Artificial Intelligence Review (2022) 55:3315–3367
https://doi.org/10.1007/s10462-021-10086-0

Hybrid marine predators algorithm for image segmentation: analysis and validations

Mohamed Abdel-Basset[1] · Reda Mohamed[1] · Mohamed Abouhawwash[2,3]

Published online: 25 October 2021

Abstract
Naturally, to analyze an image accurately, all the similar objects within it should be separated to pay attention to the most important object for reaching more details and hence achieving better accuracy. Therefore, multilevel thresholding is an indispensable image processing technique in the field of image segmentation and is employed widely to separate those similar objects. However, with increasing thresholds, the existing image segmentation techniques might suffer from exponentially-grown computational cost and low accuracy due to local optima shortage. Therefore, in this paper, a new image segmentation algorithm based on the improved marine predators algorithm (MPA) is proposed. MPA is improved using a strategy to find a number of the worst solutions within the population then tries to search for other better ones for those solutions by moving them gradually towards the best solutions to avoid accelerating to local optima and randomly within the search space based on a certain probability. In addition, this number of the worst solutions is increased with the iteration. This strategy is known as the linearly increased worst solutions improvement strategy (LIS). Also, we suggested that apply the ranking strategy based on a novel updating scheme, namely ranking-based updating strategy (RUS), on the solutions that could find better solutions in the last number iterations, perIter, in the hope of finding better solutions near it. RUS updates the particles/solutions which could not find better solutions than the best-local one in a number of consecutive iterations, with those that are generated based on a novel updating strategy. LIS is integrated with MPA to produce a new segmentation meta-heuristic algorithm abbreviated as MPALS. Also, MPALS and RUS are combined to tackle ISP in a strong variant abbreviated as HMPA for overcoming the image segmentation problem. The two proposed algorithms are validated on 14 test images and compared with seven state-of-the-arts meta-heuristic algorithms. The experimental results show the effectiveness of HMPA with increasing the threshold levels compared to the seven state-of-the-arts algorithms when segmenting an image, while their performance is roughly the same for the image with a small threshold level.

Keywords Image segmentation · Marine predators algorithm (MPA) · Linearly · Ranking-based Local Minima · Kapur's entropy

✉ Mohamed Abouhawwash
saleh1284@mans.edu.eg; abouhaww@msu.edu

Extended author information available on the last page of the article

1 Introduction

Reaching better accuracy when analyzing an image is considered an indispensable objective, but the image may be crowded with objects that are not beneficial for this analysis, and subsequently, the accuracy will reduce significantly due to paying attention to some unimportant regions within the analyzing process. As a result, separating the similar regions within the image is a common problem known as image segmentation problem (ISP) that the researchers' pit for overcoming to extract the desired regions that improve the analysis accuracy of the image in several fields such as historical newspapers (Naoum et al. 2019; Barman et al. 2020), satellite image processing (Karydas 2020), object recognition (Wang et al. 2020), and medical diagnosis (Mittal et al. 2020; Zhang et al. 2020; Sultana et al. 2020; Hassanzadeh et al. 2020).

To overcome ISP, several segmentation techniques under region-based (Aksac et al. 2017), feature selection-based clustering (Narayanan et al. 2019), edge-based (Prathusha and Jyothi 2018), and threshold-based (Han et al. 2017) have been suggested. From among those techniques, threshold-based segmentation is the simplest, fastest, and accurate. As a result, threshold-based segmentation is significantly utilized to tackle ISP (Kuruvilla et al. 2016; Oliva et al. 2014; Arora et al. 2008). Threshold-based segmentation has two categories: bi-level threshold, and multi-level threshold. In the bi-level, the image is divided into two parts: background and foreground (object). When having more than two similar regions within the image, the bi-level is skipped, and comes the role of the multi-level threshold that appeared to separate the several similar regions within the image. But although the high benefits that multi-level threshold could come true to overcome ISP for an image with several regions, it needs time increased exponentially with the required threshold number.

Some of the threshold techniques are based on a parametric approach that needs to compute some parameters under the probability density function for each region to extract the optimal threshold values. While the other techniques use a non-parametric approach that seeks to maximize some functions such as Kapur's entropy (Kapur et al. 1985), fuzzy entropy (Li et al. 2021) and Otsu function (Otsu 1979) without going to computing some parameters.

Since the multi-level threshold needs time increased exponentially with the threshold levels, the traditional techniques are inadequate for overcoming those levels especially for an image with an extremely significant level. Consequently, the researchers think in the meta-heuristic algorithms as another methodology to overcome the drawbacks of the traditional techniques due to the great success that could achieve in many fields (Abdel-Basset et al. 2018; Sayed et al. 2019; Abouhawwash and Alessio 2021; Ma et al. 2021, 2020) within less possible time. Several meta-heuristic algorithms for tackling ISP has recently been proposed, some of which will be reviewed within the next section.

Recently, a new metaheuristic algorithm known as the marine predators algorithm (MPA), which mimics the behaviors of the predators when attacking their prey, has been proposed for tackling the global optimization problem and could fulfill superior performance (Faramarzi et al. 2020). As a result, it has been applied for tackling several real optimization problems. In Soliman et al. (2020), MPA was adapted for finding the unknown parameters of triple-diode photovoltaic models and could fulfill superior outcomes compared to four other metaheuristics algorithms. It has been also applied for finding the optimal thresholds of an image by maximizing the fuzzy entropy type II as an objective function (Mahajan et al. 2021). Moreover, the MPA integrated with dominance strategy-based exploration and exploitation was applied for tackling the multiobjective optimization problems and could outperform several well-known multiobjective optimization algorithms (Abdel-Basset et al. 2021). Both MPA and political optimizers (PO) were applied for finding the unknown parameters of fuel cells and the experimental findings show that MPA is better than PO and some of the compared metaheuristic algorithms (Diab et al. 2020). Compared to some of the evolutionary algorithms like genetic algorithm and differential evolution, in addition to the particle swarm optimization algorithm as one of the well-known swarm-based optimization algorithms, MPA could fulfill terrible success for tackling several optimization problems due to having some strong characteristics which have aided avoiding stuck into local minima by exploring several regions within the search space in less number of function evaluations to increase the convergence speed (Faramarzi et al. 2020). Moreover, MPA has additional merit, increasing the exploration capability, known as fish aggregating devices (FADs) that have employed various updating manners to reposition each solution to other regions in the search space for preserving the solution diversity within various optimization phases.

Furthermore, MPA was integrated with Success History based Adaptive Differential Evolution (SHADE) algorithm to utilize the features of each one for producing a new strong variant called hybrid Marine Predators—Success History based Adaptive Differential Evolution (MP-SHADE) algorithm for estimating the unknown parameters of the single and double diode photovoltaic models. A combination of both MPA and teaching and learning-based optimization could produce a new strong one having higher exploration capability to avoid stuck into local minima as an attempt to find better outcomes. This variant was abbreviated as TLMPA and applied for tackling IEEE CEC-2017 benchmark functions and four engineering design problems. Many other variants of MPA have been recently published for tackling other real optimization problems: parameter identification of single and double diode models (Ridha 2020), a new configuration of an autonomous CHP system based on improved MPA (Wang et al. 2021), Multi-Regional Optimal Power Flow (Swief et al. 2021), and several else (Kapur et al. 1985; Yu et al. 2021; Panagant et al. 2021; Shaheen et al. 2021, 2020; Durmus 2021; Liu and Yang 2021; Elsayed et al. 2021; Ramezani et al. 2021; Riad et al. 2021; Ghoneimy et al. 2021).

Within our work, we try to support another technique that tries to exploit the individuals of the population significantly within the optimization process. This technique is based on selecting a number of the individuals with the worst fitness values, this number increases linearly with the iteration, then each individual within those individuals will be updated either with a small step size, which increases gradually with increasing the current iteration, toward the best-so-far one to explore a huge number of solutions between this worst and the best-so-far one, in addition to avoiding stuck into local minima at the start of the optimization process, or randomly within the search space to avoid stuck into local minima, the trade-off between those two different updates: toward the best-so-far and randomly within the search space is based on a certain probability illustrated within the experiments section in the parameter tuning. . After that, due to the significant success achieved by MPA in several fields as mentioned before, in addition to its need for more improvements to balance between the exploration and exploitation operators, it has been used to be integrated with this technique to propose a new variant, MPALS, having a strong exploitation operator for tackling ISP with threshold levels up to 40. However, this variant still suffers from falling into local minim because of the low exploration operator, therefore, another strategy known as ranking-based updating strategy (RUS) has been proposed with a novel updating scheme to replace the unbeneficial solutions which could not come true better solution within a consecutive number perIter of the iterations with those which improve both exploration and exploitation operators of the optimization algorithms. then, MPALS is combined with RUS to develop a new variant called HMPA. Both HMPA and MPALS were compared with a number of recently proposed well-established optimization algorithms. From involving those algorithms, the hybrid marine predators algorithm with ranking-based diversity strategy that is recently proposed for tackling ISP for covid19 images. After completing the comparison, it was notified that HMPA is superior on significant threshold levels and converged with the small threshold levels. The main contributions introduced within this paper are:

1. Proposing two strategies known as linearly increased the worst solutions improvement strategy and ranking-based updating strategy (RUS) to utilize each individual within the optimization process as possible for reaching better outcomes.
2. Integrating these strategies with MPA to propose two variants: the first is based on integrating MPA with LIS (MPALS), and the second improves MPALS by RUS (HMPA) for tackling ISP with threshold levels reaching 40.
3. Those two variants were extensively validated on 14 test images taken from Berkeley Segmentation Dataset, and compared with a number of recently proposed optimization algorithms to show the superiority of the proposed.
4. After validation and comparison, we see that HMPA could be converged with the small threshold levels and significantly outperform with the high threshold levels.

Within the following sections in this paper: some of the previous works done on ISP are reviewed in Sect. 2, the Kapur's entropy is described in Sects. 3, 4 overviews the marine predators algorithm. Section 5 designs the paces of developing MPA with a linear population improvement strategy for tackling ISP. Section 6 validates and compares the proposed algorithm on some test images, and Sect. 7 shows the conclusions about our proposition and future works.

2 Literature review

There are several meta-heuristic algorithms proposed for tackling ISP such as, ant colony optimization algorithm (Kaveh and Talatahari 2010), whale optimization algorithm (WOA) (Abd El Aziz et al. 2017), multi-verse optimizer (Kandhway and Bhandari 2019), particle swarm optimization (PSO) (Guo and Li 2007; Xiong et al. 2020; Di Martino and Sessa 2020), cuckoo search (CS) (Agrawal et al. 2013), locust search algorithm (LSA) (Cuevas et al. 2020), honey bee mating optimization (HBM) (Horng 2010), symbiotic organisms search (SOS) (Chakraborty et al. 2019), harris hawk optimization algorithm (HHA) (Bao et al. 2019), and moth-flame optimization (MFA) (Abd El Aziz et al. 2017), flower pollination algorithm (FPA) (Wang et al. 2015), crow search algorithm (Oliva et al. 2017), an improved grey wolf optimizer (IGWO) (Yao et al. 2019), genetic algorithm (GA) (Elsayed et al. 2014), bee colony algorithm (BCA) (Huo et al. 2020), marine predators algorithm (MPA) and improved MPA (IMPA) (Abdel-Basset et al. 2020a), equilibrium optimizer (EO) (Abdel-Basset et al. 2020c), bacterial Foraging Algorithm (BFA) (Sanyal et al. 2011), and firefly optimization algorithm (FFA) (Erdmann et al. 2015). Through this section, some of this algorithm will be surveyed briefly.

Abd El Aziz et al. (2017) developed both WOA and MFA for overcoming ISP by maximizing the Otsu's method on small threshold levels reaching 6, but its performance on significant threshold levels is not known even now as its main limitations. In addition, Agrawal et al. (2013) proposed CS for overcoming ISP by using the tsallis entropy as a fitness function. The CS was verified a standard benchmark of test images, and compared with 4 optimization algorithms: artificial bee colony (ABC) algorithm, particle swarm optimization (PSO) , bacteria foraging optimization (BFO) and genetic algorithm (GA). According to the discussion of the authors, the CS was competitive with those algorithms in terms of only the CPU time and the objective function. This algorithm has been investigated for only 5-threshold levels, higher than that, its performance is not known as its main shortage. Chakraborty et al. (2019) proposed SOS enhanced by the opposition-based learning to increase the convergcne speed toward the optimal solution and avoid the local minima that may deteriorite the perforamcne of SOS for tackling ISP of color images. The performance of the improved SOS was validated using a set of the color images taken from Berkeley Segmentation Dataset (BSDS) and another set gathered for the COCO dataset. The experimental results of the improved SOS in comparison to a number of the existing algorithms: Bat algorithm (BA), CS, PSO, and ABC show the superiority of the improved SOS in terms of the objective values. Also, the performance of this algorithm is not known with increasing threshold levels and subsequently not preferred for tackling any image with a significant number of threshold levels.

