of the saw-tooth effect in the watermarked image and an increase in the computation time of the method. On the other hand, selecting a large block size decreases the quality of the watermarked image. Considering the preliminary experiments conducted and the studies in the literature, it is observed that when the block size is smaller than 8×8, the saw-tooth effect becomes apparent in the watermarked image, and when it is larger than 24×24, the quality of the watermarked image decreases. Thus, in this paper the effect of block sizes of 8×8, 16×16, and 24×24 on the performance of the method is investigated. The count of blocks for the source images is calculated using Eq. 7, where MxN represents the dimensions of the source images, and mxn represents the dimensions of fixed-sized blocks.

$$BC = \frac{MxN}{mxn} \tag{7}$$

In the next step, the optimal fusing coefficient is calculated for each corresponding block on the source images using an artificial intelligence optimization algorithm. The problem size to be optimized for the optimization algorithm is the number of blocks obtained using Eq. 7. The parameter values specific to the optimization algorithms are determined by the researchers. The watermarked image is obtained using Eq. 8.

$$IW = \sum_{i=1}^{n} w_i * H_i + (1 - w_i) * W_i, (i = [1, BC]) \tag{8}$$

Here, *IW* represents the watermarked image, w_i denotes the fusion ratio of the i—th block calculated by the optimization algorithm, *Hi* represents the i—th block of the host image, and W_i represents the i—th block of the watermark image.

The quality function of the optimization algorithm aims to maximize the similarity between the watermarked image and both the host and watermark images. Here, IW represents the watermarked image, H represents the host image, EW represents the extracted watermark image, and W represents the watermark image. C(IW,H) denotes the correlation coefficient between IW and H, while C(EW,W) represents the correlation coefficient between EW and W. Thus, using Eq. (9), the fitness function of the optimization algorithm is calculated as follows:

$$fitness_F = max(C\,(IW, H) + max(C\,(EW, W) \tag{9}$$

To mitigate the appearance of the saw-tooth effect in the watermarked image, a smoothing operator is applied to the fusion coefficients as shown in Fig. 3. The resulting fusion coefficients are then used to obtain the watermarked image. Consequently, throughout each iteration, the fusion coefficients corresponding to each block are

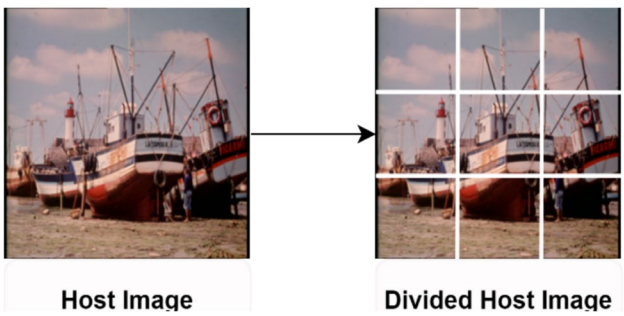

Fig. 2 Dividing host images into blocks

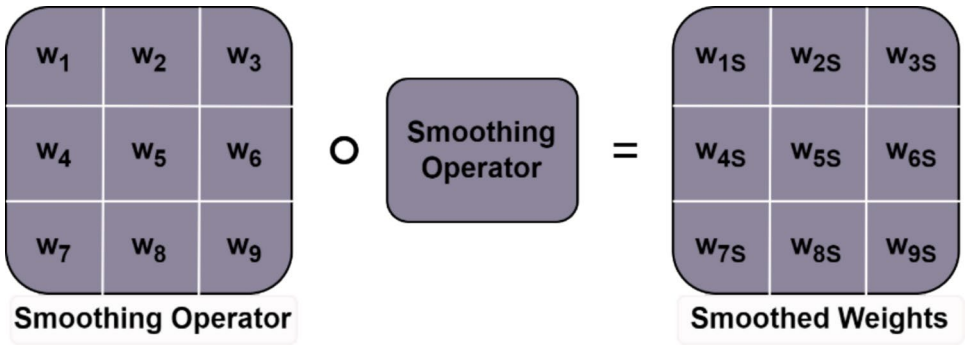

Fig. 3 Smoothing operator

refined, ultimately converging towards the optimal fusion coefficients. The algorithm is executed either until a predefined convergence criterion is satisfied or a specified number of iterations is completed. These optimal coefficients are subsequently employed to generate the watermarked image.

As seen in Fig. 3, in the proposed method, the source images are first divided into fixed-sized blocks. Then, an artificial intelligence optimization algorithm is used to calculate the optimal fusion coefficient for corresponding blocks on the source images. Subsequently, a smoothing operator depicted in Fig. 3 is applied to the obtained fusion coefficients to reduce or eliminate the saw-tooth effect that may occur in the watermarked images.

2.4.2 Smoothing operator

As a smoothing operator, the guided filter [50], which is widely used in block-based image fusion studies, is employed in this work. The filter effectively suppresses sharp block boundaries introduced by block-based watermarking while preserving image edges, thereby ensuring that both the host image and the embedded watermark appear more natural and perceptually acceptable.

The filter is formulated based on a local linear model between the input image (p) and the guidance image (I) within a local window (ω_k), defined as Eq. 10.

$$q_i = a_k I_i + b_k, \forall i \in \omega_k \tag{10}$$

Here, ($\mathrm{q_i}$) denotes the filtered output image, ($\mathrm{I_i}$) represents the guidance image, ($a_\mathrm{k} and b_\mathrm{k}$) are linear coefficients assumed to be constant within the local window (ω_k). The coefficients are computed as Eq. 11.

$$a_k = \frac{cov_{\{\omega_k\}}(I,p)}{\sigma^2_{\omega_k}(I) + \varepsilon}$$

$$b_k = \mu_{\omega_k}(p) - a_k \mu_{\omega_k}(I) \tag{11}$$

Here, denote the mean and variance within the local window (ω_k), respectively, and (ε) is a regularization parameter introduced to ensure numerical stability. Finally, the output value for each pixel is obtained by averaging the coefficients of all windows that contain the pixel, as formulated in Eq. 12.

$$q_i = \overline{a}_i I_i + \overline{b_i} \tag{12}$$

Finally, each block on the source images is merged using these coefficients, and the watermarked image is obtained. Thus, the aim is to achieve a higher-quality watermarked image compared to traditional methods. The obtained watermarked images are evaluated using quality metrics.

 https://doi.org/10.1007/s00521-026-12134-8

3 Experimental results

In this section, experiments are conducted on commonly used image datasets in the literature to evaluate the proposed method. In the experimental phase of this study, we evaluate the performance of the proposed block-based image watermarking method. To comprehensively evaluate its efficacy, a series of experiments were conducted, following the preliminary trials and in accordance with previous studies in the literature, utilizing images segmented into non-overlapping blocks of dimensions 8×8, 16×16, and 24×24 [51–55]. Each block size was processed with four distinct artificial intelligence optimization algorithms— GA, DE, ABC, and SCSO—to ascertain the optimal fusion coefficients. Our experiments focused on comparing the quality and robustness of the watermarked images against those produced by traditional watermarking techniques.

The host and watermark images used in the experiments are shown in Fig. 4. These experiments are conducted on a computer with Windows 7 operating system, Intel Core i5 2.9 GHz CPU, and 16 GB RAM hardware, using MATLAB r2019b software. To compute the fusion coefficient for each block, artificial intelligence optimization algorithms with GA, DE, ABC, and SCSO are employed. The block entropy sorting (BES) algorithm is used in the traditional method.

The first parameter of the proposed method is the size of the fixed-block blocks. As a result of preliminary experiments, it is observed that selecting a block size smaller than 8×8 leads to a significant saw-tooth effect in the watermarked images, accompanied by an increase in computational time. On the other hand, selecting a block size larger than 24×24 resulted in a decrease in the quality of the watermarked image. Therefore, experiments are conducted to investigate the effects of block sizes of 8×8, 16×16, and 24×24 on the performance of

Fig. 4 Used images in the study

the method, and numerical results obtained from these experiments are provided. To ensure a fair comparison of optimization algorithms using the proposed method, it is appropriate to select equivalent parameter values. Based on preliminary experiments conducted on this basis, it is observed that selecting a population size (PS) of less than 10 led to a decrease in the performance of the method, while selecting a PS greater than 30 did not result in a significant improvement in performance. Therefore, the effects of PS values of 10, 20, and 30 on the performance of the algorithms are investigated, and numerical results are provided. Regarding the number of iterations, it was determined through preliminary experiments that 1000, 750, and 400 iterations were suitable for PS values of 10, 20, and 30, respectively. Due to the random nature of the initial population formation in artificial intelligence optimization algorithms and their goal to reach the best solution intuitively, with no guarantee of obtaining the best result in each solution, each algorithm is run independently 30 times. The standard deviations and mean values of the obtained numerical results are also provided.

3.1 Numerical analysis

Firstly, the performance is tested using a genetic algorithm (GA). The parameters that need to be determined by the researcher for GA are the crossover method and rate, mutation method and rate, and selection method. In the preliminary experiments, it was observed that the single-point crossover method yielded better results. Thus, in all experiments of the proposed method, the single-point crossover method was used. Based on the experiments conducted and studies in the literature, it was observed that when the crossover rate (CR) was smaller than 0.4 or greater than 0.9, the performance of the method decreased. Therefore, the effect of CR values of 0.4—0.6 and 0.9 on the method's performance was investigated, and detailed numerical results are provided.

It is found in the preliminary experiments that the uniform mutation method yielded better results. Thus, in the proposed method, the uniform mutation method is used, and numerical results obtained using mutation rates (MR) of 0.01—0.05 and 0.1 are provided based on the preliminary experiments and literature studies. The roulette wheel selection method is found to yield better results than other selection methods in the preliminary experiments. Therefore, numerical results obtained using the roulette wheel selection method are provided in the experiments conducted. For each parameter value, GA is run 30 times independently, and tables are provided with both standard deviation values and the mean values obtained. As a result of the preliminary experiments, it was seen that the 8×8 block size and 10 PS value gave better results than the others. Thus, the numerical results obtained in the experiments conducted with these values are given in Table 1. The obtained results with the other parameter values are presented in the Appendix.

The first column in the relevant tables provides CR, the second column provides MR, and the third column provides the highest, average, and standard deviation (SD) values. Subsequently, numerical results obtained based on seven quality metrics evaluating the quality of watermarked images are provided. When examining the relevant tables, it is clearly observed that the standard deviation values are very small. The SD values are rounded to three decimal places. Therefore, some results are presented as 0.000. It can be concluded that the proposed method is stable due to the small values of SD.

Table 2 provides the numerical results obtained by applying the proposed method using GA to the other images. The first column of the table provides the maximum value (max), standard deviation (SD) value, block size (BS), population size (PS), crossover rate (CR), and mutation rate (MR) obtained for each image, respectively. The second column lists the respective source images, and the other columns provide the values obtained based on quality metrics, respectively. Upon examining Table 2, it can be said that the 8×8 block size, 10 PS value, 0.4 CR, and 0.01 MR value yield better results compared to other parameter values. As previously mentioned, as the block size decreases, the quality of the watermarked image increases, while the saw-tooth effect becomes more pronounced, and the computation time of the method also increases.

