



An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation

Essam H. Houssein^{a,*}, Kashif Hussain^b, Laith Abualigah^{c,d}, Mohamed Abd Elaziz^e,
Waleed Alomoush^f, Gaurav Dhiman^g, Youcef Djenouri^h, Erik Cuevasⁱ

^a Faculty of Computers and Information, Minia University, Egypt

^b Department of Computer Science, Bahria University, Karachi, Pakistan

^c Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

^d School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia

^e Faculty of Science, Zagazig University, Egypt

^f School of Information Technology, Skyline University College, Sharjah, P.O. Box 1797, United Arab Emirates

^g Department of Computer Science, Government Bikram College of Commerce, Patiala 147001, Punjab, India

^h SINTEF Digital, Oslo, Norway

ⁱ Departamento de Electrónica, Universidad de Guadalajara, CUCEI, Av. Revolución 1500, Guadalajara, Jal, Mexico

ARTICLE INFO

Article history:

Received 11 February 2021

Received in revised form 25 July 2021

Accepted 27 July 2021

Available online 30 July 2021

Keywords:

Meta-heuristics

Marine predators algorithm

Opposition-based learning

Image segmentation

Kapur's entropy and Otsu method

Multilevel thresholding

ABSTRACT

A recent meta-heuristic algorithm called Marine Predators Algorithm (MPA) is enhanced using Opposition-Based Learning (OBL) termed MPA-OBL to improve their search efficiency and convergence. A comprehensive set of experiments are performed to evaluate the MPA-OBL and prove the impact influence of merging OBL strategy with the original MPA in enhancing the quality of the solutions and the acceleration of the convergence speed, using IEEE CEC'2020 benchmark problems as recently complex optimization benchmark. In order to evaluate the performance of the proposed MPA-OBL, the effectiveness of conjunction of OBL with the original MPA and the other counterparts are calculated and compared with LSHADE with semi-parameter adaptation hybrid with CMA-ES (LSHADE_SPACMA-OBL), Restart covariance matrix adaptation ES (CMA_ES-OBL), Differential evolution (DE-OBL), Harris hawk optimization (HHO-OBL), Sine cosine algorithm (SCA-OBL), Salp swarm algorithm (SSA-OBL), and the original MPA. The extensive results and comparisons in terms of optimization metrics have revealed the superiority of the proposed MPA-OBL in solving the IEEE CEC'2020 benchmark problems and improving the convergence speed. Moreover, as a sequel to the proposed MPA-OBL, also, we have conducted experiments using two objective functions of Otsu and Kapur's methods over a variety of benchmark images at different level of thresholds based on three commonly evaluation matrices namely Peak signal-to-noise ratio (PSNR), Structural similarity (SSIM), and Feature similarity (FSIM) indices are analyzed qualitatively and quantitatively. Eventually, the statistical post-hoc analysis reveal that the MPA-OBL obtains highly efficient and reliable results in comparison with the other competitor algorithms.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Image thresholding is an important operation in computer vision for processing and analyzing images in the fields of medical, engineering, agriculture, industry, and so on. It is generally applied in image segmentation to ensure effective feature extraction

for medical and biomedical image processing, robotic vision, and pattern recognition [1]. The purpose is to subdivide an image into several non-overlapping groups of pixels, also known as image segments, based on certain thresholding value(s), in order to isolate objects from within the image [2]. Image thresholding techniques are considered as easy to implement and produce efficient segmentation results [3]. These techniques are categorized into bi-level and multi-level thresholding. In the prior category, a single threshold value is used to separate the image into two homogeneous foreground and background areas, whereas the latter techniques are applied to segment an image into more than two areas, based on pixels intensities known as histogram [4]. When segmenting an image, the selection of thresholding values

* Corresponding author.

E-mail addresses: essam.halim@mu.edu.eg (E.H. Houssein),

kashifhussain.bukc@bahria.edu.pk (K. Hussain), Aligah.2020@gmail.com

(L. Abualigah), abd_el_aziz_m@yahoo.com (M.A. Elaziz),

waleedomoush@yahoo.com (W. Alomoush), gdhiman0001@gmail.com

(G. Dhiman), Youcef.Djenouri@sintef.no (Y. Djenouri), erik.cuevas@cucei.udg.mx

(E. Cuevas).

is critical because of the presence of massive image thresholds, hence it is formulated into an optimization problem to be solved by either parametric or non-parametric techniques [5,6]. This study focuses on parametric approach, in which some statistical parameters are calculated for each class in the image. To obtain the optimal threshold values, taking the advantages of one of the combined methods such as Otsu's (by maximizing between-class variance) [5] and Kaptur's (by maximizing the entropy of the classes) [6] methods. However, in these approaches, achieving the optimal threshold values for multi-level cannot be defined accurately. Hence, multi-level thresholding is considered as a challenge needed to be optimized. To overcome these problems, meta-heuristic techniques are widely utilized in the related literature.

In the last decades, researchers have extensively proved the ability of meta-heuristic algorithms for solving several types of hard optimization problems in the areas of engineering [7–11], communications [12], bioinformatics [13], drug design [14], and feature selection [15], to name a few, mainly due to the fact that such algorithms are general purpose and easy to implement [16–18]. As compared to deterministic methods, meta-heuristic algorithms perform search on a problem landscape with the help a group of search agents which work as candidate solutions generated over and over again in an iterative process using heuristic operators. These operators when applied in different orders, form different search strategies, and these search mechanisms have been inspired from numerous sources from natural and man-made processes. This way, meta-heuristic algorithms are categorized into: natural evolution-based, swarm intelligence-based [19], chemistry and physics-based [20,21]. Some of the most successful meta-heuristic algorithms are genetic algorithm (GA) [22], differential evolution (DE) [23], and particle swarm optimization (PSO) [24], to list a few. Other than these classic methods, some of the recently introduced meta-heuristic techniques have also earned ample appreciation from researcher while investigated in different domains, such as gravitational search algorithm (GSA) [25], Gray wolf optimization (GWO) [26], Arithmetic Optimization Algorithm (AOA) [27], Aquila Optimizer (AO) [28], Harris hawks optimization (HHO) [19], Moth-flame optimization (MFO) [29], Bees Swarm Optimization (BSO) [30]. Marine predators algorithm (MPA) is also one of the most recent additions, introduced in 2020 by Faramarzi et al. [31], which has shown high quality results, as compared to several classical and the latest counterparts.

In literature, the image thresholding clearly shows the efficacy of meta-heuristic algorithms in the relevant paradigm [32–34]. There are numerous examples of meta-heuristic applications in this regard [35,36], however, following are given a few prominent state-of-the-art research works. Upadhyay and Chhabra [37] used crow search algorithm (CSA) for maximizing Kapur's method to tackle the problems of multi-thresholding. The proposed model was compared with a set of meta-heuristic algorithms namely PSO, DE, GWO, MFO and CS. The authors chose CSA because of its balance between exploration and exploitation, as well as, less parameters to tune. Through majority of commonly used evaluation metrics, the authors contended to have achieved comparatively better results while tested on a set of benchmark image using multiple threshold values. Despite success in this work, CSA suffers from slow convergence. In [38], Khairuzzaman and Chaudhury produced efficient image segmentation results with the help of GWO while finding optimal set of thresholds using Otsu's and Kapur's functions. Compared to bacterial foraging optimization (BFO) and PSO, GWO converged to better optimum solutions, however the proposed algorithms also posed certain disadvantages: (a) its efficacy reduced when employed on noisy images and (b) GWO was slower than PSO with regards to computational time. The research maintained a major weakness that

it did not provide comprehensive comparison with many other popular and established meta-heuristic algorithms, merely used PSO and BFO for comparison. In [39], a modified grasshopper optimization algorithm (GOA) with Lévy flight was introduced based on Tsallis cross-entropy as objective function, in order to optimize threshold values for multilevel image thresholding. The proposed model was tested on benchmark images and lant stomata. When compared with standard GOA, WOA, flower pollination algorithm (FPA), PSO, and BA, the proposed GOA variant produced better segmentation accuracy with enhanced convergence on multilevel segmentation on energy-based Tsallis entropy. A shortcoming of this study may be considered as it did not experiment relatively increased thresholds for high-dimensional optimization problems.

The study in [40] found optimal threshold values for grayscale images using EO and Kapur's entropy as objective function. For achieving the enhanced search ability, the researchers improved EO with adaptive parameters, and evaluated using several solution quality metrics such as signal-to-noise ratio, structured similarity index, some accuracy measures like mean absolute error, as well as, computation time for resource complexity. When compared with WOA, BA, SCA, SSA, HHO, CSA and PSO, the proposed EO proved to have outperformed the compared techniques. The significant of this study can be determined with the level of thresholds used in its experiment. The researchers used up to 50 threshold levels. However, the proposed EO variant comparatively underperformed with respect to standard deviation values and computational time. Another recent meta-heuristic technique is HHO which was implemented in [2] in the similar domain using Otsu's and Kapur's objective functions. According to comparisons with PSO, DE, harmony search (HS), ABC, and SCA, its can be asserted that the proposed algorithm produced efficient results in terms of quality, consistency, and accuracy. On the other hand, the results of HHO were also compared with two techniques in machine learning paradigm, K-means and fuzzy IterAg, to discover that these techniques performed the least in the overall image segmentation exercise. On few of the downsides of this study, it was not evaluated on color images and the number of thresholds were set manually. Meanwhile, Díaz-Cortés et al. [41] resolved the problem of unclear regional borders in the low resolution thermography images in health-care with the help of dragonfly algorithm (DA). This practical research applied DA technique to find optimum threshold values for energy curves in thermal images for breast-cancer diagnosis. Based on Otsu's and Kapur's objective functions, the authors evaluated solution quality and found that DA outperformed GA, PSO, runner-root algorithm (RRA), and the krill-herd algorithm (KHA) on a set of 8 images retrieved from the DA-Breast Thermography database.

Most of the meta-heuristic techniques applied on a diversified range of optimization problems in literature often come with certain shortcoming, such as trapping in local regions, early conversion, lacking global search ability. This lays grounds for researchers to propose the modified and hybrid versions. Opposition-based learning (OBL) is one of the most effective approach for improving search efficiency of meta-heuristic algorithms [42]. First introduced by Tizhoosh [43] in 2005, OBL has been integrated with meta-heuristic techniques in several ways for enhancing explorative search ability. For instance, elephant herding optimization (EHO) was enhanced by OBL learning and dynamic cauchy mutation (DCM) in [44] to solve multi-thresholding problem. According to the authors, OBL resolved slow convergence and DCM addressed premature convergence in EHO. The study employed Otsu's and Kapur's methods to estimate threshold values for segmenting an image. When compared with other four popular meta-heuristic algorithms CS, ABC, BA, PSO, and a classical dynamic programming method, results of the

proposed algorithm remained outstanding. Similarly, another related but reasonably old work by Rahnamayan and Tizhoosh [45] also utilized OBL-based enhancement strategy with DE while implemented for gray-level and bi-level images thresholding problem, and found improved results as compared to the original DE. Apart from a couple of example studies related to OBL and meta-heuristic algorithms with respect to image thresholding problems, there is no more research which could be easily found in the relevant literature, to the best of authors' knowledge. However, OBL-based meta-heuristic algorithms are often implemented on various other optimization problems. For this reason, this particular paper aims at further deepening the research in image segmentation area by using the latest marine predators algorithm (MPA); in fact, it happens to be the first instance of MPA applied on the CEC'2020 benchmark functions and image segmentation. Faramarzi et al. [31] introduced MPA in August 2020 with the inspirations of general foraging behavior of ocean predators, which demonstrated superior performance against many other well-known counterparts on various mathematical and engineering benchmark problems. Despite efficacy in search mechanisms in MPA, there are certain areas where it can be further improved to prove its efficiency on different optimization problems like CEC'2020 benchmark functions and a real world problem such as image thresholding problems. Since, the movement of search agents in MPA is based on velocity, the algorithm may miss some important search regions. To address this, we propose to integrate OBL for generating solutions from potential regions in order to explore search space more rigorously. More importantly, we improve its local search ability with the help of solutions generated from around promising region – avoiding trap in local optima. Consequently, the proposed MPA variant comes with trade-off balance between exploration and exploitation. The proposed MPA-OBL is then tested on CEC'2020 test suit functions then it is applied to tackle the multi-level image thresholding problems. Besides, to approve the robustness of the proposed MPA-OBL, the effectiveness of conjunction of OBL with several optimization algorithms namely LSHADE_SPACMA [46], CMA_ES [47], DE [23], HHO [19], SCA [48], and SSA [49] are used for comparison purpose.

To be specific, the contributions of this manuscript can be summarized as follows:

- Adapting MPA based on OBL called MPA-OBL for solving optimization and image segmentation problems.
- The effectiveness of the MPA-OBL is assessed on CEC'2020 test suite.
- MPA-OBL is applied for thresholding segmentation using two objective functions of Otsu and Kapur.
- Verify the quality of the segmentation in terms of the PSNR, SSIM, FSIM.
- Extensive results show the more stable performance of the proposed MPA-OBL.
- The performance of the proposed algorithm is compared with several well-known meta-heuristics.

The rest of this paper is organized as follows: Section 2.3 describes the mathematical model of Otsu and Kapur's methods. Section 2 briefly describes the MPA algorithm and OBL search strategy. Section 3 presents the proposed algorithm. The experimental results are analyzed and discussed in detail in Section 5. In the end in Section 6, the study is duly concluded with intimation of potential future research directions.

2. Preliminaries

This section presents the inspiration, mathematical model, algorithm of the marine predator algorithm (MPA), also describes the opposition-based learning (OBL) strategy.

2.1. Marine predator algorithm

Marine predators algorithm (MPA) [31] is a novel nature-inspired meta-heuristic algorithm. Like other meta-heuristic algorithms, MPA is presented to tackle real-world optimization problems. The inspiration of MPA comes from the widespread foraging behavior in ocean predators and predator-prey encounters or interactions. Here, a predator optimally strategies the encounter rate to maximize the chances of survival in the natural environments. With two basic random walk mechanisms, MPA performs a search with the help of Lévy flight and Brownian motion. The prior type of random walk is often implemented in meta-heuristic algorithms and supposed to be most effective in avoiding solution stagnancy by performing a constructive search in local regions [50]. On the other hand, the latter is an established stochastic apparatus for global search. The inventors of MPA combined the search efficiency of both the random walk strategies to maximize the trade-off balance between exploration and exploitation.

Similar to various other population-based meta-heuristic algorithms, MPA also initializes search process by randomly locating N search agents around search space by Eq. (1):

$$\vec{X}_i = \vec{lb}_i + r \times (\vec{ub}_i - \vec{lb}_i); i \in \{1, 2, \dots, N\} \quad (1)$$

where r denotes a random variable between [0,1], whereas \vec{lb}_i and \vec{ub}_i are two vectors represent the lower and upper bounds for the search to be performed within. During initialization, along the main population matrix, another $N \times D$ matrix comprising of search agents with best fitness values is created, where N and D denote population size and problem dimensions respectively. MPA calls it Elite:

$$Elite = \begin{bmatrix} X_{1,1}^l & X_{1,2}^l & \dots & X_{1,D}^l \\ X_{2,1}^l & X_{2,2}^l & \dots & X_{2,D}^l \\ \vdots & \vdots & \vdots & \vdots \\ X_{N,1}^l & X_{N,2}^l & \dots & X_{N,D}^l \end{bmatrix}_{N \times D} \quad (2)$$

where X^l represents a vector with top fitness.

Another matrix of the same dimension as Elite is called Prey which the predators update their positions depending on it. In a single term, the initialization produces the initial Prey of which the fittest one (predator) builds the Elite . The following is a representation of the Prey:

$$Prey = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,D} \\ X_{2,1} & X_{2,2} & \dots & X_{2,D} \\ X_{3,1} & X_{3,2} & \dots & X_{3,D} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{N,1} & X_{N,2} & \dots & X_{N,D} \end{bmatrix}_{N \times D} \quad (3)$$

Here, X_{ij} denotes the j th dimension of the i th prey. The optimization process is mainly depends on these two matrices.

After initialization, the main iterative search process starts, which is comprised of three phases that mimic different scenarios between predator and the prey, devising different search strategies. These phases are based on iterations $t \in \{1, 2, 3 \dots t_{max}\}$ where t_{max} is maximum iterations. Note that during these phases, MPA updates candidate solutions dimension-wise.

Phase 1 - high velocity ($t < \frac{t_{max}}{3}$). This mimics the scenario when prey is moving faster than the predator. This strategy

enforces exploration and spends more of the initial iterations. The dimension-wise solution update process is carried out using Eq. (4):

$$\begin{aligned} \overrightarrow{\text{Stepsize}}_i &= \vec{R}_B \otimes (\overrightarrow{\text{Elite}}_i - \vec{R}_B \otimes \overrightarrow{\text{Prey}}_i) \quad i = 1, \dots, n \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Prey}}_i + P \cdot \vec{R} \otimes \overrightarrow{\text{Stepsize}}_i \end{aligned} \quad (4)$$

where R_B is a random vector that contains numbers in the range [0,1] based on the normal distribution to represent the Brownian motion. The notation \otimes presents the entry-wise multiplications process. A constant number $P = 0.5$, and a vector of uniform random numbers $R \in [0, 1]$. In this phase, the velocity of predators and prey is high which helps explore far-reached locations in the search space.

Phase 2 - unit velocity ($\frac{1}{3}t_{max} < t < \frac{2}{3}t_{max}$). Both prey and predator move on the same speed. This phase translates transmission stage from exploration to exploitation. In terms of MPA philosophy, this is the situation when prey and predators are traveling in the same pace. Therefore, the population is equally divided into two groups: prey group for exploration using Lévy motion and the predator group for exploitation using Brownian motion. The first half uses Eqs. (5) and (6).

$$\begin{aligned} \overrightarrow{\text{Stepsize}}_i &= \vec{R}_L \otimes (\overrightarrow{\text{Elite}}_i - \vec{R}_L \otimes \overrightarrow{\text{Prey}}_i) \quad i = 1, \dots, n/2 \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Prey}}_i + P \cdot \vec{R} \otimes \overrightarrow{\text{Stepsize}}_i \end{aligned} \quad (5)$$

where \vec{R}_L is a random number based on Lévy distribution. The process of multiplication \vec{R}_L and $\overrightarrow{\text{Prey}}_i$ mimics the predator motion in Lévy, while adding the step size to prey position simulates the movement of prey.

$$\begin{aligned} \overrightarrow{\text{Stepsize}}_i &= \vec{R}_B \otimes (\vec{R}_B \otimes \overrightarrow{\text{Elite}}_i - \overrightarrow{\text{Prey}}_i) \quad i = n/2, \dots, n \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Elite}}_i + P \cdot CF \otimes \overrightarrow{\text{Stepsize}}_i \end{aligned} \quad (6)$$

$CF = \left(1 - \frac{t}{t_{max}}\right)^{\left(2 \cdot \frac{t}{t_{max}}\right)}$ is an adaptive parameter used to control the step size for predators. The process of multiplication \vec{R}_B and $\overrightarrow{\text{Elite}}_i$ mimics the predator motion in Brownian, while prey updates its position based on the Brownian motion.

Phase 3 - low velocity ($t > \frac{2}{3}t_{max}$). predator is moving faster than the prey. This phase enforces exploitation in the last iterations of search process, using Eq. (7):

$$\begin{aligned} \overrightarrow{\text{Stepsize}}_i &= \vec{R}_L \otimes (\vec{R}_L \otimes \overrightarrow{\text{Elite}}_i - \overrightarrow{\text{Prey}}_i) \quad i = 1, \dots, n \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Elite}}_i + P \cdot CF \otimes \overrightarrow{\text{Stepsize}}_i \end{aligned} \quad (7)$$

The process of multiplication \vec{R}_L and $\overrightarrow{\text{Elite}}_i$ simulates the predator movement in Lévy strategy, while prey updates its position based on adding the step size to Elite position.

After a round of iteration, all candidate solutions are evaluated and updated for their personal best fitness, as well as, the best solution is updated. Additionally, in order to inject diversification in candidate solutions, MPA uses a concept in marine predators called eddy formation in which the predators consider longer jumps in different directions in search for more food. For this purpose, MPA uses Eq. (8):

$$\overrightarrow{\text{Prey}}_i = \begin{cases} \overrightarrow{\text{Prey}}_i + CF[\vec{lb} + r \otimes (\vec{ub} - \vec{lb})] \otimes \vec{v} & r \leq FADs \\ \overrightarrow{\text{Prey}}_i + [FADs(1 - r) + r](\overrightarrow{\text{Prey}}_{r1} - \overrightarrow{\text{Prey}}_{r2}) & \text{else} \end{cases} \quad (8)$$

where $\overrightarrow{\text{Prey}}_i$, $\overrightarrow{\text{Prey}}_{r1}$, and $\overrightarrow{\text{Prey}}_{r2}$ are vectors for i th candidate solution, a randomly selected solution, and another randomly selected

candidate solution, respectively; whereas, r denotes a random variable between [0,1], $FADs$ a constant with value 0.2, \vec{v} a binary vector containing 0 and 1 for variation in solution.

2.2. Opposition-based learning

Opposition-based learning (OBL) is a useful search strategy to avoid stagnancy in candidate solutions [51]. Proposed by HR. Tizhoosh [43], OBL enhances the exploitation ability of a search mechanism. Generally in meta-heuristic algorithms, when the initial solutions are closer to the optimal location then convergence happens quickly; otherwise, late convergence is expected. Here, OBL strategy generates new solutions by considering opposite search regions which may prove to be closer to the global optimum.

To better understand the OBL, assume the opposite of a real number $x \in [lb, ub]$ can be calculated as $Opp = (ub + lb) - x$ where Opp is the opposite variable. Hence, for N-dimension real numbers, the previous formulation can be generalized as follows:

$$\overrightarrow{\text{Opp}}_i = (\vec{ub}_i + \vec{lb}_i) - \vec{X}_i \quad (9)$$

Keeping in view the advantages of OBL, researchers in meta-heuristic community have overwhelmingly utilized in many meta-heuristic optimization algorithms. For example, in [52], the OBL strategy is used to enhance the search-efficiency of the HHO algorithm. While the authors in [53], proposed a modified version of the SCA based on OBL, this modification is to jump out from local optima during the search process. In [54], an improved grasshopper optimization algorithm using OBL is presented for solving benchmark functions and engineering problems. Also in [55], OBL is used to improve the shuffled frog-leaping algorithm (SFLA), where OBL is embedded into the memeplexes before the frog initiates foraging. Gray wolf optimizer(GWO) is hybridized with OBL in [56] to help the population avoid stagnancy in solutions, also presents a modified parameter "C" strategy to balance between exploration and exploitation in GWO. In [57], the authors used the OBL to improve the exploration of SSA. However, they used OBL with every iteration in SSA. Therefore, this increases the search complexity of the algorithm, since in each iteration it determines the opposite solutions for all solutions and takes elite solutions. According to this current research work, OBL is used only in the main process of the proposed MPA-OBL to improve the rate of convergence and prevent stuck in local optima of MPA, then MPA-OBL is used to tackle the problem of multi-thresholding for image segmentation by maximizing two objective functions namely Otsu and Kapur. Fig. 1 illustrates the role of OBL in generating new efficient solutions in the hybrid MPA-OBL algorithm.

2.3. Image thresholding methods

This section explains the mathematical model of two common methods used in image thresholding namely Otsu [5] and Kapur [6]. The segmentation process is based on the image histogram [58], which carries the information of pixel's distribution over the image. These two methods take the image histogram as an input, and then to each method's criteria, find the optimal threshold values.

2.3.1. Otsu's method

Otsu [5] is one of the segmentation techniques used to find the optimal threshold values of an image based on maximizing the between-class variance, this section describes the mathematical model of Otsu objective function. This approach considers L intensity levels of a gray image and the probability distribution is

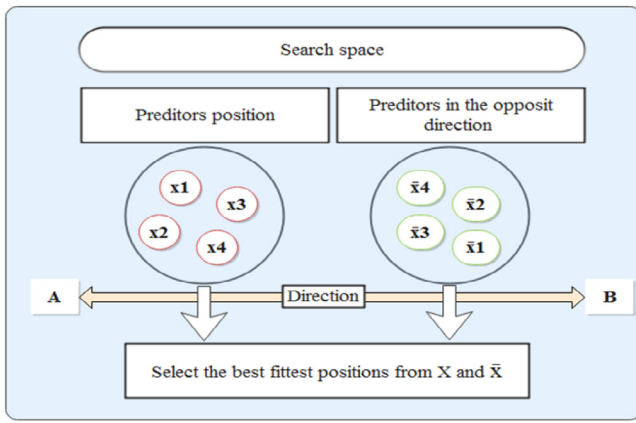


Fig. 1. Illustration of how OBL is used in MPA.

computed as in Eq. (10). This method can be used for color images as RGB where Otsu is applied to each channel separately:

$$h_i = \frac{h_i}{NP}, \sum_{i=1}^{NP} Ph_i = 1 \quad (10)$$

where i is an intensity level defined in the range of $(0 \leq i \leq L - 1)$. NP is the total number of pixels in an image. h_i denotes the number of occurrence of intensity i in the image represented by the histogram. The histogram is normalized in a probability distribution Ph_i . Based on the probability distribution or threshold value (th), the classes are computed for bi-level segmentation as follows:

$$C_1 = \frac{Ph_1}{\omega_0(th)}, \dots, \frac{Ph_{th}}{\omega_0(th)} \text{ and } C_2 = \frac{Ph_{th+1}^c}{\omega_1(th)}, \dots, \frac{Ph_L}{\omega_1(th)} \quad (11)$$

where $\omega_0(th)$ and $\omega_1(th)$ are cumulative probability distributions for C_1 and C_2 , as it is shown by Eq. (12).

$$\omega_0(th) = \sum_{i=1}^{th} Ph_i \text{ and } \omega_1(th) = \sum_{i=th+1}^L Ph_i \quad (12)$$

It is mandatory to find the average intensity levels μ_0 and μ_1 that define the classes using Eq. (13). Once those values are calculated, the Otsu based between-class σ_B^2 is calculated using Eq. (14).

$$\mu_0 = \sum_{i=1}^{th} \frac{iPh_i}{\omega_0(th)} \text{ and } \mu_1 = \sum_{i=th+1}^L \frac{iPh_i}{\omega_1(th)} \quad (13)$$

$$\sigma_B^2 = \sigma_1 + \sigma_2 \quad (14)$$

Notice that σ_1 and σ_2 in Eq. (14) are the variances of C_1 and C_2 which are defined as follow:

$$\sigma_1 = \omega_0(\mu_0 + \mu_T)^2 \text{ and } \sigma_2 = \omega_1(\mu_1 + \mu_T)^2 \quad (15)$$

where $\mu_T = \omega_0\mu_0 + \omega_1\mu_1$ and $\omega_0 + \omega_1 = 1$ based on the values σ_1 and σ_2 , Eq. (16) presents the objective function. Therefore, the optimization problem is reduced to find the intensity level that maximizes Eq. (16)

$$F_{otsu}(th) = \max(\sigma_B^2(th)) \text{ where } 0 \leq th \leq L - 1 \quad (16)$$

where $\sigma_B^2(th)$ is the Otsu's variance for a given th value. Otsu's method is applied for a single component of an image, that means for RGB images it is necessary to apply separation into single component images. The previous illustration of such bi-level method can be modified for multiple thresholds. The objective function $F_{otsu}(th)$ in Eq. (16) can also be modified for multiple

thresholds as follows:

$$F_{otsu}(TH) = \max(\sigma_B^2(th)) \text{ where } 0 \leq th \leq L - 1 \text{ and } i = [1, 2, 3, \dots, k] \quad (17)$$

where $TH = [th_1, th_2, \dots, th_k - 1]$ is a vector containing multiple thresholds, L denotes maximum gray level, whereas the variances are computed through Eq. (18)

$$N\sigma_B^2 = \sum_{i=1}^k \sigma_i = \sum_{i=1}^k \omega_i(\mu_i - \mu_T)^2 \quad (18)$$

where i represents a specific class. ω_i and μ_j are the probability of occurrence and the mean of a class respectively. For multi-level thresholding, such values are obtained as:

$$\omega_{k-1}(th) = \sum_{i=th_k+1}^L Ph_i \quad (19)$$

for mean values:

$$\mu_{k-1} = \sum_{i=th_k+1}^L \frac{iPh_i}{\omega_1(th_k)} \quad (20)$$

2.3.2. Kapur's method

Another thresholding technique used to apply the concept of segmentation is the Kapur's method [6]. Kapur's method selects the optimal threshold values based on maximizing the entropy. The mathematical model is described as follows:

$$F_{kapur}(th) = H_1 + H_2 \quad (21)$$

where the entropies H_1 and H_2 are computed as:

$$H_1 = \sum_{i=1}^{th} \frac{Ph_i}{\omega_0} \ln\left(\frac{Ph_i}{\omega_0}\right) \text{ and } H_2 = \sum_{i=th+1}^L \frac{Ph_i}{\omega_1} \ln\left(\frac{Ph_i}{\omega_1}\right) \quad (22)$$

where Ph_i is the probability distribution of the intensity levels which is obtained using Eq. (10), $\omega_0(th)$ and $\omega_1(th)$ are probabilities distributions for the classes C_1 and C_2 . $\ln(\cdot)$ is the natural logarithm. Similar to the Otsu's method, the entropy-based approach can be modified for multi-thresholding values; for such a case, it is necessary to divide the image into k classes using $k - 1$ thresholds. The objective function then can be modified as follows:

$$F_{kapur}(TH) = \sum_{i=1}^k H_i \quad (23)$$

where $TH = [th_1, th_2, \dots, th_{k-1}]$ is a vector that contains the multiple thresholds. Each entropy is computed separately with its respective (th) value, so Eq. (23) is expanded for k entropies as:

$$H_k^c = \sum_{i=th_{k+1}}^L \frac{Ph_i}{\omega_{k-1}} \ln\left(\frac{Ph_i}{\omega_{k-1}}\right) \quad (24)$$

Here, the values of the probability occurrence ($\omega_0^c, \omega_1, \dots, \omega_{k-1}$) of the k classes are obtained using Eq. (19) and the probability distribution Ph_i with Eq. (10).

3. The proposed MPA-OBL

In this section, the proposed algorithm MPA-OBL is explained in detail. OBL is a local search strategy aims to avoid the drawbacks of the random population and increases the convergence of the algorithm by improving the diversity of its solutions. As a result, the smaller steps assist the whole search to scan the

promising region rigorously. The OBL strategy can be applied to the MPA to enhance the search process as following:

$$\vec{Opp}_i = (\vec{ub}_i + \vec{lb}_i) - \vec{X}_i \quad (25)$$

where \vec{Opp}_i is a vector maintaining solution generated by applying OBL. Following is given further detail on the phases of the proposed model for image thresholding.