Furthermore, Bhandari et al. (2015) proposed the modified ABC to find the optimal threshold values for the satellite image segmentation using different objective functions: Kapur's, Otsu and Tsallis. In the modified ABC (MABC), the chaotic maps and opposition-based learning was employed during generating the initial population to improve the convergence speed. This modified version was compared with the standard ABC, PSO and GA under various objective functions, but this algorithm was limited in terms of CPU runtime and algorithm complexity compared to those compared algorithms. In Erdmann et al. (2015), FFA was proposed to tackle ISP, but its performance was weak, so the improved one (IFFA) (Chen et al. 2016), that was improved using the Cauchy mutation and neighborhood strategy to overcome the local minima and improve its exploration capability, has been developed.

Maitra and Chatterjee (2008) improved PSO with cooperative and comprehensive learning for overcoming the dimensionality curse and increasing the early convergence, respectively. Also, PSO (Liu et al. 2015) modified using adaptive inertia, and the population has been proposed for tackling ISP. BFA (Sanyal et al. 2011) has been proposed for tackling ISP of grey images by using fuzzy entropy to alter the bacterium between intensification and diversification operators. Furthermore, BFA (Tang et al. 2017) is integrated with PSO to support the global search capability in addition to the weak bacterium, which selects a random strong one to reach a location near it. In Yao et al. (2019), an improved grey wolf optimizer (IGWO) has been proposed for tackling ISP. IGWO was improved using a good point set method to initialize the population, this algorithm has a high ability to get rid of local optima and finding better solutions. This algorithm has not discussed the CPU runtime in addition to using a small number of threshold levels up to 5 and consequently, it is not a good alternative to the existing image segmentation techniques.

An improved Bat algorithm (IBA) (Mokhtari and Kimour 2019) was proposed for tackling ISP. IBA accelerated the convergence and increased the diversity between the members of the population until disposing of stuck into local minima using both the crossover operator and chaotic search, respectively. In Xu et al. (2019), A dragonfly algorithm (DA) and differential evolution (DE) have been proposed for tackling the color image segmentation problem. DA has a high ability on getting out of local minima so it can reach a better solution, in addition to integrating DE as a local search strategy to improve the precision of the solution. A modified spherical search optimizer (MSSO) (Naji Alwerfali et al. 2020) has been proposed and modified by the sine-cosine algorithm (SCA) to increase the exploitation capability of this algorithm.

In Abdel-Basset et al. (2020c), equilibrium optimizer (EO) was adapted for tackling the ISP by maximizing the Kapur's entropy. EO was verified using a collection of the test images extracted from Berkeley Segmentation Dataset. In addition, EO was compared with a number of the popular optimization algorithms, such as WOA, BA, SCA, salp swarm algorithm, Harris hawks algorithm, crow search algorithm, and PSO to see its efficacy under a set of performance metrics. EO still suffers from falling into local minima and low convergence speed which stand as an immune obstacle front reaching the optimal threshold levels. Furthermore, in Abdel-Basset et al. (2020a) the standard MPA in addition to an improved one by ranking based diversity reduction strategy has been suggested for segmenting the x-ray images infected by covid19 under the Kapur's entropy. The standard MPA was improved using this novel method to accelerate the convergence speed toward the best-so-far solution as an attempt to find better outcomes. This improved MPA was compared with a number of the recent algorithms to see its efficacy. The experimental results proved its efficacy over the standard one and the compared ones. Also, as discussed in the experiments later that this improved MPA had two shortcomings: local minima and low convergence speed.

Chouksey et al. (2020) has investigated the performance of the antlion optimization (ALO) and multiverse optimization (MVO) algorithms for tackling the multilevel image segmentation problem based on two objective functions: kapur's entropy and Otsu method. Those two algorithms were compared with evolutionary algorithms under the objective value, the stability, feature similarity index (FSIM), peak signal to noise ratio(PSNR), and structural similarity index (SSIM), and CPU time. The experiments show that those algorithms could come true better outcomes over the compared algorithm, in general. Especially, the MVO could be more converged than ALO. However, those algorithms have not experimented with thresholds greater than 5 as their main limitation. Shahabi et al. (2019) proposed the crow search algorithm (CSA) for estimating the optimal threshold values using the Otsu function. This algorithm was compared the improved PSO, FFA, the fuzzy version of FFA to see its efficacy. The experimental outcomes proved the efficacy of the CSA over those algorithms for time and uniformity.

Further, the water wave optimization (WWO) algorithm modified by opposition based learning strategy and ranking-based mutation strategy to improve both the diversity of the individuals within the population and the selection probability, respectively, has been proposed by Yan et al. (2020) to select the optimal threshold values for the underwater ISP. The modified WWO (MWWO) was compared with the other algorithms based on the PSNR, the SSIM, the CPU time, the objective values, and the Wilcoxon rank sum test. The experimental outcomes show the superiority of the MWWO over the other algorithms. However, its performance for threshold levels higher than 6 has not been investigated, and hence when increasing the threshold levels, this algorithm is not preferred for tackling this problem.

3 Kapur's entropy

Within this section, we will describe the Kapur's entropy used to extract the optimal threshold values by maximizing the entropy of the segmented regions [14]. Let's see the mathematical model of this method, supposing that the threshold values that segment an image with k similar regions are $t_0, t_1, t_2, \ldots$,and t_k, then the Kapur's entropy seek for maximizing the following formula until reaching the optimal threshold values:

$$T(t_0, t_1, t_2, \ldots, t_n) = T_0 + T_1 + T_2 + \cdots + T_n \tag{1}$$

$$T_0 = -\sum_{i=0}^{t_0-1} \frac{X_i}{W_0} * ln\frac{X_i}{W_0}, X_i = \frac{N_i}{W}, W_0 = \sum_{i=0}^{t_1-1} X_i \tag{2}$$

$$T_1 = -\sum_{i=t_0}^{t_1-1} \frac{X_i}{W_1} * ln\frac{X_i}{W_1}, X_i = \frac{N_i}{W}, W_1 = \sum_{i=t_0}^{t_1-1} X_i \tag{3}$$

$$T_2 = -\sum_{i=t_1}^{t_2-1} \frac{X_i}{W_2} * ln\frac{X_i}{W_2}, X_i = \frac{N_i}{W}, W_2 = \sum_{i=t_1}^{t_2-1} X_i \tag{4}$$

$$T_k = -\sum_{i=t_k}^{L-1} \frac{X_i}{W_k} * ln\frac{X_i}{W_k}, X_i = \frac{N_i}{W}, W_k = \sum_{i=t_k}^{L-1} X_i \tag{5}$$

$T_0, T_1, T_2, \ldots$, and T_k expresses the entropies of similar regions, and N_i refers to the count of pixels with a value equal to i. $W_0, W_1, W_2, \ldots$, and W_k is a phrase about the probabilities of the different regions in proportion to the whole pixel W within an image. Finally, Eq. 1 is used in our proposition as a fitness function to find the optimal threshold values until overcoming ISP.

4 Marine predators algorithm (MPA)

Recently, Faramarzi et al. (2020) proposed a novel meta-heuristic algorithm, namely marine predators algorithm (MPA), that mimics the behavior of the predators when attacking their prey. Specifically, the predators tradeoff between the $l\hat{e}vy's$ flight and Brownian strategy according to the velocity from the prey to predators when searching for their prey. Mathematically, MPA is formulated as follows:

Like most meta-heuristic algorithms, at the outset of the optimization process, the prey of size N will be distributed within the search space using the following equation:

$$\mathbf{prey} = \mathbf{X}_{min} + \mathbf{r} * (\mathbf{X}_{max} - \mathbf{X}_{min}) \tag{6}$$

Where $\mathbf{r}$ is a vector to contain numerical values generated randomly at the interval of 0 and 1, and $\mathbf{X}_{min}$, and $\mathbf{X}_{max}$ are two vectors contain the maximum and minimum boundaries of the search space for the problem.

After that, the fitness function is calculated and the prey with the highest fitness is selected as the Top predator within the optimization process to construct the Elite (E) matrix. This matrix is constructed as follows: $E = \begin{bmatrix} A^I_{1,1} & A^I_{1,2} & \ldots & A^I_{1,d} \\ A^I_{2,1} & A^I_{2,2} & \ldots & A^I_{2,d} \\ \ldots & \ldots & \ldots & \ldots \\ A^I_{N,1} & A^I_{N,2} & \ldots & A^I_{N,d} \end{bmatrix}$

$\mathbf{A}^I$is the top predator and repeated N times to build up the elite matrix. N indicates the individual numbers in the population, and d is dimension number in each individual. Then, within the optimization process, the predators are moved towards a prey matrix formulated as and initialized randomly within the search space:

$$prey = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,d} \\ A_{2,1} & A_{2,2} & \dots & A_{2,d} \\ \dots & \dots & \dots & \dots \\ A_{N,1} & A_{N,2} & \dots & A_{N,d} \end{bmatrix}$$

Within the optimization process, the updating of each prey will be divided into three stages according to the velocity ratio from the prey to the predators.

1. High-velocity ratio The velocity ratio from the prey to predators is high in this phase, where the prey is moved quickly searching for their food and the predators monitor their movements. Therefore, the predators in this phase don't need to move at all, because the prey will reach them itself. This phase occurs at the start of the optimization process, where the algorithm searches for better solution in all regions within the search space of the problem. This phase is mathematically formulated as follows:
   while $t < \frac{1}{3} * t_{max}$

$$\mathbf{S}_i = \mathbf{R}_B \otimes (\mathbf{E}_i - \mathbf{R}_B \otimes \mathbf{Prey}_i) \tag{7}$$

$$\mathbf{Prey}_i = \mathbf{Prey}_i + P.\mathbf{R} \otimes \mathbf{S}_i \tag{8}$$

   where $\mathbf{R}_B$ is a vector assigned randomly using the normal distribution to represent the Brownian strategy, $\otimes$ is the entry-wise multiplication, P is a fixed value and assigned 0.5 as recommended in the original paper, $\mathbf{R}$ is a vector generated randomly within 0 and 1, t indicates the current iteration, and t_{max} expresses the maximum iteration.
2. Unit velocity ratio.
   This stage occurs at the intermediate stage of the optimization process where the exploration is converted into the exploitation phase. So this stage is neither exploration nor exploration but it is a mix of them. Based on that, MPA within this phase divides the population into two halves: the first half will move using exploration steps while the other has exploitation steps. Finally, this stage will be mathematically formulated as follows:
   while $\frac{1}{3} * t_{max} < t < \frac{2}{3} * t_{max}$

- For the first half of the population

$$\mathbf{S}_i = \mathbf{R}_L \otimes (\mathbf{E}_i - \mathbf{R}_L \otimes \mathbf{Prey}_i) \tag{9}$$

$$\mathbf{Prey}_i = \mathbf{Prey}_i + P.\mathbf{R} \otimes \mathbf{S}_i \tag{10}$$

- For the second half of the population

$$\mathbf{S}_i = \mathbf{R}_B \otimes (\mathbf{R}_B \otimes \mathbf{E}_i - \mathbf{Prey}_i) \tag{11}$$

$$\mathbf{Prey}_i = \mathbf{E}_i + P * CF \otimes \mathbf{S}_i \tag{12}$$

where $\mathbf{R}_L$ is a vector generated randomly based on the *lévy* distribution, and CF is an adaptive parameter designed to manage the step size and created using the following equation:

$$CF = (1 - \frac{t}{t_{max}})^{2\frac{t}{t_{max}}} \tag{13}$$

3. Low velocity-ratio:
This stage occurs at the end of the optimization process where the exploration operator is completely converted into the exploitation operator and mathematically modeled as: while $t > \frac{2}{3} * t_{max}$

$$\mathbf{S}_i = \mathbf{R}_L \otimes (\mathbf{R}_L \otimes \mathbf{E}_i - \mathbf{Prey}_i) \tag{14}$$

$$\mathbf{Prey}_i = \mathbf{E}_i + P * CF \otimes \mathbf{S}_i \tag{15}$$

Some other factors such as eddy formulation and fish aggregating devices (FADs) affect significantly the behaviors of the predators. Based on some studies, as a result of FADs, the predators spend 20% of their search time exploring another environment around the search space with abundant prey, while the other time they search for better solution within the surrounding environment. FADs could be mathematically computed according to the following formula:

$$\mathbf{prey}_i = \begin{cases} \mathbf{prey}_i + CF[\mathbf{X}_{min} + \mathbf{r}_2(\mathbf{X}_{max} - \mathbf{X}_{min})] \otimes \mathbf{U} & if\ r < FADs \\ \mathbf{prey}_i + [FADs(1-r) + r](\mathbf{prey}_{r1} - \mathbf{prey}_{r2}) & if\ r \geqslant FADs \end{cases} \tag{16}$$

r is a number generated randomly within 0 and 1. Where, $\mathbf{r}_2$ indicates the index of a prey selected randomly from the population. $\mathbf{U}$ is a binary vector including 0 and 1. FADs=0.2 indicates the probability of impacting FADs on the optimization process. r_1 indicates the index of a prey selected randomly from the population.