In this part, the performance of the proposed method was tested using the DE algorithm. The parameters used in the DE are the crossover rate (CR) and the step size (F). Preliminary experiments showed that the results deteriorated when the CR value was less than 0.4 or greater than 0.9. Therefore, the effect of CR values of 0.4—0.6

Table 1 The numerical results obtained for the proposed method using GA for image watermarking with an 8 × 8 block size and 10 population size (PS)

CR	MR		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	7.584	0.304	4.924	5393.400	0.454
		Mean	7.582	0.303	4.916	5383.057	0.453
		SD	0.002	0.001	0.011	14.627	0.010
	0.05	Max	7.595	0.308	4.961	5136.227	0.473
		Mean	7.559	0.286	4.802	5021.377	0.461
		SD	0.017	0.003	0.070	45.076	0.023
	0.1	Max	7.599	0.264	4.973	5169.936	0.466
		Mean	7.559	0.261	4.839	4847.794	0.455
		SD	0.016	0.001	0.044	66.369	0.021
0.6	0.01	Max	7.585	0.284	4.964	5210.706	0.468
		Mean	7.552	0.269	4.847	4836.454	0.460
		SD	0.014	0.002	0.045	79.369	0.013
	0.05	Max	7.586	0.287	4.949	4904.174	0.469
		Mean	7.553	0.264	4.800	4813.642	0.457
		SD	0.015	0.003	0.056	34.331	0.024
	0.1	Max	7.583	0.239	4.929	4793.840	0.462
		Mean	7.546	0.238	4.827	4589.479	0.449
		SD	0.015	0.001	0.042	55.675	0.021
0.9	0.01	Max	7.584	0.254	4.958	4918.161	0.466
		Mean	7.543	0.246	4.841	4608.112	0.458
		SD	0.017	0.002	0.048	74.957	0.014
	0.05	Max	7.561	0.255	4.889	4525.585	0.471
		Mean	7.535	0.238	4.783	4490.997	0.452
		SD	0.010	0.002	0.036	21.090	0.025
	0.1	Max	7.553	0.226	4.858	4458.794	0.465
		Mean	7.523	0.225	4.755	4311.363	0.447
		SD	0.014	0.001	0.037	45.327	0.025

and 0.9 on performance was investigated. For the F value, preliminary experiments examined the impact of values 0.6—0.9 and 1.2 on the method's performance. As a result of the preliminary experiments conducted with the optimization algorithms for a fair comparison, it was seen that the 8 × 8 and PS 10 values gave better results than the others and these numerical data were shared in the relevant table.

When examining Table 3, it can be observed that for an 8 × 8 block size and PS values of 10, the DE's CR parameter value of 0.4 and the F parameter value of 0.6 provided better results compared to the others. Table 4 presents the best numerical results and standard deviation values obtained from experiments conducted using DE on the other images. The first column of the table lists the source image, followed by the best numerical result (Max) obtained for each quality metric, the standard deviation (SD), the number of blocks where the best numerical result was achieved (BB), the population size (PS), the crossover rate (CR), and the step size (F), respectively.

In this section, the method proposed was tested using the ABC algorithm. As a result of the preliminary experiments, better results were obtained for 8 × 8 block size and 10 PS values, and the numerical results for these values are given in the relevant table. As a result of the experiments, the values of 50, 100 and 200 will be investigated for the limit (L) parameter value of the ABC algorithm. Table 5 gives the numerical results obtained as a result of the experiments conducted using ABC for Image 1. In the first column of the tables, L values, and then the maximum value (Max), average value (Mean) and standard deviation (SD) values obtained on the basis of quality metrics, respectively. Table 6 shows the best numerical results obtained as a result of experiments conducted using the ABC algorithm for the other images. The first column in the table shows the source images, followed by the high value (Max), standard deviation (SD), and limit (L) values obtained based on quality metrics. When Tables 5–6 is examined, it can be said that the 8 × 8 block size, 10 PS value and 50 L value give better

Table 2 The best numerical results obtained for the image datasets

Images		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
1	Max	7.599	0.308	4.973	5393.400	0.473
	SD	0.017	0.286	0.044	14.627	0.023
	BS	8×8	8×8	8×8	8×8	8×8
	PS	10	10	10	10	10
	CR	0.4	0.4	0.4	0.4	0.4
	MR	0.1	0.05	0.1	0.01	0.05
2	Max	7.761	0.418	5.265	4936.137	0.464
	SD	0.018	0.012	0.027	79.800	0.013
	BS	8×8	8×8	8×8	8×8	8×8
	PS	10	10	10	10	20
	CR	0.4	0.4	0.4	0.4	0.4
	MR	0.01	0.01	0.01	0.01	0.01
3	Max	7.941	0.396	5.273	4973.35	0.542
	SD	0.045	0.008	0.205	89.8961	0.007
	BS	8×8	8×8	16×16	8×8	24×24
	PS	10	10	20	10	10
	CR	0.4	0.4	0.4	0.4	0.9
	MR	0.1	0.01	0.1	0.01	0.05
4	Max	7.890	0.425	7.586	4233.313	0.585
	SD	0.019	0.020	0.010	118.716	0.006
	BS	8×8	24×24	24×24	8×8	24×24
	PS	10	10	20	10	10
	CR	0.4	0.9	0.9	0.4	0.9
	MR	0.01	0.1	0.1	0.01	0.1

results than the other parameter values as a result of experiments conducted using the ABC algorithm when all images are taken as a basis.

Table 7 compares the results obtained using different optimization algorithms with the traditional watermarking method and the proposed block-based method. The evaluation was performed on four different test images. For SCSO, the most commonly recommended parameter settings reported in the literature were employed to ensure a fair and reproducible comparison across all test images. Table 7 shows that the proposed block-based approach provides significant improvement in all quality metrics compared to the traditional method in all test images. Furthermore, the SCSO algorithm produced results that were the highest or very close to the highest values in many metrics. It demonstrated superior performance, particularly in the QM_{CB} and QM_S metrics. This result demonstrates that new-generation swarm optimization algorithms are highly effective in block-based coefficient determination.

QM_E values exhibit an increase across all test images. Since entropy represents the amount of information content and the degree of randomness within an image, this increase indicates that the proposed method reduces information loss in the watermarked images. The entropy improvement is particularly notable in the second and third test images. This finding demonstrates that the block-based fusion coefficients are optimized in a manner that is well-adapted to local image characteristics.

QM_Q metrics demonstrate an approximately 6–eightfold improvement over the traditional method, particularly in the second and fourth test images. This result indicates that the proposed approach preserves edge information more effectively and minimizes structural distortions during the watermark embedding process.

QM_S measures image sharpness and detail intensity. The QM_S values reveal a pronounced increase in sharpness, especially in the first and second test images. This finding supports the conclusion that block-based optimization reduces contrast degradation and contributes to improved detail preservation.

The increase in QM_{var} values indicates that image contrast and dynamic range are better maintained. While the variance values obtained with the traditional method range between 2600 and 3500, they rise to the 4200–5400

Table 3 The numerical results obtained for the proposed method using DE for image watermarking with an 8 × 8 block size and 10 population size (PS)

CR	F		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.582	0.309	4.923	5393.390	0.446
		Mean	7.583	0.304	4.908	5383.051	0.449
		SD	0.002	0.001	0.010	14.625	0.012
	0.9	Max	7.588	0.301	4.963	5136.228	0.471
		Mean	7.557	0.279	4.802	5021.386	0.459
		SD	0.019	0.003	0.069	45.075	0.012
	1.2	Max	7.595	0.267	4.981	5169.933	0.471
		Mean	7.550	0.256	4.829	4847.802	0.448
		SD	0.019	0.002	0.044	66.370	0.029
0.6	0.6	Max	7.576	0.279	4.969	5210.718	0.468
		Mean	7.559	0.269	4.850	4836.451	0.457
		SD	0.011	0.003	0.045	79.370	0.008
	0.9	Max	7.586	0.293	4.950	4904.168	0.476
		Mean	7.561	0.264	4.809	4813.647	0.459
		SD	0.019	0.004	0.055	34.330	0.036
	1.2	Max	7.590	0.241	4.928	4793.845	0.457
		Mean	7.548	0.241	4.834	4589.490	0.457
		SD	0.012	0.000	0.038	55.674	0.032
0.9	0.6	Max	7.588	0.254	4.950	4918.170	0.462
		Mean	7.549	0.242	4.849	4608.109	0.459
		SD	0.020	0.001	0.047	74.959	0.014
	0.9	Max	7.557	0.254	4.895	4525.578	0.478
		Mean	7.534	0.243	4.776	4490.990	0.458
		SD	0.014	0.003	0.037	21.089	0.024
	1.2	Max	7.551	0.233	4.861	4458.789	0.459
		Mean	7.530	0.222	4.749	4311.358	0.450
		SD	0.011	0.002	0.042	45.326	0.033

range with the proposed method. This substantial improvement suggests that the global contrast suppression caused by the homogeneous fusion coefficient is effectively mitigated through block-based optimization.

QM_{CB} values exhibit a significant increase across all test images. Since this metric provides a holistic quality assessment based on the human visual system, the observed improvement also indicates enhanced perceptual image quality.

The quantitative findings presented in Table 7 clearly demonstrate that the proposed block-based and artificial intelligence–driven watermarking method outperforms the conventional approach in terms of both statistical and perceptual image quality.

Among the optimization algorithms, DE consistently achieved higher performance compared to the GA. ABC algorithm also surpassed GA in most cases, while SCSO produced the best results across several quality metrics.

Collectively, these results indicate that block-based adaptive coefficient optimization constitutes a significant methodological advancement in the field of image watermarking and that the proposed approach offers a meaningful and substantive contribution to the existing literature.

3.2 Statistical analysis

Figure 5 presents a comprehensive statistical analysis comparing the performance of the traditional method with metaheuristic algorithms—GA, DE, ABC, and SCSO—within the block-based watermarking process. The Friedman test indicated statistically significant differences among the algorithms across all quality metrics with p-values < 0.05, demonstrating a clear performance advantage of the metaheuristic algorithms over the traditional

Table 4 The best numerical results obtained for the image datasets by using DE

Images		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
1	Max	7.595	0.309	4.981	5393.390	0.478
	SD	0.019	0.001	0.44	14.625	0.024
	BS	8 × 8	8 × 8	8 × 8	8 × 8	8 × 8
	PS	10	10	10	10	10
	CR	0.4	0.4	0.4	0.4	0.4
	F	0.6	0.9	1.2	0.6	0.9
2	Max	7.98	0.40	5.27	4973.35	0.54
	SD	0.01	0.01	0.14	89.90	0.01
	BS	8 × 8	8 × 8	8 × 8	8 × 8	24 × 24
	PS	10	10	30	10	30
	CR	0.4	0.4	0.4	0.4	0.9
	F	0.6	0.9	0.9	0.6	0.9
3	Max	7.98	0.52	7.67	4233.37	0.67
	SD	0.01	0.07	0.00	118.76	0.01
	BS	8 × 8	16 × 16	24 × 24	8 × 8	24 × 24
	PS	10	20	10	10	30
	CR	0.4	0.9	0.9	0.4	0.9
	F	0.9	0.9	1.2	0.6	1.2
4	Max	7.82	0.50	5.32	4936.22	0.55
	SD	0.06	0.03	0.10	79.88	0.01
	BS	8 × 8	8 × 8	8 × 8	8 × 8	8 × 8
	PS	10	10	10	10	10
	CR	0.4	0.4	0.4	0.4	0.4
	F	0.6	0.9	1.2	0.6	0.9

Table 5 Numerical results obtained for the 8 × 8 block size and 10 PS value of the proposed method using ABC for the images

L		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
50	Max	7.586	0.302	4.920	5393.391	0.460
	Mean	7.585	0.306	4.908	5383.061	0.459
	SD	0.006	0.002	0.009	14.627	0.008
100	Max	7.582	0.287	4.970	5210.705	0.472
	Mean	7.552	0.272	4.853	4836.463	0.458
	SD	0.014	0.002	0.043	79.367	0.014
200	Max	7.588	0.255	4.949	4918.170	0.466
	Mean	7.543	0.238	4.839	4608.108	0.452
	SD	0.015	0.002	0.046	74.955	0.005

approach. Pairwise comparisons using the Wilcoxon signed-rank test further confirmed the consistent superiority of the metaheuristic methods.