3.1. Primitive steps in MPA-OBL

The first step of the proposed model is to read the image from the benchmark dataset used in gray-scale, then it obtains the histogram of the selected image and calculates the probability distribution using Eq. (10), the next step is to initialize the MPA-OBL parameters such as maximum iterations t_{max} , population size N , FADs, constant value P , and problem dimensions D . Like many other meta-heuristic algorithms, MPA-OBL starts with the random initialization of the first population x^0 and saving results.

3.2. Optimization scenarios

After the concept of the OBL strategy is applied to calculate the \vec{Opp}_i vector using Eq. (25), this phase starts by evaluating the \vec{X}_i and \vec{Opp}_i populations using Otsu F_{otsu} Eq. (17) or Kapur F_{kapur} Eq. (23) objective function then comparing the fitness of \vec{X}_i and \vec{Opp}_i and saving the global best solution with the highest fitness. The optimization process is divided into three main phases of optimization as discussed in Section 2.1. After applying the three phases of the optimization, the proposed algorithm calculates and compares the fitness of \vec{X}_i and \vec{Opp}_i after using Otsu Eq. (17) or Kapur Eq. (23) and updates the global best solution found so far.

3.3. Final steps of MPA-OBL

Now, after accomplishing the optimization scenarios, memory saving, Elite/Fitness update, and the FADs effect using Eq. (8) over the iterations. the proposed algorithm selects the best solution according to the best threshold values onto the image. Algorithm 1 illustrates the pseudo-code of the proposed MPA-OBL algorithm.

3.4. Computational complexity of the proposed MPA-OBL

This section explains the time and space costs maintained by the proposed method.

3.4.1. Time complexity

Firstly, the MPA-OBL produces N number of search agents each with size D , so the complexity of initialization can be represented as $\mathcal{O}(N \times D)$ time complexity. Besides, the MPA-OBL calculates the fitness of each search agent with complexity of $\mathcal{O}(t_{max} \times N \times D)$, where t_{max} indicates the cumulative number of iterations. In addition, MPA-OBL needs $\mathcal{O}(T)$ time complexity to perform T number of its main operations (phase1, phase2, phase3, memory saving, FADs effect, and the OBL). Hence, the total time complexity of the proposed MPA-OBL can be represented by $\mathcal{O}(t_{max} \times T \times N \times D)$.

3.4.2. Space complexity

Space complexity determines the total amount of space occupied by the algorithm. Now, MPA-OBL uses the space complexity of $\mathcal{O}(N \times D)$.

Algorithm 1 The proposed MPA-OBL algorithm

Set parameters values $t_{max} = 350, N = 50, P = 0.5, FADs = 0.2$;
Randomly initialize the population x_i^0 with dimensions D .
while $t \leq t_{max}$ **do**
 for $i \leq N$ **do**
 Evaluate \vec{X}_i using Otsu Eq. (17) or Kapur Eq. (23) and store results in fit_i .
 Apply OBL on the population \vec{X}_i using Eq. (25) and save result in \vec{Opp}_i .
 Evaluate \vec{Opp}_i using Otsu Eq. (17) or Kapur Eq. (23) and store results in $FitOpp_i$.
 if $Fit_i < FitOpp_i$ **then**
 $\vec{X}_i = \vec{Opp}_i$;
 end if
 end for
 Accomplish memory saving.
 if $t < \frac{t_{max}}{3}$ **then**
 Update candidate solutions using Eq. (4).
 else if $\frac{t_{max}}{3} < t < 2 \times \frac{t_{max}}{3}$ **then**
 Update first half of the solutions using Eq. (5) and another half using Eq. (6).
 else if $t > 2 \times \frac{t_{max}}{3}$ **then**
 Update candidate solutions using Eq. (7).
 end if
 Update best solution found so far.
 Apply FADs effect and update based on Eq. (8).
 for each candidate solution **do**
 Evaluate \vec{X}_i after update using Otsu Eq. (17) or Kapur Eq. (23) and store results in Fit_i .
 Evaluate \vec{Opp}_i after update using Otsu Eq. (17) or Kapur Eq. (23) and store results in $FitOpp_i$.
 if $Fit_i < FitOpp_i$ **then**
 Set $\vec{X}_i = \vec{Opp}_i$;
 end if
 end for
 Accomplish memory saving and Elite/Fitness update.
 Apply FADs effect using Eq. (8).
 Save the results.
end while
Return the best solution which contains the best threshold values onto the image.

4. Performance evaluation of MPA-OBL

4.1. Definition of CEC'2020 benchmark functions

This section introduces the evaluation process of the new version of the MPA called MPA-OBL. Consequently, The IEEE Congress on Evolutionary Computation (CEC) [59] were chosen as test problems to measure the performance of the proposed algorithms. Initially, the CEC'2020 benchmark functions contained 10 test functions, including unimodal, multimodal, hybrid, and composition functions. Table 1 illustrates the mathematical formulation and properties of the CEC'2020 benchmark test; 'Fi*' denotes the optimum global value.

Fig. 2 provides a 2D visualization of the CEC'2020 functions to ease the understanding of differences and the nature of each problems.

4.2. Statistical results on CEC'2020 benchmark functions

As mentioned above, CEC'2020 benchmark functions are used to evaluate the performance of the proposed MPA-OBL including quantitatively and qualitatively metrics. Quantitative metrics

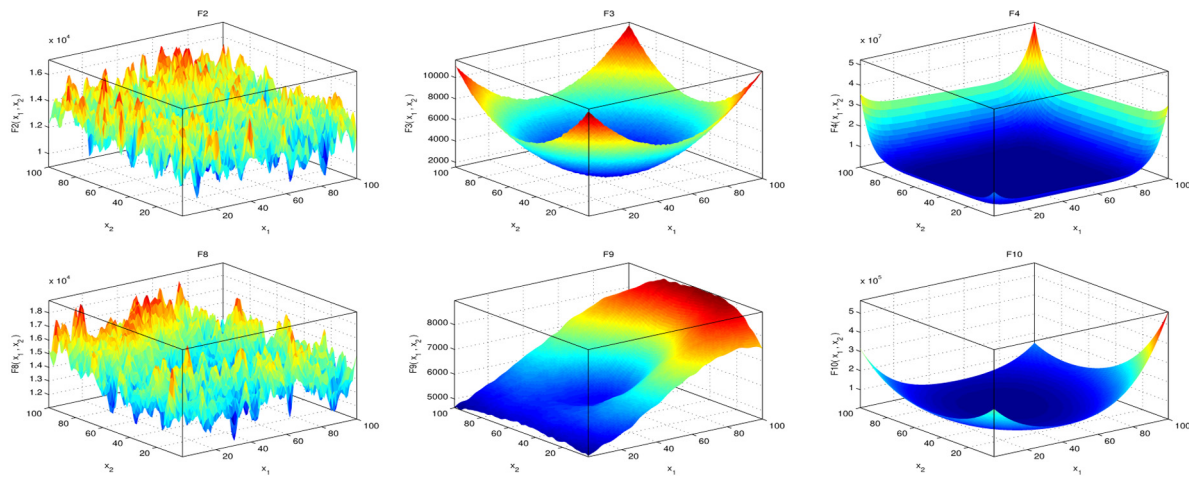


Fig. 2. View of some CEC'2020 benchmark functions.

Table 1
CEC 2020 benchmark test.

No.	Function description	Fi*
Unimodal function		
F1	Shifted and Rotated Bent Cigar function	100
Multimodal shifted and rotated functions		
F2	Shifted and Rotated Schwefel's function	1100
F3	Shifted and Rotated Lunacek bi-Rastrigin function	700
F4	Expanded Rosenbrock's plus Griewangk's function	1900
Hybrid functions		
F5	Hybrid function 1 (N = 3)	1700
F6	Hybrid function 2 (N = 4)	1600
F7	Hybrid function 3 (N = 5)	2100
Composition functions		
F8	Composition function 1 (N = 3)	2200
F9	Composition function 2 (N = 4)	2400
F10	Composition function 3 (N = 5)	2500

are the mean and standard deviation (STD) for best solutions obtained by the proposed algorithm and all other algorithms used in the comparison. Moreover, the qualitative metrics include search history, average fitness history, and the convergence curve. For fair evaluation, the proposed MPA-OBL results were compared with other seven algorithms namely LSHADE_SPACMA-OBL, CMA_ES-OBL, DE-OBL, HHO-OBL, SCA-OBL, SSA-OBL, and the original MPA. The termination criteria for all considered optimization algorithms is satisfied while reaching to the number of 30000 function evaluations. Table 4 presents the parameters settings for each algorithm. Table 3 shows the mean of time in seconds taken by each algorithm over 30 run. Table 2 provides the mean and STD of the best value obtained from the proposed algorithm and all other competed algorithms for each CEC'2020 benchmark function with 30-dimension, the best results (minimum values) is highlighted in bold. Results in terms of mean and STD reveal the superiority of the proposed algorithm in solving eight of CEC'2020 benchmark functions compared to other competed algorithms. Moreover, MPA-OBL gains the first rank in terms of Friedman mean rank sum test.

Fig. 3 summarizes the better statistical results of tackling the CEC'2020 test suit experiment in terms of worst, best, mean, and STD. As shown, the proposed MPA-OBL reaches the mean and best optimal solutions for most of the test functions, as compared to other methods. Also, the standard deviation (STD) of solutions found with MPA-OBL and others suggest that the proposed method produced almost similar kind of solutions everytime.

In case of HHO-OBL, the bar for worst solutions is high, which means this algorithms generated poor solutions for most of the functions used to evaluate the performance of these algorithms. Overall, Fig. 3 indicates superiority over counterpart methods when taking the test functions collectively.

4.3. Boxplot behavior analysis

The data distribution characteristics can be displayed by the boxplot analysis. Since this class of functions are associated with too many local minima, in order to have a better understanding of distribution of results, the boxplot of results for each algorithm and function are shown in Fig. 4. Boxplots are great plots to depict data distributions into quartiles. The minimum and maximum are the lowest and largest data points reached by the algorithm, which are the edges of the whiskers. The lower quartile and upper quartile are delimited by the ends of the rectangles. A narrow boxplot signifies a high agreement between data. Fig. 4 shows the results of ten functions boxplot for Dim = 30. The boxplots of the proposed MPA-OBL algorithm are, for most functions, very narrow compared to other algorithm distribution, and so, with the lowest values. Indeed, the proposed MPA-OBL algorithm performs better than the other algorithms on most of the test functions, but yields to limited performance only on F1, and F10.

4.4. Convergence behavior analysis

This subsection presents a convergence analysis of the proposed MPA-OBL compared to the other competitor algorithms. Fig. 5 presents the convergence curves of LSHADE_SPACMA-OBL, CMA_ES-OBL, DE-OBL, HHO-OBL, SCA-OBL, SSA-OBL, and the original MPA for the CEC'2020 functions. The proposed algorithm reaches a stable point for all functions. This behavior suggests that the proposed algorithm converges. Moreover, the proposed algorithm achieves the lowest average of the best so-far solutions the fastest for most functions. This fast convergence to the (near)-optimal solution is noticed, and makes the proposed MPA-OBL algorithm a promising optimization algorithm to solve problems that require fast computation, such as online optimization problems.

4.5. Qualitative metrics analysis

Even though, the previous result analyses confirm the high performance of the proposed MPA-OBL algorithm, performing more experiments and analyses would allow us to draw stronger

Table 2

The mean, and STD of fitness values for 30 runs obtained with the different algorithms on the CEC'2020 functions with $Dim = 30$.

Functions	Measure	LSHADE_SPACMA-OBL	CMA_ES-OBL	DE-OBL	HHO-OBL	SCA-OBL	SSA-OBL	MPA	MPA-OBL
F1	BEST	1.56E+09	1.01E+02	1.05E+09	1.34E+10	1.01E+02	1.61E+07	5.37E+03	1.71E+02
	WORST	1.05E+10	8.33E+04	2.80E+09	3.37E+10	2.08E+04	4.45E+07	2.76E+05	3.49E+04
	MEAN	4.56E+09	1.75E+04	1.61E+09	1.87E+10	3.31E+03	3.10E+07	7.85E+04	6.63E+03
	STD	2.00E+09	2.28E+04	4.15E+08	3.90E+09	5.19E+03	6.68E+06	6.79E+04	7.44E+03
F2	BEST	6.83E+03	8.03E+03	7.57E+03	7.84E+03	3.89E+03	4.12E+03	3.43E+03	2.99E+03
	WORST	9.80E+03	9.40E+03	8.97E+03	9.37E+03	6.53E+03	7.29E+03	5.75E+03	4.65E+03
	MEAN	8.66E+03	8.96E+03	8.49E+03	8.76E+03	5.24E+03	5.59E+03	4.80E+03	3.89E+03
	STD	8.98E+02	3.42E+02	3.12E+02	3.30E+02	6.66E+02	8.38E+02	6.98E+02	4.85E+02
F3	BEST	1.01E+03	7.47E+02	1.05E+03	1.13E+03	8.41E+02	1.13E+03	9.46E+02	7.55E+02
	WORST	1.21E+03	9.21E+02	1.20E+03	1.32E+03	1.10E+03	1.42E+03	1.36E+03	8.45E+02
	MEAN	1.09E+03	8.65E+02	1.10E+03	1.22E+03	9.18E+02	1.28E+03	1.14E+03	8.04E+02
	STD	5.11E+01	6.75E+01	3.82E+01	4.77E+01	5.87E+01	7.92E+01	9.81E+01	2.20E+01
F4	BEST	1.94E+03	1.90E+03	2.72E+03	1.90E+03	1.90E+03	1.90E+03	1.90E+03	1.90E+03
	WORST	2.81E+03	1.91E+03	1.26E+04	1.94E+03	1.93E+03	1.90E+03	1.92E+03	1.90E+03
	MEAN	2.06E+03	1.90E+03	6.59E+03	1.91E+03	1.91E+03	1.90E+03	1.91E+03	1.90E+03
	STD	1.72E+02	9.71E-01	2.73E+03	8.68E+00	5.29E+00	0.00E+00	5.34E+00	3.81E-09
F5	BEST	5.14E+05	7.56E+05	4.44E+06	1.74E+06	3.18E+04	1.00E+06	3.69E+04	7.62E+04
	WORST	2.53E+07	1.04E+07	2.13E+07	3.95E+07	2.99E+06	1.13E+07	7.03E+06	3.18E+05
	MEAN	5.76E+06	4.18E+06	1.18E+07	1.22E+07	1.19E+06	4.56E+06	2.02E+06	1.69E+05
	STD	5.20E+06	2.29E+06	3.69E+06	7.25E+06	6.90E+05	2.65E+06	1.54E+06	5.97E+04
F6	BEST	2.48E+03	1.81E+03	3.04E+03	3.25E+03	2.06E+03	2.36E+03	2.09E+03	1.71E+03
	WORST	4.14E+03	3.27E+03	4.16E+03	4.39E+03	3.35E+03	3.91E+03	3.20E+03	2.27E+03
	MEAN	3.57E+03	2.63E+03	3.70E+03	3.80E+03	2.64E+03	2.88E+03	2.61E+03	1.87E+03
	STD	4.38E+02	4.35E+02	2.23E+02	2.53E+02	3.33E+02	4.21E+02	2.74E+02	1.18E+02
F7	BEST	1.24E+05	7.55E+05	3.62E+05	9.36E+05	3.54E+04	1.05E+05	5.61E+04	2.11E+04
	WORST	8.11E+06	6.92E+06	4.09E+06	1.43E+07	1.52E+06	4.62E+06	1.68E+06	2.79E+05
	MEAN	2.07E+06	2.83E+06	1.95E+06	3.48E+06	4.52E+05	8.54E+05	3.95E+05	1.05E+05
	STD	2.30E+06	1.76E+06	8.87E+05	2.54E+06	4.03E+05	9.04E+05	4.21E+05	6.02E+04
F8	BEST	2.62E+03	8.97E+03	8.66E+03	4.05E+03	2.30E+03	2.33E+03	2.30E+03	2.30E+03
	WORST	1.00E+04	1.07E+04	1.05E+04	1.08E+04	8.52E+03	9.06E+03	2.60E+03	2.31E+03
	MEAN	4.74E+03	1.01E+04	9.69E+03	8.87E+03	5.47E+03	6.77E+03	2.33E+03	2.30E+03
	STD	2.33E+03	4.24E+02	4.49E+02	2.45E+03	2.23E+03	1.89E+03	6.44E+01	2.10E+00
F9	BEST	3.03E+03	2.83E+03	3.03E+03	3.17E+03	2.88E+03	3.25E+03	2.88E+03	2.87E+03
	WORST	3.15E+03	3.00E+03	3.07E+03	3.31E+03	3.00E+03	3.77E+03	3.26E+03	2.96E+03
	MEAN	3.08E+03	2.85E+03	3.06E+03	3.23E+03	2.94E+03	3.53E+03	3.07E+03	2.90E+03
	STD	2.76E+01	2.95E+01	1.10E+01	3.47E+01	3.21E+01	1.24E+02	1.00E+02	2.18E+01
F10	BEST	3.04E+03	2.88E+03	3.04E+03	3.23E+03	2.89E+03	2.90E+03	2.89E+03	2.88E+03
	WORST	3.47E+03	2.88E+03	3.21E+03	3.77E+03	3.00E+03	3.00E+03	3.04E+03	2.93E+03
	MEAN	3.18E+03	2.88E+03	3.11E+03	3.44E+03	2.92E+03	2.94E+03	2.94E+03	2.89E+03
	STD	1.03E+02	1.13E-01	5.10E+01	1.64E+02	2.77E+01	2.88E+01	3.80E+01	9.77E+00
Fridman mean rank	6.05	4.00	5.54	6.85	3.21	5.04	3.79	1.53	
Rank	7	4	6	8	2	5	3	1	

Table 3

The mean of time values for 30 runs obtained with the different algorithms on the CEC'2020 functions with $Dim= 30$.

Functions	LSHADE_SPACMA-OBL	CMA_ES-OBL	DE-OBL	HHO-OBL	SCA-OBL	SSA-OBL	MPA	MPA-OBL
F1	7.90E-01	2.35E+00	8.88E-01	7.15E-01	1.75E+00	7.48E-01	1.21E+00	2.61E+00
F2	7.86E-01	1.84E+00	8.48E-01	7.24E-01	1.74E+00	7.19E-01	1.21E+00	2.60E+00
F3	7.83E-01	1.82E+00	8.49E-01	7.07E-01	1.74E+00	7.13E-01	1.20E+00	2.61E+00
F4	7.83E-01	1.81E+00	8.52E-01	7.10E-01	1.74E+00	7.18E-01	1.20E+00	2.59E+00
F5	7.96E-01	1.84E+00	8.52E-01	7.13E-01	1.75E+00	7.15E-01	1.20E+00	2.59E+00
F6	7.85E-01	1.92E+00	8.42E-01	7.12E-01	1.75E+00	7.12E-01	1.19E+00	2.59E+00
F7	7.92E-01	1.79E+00	8.47E-01	7.18E-01	1.74E+00	7.12E-01	1.19E+00	2.59E+00
F8	7.94E-01	1.84E+00	8.39E-01	7.22E-01	1.77E+00	7.24E-01	1.19E+00	2.59E+00
F9	7.91E-01	1.82E+00	8.55E-01	7.11E-01	1.75E+00	7.16E-01	1.20E+00	2.60E+00

conclusions regarding the algorithm performance for real problem-solving. For instance, monitoring the behavior of the particles, i.e., search agents, provides more insights into the optimization search process and algorithm convergence. The qualitative analysis of the proposed MPA-OBL algorithm are illustrated in Fig. 6. Notably, the agent's behaviors are displayed in Fig. 6, which includes 2D views of the functions, search history, average fitness history, and convergence curve.

The following points are observed from this the qualitative analysis:

- *In terms of domain's topology – functions in 2D views :* The first column of Fig. 6 illustrates the function in 2-dimensional space. The functions possess distinctive topology, which provides insight into determining the type/shape of functions the algorithm yields the best performance.
- *Regarding the search history:* The second column of Fig. 6 displays the search history of agents, from the first to the last iteration. The search space is depicted with counter lines, for which the gradation from the blue lines to red lines means an increased fitness value. Search history shows that

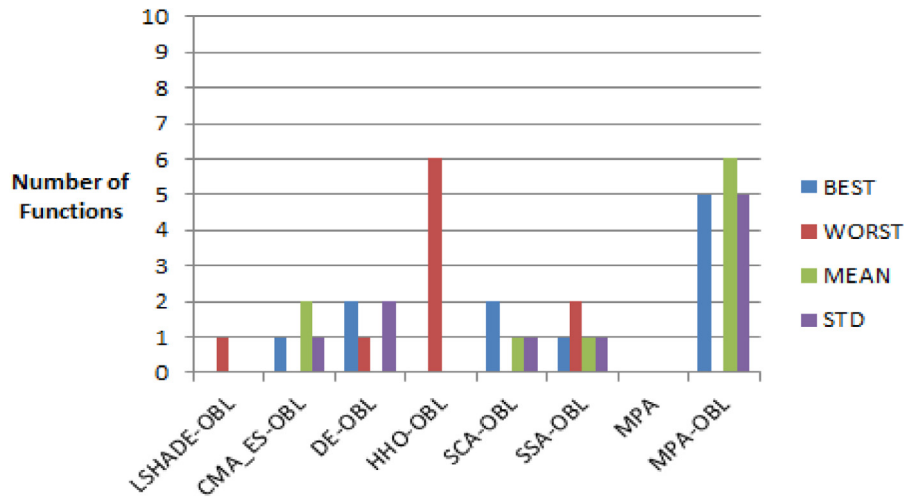


Fig. 3. Summary of better values for the Best, Worst, Mean, and STD over the CEC'2020 test suit functions obtained by each algorithm.

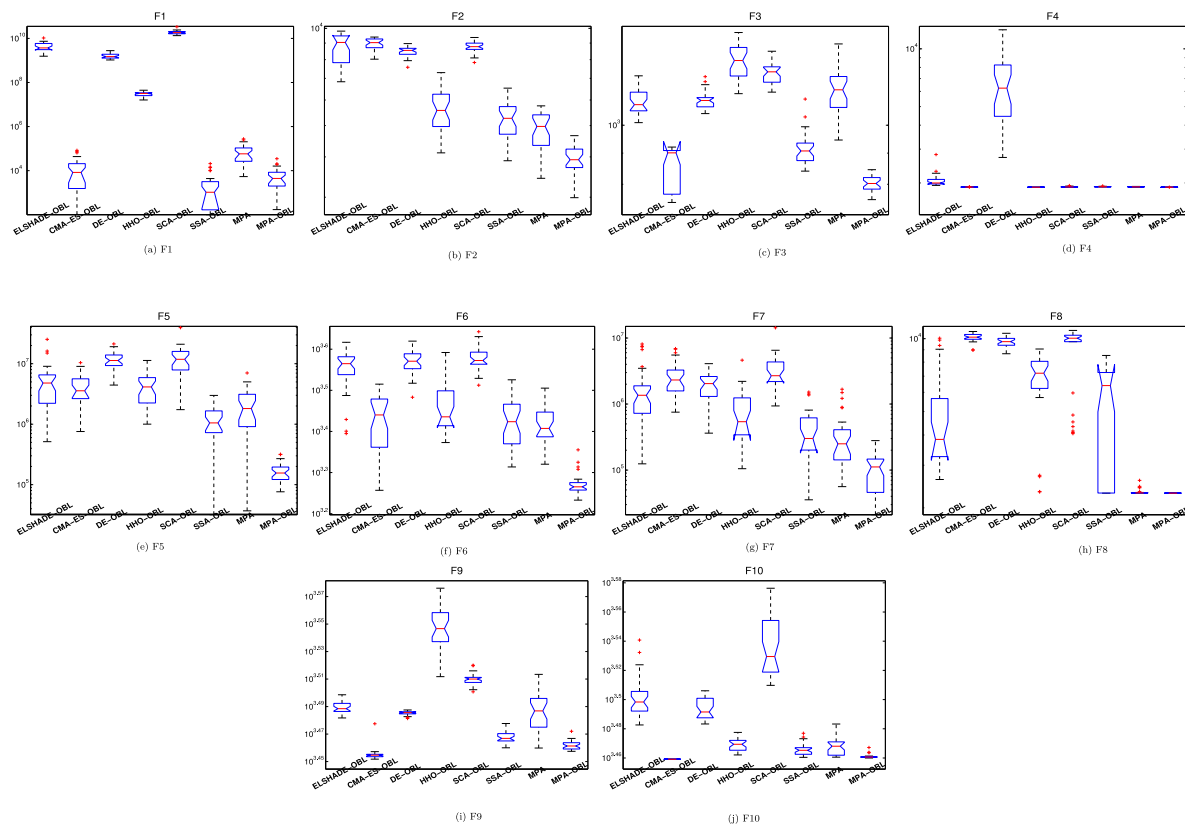


Fig. 4. Boxplots of the results obtained the algorithms over CEC'2020 functions with Dim = 30.

for some functions the proposed MPA-OBL is able to find the areas where the fitness values are the lowest.

- *In terms of average fitness history:* The third column of Fig. 6 presents the average fitness history, i.e., the fitness value averages as a function of the iteration number. This average provides insight into the general behavior of the agents and the role they play in the optimization process. The history curves are all decreasing, which means that the population improves at each iteration. This constant improvement substantiates a collaborative searching behavior and supports the efficacy of the updating particle law.

- *In terms of convergence curve and optimization history :* The fourth column reveals the progress of fitness over a number of iterations. When optimization history is decreasing, it indicates that the solutions are optimized during iterations until reaching the optimal solution.

5. Experimental results for multilevel thresholding

This section introduces the experimental environment for the proposed algorithm MPA-OBL. The benchmark images are introduced first followed by the experimental settings used in this research work. Moreover, the analysis of the results obtained by the proposed algorithm in terms of fitness, PSNR, FSIM, and

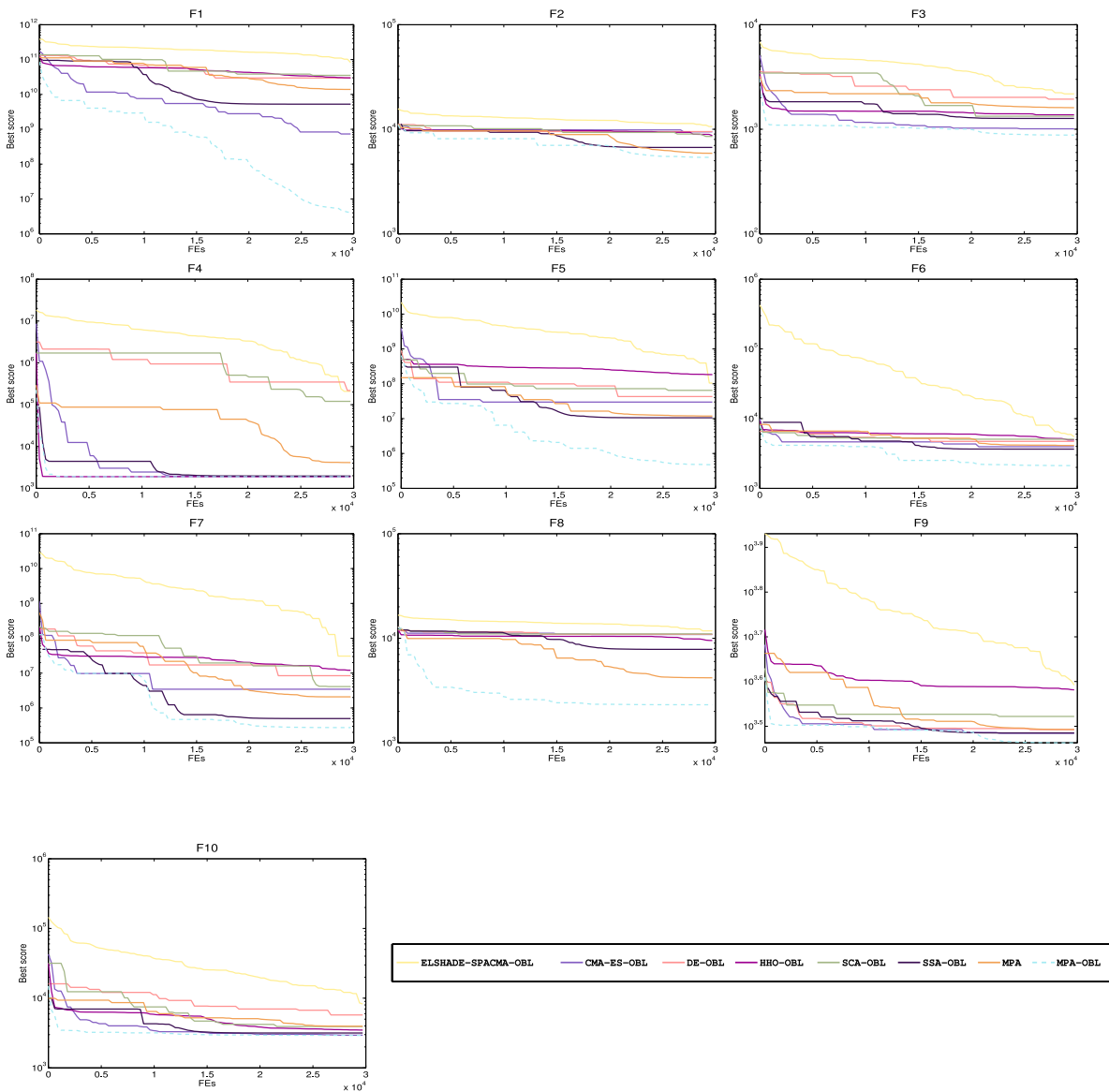


Fig. 5. Curves of convergence of the algorithms obtained on different CEC'2020 functions with $Dim = 30$.

SSIM. The statistical analysis used for comparing the proposed algorithm with other competed algorithms is also demonstrated here. This section is organized as follows: Section 5.2 shows the benchmark images used in the experiments. Section 5.3 describes the experimental settings. Section 5.4 illustrates the evaluation metrics used in the comparison. Section 5.5 shows Otsu's results in terms of fitness, PSNR, SSIM, FSIM, time and Wilcoxon test. Section 5.6 presents Kapur's results in terms of fitness, PSNR, SSIM, FSIM, time, and Wilcoxon test.