After each updating process, MPA compares the fitness of the updated solutions with the fitness for the previous solution to see if this update improves in the positions or not. If the updated position of each solution is better than the old one, the updated will be stored to be compared with the next generation, and if the old is better than the updated one then the old solution will be used within the next generation instead of the updated one. This process is known as memory saving. The pseudo-code of MPA is exposed in Algorithm 1, and the same steps are also shown in Fig. 1. In this pseudo-code and figure, $t++$ is the same as $t = t + 1$ and is responsible for moving the algorithm to the next generation for satisfying the termination condition related to a maximum generation.

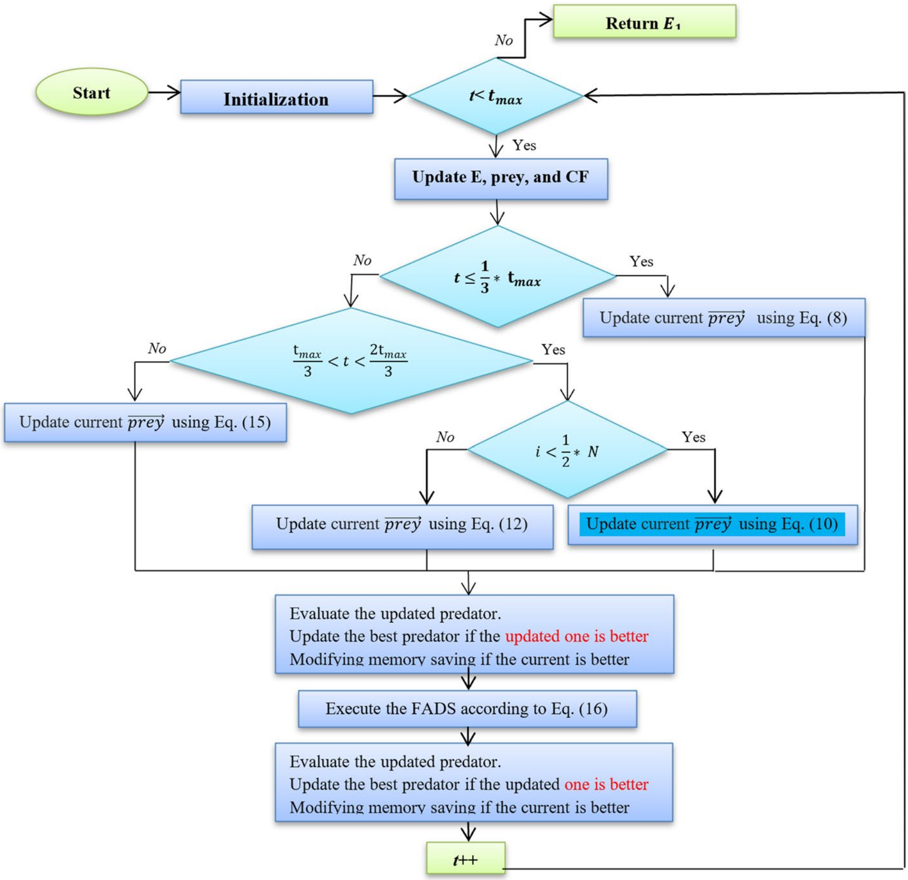


Fig. 1 Depiction of the MPA steps

Algorithm 1 The Marine predator algorithm (MPA)

Initialize $prey_i (i = 1, 2, 3, \ldots, N)$, P=0.5.
while $(t < t_{max})$
  compute the fitness for each $prey_i$.
  $best_{Predator}$=the fittest solution.
  $Best_{fit}$= the fitness value of $best_{Predator}$.
  Implement the memory saving.
  Build E matrix at the first iteration and update it later if there is better
  compute CF according to Eq. 13.
  for each i prey **do**
    if $(t < 1/3 * t_{max})$
      move the current $\mathbf{prey}_i$ to another position based on Eq. 8.
    elseIF $(1/3 * t_{max} < t < 2/3 * t_{max})$ then
      if $(i < 1/2 * N)$
        move the current $\mathbf{prey}_i$ to another position based on Eq. 10.
      else
        move the current $\mathbf{prey}_i$ to another position based on Eq. 12.
      end if
      move the current $\mathbf{prey}_i$ to another position based on Eq. 15.
    end if
  end for
  Compute the fitness for each $\mathbf{prey}_i$.
  Update E, if there is a newly updated solution better than the current $best_{Predator}$.
  implement the memory saving
  Execute the FADs according to Eq. 16
  t++
end while
Return $Best_{fit}$

5 Proposed work

In this section, the marine predators algorithm improved using an improvement strategy that selects a number, increasing gradually with the iteration, of the worst solutions and updates them toward the best-so-far solution and randomly within the search space is illustrated in this section. Our strategy is differentiated of the evolutionary population dynamics (EPD) (Saremi et al. 2015) by that, in EPD, the worst N/2 solutions are omitted from the population and repositioned randomly them again around the best solution. On the contrary, in our strategy, a number of the worst solutions, increasing linearly with the iteration, will be selected each iteration to trade-off between moving: most the dimensions within each one toward the best-so-far solutions, and the others are randomly reinitialized within the search space of the problem.

Our strategy works gradually on increasing the convergence of each worst one toward the best so-far solutions until accelerating the convergence speed and finding better solutions. Generally speaking, when moving each worst solution toward the best-so-far, the convergence toward the best solution increases significantly, but what if the best-so-far solution is local optima, if that, then the diversity of population are already reduced. Since the diversity of the population fade away and at the same time the best-so-far solution is local minima, no better solutions could be obtained after that. Therefore, our strategy manipulates in some of the dimensions of the worst repositioned solutions, to help in disposing of the local optima, and at the same time to keep the diversity of the population.

Our strategy methodology is contradicted with EPD because EPD repositions the worst half of solutions around the best-so-far solutions using a random number to specify the distance that will add to the best-so-far solution, thus the obtained distance is based on the random number generated and the probability of reaching better solutions is relied on this generated number. On the contrary, our proposed moves the worst solutions toward the best-so-far solutions, and the distance between the best and the worst reduces gradually each update until exploiting the best-so-far solutions deeply in addition to disposing of the local minima by reinitializing randomly within the search space some of the dimensions within the worst solutions to get rid of local minima. Within this section, Initialization, evaluation, and improvement will be illustrated in detail and mathematically.

5.1 Initialization

Before starting the optimization process, a group of N prey (population size) will be defined, where each one in the group will have a number of dimensions according to the threshold level acquired. Those dimensions will be distributed within the search space according to Eq. 17. In this phase, a number N of prey with a number of the threshold is predefined.

$$\mathbf{prey}_i = \mathbf{min} + \mathbf{r} \cdot (\mathbf{max} - \mathbf{min}) \tag{17}$$

where $\mathbf{r}$ is a vector including values generated randomly between 0 and 1, **min**,and **max** are the vectors including the upper and lower bound of the grey level for an image within its histogram, respectively? For example, assuming that the upper bound grey level is equal to 255, the lower bound is 0, and the threshold level of 8 is required, then each solution within the population will be represented as shown in Fig. 1 (truncated the digits after the decimal point to become integer numbers), in addition to adding the first and the last cell with the lower and the upper grey level for the image. Each cell within Fig. 2 indicates the threshold value used to find the optimal threshold values for threshold level 8.

5.2 Evaluation

After each generation, the fitness value for each solution will be calculated using Eq. 1 and the solutions under the new positions with higher fitness value will stay inside the population for the next generation; otherwise, the old positions will be used again within the next generation. So, the evaluation steps significantly affect the performance of MPA, because the fittest solution (the best-so-far solution/predator) and the solutions used within the next generation are specified according to this step.

5.3 Linearly increased the worst solutions improvement strategy(LIS)

Recently, a new strategy (Abdel-Basset et al. 2020b) known as a linear reduction diversity technique has been proposed to update a number, increasing with each iteration, of the worst solution toward the best-so far solution to increase the convergence speed toward the

$\overrightarrow{prey}_i$	0	10	40	80	120	180	200	230	240	255

Fig. 2 Initial representation of the solution for tackling ISP

best-f-so-far solution, in addition to improving the quality of the obtained solution. But, updating toward the best-so-far may accelerate the convergence toward the local minima, therefore in this paper, we tried to get rid of this problem based on another position update scheme. After calculating the fitness for each solution, a number of the worst solutions will be updated using two ways: the first one is based on updating those solutions toward the best solution with a controlling factor to determine the step size taken in the direction of this best solution to avoid stuck into local minima if the best is so. This factor, namely C, will take small steps increased gradually with increasing the current iteration even avoiding stuck into local minima at the start of the optimization process because the best-so-far might be local minima. Additionally, the second way is based on updating the current solution in the direction of the best-so-far solution using a random number generated within 0 and 1 based on the uniform distribution but will be added to this solution another step size generated based on the upper and lower bound of the problem. The number of the worst solutions is related to the iteration, where, at the start of the iteration, a small number of solutions will be selected, and with the iteration, this number will be gradually increased until maximizing at the end of the iteration. Generally, Eq. 18 represents the mathematical formula for calculating this number symbolized *np*:

$$np = N * \frac{t}{t_{max}} \tag{18}$$

Where N is the population size, and np indicates the number of the worst solutions increased linearly with increasing the current iteration t. After calculating np, those worst solutions will be updated according to Eq. 19.

$$\mathbf{prey}_{w,j} = \begin{cases} min + r.(max - min) & if\ r_3 < ER \\ E_{w,j} + r * (E_{w,j} - prey_{w,j}) & otherwise \end{cases} \tag{19}$$

$$C = 1 - e^{-t*r} \tag{20}$$

$prey_{w,j}$ indicates the jth dimension of the wth worst particle. r_3 is a random number between 0 and 1. min refers to the lower bound of the jth dimension, while max is the upper bound of the same dimension. r is a number generated randomly between 0 and 1. $E_{w,j}$ indicate the jth dimension of the row w in the elite matrix (E). *ER* expresses the exploration rate used to get the solution out of the local minima problem, its value will be discussed in the parameter settings. This strategy is known as the linearly increased the worst solutions improvement strategy (LIS). Finally, the pseudo-code of the hybrid MPA-LIS is shown in Algorithm 2.

Algorithm 2 Hybrid MPA-LIS (HMPALS)

Initialize prey, P=0.5.
Compute the fitness for each solution using Eq. 1.
Implement the memory saving.
Build E matrix.
while $(t < t_{max})$
  Compute CF according to Eq. 13.
  for each i prey **do**
    if $(t < 1/3 * t_{max})$
      move the current $\mathbf{prey}_i$ to another position based on Eq. 8.
    elseIF $(1/3 * t_{max} < t < 2/3 * t_{max})$ then:
      if $(i < 1/2 * N)$
        move the current $\mathbf{prey}_i$ to another position based on Eq. 10.
      else
        move the current $\mathbf{prey}_i$ to another position based on Eq. 12.
      end if
      move the current $\mathbf{prey}_i$ to another position based on Eq. 15.
    end if
  end for
  Compute the fitness for each $\mathbf{prey}_i$ using Eq. 1.
  Update E, if there is better.
  Implement the memory saving.
  // Applying FADs
  Execute the FADs according to Eq. 16.
  Compute the fitness for each $\mathbf{prey}_i$ using Eq. 1.
  Update E, if there is better.
  Implement the memory saving.
  //Applying LIS strategy to update the worst solutions
  Compute np.
  Update the worst np solutions using Eq. 19.
  Compute the fitness for those updated worst solutions using Eq. 1.
  Update E, if there is better
  Implement the memory saving.
  t++
end while

5.4 Ranking-based updating strategy

Recently, in Abdel-Basset et al. (2020a), we proposed a method called ranking-based updating strategy which is based on replacing those solutions that could not fulfill better solutions in the last perIter iterations with other ones to improve the outcomes produced by the optimization algorithms; in our experiments, perIter is set to 3 as recommended in the original work. But, our previous proposition was based on updating those solutions which passed the unallowed rank by steering them to the right direction of the best-so-far solution in the hope of finding better solutions there. However, moving in the direction of the best-so-far solution might take the algorithm to local minima and hence no better solution will be achieved within the remaining iterations. Therefore, herein, a new updating scheme has been proposed for updating those solutions based twofold: the first one is based on searching for a better solution around one selected randomly from the population as an attempt to increase the exploration capability for reaching other regions might involve better solutions; the second one is based on turning around the best-so-far solution in a shrinking circle with the iteration to strengthen the exploitation capability. The exchange between those twofold is achieved based on a probability Pr ranging between 0 and 1 and picked by the researchers according to their experiments. the optimal value to this parameter within our experiments as discussed later. Generally, the mathematical model of this updating scheme is as follows:

$$prey_{w,j} = \begin{cases} prey_{a,j} + D * (prey_{b,j} - prey_{c,j}) + r * (prey_{a,j} - prey_{d,j}) & if\ r_3 < Pr \\ E_{i,j} + r_1 * A * (prey_{b,j} - prey_{c,j}) & otherwise \end{cases} \tag{21}$$

$$a = 1 - \frac{t}{t_{max}} \tag{22}$$

$$A = 2 * a * r_2 - a \tag{23}$$

$$D = e^{-A*t} * \cos 2\pi r \tag{24}$$

where a, b, c, and d are indices of four solutions selected randomly from the population. r, r_1, and r_3 are three random numbers ranging between 0 and 1. Algorithm 3 described the steps of the ranking-based updating strategy (RUS) for replacing the solutions which pass three consecutive iterations without any better solution than the best-local one with those created using Eq. 21. Finally, this strategy is integrated with the MPALS variant as listed in algorithm 4 to propose a new strong one, namely HMPA, having a high ability to balance between the exploration and exploitation capability for reaching better solutions in less number of function evaluations. The RUS give the choice to the algorithms for determining if the exploration operator will be applied more than the exploitation operator and this will make applicable for several metaheuristic algorithms because some algorithms might have low exploitation operator while the others might suffer from weakening the exploration capability, so this method could help the two sets based on the required need.