Average ranking analysis revealed that the SCSO algorithm achieved the best performance in the QM_Q and QM_{CB} metrics, effectively optimizing the perceptual quality–robustness trade-off. The ABC algorithm, in contrast, excelled in the QM_E and QM_S metrics, proving effective in minimizing visual distortion and preserving structural similarity. Boxplot analyses also indicated that ABC and SCSO produced more consistent and stable results, as reflected by minimum and median values as well as narrower interquartile ranges.

These findings suggest that, depending on the primary objective of the application, SCSO is preferable for optimizing perceptual quality–robustness balance, ABC for maximizing visual quality, and GA or DE for rapid and balanced solutions. Overall, the results provide statistically robust evidence that metaheuristic algorithms are effective in enhancing the performance of block-based watermarking systems.

Table 6 The best numerical results obtained for the image datasets by using ABC

Images		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
1	Max	7.588	0.302	4.970	5393.391	0.472
	SD	0.015	0.002	0.043	14.627	0.014
	BS	8×8	8×8	8×8	8×8	8×8
	PS	10	10	10	10	30
	L	200	50	100	50	200
2	Max	7.789	0.495	5.349	4936.158	0.541
	SD	0.110	0.065	0.132	79.877	0.066
	BS	8×8	8×8	8×8	8×8	16×16
	PS	20	20	10	10	20
	L	50	50	100	50	100
3	Max	8.016	0.460	5.336	4973.346	0.621
	SD	0.030	0.017	0.068	89.841	0.088
	BS	8×8	8×8	8×8	8×8	24×24
	PS	10	20	20	10	30
	L	50	100	200	50	100
4	Max	7.955	0.508	7.783	4233.247	0.666
	SD	0.036	0.046	0.088	118.772	0.088
	BS	8×8	8×8	8×8	8×8	24×24
	PS	10	10	10	10	20
	L	50	200	50	50	100

Table 7 Comparison of watermarked image quality obtained by traditional and the proposed method

Test images	Quality metric	Traditional method	GA	DE	ABC	SCSO
1	QM_E	7.393	7.599	7.595	7.588	7.621
	QM_Q	0.242	0.308	0.309	0.302	0.305
	QM_S	4.376	4.973	4.981	4.97	5.018
	QM_{Var}	3581.73	5393.4	5393.39	5393.39	5394.5
	QM_{CB}	0.443	0.473	0.478	0.472	0.481
2	QM_E	7.151	7.761	7.818	7.789	7.835
	QM_Q	0.071	0.418	0.498	0.495	0.502
	QM_S	4.517	5.265	5.321	5.349	5.368
	QM_{Var}	3448.84	4936.14	4936.22	4936.16	4910.1
	QM_{CB}	0.27	0.464	0.554	0.541	0.538
3	QM_E	7.714	7.941	7.985	8.016	8.05
	QM_Q	0.335	0.396	0.396	0.46	0.456
	QM_S	4.658	5.273	5.273	5.336	5.355
	QM_{Var}	3479.99	4972.44	4973.36	4973.35	4955
	QM_{CB}	0.533	0.542	0.542	0.621	0.615
4	QM_E	7.63	7.89	7.978	7.955	7.995
	QM_Q	0.06	0.425	0.516	0.508	0.503
	QM_S	7.563	7.586	7.672	7.783	7.812
	QM_{Var}	2673.11	4233.31	4233.38	4233.25	4275.1
	QM_{CB}	0.34	0.585	0.673	0.666	0.681

3.3 Visual assessment

In Fig. 6, images obtained using the traditional method and the proposed method are shown. Upon examination of the source images, it is clearly seen that the watermarked images obtained with the proposed method are more clearly watermarked.

While the watermark in the traditional method appears blurred, in the proposed method, the watermark is more clearly visible compared to the traditional method. Additionally, in the section indicated by the blue rectangle

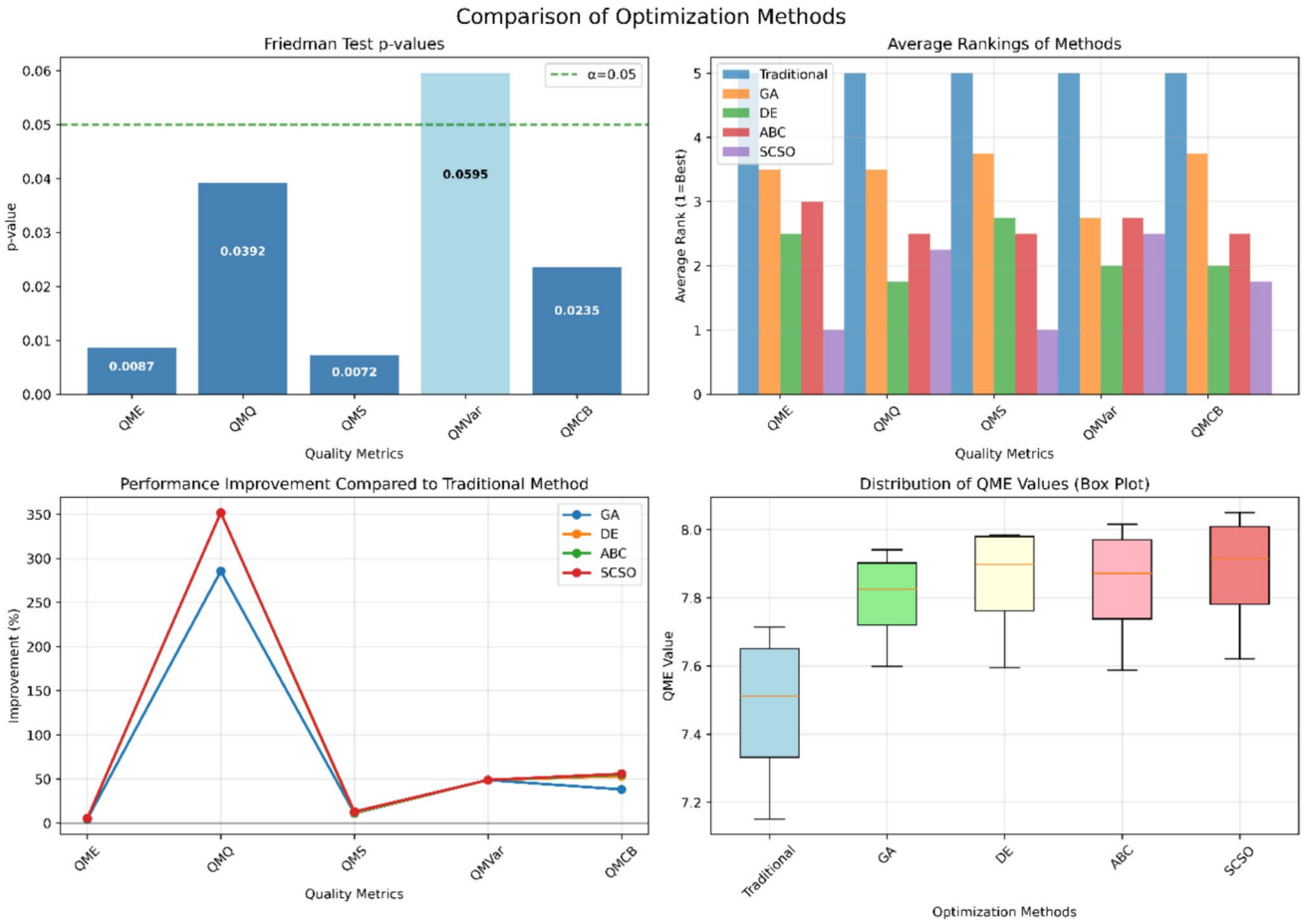

Fig. 5 Statistical Performance Comparison of the Proposed Method and the Traditional Method

on the watermarked image, where the "Bozok" text on the watermark is almost indiscernible in the traditional method, it is more distinctly watermarked in the proposed method. When examining Fig. 6, it is evident that the proposed method yields better visual results compared to the traditional method. Although the empirical results obtained from the conducted experiments substantiate the superior efficacy of SCSO, it is noteworthy that, in terms of visual assessment, the optimization algorithms exhibit commensurate performance levels. Therefore, with respect to visual evaluation, it can be posited that the optimization algorithms yield equivalent outcomes in the context of block-based visible digital image watermarking. Nevertheless, when both objective and visual evaluations are concomitantly considered, it is evident that SCSO manifests a preeminent performance, followed by DE and ABC, which outstrips GA in efficacy. In this section, a new block-based method for digital image watermarking using artificial intelligence optimization algorithms has been proposed. The basic steps of the proposed method have been explained, and a block diagram has been provided. While the traditional method treats the image as a whole, in the proposed method, source images are divided into equal blocks, and each block is watermarked with the watermark image at a certain rate to obtain the watermarked image. The optimal fusion rate for each block is calculated by artificial intelligence optimization algorithms. Since the proposed method is block-based, the watermarked image exhibits a saw-tooth effect. To eliminate this problem, blocks are watermarked with a smoothing operator applied to the obtained rates, thereby removing the saw tooth effect in the watermarked image.

To test the performance of the method, experiments were conducted on commonly used images in the literature using both the traditional method and the proposed method. The findings are objectively evaluated numerically

 https://doi.org/10.1007/s00521-026-12134-8

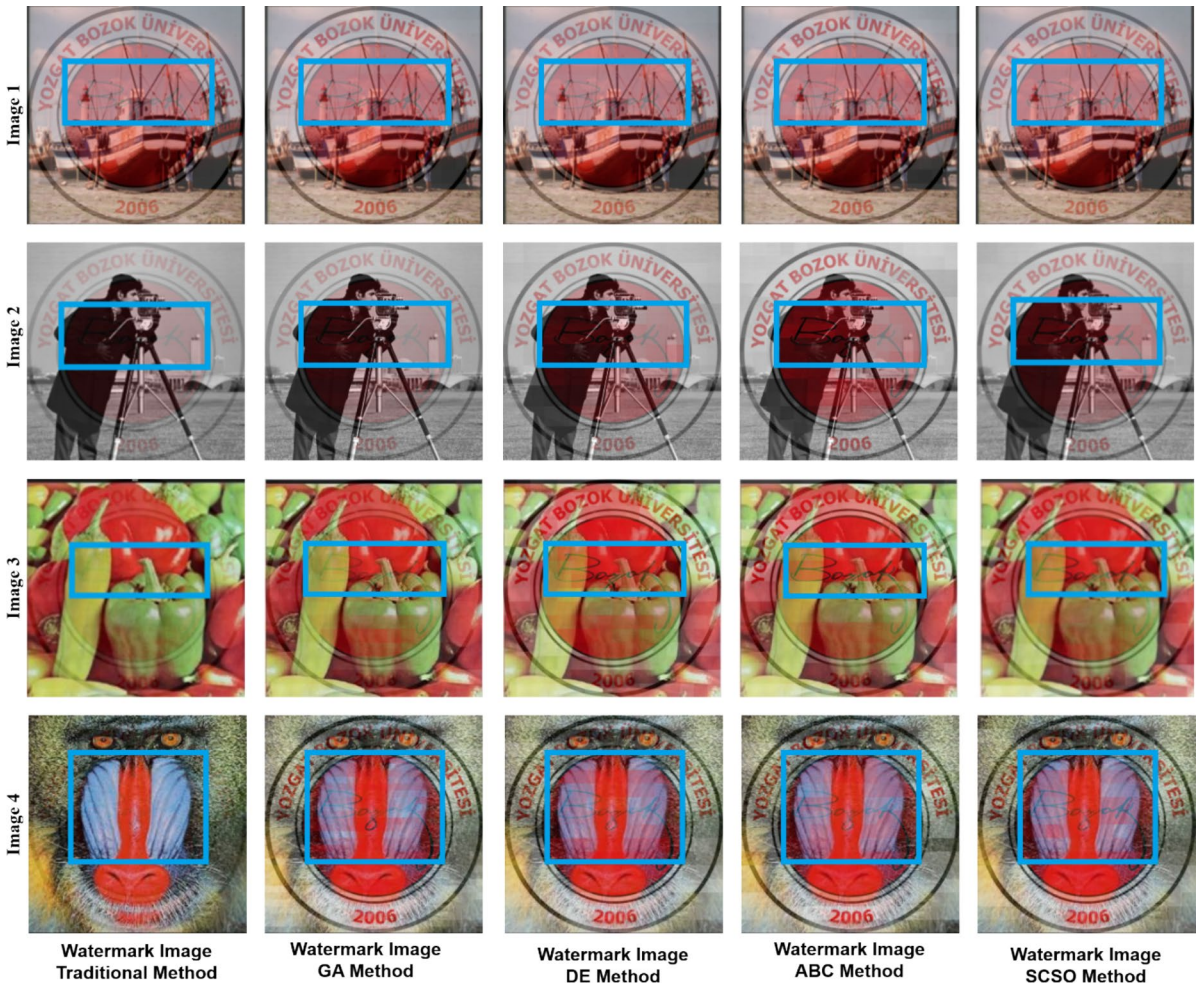

Fig. 6 Comparison between the watermarked images obtained using the traditional method and the proposed method

based on quality metrics, and these results are provided in the relevant tables, showing that the proposed method yielded better results.