5.1. Parameter settings

For fair comparison, the algorithms are tested over 30 independent runs, on two runs of experiments. The maximum number of iterations is 1,000 for each benchmark function. Table 4 presents the parameters setting of each algorithm. Generally, setting algorithm parameters to their default values is a fair and appropriate practice, which is the reason behind the parametrization employed in this study. Moreover, using the default values lessen the risk of bias in the comparison since no algorithm could be advantaged with a better parametrization. This study employs

qualitative metrics and quantitative metrics to measure algorithm performances.

5.2. Benchmark images

The proposed MPA-OBL along with other meta-heuristic algorithms are evaluated on ten benchmark gray-scale images. The test images are Cameraman, Lena, Baboon, . . . , and others presented in Table 5. The purpose behind choosing these images is just for testing the performance and the quality of the segmented images.

5.3. Experimental setup

The proposed algorithm MPA-OBL is compared with seven meta-heuristic algorithms which are LSHADE_SPACMA-OBL, CMA_ES-OBL, DE-OBL, HHO-OBL, SCA-OBL, SSA-OBL, and the original MPA. To have a fair comparison, all algorithms are executed 30 runs per image, each run with 350 internal iterations, and the number of the population is set to 30. The experiments were

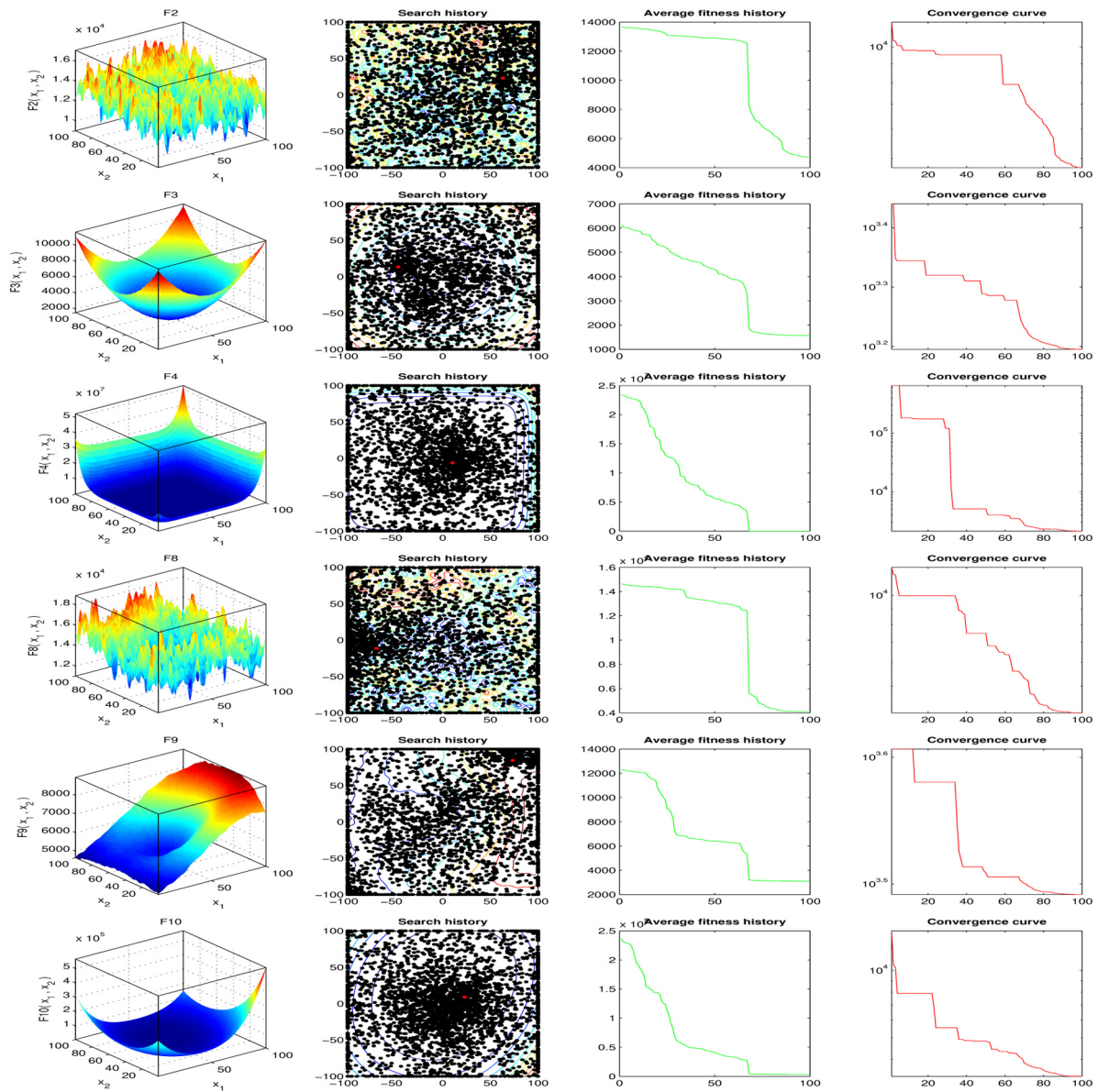


Fig. 6. Qualitative metrics on F2, F3, F4, F8, F9, and F10: 2D views of the functions, search history, average fitness history, and convergence curve.

Table 4

Parameter settings of MPA-OBL and all other competed algorithms.

Algorithms	Parameters setting
Common settings	Population size: $N = 30$ Maximum iterations: $t_{max} = 1000$ Problem dimensions $Dim = 30$ Number of independent runs 30
LSHADE_SPACMA-OBL	Population-size = 30 Pbest = 0.1, Arc rate = 2
CMA_ES-OBL	$\alpha_{mu} = 2$
DE-OBL	Scaling Factor =[0.2 0.8]; Crossover Probability = 0.2
HHO-OBL	beta = 1.5
SCA-OBL	$A = 2$
SSA-OBL	$c1=0.5$
MPA and MPA-OBL	$FADs = 0.2, P = 0.5$

Table 5
The set of benchmark images and relative histograms.

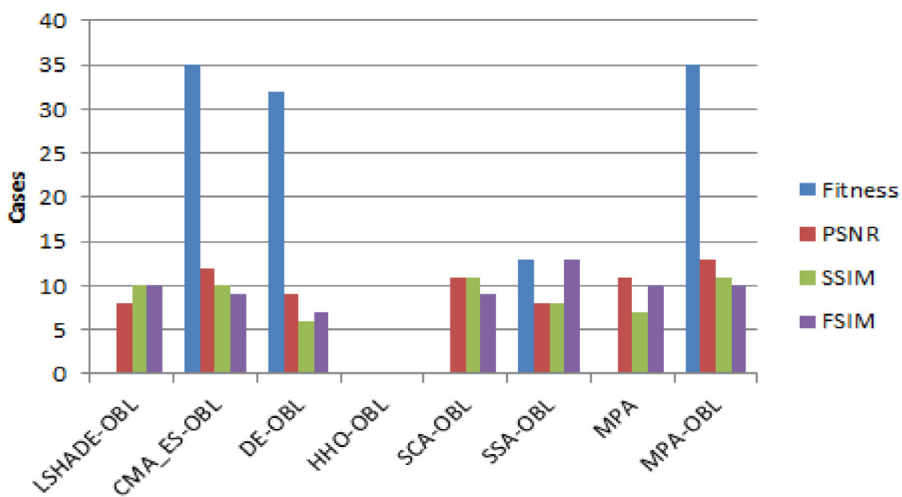
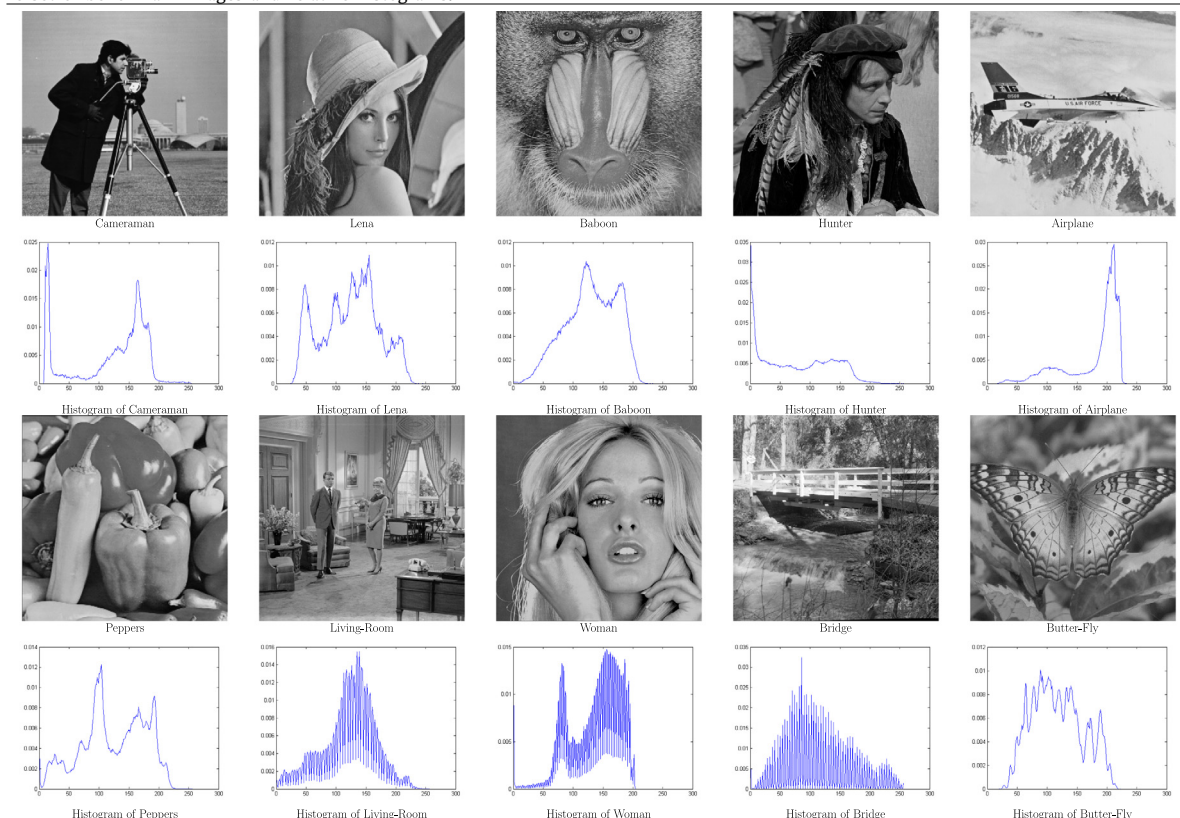




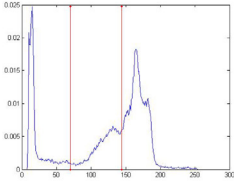
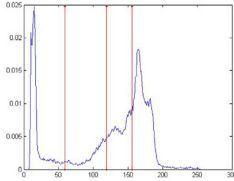
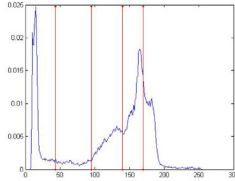
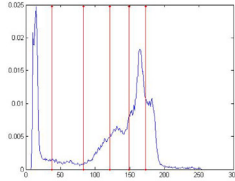




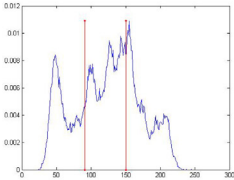
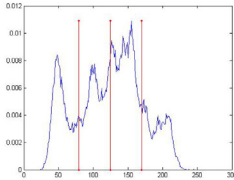
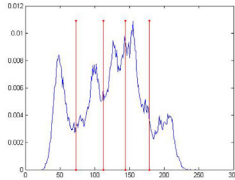
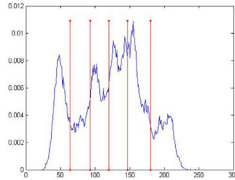
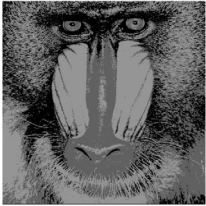
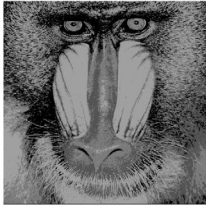
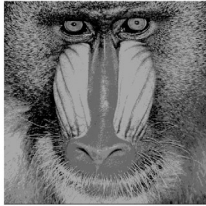
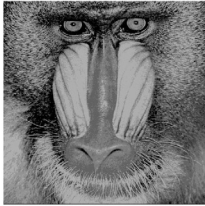
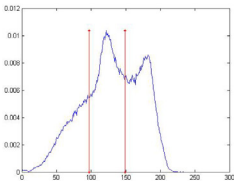
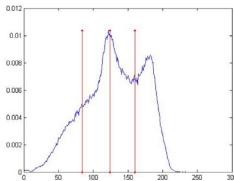
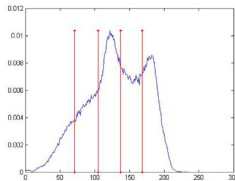
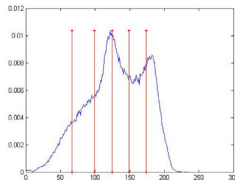


Fig. 7. Summary of Fitness, PSNR, SSIM, and FSIM number of best cases for all thresholds obtained by each algorithm based on Otsu's objective function.

Table 6
Results after applying MOA-OBL on Otsu's method to the set of benchmark images.

Image	nTh = 2	nTh = 3	nTh = 4	nTh = 5
Cameraman				
				
Lena				
				
Baboon				
				

performed on test images with different numbers of thresholds [nTh = 2, 3, 4, 5].

5.4. Evaluation measurements

To evaluate the performance and quality of the segmented images at different numbers of thresholds, three metrics namely peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and feature similarity (FSIM) are used.

PSNR. is a valid quality measure used to find the difference between the original image and segmented one quality [60]. PSNR





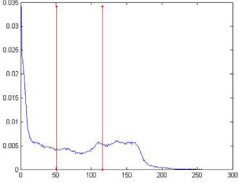
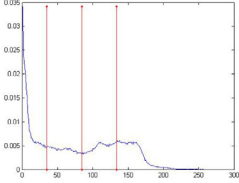
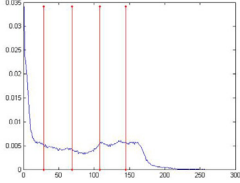
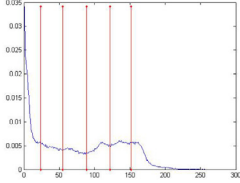




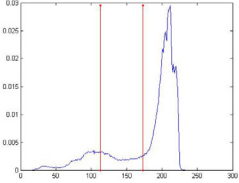
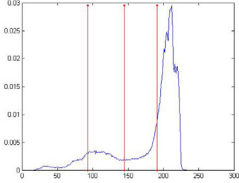
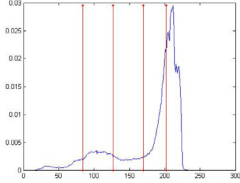
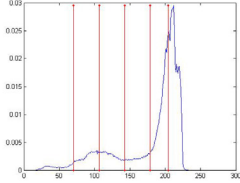




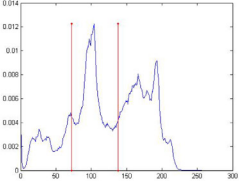
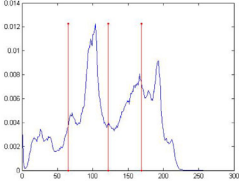
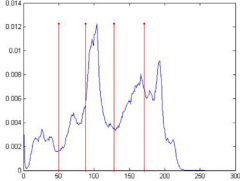
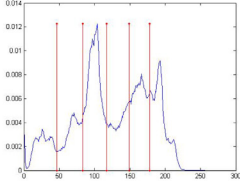
can be described as follows:

$$PSNR = 20 \log_{10} \frac{255}{RMSE}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^M \sum_{j=1}^N ((I(i, j) - Seg(i, j))^2)}{M \times N}} \tag{26}$$

where RMSE is the root-mean-squared error, I and Seg are the original and segmented images of size M × N, respectively.

Table 7
Results after applying MPA-OBL on Otsu's method to the set of benchmark images.

Image	nTh = 2	nTh = 3	nTh = 4	nTh = 5
				
Hunter				
				
Airplane				
				
Pepper				

SSIM. is another measure of the similarity between original and segmented image [61]. It can be described as follows:

$$SSIM(I, Seg) = \frac{(2\mu_1\mu_{Seg} + c_1)(2\sigma_{1,Seg} + c_2)}{(\mu_1^2 + \mu_{Seg}^2 + c_1)(\sigma_1^2 + \sigma_{Seg}^2 + c_2)} \quad (27)$$

where μ_1 and μ_{Seg} are the mean intensities of the original image I and segmented image (Seg) respectively, σ_1 and σ_{Seg} are the standard deviations of the original image I and segmented image Seg respectively, $\sigma_{1,Seg}$ is the covariance of the original image I and segmented image Seg , c_1 and c_2 are just two constants used.

FSIM. maps the features and measures the similarities [62]. To describe **FSIM**, it is needed to describe two criteria more clearly. They are phase congruency (PC) and gradient magnitude (GM). Phase congruency (PC) is a new method for detecting image features. While GM is the computation of the image gradient, which is very traditional in digital image processing. At first, calculate the similarity of these two images as:

$$S_{PC} = \frac{2PC_1PC_2 + T_1}{PC_1^2 + PC_2^2 + T_1} \quad (28)$$

where T_1 is a positive constant which increases the stability of S_{PC} , PC_1 and PC_2 are the PC for original and segmented images

Table 8
Results after applying MPA-OBL on Otsu's method to the set of benchmark images.





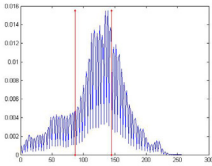
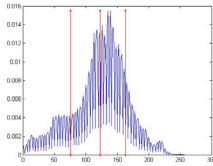
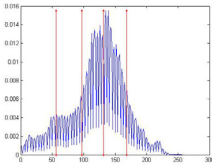
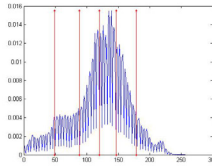




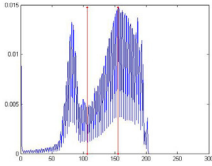
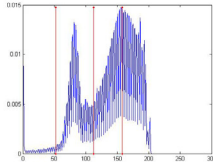
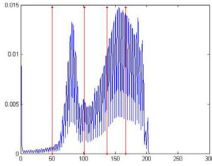
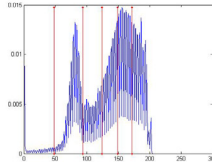




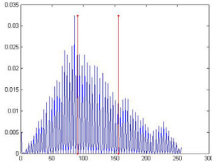
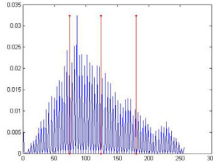
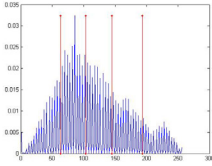
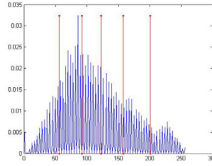


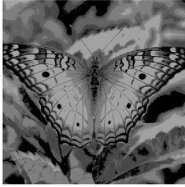

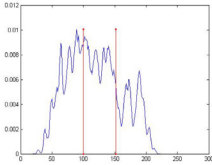
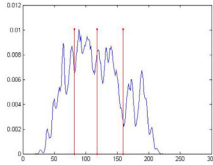
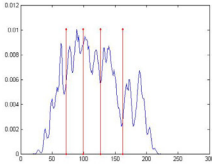
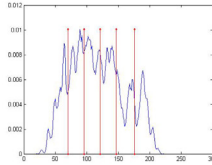
Image	nTh = 2	nTh = 3	nTh = 4	nTh = 5
Living-Room				
				
Woman				
				
Bridge				
				
Butter-Fly				
				

Table 9
Otsu's best threshold values.

Test image	nTh	LSHADE_SPACMA-OBL	CMA_ES-OBL	DE-OBL	HHO-OBL	SCA-OBL	SSA-OBL	MPA	MPA-OBL
Cameraman	2	70 144	70 144	70 144	70 144	70 144	70 144	70 144	70 144
	3	59 119 156	59 119 156	59 119 156	57 118 155	59 119 156	59 119 156	59 119 156	59 119 156
	4	42 95 140 170	42 95 140 170	42 95 140 170	34 89 136 167	42 95 140 170	42 95 140 170	42 95 140 170	42 95 140 170
Lena	5	35 82 122 149 173	36 82 122 149 173	36 82 122 149 173	18 62 109 144 172	36 82 122 149 173	36 82 122 149 173	36 82 122 149 173	36 82 122 149 173
	2	91 150	91 150	91 150	90 149	91 150	91 150	91 150	91 150
	3	79 125 170	79 125 170	79 125 170	79 125 169	79 125 170	79 125 170	79 125 170	79 125 170
Baboon	4	73 112 144 179	73 112 143 178	73 112 144 179	45 98 135 174	73 112 143 178	73 112 144 179	73 112 144 179	73 112 144 179
	5	72 108 135 159 187	64 92 119 147 180	72 108 135 159 187	11 70 110 142 178	72 108 135 159 187	64 92 119 147 180	72 108 135 159 187	72 108 135 159 187
	2	97 149	97 149	97 149	96 149	97 149	97 149	97 149	97 149
Hunter	3	85 125 161	85 125 161	85 125 161	75 120 159	85 125 161	85 125 161	85 125 161	85 125 161
	4	72 106 137 168	72 106 137 168	72 106 137 168	49 98 133 166	72 106 137 168	72 106 137 168	72 106 137 168	72 106 137 168
	5	66 98 124 148 174	66 97 123 148 174	67 99 125 149 174	14 66 104 136 168	67 99 125 149 174	67 99 125 149 174	67 99 125 149 174	67 99 125 149 174
Airplane	2	51 116	51 116	51 116	51 116	51 116	51 116	51 116	51 116
	3	36 86 135	36 86 135	36 86 135	35 85 134	36 86 135	36 86 135	36 86 135	36 86 135
	4	30 71 110 146	30 71 110 146	30 71 110 146	27 65 104 143	30 71 110 146	30 71 110 146	30 71 110 146	30 71 110 146
Pepper	5	22 53 88 122 152	22 53 88 122 152	22 53 88 122 152	12 41 80 116 149	22 53 88 122 152	22 53 88 122 152	22 53 88 122 152	22 53 88 122 152
	2	113 173	113 173	113 173	112 173	113 173	113 173	113 173	113 173
	3	93 145 191	93 145 191	93 145 191	92 144 190	93 145 191	93 145 191	93 145 191	93 145 191
Living-Room	4	84 129 172 203	84 129 172 203	84 129 172 203	62 117 163 199	84 129 172 203	84 129 172 203	84 129 172 203	84 129 172 203
	5	66 106 142 179 204	68 106 142 179 204	69 106 142 179 204	24 87 127 170 201	68 106 142 179 204	68 106 142 179 204	68 106 142 179 204	68 106 142 179 204
	2	72 138	72 138	72 138	71 138	72 138	72 138	72 138	72 138
Woman	3	65 122 169	65 122 169	65 122 169	62 120 167	65 122 169	65 122 169	65 122 169	65 122 169
	4	50 88 128 171	50 88 128 171	50 88 128 171	33 82 126 170	50 88 128 171	50 88 128 171	50 88 128 171	50 88 128 171
	5	48 85 118 151 180	48 85 118 150 179	49 85 118 150 179	17 58 96 134 173	48 85 118 150 179	48 85 118 150 179	48 85 118 150 179	48 85 118 150 179
Bridge	2	87 145	87 145	87 145	87 145	87 145	87 145	87 145	87 145
	3	76 123 163	76 123 163	76 123 163	73 122 162	76 123 163	76 123 163	76 123 163	76 123 163
	4	56 97 132 168	56 97 132 168	56 97 132 168	43 92 130 166	56 97 132 168	56 97 132 168	56 97 132 168	56 97 132 168
Butter-Fly	5	45 86 119 146 178	49 88 120 146 178	49 88 120 146 178	25 71 108 139 173	49 88 120 146 178	49 88 120 146 178	49 88 120 146 178	49 88 120 146 178
	2	106 155	106 155	106 155	106 155	106 155	106 155	106 155	106 155
	3	53 112 158	53 112 158	53 112 158	49 112 157	53 112 158	53 112 158	53 112 158	53 112 158
MPA	4	49 101 137 167	50 101 137 167	50 101 137 167	24 99 135 166	50 101 137 167	50 101 137 167	50 101 137 167	50 101 137 167
	5	47 95 124 150 173	48 94 124 149 172	48 95 125 150 173	5 58 104 139 167	48 94 124 149 172	48 95 125 150 173	48 95 125 150 173	48 95 125 150 173
	2	91 156	91 156	91 156	91 156	91 156	91 156	91 156	91 156
MPA-OBL	3	75 124 180	75 124 180	75 124 180	71 122 179	75 124 180	75 124 180	75 124 180	75 124 180
	4	63 103 145 193	63 103 145 193	63 103 145 193	50 96 140 190	63 103 145 193	63 103 145 193	63 103 145 193	63 103 145 193
	5	55 91 124 159 201	55 88 120 157 200	55 91 124 159 201	19 70 107 147 197	55 88 120 157 200	55 88 120 157 200	55 88 120 157 200	55 88 120 157 200
SSIM	2	100 152	100 152	100 152	100 152	100 152	100 152	100 152	100 152
	3	81 118 159	81 118 159	81 118 159	76 116 159	81 118 159	81 118 159	81 118 159	81 118 159
	4	72 99 127 162	71 99 127 162	72 99 127 162	33 88 122 160	71 99 127 162	72 99 127 162	72 99 127 162	72 99 127 162
SSIM	5	71 97 124 152 179	71 98 124 152 179	71 98 124 152 179	24 73 106 134 166	71 98 124 152 179	71 98 124 152 179	71 98 124 152 179	71 98 124 152 179

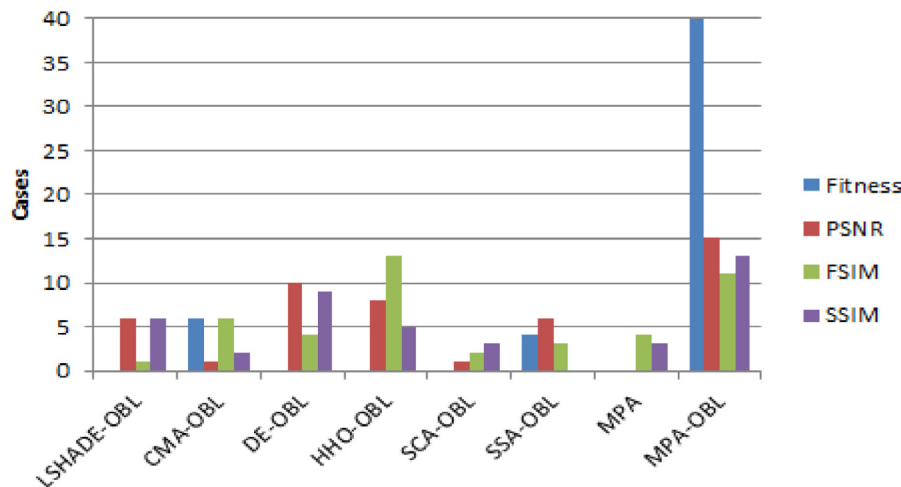


Fig. 8. Summary of Fitness, PSNR, SSIM, and SSIM number of best cases for all thresholds obtained by each algorithm based on Kapur's objective function.

respectively, then compute S_G which is the similarity from G_1 and G_2 as:

$$S_G = \frac{2G_1G_2 + T_2}{G_1^2 + G_2^2 + T_2} \quad (29)$$

where G_1 and G_2 are the gradients of original image and segmented one respectively, T_2 is a positive constant which depends on the dynamic range of GM values. From Eqs. (28) and (29), calculate the similarity as follow:

$$S_L(x) = [S_{PC}(x)]^\alpha [S_G(x)]^\beta \quad (30)$$

where the parameters α and β are used to adjust the relative importance of PC and GM features. Note that the high values of Fitness, PSNR, SSIM, and FSIM indicates high performance of the algorithm.

Additionally, to statistically determine the significance of the proposed algorithm, the Wilcoxon rank-sum and Friedman mean

rank tests are used to show the difference between the performance of the proposed algorithm MPA-OBL and all other algorithms used in the comparison.

Wilcoxon rank-sum test. is a non-parametric test used to compare the results obtained by each pair of algorithms [63]. This test is based on two hypotheses: the first is the null hypothesis, and the other is the alternative hypothesis. The null hypothesis makes assumptions that there is no difference between the ranks of the results obtained by a pair of algorithms, and the alternative hypothesis considers that there is a difference between the ranks of the results obtained by a pair of algorithms. Wilcoxon rank-sum test is based here on a 5% significant level. The (P) and (H) values in terms of fitness is reported, while using Otsu's method is reported in Table 15, and Kapur's method is reported in Table 25. According to (P) and (H) values, if $(P > 0.05)$ or $(H = 0)$, then the null hypothesis is true, whereas if $(P < 0.05)$ or $(H = 1)$, then the alternative hypothesis is true.

Table 10
MPA-OBL Otsu's in terms of fitness values over all competed algorithms.