Algorithm 3 Ranking-based updating strategy.

CR: a vector of size N initialized with 0s value.
i = 0
perIter = 3
while $(i < N)$
  if $(fit(\mathbf{prey}_i) > fit(\mathbf{prey}_{old_i}))$
    CR_i^{++}.
  else
    $CR_i = 0$
  end if
  i++
end while
for each i particle **do**
  if $(CR_i > perIter)$
    Update **prey** toward the best one using Eq. 21.
  end if
end for

Algorithm 4 hybrid MPALS and RUS (HMPA)

```
Initialize prey, P=0.5.
Compute the fitness for each solution using Eq. 1.
Implement the memory saving.
Build E matrix.
while (t < t_max)
   Compute CF according to Eq. 13.
   for each i prey do
      if (t < 1/3 * t_max)
         move the current prey_i to another position based on Eq. 8.
      elseIF (1/3 * t_max < t < 2/3 * t_max)
         if (i < 1/2 * N)
            move the current prey_i to another position based on Eq. 10.
         else
            move the current prey_i to another position based on Eq. 12.
         end if
         move the current prey_i to another position based on Eq. 15.
      end if
   end for
   Compute the fitness for each prey_i using Eq. 1.
   Update E, if there is better
   Implement the memory saving.
   // Applying FADs
   Execute the FADs according to Eq. 16.
   Compute the fitness for each prey_i using Eq. 1.
   Update E, if there is better.
   Implement the memory saving.
   //Applying LIS and RUS strategy to update the worst solutions
   Applying algorithm 3.
   compute np
   Update the worst np solutions using Eq. 19.
   Compute the fitness for those updated worst solutions using Eq. 1.
   Update E, if there is better
   Implement the memory saving.
   t++
end while
```

Some of the advantages of the proposed algorithms are that both the proposed strategy integrated work on increasing the convergence speed and at the same time avoiding falling into local optima and subsequently the probability of finding better solutions is significant. Although the significant success achieved by both HMPA and MPALS on the various threshold-levels, it consumes more a bit computational cost compared to some of the competing algorithms as their main limitation tackled in future work.

6 Results and discussion

6.1 Test images description and experimental settings

To validate our proposed algorithms, 14 test images taken from Berkeley Segmentation Dataset (BSDS500) [75] are used to observe their effectiveness. These images are named as “12003”, “61060”, “38092”, “232038”, “108082”, “148089”, “189003”,“108070”, “227092”, “277036”, “Barbara”, “Airplane”, “Mand”, “lena”. Besides, the original images and the histogram of each one is shown in Figs. 3 and 4.

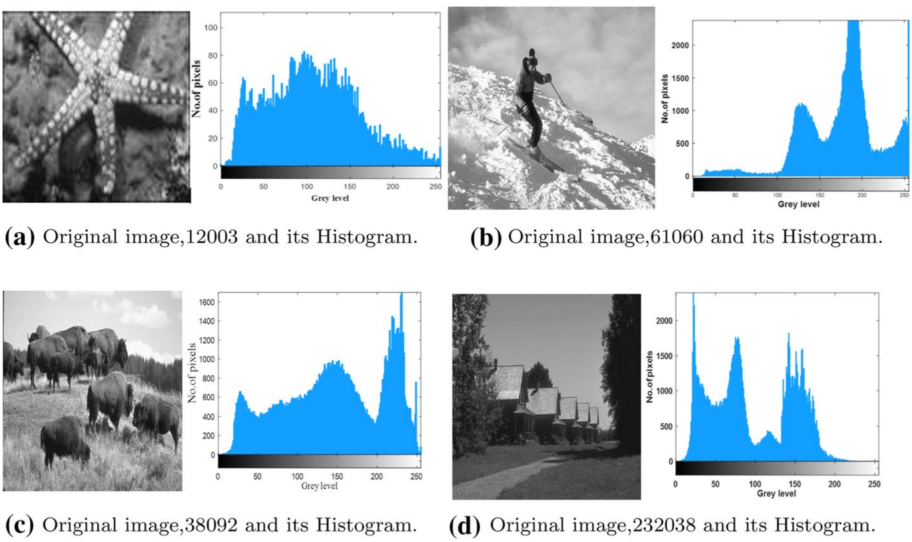
A number of algorithms, such as HHA (Bao et al. 2019), WOA (Mirjalili and Lewis 2016), slime mould optimizer (SMA) (Li et al. 2020), improved tunicate swarm algorithm (ITSA) (Houssein et al. 2021), flower pollination algorithm (FPA) (Yang 2012), EO (Abdel-Basset et al. 2020c), and IMPA (Abdel-Basset et al. 2020a), are employed to check the effectiveness of our proposed algorithms, whose parameters are as recommended in the cited papers, except the population size N set to 30, and the maximum iterations t_{max} equal 300 for a fair comparison. Those algorithms were employed because they have been recently applied for tackling several optimization problems: image segmentation, parameter estimation of photovoltaic models, DNA fragment assembly problem, global optimization, and several else, and could come true superior outcomes. Table 1 lists the parameter settings of those compared algorithms.

Regarding the parameters of the proposed algorithm, ER of the MPALS algorithm is tuned on 12003 under the threshold level 40 and displayed in Fig. 5, which shows the superiority of $ER = 0.9$ against the other values. For the parameter Pr of HMPA, it is checked on some values such as 0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0 and the outcomes are illustrated on Fig. 6. This figure shows that $Pr = 0.05$ is the best. Some values for the parameter β are observed to see the best value for it, after observing and exposing the output on Fig. 7, $\beta = 0.01$ is the best.

Finally, a device equipped with Windows 10 and having 32GB of RAM, AND Corei7 Intel CPU with 2.40GH as speed is used to conduct those experiments.

6.2 Performance metrics

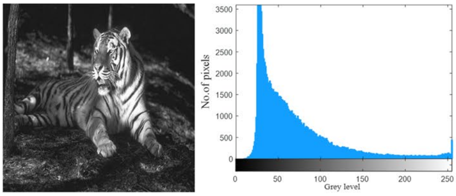
In this section, the performance metrics used to observe the performance of the algorithm will be discussed in brief. Those metrics are standard deviation (SD), Peak Signal to noise



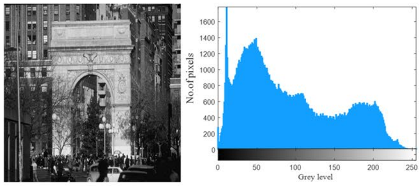
(a) Original image,12003 and its Histogram. **(b)** Original image,61060 and its Histogram.

(c) Original image,38092 and its Histogram. **(d)** Original image,232038 and its Histogram.

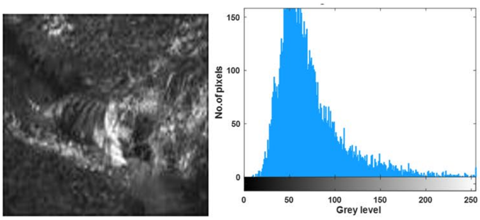
Fig. 3 Illustration the original image and its histogram used in our experiment



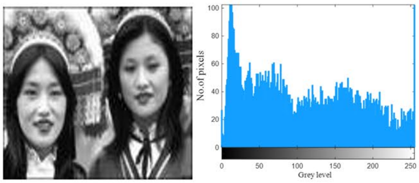
(a) Original image,108082 and its Histogram.



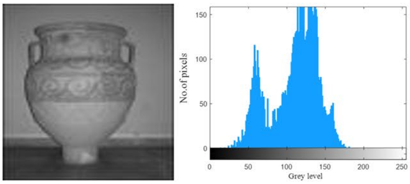
(b) Original image,148089 and its Histogram.



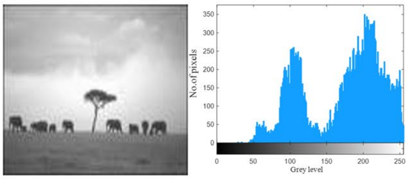
(c) Original image,108070 and its Histogram.



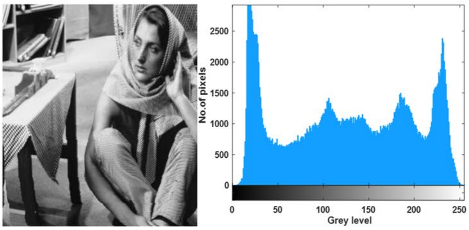
(d) Original image,189003 and its Histogram.



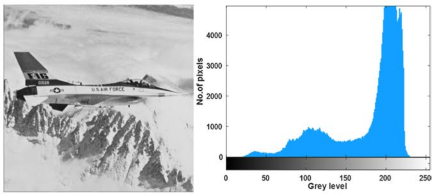
(e) Original image,227092 and its Histogram.



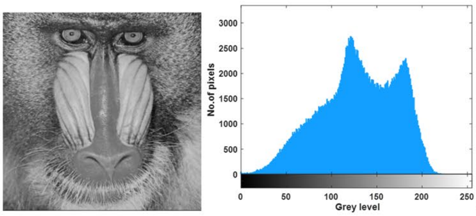
(f) Original image,277036 and its Histogram.



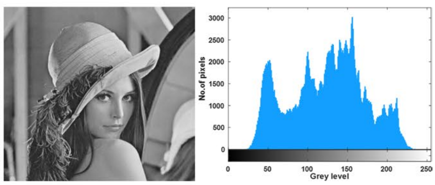
(g) Original image,Barbara and its Histogram.



(h) Original image,Airplane and its Histogram.



(i) Original image,Mand and its Histogram.



(j) Original image,Lena and its Histogram.

Fig. 4 Illustration the original image and its histogram used in our experiment

Table 1 Parameter settings of competing algorithm

Algorithm	Parameter	Value
FPA (Yang 2012)	Probability switch p	0.8
WOA (Mirjalili and Lewis 2016)	Constant (a)	Is linearly decreased from 2 to 0
HHA (Bao et al. 2019)	Energy of a rabbit (E)	$E \in [0, 2]$
SMA (Li et al. 2020)	Constant (z)	0.03
	Constant a_1	1
EO (Abdel-Basset et al. 2020c)	Constant a_2	2
	Constant (P)	0.5
IMPA (Abdel-Basset et al. 2020a)	Constant(FADs)	0.2
	Constant (perIter)	3
ITSA (Houssein et al. 2021)	P_{min}	1
	P_{max}	4
	pr	0.3

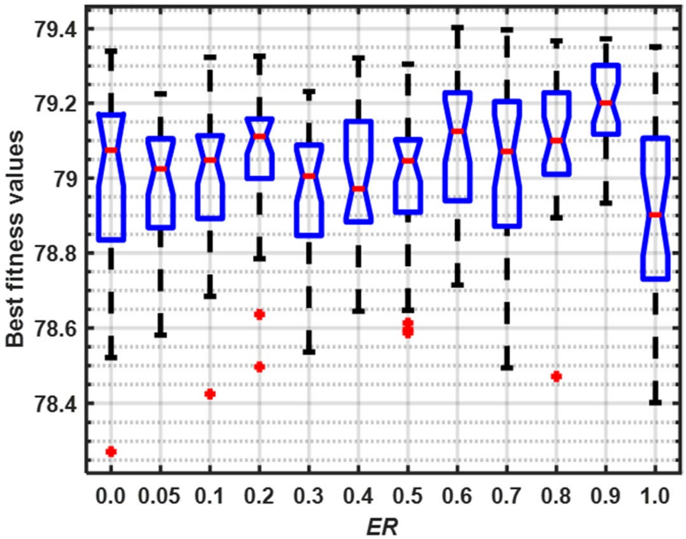


Fig. 5 Tuning the parameter ER of MPALS under 40 threshold level for 12003

ratio (PSNR), Structured similarity index metric (SSIM), universal quality index (UQI), fitness value under Kapur's entropy, and CPU time.