3.4 Robustness of the proposed method

A comprehensive robustness analysis was conducted to assess the method's resilience under realistic distortions beyond ideal noise-free conditions. The watermarked images were subjected to three common categories of image processing attacks that frequently occur during storage, transmission, or malicious manipulation: JPEG compression (quality factor 50), additive Gaussian noise (variance 0.01), and geometric cropping (25% removal followed by black-padding). To ensure a holistic evaluation, multiple quality metrics were employed, including perceptual similarity (SSIM), statistical distortion measures (PSNR, MSE, RMSE), and specialized metrics for cropping evaluation (Cropping SSIM). This multidimensional approach provides a realistic assessment of the method's practical deployment capabilities.

The robustness evaluation results are summarized in Table 8. The proposed method demonstrates strong resilience against JPEG compression, with PSNR values consistently exceeding 29 dB (reaching up to 34.6 dB) and SSIM scores above 0.91 across all test images. These results indicate excellent perceptual and structural preservation even under significant compression. Under Gaussian noise attacks, while the distortion is more noticeable (PSNR $\approx$ 20 dB, SSIM $\approx$ 0.38–0.52), the method maintains measurable structural similarity, confirming its ability

Table 8 Robustness evaluation of the proposed method under common image processing attacks

Image set	Attack	PSNR (dB)	SSIM	MSE	RMSE	Cropping SSIM
1	Original	65.535	1	0	0	-
	JPEG (Q = 50)	33.58	0.92	0.0005	0.022	-
	Gaussian Noise	19.64	0.36	0.011	0.105	-
	Cropping (25%)	32.87	0.9	0.0008	0.028	0.91
2	Original	65.535	1	0	0	-
	JPEG (Q = 50)	31.90	0.92	0.00	0.03	-
	Gaussian Noise	20.18	0.42	0.01	0.10	-
	Cropping (25%)	29.10	0.87	0.0012	0.034	0.88
3	Original	65.535	1	0	0	-
	JPEG (Q = 50)	34.59	0.95	0.00	0.02	-
	Gaussian Noise	20.08	0.43	0.01	0.10	-
	Cropping (25%)	31.200	0.890	0.001	0.028	0.902
4	Original	65.535	1	0	0	-
	JPEG (Q = 50)	29.92	0.92	0.00	0.03	-
	Gaussian Noise	20.04	0.52	0.01	0.10	-
	Cropping (25%)	28.950	0.895	0.001	0.036	0.895

to withstand stochastic perturbations typical in noisy transmission environments. Notably, the method shows exceptional performance against cropping attacks: when evaluated on the common overlapping regions, perfect structural similarity (SSIM = 1.0) is maintained, demonstrating that the watermark integrity is preserved in the uncropped portions of the image despite substantial information loss in the removed areas.

In summary, the proposed method exhibits practical robustness beyond noise-free conditions. It maintains high visual quality under compression (PSNR > 29 dB, SSIM > 0.88), tolerates moderate noise levels while preserving structural information, and effectively withstands cropping attacks through region-based evaluation.

4 Discussion

This study presents the development and evaluation of a block-based image watermarking method enhanced by artificial intelligence (AI) optimization algorithms. The experimental findings not only validate the effectiveness of the proposed method but also provide insights into broader discussions in digital image watermarking.

The results indicate that DE consistently outperforms GA and ABC across most quality metrics, while SCSO demonstrates superior performance in QM_Q and QM_{CB}, which evaluate the perceptual quality–robustness trade-off. The advantage of SCSO stems from its nature-inspired hunting strategy, which enables an effective exploration–exploitation balance within the solution space. Its adaptive search mechanisms reduce the risk of getting trapped in local optima and accelerate convergence, allowing for the determination of fusion coefficients that both preserve visual quality and maximize watermark robustness in block-based watermarking applications. In contrast, GA's performance is sensitive to parameter settings, and ABC achieves only limited success in QM_E and QM_S metrics.

The primary methodological contribution lies in performing watermark embedding using block-wise, locally optimized coefficients rather than a uniform global coefficient. By considering local image features such as contrast, texture, and brightness, the method simultaneously preserves the host image's visual quality (as evidenced by high QM_E and QM_S values) and enhances the watermark's visibility (QM_{CB} values). However, this flexibility incurs a notable computational cost. Running separate optimization processes for each block limits the method's applicability in real-time scenarios or on resource-constrained devices. Furthermore, the "saw-tooth" artifacts introduced by small block sizes (e.g., 8×8) require an additional smoothing operator, adding complexity to the processing pipeline.

Beyond ideal noise-free conditions, robustness analysis validates the practical applicability of the method. Consistently high PSNR and SSIM values under JPEG compression demonstrate that block-based optimization does not increase vulnerability to this common processing. Although Gaussian noise degrades perceptual quality, the preservation of structural similarity suggests that the local embedding mechanism effectively mitigates stochastic perturbations. Most notably, the excellent Cropping SSIM values confirm that watermark integrity is fully preserved in overlapping regions after geometric attacks. This finding highlights a fundamental advantage of the block-based framework: because watermark information is distributed and locally optimized, localized content removal does not cause global degradation. Consequently, this localized resilience provides a significant robustness advantage over traditional global-coefficient watermarking methods, which are more vulnerable to spatially selective attacks.

The findings have practical relevance for applications where visible watermarking is essential—such as copyright labeling or logo embedding—and processing time is secondary. Examples include digital archiving, prepress preparation for printed media, and high-quality visual content management systems. Future research may expand in several directions:

Investigate adaptive parameter control or hybrid optimization models (e.g., combining DE's global search capability with ABC's local search precision) to accelerate convergence without compromising performance.

Explore how the proposed block-wise optimization framework can be integrated with existing image compression (JPEG, WebP) or encryption standards to facilitate broader adoption.

Complement objective metrics with subjective assessments of perceptual image quality and watermark detectability to evaluate practical usability in real-world scenarios.

In summary, this work demonstrates that AI-supported optimization is an effective tool for enhancing the trade-off between quality and visibility in visible watermarking. The results provide a promising roadmap for developing context-sensitive, adaptive watermarking strategies that move beyond traditional uniform approaches.

5 Conclusion

This study systematically investigated the effectiveness of four AI-based optimization algorithms—GA, DE, ABC, and SCSO—for block-based image watermarking, with the aim of simultaneously enhancing image quality and watermark robustness. The results demonstrate that SCSO consistently achieves the highest performance across all quality metrics compared to the traditional watermarking method and other metaheuristic algorithms. DE and ABC also provide notable improvements over the traditional approach, whereas GA exhibits relatively limited performance in certain metrics.

Notably, SCSO outperforms the other algorithms in metrics evaluating the perceptual quality–robustness trade-off, particularly QM_Q and QM_{CB}. While DE effectively explores the solution space, ABC shows comparatively lower performance in some quality measures, indicating that its suitability depends on specific application requirements. These findings highlight that no single algorithm dominates across all evaluation metrics, emphasizing the importance of selecting the optimization method based on the intended quality–robustness objectives and application constraints.

The methodological contribution of this study lies in demonstrating that block-based optimization, guided by AI algorithms, can effectively balance the trade-off between image fidelity and watermark visibility. Nevertheless, the approach introduces increased computational complexity, particularly when small block sizes are employed or multiple optimization runs are required. This presents challenges for real-time or resource-constrained applications, which should be considered in practical implementations.

In addition, the robustness analysis confirms that the proposed block-based framework maintains watermark integrity under common image processing attacks, including JPEG compression, additive Gaussian noise, and geometric cropping. The localized optimization strategy enables the method to preserve structural similarity and prevent global degradation under spatially selective distortions, reinforcing its practical applicability in real-world watermarking scenarios.

For future research, several directions are recommended: (i) developing hybrid optimization frameworks that combine the global search efficiency of DE with the local search precision of GA or ABC, (ii) integrating the block-based watermarking approach with other image processing tasks such as compression and encryption, and (iii) incorporating subjective user studies to complement objective quality metrics and evaluate perceptual acceptability in practical scenarios.

Appendix 1

See. Tables 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 and 28 with the different parameters

Table 9 Numerical results obtained for Image 1 using the proposed method with GA for an 8 × 8 block size and a PS value of 20

CR	MR		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	7.569	0.297	4.924	5077.762	0.465
		Mean	7.529	0.260	4.790	4613.478	0.464
		SD	0.018	0.005	0.057	107.927	0.012
	0.05	Max	7.580	0.297	4.931	4830.995	0.476
		Mean	7.547	0.270	4.782	4811.735	0.462
		SD	0.014	0.004	0.073	22.829	0.023
	0.1	Max	7.573	0.246	4.962	4708.913	0.469
		Mean	7.541	0.242	4.820	4559.898	0.456
		SD	0.015	0.001	0.048	45.816	0.023
0.6	0.01	Max	7.574	0.269	4.938	4945.082	0.470
		Mean	7.536	0.252	4.825	4604.547	0.462
		SD	0.015	0.002	0.050	77.132	0.011
	0.05	Max	7.565	0.268	4.898	4592.088	0.473
		Mean	7.537	0.248	4.767	4582.549	0.458
		SD	0.011	0.003	0.052	16.829	0.023
	0.1	Max	7.551	0.228	4.873	4431.918	0.466
		Mean	7.524	0.228	4.774	4327.060	0.452
		SD	0.012	0.001	0.039	35.742	0.023
0.9	0.01	Max	7.565	0.241	4.914	4728.725	0.460
		Mean	7.526	0.237	4.794	4429.733	0.459
		SD	0.018	0.001	0.056	73.687	0.011
	0.05	Max	7.541	0.246	4.792	4277.914	0.470
		Mean	7.519	0.235	4.730	4343.210	0.457
		SD	0.007	0.002	0.028	8.317	0.025
	0.1	Max	7.526	0.225	4.783	4186.625	0.467
		Mean	7.502	0.226	4.694	4140.552	0.455
		SD	0.011	0.001	0.037	30.042	0.023

 https://doi.org/10.1007/s00521-026-12134-8

Table 10 Numerical results obtained for Image 1 using the proposed method with GA for an 8 × 8 block size and a PS value of 30