Test image	nTh	LSHADE_SPACMA-OBL		CMA_ES-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	1.73E+01	4.65E-02	1.72E+01	1.08E-14	1.72E+01	1.08E-14	1.46E+01	2.17E+00	1.73E+01	4.82E-02	1.72E+01	1.08E-14	1.72E+01	1.08E-14	1.72E+01	1.08E-14
	3	2.02E+01	9.53E-02	2.02E+01	7.23E-15	2.02E+01	7.23E-15	1.78E+01	4.14E-01	2.02E+01	1.34E-01	2.02E+01	6.55E-03	2.02E+01	0.00E+00	2.02E+01	7.23E-15
	4	2.16E+01	2.65E-01	2.15E+01	0.00E+00	2.15E+01	0.00E+00	2.06E+01	3.14E-01	2.12E+01	4.87E-01	2.15E+01	2.78E-02	2.15E+01	7.21E-15	2.15E+01	0.00E+00
Test2	2	2.30E+01	3.20E-01	2.33E+01	7.23E-15	2.33E+01	7.23E-15	2.16E+01	2.04E-01	2.26E+01	5.22E-01	2.32E+01	2.71E-01	2.33E+01	1.44E-14	2.33E+01	7.23E-15
	3	1.54E+01	2.54E-02	1.54E+01	7.23E-15	1.54E+01	7.23E-15	1.27E+01	4.63E-01	1.54E+01	4.02E-02	1.54E+01	7.23E-15	1.54E+01	7.21E-15	1.54E+01	7.23E-15
	4	1.74E+01	8.35E-02	1.74E+01	1.08E-14	1.74E+01	1.08E-14	1.57E+01	2.68E-01	1.74E+01	1.12E-01	1.74E+01	1.08E-14	1.74E+01	1.08E-14	1.74E+01	1.08E-14
Test3	2	1.88E+01	1.44E-01	1.88E+01	7.23E-15	1.88E+01	7.23E-15	1.76E+01	1.37E-01	1.85E+01	3.25E-01	1.88E+01	3.20E-03	1.88E+01	0.00E+00	1.88E+01	7.23E-15
	3	1.98E+01	3.83E-01	1.94E+01	1.45E-14	1.94E+01	1.94E-03	1.89E+01	2.32E-01	1.94E+01	8.43E-01	1.98E+01	3.70E-01	1.94E+01	1.44E-14	1.98E+01	3.53E-01
	4	1.54E+01	2.16E-02	1.54E+01	1.08E-14	1.54E+01	1.08E-14	1.12E+01	2.13E-01	1.54E+01	6.15E-02	1.54E+01	1.08E-14	1.54E+01	1.26E-14	1.54E+01	1.08E-14
Test4	2	1.79E+01	2.89E-01	1.77E+01	7.23E-15	1.77E+01	7.23E-15	1.56E+01	2.03E-01	1.78E+01	5.32E-01	1.77E+01	3.98E-03	1.77E+01	7.21E-15	1.77E+01	7.23E-15
	3	2.02E+01	3.29E-01	2.02E+01	3.61E-15	2.02E+01	3.61E-15	1.79E+01	1.70E-01	1.97E+01	1.01E+00	2.02E+01	4.10E-02	2.02E+01	1.08E-14	2.02E+01	3.61E-15
	4	2.19E+01	5.04E-01	2.16E+01	2.22E-03	2.16E+01	2.24E-02	2.05E+01	2.12E-01	2.13E+01	9.79E-01	2.16E+01	6.22E-02	2.16E+01	7.21E-15	2.16E+01	7.23E-15
Test5	2	1.79E+01	1.44E-02	1.79E+01	0.00E+00	1.79E+01	0.00E+00	1.56E+01	1.20E+00	1.79E+01	1.21E-02	1.79E+01	0.00E+00	1.79E+01	0.00E+00	1.79E+01	0.00E+00
	3	2.03E+01	4.80E-02	2.03E+01	3.61E-15	2.03E+01	3.61E-15	1.85E+01	5.62E-01	2.03E+01	8.74E-02	2.03E+01	3.91E-03	2.03E+01	1.00E-14	2.03E+01	3.61E-15
	4	2.21E+01	9.04E-02	2.22E+01	0.00E+00	2.22E+01	0.00E+00	2.06E+01	2.85E-01	2.20E+01	1.53E-01	2.22E+01	2.06E-02	2.22E+01	7.21E-15	2.22E+01	0.00E+00
Test6	2	2.35E+01	1.87E-01	2.37E+01	0.00E+00	2.37E+01	0.00E+00	2.25E+01	3.02E-01	2.28E+01	5.14E-01	2.37E+01	1.10E-02	2.37E+01	0.00E+00	2.37E+01	0.00E+00
	3	1.50E+01	2.59E-02	1.50E+01	9.03E-15	1.50E+01	9.03E-15	1.23E+01	7.28E-01	1.50E+01	7.14E-02	1.50E+01	9.03E-15	1.50E+01	1.26E-14	1.50E+01	9.03E-15
	4	1.88E+01	1.99E-01	1.88E+01	1.45E-14	1.88E+01	1.23E-02	1.56E+01	6.34E-01	1.88E+01	3.41E-01	1.88E+01	2.34E-02	1.88E+01	1.44E-14	1.88E+01	1.45E-14
Test7	2	2.08E+01	2.97E-01	2.07E+01	3.61E-15	2.07E+01	1.07E-02	1.92E+01	3.79E-01	2.08E+01	6.66E-01	2.07E+01	2.29E-02	2.07E+01	1.08E-14	2.07E+01	3.61E-15
	3	2.28E+01	4.12E-01	2.32E+01	3.61E-15	2.32E+01	1.30E-02	2.09E+01	5.43E-01	2.19E+01	8.07E-01	2.31E+01	3.82E-02	2.32E+01	1.08E-14	2.32E+01	3.61E-15
	4	1.63E+01	7.02E-03	1.63E+01	1.08E-14	1.63E+01	1.08E-14	1.17E+01	3.85E-01	1.63E+01	1.77E-02	1.63E+01	1.08E-14	1.63E+01	1.08E-14	1.63E+01	1.08E-14
Test8	2	1.84E+01	4.94E-02	1.84E+01	1.45E-14	1.84E+01	1.45E-14	1.72E+01	7.39E-01	1.83E+01	1.15E-01	1.84E+01	1.45E-14	1.84E+01	1.44E-14	1.84E+01	1.45E-14
	3	2.07E+01	1.18E-01	2.07E+01	3.61E-15	2.07E+01	3.61E-15	1.85E+01	1.64E-01	2.03E+01	6.92E-01	2.07E+01	7.53E-03	2.07E+01	1.08E-14	2.07E+01	3.61E-15
	4	2.21E+01	1.78E-01	2.23E+01	1.81E-14	2.23E+01	1.81E-14	2.09E+01	1.96E-01	2.15E+01	5.44E-01	2.23E+01	8.06E-03	2.23E+01	1.80E-14	2.23E+01	1.81E-14
Test9	2	1.60E+01	1.30E-02	1.60E+01	1.08E-14	1.60E+01	1.08E-14	1.40E+01	6.49E-01	1.60E+01	3.16E-02	1.60E+01	1.08E-14	1.60E+01	5.41E-15	1.60E+01	1.08E-14
	3	1.82E+01	9.34E-02	1.82E+01	7.23E-15	1.82E+01	7.23E-15	1.63E+01	1.76E-01	1.81E+01	1.79E-01	1.82E+01	1.02E-02	1.82E+01	7.21E-15	1.82E+01	7.23E-15
	4	2.05E+01	1.62E-01	2.07E+01	1.81E-14	2.07E+01	1.81E-14	1.84E+01	1.38E-01	2.03E+01	7.57E-01	2.07E+01	1.05E-02	2.07E+01	1.80E-14	2.07E+01	1.81E-14
Test10	2	2.20E+01	2.34E-01	2.22E+01	7.23E-15	2.22E+01	7.23E-15	2.09E+01	2.17E-01	2.14E+01	6.03E-01	2.22E+01	3.06E-02	2.22E+01	1.44E-14	2.22E+01	7.23E-15
	3	1.46E+01	2.43E-02	1.46E+01	0.00E+00	1.46E+01	0.00E+00	1.26E+01	4.74E-01	1.46E+01	3.82E-02	1.46E+01	0.00E+00	1.46E+01	0.00E+00	1.46E+01	0.00E+00
	4	1.91E+01	3.03E-01	1.92E+01	3.61E-15	1.92E+01	3.61E-15	1.47E+01	1.16E-01	1.87E+01	9.59E-01	1.92E+01	1.64E-02	1.92E+01	3.60E-15	1.92E+01	1.14E-01
Test11	2	2.09E+01	4.06E-01	2.12E+01	0.00E+00	2.12E+01	0.00E+00	1.80E+01	1.71E+00	1.90E+01	2.31E+00	2.11E+01	5.77E-02	2.12E+01	7.21E-15	2.12E+01	0.00E+00
	3	2.20E+01	7.35E-01	2.24E+01	3.61E-15	2.24E+01	8.30E-03	2.05E+01	1.42E+00	2.13E+01	1.17E+00	2.24E+01	6.31E-02	2.24E+01	1.08E-14	2.24E+01	3.61E-15
	4	1.40E+01	2.04E-02	1.35E+01	3.61E-15	1.40E+01	1.43E-02	1.10E+01	1.94E-01	1.34E+01	1.93E-02	1.40E+01	1.40E-02	1.40E+01	1.42E-02	1.40E+01	1.42E-02
Test12	2	1.68E+01	1.84E-01	1.66E+01	3.61E-15	1.66E+01	1.08E-14	1.44E+01	1.67E-01	1.69E+01	1.87E-01	1.66E+01	1.35E-01	1.66E+01	1.08E-14	1.66E+01	8.18E-02
	3	1.89E+01	1.53E-01	1.89E+01	0.00E+00	1.89E+01	0.00E+00	1.70E+01	2.63E-01	1.86E+01	4.09E-01	1.89E+01	1.04E-01	1.89E+01	7.21E-15	1.89E+01	0.00E+00
	4	2.06E+01	2.77E-01	1.99E+01	0.00E+00	2.06E+01	2.15E-02	1.92E+01	2.15E-01	2.02E+01	6.36E-01	2.06E+01	1.07E-01	2.06E+01	2.14E-02	2.06E+01	2.15E-02
Test13	2	1.37E+01	6.84E-02	1.37E+01	0.00E+00	1.37E+01	0.00E+00	1.12E+01	1.82E-01	1.37E+01	7.98E-02	1.37E+01	0.00E+00	1.37E+01	0.00E+00	1.37E+01	0.00E+00
	3	1.69E+01	1.38E-01	1.70E+01	7.23E-15	1.70E+01	7.23E-15	1.38E+01	8.89E-02	1.70E+01	3.20E-01	1.70E+01	7.23E-15	1.70E+01	0.00E+00	1.70E+01	7.23E-15
	4	1.90E+01	5.29E-01	1.91E+01	1.08E-14	1.91E+01	1.08E-14	1.71E+01	1.43E-01	1.81E+01	1.03E+00	1.91E+01	5.83E-02	1.91E+01	1.08E-14	1.91E+01	1.08E-14
Test14	2	1.97E+01	7.35E-01	1.97E+01	0.00E+00	1.97E+01	0.00E+00	1.93E+01	6.15E-01	1.92E+01	7.76E-01	1.97E+01	4.50E-02	1.97E+01	0.00E+00	1.97E+01	0.00E+00
	3	1.97E+01	7.35E-01	1.97E+01	0.00E+00	1.97E+01	0.00E+00	1.93E+01	6.15E-01	1.92E+01	7.76E-01	1.97E+01	4.50E-02	1.97E+01	0.00E+00	1.97E+01	0.00E+00
	4	1.97E+01	7.35E-01	1.97E+01	0.00E+00	1.97E+01	0.00E+00	1.93E+01	6.15E-01	1.92E+01	7.76E-01	1.97E+01	4.50E-02	1.97E+01	0.00E+00	1.97E+01	0.00E+00
Fridman mean rank		4.92		5.21		5.12		1.03		3.92		5.32		5.15		5.33	
Rank		6		3		5		8		7		2		4		1	

Table 11
MPA-OBL Otsu's in terms of PSNR values over all competed algorithms.

Test image	nTh	LSHADE_SPACMA-OBL		CMA_ES-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	1.73E+01	4.65E-02	1.72E+01	1.08E-14	1.72E+01	1.08E-14	1.46E+01	2.17E+00	1.73E+01	4.82E-02	1.72E+01	1.08E-14	1.72E+01	1.08E-14	1.72E+01	1.08E-14
	3	2.02E+01	9.53E-02	2.02E+01	7.23E-15	2.02E+01	7.23E-15	1.78E+01	4.14E-01	2.02E+01	1.34E-01	2.02E+01	6.55E-03	2.02E+01	0.00E+00	2.02E+01	7.23E-15
	4	2.16E+01	2.65E-01	2.15E+01	0.00E+00	2.15E+01	0.00E+00	2.06E+01	3.14E-01	2.12E+01	4.87E-01	2.15E+01	2.78E-02	2.15E+01	7.21E-15	2.15E+01	0.00E+00
Test2	5	2.30E+01	3.20E-01	2.33E+01	7.23E-15	2.33E+01	7.23E-15	2.16E+01	2.04E-01	2.26E+01	5.22E-01	2.32E+01	2.71E-01	2.33E+01	1.44E-14	2.33E+01	7.23E-15
	2	1.54E+01	2.54E-02	1.54E+01	7.23E-15	1.54E+01	7.23E-15	1.27E+01	4.63E-01	1.54E+01	4.02E-02	1.54E+01	7.23E-15	1.54E+01	7.21E-15	1.54E+01	7.23E-15
	3	1.74E+01	8.35E-02	1.74E+01	1.08E-14	1.74E+01	1.08E-14	1.57E+01	2.68E-01	1.74E+01	1.12E-01	1.74E+01	1.08E-14	1.74E+01	1.08E-14	1.74E+01	1.08E-14
Test3	4	1.88E+01	1.44E-01	1.88E+01	7.23E-15	1.88E+01	7.23E-15	1.76E+01	1.37E-01	1.85E+01	3.25E-01	1.88E+01	3.20E-03	1.88E+01	0.00E+00	1.88E+01	7.23E-15
	5	1.98E+01	3.83E-01	1.94E+01	1.45E-14	1.94E+01	1.94E-03	1.89E+01	2.32E-01	1.94E+01	8.43E-01	1.98E+01	3.70E-01	1.94E+01	1.44E-14	1.98E+01	3.53E-01
	2	1.54E+01	2.16E-02	1.54E+01	1.08E-14	1.54E+01	1.08E-14	1.12E+01	2.13E-01	1.54E+01	6.15E-02	1.54E+01	1.08E-14	1.54E+01	1.26E-14	1.54E+01	1.08E-14
Test4	3	1.79E+01	2.89E-01	1.77E+01	7.23E-15	1.77E+01	7.23E-15	1.56E+01	2.03E-01	1.78E+01	5.32E-01	1.77E+01	3.98E-03	1.77E+01	7.21E-15	1.77E+01	7.23E-15
	4	2.02E+01	3.29E-01	2.02E+01	3.61E-15	2.02E+01	3.61E-15	1.79E+01	1.70E-01	1.97E+01	1.01E+00	2.02E+01	4.10E-02	2.02E+01	1.08E-14	2.02E+01	3.61E-15
	5	2.19E+01	5.04E-01	2.16E+01	2.22E-03	2.16E+01	2.24E-02	2.05E+01	2.12E-01	2.13E+01	9.79E-01	2.16E+01	6.22E-02	2.16E+01	7.21E-15	2.16E+01	7.23E-15
Test5	2	1.79E+01	1.44E-02	1.79E+01	0.00E+00	1.79E+01	0.00E+00	1.56E+01	1.20E+00	1.79E+01	1.21E-02	1.79E+01	0.00E+00	1.79E+01	0.00E+00	1.79E+01	0.00E+00
	3	2.03E+01	4.80E-02	2.03E+01	3.61E-15	2.03E+01	3.61E-15	1.85E+01	5.62E-01	2.03E+01	8.74E-02	2.03E+01	3.91E-03	2.03E+01	1.08E-14	2.03E+01	3.61E-15
	4	2.21E+01	9.04E-02	2.22E+01	0.00E+00	2.22E+01	0.00E+00	2.06E+01	2.85E-01	2.20E+01	1.53E-01	2.22E+01	2.06E-02	2.22E+01	7.21E-15	2.22E+01	0.00E+00
Test6	5	2.35E+01	1.87E-01	2.37E+01	0.00E+00	2.37E+01	0.00E+00	2.25E+01	3.02E-01	2.28E+01	5.14E-01	2.37E+01	1.10E-02	2.37E+01	0.00E+00	2.37E+01	0.00E+00
	2	1.50E+01	2.59E-02	1.50E+01	9.03E-15	1.50E+01	9.03E-15	1.23E+01	7.28E-01	1.50E+01	7.14E-02	1.50E+01	9.03E-15	1.50E+01	1.26E-14	1.50E+01	9.03E-15
	3	1.88E+01	1.99E-01	1.88E+01	1.45E-14	1.88E+01	1.23E-02	1.56E+01	6.34E-01	1.88E+01	3.41E-01	1.88E+01	2.34E-02	1.88E+01	1.44E-14	1.88E+01	1.45E-14
Test7	4	2.08E+01	2.97E-01	2.07E+01	3.61E-15	2.07E+01	1.07E-02	1.92E+01	3.79E-01	2.08E+01	6.66E-01	2.07E+01	2.29E-02	2.07E+01	1.08E-14	2.07E+01	3.61E-15
	5	2.28E+01	4.12E-01	2.32E+01	3.61E-15	2.32E+01	1.30E-02	2.09E+01	5.43E-01	2.19E+01	8.07E-01	2.31E+01	3.82E-02	2.32E+01	1.08E-14	2.32E+01	3.61E-15
	2	1.63E+01	7.02E-03	1.63E+01	1.08E-14	1.63E+01	1.08E-14	1.17E+01	3.85E-01	1.63E+01	1.77E-02	1.63E+01	1.08E-14	1.63E+01	1.08E-14	1.63E+01	1.08E-14
Test8	3	1.84E+01	4.94E-02	1.84E+01	1.45E-14	1.84E+01	1.45E-14	1.72E+01	7.39E-01	1.83E+01	1.15E-01	1.84E+01	1.45E-14	1.84E+01	1.44E-14	1.84E+01	1.45E-14
	4	2.07E+01	1.18E-01	2.07E+01	3.61E-15	2.07E+01	3.61E-15	1.85E+01	1.64E-01	2.03E+01	6.92E-01	2.07E+01	7.53E-03	2.07E+01	1.08E-14	2.07E+01	3.61E-15
	5	2.21E+01	1.78E-01	2.23E+01	1.81E-14	2.23E+01	1.81E-14	2.09E+01	1.96E-01	2.15E+01	5.44E-01	2.23E+01	8.06E-03	2.23E+01	1.80E-14	2.23E+01	1.81E-14
Test9	2	1.60E+01	1.30E-02	1.60E+01	1.08E-14	1.60E+01	1.08E-14	1.40E+01	6.49E-01	1.60E+01	3.16E-02	1.60E+01	1.08E-14	1.60E+01	5.41E-15	1.60E+01	1.08E-14
	3	1.82E+01	9.34E-02	1.82E+01	7.23E-15	1.82E+01	7.23E-15	1.63E+01	1.76E-01	1.81E+01	1.79E-01	1.82E+01	1.02E-02	1.82E+01	7.21E-15	1.82E+01	7.23E-15
	4	2.05E+01	1.62E-01	2.07E+01	1.81E-14	2.07E+01	1.81E-14	1.84E+01	1.38E-01	2.03E+01	7.57E-01	2.07E+01	1.05E-02	2.07E+01	1.80E-14	2.07E+01	1.81E-14
Test10	5	2.20E+01	2.34E-01	2.22E+01	7.23E-15	2.22E+01	7.23E-15	2.09E+01	2.17E-01	2.14E+01	6.03E-01	2.22E+01	3.06E-02	2.22E+01	1.44E-14	2.22E+01	7.23E-15
	2	1.46E+01	2.43E-02	1.46E+01	0.00E+00	1.46E+01	0.00E+00	1.26E+01	4.74E-01	1.46E+01	3.82E-02	1.46E+01	0.00E+00	1.46E+01	0.00E+00	1.46E+01	0.00E+00
	3	1.91E+01	3.03E-01	1.92E+01	3.61E-15	1.92E+01	3.61E-15	1.47E+01	1.16E-01	1.87E+01	9.59E-01	1.92E+01	1.64E-02	1.92E+01	3.60E-15	1.92E+01	1.14E-01
Test11	4	2.09E+01	4.06E-01	2.12E+01	0.00E+00	2.12E+01	0.00E+00	1.80E+01	1.71E+00	1.90E+01	2.31E+00	2.11E+01	5.77E-02	2.12E+01	7.21E-15	2.12E+01	0.00E+00
	5	2.20E+01	7.35E-01	2.24E+01	3.61E-15	2.24E+01	8.30E-03	2.05E+01	1.42E+00	2.13E+01	1.17E+00	2.24E+01	6.31E-02	2.24E+01	1.08E-14	2.24E+01	3.61E-15
	2	1.40E+01	2.04E-02	1.35E+01	3.61E-15	1.40E+01	1.43E-02	1.10E+01	1.94E-01	1.34E+01	1.93E-02	1.40E+01	1.40E-02	1.40E+01	1.42E-02	1.40E+01	1.42E-02
Test12	3	1.68E+01	1.84E-01	1.66E+01	3.61E-15	1.66E+01	1.08E-14	1.44E+01	1.67E-01	1.69E+01	1.87E-01	1.66E+01	1.35E-01	1.66E+01	1.08E-14	1.66E+01	8.18E-02
	4	1.89E+01	1.53E-01	1.89E+01	0.00E+00	1.89E+01	0.00E+00	1.70E+01	2.63E-01	1.86E+01	4.09E-01	1.89E+01	1.04E-01	1.89E+01	7.21E-15	1.89E+01	0.00E+00
	5	2.06E+01	2.77E-01	1.99E+01	0.00E+00	2.06E+01	2.15E-02	1.92E+01	2.15E-01	2.02E+01	6.36E-01	2.06E+01	1.07E-01	2.06E+01	2.14E-02	2.06E+01	2.15E-02
Test13	2	1.37E+01	6.84E-02	1.37E+01	0.00E+00	1.37E+01	0.00E+00	1.12E+01	1.82E-01	1.37E+01	7.98E-02	1.37E+01	0.00E+00	1.37E+01	0.00E+00	1.37E+01	0.00E+00
	3	1.69E+01	1.38E-01	1.70E+01	7.23E-15	1.70E+01	7.23E-15	1.38E+01	8.89E-02	1.70E+01	3.20E-01	1.70E+01	7.23E-15	1.70E+01	0.00E+00	1.70E+01	7.23E-15
	4	1.90E+01	5.29E-01	1.91E+01	1.08E-14	1.91E+01	1.08E-14	1.71E+01	1.43E-01	1.81E+01	1.03E+00	1.91E+01	5.83E-02	1.91E+01	1.08E-14	1.91E+01	1.08E-14
5	1.97E+01	7.35E-01	1.97E+01	0.00E+00	1.97E+01	0.00E+00	1.93E+01	6.15E-01	1.92E+01	7.76E-01	1.97E+01	4.50E-02	1.97E+01	0.00E+00	1.97E+01	0.00E+00	

Table 12
MPA-OBL Otsu's in terms of SSIM values over all competed algorithms.

Test image	nTh	LSHADE_SPACMA-OBL		CMA_ES-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	7.68E-01	1.04E-03	7.68E-01	2.26E-16	7.68E-01	2.26E-16	7.16E-01	4.36E-02	7.68E-01	2.05E-03	7.68E-01	2.26E-16	7.68E-01	4.51E-16	7.68E-01	2.26E-16
	3	8.08E-01	4.83E-03	8.07E-01	5.65E-16	8.07E-01	5.65E-16	7.68E-01	5.21E-03	8.08E-01	7.24E-03	8.07E-01	5.65E-16	8.07E-01	7.89E-16	8.07E-01	4.43E-05
	4	8.30E-01	3.01E-03	8.31E-01	2.26E-16	8.31E-01	2.26E-16	8.09E-01	3.62E-03	8.26E-01	6.66E-03	8.31E-01	2.26E-16	8.31E-01	4.51E-16	8.31E-01	4.08E-04
	5	8.55E-01	6.31E-03	8.58E-01	2.26E-16	8.58E-01	2.26E-16	8.31E-01	2.49E-03	8.49E-01	7.51E-03	8.58E-01	2.26E-16	8.58E-01	3.38E-16	8.57E-01	3.69E-03
Test2	2	6.46E-01	2.36E-03	6.46E-01	1.13E-16	6.46E-01	1.13E-16	5.19E-01	1.23E-02	6.46E-01	3.57E-03	6.46E-01	1.13E-16	6.46E-01	1.13E-16	6.46E-01	1.13E-16
	3	7.22E-01	6.05E-03	7.22E-01	2.26E-16	7.22E-01	2.26E-16	6.49E-01	5.31E-03	7.25E-01	8.42E-03	7.22E-01	2.26E-16	7.22E-01	4.51E-16	7.22E-01	2.26E-16
	4	7.65E-01	6.56E-03	7.65E-01	2.26E-16	7.65E-01	2.26E-16	7.23E-01	3.21E-03	7.57E-01	1.13E-02	7.65E-01	2.26E-16	7.65E-01	4.51E-16	7.65E-01	1.29E-04
	5	7.97E-01	1.29E-02	7.99E-01	1.18E-02	7.86E-01	1.18E-04	7.65E-01	6.88E-03	7.86E-01	2.60E-02	7.86E-01	2.26E-16	7.86E-01	2.25E-16	7.99E-01	1.25E-02
Test3	2	7.89E-01	1.11E-03	7.89E-01	1.13E-16	7.89E-01	1.13E-16	5.45E-01	9.88E-03	7.89E-01	3.11E-03	7.89E-01	1.13E-16	7.89E-01	1.13E-16	7.89E-01	1.13E-16
	3	8.59E-01	7.57E-03	8.54E-01	5.65E-16	8.54E-01	5.65E-16	7.93E-01	4.95E-03	8.55E-01	1.44E-02	8.54E-01	5.65E-16	8.54E-01	7.89E-16	8.54E-01	4.52E-05
	4	9.13E-01	7.52E-03	9.12E-01	2.26E-16	9.12E-01	2.26E-16	8.58E-01	3.96E-03	9.02E-01	2.35E-02	9.12E-01	3.26E-16	9.12E-01	3.38E-16	9.13E-01	9.37E-04
	5	9.37E-01	7.94E-03	9.33E-01	3.39E-16	9.33E-01	3.54E-04	9.18E-01	3.87E-03	9.29E-01	1.66E-02	9.33E-01	3.85E-06	9.33E-01	5.63E-16	9.34E-01	1.00E-03
Test4	2	7.58E-01	1.03E-03	7.58E-01	2.26E-16	7.58E-01	2.26E-16	6.36E-01	3.07E-02	7.58E-01	1.04E-03	7.58E-01	2.26E-16	7.58E-01	2.25E-16	7.58E-01	2.26E-16
	3	8.47E-01	2.21E-03	8.47E-01	3.39E-16	8.47E-01	3.39E-16	7.67E-01	7.07E-03	8.47E-01	3.23E-03	8.47E-01	3.39E-16	8.47E-01	5.63E-16	8.47E-01	4.19E-04
	4	8.89E-01	6.72E-03	8.86E-01	3.39E-16	8.86E-01	3.39E-16	8.51E-01	3.21E-03	8.89E-01	7.44E-03	8.86E-01	3.39E-16	8.86E-01	3.38E-16	8.89E-01	2.74E-03
	5	9.19E-01	3.01E-03	9.22E-01	4.52E-16	9.22E-01	4.52E-16	8.94E-01	5.13E-03	9.09E-01	1.11E-02	9.22E-01	4.52E-16	9.22E-01	5.63E-16	9.22E-01	2.06E-04
Test5	2	8.40E-01	9.49E-04	8.39E-01	2.26E-16	8.39E-01	2.26E-16	7.34E-01	2.25E-02	8.40E-01	2.63E-02	8.39E-01	2.26E-16	8.39E-01	2.25E-16	8.39E-01	2.26E-16
	3	8.92E-01	7.41E-03	8.89E-01	2.26E-16	8.90E-01	7.91E-04	8.39E-01	9.11E-03	8.92E-01	1.00E-02	8.89E-01	2.26E-16	8.89E-01	2.25E-16	8.90E-01	1.50E-03
	4	9.12E-01	6.45E-03	9.10E-01	4.52E-16	9.10E-01	2.99E-04	8.91E-01	4.95E-03	9.13E-01	1.35E-02	9.10E-01	4.52E-16	9.10E-01	4.51E-16	9.10E-01	5.32E-04
	5	9.41E-01	7.45E-03	9.46E-01	3.39E-16	9.46E-01	2.77E-04	9.10E-01	4.11E-03	9.28E-01	1.25E-02	9.46E-01	3.39E-16	9.46E-01	1.13E-16	9.45E-01	6.98E-04
Test6	2	7.59E-01	3.40E-04	7.59E-01	1.13E-16	7.59E-01	1.13E-16	4.68E-01	9.21E-03	7.59E-01	1.71E-03	7.59E-01	1.13E-16	7.59E-01	3.38E-16	7.59E-01	1.13E-16
	3	7.98E-01	2.77E-03	7.99E-01	0.00E+00	7.99E-01	0.00E+00	7.66E-01	7.39E-03	7.99E-01	4.87E-03	7.99E-01	0.00E+00	7.99E-01	0.00E+00	7.99E-01	0.00E+00
	4	8.59E-01	3.18E-03	8.61E-01	0.00E+00	8.61E-01	0.00E+00	8.01E-01	2.25E-03	8.51E-01	1.64E-02	8.61E-01	0.00E+00	8.61E-01	0.00E+00	8.61E-01	1.35E-04
	5	8.83E-01	3.15E-03	8.84E-01	5.65E-16	8.84E-01	5.65E-16	8.63E-01	1.69E-03	8.75E-01	1.03E-02	8.84E-01	5.65E-16	8.84E-01	5.63E-16	8.84E-01	2.53E-04
Test7	2	7.31E-01	8.13E-04	7.31E-01	0.00E+00	7.31E-01	0.00E+00	6.25E-01	2.47E-02	7.31E-01	1.55E-03	7.31E-01	0.00E+00	7.31E-01	0.00E+00	7.31E-01	0.00E+00
	3	8.08E-01	3.72E-03	8.08E-01	2.26E-16	8.08E-01	2.26E-16	7.39E-01	5.90E-03	8.06E-01	6.69E-03	8.08E-01	2.26E-16	8.08E-01	2.25E-16	8.08E-01	2.14E-04
	4	8.81E-01	8.75E-03	8.86E-01	5.65E-16	8.86E-01	5.65E-16	8.11E-01	3.15E-03	8.77E-01	2.49E-02	8.86E-01	5.65E-16	8.86E-01	5.63E-16	8.86E-01	5.68E-04
	5	9.14E-01	7.23E-03	9.18E-01	1.13E-16	9.18E-01	1.13E-16	8.92E-01	4.66E-03	9.03E-01	1.56E-02	9.18E-01	1.13E-16	9.18E-01	0.00E+00	9.20E-01	1.49E-03
Test8	2	7.00E-01	1.01E-03	7.01E-01	2.26E-16	7.01E-01	2.26E-16	6.03E-01	1.68E-02	7.00E-01	1.85E-03	7.01E-01	2.26E-16	7.01E-01	4.51E-16	7.01E-01	2.26E-16
	3	8.83E-01	6.10E-03	8.83E-01	2.48E-03	8.83E-01	2.26E-16	7.01E-01	3.51E-03	8.71E-01	3.60E-02	8.83E-01	2.26E-16	8.83E-01	2.25E-16	8.83E-01	3.36E-04
	4	9.18E-01	6.60E-03	9.23E-01	2.26E-16	9.23E-01	2.26E-16	8.39E-01	6.06E-02	8.63E-01	7.05E-02	9.23E-01	2.26E-16	9.23E-01	1.13E-16	9.22E-01	7.64E-04
	5	9.34E-01	1.18E-02	9.42E-01	3.39E-16	9.42E-01	8.41E-05	9.05E-01	3.78E-02	9.20E-01	2.17E-02	9.42E-01	3.39E-16	9.42E-01	5.63E-16	9.42E-01	6.33E-04
Test9	2	6.14E-01	4.07E-04	6.14E-01	1.19E-04	6.14E-01	1.20E-04	3.84E-01	1.53E-02	5.85E-01	9.46E-04	5.85E-01	3.39E-16	6.14E-01	1.20E-04	6.14E-01	1.18E-04
	3	7.68E-01	1.00E-02	7.58E-01	3.85E-03	7.57E-01	2.26E-16	6.32E-01	8.55E-03	7.71E-01	1.09E-02	7.59E-01	0.00E+00	7.57E-01	2.25E-16	7.59E-01	6.43E-03
	4	8.50E-01	7.33E-03	8.46E-01	4.52E-16	8.46E-01	4.52E-16	7.71E-01	1.19E-02	8.35E-01	2.09E-02	8.48E-01	5.65E-16	8.46E-01	4.51E-16	8.47E-01	3.96E-03
	5	8.95E-01	9.47E-03	8.92E-01	4.86E-05	8.92E-01	4.86E-05	8.55E-01	7.13E-03	8.85E-01	2.27E-02	8.68E-01	0.00E+00	8.92E-01	4.83E-05	8.93E-01	3.39E-03
Test10	2	6.01E-01	4.10E-03	5.99E-01	1.13E-16	5.99E-01	1.13E-16	4.24E-01	8.00E-03	6.01E-01	4.96E-03	5.99E-01	1.13E-16	5.99E-01	1.13E-16	5.99E-01	1.13E-16
	3	7.62E-01	7.06E-03	7.64E-01	2.26E-16	7.64E-01	2.26E-16	6.00E-01	2.31E-03	7.65E-01	1.62E-02	7.64E-01	2.26E-16	7.64E-01	4.51E-16	7.64E-01	2.26E-16
	4	8.36E-01	1.84E-02	8.41E-01	5.65E-16	8.41E-01	5.65E-16	7.67E-01	3.65E-03	8.05E-01	3.68E-02	8.41E-01	5.65E-16	8.41E-01	5.63E-16	8.42E-01	1.90E-03
	5	8.55E-01	2.23E-02	8.57E-01	3.39E-16	8.57E-01	3.39E-16	8.45E-01	1.95E-02	8.43E-01	2.61E-02	8.57E-01	3.39E-16	8.57E-01	4.51E-16	8.57E-01	1.41E-03

Table 13
MPA-OBL Otsu's in terms of FSIM values over all competed algorithms.