1. Standard deviation (SD): SD is used to measure the stability of the outputs obtained by each algorithm within several runs and mathematically calculated using the following equation:

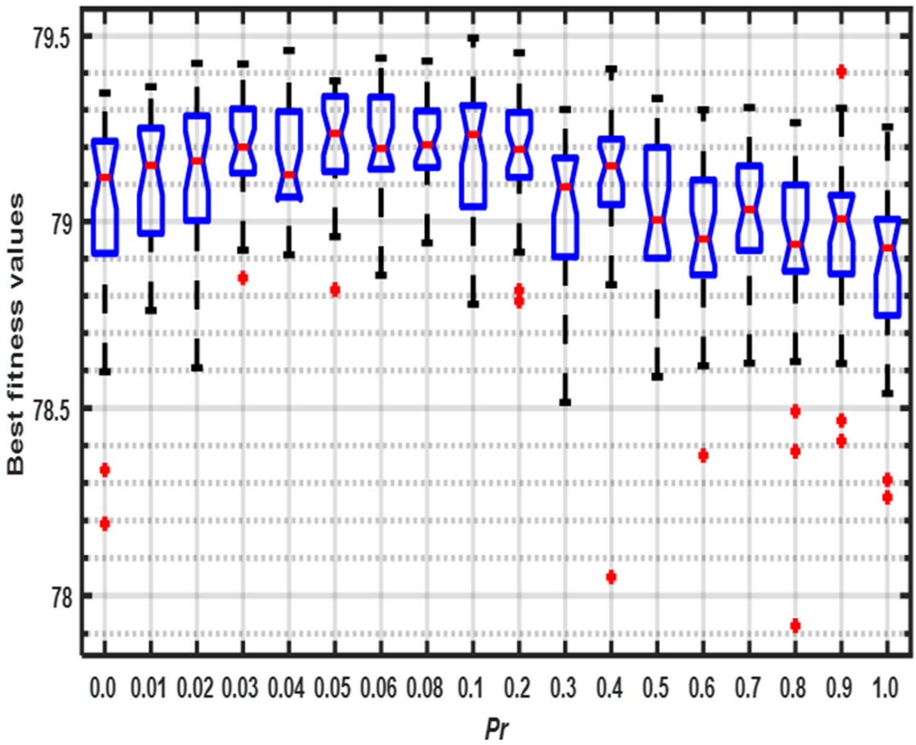


Fig. 6 Tuning the parameter Pr of HMPA under the threshold level 40 on 12003

$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(f_i - \bar{f})^2} \tag{25}$$

n indicates the number of independent runs, f_i is the fitness value under Kapur's entropy of the i^{th} run, and $\bar{f}$ is the average of the fitness value within the independent runs. The algorithm with the smallest SD is considered the best.

2. Peak signal to noise ratio (PSNR): PSNR (Hore et al. 2010) is a quality indicator used to check the quality of the segmented image compared with the original image. This indicator calculates the mean square error (MSE) between the original and segmented images using Eq. 27, then take a log with base 10 for the square of 255 divided by MSE. The mathematical model of PSNR is as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{26}$$

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \mid A(i,j) - S(i,j) \mid}{M * N} \tag{27}$$

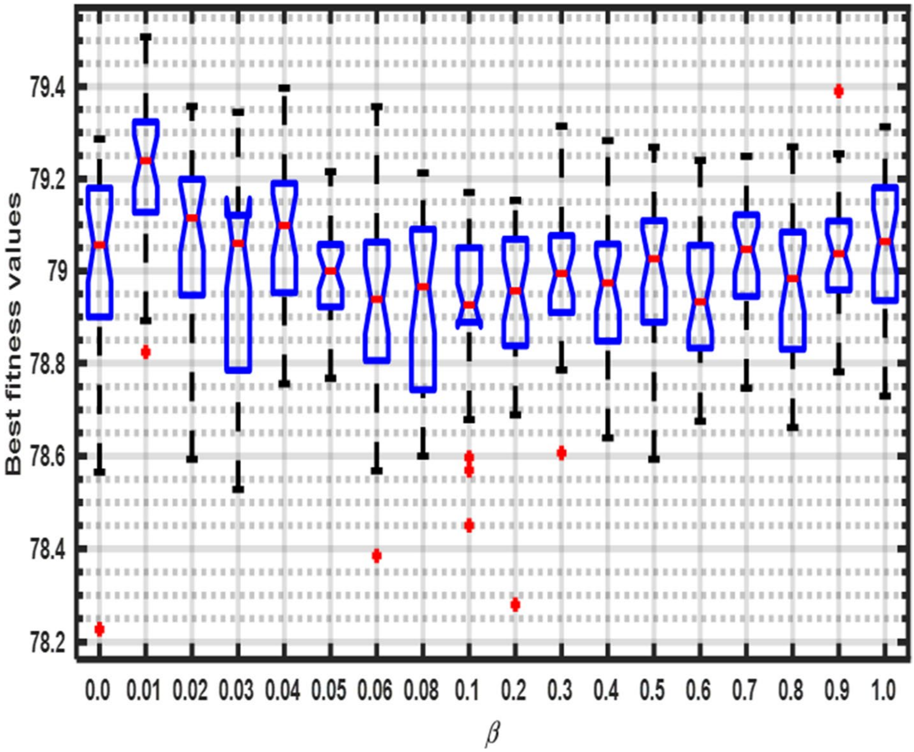


Fig. 7 Adjusting the parameter β of HMPA under the threshold level 40 on 12003

$A(i, j)$, $S(i, j)$ are the grey value of both segmented and source images within i^{th} row and j^{th} the column in matrices A, and B that have the size of rows and columns equal to M, and N, respectively.

3. Structured similarity index metric (SSIM): SSIM (Hore et al. 2010) is different about PSNR because it takes in its consideration the structure of the segmented image compared with the original image. Generally, it measures the brightness, structure similarity, and contrast distortion between the source and predicted images. Mathematically, SSIM is formulated as following:

$$SSIM(O, S) = \frac{(2\mu_o\mu_s + a)(2\sigma_{os} + b)}{(\mu_o^2 + \mu_s^2 + a)(\sigma_o^2 + \sigma_s^2 + b)} \tag{28}$$

where μ_0, μ_s express the mean intensities of both original and predicted image; σ_0 and σ_s indicate SD of the same two images respectively; a,and b is fixed values and assigned to 0.001 and 0.003 respectively. This metric is maximized until reaching better accuracy.

6.3 Stability and CPU time

Starting with stability, to measure the convergence between the outcomes produced by each algorithm, the average of SD was calculated on all datasets within 30 independent runs for each one and displayed in Fig. 8. Figure 8 shows that HMPA could produce more converged outcomes in comparison with the other algorithm. As a result, the performance of our proposed algorithm, HMPA, is more stable within 30 independent runs. Ending with CPU time, the improvements on MPA may significantly affect running time, so we calculate the running time result of each algorithm and then show those outcomes within Fig. 9. Inspecting Fig. 9 told us that the improvements done on MPA don't affect significantly the speedup of MPA and hence it is a strong alternative to the existing image segmentation techniques because it could fulfill better outcomes within almost converged CPU time.

6.4 Graphically performance analysis

The effectiveness of the proposed algorithms compared with the other algorithms under different performance metrics will be observed within this section. Generally, this section is organized under graphical depictions as follows:

1. Comparison of fitness values.
2. Comparison under PSNR values.
3. Comparison under SSIM values.
4. Comparison under various threshold levels: small and large levels.
5. Depiction of the fitness values using the interval plot.

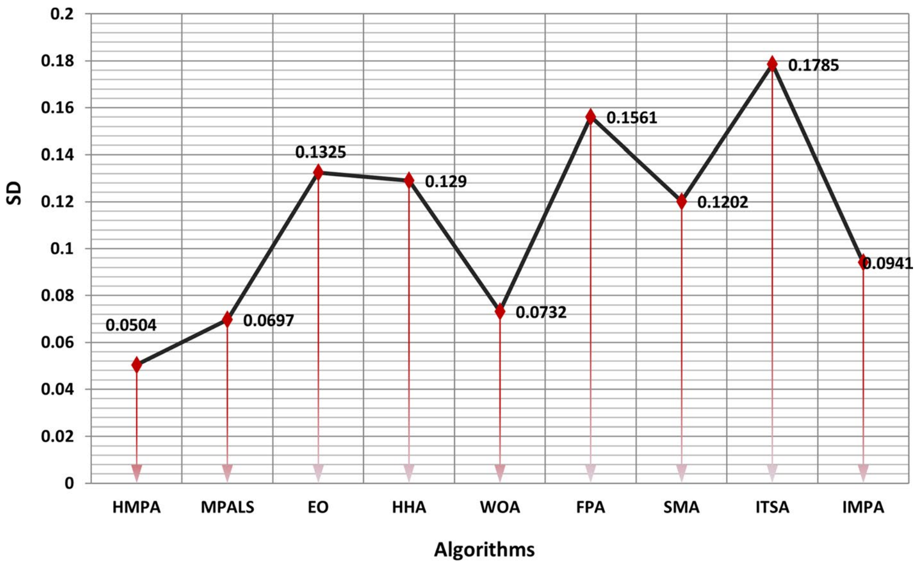


Fig. 8 Comparison the average of SD values obtained by each algorithm

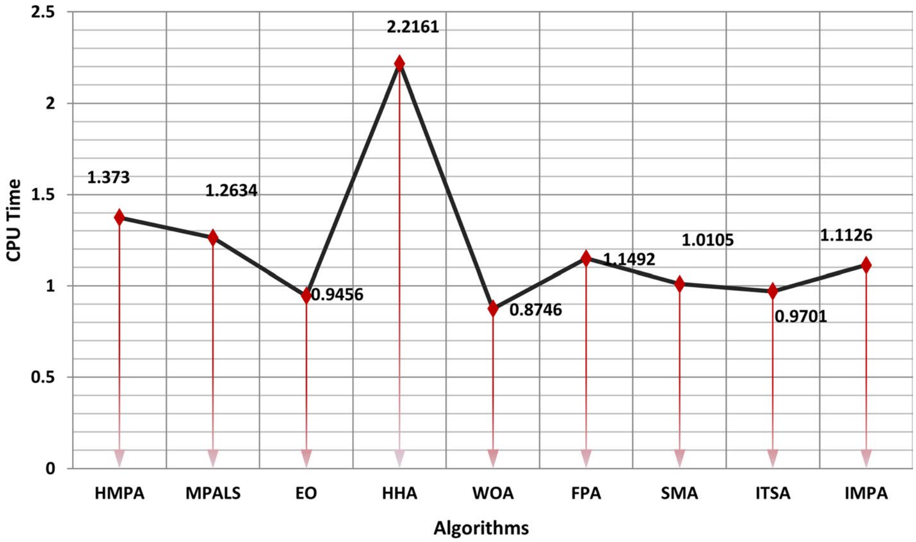


Fig. 9 Comparison of the CPU Time values obtained by each algorithm

(1) Comparison under Fitness value.

Figure 10 shows the average of the fitness values obtained by each algorithm within 30 independent runs under all threshold levels on all the test images. Observing Fig. 10 shows that HMPA could reach the first rank with a value of 35.17, MPALS reach the second rank with 34.81, while FPA came as the last one with a value of 33.91. This superiority is

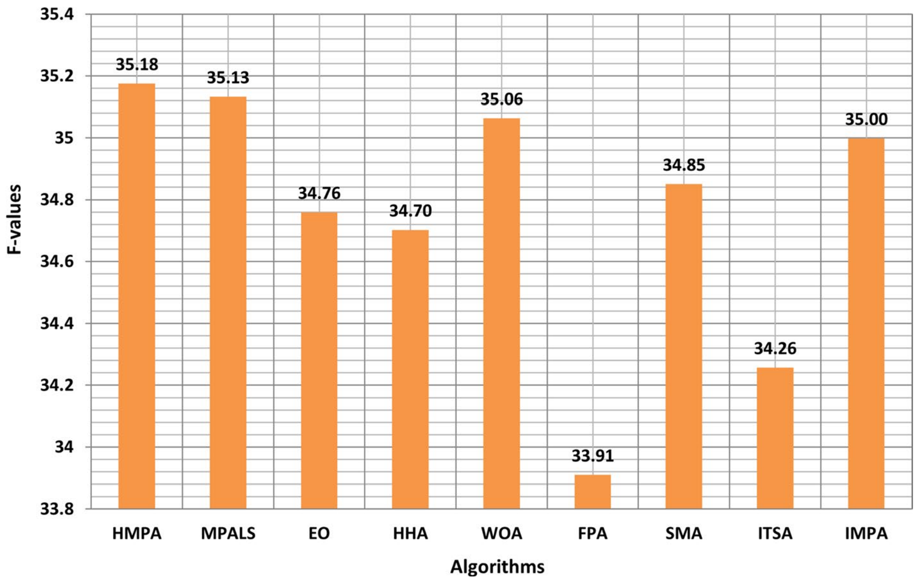


Fig. 10 Depiction of the average fitness values under all threshold levels for all test images

caused by both LIS and RUS that could accelerate the convergence toward the best solution and at the same time take the solutions out of the local optima problem.

(2) Comparison under PSNR values.