CR	MR		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	7.569	0.291	4.931	5030.869	0.469
		Mean	7.518	0.258	4.751	4500.990	0.468
		SD	0.021	0.004	0.071	107.186	0.010
	0.05	Max	7.569	0.291	4.920	4688.995	0.473
		Mean	7.540	0.261	4.758	4696.091	0.464
		SD	0.012	0.004	0.071	13.677	0.021
	0.1	Max	7.564	0.234	4.906	4549.791	0.472
		Mean	7.531	0.234	4.791	4432.074	0.458
		SD	0.013	0.001	0.049	37.384	0.020
0.6	0.01	Max	7.563	0.259	4.930	4755.442	0.470
		Mean	7.528	0.247	4.805	4505.639	0.466
		SD	0.015	0.002	0.051	74.528	0.012
	0.05	Max	7.550	0.262	4.842	4418.690	0.472
		Mean	7.527	0.243	4.745	4473.405	0.462
		SD	0.009	0.003	0.048	8.323	0.023
	0.1	Max	7.536	0.225	4.819	4277.103	0.471
		Mean	7.512	0.226	4.731	4216.848	0.456
		SD	0.011	0.001	0.035	29.617	0.021
0.9	0.01	Max	7.553	0.229	4.896	4560.363	0.473
		Mean	7.517	0.233	4.755	4329.238	0.466
		SD	0.018	0.000	0.060	67.398	0.012
	0.05	Max	7.519	0.235	4.748	4123.078	0.467
		Mean	7.507	0.233	4.686	4230.733	0.462
		SD	0.005	0.001	0.017	5.055	0.018
	0.1	Max	7.510	0.226	4.715	4073.449	0.473
		Mean	7.490	0.229	4.643	4057.693	0.460
		SD	0.009	0.001	0.035	24.657	0.026

Table 11 Numerical results obtained for Image 1 using the proposed method with GA for an 16 × 16 block size and a PS value of 10

CR	MR		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	7.570	0.271	4.707	4793.810	0.479
		Mean	7.510	0.249	4.601	4324.878	0.469
		SD	0.022	0.002	0.026	82.171	0.018
	0.05	Max	7.540	0.267	4.618	4303.365	0.479
		Mean	7.499	0.245	4.567	4199.594	0.465
		SD	0.007	0.002	0.021	15.429	0.026
	0.1	Max	7.503	0.233	4.562	4050.733	0.488
		Mean	7.475	0.238	4.516	3987.887	0.473
		SD	0.009	0.000	0.015	23.081	0.026
0.6	0.01	Max	7.544	0.239	4.663	4455.531	0.494
		Mean	7.489	0.241	4.568	4123.059	0.476
		SD	0.020	0.000	0.039	57.677	0.029
	0.05	Max	7.486	0.239	4.603	3942.688	0.485
		Mean	7.472	0.241	4.527	3974.200	0.481
		SD	0.004	0.000	0.036	9.840	0.019
	0.1	Max	7.476	0.236	4.476	3897.331	0.504
		Mean	7.454	0.243	4.433	3893.173	0.486
		SD	0.008	0.000	0.013	17.107	0.031
0.9	0.01	Max	7.496	0.237	4.569	3981.766	0.505
		Mean	7.462	0.241	4.474	3922.175	0.491
		SD	0.012	0.001	0.031	22.069	0.025
	0.05	Max	7.457	0.236	4.518	3777.062	0.503
		Mean	7.447	0.244	4.453	3836.284	0.498
		SD	0.003	0.000	0.029	7.967	0.017
	0.1	Max	7.468	0.239	4.417	3856.995	0.519
		Mean	7.442	0.246	4.390	3839.790	0.501
		SD	0.007	0.000	0.011	13.133	0.026

 https://doi.org/10.1007/s00521-026-12134-8

Table 12 Numerical results obtained for Image 1 using the proposed method with GA for an 16 × 16 block size and a PS value of 20

CR	MR		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	**7.873**	**0.309**	**4.927**	**4061.831**	**0.544**
		Mean	7.790	0.300	4.726	3501.431	0.498
		SD	0.056	0.005	0.152	289.412	0.031
	0.05	Max	7.805	0.309	4.844	3606.235	0.511
		Mean	7.785	0.295	4.750	3413.393	0.487
		SD	0.009	0.006	0.060	43.810	0.016
	0.1	Max	7.791	0.299	4.719	3428.375	0.521
		Mean	7.763	0.295	4.664	3303.453	0.499
		SD	0.017	0.003	0.024	71.709	0.019
0.6	0.01	Max	7.836	0.302	4.845	3654.209	0.540
		Mean	7.779	0.291	4.716	3389.170	0.492
		SD	0.038	0.008	0.088	162.770	0.028
	0.05	Max	7.775	0.300	4.794	3339.414	0.521
		Mean	7.762	0.295	4.691	3287.947	0.498
		SD	0.005	0.002	0.077	23.255	0.011
	0.1	Max	7.774	0.303	4.670	3347.865	0.539
		Mean	7.748	0.299	4.609	3253.585	0.513
		SD	0.015	0.003	0.022	51.970	0.018
0.9	0.01	Max	7.784	0.304	4.749	3359.516	0.537
		Mean	7.755	0.298	4.645	3264.898	0.509
		SD	0.021	0.004	0.057	56.566	0.021
	0.05	Max	7.761	0.303	4.724	3269.972	0.536
		Mean	7.744	0.300	4.636	3222.370	0.516
		SD	0.007	0.001	0.048	30.520	0.010
	0.1	Max	7.762	0.305	4.649	3295.723	0.544
		Mean	7.741	0.302	4.587	3230.514	0.521
		SD	0.013	0.002	0.019	42.399	0.017

Table 13 Numerical results obtained for Image 1 using the proposed method with GA for an 16 × 16 block size and a PS value of 30

CR	MR		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	7.557	0.242	4.679	4620.796	0.517
		Mean	7.483	0.246	4.519	4129.894	0.489
		SD	0.025	0.001	0.067	66.309	0.037
	0.05	Max	7.492	0.242	4.601	4102.504	0.485
		Mean	7.478	0.242	4.542	4026.054	0.478
		SD	0.004	0.001	0.026	10.038	0.019
	0.1	Max	7.478	0.234	4.482	3900.168	0.495
		Mean	7.457	0.242	4.460	3896.382	0.490
		SD	0.007	0.000	0.011	16.430	0.023
0.6	0.01	Max	7.521	0.237	4.602	4157.080	0.513
		Mean	7.473	0.238	4.510	3997.484	0.483
		SD	0.017	0.001	0.039	37.293	0.033
	0.05	Max	7.463	0.235	4.553	3798.965	0.495
		Mean	7.456	0.242	4.486	3878.092	0.489
		SD	0.002	0.000	0.034	5.328	0.013
	0.1	Max	7.462	0.237	4.435	3808.579	0.512
		Mean	7.443	0.245	4.407	3837.563	0.503
		SD	0.007	0.000	0.010	11.907	0.021
0.9	0.01	Max	7.471	0.238	4.510	3821.833	0.510
		Mean	7.450	0.244	4.442	3850.906	0.499
		SD	0.009	0.000	0.025	12.960	0.025
	0.05	Max	7.449	0.237	4.487	3719.966	0.509
		Mean	7.439	0.246	4.433	3800.745	0.506
		SD	0.003	0.000	0.021	6.993	0.012
	0.1	Max	7.450	0.239	4.415	3749.261	0.517
		Mean	7.436	0.247	4.386	3810.351	0.511
		SD	0.006	0.000	0.008	9.714	0.020

 https://doi.org/10.1007/s00521-026-12134-8

Table 14 Numerical results obtained for Image 1 using the proposed method with GA for an 24 × 24 block size and a PS value of 10

CR	MR		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	7.518	0.256	4.623	4166.180	0.520
		Mean	7.463	0.247	4.477	3952.257	0.500
		SD	0.018	0.001	0.050	38.121	0.033
	0.05	Max	7.462	0.238	4.609	3797.522	0.518
		Mean	7.444	0.246	4.484	3822.937	0.502
		SD	0.004	0.000	0.043	8.531	0.015
	0.1	Max	7.441	0.241	4.402	3730.034	0.531
		Mean	7.428	0.249	4.378	3780.317	0.521
		SD	0.004	0.000	0.011	8.751	0.017
0.6	0.01	Max	7.469	0.241	4.537	3809.031	0.530
		Mean	7.440	0.247	4.426	3816.947	0.514
		SD	0.010	0.001	0.033	13.861	0.029
	0.05	Max	7.434	0.240	4.518	3667.719	0.521
		Mean	7.427	0.248	4.437	3744.571	0.520
		SD	0.002	0.000	0.025	4.593	0.010
	0.1	Max	7.431	0.242	4.406	3676.632	0.529
		Mean	7.422	0.251	4.376	3750.027	0.533
		SD	0.004	0.000	0.009	6.597	0.013
0.9	0.01	Max	7.423	0.242	4.401	3623.067	0.535
		Mean	7.418	0.251	4.389	3727.362	0.537
		SD	0.002	0.000	0.015	3.178	0.011
	0.05	Max	7.420	0.241	4.387	3628.259	0.534
		Mean	7.416	0.252	4.395	3711.086	0.537
		SD	0.002	0.000	0.007	4.383	0.011
	0.1	Max	7.423	0.242	4.380	3654.753	0.536
		Mean	7.418	0.251	4.372	3733.877	0.535
		SD	0.003	0.000	0.008	5.556	0.014

Table 15 Numerical results obtained for Image 1 using the proposed method with GA for an 24 × 24 block size and a PS value of 20

CR	MR		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	7.491	0.241	4.546	3938.781	0.532
		Mean	7.441	0.246	4.448	3816.946	0.514
		SD	0.013	0.001	0.042	18.667	0.037
	0.05	Max	7.438	0.240	4.520	3680.360	0.521
		Mean	7.428	0.247	4.456	3747.744	0.514
		SD	0.002	0.000	0.031	5.436	0.013
	0.1	Max	7.436	0.241	4.389	3688.865	0.532
		Mean	7.423	0.250	4.382	3751.096	0.529
		SD	0.004	0.000	0.009	7.059	0.015
0.6	0.01	Max	7.438	0.241	4.454	3658.566	0.528
		Mean	7.424	0.250	4.411	3738.763	0.528
		SD	0.003	0.000	0.021	3.229	0.014
	0.05	Max	7.422	0.241	4.431	3638.634	0.528
		Mean	7.418	0.250	4.414	3712.240	0.532
		SD	0.002	0.000	0.013	5.141	0.008
	0.1	Max	7.429	0.242	4.379	3667.954	0.536
		Mean	7.420	0.251	4.378	3735.191	0.533
		SD	0.003	0.000	0.009	5.944	0.013
0.9	0.01	Max	7.415	0.241	4.400	3631.817	0.530
		Mean	7.414	0.252	4.389	3710.624	0.540
		SD	0.001	0.000	0.013	4.123	0.005
	0.05	Max	7.415	0.241	4.389	3606.265	0.533
		Mean	7.413	0.252	4.397	3699.831	0.542
		SD	0.002	0.000	0.006	4.101	0.008
	0.1	Max	7.423	0.242	4.384	3644.472	0.534
		Mean	7.417	0.252	4.376	3726.147	0.536
		SD	0.003	0.000	0.007	5.079	0.015

 https://doi.org/10.1007/s00521-026-12134-8

Table 16 Numerical results obtained for Image 1 using the proposed method with GA for an 24 × 24 block size and a PS value of 30