Test image	nTh	LSHADE_SPACMA-OBL		CMA_ES-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	7.71E-01	1.15E-03	7.71E-01	3.39E-16	7.71E-01	3.39E-16	7.42E-01	2.78E-02	7.71E-01	2.69E-03	7.71E-01	3.39E-16	7.71E-01	3.38E-16	7.71E-01	3.39E-16
	3	8.15E-01	2.82E-03	8.15E-01	0.00E+00	8.15E-01	0.00E+00	7.73E-01	5.60E-03	8.15E-01	4.35E-03	8.15E-01	1.22E-05	8.15E-01	2.25E-16	8.15E-01	0.00E+00
	4	8.47E-01	3.75E-03	8.48E-01	3.39E-16	8.48E-01	3.39E-16	8.23E-01	5.17E-03	8.42E-01	8.26E-03	8.47E-01	6.24E-04	8.48E-01	5.63E-16	8.48E-01	3.39E-16
	5	8.81E-01	5.38E-03	8.86E-01	5.65E-16	8.86E-01	5.65E-16	8.51E-01	3.38E-03	8.73E-01	6.63E-03	8.85E-01	4.46E-03	8.86E-01	7.89E-16	8.86E-01	5.65E-16
Test2	2	6.99E-01	1.40E-03	6.98E-01	1.13E-16	6.98E-01	1.13E-16	6.42E-01	1.89E-02	6.99E-01	2.13E-03	6.98E-01	1.13E-16	6.98E-01	3.38E-16	6.98E-01	1.13E-16
	3	7.54E-01	4.65E-03	7.54E-01	2.26E-16	7.54E-01	2.26E-16	7.09E-01	9.76E-03	7.54E-01	6.59E-03	7.54E-01	2.26E-16	7.54E-01	2.25E-16	7.54E-01	2.26E-16
	4	8.01E-01	3.04E-03	8.04E-01	3.39E-16	8.04E-01	3.39E-16	7.59E-01	4.88E-03	7.94E-01	1.34E-02	8.04E-01	2.60E-04	8.04E-01	3.38E-16	8.04E-01	3.39E-16
	5	8.28E-01	5.75E-03	8.33E-01	6.00E-04	8.33E-01	8.83E-05	8.04E-01	1.06E-02	8.13E-01	1.68E-02	8.33E-01	6.62E-04	8.33E-01	3.38E-16	8.33E-01	3.39E-16
Test3	2	8.45E-01	8.57E-04	8.45E-01	1.13E-16	8.45E-01	1.13E-16	7.16E-01	1.54E-02	8.46E-01	1.96E-03	8.45E-01	1.13E-16	8.45E-01	1.13E-16	8.45E-01	1.13E-16
	3	8.88E-01	3.29E-03	8.88E-01	5.65E-16	8.88E-01	5.65E-16	8.54E-01	1.01E-02	8.85E-01	8.25E-03	8.88E-01	4.69E-04	8.88E-01	7.89E-16	8.88E-01	5.65E-16
	4	9.23E-01	5.90E-03	9.24E-01	3.39E-16	9.24E-01	3.39E-16	8.95E-01	8.09E-03	9.17E-01	1.52E-02	9.24E-01	5.93E-04	9.24E-01	3.38E-16	9.24E-01	3.39E-16
	5	9.43E-01	6.56E-03	9.40E-01	5.65E-16	9.40E-01	2.70E-04	9.32E-01	7.00E-03	9.35E-01	1.36E-02	9.40E-01	8.42E-04	9.40E-01	5.63E-16	9.40E-01	7.05E-05
Test4	2	7.95E-01	4.40E-04	7.95E-01	2.26E-16	7.95E-01	2.26E-16	7.24E-01	3.13E-02	7.95E-01	3.08E-04	7.95E-01	2.26E-16	7.95E-01	4.51E-16	7.95E-01	2.26E-16
	3	8.73E-01	5.01E-04	8.74E-01	4.52E-16	8.74E-01	4.52E-16	8.16E-01	1.70E-02	8.73E-01	8.28E-04	8.74E-01	1.22E-04	8.74E-01	5.63E-16	8.74E-01	4.52E-16
	4	9.11E-01	1.21E-03	9.11E-01	4.52E-16	9.11E-01	4.52E-16	8.84E-01	9.38E-03	9.09E-01	2.26E-03	9.12E-01	4.12E-04	9.11E-01	5.63E-16	9.11E-01	4.52E-16
	5	9.35E-01	2.36E-03	9.37E-01	3.39E-16	9.37E-01	3.39E-16	9.20E-01	7.40E-03	9.23E-01	9.36E-03	9.37E-01	1.73E-04	9.37E-01	4.51E-16	9.37E-01	3.39E-16
Test5	2	8.11E-01	7.78E-04	8.11E-01	2.26E-16	8.11E-01	2.26E-16	7.42E-01	1.78E-02	8.12E-01	1.21E-03	8.11E-01	2.26E-16	8.11E-01	2.25E-16	8.11E-01	2.26E-16
	3	8.52E-01	3.99E-03	8.50E-01	1.13E-16	8.50E-01	4.65E-04	8.15E-01	1.15E-02	8.52E-01	5.14E-03	8.50E-01	8.80E-04	8.50E-01	1.13E-16	8.50E-01	1.13E-16
	4	8.85E-01	5.29E-03	8.84E-01	3.39E-16	8.84E-01	4.88E-04	8.52E-01	4.80E-03	8.80E-01	1.28E-02	8.85E-01	6.63E-04	8.84E-01	3.38E-16	8.84E-01	3.39E-16
	5	9.13E-01	7.52E-03	9.18E-01	4.52E-16	9.18E-01	3.80E-04	8.82E-01	7.64E-03	8.97E-01	1.35E-02	9.17E-01	8.23E-04	9.18E-01	4.51E-16	9.18E-01	4.52E-16
Test6	2	7.24E-01	5.59E-05	7.24E-01	3.39E-16	7.24E-01	3.39E-16	6.66E-01	1.72E-02	7.24E-01	4.36E-04	7.24E-01	3.39E-16	7.24E-01	3.38E-16	7.24E-01	3.39E-16
	3	7.81E-01	8.69E-04	7.81E-01	3.39E-16	7.81E-01	3.39E-16	7.46E-01	1.78E-02	7.82E-01	2.19E-03	7.81E-01	3.39E-16	7.81E-01	3.38E-16	7.81E-01	3.39E-16
	4	8.22E-01	2.01E-03	8.23E-01	5.65E-16	8.23E-01	5.65E-16	7.89E-01	8.38E-03	8.15E-01	1.16E-02	8.23E-01	1.79E-04	8.23E-01	7.89E-16	8.23E-01	5.65E-16
	5	8.56E-01	3.22E-03	8.58E-01	2.26E-16	8.58E-01	2.26E-16	8.31E-01	7.85E-03	8.42E-01	1.19E-02	8.58E-01	2.81E-04	8.58E-01	2.25E-16	8.58E-01	2.26E-16
Test7	2	7.57E-01	3.67E-04	7.57E-01	4.52E-16	7.57E-01	4.52E-16	6.60E-01	3.42E-02	7.57E-01	7.32E-04	7.57E-01	4.52E-16	7.57E-01	4.51E-16	7.57E-01	4.52E-16
	3	8.29E-01	1.14E-03	8.29E-01	2.26E-16	8.29E-01	2.26E-16	7.76E-01	1.39E-02	8.27E-01	2.90E-03	8.29E-01	1.52E-05	8.29E-01	2.25E-16	8.29E-01	2.26E-16
	4	8.80E-01	3.75E-03	8.83E-01	3.39E-16	8.83E-01	3.39E-16	8.39E-01	7.89E-03	8.73E-01	1.62E-02	8.83E-01	8.29E-05	8.83E-01	3.38E-16	8.83E-01	3.39E-16
	5	9.09E-01	3.63E-03	9.11E-01	5.65E-16	9.11E-01	5.65E-16	8.91E-01	6.84E-03	8.97E-01	1.27E-02	9.12E-01	1.07E-03	9.11E-01	5.63E-16	9.11E-01	5.65E-16
Test8	2	7.58E-01	4.43E-04	7.57E-01	0.00E+00	7.57E-01	0.00E+00	6.73E-01	3.15E-02	7.58E-01	5.13E-04	7.57E-01	0.00E+00	7.57E-01	2.25E-16	7.57E-01	0.00E+00
	3	8.35E-01	2.72E-03	8.35E-01	1.27E-03	8.35E-01	2.26E-16	7.65E-01	1.33E-02	8.29E-01	1.52E-02	8.35E-01	9.75E-05	8.35E-01	4.51E-16	8.35E-01	2.26E-16
	4	8.92E-01	4.58E-03	8.94E-01	0.00E+00	8.94E-01	0.00E+00	8.25E-01	2.76E-02	8.64E-01	3.15E-02	8.93E-01	3.48E-04	8.94E-01	0.00E+00	8.94E-01	0.00E+00
	5	9.19E-01	8.51E-03	9.28E-01	1.13E-16	9.28E-01	5.63E-05	8.92E-01	1.94E-02	8.97E-01	1.19E-02	9.28E-01	3.42E-04	9.28E-01	1.13E-16	9.28E-01	1.13E-16
Test9	2	7.66E-01	3.67E-04	7.66E-01	4.52E-05	7.66E-01	4.56E-05	6.62E-01	2.16E-02	7.57E-01	1.17E-03	7.66E-01	4.47E-05	7.66E-01	4.53E-05	7.66E-01	2.26E-16
	3	8.37E-01	3.00E-03	8.34E-01	8.30E-04	8.34E-01	2.26E-16	7.94E-01	1.46E-02	8.38E-01	4.32E-03	8.34E-01	1.18E-03	8.34E-01	2.25E-16	8.35E-01	2.26E-16
	4	8.88E-01	4.39E-03	8.86E-01	0.00E+00	8.86E-01	0.00E+00	8.50E-01	1.02E-02	8.81E-01	1.09E-02	8.86E-01	1.45E-03	8.86E-01	0.00E+00	8.87E-01	2.26E-16
	5	9.13E-01	4.66E-03	9.15E-01	2.02E-05	9.15E-01	2.02E-05	8.97E-01	6.94E-03	9.07E-01	1.11E-02	9.15E-01	6.30E-04	9.15E-01	2.01E-05	9.06E-01	3.39E-16
Test10	2	7.30E-01	1.07E-03	7.29E-01	3.39E-16	7.29E-01	3.39E-16	6.63E-01	1.18E-02	7.30E-01	1.16E-03	7.29E-01	3.39E-16	7.29E-01	5.63E-16	7.29E-01	3.39E-16
	3	8.02E-01	2.19E-03	8.03E-01	5.65E-16	8.03E-01	5.65E-16	7.33E-01	7.10E-03	8.02E-01	4.35E-03	8.03E-01	5.65E-16	8.03E-01	5.63E-16	8.03E-01	5.65E-16
	4	8.50E-01	4.40E-03	8.50E-01	5.65E-16	8.50E-01	5.65E-16	8.12E-01	8.39E-03	8.28E-01	2.09E-02	8.50E-01	5.01E-04	8.50E-01	7.89E-16	8.50E-01	5.65E-16
	5	8.73E-01	8.60E-03	8.78E-01	0.00E+00	8.78E-01	0.00E+00	8.56E-01	1.55E-02	8.56E-01	1.21E-02	8.78E-01	4.16E-04	8.78E-01	0.00E+00	8.78E-01	0.00E+00

Table 14

The mean of time values for 30 runs obtained with the different algorithms for the segmentation experiment based on Otsu's objective function.

Test image	nTh	LSHADE_SPACMA-OBL		CMA_ES-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	1.4717	0.0353	1.444	0.0752	1.024	0.1037	1.2474	0.3841	0.6711	0.0168	0.7515	0.0194	1.4717	0.0353	0.6455	0.0147
	3	1.7366	0.015	1.5868	0.0412	1.1861	0.0104	1.3358	0.0086	0.807	0.0055	0.8798	0.0056	1.7366	0.015	0.7873	0.025
	4	2.0098	0.0279	1.7411	0.0141	1.374	0.0118	1.5559	0.0069	0.9404	0.0076	1.0137	0.0075	2.0098	0.0279	0.9178	0.0279
	5	2.2832	0.039	1.9667	0.1115	1.5821	0.0132	1.7849	0.0098	1.0832	0.0058	1.1457	0.011	2.2832	0.039	1.0553	0.0154
	5	2.4631	0.0151	1.3654	0.0189	0.9873	0.0044	1.0997	0.0066	0.6648	0.0073	0.7524	0.0056	1.4631	0.0151	0.648	0.0124
Test2	3	1.7398	0.0126	1.5005	0.0182	1.1779	0.017	1.3351	0.0165	0.8086	0.0139	0.8804	0.0045	1.7398	0.0126	0.7741	0.0228
	4	2.014	0.0324	1.7122	0.0369	1.3957	0.0565	1.5644	0.0217	0.9786	0.0659	1.0125	0.0118	2.014	0.0324	0.8945	0.006
	5	2.2767	0.0191	1.9142	0.1475	1.5794	0.0608	1.7816	0.0064	1.109	0.0191	1.1471	0.0182	2.2767	0.0191	1.0544	0.0671
	2	1.4725	0.03	1.3631	0.0244	0.9778	0.0209	1.1162	0.0223	0.6716	0.0112	0.7584	0.0181	1.4725	0.03	0.6464	0.0106
	3	1.753	0.0159	1.5702	0.0395	1.1714	0.0087	1.3586	0.0236	0.8235	0.0189	0.8867	0.0131	1.753	0.0159	0.7724	0.0065
Test3	4	2.017	0.0431	1.7552	0.0255	1.3737	0.0245	1.5787	0.0715	0.9406	0.0076	1.0195	0.0074	2.017	0.0431	0.9066	0.0167
	5	2.3084	0.0425	1.958	0.0315	1.5814	0.054	1.7914	0.0079	1.0777	0.0126	1.1456	0.012	2.3084	0.0425	1.0372	0.0145
	2	1.4679	0.015	1.3602	0.0876	0.9749	0.0054	1.1295	0.0325	0.6562	0.0045	0.7456	0.0063	1.5814	0.054	0.6354	0.0033
	3	1.753	0.033	1.5165	0.0188	1.2027	0.0062	1.3499	0.0457	0.7997	0.0052	0.8789	0.0061	0.9749	0.0054	0.7693	0.0054
	4	2.0309	0.0795	1.7023	0.0146	1.38	0.0211	1.572	0.0136	0.9355	0.0061	1.0106	0.0071	1.2027	0.0062	0.926	0.0655
Test5	5	2.2861	0.0471	1.9034	0.0407	1.5715	0.019	1.8135	0.0361	1.0768	0.0093	1.1393	0.0104	1.38	0.0211	1.0416	0.0208
	2	1.4669	0.0125	1.377	0.019	0.9821	0.0278	1.1024	0.0092	0.6589	0.007	0.7508	0.005	1.5715	0.019	0.6341	0.0048
	3	1.7428	0.0185	1.5732	0.0692	1.1851	0.033	1.3428	0.0056	0.7986	0.0049	0.886	0.0173	0.9821	0.0278	0.766	0.0053
	4	2.0156	0.0125	1.753	0.0152	1.4105	0.0384	1.5656	0.0095	0.9371	0.0059	1.0164	0.014	1.1851	0.033	0.8951	0.0051
	5	2.2803	0.0219	1.969	0.1355	1.5672	0.0204	1.8017	0.0546	1.0823	0.0438	1.1519	0.0244	1.4105	0.0384	1.0283	0.0067
Test6	2	1.4689	0.0113	1.3828	0.0164	0.9765	0.0351	1.1181	0.0052	0.6611	0.0042	0.7556	0.0062	1.5672	0.0204	0.6397	0.0418
	3	1.751	0.039	1.5752	0.0326	1.173	0.0124	1.3451	0.0093	0.8007	0.0052	0.8831	0.0213	0.9765	0.0351	0.7681	0.006
	4	2.0119	0.0204	1.7767	0.0737	1.4015	0.0205	1.5641	0.0049	0.9414	0.0198	1.0197	0.0308	1.173	0.0124	0.9064	0.0239
	5	2.2975	0.0754	1.9608	0.0326	1.5668	0.0112	1.7853	0.0125	1.0809	0.0157	1.1474	0.0161	1.4015	0.0205	1.0344	0.0072
	2	1.4664	0.0206	1.3787	0.0149	0.9692	0.0091	1.102	0.005	0.6571	0.0043	0.7475	0.0069	1.5668	0.0112	0.6351	0.0069
Test7	3	1.7464	0.0166	1.5734	0.0261	1.1919	0.0309	1.3354	0.0133	0.8036	0.0157	0.8941	0.0186	1.3354	0.0133	0.7716	0.0053
	4	2.0148	0.0379	1.7627	0.0208	1.3897	0.0322	1.5762	0.0606	0.9407	0.0097	1.0125	0.0075	1.5762	0.0606	0.9314	0.0308
	5	2.2888	0.0113	1.9627	0.0686	1.5884	0.0433	1.7878	0.0225	1.0808	0.0063	1.1419	0.0079	1.7878	0.0225	1.0428	0.0376
	2	1.4626	0.0165	1.3263	0.0166	0.9921	0.0103	1.1152	0.0073	0.6599	0.005	0.7478	0.0071	1.1152	0.0073	0.6344	0.0053
	3	1.7458	0.0134	1.5424	0.0312	1.1767	0.0274	1.3385	0.0558	0.8035	0.0062	0.8841	0.0065	1.3385	0.0558	0.7725	0.0084
Test8	4	2.009	0.0203	1.7714	0.038	1.3872	0.0098	1.5625	0.0182	0.9448	0.0121	1.0136	0.0088	1.5625	0.0182	0.9	0.0074
	5	2.2872	0.0757	1.9491	0.0293	1.5641	0.0201	1.7805	0.0266	1.0923	0.0349	1.1455	0.0087	1.7805	0.0266	1.0452	0.0344
	2	1.4651	0.0155	1.3898	0.0433	0.9915	0.0429	1.125	0.008	0.6603	0.0047	0.7538	0.0331	1.125	0.008	0.6327	0.0066
	3	1.7511	0.0529	1.5735	0.0226	1.2139	0.0235	1.3284	0.0137	0.7965	0.0053	0.9048	0.0488	1.3284	0.0137	0.7704	0.0054
	4	2.0128	0.0194	1.5301	0.1939	1.3753	0.0361	1.5604	0.0065	0.9456	0.0509	1.0208	0.0476	1.5604	0.0065	0.9012	0.0088
Test10	5	2.2782	0.0295	1.4904	0.0189	1.6046	0.0236	1.7917	0.013	1.0855	0.0177	1.1488	0.0175	1.0125	0.0075	1.034	0.0098
	2	1.4588	0.0162	1.1273	0.0269	0.9665	0.0078	1.1178	0.0046	0.667	0.0157	0.7489	0.006	1.1419	0.0079	0.6343	0.0118
	3	1.7491	0.0422	1.2412	0.0338	1.1766	0.0291	1.3293	0.0076	0.8176	0.0241	0.8766	0.0067	0.7478	0.0071	0.7684	0.0066
	4	2.0051	0.0121	1.3843	0.0426	1.3682	0.007	1.5605	0.0069	0.94	0.0068	1.017	0.0084	0.8841	0.0065	0.8987	0.0086
	5	2.284	0.0562	1.5307	0.0461	1.5754	0.0395	1.7852	0.0362	1.0768	0.0117	1.1432	0.0102	1.0136	0.0088	1.0502	0.0619

Friedman mean rank test. is also a non-parametric test that is used to compare three or more matched groups, also check the performance of the competitive algorithms. Friedman's mean rank is used to specify the overall rank among the competitor algorithms [64]. The Friedman statistic requires calculating the mean ranked value. Then, a comparison is needed to review the critical values obtained for the considered significance level with Friedman statistics to see whether the null hypothesis is rejected or not. The relative results for Otsu's methods are shown in Table 10, whereas Table 20 reports statistical significance of Kapur's method.

5.5. Results analysis of MPA-OBL based Otsu objective function

This section presents and discusses the numerical results of the proposed algorithm MPA-OBL results, where Otsu's between-class variance Eq. (17) is employed as an objective function. Tables 6–8, show the segmented images obtained from the proposed algorithm MPA-OBL with a different number of thresholds [nTh = 2, 3, 4, 5] for all the test benchmark images used in the experiments. Results also include the distribution of the best-selected thresholding values over the histogram related to each tested image. Table 9 shows the best thresholding values selected by the proposed algorithm and all other competed algorithms, while Tables 10–14 contain the mean and standard deviation results of fitness also the PSNR, SSIM, FSIM, and time, respectively. The higher the mean value of an algorithm, the more accurate and efficient one. On the other hand, the less value in terms of the standard deviation of an algorithm is the more stable one.

According to Table 10, in terms of the mean values of fitness results indicate that the proposed MPA-OBL and the CMA_ES-OBL algorithms is in the first rank by 35 higher fitness values over the 40 experiments, the DE-OBL algorithm is in the second rank by 32 higher fitness values, and the SSA-OBL in the third rank with 13 higher fitness cases while the rest of the algorithms gain no higher fitness cases.

From PSNR results shown in Table 11 in terms of the mean values related to PSNR, it is noticed that the proposed MPA-OBL in the first rank by 13 higher cases, while CMA_ES-OBL comes in the second rank by 12 higher cases, SCA-OBL and MPA algorithms came in the third rank with 11 higher cases. Moreover, DE-OBL came in the fourth rank by 9 cases, LSHADE_SPACMA-OBL and SSA-OBL in the fifth rank by 8 cases, HHO-OBL in the last rank with no higher cases.

In Table 12, in terms of mean results of SSIM, MPA-OBL and SCA-OBL algorithms come in the first rank by 11 higher SSIM cases, next LSHADE_SPACMA-OBL, CMA_ES-OBL come in the second rank by 10 higher SSIM cases, then SSA-OBL in the third rank by 8 higher SSIM cases, SSA come in the fourth rank by 7 higher SSIM cases, DE-OBL in the fifth rank by 6 higher SSIM cases, while HHO-OBL comes in the last rank with no higher SSIM cases.

Table 13 reports mean values that show that SSA-OBL come in the first rank with 13 higher FSIM cases. Moreover, MPA-OBL, MPA, and LSHADE_SPACMA-OBL come in the second rank by 10 higher SSIM cases. CMA_ES-OBL and SCA-OBL take the third-ranking place by 9 higher SSIM cases, while DE-OBL comes in the fourth place by 7 higher SSIM cases, finally HHO-OBL in the last place with no higher SSIM cases.

According to Wilcoxon rank-sum test, Table 15, show the rank-sum results for fitness, using the Otsu's method. While applying the rank-sum test between the proposed algorithm MPA-OBL and each of the following algorithms (LSHADE_SPACMA-OBL, CMA_ES-OBL, DE-OBL, HHO-OBL, SCA-OBL, SSA-OBL, and the original MPA) it is noticed there is a difference of LSHADE_SPACMA-OBL, HHO-OBL, SCA-OBL, and the original MPA algorithm in contrast to MPA-OBL; this means the proposed algorithm MPA-OBL has a significant development, but in comparison to the CMA_ES-OBL and DE-OBL it shows that they have the similar behavior which is shown in tables when (P > 0.05) or NaN values.

Table 15

Comparison of P and H values obtained from the Wilcoxon signed-rank test between the pairs of the proposed MPA-OBL vs LSHADE_SPACMA-OBL, MPA-OBL vs CMA_ES-OBL, MPA-OBL vs DE-OBL, MPA-OBL vs HHO-OBL, MPA-OBL vs SCA-OBL, MPA-OBL vs SSA-OBL, and MPA-OBL vs MPA for Otsu's method in terms of fitness results.

Test image	nTh	LSHADE_SPACMA-OBL		CMA_ES-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA	
		p	h	p	h	p	h	p	h	p	h	p	h	p	h
Test1	2	0.000660315	1	NaN	0	NaN	0	1.69E-14	1	1.51E-09	1	NaN	0	1.69E-14	1
	3	4.57E-12	1	NaN	0	NaN	0	2.06E-13	1	1.21E-12	1	0.333710696	0	2.06E-13	1
	4	1.21E-12	1	NaN	0	NaN	0	1.03E-12	1	1.21E-12	1	0.005554612	1	1.03E-12	1
	5	1.21E-12	1	NaN	0	NaN	0	1.20E-12	1	1.21E-12	1	8.75E-07	1	1.20E-12	1
Test2	2	5.60E-09	1	NaN	0	NaN	0	1.69E-14	1	1.67E-08	1	NaN	0	5.60E-09	1
	3	4.57E-12	1	NaN	0	NaN	0	1.69E-14	1	1.21E-12	1	NaN	0	4.57E-12	1
	4	1.21E-12	1	NaN	0	NaN	0	8.73E-14	1	1.21E-12	1	0.00269767	1	1.21E-12	1
	5	1.21E-12	1	2.17E-07	1	0.081493283	0	9.22E-13	1	1.21E-12	1	5.54E-11	1	1.21E-12	1
Test3	2	6.59E-05	1	NaN	0	NaN	0	1.69E-14	1	8.58E-07	1	NaN	0	8.58E-07	1
	3	1.21E-12	1	NaN	0	NaN	0	1.69E-14	1	1.21E-12	1	0.160741998	0	1.21E-12	1
	4	1.21E-12	1	NaN	0	NaN	0	6.13E-14	1	1.21E-12	1	0.000304818	1	1.21E-12	1
	5	1.72E-12	1	0.333710696	0	0.304901788	0	1.55E-12	1	1.72E-12	1	3.73E-11	1	1.72E-12	1
Test4	2	2.91E-05	1	NaN	0	NaN	0	1.69E-14	1	6.43E-05	1	NaN	0	NaN	0
	3	1.21E-12	1	NaN	0	NaN	0	1.69E-14	1	1.21E-12	1	0.333710696	0	0.333710696	0
	4	1.21E-12	1	NaN	0	NaN	0	8.24E-13	1	1.21E-12	1	1.62E-08	1	NaN	0
	5	1.21E-12	1	NaN	0	NaN	0	1.09E-12	1	1.21E-12	1	1.23E-05	1	NaN	0
Test5	2	3.41E-07	1	NaN	0	NaN	0	1.69E-14	1	6.03E-10	1	NaN	0	3.41E-07	1
	3	1.21E-12	1	NaN	0	0.333710696	0	1.55E-13	1	1.21E-12	1	0.04177393	1	1.21E-12	1
	4	1.21E-12	1	NaN	0	0.333710696	0	4.52E-13	1	1.21E-12	1	0.081522972	0	1.21E-12	1
	5	1.21E-12	1	NaN	0	0.333710696	0	1.12E-12	1	1.21E-12	1	6.42E-05	1	1.21E-12	1
Test6	2	7.28E-07	1	NaN	0	NaN	0	1.69E-14	1	1.48E-08	1	NaN	0	7.28E-07	1
	3	1.21E-12	1	NaN	0	NaN	0	4.27E-13	1	1.21E-12	1	NaN	0	1.21E-12	1
	4	1.21E-12	1	NaN	0	NaN	0	7.39E-13	1	1.21E-12	1	0.005510225	1	1.21E-12	1
	5	1.21E-12	1	NaN	0	NaN	0	1.20E-12	1	1.21E-12	1	0.000144756	1	1.21E-12	1
Test7	2	3.42E-07	1	NaN	0	NaN	0	1.69E-14	1	1.28E-07	1	NaN	0	1.69E-14	1
	3	1.21E-12	1	NaN	0	NaN	0	2.01E-13	1	1.21E-12	1	0.333710696	0	2.01E-13	1
	4	1.21E-12	1	NaN	0	NaN	0	6.60E-13	1	1.21E-12	1	0.160741998	0	6.60E-13	1
	5	1.21E-12	1	NaN	0	NaN	0	1.16E-12	1	1.21E-12	1	1.67E-08	1	1.16E-12	1
Test8	2	1.30E-07	1	NaN	0	NaN	0	1.69E-14	1	4.40E-08	1	NaN	0	1.30E-07	1
	3	1.21E-12	1	0.333710696	0	NaN	0	1.69E-14	1	1.21E-12	1	0.333710696	0	1.21E-12	1
	4	1.21E-12	1	NaN	0	NaN	0	1.20E-12	1	1.21E-12	1	0.005539794	1	1.21E-12	1
	5	1.21E-12	1	NaN	0	0.333710696	0	1.21E-12	1	1.21E-12	1	4.61E-08	1	1.21E-12	1
Test9	2	6.74E-13	1	1.69E-14	1	1.69E-14	1	1.69E-14	1	5.38E-11	1	1.69E-14	1	6.74E-13	1
	3	1.21E-12	1	1.20E-13	1	1.69E-14	1	4.16E-14	1	1.21E-12	1	6.12E-14	1	1.21E-12	1
	4	1.21E-12	1	1.69E-14	1	1.69E-14	1	1.02E-12	1	1.21E-12	1	5.94E-13	1	1.02E-12	1
	5	1.21E-12	1	1.69E-14	1	1.69E-14	1	1.16E-12	1	1.21E-12	1	5.82E-13	1	1.16E-12	1
Test10	2	5.65E-09	1	NaN	0	NaN	0	1.69E-14	1	4.50E-08	1	NaN	0	1.69E-14	1
	3	4.57E-12	1	NaN	0	NaN	0	1.69E-14	1	1.21E-12	1	NaN	0	1.69E-14	1
	4	1.21E-12	1	NaN	0	NaN	0	1.59E-13	1	1.21E-12	1	0.001305464	1	1.59E-13	1
	5	1.21E-12	1	NaN	0	NaN	0	1.14E-12	1	1.21E-12	1	8.54E-07	1	1.14E-12	1

Based on the Friedman mean rank, it is clear in the last row of Table 10. It can be noticed that the proposed algorithm MPA-OBL is placed as the best performing method (ranked first) among all other algorithms used in the comparison with Friedman mean rank value of 5.33, while SSA-OBL takes the second rank with a mean rank value of 5.32, CMA_ES-OBL in the third rank with a mean rank value of 5.21, in the fourth rank MPA takes place with a mean value of 5.15, DE-OBL is the fifth rank with 5.12, then LSHADE_SPACMA-OBL takes place with 4.92 mean value, while SCA-OBL comes in the seventh rank by a mean value of 3.92, finally HHO-OBL in the last rank with a mean value of 1.03.