In this section, the quality of the predicted image using the proposed algorithms and the others will be compared under the PSNR metric. After calculating the average PSNR values within 30 runs for each level on each image, and calculating the averages under all levels on all images, they are shown in Fig. 11. After observing Fig. 11, it is concluded that the proposed algorithm, HMPA, could achieve the best value compared with the others and occupies the first rank with a value of 23.45, and both WOA and MPALS could come in the second rank with a value of 25.03, while FPA came in the last rank with 24.06. Ultimately, it is concluded that HMPA could reach the threshold values which accurately separate the objects in an image compared to the other compared algorithm, and subsequently, HMPA considers a strong alternative to the existing techniques for tackling the image segmentation problem.

(3) Comparison under SSIM values.

In this part, the mean of the SSIM values obtained by each algorithm within 30 independent runs on all threshold levels under all the images will be discussed to see any algorithm could reach better accuracy. According to Fig. 12, HMAP could be the best with a value of 0.9320 and EO, WOA, and SMA as the second-best ones with 0.9318, while FPA is the worst with 0.9300. Unfortunately, MPALS could only overcome ITSA and FPA over this metric as the main limitation to this variant solved using the RUS, which could explore more intractable regions for fulfilling better outcomes for all employed performance metrics: F-value, PSNR, SSIM, SD, and CPU time.

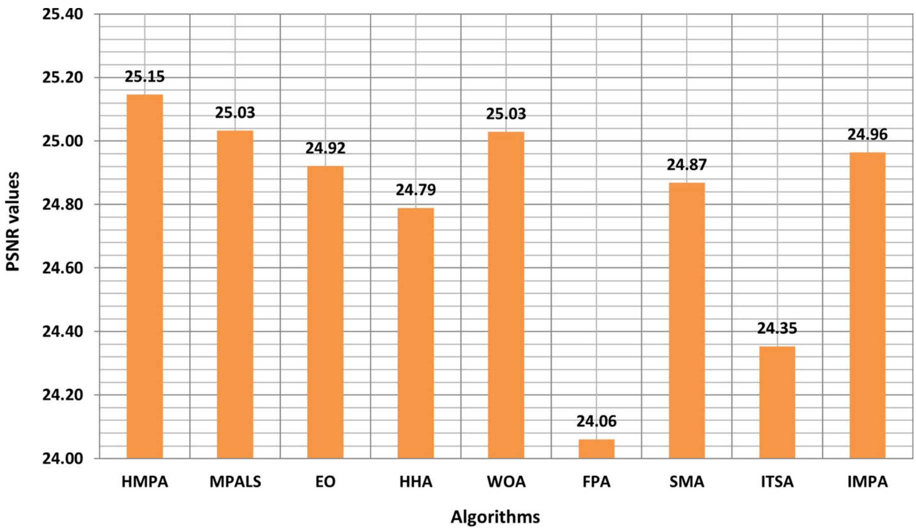


Fig. 11 Depiction of the average PSNR values under all threshold levels for all test images

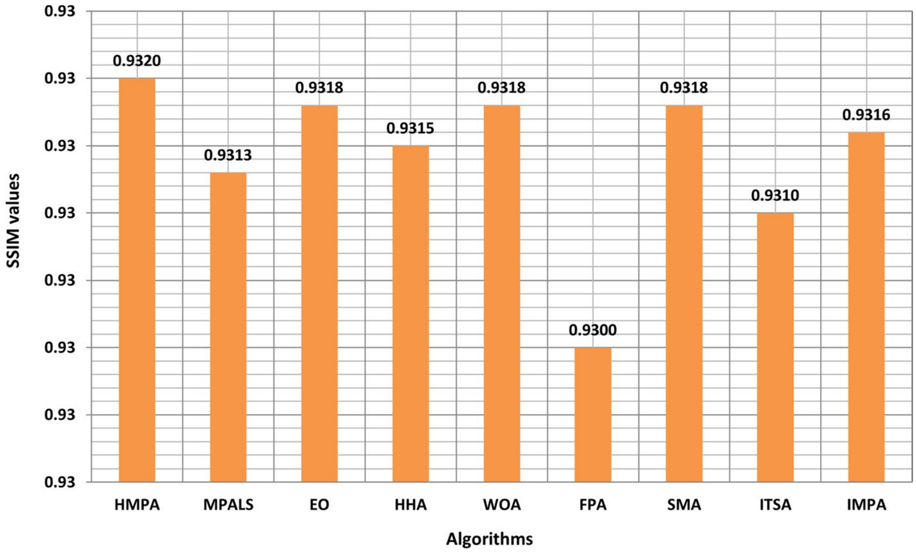


Fig. 12 Depiction of the average SSIM values under all threshold levels for all test images

(4) Comparison under various threshold levels: small and large levels.

In this section, the various algorithms have been compared based on the threshold level to see their performance with small and high threshold levels. After computing the average of SSIM, PSNR, and F-value on the threshold levels ranging between 2 and 7 as the small levels and presenting the outcomes in Fig. 13, it is notified that all algorithms except FPA are almost converged under those three performance metrics and subsequently any of them could be used to segment any image with small threshold levels ranging between 2 and 7. However, with the high threshold levels ranging between 10 and 40, HMPA could be superior for the three employed performance metrics: SSIM, PSNR, and F-value as depicted in Fig. 14. In general, HMPA considers a strong alternative for tackling the image segmentation problem for images with small or high threshold levels contradicted most of the compared algorithms which have good performance for only the small levels.

(5) Comparison under interval plot.

Within this section, the Boxplot is used to check the performance of the algorithms under Fitness values on each image with a threshold level (T) equal to 40. Specifically speaking, each algorithm is run 30 times on some test images under T=40 and the fitness values within those 30 runs are drawn in Figs. 15, 16 for this algorithm. Figs. 15, 16 prove the superiority of HMPA under T=40 on all the test images, where HMPA could come in the first rank superior to all the other algorithms, while MPALS occupies the second rank on all the test images and the third rank is occupied by WOA. These experiments show the superiority of HMPA on the images with high threshold levels, so it is considered as an outstanding alternative to solve the multilevel thresholding image segmentation, especially with high threshold levels.

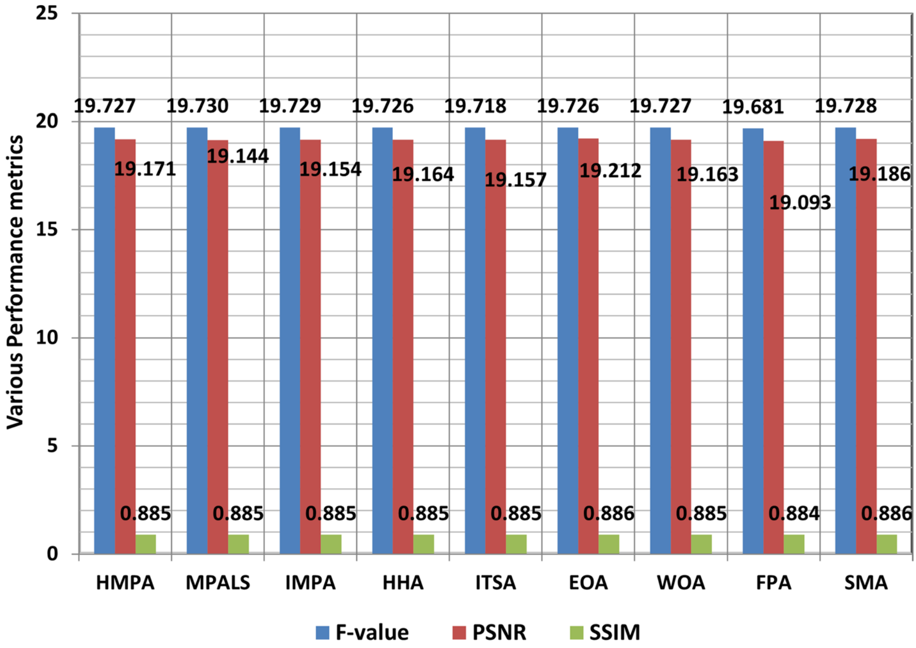


Fig. 13 Comparison based on various performance metrics: F-value. PSNR, and SSIM on small threshold levels ranging between 2 and 7

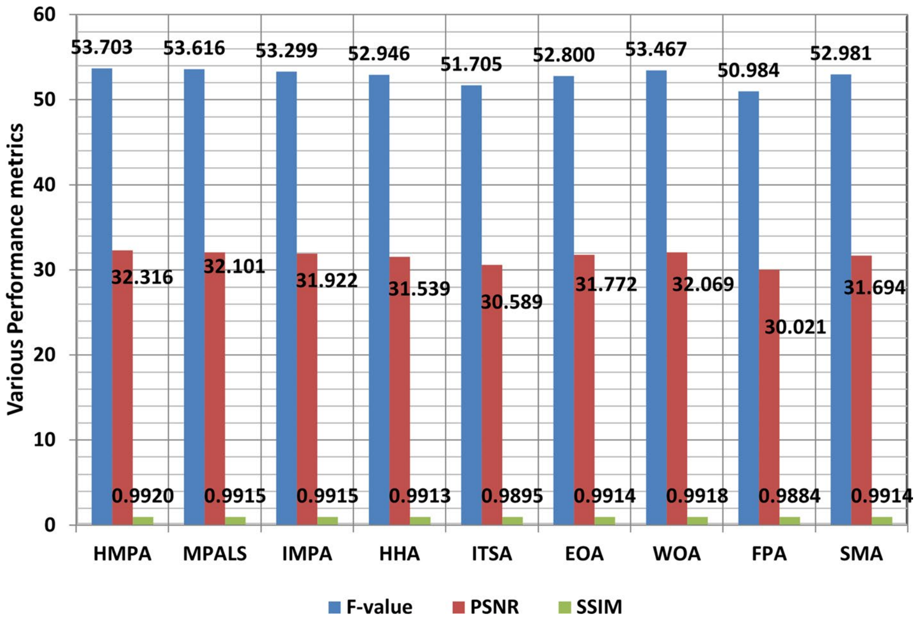
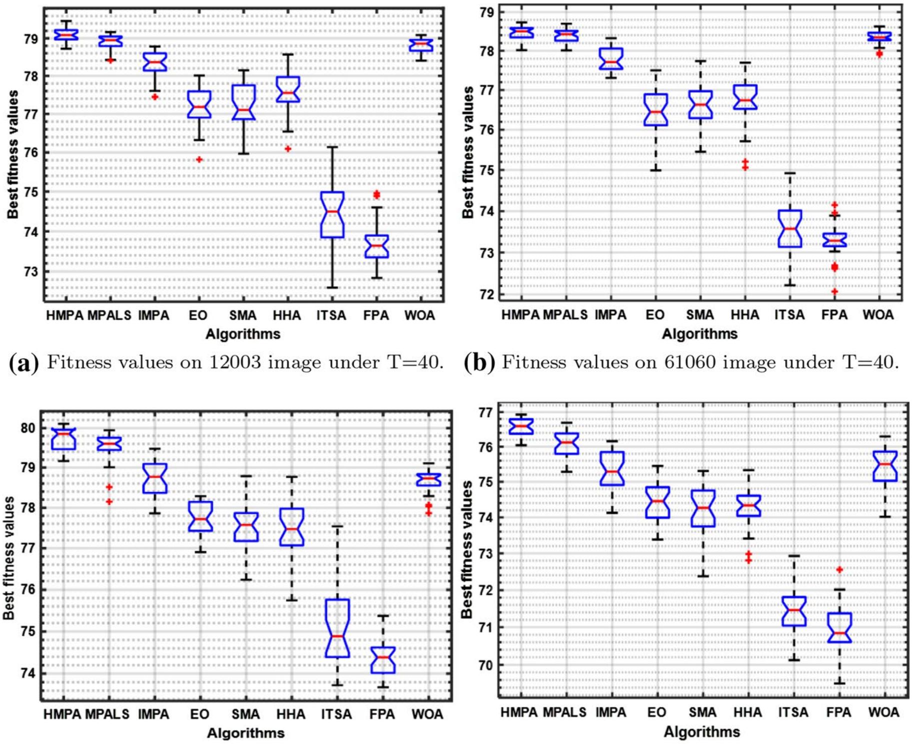


Fig. 14 Comparison based on various performance metrics: F-value. PSNR, and SSIM on large threshold levels ranging between 10 and 40



(a) Fitness values on 12003 image under T=40. **(b)** Fitness values on 61060 image under T=40.