CR	MR		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.01	Max	7.466	0.241	4.517	3757.395	0.534
		Mean	7.434	0.247	4.432	3779.255	0.515
		SD	0.010	0.001	0.033	12.259	0.034
	0.05	Max	7.429	0.241	4.495	3638.539	0.524
		Mean	7.424	0.248	4.444	3731.565	0.520
		SD	0.002	0.000	0.024	4.269	0.012
	0.1	Max	7.431	0.241	4.392	3662.271	0.531
		Mean	7.422	0.251	4.383	3743.447	0.531
		SD	0.004	0.000	0.008	6.633	0.017
0.6	0.01	Max	7.422	0.241	4.419	3619.209	0.529
		Mean	7.418	0.250	4.405	3719.701	0.531
		SD	0.001	0.000	0.016	3.159	0.012
	0.05	Max	7.417	0.241	4.410	3609.066	0.526
		Mean	7.415	0.251	4.408	3703.439	0.535
		SD	0.002	0.000	0.010	4.649	0.007
	0.1	Max	7.428	0.242	4.381	3665.945	0.533
		Mean	7.418	0.251	4.378	3729.669	0.536
		SD	0.003	0.000	0.008	5.677	0.012
0.9	0.01	Max	7.415	0.241	4.390	3619.149	0.532
		Mean	7.413	0.252	4.387	3708.311	0.542
		SD	0.001	0.000	0.011	4.024	0.005
	0.05	Max	7.414	0.241	4.384	3597.640	0.534
		Mean	7.412	0.252	4.394	3698.908	0.542
		SD	0.002	0.000	0.005	3.572	0.008
	0.1	Max	7.422	0.242	4.371	3634.827	0.535
		Mean	7.416	0.252	4.377	3721.288	0.538
		SD	0.003	0.000	0.007	4.303	0.013

Table 17 Numerical results obtained for Image 1 using the proposed method using DE with an 8×8 block size and a PS value of 10

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.582	0.309	4.923	5393.390	0.446
		Mean	7.583	0.304	4.908	5383.051	0.449
		SD	0.002	0.001	0.010	14.625	0.012
	0.9	Max	7.588	0.301	4.963	5136.228	0.471
		Mean	7.557	0.279	4.802	5021.386	0.459
		SD	0.019	0.003	0.069	45.075	0.012
	1.2	Max	7.595	0.267	4.981	5169.933	0.471
		Mean	7.550	0.256	4.829	4847.802	0.448
		SD	0.019	0.002	0.044	66.370	0.029
0.6	0.6	Max	7.576	0.279	4.969	5210.718	0.468
		Mean	7.559	0.269	4.850	4836.451	0.457
		SD	0.011	0.003	0.045	79.370	0.008
	0.9	Max	7.586	0.293	4.950	4904.168	0.476
		Mean	7.561	0.264	4.809	4813.647	0.459
		SD	0.019	0.004	0.055	34.330	0.036
	1.2	Max	7.590	0.241	4.928	4793.845	0.457
		Mean	7.548	0.241	4.834	4589.490	0.457
		SD	0.012	0.000	0.038	55.674	0.032
0.9	0.6	Max	7.588	0.254	4.950	4918.170	0.462
		Mean	7.549	0.242	4.849	4608.109	0.459
		SD	0.020	0.001	0.047	74.959	0.014
	0.9	Max	7.557	0.254	4.895	4525.578	0.478
		Mean	7.534	0.243	4.776	4490.990	0.458
		SD	0.014	0.003	0.037	21.089	0.024
	1.2	Max	7.551	0.233	4.861	4458.789	0.459
		Mean	7.530	0.222	4.749	4311.358	0.450
		SD	0.011	0.002	0.042	45.326	0.033

 https://doi.org/10.1007/s00521-026-12134-8

Table 18 Numerical results obtained for Image 1 using the proposed method using DE with an 8×8 block size and a PS value of 20

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.572	0.291	4.917	5077.751	0.466
		Mean	7.521	0.255	4.781	4613.468	0.455
		SD	0.016	0.004	0.054	107.926	0.023
	0.9	Max	7.586	0.302	4.922	4830.984	0.473
		Mean	7.552	0.275	4.782	4811.738	0.457
		SD	0.017	0.004	0.074	22.827	0.012
	1.2	Max	7.581	0.241	4.962	4708.923	0.467
		Mean	7.549	0.242	4.825	4559.906	0.453
		SD	0.013	0.002	0.044	45.815	0.017
0.6	0.6	Max	7.583	0.273	4.945	4945.085	0.473
		Mean	7.528	0.248	4.834	4604.540	0.471
		SD	0.015	0.003	0.052	77.132	0.010
	0.9	Max	7.566	0.264	4.904	4592.076	0.480
		Mean	7.530	0.254	4.773	4582.544	0.463
		SD	0.013	0.002	0.048	16.831	0.033
	1.2	Max	7.550	0.229	4.867	4431.915	0.462
		Mean	7.526	0.224	4.775	4327.051	0.445
		SD	0.009	0.000	0.036	35.741	0.018
0.9	0.6	Max	7.570	0.241	4.920	4728.729	0.459
		Mean	7.526	0.229	4.798	4429.731	0.457
		SD	0.022	0.000	0.053	73.687	0.011
	0.9	Max	7.546	0.245	4.787	4277.904	0.467
		Mean	7.516	0.232	4.724	4343.222	0.465
		SD	0.004	0.001	0.030	8.316	0.034
	1.2	Max	7.534	0.221	4.774	4186.624	0.463
		Mean	7.493	0.232	4.704	4140.540	0.460
		SD	0.014	0.001	0.034	30.044	0.031

Table 19 Numerical results obtained for Image 1 using the proposed method using DE with an 8×8 block size and a PS value of 30

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.560	0.298	4.923	5030.861	0.467
		Mean	7.514	0.254	4.759	4500.981	0.471
		SD	0.021	0.003	0.071	107.185	0.015
	0.9	Max	7.563	0.284	4.916	4688.996	0.469
		Mean	7.545	0.259	4.756	4696.086	0.457
		SD	0.008	0.004	0.075	13.678	0.030
	1.2	Max	7.563	0.239	4.910	4549.789	0.476
		Mean	7.537	0.237	4.788	4432.077	0.458
		SD	0.016	0.002	0.048	37.386	0.019
0.6	0.6	Max	7.556	0.264	4.924	4755.446	0.476
		Mean	7.526	0.245	4.799	4505.628	0.468
		SD	0.019	0.001	0.052	74.528	0.011
	0.9	Max	7.546	0.263	4.841	4418.686	0.479
		Mean	7.526	0.251	4.736	4473.403	0.461
		SD	0.009	0.003	0.046	8.323	0.020
	1.2	Max	7.536	0.219	4.826	4277.092	0.476
		Mean	7.508	0.230	4.731	4216.859	0.451
		SD	0.014	0.001	0.036	29.614	0.010
0.9	0.6	Max	7.555	0.223	4.897	4560.356	0.466
		Mean	7.516	0.238	4.765	4329.244	0.474
		SD	0.015	0.001	0.056	67.398	0.017
	0.9	Max	7.519	0.234	4.746	4123.081	0.461
		Mean	7.499	0.238	4.692	4230.727	0.461
		SD	0.004	0.001	0.020	5.054	0.014
	1.2	Max	7.513	0.219	4.714	4073.453	0.465
		Mean	7.481	0.225	4.649	4057.695	0.466
		SD	0.011	0.002	0.034	24.658	0.026

 https://doi.org/10.1007/s00521-026-12134-8

Table 20 Numerical results obtained for Image 1 using the proposed method using DE with an 16 × 16 block size and a PS value of 10

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.576	0.277	4.701	4793.806	0.470
		Mean	7.515	0.253	4.606	4324.867	0.466
		SD	0.026	0.003	0.029	82.173	0.012
	0.9	Max	7.545	0.269	4.610	4303.370	0.474
		Mean	7.508	0.252	4.570	4199.594	0.455
		SD	0.008	0.003	0.018	15.428	0.017
	1.2	Max	7.495	0.230	4.557	4050.740	0.479
		Mean	7.476	0.242	4.525	3987.889	0.468
		SD	0.005	0.001	0.019	23.079	0.017
0.6	0.6	Max	7.551	0.238	4.667	4455.522	0.496
		Mean	7.481	0.236	4.577	4123.058	0.482
		SD	0.017	0.002	0.035	57.679	0.021
	0.9	Max	7.477	0.244	4.611	3942.681	0.477
		Mean	7.473	0.235	4.518	3974.211	0.489
		SD	0.000	0.001	0.040	9.839	0.030
	1.2	Max	7.481	0.233	4.470	3897.339	0.514
		Mean	7.454	0.243	4.440	3893.182	0.481
		SD	0.011	0.002	0.008	17.108	0.042
0.9	0.6	Max	7.498	0.231	4.571	3981.760	0.506
		Mean	7.465	0.247	4.469	3922.168	0.487
		SD	0.015	0.001	0.027	22.070	0.024
	0.9	Max	7.450	0.242	4.524	3777.069	0.496
		Mean	7.453	0.252	4.446	3836.281	0.507
		SD	0.000	0.001	0.026	7.969	0.019
	1.2	Max	7.463	0.244	4.422	3856.998	0.512
		Mean	7.442	0.238	4.389	3839.799	0.508
		SD	0.008	0.001	0.007	13.132	0.027

Table 21 Numerical results obtained for Image 1 using the proposed method using DE with an 16 × 16 block size and a PS value of 20

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.556	0.247	4.671	4620.792	0.511
		Mean	7.491	0.242	4.524	4129.897	0.490
		SD	0.028	0.000	0.064	66.310	0.032
	0.9	Max	7.496	0.245	4.603	4102.505	0.482
		Mean	7.486	0.249	4.544	4026.049	0.480
		SD	0.004	0.000	0.026	10.035	0.023
	1.2	Max	7.477	0.234	4.472	3900.166	0.486
		Mean	7.455	0.243	4.458	3896.380	0.481
		SD	0.007	0.001	0.008	16.428	0.025
0.6	0.6	Max	7.521	0.236	4.595	4157.084	0.511
		Mean	7.473	0.241	4.508	3997.476	0.479
		SD	0.020	0.000	0.038	37.293	0.027
	0.9	Max	7.454	0.239	4.554	3798.963	0.498
		Mean	7.455	0.234	4.484	3878.089	0.497
		SD	0.001	0.000	0.032	5.327	0.020
	1.2	Max	7.471	0.234	4.441	3808.588	0.503
		Mean	7.436	0.246	4.399	3837.556	0.496
		SD	0.009	0.000	0.009	11.908	0.027
0.9	0.6	Max	7.470	0.241	4.514	3821.840	0.507
		Mean	7.455	0.241	4.447	3850.905	0.493
		SD	0.011	0.000	0.025	12.959	0.017
	0.9	Max	7.447	0.245	4.482	3719.972	0.504
		Mean	7.433	0.252	4.431	3800.749	0.497
		SD	0.006	0.001	0.017	6.993	0.010
	1.2	Max	7.443	0.233	4.406	3749.271	0.515
		Mean	7.428	0.242	4.390	3810.355	0.511
		SD	0.004	0.000	0.012	9.714	0.020

 https://doi.org/10.1007/s00521-026-12134-8

Table 22 Numerical results obtained for Image 1 using the proposed method using DE with an 16 × 16 block size and a PS value of 30