Fig. 7 summarizes the results of segmentation experiment in terms of fitness, PSNR, SSIM, and SSIM based on Otsu's objective function.

5.6. Results analysis of MPA-OBL based Kapur objective function

This section presents and discusses the MPA-OBL proposed approach results based on maximizing the Kapur objective function. Tables 16–18, show the segmented images of all benchmark images used in the experiments each benchmark image include four thresholding number [nTh = 2, 3, 4, 5], also these tables show the distribution of the best selected thresholding values over the histogram of each test image with all thresholding values. While Table 19 presents the best of thresholding values obtained. Also,

Tables 20–24 show the mean and standard deviation of Fitness, PSNR, SSIM, FSIM, and time respectively.

From fitness results Table 20, in terms of mean, it is noticed that the proposed algorithm MPA-OBL comes in the first rank with 40 higher fitness values over the 40 experiments, the CMA_ES-OBL is in the second rank by won 6 higher cases, but SSA-OBL comes in the third rank by 4 higher cases, while the rest of algorithms gain zero higher values.

Table 21 provides the mean and standard deviation of PSNR. In terms of mean results it is noticeable that the MPA-OBL comes in the first rank by providing 15 cases of the higher PSNR values, in the second rank DE-OBL comes with 10 cases only, HHO-OBL in the third rank by 8 higher PSNR cases, LSHADE_SPACMA-OBL and SSA-OBL only produces 6 cases and comes in the fourth rank, while CMA_ES-OBL and SCA-OBL come in the fifth rank by only one higher case. Finally, the original MPA algorithm comes in the last rank with zero higher PSNR values.

Table 22 presents the mean of SSIM, from results of mean values it is noticed that HHO-OBL comes in the first rank with 13 higher cases of SSIM, then the MPA-OBL comes with 11 cases in the second rank, CMA_ES-OBL comes in the third rank by just 6 higher cases, while MPA comes in the fourth rank by 4 higher SSIM values, SSA-OBL comes in the fifth rank by 3 cases only, SCA-OBL in the sixth rank with 2 higher cases. Finally, LSHADE_SPACMA-OBL has only one higher cases of SSIM.

Table 16
Results after applying MPA-OBL on Kapur to the set of benchmark images.





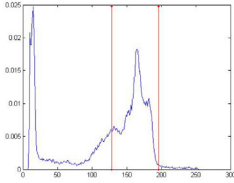
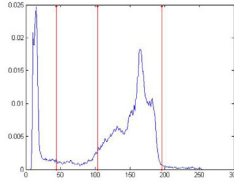
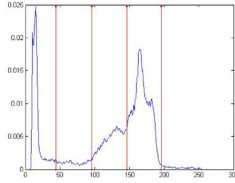
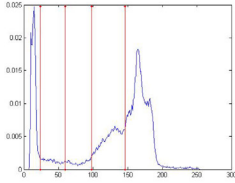




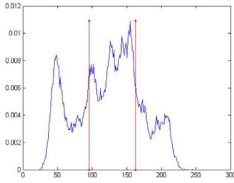
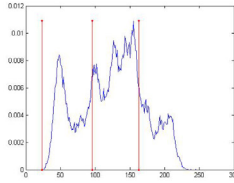
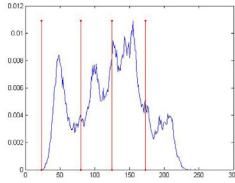
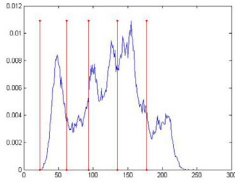
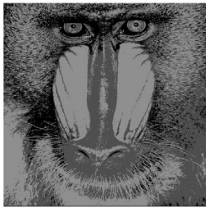
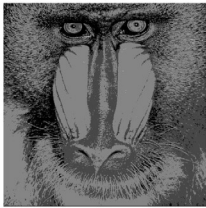
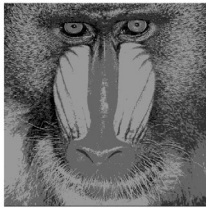
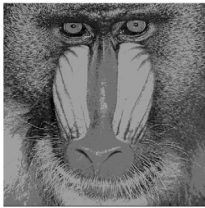
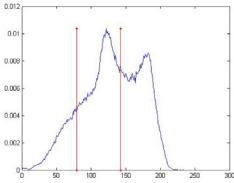
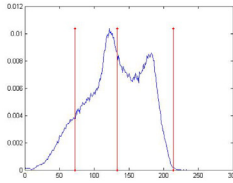
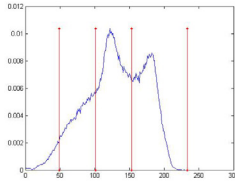
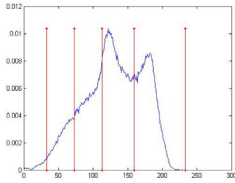
Image	nTh = 2	nTh = 3	nTh = 4	nTh = 5
				
Cameraman				
				
Lena				
				
Baboon				






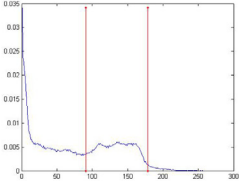
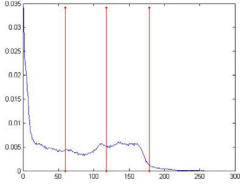
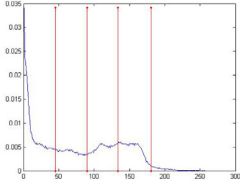
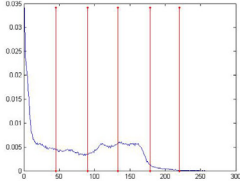





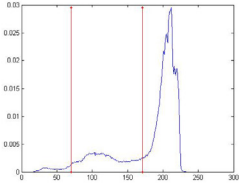
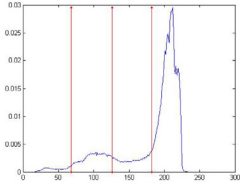
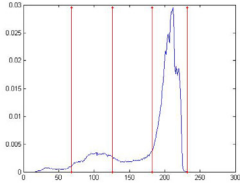
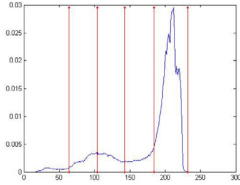





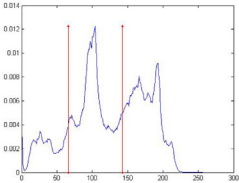
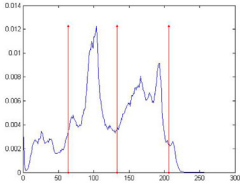
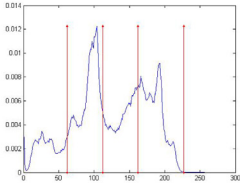
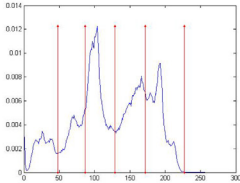
Table 23 presents the mean and standard deviation of FSIM. In terms of mean, results indicate that MPA-OBL comes in the first rank by 13 higher cases, then DE-OBL comes in the second rank by 9 cases, while LSHADE_SPACMA-OBL comes with 6 cases in the third rank, HHO-OBL comes in the fourth rank by 5 cases, while SCA-OBL and MPA comes in the fifth rank with 3 higher cases. At the last rank SSA-OBL gain no higher cases.

Meanwhile the following Table 25 show the Wilcoxon rank-sum test results applied on fitness while maximizing the Kapur's fitness function, there is a noticeable difference between all competing algorithms in contrast to MPA-OBL which indicates that the proposed algorithm has a significant development.

Finally, Friedman mean rank test results presented in the last two rows in Table 20, we can notice that the proposed algorithm MPA-OBL comes in the first rank with a mean value of 5.88, while SSA-OBL comes in the second rank with a mean value of 5.74, DE-OBL takes the third place in the ranking with a mean value of 5.71, HHO-OBL in the fourth-ranking order with a mean value of 2.85, while the original MPA takes the fifth place with 4.44 mean value, while CMA_ES-OBL in the sixth place with a mean value of 3.33, LSHADE_SPACMA-OBL in the seventh place with mean value of 3.15. Finally, SCA-OBL in the last rank with a mean value of 2.85.

Fig. 8 presents the results of segmentation experiment in terms of fitness, PSNR, SSIM, and SSIM based on Kapur's objective function.

Table 17
Results after applying MPA-OBL on Kapur to the set of benchmark images.

Image	nTh = 2	nTh = 3	nTh = 4	nTh = 5
				
Hunter				
				
Airplane				
				
Pepper				

Overall, in both Otsu and Kapur's objective functions the segmentation effect of MPA-OBL represents the dominance of the proposed MPA-OBL comparing to other algorithms.

5.7. The pros and cons of the MPA-OBL algorithm

This section mentions some of the advantages and drawbacks pertaining to proposed method MPA-OBL. Regarding the main advantages of the proposed algorithm, it can be concluded in its simplicity to understand and implementation. The experimental

results reveal that it performs well on intensity image segmentation problems and gives competitive results compared to the counterpart algorithms, the quality of the segmented images give good results. Besides, the MPA-OBL produces high convergence speed which indicates that the proposed algorithm avoids trapping in local optima and successfully balances between exploration and exploitation phases, due to fast determination of threshold values, along with high accuracy presented in the results. The main drawbacks of the proposed algorithm are that MPA-OBL uses only the intensity information of the image to perform segmentation tasks resulting in the sensitivity to noise

Table 18
Results after applying MPA-OBL on Kapur to the set of benchmark images.





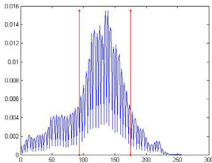
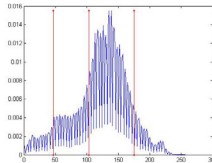
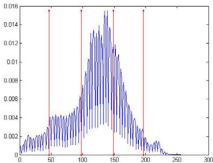
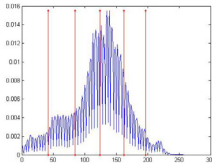




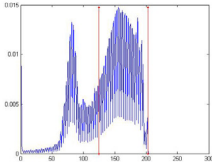
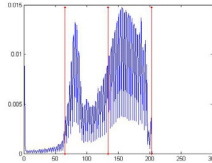
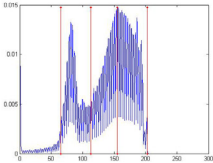
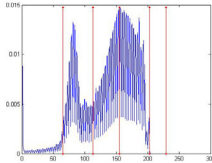




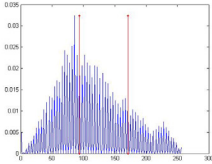
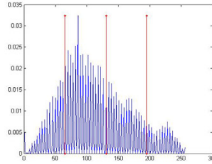
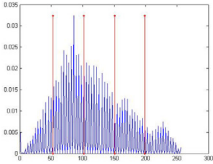
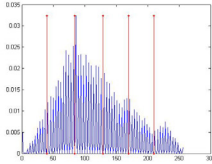




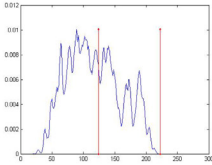
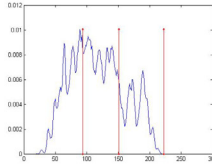
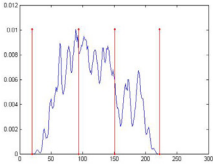
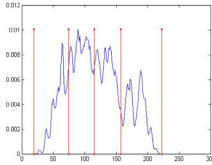
Image	nTh = 2	nTh = 3	nTh = 4	nTh = 5
Living-Room				
				
Woman				
				
Bridge				
				
Butter-Fly				
				

Table 19
Kapur's best threshold values.

Test image	nTh	LSHADE_SPACMA-OBL	CMA_ES-OBL	DE-OBL	HHO-OBL	SCA-OBL	SSA-OBL	MPA	MPA-OBL
Cameraman	2	128 196	128 196	128 196	127 195	128 196	123 190	128 196	128 196
	3	44 103 196	44 103 196	63 118 196	43 104 195	45 104 196	44 103 196	44 103 196	44 103 196
	4	42 96 145 196	44 96 146 196	42 96 145 196	38 93 143 194	43 96 145 196	43 96 146 196	44 96 146 196	44 96 146 196
	5	22 59 98 145 196	24 60 98 146 196	26 63 99 146 196	25 60 100 144 197	27 67 106 154 199	25 63 101 149 197	24 60 98 146 196	24 60 98 146 196
	2	96 163	96 163	96 163	95 163	96 163	96 163	96 163	96 163
Lena	3	23 96 163	47 108 167	54 111 168	29 101 164	31 100 164	68 119 171	23 96 163	23 96 163
	4	23 80 125 173	23 79 124 173	24 81 125 173	22 75 121 170	23 80 125 173	37 85 128 174	23 80 125 173	23 80 125 173
	5	23 62 95 135 177	23 65 99 139 179	23 70 108 146 181	21 67 104 140 180	23 68 104 143 181	32 70 104 143 181	23 62 94 135 177	23 62 94 135 177
	2	79 143	79 143	79 143	79 143	79 143	79 143	79 143	79 143
	3	58 116 185	50 103 157	79 143 233	47 100 152	55 110 171	49 101 153	72 133 214	72 133 214
Baboon	4	46 99 152 233	40 87 133 197	48 100 152 233	37 87 139 214	42 90 138 205	34 77 119 171	49 101 153 233	49 101 153 233
	5	32 72 112 159 233	33 72 112 157 227	33 73 113 159 233	31 68 112 163 228	33 73 112 155 220	31 68 107 147 201	33 73 113 159 233	33 73 113 159 233
	2	91 179	91 179	91 179	90 178	91 179	91 179	91 179	91 179
	3	59 118 179	60 118 179	60 118 179	58 116 179	60 118 179	60 118 179	60 118 179	60 118 179
	4	44 90 133 180	44 89 133 180	44 90 134 181	43 87 132 180	45 90 134 181	44 90 134 181	45 90 134 181	45 90 134 181
Hunter	5	40 85 130 178 218	42 86 128 172 213	44 90 133 179 220	36 77 117 162 202	38 78 118 159 199	35 74 114 155 195	45 90 133 179 220	45 90 133 179 220
	2	70 171	70 171	70 171	70 171	70 171	70 171	70 171	70 171
	3	68 126 182	68 126 182	68 127 183	68 126 180	68 126 182	68 126 182	68 126 182	68 126 182
	4	67 125 181 232	66 116 165 211	63 121 177 227	64 116 165 215	67 122 175 223	66 115 164 210	68 126 182 232	68 126 182 232
	5	42 86 134 182 232	59 98 138 179 225	48 93 137 181 229	26 71 117 163 213	53 94 134 176 222	51 87 127 171 217	63 103 142 184 232	64 104 143 184 232
Pepper	2	66 143	66 143	66 143	66 142	66 143	66 143	66 143	66 143
	3	62 116 171	61 111 161	63 120 180	62 115 167	62 115 169	61 112 162	65 135 212	64 133 206
	4	61 111 161 227	61 111 161 225	61 111 161 227	60 108 161 225	61 111 160 225	57 104 156 222	62 112 162 227	62 112 162 227
	5	45 84 126 170 227	48 86 128 172 227	52 94 136 179 227	42 81 125 172 226	51 91 132 175 227	42 83 125 170 227	48 87 129 172 227	48 87 129 172 227
	2	94 175	92 173	93 174	90 173	93 174	93 174	94 175	94 175
Living-Room	3	47 103 175	47 103 175	47 103 175	48 104 174	47 103 175	47 103 175	47 103 175	47 103 175
	4	46 98 149 197	46 97 148 195	47 100 152 197	46 96 144 191	47 99 150 196	46 97 148 196	47 98 149 197	47 98 149 197
	5	41 83 121 161 197	41 84 123 162 197	43 86 126 165 202	35 73 112 155 195	43 86 124 163 199	39 79 116 159 197	42 85 124 162 197	42 85 124 162 197
	2	125 203	125 203	125 203	125 203	125 203	108 189	125 203	125 203
	3	65 133 203	65 134 203	72 136 203	65 131 205	65 134 203	65 134 203	65 134 203	65 134 203
Woman	4	65 112 154 203	65 113 155 203	65 115 157 203	63 111 155 204	65 113 155 203	65 112 154 203	65 113 155 203	65 113 155 203
	5	64 102 141 184 216	65 102 139 178 212	64 108 147 188 218	52 93 128 165 209	64 105 142 182 215	61 94 132 172 209	65 113 155 203 229	65 113 155 203 229
	2	94 171	95 172	94 171	96 171	93 172	94 171	94 171	94 171
	3	65 130 193	65 130 193	64 130 193	63 128 190	65 129 192	65 130 194	65 131 195	65 131 195
	4	53 100 148 198	53 102 149 199	52 101 149 199	51 100 148 199	53 100 149 199	52 100 148 198	53 102 151 199	53 102 151 199
Bridge	5	40 80 123 168 207	41 82 125 167 207	43 86 128 170 209	41 82 125 165 209	43 85 128 168 208	38 77 119 161 203	40 84 130 171 211	40 84 129 170 210
	2	124 222	122 217	124 222	124 222	124 222	97 159	124 222	124 222
	3	93 150 222	91 149 219	94 151 222	87 144 222	94 151 222	85 139 203	94 151 222	94 151 222
	4	58 108 155 222	72 114 156 222	58 108 154 220	47 101 155 223	47 115 157 222	59 106 150 212	19 94 151 222	20 94 151 222
	5	31 81 121 161 222	36 85 124 163 222	53 96 134 170 222	21 69 111 159 222	42 87 126 166 219	37 82 122 163 222	19 74 115 157 222	19 74 115 157 222

and inefficient for advanced image segmentation problems. The experimental data-set not efficient enough to measure the performance of the segmentation process. The proposed algorithm is simple but it has a disadvantage that it is computationally expensive. Results in terms of standard deviation are not good enough to compete with the other algorithms.

6. Conclusion

This paper introduces a new version of an optimization algorithm namely marine predator algorithm (MPA) that suffers from its poor convergence and local optima trapping. The proposed improved version is called MPA-OBL, and it came from the migration of the original MPA with an efficient local search called opposite-based learning (OBL) to enhance the exploitation phase and avoid local optima. The verification process to evaluate the performance of the proposed MPA-OBL using CEC'2020 benchmark functions revealed that the proposed algorithm can outperform other methods of different quality and statistical measures. On the other hand, segmentation is considered as a crucial step that should be performed in right manner for effective image analysis. One of the most important and effective approaches for image segmentation is thresholding. So, the proposed MPA-OBL is used to identify the optimal thresholding values for a set of gray-level benchmark images. The performance of the proposed algorithm MPA-OBL is compared with seven meta-heuristic algorithms namely LSHADE_SPACMA-OBL, CMA_ES-OBL, DE-OBL, HHO-OBL, SCA-OBL, SSA-OBL, and the original MPA. The comparison is performed by applying the algorithms on a set of benchmark images with threshold numbers *nTh* from 2 to 5. The results obtained by each algorithm in terms of Fitness, matrices used to test the segmented image quality such as PSNR, SSIM, FSIM. Also, Friedman and Wilcoxon tests statistically revealed that our proposed algorithm MPA-OBL has significant performance upgradation and outperforms competitive methods in terms of fitness in case of using Otsu or Kapur objective function. Moreover, it produces good results in terms of PSNR, SSIM, FSIM comparing with the other competed algorithms. While the Wilcoxon result indicates that our proposed method produces an improvement in

performance comparing with all other selected algorithms, also from Friedman test our proposed method comes in the first rank while in case of using Otsu or Kapur as an objective function. In future, we will: (a) apply the proposed MPA-OBL on additional real world problems. (b) extend the tested images data-set and increase the number of thresholds to attain more reliable results, (c) while the proposed algorithm could not outperform some algorithms in terms of standard deviation results, we will seek to hybridize the proposed algorithm with another meta-heuristic optimization algorithms to improve the segmentation results, (d) we will test the proposed algorithm performance on solving additional types of problems.

CRedit authorship contribution statement

Essam H. Houssein: Supervision, Methodology, Conceptualization, Software, Formal analysis, Writing – review & editing. **Kashif Hussain:** Methodology, Formal analysis, Resources, Writing – original draft, Writing – review & editing. **Laith Abualigah:** Methodology, Formal analysis, Writing – review & editing. **Mohamed Abd Elaziz:** Methodology, Formal analysis, Writing – review & editing. **Waleed Alomoush:** Methodology, Formal analysis, Writing – review & editing. **Gaurav Dhiman:** Data curation, Formal analysis, Writing – review & editing. **Youcef Djenouri:** Methodology, Formal analysis, Writing – review & editing. **Erik Cuevas:** Supervision, Formal analysis, Writing – review & editing. All authors read and approved the final paper..

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

No funding was received for this work.

Table 20
MPA-OBL Kapur's in terms of Fitness values over all competed algorithms.

Test image	nTh	LSHADE_SPACMA-OBL		CMA-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	1.75E+01	5.54E-02	1.73E+01	1.51E-01	1.76E+01	7.23E-15	1.76E+01	6.18E-03	1.76E+01	7.09E-03	1.76E+01	1.08E-14	1.76E+01	3.60E-15	1.76E+01	1.08E-14
	3	2.19E+01	6.27E-02	2.18E+01	1.20E-01	2.20E+01	1.45E-14	2.19E+01	7.13E-02	2.20E+01	3.04E-02	2.20E+01	1.97E-02	2.20E+01	3.60E-15	2.20E+01	3.61E-15
	4	2.64E+01	9.25E-02	2.60E+01	2.37E-01	2.66E+01	1.18E-13	2.65E+01	1.49E-01	2.64E+01	6.56E-02	2.66E+01	2.64E-03	2.66E+01	1.08E-14	2.66E+01	1.08E-14
Test2	5	3.03E+01	1.04E-01	2.98E+01	2.53E-01	3.06E+01	1.65E-13	3.04E+01	1.22E-01	3.02E+01	1.39E-01	3.05E+01	6.73E-02	3.05E+01	3.60E-15	3.06E+01	1.08E-14
	2	1.78E+01	1.40E-03	1.78E+01	3.28E-02	1.78E+01	0.00E+00	1.78E+01	1.00E-03	1.78E+01	2.73E-03	1.78E+01	2.17E-04	1.78E+01	3.60E-15	1.78E+01	3.61E-15
	3	2.22E+01	8.25E-02	2.20E+01	8.52E-02	2.22E+01	5.25E-02	2.21E+01	8.57E-02	2.23E+01	5.07E-02	2.23E+01	7.21E-02	2.23E+01	3.60E-15	2.23E+01	3.61E-15
Test3	4	2.64E+01	1.87E-01	2.60E+01	2.23E-01	2.65E+01	9.32E-02	2.63E+01	3.20E-01	2.64E+01	9.09E-02	2.66E+01	2.83E-03	2.66E+01	1.44E-14	2.66E+01	7.23E-15
	5	3.03E+01	1.13E-01	2.97E+01	2.82E-01	3.04E+01	3.28E-03	3.01E+01	3.87E-01	3.01E+01	1.80E-01	3.05E+01	1.22E-02	3.05E+01	1.80E-14	3.05E+01	2.17E-14
	2	1.76E+01	6.24E-04	1.76E+01	1.53E-02	1.76E+01	0.00E+00	1.76E+01	1.14E-03	1.76E+01	1.39E-03	1.76E+01	2.07E-13	1.76E+01	3.60E-15	1.76E+01	3.61E-15
Test4	3	2.21E+01	7.96E-03	2.20E+01	4.61E-02	2.21E+01	1.08E-14	2.21E+01	3.78E-02	2.21E+01	2.37E-02	2.21E+01	2.98E-02	2.21E+01	4.15E-02	2.21E+01	4.55E-02
	4	2.62E+01	1.71E-02	2.60E+01	1.21E-01	2.63E+01	1.31E-01	2.65E+01	9.75E-02	2.63E+01	1.41E-01	2.64E+01	1.96E-01	2.65E+01	7.21E-15	2.66E+01	7.23E-15
	5	3.00E+01	1.28E-01	2.97E+01	1.56E-01	3.06E+01	1.01E-01	3.05E+01	1.16E-01	3.01E+01	2.33E-01	3.05E+01	2.89E-01	3.06E+01	1.08E-14	3.07E+01	1.08E-14
Test5	2	1.79E+01	1.84E-02	1.78E+01	4.49E-02	1.79E+01	3.61E-15	1.79E+01	1.05E-02	1.79E+01	4.04E-03	1.79E+01	1.45E-14	1.79E+01	1.44E-14	1.79E+01	1.45E-14
	3	2.26E+01	7.65E-03	2.24E+01	8.60E-02	2.26E+01	3.61E-15	2.26E+01	1.71E-02	2.26E+01	1.39E-02	2.26E+01	1.04E-03	2.26E+01	1.08E-14	2.26E+01	3.61E-15
	4	2.67E+01	3.97E-02	2.65E+01	9.08E-02	2.68E+01	1.81E-14	2.68E+01	8.42E-02	2.67E+01	4.96E-02	2.68E+01	3.08E-02	2.68E+01	3.60E-15	2.68E+01	3.61E-15
Test6	5	3.06E+01	6.11E-02	3.03E+01	1.23E-01	3.08E+01	1.08E-14	3.07E+01	1.12E-01	3.05E+01	7.90E-02	3.08E+01	6.74E-02	3.08E+01	3.60E-15	3.08E+01	0.00E+00
	2	1.76E+01	3.36E-03	1.76E+01	1.65E-02	1.76E+01	1.08E-14	1.76E+01	7.37E-03	1.76E+01	9.97E-04	1.76E+01	9.89E-14	1.76E+01	3.60E-15	1.76E+01	3.61E-15
	3	2.24E+01	9.71E-03	2.22E+01	1.35E-01	2.24E+01	1.08E-14	2.24E+01	6.31E-02	2.23E+01	2.04E-02	2.24E+01	1.01E-04	2.24E+01	1.08E-14	2.24E+01	1.08E-14
Test7	4	2.64E+01	8.87E-02	2.61E+01	1.82E-01	2.66E+01	2.08E-01	2.68E+01	2.12E-01	2.64E+01	1.80E-01	2.67E+01	2.76E-01	2.69E+01	1.80E-14	2.70E+01	1.81E-14
	5	3.02E+01	1.39E-01	2.97E+01	2.28E-01	3.07E+01	2.98E-01	3.07E+01	2.36E-01	3.02E+01	3.29E-01	3.08E+01	2.98E-01	3.09E+01	1.08E-04	3.10E+01	1.08E-14
	2	1.82E+01	1.33E-03	1.81E+01	2.06E-02	1.82E+01	0.00E+00	1.82E+01	6.76E-04	1.82E+01	1.67E-03	1.82E+01	7.23E-15	1.82E+01	7.21E-15	1.82E+01	7.23E-15
Test8	3	2.26E+01	7.82E-03	2.25E+01	6.95E-02	2.26E+01	1.91E-03	2.26E+01	2.77E-02	2.26E+01	1.43E-02	2.26E+01	3.36E-03	2.26E+01	3.30E-03	2.26E+01	3.89E-03
	4	2.66E+01	8.32E-02	2.64E+01	1.21E-01	2.70E+01	7.73E-02	2.69E+01	1.21E-01	2.68E+01	1.23E-01	2.70E+01	1.49E-01	2.70E+01	1.80E-14	2.71E+01	1.81E-14
	5	3.06E+01	1.71E-01	3.03E+01	1.79E-01	3.11E+01	1.00E-03	3.09E+01	1.70E-01	3.07E+01	1.89E-01	3.11E+01	1.44E-02	3.10E+01	0.00E+00	3.11E+01	3.61E-15
Test9	2	1.79E+01	7.43E-03	1.78E+01	3.47E-02	1.79E+01	1.46E-04	1.79E+01	7.05E-03	1.79E+01	4.66E-03	1.79E+01	2.68E-04	1.79E+01	3.60E-15	1.79E+01	1.08E-14
	3	2.24E+01	1.42E-02	2.23E+01	7.20E-02	2.24E+01	1.08E-14	2.24E+01	5.37E-02	2.24E+01	1.64E-02	2.24E+01	2.89E-03	2.24E+01	1.08E-14	2.24E+01	1.08E-14
	4	2.66E+01	3.58E-02	2.63E+01	1.28E-01	2.66E+01	6.76E-05	2.66E+01	4.24E-02	2.65E+01	5.56E-02	2.66E+01	1.16E-02	2.66E+01	2.52E-14	2.66E+01	1.81E-14
Test10	5	3.04E+01	7.60E-02	3.01E+01	1.75E-01	3.05E+01	1.18E-03	3.04E+01	1.10E-01	3.02E+01	9.57E-02	3.05E+01	1.91E-02	3.05E+01	2.88E-14	3.05E+01	2.17E-14
	2	1.74E+01	3.23E-01	1.73E+01	2.00E-01	1.77E+01	3.68E-02	1.77E+01	8.46E-02	1.78E+01	1.78E-02	1.78E+01	7.71E-05	1.78E+01	1.44E-14	1.78E+01	1.45E-14
	3	2.21E+01	1.83E-01	2.16E+01	3.63E-01	2.24E+01	3.61E-15	2.21E+01	2.41E-01	2.22E+01	7.10E-02	2.24E+01	4.59E-03	2.24E+01	1.08E-14	2.24E+01	1.08E-14
Test11	4	2.60E+01	2.01E-01	2.55E+01	3.68E-01	2.64E+01	1.81E-14	2.61E+01	2.67E-01	2.61E+01	1.82E-01	2.64E+01	1.61E-02	2.64E+01	1.08E-14	2.65E+01	3.61E-15
	5	2.96E+01	2.67E-01	2.88E+01	4.16E-01	3.01E+01	0.00E+00	2.98E+01	3.63E-01	2.96E+01	2.66E-01	3.01E+01	2.63E-02	3.01E+01	7.21E-15	3.02E+01	1.08E-14
	2	1.57E+01	1.53E-02	1.56E+01	5.58E-02	1.57E+01	5.42E-15	1.57E+01	9.27E-03	1.58E+01	4.30E-03	1.57E+01	1.87E-04	1.57E+01	9.01E-15	1.58E+01	5.42E-15
Test12	3	1.94E+01	3.30E-02	1.92E+01	1.18E-01	1.95E+01	4.42E-04	1.95E+01	4.88E-02	1.95E+01	2.01E-02	1.95E+01	5.64E-03	1.95E+01	1.08E-14	1.95E+01	1.19E-04
	4	2.27E+01	5.42E-02	2.24E+01	1.66E-01	2.29E+01	0.00E+00	2.28E+01	8.13E-02	2.28E+01	4.44E-02	2.29E+01	1.19E-02	2.29E+01	1.44E-14	2.29E+01	1.12E-03
	5	2.57E+01	8.31E-02	2.53E+01	2.27E-01	2.59E+01	8.46E-04	2.58E+01	8.56E-02	2.57E+01	9.04E-02	2.59E+01	1.79E-02	2.59E+01	9.73E-04	2.60E+01	1.81E-14
Test13	2	1.74E+01	6.59E-02	1.74E+01	7.90E-02	1.76E+01	1.32E-01	1.77E+01	4.46E-02	1.77E+01	2.01E-02	1.78E+01	7.23E-15	1.77E+01	0.00E+00	1.78E+01	7.23E-15
	3	2.17E+01	2.65E-01	2.16E+01	2.45E-01	2.23E+01	1.52E-01	2.23E+01	2.46E-01	2.23E+01	8.86E-02	2.24E+01	6.71E-04	2.24E+01	1.08E-14	2.25E+01	3.61E-15
	4	2.59E+01	3.25E-01	2.56E+01	3.05E-01	2.65E+01	3.61E-15	2.63E+01	2.85E-01	2.62E+01	2.39E-01	2.66E+01	2.18E-01	2.66E+01	0.00E+00	2.67E+01	1.74E-02
Test14	5	2.98E+01	3.14E-01	2.93E+01	4.00E-01	3.06E+01	5.02E-02	3.02E+01	4.11E-01	3.01E+01	2.53E-01	3.05E+01	3.74E-01	3.07E+01	1.44E-14	3.09E+01	1.45E-14
	Fridman mean rank	3.15		3.33		5.71		4.90		2.85		5.74		4.44		5.88	
	rank	7		6		3		8		4		2		5		1	

Table 21
MPA-OBL Kapur's in terms of PSNR values over all competed algorithms.