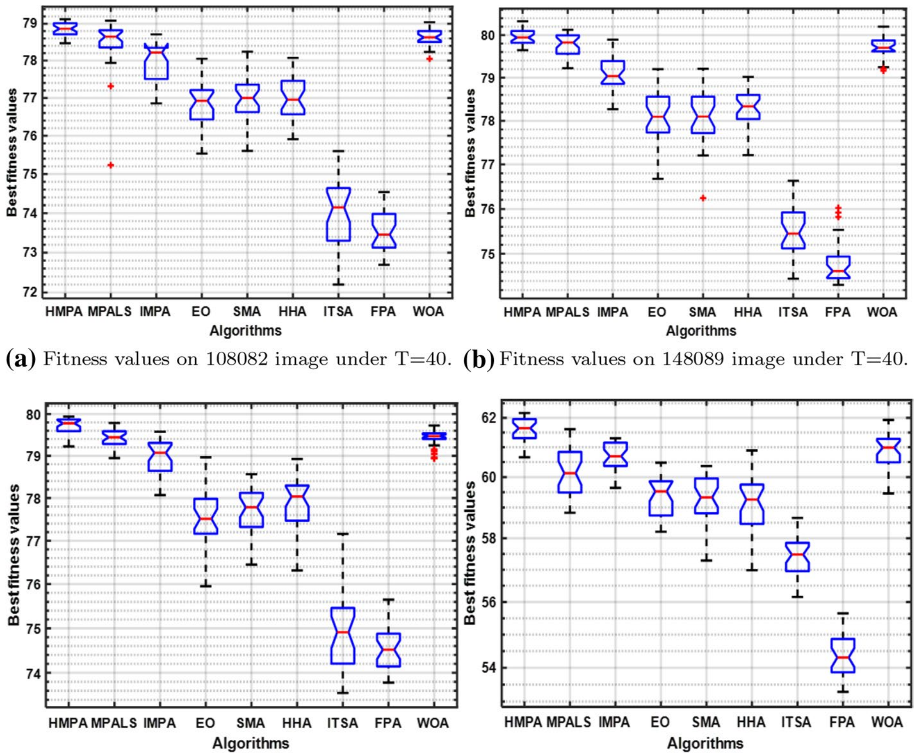
(c) Fitness values on 38092 image under T=40. **(d)** Fitness values on 232038 image under T=40.

Fig. 15 Fitness values on all images under T = 40

6.5 Descriptive performance analysis

In this section, the various performance metrics will be measured according to each algorithm within 30 independent runs ad introduced within the following tables. Broadly speaking, Tables 2 and 3 introduces the F-values obtained by each algorithm on each test image. Inspecting Tables 2 and 3 shows that HMPA could be competitive and superior in 110 cases out of 154, while MPALS could achieve the best and equal with some other algorithms in 84 out of 154 in terms of the F-values. Both MPALS and HMPA as our proposed variants could be competitive to the others, which reach the same outcomes, in 52 cases, while could be superior in 83 others, and hence those proposed variants are strong alternatives to solve the image segmentation problems, especially HMPA which could outperform the other variants for most test cases.

In Tables 4 and 5, the PSNR values obtained based on the segmented images achieved based on the fitness values assigned in Tables 2 and 3 are exposed on each threshold level for each image. Inspecting this table show that HMPA could be the best and equal with the other algorithms in 70 test cases, while MPALS could be superior and competitive in 37 cases. Based on that, HMPA considers the best-proposed variant, and the best compared to the other compared algorithms since the nearest compared algorithm in terms of the PSNR is EO which could be superior and competitive in only 42 cases.



(a) Fitness values on 108082 image under T=40. **(b)** Fitness values on 148089 image under T=40.

(c) Fitness values on 189003 image under T=40. **(d)** Fitness values on 108070 image under T=40.

Fig. 16 Fitness values on all images under T = 40

In Tables 6 and 7, the SSIM values are exposed on each threshold level for each image. From this table, it is concluded that HMPA could be the best and equal with the other algorithms in 90 test cases, while MPALS could be superior and competitive in 57 cases. Meanwhile, the two-proposed variants could reach better and competitive outcomes compared to the other algorithms in 102 cases. Based on that, HMPA considers the best.

6.6 Comparison under Wilcoxon rank-sum test

In this section, the Wilcoxon rank-sum test (Lam and Longnecker 1983) is used to compare the outcomes of HMPA with those obtained by the competing algorithms on some test images used in our experiments and presenting the outcomes in Table 8. This test returns the p-value of the two-sided Wilcoxon rank-sum test, and if this p-value is less than a significant level recommended 5% then there is a difference between paired data; otherwise, the null hypothesis will be accepted, which no difference between paired data. In our experiments which compare the data obtained by HMPA with each competing algorithm and their outcomes are presented in Table 8, it is obvious that the p-value for most test cases on threshold levels higher than 7 is less than 5%, and hence there is a difference between the outcomes of the HMPA and the other competitors. However, for threshold levels lower than 7, the outcomes of HMPA were significantly different from those of some

Table 2 Performance evaluations using F-value on all images

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
12003	HMPA	**13.0262**	**16.2486**	**19.2960**	22.1130	**24.7665**	27.2948	**34.2136**	**44.4402**	**53.1379**	**67.4516**	**79.1005**
	MPALS	**13.0262**	**16.2486**	**19.2960**	**22.1133**	**24.7665**	**27.2955**	**34.2106**	44.3681	53.1219	67.4465	78.9709
	IMPA	**13.0262**	**16.2486**	**19.2960**	**22.1133**	24.7662	27.2937	34.1948	44.2633	52.8711	66.8537	78.1449
	EO	**13.0262**	**16.2486**	**19.2960**	22.1132	24.7661	27.2781	34.1607	44.2388	52.5248	66.3570	77.2095
	SMA	**13.0262**	16.2484	19.2957	22.1126	24.7644	27.2864	34.1934	44.3401	52.7725	66.5107	76.9663
	HHA	**13.0262**	16.2485	19.2950	22.1132	24.7656	27.2925	34.1994	44.3064	52.7589	66.4856	77.5565
	ITSA	**13.0262**	16.2483	19.2924	22.1095	24.7556	27.2686	34.0895	43.8022	51.7693	64.4218	74.4240
	FPA	**13.0262**	16.2476	19.2857	22.0871	24.7112	27.1792	33.8803	43.2296	51.1052	63.5780	73.7308
	WOA	**13.0262**	16.2485	19.2956	22.1131	24.7659	27.2944	34.2058	44.4341	53.0827	67.3240	78.7980
61060	HMPA	**12.7760**	15.9360	**19.0715**	21.7968	24.4427	26.9564	33.8090	43.8732	52.4651	**66.8101**	**78.4702**
	MPALS	**12.7760**	**15.9612**	**19.0715**	**21.8236**	24.4471	**26.9623**	**33.8126**	**43.8743**	**52.4959**	66.7598	78.3054
	IMPA	**12.7760**	15.9586	**19.0715**	**21.8236**	24.4469	26.9611	33.7998	43.7496	52.2823	66.3133	77.5785
	EO	**12.7760**	15.9561	**19.0715**	21.7738	24.4603	26.9033	33.7852	43.5996	51.9623	65.5818	76.4623
	SMA	**12.7760**	**15.9612**	**19.0715**	21.8158	**24.4555**	26.9354	33.8201	43.7208	52.1391	65.8079	76.5022
	HHA	**12.7760**	**15.9612**	19.0714	21.7998	24.4499	26.9209	33.8035	43.7474	52.2410	65.9761	76.7072
	ITSA	**12.7760**	15.9609	19.0688	21.7961	24.4326	26.9059	33.7003	43.3534	51.2062	63.9900	73.5589
	FPA	**12.7760**	15.9597	19.0527	21.7559	24.3265	26.7529	33.3989	42.7275	50.4755	63.0534	73.2689
	WOA	**12.7760**	15.9561	19.0714	21.7961	24.4546	26.9070	33.8037	43.8358	52.4526	66.7724	78.3516
38092	HMPA	**12.9773**	**16.1803**	**19.1849**	**21.9897**	**24.5939**	27.0578	34.2165	44.4489	**53.0741**	**67.7324**	**79.7253**
	MPALS	**12.9773**	**16.1803**	**19.1849**	**21.9897**	**24.5939**	**27.0604**	34.1779	**44.4692**	53.0561	67.6700	79.5990
	IMPA	**12.9773**	**16.1803**	**19.1849**	**21.9897**	24.5936	27.0576	34.1596	44.3111	52.9065	67.2159	78.6479
	EO	**12.9773**	**16.1803**	**19.1849**	**21.9897**	24.5932	27.0487	34.1377	44.1459	52.6201	66.6243	77.7269
	SMA	**12.9773**	**16.1803**	19.1848	21.9895	24.5933	27.0528	**34.2407**	44.3433	52.8407	66.7029	77.6195
	HHA	**12.9773**	**16.1803**	**19.1849**	21.9896	24.5937	27.0571	34.1695	44.2498	52.7537	66.8164	77.4952
	ITSA	**12.9773**	16.1801	19.1840	21.9856	24.5794	27.0310	34.0455	43.8333	51.8276	64.5886	75.0656

Table 2 (continued)

ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
		Threshold level (T)										
	FPA	**12.9773**	16.1798	19.1782	21.9620	24.5248	26.9561	33.7820	43.1767	51.2895	64.1631	74.3765
	WOA	**12.9773**	**16.1803**	19.1848	**21.9897**	24.5937	27.0589	34.2316	44.4040	53.0668	67.3431	78.7369
232038	HMPA	**11.9894**	**15.2615**	18.0535	20.8465	23.3736	25.7850	**32.6265**	**42.4824**	**50.8935**	**65.0718**	**76.5743**
	MPALS	**11.9894**	**15.2615**	**18.0647**	20.8422	23.3840	25.8017	32.5855	42.2584	50.4942	64.2486	75.4615
	IMPA	**11.9894**	**15.2615**	18.0630	**20.8513**	**23.3870**	25.8034	32.6181	42.3883	50.8759	64.8970	76.2589
	EO	**11.9894**	**15.2615**	18.0640	**20.8513**	23.3541	**25.8174**	32.5901	42.1200	50.3266	63.6009	74.4166
	SMA	**11.9894**	**15.2615**	18.0705	20.8505	23.3767	25.8112	32.6015	42.3107	50.6426	64.0702	74.4976
	HHA	**11.9894**	**15.2615**	18.0676	20.8196	23.3501	25.7771	32.5552	42.1880	50.4251	63.6507	74.2302
	ITSA	**11.9894**	15.2597	18.0602	20.8115	23.3318	25.7540	32.3807	41.7376	49.5379	61.9764	71.5020
	FPA	**11.9894**	15.2579	18.0401	20.7431	23.2564	25.6103	32.0471	41.1592	48.8792	61.1080	70.9056
	WOA	**11.9894**	15.2614	18.0602	20.8333	23.3489	25.7981	32.5638	42.3077	50.7038	64.3932	75.5097
108082	HMPA	**12.5693**	**15.8063**	**18.8167**	21.5955	**24.2374**	26.7373	33.5750	**43.6859**	**52.3720**	**67.0131**	**78.8480**
	MPALS	**12.5693**	**15.8063**	**18.8167**	21.5958	24.2370	26.7365	33.5683	43.5564	51.9991	66.2213	77.7627
	IMPA	**12.5693**	**15.8063**	**18.8167**	**21.5959**	**24.2374**	**26.7374**	**33.5788**	43.6180	52.3575	66.9093	78.7046
	EO	**12.5693**	**15.8063**	18.8159	21.5936	24.2362	26.7306	33.5240	43.4311	51.8002	65.7687	76.8378
	SMA	**12.5693**	15.8062	18.8166	21.5943	24.2360	26.7350	33.5440	43.5756	52.1084	66.0159	76.5886
	HHA	**12.5693**	**15.8063**	**18.8167**	21.5949	24.2370	26.7368	33.5681	43.6015	52.1346	65.9995	77.0055
	ITSA	**12.5693**	15.8061	18.8151	21.5912	24.2275	26.7215	33.4566	43.1682	51.1319	64.0936	74.0468
	FPA	**12.5693**	15.8057	18.8110	21.5733	24.1779	26.6020	33.2269	42.5672	50.4840	63.2370	73.5363
	WOA	**12.5693**	**15.8063**	**18.8167**	21.5952	24.2370	26.7370	33.5800	43.6928	52.4642	66.9466	78.6898
148089	HMPA	**12.9478**	**16.1630**	19.1835	**22.0607**	**24.7663**	27.3154	**34.4723**	44.5693	**53.3823**	**68.0859**	**79.9684**
	MPALS	**12.9478**	**16.1630**	**19.1936**	**22.0607**	**24.7663**	27.3186	34.4722	**44.5833**	53.3489	68.0141	79.9142
	IMPA	**12.9478**	**16.1630**	**19.1936**	**22.0607**	24.7659	27.3166	34.4618	44.5439	53.1590	67.5191	79.1512
	EO	**12.9478**	**16.1630**	**19.1936**	**22.0607**	24.7655	**27.3188**	34.3899	44.3684	52.9169	66.8248	78.0685