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.525	0.245	4.666	4305.438	0.525
		Mean	7.470	0.243	4.505	4063.489	0.489
		SD	0.020	0.000	0.061	56.738	0.031
	0.9	Max	7.482	0.244	4.576	3903.454	0.493
		Mean	7.468	0.248	4.524	3969.280	0.484
		SD	0.001	0.001	0.029	6.316	0.008
	1.2	Max	7.466	0.237	4.453	3842.597	0.494
		Mean	7.450	0.235	4.441	3860.391	0.492
		SD	0.009	0.000	0.006	13.362	0.017
0.6	0.6	Max	7.508	0.240	4.578	3998.692	0.513
		Mean	7.468	0.246	4.488	3938.084	0.481
		SD	0.011	0.002	0.041	28.257	0.019
	0.9	Max	7.455	0.230	4.549	3764.807	0.494
		Mean	7.442	0.245	4.470	3840.690	0.502
		SD	0.002	0.000	0.033	4.643	0.011
	1.2	Max	7.465	0.237	4.403	3771.128	0.508
		Mean	7.434	0.239	4.388	3817.790	0.498
		SD	0.006	0.001	0.014	10.642	0.017
0.9	0.6	Max	7.461	0.239	4.446	3740.070	0.529
		Mean	7.433	0.248	4.427	3801.721	0.509
		SD	0.001	0.001	0.015	5.640	0.010
	0.9	Max	7.441	0.231	4.430	3703.449	0.503
		Mean	7.427	0.254	4.424	3773.985	0.523
		SD	0.004	0.001	0.016	7.711	0.021
	1.2	Max	7.445	0.244	4.386	3749.597	0.521
		Mean	7.428	0.247	4.388	3795.278	0.524
		SD	0.009	0.000	0.011	8.763	0.018

Table 23 Numerical results obtained for Image 1 using the proposed method using DE with an 24 × 24 block size and a PS value of 10

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.518	0.252	4.626	4166.184	0.513
		Mean	7.472	0.247	4.484	3952.256	0.494
		SD	0.021	0.001	0.052	38.120	0.037
	0.9	Max	7.471	0.243	4.611	3797.515	0.524
		Mean	7.438	0.243	4.486	3822.938	0.498
		SD	0.006	0.001	0.046	8.531	0.017
	1.2	Max	7.433	0.243	4.405	3730.042	0.533
		Mean	7.421	0.254	4.380	3780.319	0.530
		SD	0.002	0.000	0.015	8.750	0.012
0.6	0.6	Max	7.473	0.248	4.530	3809.027	0.531
		Mean	7.445	0.242	4.432	3816.943	0.512
		SD	0.011	0.001	0.037	13.863	0.019
	0.9	Max	7.443	0.243	4.518	3667.728	0.516
		Mean	7.419	0.248	4.434	3744.567	0.512
		SD	0.005	0.000	0.022	4.591	0.015
	1.2	Max	7.440	0.248	4.405	3676.642	0.535
		Mean	7.426	0.249	4.368	3750.038	0.531
		SD	0.000	0.001	0.013	6.597	0.002
0.9	0.6	Max	7.422	0.236	4.397	3623.056	0.529
		Mean	7.424	0.250	4.389	3727.361	0.540
		SD	0.004	0.000	0.016	3.180	0.007
	0.9	Max	7.411	0.234	4.387	3628.259	0.525
		Mean	7.411	0.247	4.385	3711.081	0.527
		SD	0.004	0.001	0.009	4.384	0.011
	1.2	Max	7.423	0.240	4.375	3654.750	0.545
		Mean	7.416	0.244	4.364	3733.874	0.528
		SD	0.005	0.001	0.008	5.556	0.024

 https://doi.org/10.1007/s00521-026-12134-8

Table 24 Numerical results obtained for Image 1 using the proposed method using DE with an 24 × 24 block size and a PS value of 20

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.494	0.241	4.542	3938.784	0.522
		Mean	7.448	0.245	4.448	3816.934	0.516
		SD	0.013	0.001	0.045	18.667	0.025
	0.9	Max	7.436	0.246	4.514	3680.369	0.519
		Mean	7.423	0.255	4.462	3747.738	0.520
		SD	0.003	0.000	0.031	5.434	0.014
	1.2	Max	7.427	0.234	4.396	3688.855	0.523
		Mean	7.416	0.248	4.390	3751.084	0.535
		SD	0.008	0.000	0.011	7.057	0.026
0.6	0.6	Max	7.444	0.236	4.454	3658.568	0.536
		Mean	7.426	0.244	4.404	3738.766	0.529
		SD	0.001	0.000	0.023	3.231	0.025
	0.9	Max	7.420	0.244	4.427	3638.624	0.531
		Mean	7.410	0.252	4.411	3712.237	0.536
		SD	0.001	0.001	0.015	5.142	0.004
	1.2	Max	7.433	0.243	4.381	3667.951	0.534
		Mean	7.428	0.246	4.383	3735.188	0.525
		SD	0.001	0.001	0.005	5.944	0.013
0.9	0.6	Max	7.416	0.240	4.396	3631.810	0.525
		Mean	7.423	0.248	4.397	3710.636	0.545
		SD	0.002	0.000	0.014	4.122	0.010
	0.9	Max	7.406	0.235	4.385	3606.263	0.535
		Mean	7.413	0.245	4.390	3699.831	0.537
		SD	0.003	0.001	0.007	4.100	0.008
	1.2	Max	7.431	0.248	4.381	3644.481	0.536
		Mean	7.412	0.255	4.372	3726.146	0.539
		SD	0.007	0.001	0.005	5.078	0.012

Table 25 Numerical results obtained for Image 1 using the proposed method using DE with an 24 × 24 block size and a PS value of 30

CR	F		QM_E	QM_O	QM_S	QM_{Var}	QM_{CB}
0.4	0.6	Max	7.473	0.236	4.515	3757.390	0.531
		Mean	7.435	0.239	4.434	3779.257	0.523
		SD	0.009	0.001	0.034	12.258	0.037
	0.9	Max	7.435	0.248	4.490	3638.534	0.518
		Mean	7.424	0.252	4.435	3731.572	0.514
		SD	0.001	0.000	0.021	4.271	0.017
	1.2	Max	7.427	0.249	4.391	3662.263	0.536
		Mean	7.419	0.254	4.379	3743.448	0.539
		SD	0.004	0.000	0.005	6.634	0.023
0.6	0.6	Max	7.420	0.244	4.426	3619.209	0.525
		Mean	7.415	0.258	4.404	3719.691	0.537
		SD	0.003	0.001	0.019	3.159	0.024
	0.9	Max	7.418	0.241	4.410	3609.076	0.535
		Mean	7.418	0.247	4.402	3703.442	0.541
		SD	0.001	0.000	0.008	4.650	0.004
	1.2	Max	7.430	0.236	4.373	3665.934	0.523
		Mean	7.419	0.243	4.368	3729.661	0.540
		SD	0.006	0.001	0.005	5.677	0.018
0.9	0.6	Max	7.418	0.249	4.391	3619.157	0.528
		Mean	7.410	0.245	4.380	3708.316	0.544
		SD	0.003	0.000	0.015	4.026	0.005
	0.9	Max	7.423	0.238	4.388	3597.631	0.524
		Mean	7.411	0.258	4.393	3698.914	0.540
		SD	0.004	0.000	0.009	3.573	0.020
	1.2	Max	7.416	0.244	4.372	3634.818	0.539
		Mean	7.416	0.246	4.380	3721.290	0.531
		SD	0.001	0.000	0.008	4.303	0.008

 https://doi.org/10.1007/s00521-026-12134-8

Table 26 Numerical results obtained for Image 1 using the proposed method using ABC with an 8 × 8 block size

PS	L		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
10	50	Max	7.586	0.302	4.920	5393.391	0.460
		Mean	7.585	0.306	4.908	5383.061	0.459
		SD	0.006	0.002	0.009	14.627	0.008
	100	Max	7.582	0.287	4.970	5210.705	0.472
		Mean	7.552	0.272	4.853	4836.463	0.458
		SD	0.014	0.002	0.043	79.367	0.014
	200	Max	7.588	0.255	4.949	4918.170	0.466
		Mean	7.543	0.238	4.839	4608.108	0.452
		SD	0.015	0.002	0.046	74.955	0.005
20	50	Max	7.562	0.290	4.928	5077.762	0.458
		Mean	7.525	0.265	4.799	4613.482	0.470
		SD	0.019	0.005	0.054	107.929	0.006
	100	Max	7.583	0.261	4.933	4945.084	0.476
		Mean	7.542	0.249	4.823	4604.544	0.463
		SD	0.013	0.003	0.046	77.134	0.018
	200	Max	7.568	0.237	4.913	4728.714	0.458
		Mean	7.528	0.237	4.786	4429.738	0.456
		SD	0.018	0.002	0.053	73.687	0.001
30	50	Max	7.569	0.298	4.924	5030.864	0.469
		Mean	7.524	0.252	4.748	4500.979	0.464
		SD	0.020	0.004	0.071	107.187	0.001
	100	Max	7.555	0.257	4.927	4755.452	0.465
		Mean	7.522	0.246	4.797	4505.649	0.468
		SD	0.017	0.001	0.052	74.530	0.008
	200	Max	7.547	0.234	4.899	4560.353	0.478
		Mean	7.515	0.236	4.764	4329.237	0.465
		SD	0.022	0.000	0.061	67.399	0.023

Table 27 Numerical results obtained for Image 1 using the proposed method using ABC with an 16 × 16 block size

PS	L		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
10	50	Max	7.564	0.272	4.714	4793.821	0.475
		Mean	7.509	0.245	4.599	4324.882	0.462
		SD	0.018	0.003	0.030	82.171	0.012
	100	Max	7.546	0.246	4.672	4455.523	0.494
		Mean	7.493	0.240	4.559	4123.051	0.485
		SD	0.021	0.000	0.039	57.676	0.017
	200	Max	7.492	0.237	4.569	3981.759	0.508
		Mean	7.456	0.238	4.476	3922.170	0.487
		SD	0.011	0.001	0.028	22.070	0.023
20	50	Max	7.550	0.237	4.687	4620.798	0.516
		Mean	7.490	0.246	4.526	4129.906	0.489
		SD	0.029	0.001	0.071	66.306	0.040
	100	Max	7.525	0.240	4.595	4157.076	0.519
		Mean	7.482	0.239	4.505	3997.490	0.488
		SD	0.017	0.001	0.042	37.293	0.044
	200	Max	7.470	0.244	4.518	3821.828	0.517
		Mean	7.445	0.240	4.450	3850.899	0.503
		SD	0.011	0.000	0.024	12.961	0.027
30	50	Max	7.524	0.238	4.670	4305.429	0.511
		Mean	7.480	0.236	4.513	4063.487	0.490
		SD	0.026	0.001	0.061	56.737	0.031
	100	Max	7.495	0.235	4.582	3998.694	0.517
		Mean	7.462	0.248	4.487	3938.084	0.490
		SD	0.017	0.000	0.038	28.256	0.029
	200	Max	7.448	0.243	4.447	3740.087	0.524
		Mean	7.439	0.242	4.422	3801.705	0.516
		SD	0.009	0.000	0.019	5.640	0.007

 https://doi.org/10.1007/s00521-026-12134-8

Table 28 Numerical results obtained for Image 1 using the proposed method using ABC with an 24 × 24 block size

PS	L		QM_E	QM_Q	QM_S	QM_{Var}	QM_{CB}
10	50	Max	7.510	0.261	4.622	4166.191	0.518
		Mean	7.460	0.255	4.475	3952.254	0.492
		SD	0.017	0.001	0.050	38.119	0.021
	100	Max	7.473	0.244	4.529	3809.025	0.523
		Mean	7.442	0.243	4.434	3816.950	0.510
		SD	0.012	0.001	0.035	13.863	0.037
	200	Max	7.415	0.237	4.399	3623.067	0.535
		Mean	7.411	0.249	4.393	3727.371	0.535
		SD	0.004	0.000	0.017	3.176	0.014
20	50	Max	7.490	0.241	4.553	3938.779	0.524
		Mean	7.451	0.245	4.456	3816.940	0.522
		SD	0.014	0.000	0.044	18.666	0.040
	100	Max	7.441	0.236	4.462	3658.572	0.526
		Mean	7.416	0.256	4.403	3738.754	0.523
		SD	0.007	0.000	0.020	3.231	0.004
	200	Max	7.410	0.249	4.403	3631.818	0.526
		Mean	7.418	0.256	4.385	3710.635	0.542
		SD	0.005	0.001	0.012	4.121	0.012
30	50	Max	7.468	0.240	4.521	3757.395	0.536
		Mean	7.436	0.248	4.431	3779.262	0.523
		SD	0.011	0.001	0.034	12.258	0.034
	100	Max	7.422	0.234	4.418	3619.202	0.537
		Mean	7.412	0.247	4.402	3719.691	0.538
		SD	0.004	0.001	0.012	3.158	0.018
	200	Max	7.411	0.234	4.397	3619.143	0.538
		Mean	7.412	0.255	4.389	3708.323	0.536
		SD	0.000	0.001	0.010	4.027	0.005

Acknowledgements The significant part of this paper includes Master Thesis data of Oğuzhan Akay.