Test image	nTh	LSHADE_SPACMA-OBL		CMA-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	1.38E+01	2.56E-01	1.32E+01	9.39E-01	1.36E+01	5.42E-15	1.36E+01	5.42E-15	1.36E+01	2.69E-01	1.36E+01	5.42E-15	1.36E+01	9.01E-15	1.35E+01	5.26E-01
	3	1.61E+01	1.81E+00	1.59E+01	1.54E+00	1.45E+01	3.61E-15	1.45E+01	3.61E-15	1.45E+01	6.32E-01	1.46E+01	7.92E-01	1.45E+01	5.41E-15	1.55E+01	1.50E+00
	4	1.96E+01	7.13E-01	1.85E+01	1.45E+00	2.02E+01	0.00E+00	2.02E+01	0.00E+00	1.94E+01	5.82E-01	2.01E+01	4.20E-02	2.02E+01	0.00E+00	1.98E+01	4.84E-01
	5	2.07E+01	7.69E-01	2.03E+01	1.05E+00	2.07E+01	1.45E-14	2.07E+01	1.45E-14	2.02E+01	1.14E+00	2.07E+01	4.65E-01	2.07E+01	1.44E-14	2.12E+01	7.98E-01
Test2	2	1.46E+01	5.29E-02	1.41E+01	5.34E-01	1.46E+01	5.42E-15	1.46E+01	5.42E-15	1.46E+01	9.35E-02	1.46E+01	5.92E-05	1.46E+01	5.41E-15	1.46E+01	4.54E-02
	3	1.72E+01	5.02E-01	1.67E+01	7.56E-01	1.66E+01	5.59E-01	1.62E+01	7.23E-15	1.59E+01	4.13E-01	1.63E+01	3.47E-01	1.62E+01	7.21E-15	1.62E+01	7.65E-01
	4	1.92E+01	1.13E-01	1.81E+01	8.95E-01	1.93E+01	3.17E-02	1.93E+01	3.61E-15	1.88E+01	4.21E-01	1.93E+01	1.46E-02	1.93E+01	3.60E-15	1.88E+01	3.59E-01
	5	2.03E+01	5.31E-01	1.93E+01	9.95E-01	2.09E+01	2.86E-02	2.09E+01	1.08E-14	2.01E+01	5.71E-01	2.09E+01	1.82E-01	2.09E+01	1.80E-14	2.05E+01	4.75E-01
Test3	2	1.60E+01	1.65E-02	1.59E+01	1.24E-01	1.60E+01	7.23E-15	1.60E+01	7.23E-15	1.60E+01	2.65E-02	1.60E+01	7.23E-15	1.60E+01	3.60E-15	1.60E+01	1.57E-02
	3	1.75E+01	1.48E+00	1.83E+01	4.90E-01	1.88E+01	0.00E+00	1.68E+01	1.28E+00	1.80E+01	1.16E+00	1.85E+01	8.38E-01	1.66E+01	1.17E+00	1.86E+01	2.03E-01
	4	1.81E+01	7.62E-01	1.96E+01	9.54E-01	1.95E+01	9.48E-01	1.88E+01	0.00E+00	1.82E+01	1.36E+00	1.95E+01	1.02E+00	1.88E+01	0.00E+00	2.03E+01	7.38E-01
	5	1.98E+01	6.65E-01	2.11E+01	1.10E+00	2.05E+01	3.49E-01	2.05E+01	7.23E-15	1.95E+01	1.23E+00	2.08E+01	8.58E-01	2.05E+01	7.21E-15	2.19E+01	5.69E-01
Test4	2	1.51E+01	1.65E-01	1.50E+01	2.75E-01	1.52E+01	1.08E-14	1.52E+01	1.08E-14	1.52E+01	6.99E-02	1.52E+01	1.08E-14	1.52E+01	9.01E-15	1.52E+01	7.15E-02
	3	1.84E+01	9.31E-02	1.80E+01	3.77E-01	1.85E+01	1.08E-14	1.85E+01	1.08E-14	1.84E+01	6.93E-02	1.85E+01	1.89E-03	1.85E+01	3.60E-15	1.84E+01	3.41E-02
	4	2.08E+01	7.87E-01	2.04E+01	3.24E-01	2.10E+01	7.23E-15	2.10E+01	7.23E-15	2.07E+01	2.38E-01	2.09E+01	4.49E-01	2.10E+01	7.21E-15	2.09E+01	1.80E-01
	5	2.05E+01	5.36E-01	2.16E+01	7.66E-01	2.11E+01	0.00E+00	2.11E+01	0.00E+00	2.14E+01	9.80E-01	2.15E+01	7.77E-01	2.11E+01	0.00E+00	2.18E+01	9.27E-01
Test5	2	1.59E+01	2.52E-01	1.59E+01	5.12E-01	1.58E+01	9.03E-15	1.58E+01	9.03E-15	1.58E+01	1.02E-01	1.58E+01	9.03E-15	1.58E+01	9.01E-15	1.58E+01	1.38E-01
	3	1.85E+01	9.06E-01	1.81E+01	1.01E+00	1.88E+01	3.61E-15	1.88E+01	3.61E-15	1.86E+01	3.34E-01	1.88E+01	6.87E-05	1.88E+01	3.60E-15	1.87E+01	2.37E-01
	4	1.86E+01	3.69E-01	1.95E+01	9.99E-01	1.96E+01	7.69E-01	1.88E+01	3.61E-15	1.85E+01	1.33E+00	1.94E+01	7.63E-01	1.88E+01	3.60E-15	2.02E+01	6.25E-01
	5	1.97E+01	6.59E-01	2.05E+01	1.40E+00	2.04E+01	2.35E-01	2.03E+01	0.00E+00	1.92E+01	1.27E+00	2.01E+01	7.72E-01	2.03E+01	1.24E-04	2.05E+01	1.14E+00
Test6	2	1.63E+01	9.67E-04	1.61E+01	2.33E-01	1.63E+01	0.00E+00	1.63E+01	0.00E+00	1.63E+01	4.42E-02	1.63E+01	0.00E+00	1.63E+01	3.60E-15	1.63E+01	2.56E-02
	3	1.71E+01	1.09E+00	1.80E+01	5.08E-01	1.82E+01	6.43E-01	1.72E+01	1.06E+00	1.81E+01	6.69E-01	1.82E+01	6.37E-01	1.68E+01	8.92E-01	1.83E+01	9.73E-02
	4	1.79E+01	5.13E-01	1.91E+01	1.13E+00	1.85E+01	4.26E-01	1.84E+01	7.23E-15	1.79E+01	6.53E-01	1.87E+01	8.04E-01	1.84E+01	0.00E+00	1.95E+01	1.32E+00
	5	1.94E+01	1.07E+00	2.00E+01	1.41E+00	2.07E+01	1.17E-04	2.07E+01	7.23E-15	1.92E+01	8.14E-01	2.04E+01	4.14E-01	2.07E+01	7.21E-15	2.02E+01	8.96E-01
Test7	2	1.45E+01	2.35E-01	1.44E+01	4.60E-01	1.46E+01	1.06E-02	1.46E+01	3.61E-15	1.45E+01	1.45E-01	1.46E+01	5.94E-02	1.46E+01	5.41E-15	1.45E+01	1.55E-01
	3	1.71E+01	3.35E-01	1.71E+01	6.53E-01	1.71E+01	1.08E-14	1.71E+01	1.08E-14	1.71E+01	3.07E-01	1.72E+01	1.03E-01	1.71E+01	3.60E-15	1.74E+01	1.77E-01
	4	1.90E+01	4.78E-01	1.89E+01	9.07E-01	1.91E+01	4.90E-03	1.91E+01	3.61E-15	1.92E+01	7.09E-01	1.92E+01	2.88E-01	1.91E+01	3.60E-15	1.93E+01	5.45E-01
	5	1.99E+01	8.71E-01	2.06E+01	8.81E-01	2.12E+01	4.72E-03	2.12E+01	3.61E-15	2.05E+01	5.97E-01	2.07E+01	4.96E-01	2.12E+01	1.08E-14	2.09E+01	5.22E-01
Test8	2	1.17E+01	1.02E+00	1.17E+01	1.40E+00	1.22E+01	3.87E-01	1.22E+01	1.81E-15	1.21E+01	2.91E-01	1.22E+01	4.37E-02	1.22E+01	1.80E-15	1.24E+01	2.00E+00
	3	1.54E+01	1.19E+00	1.53E+01	1.37E+00	1.69E+01	7.23E-15	1.69E+01	7.23E-15	1.64E+01	4.51E-01	1.68E+01	6.13E-02	1.69E+01	0.00E+00	1.64E+01	5.95E-01
	4	1.78E+01	1.96E+00	1.84E+01	1.27E+00	2.01E+01	7.23E-15	2.01E+01	7.23E-15	1.91E+01	7.61E-01	1.99E+01	1.99E-01	2.01E+01	7.21E-15	1.93E+01	5.49E-01
	5	1.97E+01	1.87E+00	1.96E+01	1.17E+00	2.20E+01	0.00E+00	2.01E+01	7.23E-15	2.03E+01	1.36E+00	2.11E+01	1.04E+00	2.01E+01	7.21E-15	2.06E+01	1.32E+00
Test9	2	1.37E+01	3.76E-01	1.36E+01	9.38E-01	1.35E+01	9.03E-15	1.27E+01	0.00E+00	1.31E+01	2.45E-01	1.37E+01	4.35E-01	1.35E+01	1.26E-14	1.35E+01	4.93E-01
	3	1.65E+01	4.84E-01	1.66E+01	8.76E-01	1.68E+01	1.16E-02	1.69E+01	5.29E-02	1.69E+01	2.70E-01	1.69E+01	1.08E-01	1.68E+01	0.00E+00	1.69E+01	2.46E-01
	4	1.87E+01	3.48E-01	1.82E+01	8.17E-01	1.89E+01	1.45E-14	1.89E+01	7.90E-03	1.88E+01	2.97E-01	1.89E+01	1.17E-01	1.89E+01	1.44E-14	1.90E+01	2.27E-01
	5	1.99E+01	6.48E-01	1.96E+01	8.84E-01	2.03E+01	6.56E-02	2.02E+01	3.61E-15	2.01E+01	6.23E-01	2.04E+01	2.32E-01	2.03E+01	6.15E-02	2.05E+01	3.07E-01
Test10	2	1.06E+01	6.08E-01	1.41E+01	1.24E+00	1.18E+01	1.55E+00	1.08E+01	1.81E-15	1.08E+01	4.78E-01	1.08E+01	1.81E-15	1.08E+01	5.41E-15	1.36E+01	1.76E+00
	3	1.39E+01	1.28E+00	1.52E+01	2.11E+00	1.44E+01	6.50E-01	1.43E+01	5.42E-15	1.46E+01	9.68E-01	1.44E+01	1.45E-01	1.43E+01	9.01E-15	1.62E+01	1.80E+00
	4	1.67E+01	1.75E+00	1.67E+01	1.61E+00	1.78E+01	7.23E-15	1.61E+01	3.40E-01	1.60E+01	1.26E+00	1.77E+01	6.93E-01	1.60E+01	0.00E+00	1.73E+01	1.62E+00
	5	1.86E+01	1.76E+00	1.84E+01	1.95E+00	1.94E+01	1.01E-01	1.94E+01	0.00E+00	1.82E+01	9.74E-01	1.97E+01	7.42E-01	1.94E+01	0.00E+00	1.83E+01	1.83E+00

Table 22
MPA-OBL Kapur's in terms of SSIM values over all competed algorithms.

Test image	nTh	LSHADE_SPACMA-OBL		CMA-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	6.52E-01	4.36E-02	6.63E-01	1.13E-16	6.70E-01	1.12E-02	6.61E-01	2.14E-02	6.64E-01	1.13E-02	6.63E-01	1.13E-16	6.63E-01	3.38E-16	6.63E-01	1.13E-16
	3	7.78E-01	1.70E-02	7.72E-01	5.65E-16	7.75E-01	1.15E-02	7.78E-01	1.14E-02	7.71E-01	1.12E-02	7.72E-01	5.65E-16	7.72E-01	5.63E-16	7.73E-01	3.75E-03
	4	8.18E-01	2.06E-02	8.33E-01	4.52E-16	8.27E-01	1.46E-02	8.31E-01	1.07E-02	8.34E-01	1.11E-02	8.33E-01	4.52E-16	8.33E-01	6.76E-16	8.35E-01	2.45E-03
	5	8.37E-01	2.04E-02	8.51E-01	3.39E-16	8.48E-01	1.27E-02	8.50E-01	1.34E-02	8.37E-01	2.49E-02	8.51E-01	3.39E-16	8.51E-01	4.51E-16	8.50E-01	6.37E-03
Test2	2	5.92E-01	3.55E-02	6.16E-01	1.13E-16	6.17E-01	4.71E-03	6.17E-01	3.43E-03	6.15E-01	5.89E-03	6.16E-01	1.13E-16	6.16E-01	1.13E-16	6.16E-01	2.67E-06
	3	7.22E-01	2.20E-02	7.30E-01	1.02E-02	7.23E-01	9.55E-03	7.20E-01	1.79E-02	7.26E-01	1.85E-02	7.38E-01	1.13E-16	7.38E-01	3.38E-16	7.36E-01	6.40E-03
	4	7.83E-01	2.69E-02	8.17E-01	6.07E-03	8.02E-01	1.82E-02	8.04E-01	2.20E-02	8.10E-01	1.91E-02	8.18E-01	3.39E-16	8.18E-01	3.38E-16	8.20E-01	2.53E-03
	5	8.17E-01	3.40E-02	8.61E-01	7.38E-04	8.25E-01	2.86E-02	8.44E-01	1.63E-02	8.38E-01	1.87E-02	8.61E-01	5.65E-16	8.61E-01	5.63E-16	8.58E-01	7.31E-03
Test3	2	8.19E-01	5.16E-03	8.23E-01	2.26E-16	8.23E-01	2.81E-08	8.23E-01	5.31E-04	8.23E-01	9.99E-04	8.23E-01	2.26E-16	8.23E-01	4.51E-16	8.23E-01	2.26E-16
	3	8.88E-01	1.10E-02	9.00E-01	5.65E-16	8.67E-01	4.28E-02	8.98E-01	2.75E-03	8.79E-01	3.28E-02	8.46E-01	3.59E-02	8.41E-01	3.29E-02	8.93E-01	2.35E-02
	4	9.07E-01	2.25E-02	9.11E-01	1.45E-02	8.82E-01	2.30E-02	9.22E-01	1.34E-02	8.80E-01	3.31E-02	9.00E-01	5.65E-16	9.00E-01	7.89E-02	9.11E-01	1.47E-02
	5	9.32E-01	1.71E-02	9.27E-01	5.05E-03	9.13E-01	1.34E-02	9.47E-01	9.38E-03	9.07E-01	2.51E-02	9.26E-01	2.26E-16	9.26E-01	2.25E-16	9.31E-01	1.21E-02
Test4	2	6.02E-01	2.61E-02	6.10E-01	1.13E-16	6.10E-01	1.16E-02	6.10E-01	7.88E-03	6.10E-01	8.15E-03	6.10E-01	1.13E-16	6.10E-01	1.13E-16	6.10E-01	1.13E-16
	3	7.54E-01	2.99E-02	7.53E-01	3.39E-16	7.61E-01	1.34E-02	7.56E-01	9.88E-03	7.59E-01	1.32E-02	7.53E-01	3.39E-16	7.53E-01	3.38E-16	7.54E-01	1.25E-03
	4	8.29E-01	3.02E-02	8.37E-01	5.65E-16	8.38E-01	3.07E-02	8.37E-01	1.35E-02	8.40E-01	1.72E-02	8.37E-01	5.65E-16	8.37E-01	5.63E-16	8.35E-01	1.38E-02
	5	8.60E-01	2.64E-02	8.39E-01	5.65E-16	8.30E-01	2.76E-02	8.59E-01	3.18E-02	8.67E-01	2.35E-02	8.39E-01	5.65E-16	8.39E-01	5.63E-16	8.57E-01	1.98E-02
Test5	2	8.76E-01	5.20E-03	8.77E-01	4.52E-16	8.77E-01	2.20E-04	8.76E-01	2.15E-03	8.77E-01	2.07E-03	8.77E-01	4.52E-16	8.77E-01	4.51E-16	8.77E-01	4.52E-16
	3	9.12E-01	1.12E-02	9.23E-01	5.65E-16	9.16E-01	1.67E-02	9.23E-01	1.73E-03	9.22E-01	3.17E-03	9.23E-01	5.65E-16	9.23E-01	7.89E-16	9.23E-01	5.23E-07
	4	9.32E-01	1.10E-02	9.34E-01	1.14E-02	9.19E-01	8.93E-03	9.42E-01	5.18E-03	9.15E-01	2.07E-02	9.23E-01	5.65E-16	9.23E-01	7.89E-16	9.33E-01	1.08E-02
	5	9.37E-01	2.09E-02	9.48E-01	3.15E-03	9.33E-01	1.49E-02	9.47E-01	7.01E-03	9.27E-01	1.15E-02	9.46E-01	0.00E+00	9.46E-01	4.72E-05	9.46E-01	6.75E-03
Test6	2	7.50E-01	2.06E-02	7.58E-01	2.26E-16	7.58E-01	1.14E-07	7.59E-01	3.40E-03	7.58E-01	4.25E-03	7.58E-01	2.26E-16	7.58E-01	4.51E-16	7.58E-01	2.26E-16
	3	7.95E-01	1.69E-02	8.01E-01	1.43E-02	7.72E-01	3.35E-02	8.05E-01	2.96E-03	8.00E-01	1.45E-02	7.79E-01	2.42E-02	7.69E-01	2.04E-02	8.02E-01	1.41E-02
	4	8.23E-01	3.06E-02	8.08E-01	1.03E-02	7.97E-01	1.71E-02	8.34E-01	3.26E-02	8.04E-01	1.97E-02	8.06E-01	5.65E-16	8.06E-01	7.89E-16	8.15E-01	1.89E-02
	5	8.48E-01	3.21E-02	8.62E-01	1.54E-10	8.37E-01	2.40E-02	8.56E-01	1.91E-02	8.41E-01	1.92E-02	8.62E-01	1.13E-16	8.62E-01	1.13E-16	8.58E-01	1.20E-02
Test7	2	6.84E-01	1.26E-02	6.85E-01	3.07E-03	6.86E-01	4.95E-03	6.88E-01	4.27E-03	6.88E-01	4.01E-03	6.84E-01	4.52E-16	6.84E-01	4.51E-16	6.88E-01	5.01E-03
	3	8.04E-01	2.35E-02	8.24E-01	4.52E-16	8.20E-01	8.27E-03	8.27E-01	3.71E-03	8.21E-01	6.19E-03	8.24E-01	4.52E-16	8.24E-01	6.76E-16	8.25E-01	1.50E-03
	4	8.57E-01	2.30E-02	8.63E-01	6.13E-05	8.59E-01	1.21E-02	8.67E-01	1.31E-02	8.65E-01	1.50E-02	8.63E-01	3.39E-16	8.63E-01	5.63E-16	8.66E-01	7.38E-03
	5	8.91E-01	1.80E-02	9.09E-01	6.14E-05	8.82E-01	2.10E-02	9.03E-01	8.57E-03	8.96E-01	9.97E-03	9.09E-01	4.52E-16	9.09E-01	5.63E-16	9.02E-01	1.01E-02
Test8	2	5.65E-01	9.88E-02	5.90E-01	1.11E-03	5.94E-01	3.05E-02	6.20E-01	9.23E-02	5.80E-01	3.06E-02	5.90E-01	1.13E-16	5.90E-01	1.13E-16	5.88E-01	5.07E-03
	3	7.65E-01	6.62E-02	8.34E-01	5.65E-16	7.66E-01	6.82E-02	8.13E-01	3.47E-02	8.20E-01	1.86E-02	8.34E-01	5.65E-16	8.34E-01	5.63E-16	8.34E-01	3.49E-03
	4	8.54E-01	4.37E-02	9.00E-01	3.39E-16	8.44E-01	5.71E-02	8.85E-01	1.15E-02	8.78E-01	2.09E-02	9.00E-01	3.39E-16	9.00E-01	3.38E-16	8.99E-01	2.19E-03
	5	8.84E-01	2.85E-02	9.29E-01	0.00E+00	8.85E-01	4.34E-02	9.03E-01	3.22E-02	9.02E-01	2.53E-02	9.00E-01	3.39E-16	9.00E-01	3.38E-16	9.17E-01	1.57E-02
Test9	2	5.88E-01	7.56E-02	5.87E-01	2.26E-16	6.05E-01	3.19E-02	5.82E-01	3.93E-02	5.63E-01	2.04E-02	5.37E-01	3.39E-16	5.87E-01	0.00E+00	6.01E-01	3.40E-02
	3	7.74E-01	5.53E-02	7.87E-01	6.86E-05	7.79E-01	1.86E-02	7.86E-01	1.49E-02	7.85E-01	1.64E-02	7.83E-01	3.57E-03	7.87E-01	1.13E-16	7.88E-01	6.36E-03
	4	8.40E-01	3.37E-02	8.66E-01	0.00E+00	8.59E-01	1.21E-02	8.65E-01	1.12E-02	8.54E-01	1.47E-02	8.64E-01	1.81E-03	8.66E-01	0.00E+00	8.68E-01	3.98E-03
	5	8.73E-01	3.20E-02	9.02E-01	1.82E-03	8.88E-01	1.72E-02	9.01E-01	9.46E-03	8.92E-01	2.09E-02	8.95E-01	6.78E-16	9.03E-01	1.87E-03	9.05E-01	7.56E-03
Test10	2	6.27E-01	8.12E-02	4.67E-01	1.05E-01	3.88E-01	5.16E-02	5.91E-01	1.27E-01	4.00E-01	4.08E-02	4.05E-01	1.13E-16	4.05E-01	2.25E-16	4.05E-01	1.13E-16
	3	6.79E-01	1.20E-01	6.43E-01	3.12E-02	6.09E-01	8.95E-02	7.35E-01	9.57E-02	6.56E-01	6.50E-02	6.37E-01	1.13E-16	6.37E-01	3.38E-16	6.43E-01	9.73E-03
	4	7.62E-01	7.97E-02	8.08E-01	5.65E-16	7.61E-01	9.00E-02	7.73E-01	7.73E-02	7.52E-01	6.14E-02	7.58E-01	9.71E-03	7.56E-01	5.63E-16	8.08E-01	2.08E-02
	5	8.19E-01	7.36E-02	8.72E-01	7.39E-03	8.29E-01	7.18E-02	8.22E-01	7.27E-02	8.38E-01	3.81E-02	8.73E-01	2.26E-16	8.73E-01	3.38E-16	8.73E-01	2.22E-02

Table 23
MPA-OBL Kapur's in terms of FSIM values over all competed algorithms.