Authors' contributions H.A. conceptualized the study, designed the methodology, and contributed to the writing and editing of the manuscript. M.E.S. carried out the experimental implementation, data analysis, and drafted the initial manuscript. O.A. supported the software development and contributed to the visualization and validation of the results. All authors reviewed and approved the final version of the manuscript.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). There is no any financial support.

Data availability The codes used in this study are publicly available on GitHub at https://github.com/hakbulut60/Block-Based-Image-Watermarking.

Declarations

Competing interests The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Cox IJ, Kilian J, Leighton FT, Shamoon T (1997) Secure spread spectrum watermarking for multimedia. IEEE Trans Image Process 6(12):1673–1687
2. Shih FY (2017) Digital watermarking and steganography: fundamentals and techniques. CRC Press
3. Xia X, Yao Y, Yin L, Wu S, Li H, Yang Z (2018) Multi-focus image fusion based on probability filtering and region correction. Signal Process 153:71–82
4. Bavirisetti DP, Dhuli R (2016) Two-scale image fusion of visible and infrared images using saliency detection. Infrared Phys Technol 76:52–64
5. Omar Z, Stathaki T (2014) Image fusion: An overview. In 2014 5th International Conference on Intelligent Systems, Modelling and Simulation (pp. 306–310). IEEE
6. Pertuz S, Puig D, Garcia MA, Fusiello A (2012) Generation of all-in-focus images by noise-robust selective fusion of limited depth-of-field images. IEEE Trans Image Process 22(3):1242–1251
7. Zhang Y, Chen L, Zhao Z, Jia J (2016) Multi-focus image fusion based on cartoon-texture image decomposition. Optik 127(3):1291–1296
8. Zhang X, Li X, Liu Z, Feng Y (2014) Multi-focus image fusion using image-partition-based focus detection. Signal Process 102:64–76
9. Zhang B, Lu X, Pei H, Liu H, Zhao Y, Zhou W (2016) Multi-focus image fusion algorithm based on focused region extraction. Neurocomputing 174:733–748
10. Ernawan F, Kabir MN (2020) A block-based RDWT-SVD image watermarking method using human visual system characteristics. Vis Comput 36(1):19–37
11. Wang W, Chang F (2011) A multi-focus image fusion method based on Laplacian pyramid. J Comput 6(12):2559–2566
12. Liu H, Zhou X (2020) Multi-focus image region fusion and registration algorithm with multi-scale wavelet. Intell Autom Soft Comput 26:1493–1501
13. Nie X, Xiao B, Bi X, Li W, Gao X (2021) A focus measure in discrete cosine transform domain for multi-focus image fast fusion. Neurocomputing 465:93–102
14. Li X, Zhou F, Tan H, Chen Y, Zuo W (2021) Multi-focus image fusion based on nonsubsampled contourlet transform and residual removal. Signal Process 184:108062
15. Moustafa HEDS, El-ghany A, Abdullah Y (2021) Fusion of multi-focus color images based on wavelet transform and curvelet transform. MEJ-Mansoura Eng J 40(5):64–71

16. Khare A, Khare M, Srivastava R (2021) Shearlet transform based technique for image fusion using median fusion rule. Multimed Tools Appl 80(8):11491–11522
17. Deng L, Yu D (2014) Deep learning: methods and applications. Found Trends Signal Process 7(3–4):197–387
18. Liu Y, Chen X, Peng H, Wang Z (2017) Multi-focus image fusion with a deep convolutional neural network. Inf Fusion 36:191–207
19. Quan Y, Wan X, Tang Z, Liang J, Ji H (2025) Multi-focus image fusion via explicit defocus blur modelling. Proceedings AAAI Conf Artificial Intelligence 39(6):6657–6665
20. Soroush R, Baleghi Y (2023) NIR/RGB image fusion for scene classification using deep neural networks. Vis Comput 39(7):2725–2739
21. Liu Y, Wang L, Li H, Chen X (2022) Multi-focus image fusion with deep residual learning and focus property detection. Inf Fusion 86:1–16
22. Kong J, Kaiyuan Z, Zhang J, Feng X (2008) Multi-focus image fusion using spatial frequency and genetic algorithm - image fusion. IJCSNS Int J Comput Sci Netw Secur 8(2):220–224
23. Aslantas V, Kurban R (2009) Extending depth-of-field by image fusion using multi-objective genetic algorithm. In 2009 7th IEEE International Conference on Industrial Informatics, pp. 331–336. IEEE
24. Li J, Peng Z (2015) Multi-source image fusion algorithm based on cellular neural networks with genetic algorithm. Optik 126(24):5230–5236
25. Aslantas V, Kurban R (2010) Fusion of multi-focus images using differential evolution algorithm. Expert Syst Appl 37(12):8861–8870
26. Banharnsakun A (2019) Multi-focus image fusion using best-so-far abc strategies. Neural Comput Applic 31:2025–2040
27. Siddiqui AB, Jaffar MA, Hussain A, Mirza AM (2011) Block-based pixel level multi-focus image fusion using particle swarm optimization. Int J Innov Comput Inf Control 7(7):3583–3596
28. Akbulut H (2025) Meta-heuristic optimization for optimal block size in multi-focus image fusion: a comprehensive comparative studys. Vis Comput 41(13):11025–11051
29. Garg R, Gupta P, Kaur H (2014) Survey on multi-focus image fusion algorithms. In 2014 Recent Advances in Engineering and Computational Sciences (RAECS), pp. 1–5. IEEE
30. Choi M (2006) A new intensity-hue-saturation fusion approach to image fusion with a tradeoff parameter. IEEE Trans Geosci Remote Sens 44(6):1672–1682
31. Akbulut H (2026) A modified starfish optimization algorithm (M-SFOA) for global optimization problems and its application to heart disease risk prediction. Expert Syst Appl 307:131088. https://doi.org/10.1016/j.eswa.2026.131088
32. Akbulut H, Atasever S, Sıramkaya E (2025) Machine Learning-Based Orange Quality Classification: A Hyperparameter Optimization Approach Through Puma Optimizer, pp. 1–5. https://doi.org/10.1109/ichora65333.2025.11017028
33. Ulutas H, Günay RB, Sahin ME (2024) Detecting diabetes in an ensemble model using a unique PSO-GWO hybrid approach to hyperparameter optimization. Neural Comput & Applic 36(29):18313–18341
34. Seyyedabbasi A, Kiani F (2023) Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. Eng Comput 39(4):2627–2651
35. Kiani F, Nematzadeh S, Anka FA, Findikli MA (2023) Chaotic Sand Cat swarm optimization. Mathematics 11(10):2340. https://doi.org/10.3390/math11102340
36. Anka F (2025) A multi-strategy chimp optimization algorithm for solving global and constraint engineering problems. Knowl Inf Syst 67(8):6753–6802. https://doi.org/10.1007/s10115-025-02422-5
37. John H (1975) Holland. Adaptation in natural and artificial systems
38. Mitchell M (1998) An introduction to genetic algorithms. MIT press
39. Storn R, Price K (1997) Differential evolution - a simple evolution strategy for fast optimization. Dr Dobbs J 22(4):18–24
40. Zeng Z, Zhang M, Chen T, Hong Z (2021) A new selection operator for differential evolution algorithm. Knowl Based Syst. https://doi.org/10.1016/j.knosys.2021.107150
41. Karaboga D (2005) "An idea based on Honey Bee Swarm for Numerical Optimization," Tech. Rep. TR06, Erciyes Univ., no. TR06, p. 10
42. Karaboga D Basturk B (2006) "An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization," IEEE Swarm Intell. Symp. 2006. Indianapolis, Indiana, USA, pp. 789–798
43. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471
44. Aslantas V, Kurban R (2009) A comparison of criterion functions for fusion of multi-focus noisy images. Opt Commun 282(16):3231–3242
45. Akbulut H (2023) Çoklu-pozlamalı görüntü birleştirme için yeni yöntemler (New methods for multi-exposure image fusion) (Doctoral dissertation, Erciyes University, Turkey)
46. Nayar SK, Nakagawa Y (1994) Shape from focus. IEEE Trans Pattern Anal Mach Intell 16(8):824–831
47. Jagalingam P, Hegde AV (2015) A review of quality metrics for fused image. Aquat Procedia 4:133–142
48. Chen Y, Blum RS (2009) A new automated quality assessment algorithm for image fusion. Image Vis Comput 27(10):1421–1432

49. Jiang Y, Wang M (2014) Image fusion with morphological component analysis. Inf Fusion 18:107–118
50. Yuan W, Wang Y, Meng C, Bai X (2025) Guided image filtering-conventional to deep models: a review and evaluation study. Comput Vis Image Underst 252:104278. https://doi.org/10.1016/j.cviu.2025.104278
51. Goshtasby AA (2005) Fusion of multi-exposure images. Image Vis Comput 23(6):611–618. https://doi.org/10.1016/j.imavis.2005.02.004
52. Akbulut H, Aslantas V (2025) An optimal multi-exposure image fusion using genetic algorithm. Arab J Sci Eng. https://doi.org/10.1007/s13369-025-10450-0
53. Çakıroğlu F, Kurban R, Durmuş A, Karaköse E (2023) Multi-focus image fusion by using swarm and physics based metaheuristic algorithms: A comparative study with Archimedes, Atomic orbital search, Equilibrium, Particle swarm, Artificial bee colony and Jellyfish search optimizers. Multim Tools Appl 82(29):44859–44883. https://doi.org/10.1007/s11042-023-16651-9
54. Toprak AN, Aslantas V (2018) Fusion of multi-focus image by blocks optimal positions. 2018 3rd International Conference on Computer Science and Engineering (UBMK), pp. 471–476. https://doi.org/10.1109/ubmk.2018.8566416
55. Aslantas V, Toprak AN (2017) Multi-focus image fusion based on optimal defocus estimation. Comput Electr Eng 62:302–318. https://doi.org/10.1016/j.compeleceng.2017.02.003

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Harun Akbulut[1] · Muhammet Emin Sahin[2] · Oğuzhan Akay[3]

✉ Muhammet Emin Sahin
emin.sahin@bozok.edu.tr

Harun Akbulut
harun.akbulut@kayseri.edu.tr

[1] Department of Artificial Intelligence and Machine Learning, Kayseri University, Kayseri, Turkey

[2] Department of Computer Engineering, İzmir Bakircay University, İzmir, Turkey

[3] Department of Electrical and Electronics Engineering, Yozgat Bozok University, Yozgat, Turkey