Test image	nTh	LSHADE_SPACMA-OBL		CMA-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Test1	2	6.78E-01	3.39E-16	6.78E-01	2.77E-02	6.78E-01	3.39E-16	6.83E-01	8.04E-03	6.79E-01	7.76E-03	6.78E-01	3.39E-16	6.78E-01	3.38E-16	6.78E-01	1.20E-02
	3	7.99E-01	2.26E-16	7.96E-01	1.18E-02	7.99E-01	2.26E-16	7.89E-01	1.06E-02	7.97E-01	3.71E-03	7.99E-01	2.04E-03	7.99E-01	2.25E-16	7.97E-01	6.24E-03
	4	8.41E-01	2.26E-16	8.30E-01	2.39E-02	8.41E-01	2.26E-16	8.35E-01	1.80E-02	8.46E-01	1.54E-02	8.45E-01	4.24E-03	8.41E-01	4.51E-16	8.42E-01	1.40E-02
	5	8.62E-01	3.39E-16	8.54E-01	2.37E-02	8.62E-01	3.39E-16	8.61E-01	1.57E-02	8.52E-01	2.62E-02	8.63E-01	6.66E-03	8.62E-01	3.38E-16	8.67E-01	1.58E-02
Test2	2	6.74E-01	4.52E-16	6.68E-01	1.19E-02	6.74E-01	4.52E-16	6.74E-01	1.99E-03	6.73E-01	2.79E-03	6.74E-01	9.59E-06	6.74E-01	4.51E-16	6.74E-01	1.79E-03
	3	6.91E-01	0.00E+00	7.26E-01	2.95E-02	7.11E-01	2.64E-02	7.37E-01	2.00E-02	6.86E-01	1.10E-02	6.96E-01	1.64E-02	6.91E-01	2.25E-16	7.05E-01	2.66E-02
	4	7.62E-01	1.13E-16	7.58E-01	2.68E-02	7.63E-01	5.79E-03	7.76E-01	1.72E-02	7.64E-01	1.46E-02	7.63E-01	2.88E-03	7.62E-01	3.38E-16	7.63E-01	1.50E-02
	5	8.12E-01	3.39E-16	7.87E-01	2.64E-02	8.12E-01	1.61E-03	8.13E-01	1.07E-02	7.99E-01	1.21E-02	8.13E-01	4.68E-03	8.12E-01	5.63E-16	8.07E-01	1.22E-02
Test3	2	8.65E-01	2.26E-16	8.60E-01	7.01E-03	8.65E-01	2.26E-16	8.65E-01	4.73E-04	8.64E-01	1.42E-03	8.65E-01	2.26E-16	8.65E-01	4.51E-16	8.65E-01	6.92E-04
	3	8.77E-01	1.90E-02	8.93E-01	1.02E-02	9.06E-01	4.52E-16	8.86E-01	2.65E-02	8.92E-01	2.14E-02	9.02E-01	1.25E-02	8.74E-01	1.74E-02	9.04E-01	2.68E-03
	4	9.06E-01	4.52E-16	9.14E-01	1.85E-02	9.14E-01	1.12E-02	8.93E-01	1.76E-02	8.88E-01	3.04E-02	9.14E-01	1.20E-02	9.06E-01	4.51E-16	9.22E-01	1.49E-02
	5	9.25E-01	1.13E-16	9.37E-01	1.70E-02	9.26E-01	5.58E-03	9.18E-01	1.06E-02	9.11E-01	2.49E-02	9.31E-01	1.41E-02	9.25E-01	0.00E+00	9.50E-01	9.37E-03
Test4	2	7.15E-01	0.00E+00	7.10E-01	1.69E-02	7.15E-01	0.00E+00	7.15E-01	7.67E-03	7.15E-01	5.50E-03	7.15E-01	0.00E+00	7.15E-01	0.00E+00	7.15E-01	5.63E-03
	3	8.24E-01	5.65E-16	8.16E-01	1.28E-02	8.24E-01	5.65E-16	8.24E-01	3.92E-03	8.24E-01	4.13E-03	8.24E-01	2.28E-04	8.24E-01	5.63E-16	8.23E-01	3.15E-03
	4	8.89E-01	5.65E-16	8.76E-01	1.37E-02	8.89E-01	5.65E-16	8.86E-01	2.36E-02	8.85E-01	7.75E-03	8.87E-01	1.11E-02	8.89E-01	5.63E-16	8.88E-01	6.80E-03
	5	8.91E-01	2.26E-16	9.00E-01	1.74E-02	8.91E-01	2.26E-16	8.79E-01	1.62E-02	9.02E-01	1.85E-02	9.02E-01	1.48E-02	8.91E-01	2.25E-16	9.05E-01	2.12E-02
Test5	2	7.91E-01	2.26E-16	7.98E-01	9.56E-03	7.91E-01	2.26E-16	7.95E-01	7.90E-03	7.91E-01	1.86E-03	7.91E-01	2.26E-16	7.91E-01	4.51E-16	7.90E-01	2.05E-03
	3	8.61E-01	3.39E-16	8.52E-01	1.27E-02	8.61E-01	3.39E-16	8.52E-01	2.12E-02	8.60E-01	2.81E-03	8.61E-01	1.53E-05	8.61E-01	5.63E-16	8.61E-01	1.66E-03
	4	8.61E-01	3.39E-16	8.78E-01	1.55E-02	8.79E-01	1.89E-02	8.57E-01	1.13E-02	8.55E-01	2.48E-02	8.76E-01	1.83E-02	8.61E-01	5.63E-16	8.94E-01	7.34E-03
	5	8.98E-01	2.26E-16	8.86E-01	2.39E-02	8.99E-01	3.62E-03	8.80E-01	1.67E-02	8.68E-01	1.67E-02	8.96E-01	1.31E-02	8.98E-01	1.82E-04	8.98E-01	1.19E-02
Test6	2	7.21E-01	3.39E-16	7.19E-01	9.27E-03	7.21E-01	3.39E-16	7.21E-01	9.27E-05	7.21E-01	1.70E-03	7.21E-01	3.39E-16	7.21E-01	3.38E-16	7.22E-01	1.36E-03
	3	7.47E-01	2.92E-02	7.70E-01	1.41E-02	7.74E-01	1.76E-02	7.46E-01	3.09E-02	7.73E-01	1.69E-02	7.74E-01	1.74E-02	7.35E-01	2.47E-02	7.80E-01	2.67E-03
	4	7.80E-01	0.00E+00	7.95E-01	2.14E-02	7.82E-01	8.12E-03	7.70E-01	1.33E-02	7.74E-01	1.23E-02	7.86E-01	1.54E-02	7.80E-01	2.25E-16	8.03E-01	2.44E-02
	5	8.25E-01	1.13E-16	8.20E-01	2.01E-02	8.25E-01	5.23E-06	8.00E-01	1.59E-02	8.01E-01	1.29E-02	8.22E-01	5.26E-03	8.25E-01	3.38E-16	8.22E-01	1.60E-02
Test7	2	7.16E-01	5.65E-16	7.19E-01	1.51E-02	7.17E-01	3.54E-03	7.19E-01	6.36E-03	7.22E-01	5.04E-03	7.21E-01	5.91E-03	7.16E-01	5.63E-16	7.22E-01	5.74E-03
	3	7.99E-01	2.26E-16	8.00E-01	1.91E-02	7.99E-01	2.26E-16	7.98E-01	9.73E-03	7.99E-01	8.36E-03	8.00E-01	2.68E-03	7.99E-01	4.51E-16	8.06E-01	5.24E-03
	4	8.55E-01	5.65E-16	8.47E-01	1.92E-02	8.55E-01	2.48E-05	8.51E-01	6.60E-03	8.58E-01	1.33E-02	8.58E-01	5.85E-03	8.55E-01	5.63E-16	8.60E-01	9.89E-03
	5	9.05E-01	2.26E-16	8.84E-01	1.80E-02	9.05E-01	1.46E-04	8.74E-01	2.11E-02	8.87E-01	8.74E-03	8.95E-01	1.12E-02	9.05E-01	2.25E-16	8.95E-01	8.56E-03
Test8	2	6.48E-01	1.13E-16	6.33E-01	3.73E-02	6.46E-01	1.06E-02	6.34E-01	2.97E-02	6.44E-01	1.06E-02	6.47E-01	1.70E-03	6.48E-01	1.13E-16	6.49E-01	4.65E-02
	3	7.46E-01	2.26E-16	7.26E-01	2.64E-02	7.46E-01	2.26E-16	7.30E-01	2.87E-02	7.37E-01	8.65E-03	7.46E-01	1.38E-03	7.46E-01	4.51E-16	7.40E-01	7.38E-03
	4	8.37E-01	3.39E-16	8.06E-01	2.59E-02	8.37E-01	3.39E-16	7.82E-01	4.93E-02	8.18E-01	2.12E-02	8.32E-01	5.66E-03	8.37E-01	5.63E-16	8.19E-01	1.74E-02
	5	8.37E-01	3.39E-16	8.31E-01	3.28E-02	8.84E-01	5.65E-16	8.30E-01	5.07E-02	8.46E-01	3.37E-02	8.62E-01	2.76E-02	8.37E-01	5.63E-16	8.55E-01	2.97E-02
Test9	2	7.47E-01	4.52E-16	7.55E-01	2.25E-02	7.63E-01	2.26E-16	7.61E-01	6.26E-03	7.52E-01	5.17E-03	7.62E-01	5.32E-03	7.63E-01	2.25E-16	7.57E-01	9.10E-03
	3	8.49E-01	5.41E-04	8.33E-01	1.75E-02	8.48E-01	5.96E-05	8.36E-01	1.65E-02	8.47E-01	3.90E-03	8.48E-01	2.01E-03	8.48E-01	5.63E-16	8.45E-01	4.82E-03
	4	9.00E-01	1.22E-04	8.72E-01	1.76E-02	9.00E-01	0.00E+00	8.91E-01	9.58E-03	8.91E-01	6.32E-03	8.99E-01	1.07E-03	9.00E-01	0.00E+00	8.96E-01	4.02E-03
	5	9.23E-01	3.39E-16	8.99E-01	1.93E-02	9.25E-01	1.57E-04	9.12E-01	1.15E-02	9.10E-01	8.39E-03	9.24E-01	1.92E-03	9.25E-01	2.67E-04	9.22E-01	2.73E-03
Test10	2	6.47E-01	1.13E-16	7.23E-01	3.00E-02	6.70E-01	3.88E-02	6.38E-01	2.27E-02	6.43E-01	1.44E-02	6.47E-01	1.13E-16	6.47E-01	1.13E-16	7.14E-01	5.18E-02
	3	7.33E-01	3.39E-16	7.44E-01	4.84E-02	7.36E-01	1.42E-02	7.23E-01	2.23E-02	7.32E-01	1.15E-02	7.34E-01	1.95E-03	7.33E-01	3.38E-16	7.70E-01	3.89E-02
	4	7.57E-01	1.01E-02	7.77E-01	4.17E-02	8.11E-01	4.52E-16	7.83E-01	3.09E-02	7.52E-01	2.73E-02	8.03E-01	2.08E-02	7.55E-01	2.25E-16	7.92E-01	3.15E-02
	5	8.34E-01	5.65E-16	8.20E-01	3.64E-02	8.35E-01	2.30E-03	8.23E-01	3.03E-02	8.02E-01	2.11E-02	8.47E-01	1.50E-02	8.34E-01	7.89E-16	8.17E-01	3.25E-02

Table 24

The mean of time values for 30 runs obtained with the different algorithms for the segmentation experiment based on Kapur's objective function.

Test image	nTh	LSHADE_SPACMA-OBL		CMA-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA		MPA-OBL	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Test1	2	0.9111	0.1174	1.9400	0.2925	1.2449	0.0971	3.7972	0.0845	2.4368	0.8439	2.2472	0.1119	1.4345	0.0218	1.6980	0.0572
	3	0.9887	0.0416	1.9370	0.2738	1.4325	0.0880	4.5817	0.0327	4.3004	0.3209	2.5900	0.1194	1.6956	0.0098	1.9606	0.0373
	4	1.1661	0.1730	2.0271	0.1201	1.4599	0.0350	6.3684	1.5260	5.1943	0.2915	2.9284	0.0764	1.9551	0.0205	2.2394	0.0385
	5	1.1942	0.2601	2.0592	0.0734	1.6100	0.0157	8.1429	0.8922	5.9765	0.2190	3.2996	0.0883	2.2294	0.0094	2.5216	0.0524
	2	0.8681	0.0137	1.5612	0.0688	1.1332	0.0124	4.1626	0.4607	3.8394	0.0930	2.2031	0.0658	1.4364	0.0364	1.6780	0.0428
Test2	3	0.9394	0.0345	1.7249	0.0926	1.2915	0.0144	4.8289	0.7654	4.5017	0.1710	2.6235	0.1690	1.6960	0.0083	1.9599	0.0476
	4	1.0212	0.0316	1.9482	0.1257	1.4488	0.0059	6.6781	1.0907	5.3415	0.1251	3.0451	0.1990	1.9654	0.0106	2.2398	0.0374
	5	1.0873	0.0356	2.2699	0.2997	1.6065	0.0057	6.4679	0.4803	3.6224	1.3594	3.3877	0.2060	2.6895	0.8851	2.5273	0.0601
	2	1.6568	0.0331	1.5494	0.0732	1.1276	0.0062	4.3205	0.0888	2.6345	0.2003	2.7745	0.0822	1.6076	0.3120	1.6788	0.0358
	3	1.8955	0.0378	1.6555	0.0386	1.3066	0.0301	5.3010	0.1216	2.9293	0.0996	3.2240	0.0816	1.6981	0.0083	1.9574	0.0337
Test3	4	2.1472	0.0623	1.8136	0.0384	1.4749	0.0311	6.2974	0.1698	3.8375	0.4973	3.6739	0.0571	1.9670	0.0418	2.2447	0.0346
	5	2.3646	0.0215	1.9661	0.0216	1.6337	0.0255	7.6452	0.7971	3.9035	0.2132	4.1684	0.1469	2.2322	0.0206	2.5438	0.0557
	2	1.6358	0.0192	1.5300	0.1570	1.1438	0.0233	4.7011	0.3897	2.4153	0.0593	2.7582	0.0931	1.4200	0.0090	1.6943	0.0409
	3	1.8794	0.0500	1.6454	0.0377	1.2967	0.0090	5.5033	0.4152	2.8602	0.0755	3.2399	0.0900	1.6908	0.0103	1.9771	0.0545
	4	2.1112	0.0372	1.8030	0.0245	1.4717	0.0096	6.2730	0.0625	3.3253	0.0489	3.6733	0.0634	1.9535	0.0076	2.2355	0.0382
Test4	5	2.3553	0.0525	1.9566	0.0195	1.6225	0.0080	7.2076	0.1989	3.7778	0.0417	4.1223	0.0459	2.2387	0.0471	2.5172	0.0502
	2	1.6379	0.0276	1.4743	0.0362	1.1594	0.0253	4.3057	0.0804	2.4321	0.0864	2.7431	0.0717	1.4423	0.0499	1.7157	0.0624
	3	1.8715	0.0138	1.6495	0.0468	1.3024	0.0387	5.2353	0.0495	2.8696	0.0406	3.1926	0.0466	1.6869	0.0093	2.0012	0.0610
	4	2.2027	0.2281	1.8006	0.0381	1.4689	0.0105	6.1704	0.0860	3.3487	0.0458	3.6745	0.0699	1.9549	0.0140	2.2598	0.0460
	5	2.3746	0.0565	1.9540	0.0392	1.6279	0.0118	7.1314	0.1488	3.8084	0.0976	4.1506	0.1021	2.2221	0.0115	2.5420	0.0562
Test5	2	1.6700	0.0523	1.4685	0.0325	1.1392	0.0302	4.3308	0.0490	2.4502	0.1245	2.7155	0.0218	1.4260	0.0126	1.6809	0.0343
	3	1.8870	0.0279	1.6324	0.0178	1.2853	0.0085	5.2470	0.0680	3.0153	0.6726	3.2018	0.0568	1.6973	0.0378	1.9663	0.0634
	4	2.1325	0.0392	1.8012	0.0393	1.4524	0.0180	6.1940	0.1114	3.3267	0.0499	3.7008	0.0941	1.9575	0.0096	2.2431	0.0417
	5	2.3595	0.0313	1.9709	0.0532	1.6121	0.0187	7.0892	0.0802	3.7911	0.0671	4.1556	0.0986	2.2164	0.0094	2.5571	0.0607
	2	1.6587	0.0684	1.4741	0.0263	1.1345	0.0273	4.3370	0.1224	2.3902	0.0432	2.7233	0.0398	1.4322	0.0524	1.6902	0.0421
Test6	3	1.9660	0.1385	1.6290	0.0106	1.2989	0.0059	6.1800	1.1015	2.8611	0.0526	3.2036	0.0589	1.6889	0.0104	2.1242	0.2832
	4	2.4218	0.4133	1.8088	0.0505	1.4759	0.0408	6.9712	0.6393	3.3392	0.0578	3.6636	0.0521	1.9647	0.0434	2.2197	0.1223
	5	2.3804	0.0647	1.9715	0.0798	1.6496	0.0134	7.9631	0.5364	3.7833	0.0603	4.1385	0.0744	2.2250	0.0193	2.4085	0.0111
	2	1.6508	0.0389	1.4653	0.0174	1.1474	0.0129	4.6406	0.1717	2.4080	0.0265	2.7149	0.0565	1.4239	0.0088	1.6278	0.0276
	3	1.6223	0.3100	1.6359	0.0223	1.2957	0.0103	5.7843	0.2296	2.8807	0.0516	3.2046	0.0654	1.6915	0.0091	1.8768	0.0087
Test7	4	1.5012	0.0251	1.7949	0.0165	1.4612	0.0147	6.9746	0.4586	3.3317	0.0589	3.6786	0.0564	1.9564	0.0081	2.1521	0.0286
	5	1.6493	0.0247	1.9579	0.0489	1.6279	0.0087	8.1870	0.6535	3.7890	0.0602	4.1877	0.1078	2.2341	0.0489	2.4214	0.0153
	2	1.1773	0.0108	1.4684	0.0312	1.1730	0.0783	4.7515	0.3404	2.4091	0.0473	2.7585	0.0877	1.4329	0.0350	1.6820	0.0258
	3	1.3368	0.0175	1.6569	0.0738	1.3021	0.0089	5.8659	0.4065	2.8756	0.0595	3.1828	0.0461	1.6992	0.0259	1.8800	0.0121
	4	1.4937	0.0164	1.8055	0.0148	1.4612	0.0084	7.2888	0.7825	3.3541	0.0717	3.6648	0.0561	1.9729	0.0473	2.1642	0.0490
Test8	5	1.6452	0.0143	1.9588	0.0209	1.6402	0.0323	7.9718	0.8481	3.8000	0.0448	4.1404	0.0532	2.2289	0.0335	2.4212	0.0132
	2	1.1792	0.0145	1.4638	0.0253	1.1688	0.0356	4.6434	0.2156	2.4093	0.0319	2.7143	0.0615	1.4242	0.0117	1.6153	0.0184
	3	1.3268	0.0140	1.6420	0.0281	1.3144	0.0114	5.6312	0.1780	2.9070	0.0762	3.1809	0.0472	1.6948	0.0323	1.8928	0.0300
	4	1.5231	0.0481	1.8089	0.0450	1.4637	0.0088	6.5840	0.1660	3.3252	0.0456	3.6484	0.0401	1.9580	0.0451	2.2089	0.0905
	5	1.6458	0.0179	1.9633	0.0425	1.6328	0.0349	7.5668	0.2257	7.9464	1.1175	4.1219	0.0588	2.2223	0.0184	2.4248	0.0456

Table 25

Comparison of P and H values obtained from the Wilcoxon signed-rank test between the pairs of the proposed MPA-OBL vs LSHADE_SPACMA-OBL, MPA-OBL vs CMA_ES-OBL, MPA-OBL vs DE-OBL, MPA-OBL vs HHO-OBL, MPA-OBL vs SCA-OBL, MPA-OBL vs SSA-OBL, and MPA-OBL vs MPA for Kapur's method in terms of fitness results.

Test image	nTh	LSHADE_SPACMA-OBL		CMA_ES-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA	
		p	h	p	h	p	h	p	h	p	h	p	h		
Test1	2	1.21E-12	1	1.21E-12	1	1.19E-07	1	1.68E-08	1	4.52E-12	1	NaN	0	1.21E-12	1
	3	1.21E-12	1	1.21E-12	1	1.19E-07	1	1.21E-12	1	1.21E-12	1	0.000308737	1	1.21E-12	1
	4	1.21E-12	1	1.21E-12	1	1.19E-07	1	1.21E-12	1	1.21E-12	1	6.10E-10	1	1.21E-12	1
	5	1.21E-12	1	1.21E-12	1	4.16E-14	1	4.57E-12	1	1.21E-12	1	1.63E-11	1	1.21E-12	1
	2	1.21E-12	1	1.21E-12	1	1.69E-14	1	1.70E-08	1	1.93E-10	1	2.92E-05	1	1.21E-12	1
Test2	3	1.21E-12	1	1.21E-12	1	4.42E-13	1	4.57E-12	1	1.21E-12	1	4.51E-12	1	1.21E-12	1
	4	1.21E-12	1	1.21E-12	1	2.71E-14	1	1.66E-11	1	1.21E-12	1	1.19E-12	1	1.66E-11	1
	5	1.21E-12	1	1.21E-12	1	2.71E-14	1	1.21E-12	1	1.21E-12	1	1.21E-12	1	1.21E-12	1
	2	1.21E-12	1	1.21E-12	1	1.69E-14	1	1.27E-05	1	5.76E-11	1	4.77E-08	1	1.27E-05	1
	3	1.01E-11	1	1.01E-11	1	2.90E-13	1	1.07E-07	1	2.14E-09	1	2.14E-09	1	1.07E-07	1
Test3	4	1.21E-12	1	1.21E-12	1	5.63E-13	1	1.21E-12	1	1.21E-12	1	1.93E-10	1	1.21E-12	1
	5	1.21E-12	1	1.21E-12	1	2.71E-14	1	1.21E-12	1	1.21E-12	1	1.93E-10	1	1.21E-12	1
	2	1.21E-12	1	1.21E-12	1	1.69E-14	1	6.22E-10	1	1.65E-11	1	NaN	0	1.21E-12	1
	3	1.21E-12	1	1.21E-12	1	1.69E-14	1	1.21E-12	1	1.21E-12	1	0.000658686	1	1.21E-12	1
	4	1.21E-12	1	1.21E-12	1	1.69E-14	1	1.21E-12	1	1.21E-12	1	3.39E-07	1	1.21E-12	1
Test4	5	1.21E-12	1	1.21E-12	1	1.69E-14	1	1.21E-12	1	1.21E-12	1	1.93E-10	1	1.21E-12	1
	2	1.21E-12	1	1.21E-12	1	1.69E-14	1	3.44E-07	1	5.75E-11	1	2.92E-05	1	1.21E-12	1
	3	1.21E-12	1	1.21E-12	1	1.69E-14	1	1.66E-11	1	1.21E-12	1	4.49E-12	1	1.21E-12	1
	4</														

Table 25 (continued).

Test image	nTh	LSHADE_SPACMA-OBL		CMA_ES-OBL		DE-OBL		HHO-OBL		SCA-OBL		SSA-OBL		MPA	
		p	h	p	h	p	h	p	h	p	h	p	h	p	h
Test10	5	1.21E-12	1	1.21E-12	1	4.16E-14	1	1.21E-12	1	1.21E-12	1	1.21E-12	1	1.21E-12	1
	2	1.21E-12	1	1.21E-12	1	2.43E-13	1	0.000661402	1	4.54E-12	1	NaN	0	0.000661402	1
	3	1.21E-12	1	1.21E-12	1	2.71E-14	1	1.93E-10	1	1.21E-12	1	0.000656249	1	1.93E-10	1
	4	2.37E-12	1	2.37E-12	1	4.16E-14	1	2.37E-12	1	2.96E-12	1	6.26E-12	1	2.37E-12	1
	5	1.21E-12	1	1.21E-12	1	2.71E-14	1	1.21E-12	1	1.21E-12	1	4.57E-12	1	1.21E-12	1

References

- [1] M.H. Merzban, M. Elbayoumi, Efficient solution of Otsu multilevel image thresholding: A comparative study, *Expert Syst. Appl.* 116 (2019) 299–309.
- [2] E. Rodríguez-Esparza, L.A. Zanella-Calzada, D. Oliva, A.A. Heidari, D. Zaldivar, M. Pérez-Cisneros, L.K. Foong, An efficient Harris Hawks-inspired image segmentation method, *Expert Syst. Appl.* (2020) 113428.
- [3] S. Aja-Fernández, A.H. Curiale, G. Vegas-Sánchez-Ferrero, A local fuzzy thresholding methodology for multiregion image segmentation, *Knowl.-Based Syst.* 83 (2015) 1–12.
- [4] H. Gao, Z. Fu, C.-M. Pun, H. Hu, R. Lan, A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm, *Comput. Electr. Eng.* 70 (2018) 931–938.
- [5] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.* 9 (1) (1979) 62–66.
- [6] J.N. Kapur, P.K. Sahoo, A.K. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, *Comput. Vis. Graph. Image Process.* 29 (3) (1985) 273–285.
- [7] E.H. Houssein, M.R. Saad, F.A. Hashim, H. Shaban, M. Hassaballah, Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems, *Eng. Appl. Artif. Intell.* 94 (2020) 103731.
- [8] E.H. Houssein, M.A. Mahdy, M.G. Eldin, D. Shebl, W.M. Mohamed, M. Abdel-Aty, Optimizing quantum cloning circuit parameters based on adaptive guided differential evolution algorithm, *J. Adv. Res.* (2020).
- [9] G. Dhiman, V. Kumar, Emperor penguin optimizer: a bio-inspired algorithm for engineering problems, *Knowl.-Based Syst.* 159 (2018) 20–50.
- [10] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, *Knowl.-Based Syst.* 165 (2019) 169–196.
- [11] S. Kaur, L.K. Awasthi, A. Sangal, G. Dhiman, Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization, *Eng. Appl. Artif. Intell.* 90 (2020) 103541.
- [12] E.H. Houssein, M.R. Saad, K. Hussain, W. Zhu, H. Shaban, M. Hassaballah, Optimal sink node placement in large scale wireless sensor networks based on Harris' hawk optimization algorithm, *IEEE Access* 8 (2020) 19381–19397.
- [13] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, A modified Henry gas solubility optimization for solving motif discovery problem, *Neural Comput. Appl.* 32 (14) (2020) 10759–10771.
- [14] E.H. Houssein, M.E. Hosney, D. Oliva, W.M. Mohamed, M. Hassaballah, A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery, *Comput. Chem. Eng.* 133 (2020) 106656.
- [15] N. Neggaz, E.H. Houssein, K. Hussain, An efficient henry gas solubility optimization for feature selection, *Expert Syst. Appl.* (2020) 113364.
- [16] E. Cuevas, J. Gálvez, O. Avalos, Introduction to optimization and metaheuristic methods, in: *Recent Metaheuristics Algorithms for Parameter Identification*, Springer, 2020, pp. 1–8.
- [17] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, *Artif. Intell. Rev.* 52 (4) (2019) 2191–2233.
- [18] G. Dhiman, A. Kaur, STOA: a bio-inspired based optimization algorithm for industrial engineering problems, *Eng. Appl. Artif. Intell.* 82 (2019) 148–174.
- [19] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.* 97 (2019) 849–872.
- [20] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: A novel physics-based algorithm, *Future Gener. Comput. Syst.* 101 (2019) 646–667.
- [21] F.A. Hashim, K. Hussain, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, *Appl. Intell.* (2020) 1–21.
- [22] J.H. Holland, Genetic algorithms, *Sci. Am.* 267 (1) (1992) 66–73.
- [23] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (4) (1997) 341–359.
- [24] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, IEEE, 1995, pp. 39–43.
- [25] E. Rashedi, H. Nezamabadi-Pour, S. Saryzadi, GSA: a gravitational search algorithm, *Inform. Sci.* 179 (13) (2009) 2232–2248.
- [26] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [27] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Engrg.* 376 (2021) 113609.
- [28] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M. Al-qaness, A.H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.* (2021).
- [29] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.* 89 (2015) 228–249.
- [30] Y. Djenouri, A. Belhadi, R. Belkebir, Bees swarm optimization guided by data mining techniques for document information retrieval, *Expert Syst. Appl.* 94 (2018) 126–136.
- [31] A. Faramarzi, M. Heidarnejad, S. Mirjalili, A.H. Gandomi, Marine predators algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.* (2020) 113377.
- [32] E.H. Houssein, B. El-dinHelmy, D. Oliva, A. A. Elngar, H. Shabana, A novel Black Widow Optimization algorithm for multilevel thresholding image segmentation, *Expert Syst. Appl.* - (2020) 114159–114176.
- [33] K.G. Dhal, A. Das, S. Ray, J. Gálvez, S. Das, Nature-inspired optimization algorithms and their application in multi-thresholding image segmentation, *Arch. Comput. Methods Eng.* (2019) 1–34.
- [34] D. Oliva, M.A. Elaziz, S. Hinojosa, Multilevel thresholding for image segmentation based on metaheuristic algorithms, in: *Metaheuristic Algorithms for Image Segmentation: Theory and Applications*, Springer, 2019, pp. 59–69.
- [35] G. Dhiman, M. Garg, A. Nagar, V. Kumar, M. Dehghani, A novel algorithm for global optimization: Rat swarm optimizer, *J. Ambient Intell. Humaniz. Comput.* (2020) 1–26.
- [36] G. Dhiman, D. Oliva, A. Kaur, K.K. Singh, S. Vimal, A. Sharma, K. Cengiz, BEPO: a novel binary emperor penguin optimizer for automatic feature selection, *Knowl.-Based Syst.* 211 (2021) 106560.
- [37] P. Upadhyay, J.K. Chhabra, Kapur's entropy based optimal multilevel image segmentation using crow search algorithm, *Appl. Soft Comput.* (2019) 105522.
- [38] A.K.M. Khairuzzaman, S. Chaudhury, Multilevel thresholding using grey wolf optimizer for image segmentation, *Expert Syst. Appl.* 86 (2017) 64–76.
- [39] H. Liang, H. Jia, Z. Xing, J. Ma, X. Peng, Modified grasshopper algorithm-based multilevel thresholding for color image segmentation, *IEEE Access* 7 (2019) 11258–11295.
- [40] M. Abdel-Basset, V. Chang, R. Mohamed, A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems, *Neural Comput. Appl.* (2020) 1–34.
- [41] M.-A. Díaz-Cortés, N. Ortega-Sánchez, S. Hinojosa, D. Oliva, E. Cuevas, R. Rojas, A. Demin, A multi-level thresholding method for breast thermograms analysis using Dragonfly algorithm, *Infrared Phys. Technol.* 93 (2018) 346–361.
- [42] N. Rojas-Morales, M.-C.R. Rojas, E.M. Ureta, A survey and classification of opposition-based metaheuristics, *Comput. Ind. Eng.* 110 (2017) 424–435.
- [43] H.R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, 1, IEEE, 2005, pp. 695–701.
- [44] F. Chakraborty, P.K. Roy, D. Nandi, Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding, *Evol. Intell.* 12 (3) (2019) 445–467.
- [45] S. Rahnamayan, H.R. Tizhoosh, Image thresholding using micro opposition-based differential evolution (micro-ode), in: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, IEEE, 2008, pp. 1409–1416.
- [46] A.W. Mohamed, A.A. Hadi, A.M. Fattouh, K.M. Jambi, LSHADE With semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, in: *2017 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2017, pp. 145–152.
- [47] I. Loshchilov, CMA-ES With restarts for solving CEC 2013 benchmark problems, in: *2013 IEEE Congress on Evolutionary Computation*, IEEE, 2013, pp. 369–376.
- [48] S. Mirjalili, SCA: A Sine cosine algorithm for solving optimization problems, *Knowl.-Based Syst.* 96 (2016) 120–133.

- [49] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Softw.* 114 (2017) 163–191.
- [50] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, A. Cosar, A survey on new generation metaheuristic algorithms, *Comput. Ind. Eng.* 137 (2019) 106040.
- [51] E. Aarts, E.H. Aarts, J.K. Lenstra, *Local Search in Combinatorial Optimization*, Princeton University Press, 2003.
- [52] S. Gupta, K. Deep, A.A. Heidari, H. Moayedi, M. Wang, Opposition-based learning Harris Hawks optimization with advanced transition rules: Principles and analysis, *Expert Syst. Appl.* (2020) 113510.
- [53] S. Gupta, K. Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, *Expert Syst. Appl.* 119 (2019) 210–230.
- [54] A.A. Ewees, M.A. Elaziz, E.H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, *Expert Syst. Appl.* 112 (2018) 156–172.
- [55] T.K. Sharma, M. Pant, Opposition based learning ingrained shuffled frog-leaping algorithm, *J. Comput. Sci.* 21 (2017) 307–315.
- [56] W. Long, J. Jiao, X. Liang, S. Cai, M. Xu, A random opposition-based learning grey wolf optimizer, *IEEE Access* 7 (2019) 113810–113825.
- [57] D. Bairathi, D. Gopalani, Opposition based salp swarm algorithm for numerical optimization, in: *International Conference on Intelligent Systems Design and Applications*, Springer, 2018, pp. 821–831.
- [58] C.A. Glasbey, An analysis of histogram-based thresholding algorithms, *CVGIP: Graph. Models Image Process.* 55 (6) (1993) 532–537.
- [59] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, N.H. Awad, Evaluating the performance of adaptive gainsharing knowledge based algorithm on CEC 2020 benchmark problems, in: *2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2020, pp. 1–8.
- [60] Q. Huynh-Thu, M. Ghanbari, Scope of validity of PSNR in image/video quality assessment, *Electron. Lett.* 44 (13) (2008) 800–801.
- [61] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.
- [62] U. Sara, M. Akter, M.S. Uddin, Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study, *J. Comput. Commun.* 7 (3) (2019) 8–18.
- [63] C. Liao, S. Li, Z. Luo, Gene selection using wilcoxon rank sum test and support vector machine for cancer classification, in: *International Conference on Computational and Information Science*, Springer, 2006, pp. 57–66.
- [64] S.W. Scheff, Chapter 8 - nonparametric statistics, in: S.W. Scheff (Ed.), *Fundamental Statistical Principles for the Neurobiologist*, Academic Press, 2016, pp. 157–182.