Table 2 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	SMA	**12.9478**	**16.1630**	**19.1936**	22.0601	24.7655	27.3156	34.4120	44.5078	53.0827	67.0968	78.2111
	HHA	**12.9478**	**16.1630**	19.1934	22.0600	24.7652	27.3167	34.4251	44.5050	53.0569	67.1294	78.3289
	ITSA	**12.9478**	16.1629	19.1919	22.0554	24.7574	27.2914	34.3146	44.0789	52.0651	64.9545	75.4968
	FPA	**12.9478**	16.1626	19.1827	22.0308	24.7056	27.1994	34.0462	43.5142	51.4772	64.3220	74.7993
	WOA	**12.9478**	**16.1630**	**19.1936**	22.0601	24.7656	27.3211	34.4474	44.5788	53.2832	67.9475	79.6025
189003	HMPA	**13.2054**	**16.4696**	19.5007	**22.3581**	25.0326	27.5744	34.5740	**44.6550**	53.2938	**67.8831**	**79.7136**
	MPALS	**13.2054**	**16.4696**	**19.5008**	**22.3581**	**25.0329**	**27.5745**	**34.5749**	44.6530	**53.3178**	67.7876	79.4895
	IMPA	**13.2054**	**16.4696**	**19.5008**	**22.3581**	25.0325	27.5732	34.5690	44.5528	53.1253	67.2542	78.7768
	EO	**13.2054**	**16.4696**	19.5005	22.3580	25.0284	27.5719	34.5339	44.4138	52.8900	66.8139	77.5620
	SMA	**13.2054**	**16.4696**	19.5005	**22.3581**	25.0319	27.5706	34.5594	44.5557	53.0181	66.9334	77.5981
	HHA	**13.2054**	**16.4696**	19.5005	22.3579	25.0324	27.5735	34.5640	44.5671	53.0982	67.0164	77.8655
	ITSA	**13.2054**	16.4695	19.4999	22.3554	25.0223	27.5556	34.4731	44.1498	52.0442	64.9163	74.9275
	FPA	**13.2054**	16.4693	19.4954	22.3340	24.9775	27.4724	34.2460	43.6624	51.5522	64.2297	74.5432
	WOA	**13.2054**	**16.4696**	19.5006	22.3580	25.0327	27.5735	34.5736	44.6301	53.2786	67.7431	79.4402
227092	HMPA	**11.0594**	13.7955	16.3538	18.7650	20.9785	23.1012	28.7518	**36.7705**	**43.3605**	**53.8887**	**61.5942**
	MPALS	**11.0594**	**13.7955**	**16.3631**	**18.7761**	20.9838	**23.1258**	**28.7622**	36.7689	43.3407	53.3630	60.3206
	IMPA	**11.0594**	**13.7955**	16.3538	18.7759	**20.9844**	23.1244	28.7485	36.6524	43.1024	53.0725	60.3119
	EO	**11.0594**	**13.7955**	**16.3631**	**18.7761**	20.9688	23.0786	28.6506	36.3285	42.5201	52.0984	59.3791
	SMA	**11.0594**	**13.7955**	**16.3631**	18.7748	20.9754	23.0911	28.7175	36.5762	42.9617	52.5778	59.4067
	HHA	**11.0594**	13.7946	16.3506	18.7695	20.9654	23.0996	28.6403	36.4008	42.5188	52.2366	59.0597
	ITSA	**11.0594**	13.7942	16.3556	18.7582	20.9268	23.0496	28.5472	35.9808	41.8248	50.7913	57.4370
	FPA	**11.0594**	13.7921	16.3239	18.6711	20.8266	22.8069	28.1012	34.8062	40.0230	48.5868	54.3894
	WOA	**11.0594**	13.7954	16.3583	18.7723	20.9687	23.1034	28.6989	36.6312	43.1405	53.2898	60.9455
277036	HMPA	**12.2424**	15.3336	**18.4297**	**21.1264**	23.4587	25.7730	32.3654	**41.6292**	**49.6277**	**62.6105**	72.9631

Table 2 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	MPALS	**12.2424**	15.3457	**18.4297**	**21.1264**	**23.4791**	25.7796	**32.3878**	41.5895	49.6000	62.5933	**72.9686**
	IMPA	**12.2424**	15.3445	**18.4297**	**21.1264**	23.4771	**25.7983**	32.3494	41.5412	49.3738	62.2023	72.1134
	EO	**12.2424**	**15.3498**	**18.4297**	**21.1264**	23.4488	25.7766	32.2796	41.3710	48.7290	60.6221	69.3596
	SMA	**12.2424**	15.3484	18.4291	21.1261	23.4675	25.7858	32.3382	41.4902	49.2417	61.5481	70.6606
	HHA	**12.2424**	15.3250	18.4277	21.1242	23.4611	25.7712	32.3351	41.4738	49.2546	61.6011	70.9646
	ITSA	**12.2424**	15.3441	18.4189	21.1023	23.4287	25.7287	32.1331	40.9056	47.9936	59.1799	67.7941
	FPA	**12.2424**	15.3317	18.4020	21.0474	23.3638	25.6304	31.8227	40.3776	47.1428	58.0004	66.4691
	WOA	**12.2424**	15.3291	18.4291	21.1253	23.4546	25.7719	32.3211	41.5945	49.5342	62.3864	72.4635
108070	HMPA	**12.5289**	**15.7032**	**18.5786**	21.2484	**23.8167**	26.2769	32.9322	**42.5770**	**50.7572**	**64.3405**	**75.0947**
	MPALS	**12.5289**	**15.7032**	**18.5786**	**21.2524**	23.8160	26.2832	**32.9412**	42.5758	50.7230	64.2061	74.8053
	IMPA	**12.5289**	**15.7032**	**18.5786**	21.2523	23.8160	**26.2846**	32.9311	42.4926	50.6478	63.7913	74.1403
	EO	**12.5289**	**15.7032**	**18.5786**	21.2412	23.8153	26.2766	32.9370	42.3525	50.1383	62.8736	73.1329
	SMA	**12.5289**	**15.7032**	18.5784	21.2470	23.8125	26.2777	32.9331	42.3986	50.4161	63.2744	73.0425
	HHA	**12.5289**	**15.7032**	18.5771	21.2513	23.8140	26.2769	32.9061	42.3935	50.4208	63.0299	72.9992
	ITSA	**12.5289**	15.7022	18.5733	21.2404	23.7981	26.2492	32.8079	41.9981	49.4379	61.3012	70.6755
	FPA	**12.5289**	15.7015	18.5572	21.2173	23.7482	26.1372	32.5283	41.3837	48.6483	60.2825	69.5329
	WOA	**12.5289**	**15.7032**	18.5774	21.2512	23.8141	26.2814	32.9233	42.4699	50.6247	63.7047	73.9346

Bold value shows the best results

Table 3 Performance evaluations using F-value on all images

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
Barbara	HMPA	**12.8944**	**16.0776**	**19.0524**	**21.8663**	**24.5163**	27.0285	33.8862	43.8030	**52.4811**	**67.2037**	**79.2864**
	MPALS	**12.8944**	**16.0776**	**19.0524**	**21.8663**	**24.5163**	**27.0286**	33.8897	**43.8122**	52.3960	67.0895	78.9849
	IMPA	**12.8944**	**16.0776**	**19.0524**	**21.8663**	24.5159	27.0282	33.8875	43.6951	52.1805	66.4355	78.1167
	EO	**12.8944**	**16.0776**	**19.0524**	**21.8663**	24.5137	27.0243	33.8759	43.5093	51.9943	65.9319	77.2050
	SMA	**12.8944**	**16.0776**	**19.0524**	21.8661	24.5152	27.0264	33.8861	43.6652	52.2117	66.1961	77.1260
	HHA	**12.8944**	16.0775	**19.0524**	21.8663	24.5161	27.0278	**33.8948**	43.6515	52.1600	66.1586	77.2514
	ITSA	**12.8944**	16.0773	19.0516	21.8640	24.5095	27.0100	33.7752	43.2863	51.2887	64.1723	74.5173
	FPA	**12.8944**	16.0769	19.0487	21.8442	24.4668	26.9338	33.4932	42.7338	50.6522	63.6220	74.0328
	WOA	**12.8944**	**16.0776**	**19.0524**	**21.8663**	24.5163	27.0284	33.9122	43.7849	52.4571	66.8121	78.7664
Airplane	HMPA	**12.2274**	**15.5081**	**18.3280**	20.9397	23.4208	**25.7816**	32.1946	41.4299	**49.5319**	**62.9455**	**73.5826**
	MPALS	**12.2274**	**15.5081**	**18.3280**	**20.9444**	23.4179	**25.7816**	**32.2006**	**41.4477**	49.5195	62.7600	73.3507
	IMPA	**12.2274**	**15.5081**	**18.3280**	**20.9444**	23.4237	25.7803	32.1913	41.4048	49.2791	62.4401	72.6822
	EO	**12.2274**	**15.5081**	**18.3280**	20.9349	23.4266	25.7767	32.1288	41.2051	48.9190	61.2739	71.0511
	SMA	**12.2274**	**15.5081**	**18.3280**	20.9416	23.4165	25.7792	32.1713	41.3664	49.2087	61.8210	71.6404
	HHA	**12.2274**	**15.5081**	**18.3278**	20.9395	23.4270	25.7804	32.1872	41.3813	49.0345	61.5561	71.3104
	ITSA	**12.2274**	15.5079	18.3257	20.9347	23.4123	25.7562	32.0726	40.8540	48.1415	59.5682	68.6942
	FPA	**12.2274**	15.5073	18.3178	20.9027	23.3325	25.6257	31.7821	40.2425	47.2591	58.3053	67.1785
	WOA	**12.2274**	**15.5081**	18.3279	20.9443	23.4270	25.7812	32.1933	41.4612	49.2387	62.2972	72.5960
Mandrill	HMPA	**12.2772**	**15.3757**	**18.2736**	**20.9959**	**23.5384**	26.0414	32.9129	**42.7622**	**51.2897**	**65.1751**	**76.4023**
	MPALS	**12.2772**	**15.3757**	**18.2736**	**20.9959**	23.5379	**26.0414**	**32.9179**	**42.7637**	51.2859	65.0837	75.9578
	IMPA	**12.2772**	**15.3757**	**18.2736**	**20.9959**	23.5363	26.0394	32.9051	42.7239	51.0273	64.5601	75.2007
	EO	**12.2772**	**15.3757**	**18.2736**	**20.9959**	23.5378	26.0389	32.8671	42.5043	50.6732	63.8579	74.3417
	SMA	**12.2772**	**15.3757**	**18.2736**	20.9958	23.5359	26.0392	32.8926	42.6785	50.9259	64.1848	74.3093
	HHA	**12.2772**	**15.3757**	**18.2736**	20.9958	23.5380	26.0376	32.8553	42.5895	50.7645	63.9880	74.2377
	ITSA	**12.2772**	**15.3757**	18.2730	20.9931	23.5315	26.0181	32.8004	42.1880	49.8101	62.0125	71.1463

Table 3 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	FPA	**12.2772**	**15.3754**	**18.2664**	20.9663	23.4908	25.8818	32.4825	41.5565	48.8883	60.8398	70.1299
	WOA	**12.2772**	**15.3757**	**18.2736**	**20.9959**	**23.5384**	26.0396	32.8978	42.7290	50.9661	64.7652	75.8249
Lena	HMPA	**12.4048**	**15.4167**	**18.1610**	**20.8085**	**23.2547**	25.5361	32.0583	41.4808	**49.7292**	**63.4561**	**74.5055**
	MPALS	**12.4048**	**15.4167**	**18.1610**	**20.8085**	**23.2547**	25.5526	32.0626	**41.4850**	49.7167	63.3655	74.2935
	IMPA	**12.4048**	**15.4167**	**18.1610**	**20.8085**	23.2543	25.5545	32.0532	41.4018	49.4810	62.9427	73.5091
	EO	**12.4048**	**15.4167**	18.1593	**20.8085**	23.2544	25.5036	32.0130	41.2126	49.1301	62.0035	71.8218
	SMA	**12.4048**	**15.4167**	18.1593	20.8083	23.2534	25.5329	32.0301	41.3603	49.4356	62.2989	72.0302
	HHA	**12.4048**	**15.4167**	18.1605	20.8083	23.2543	25.5454	32.0468	41.2830	49.2091	61.9919	71.8495
	ITSA	**12.4048**	15.4164	18.1593	20.8037	23.2406	25.5068	31.9026	40.8692	48.3668	60.0749	69.4590
	FPA	**12.4048**	15.4161	18.1494	20.7713	23.1686	25.3946	31.5934	40.2612	47.4949	59.0258	68.3088
	WOA	**12.4048**	**15.4167**	**18.1610**	20.8083	23.2545	**25.5582**	**32.0655**	41.3781	49.4402	62.4487	72.7444

Bold value shows the best results

Table 4 Performance evaluations using PSNR on all images

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
12003	HMPA	**14.4664**	17.0600	18.3656	20.3093	21.5585	22.4888	25.3361	**29.2292**	**31.8552**	**35.5342**	**37.9395**
	MPALS	**14.4664**	17.0600	18.3656	20.3492	21.5382	22.4700	25.3277	29.1136	31.7124	35.1695	37.7015
	IMPA	**14.4664**	17.0600	18.3656	20.3493	21.6213	22.5072	25.5659	29.4008	31.9203	35.2554	37.3621
	EO	**14.4664**	17.0600	18.3656	**20.3636**	**21.7099**	**22.8190**	**25.8319**	29.2168	31.8345	35.2145	37.5518
	SMA	**14.4664**	17.0516	18.3623	20.3046	21.6142	22.6220	25.4177	29.1341	31.5422	34.7098	36.5032
	HHA	**14.4664**	17.0586	18.3615	20.3528	21.6158	22.4904	25.3795	28.9297	31.1825	34.2928	36.4958
	ITSA	**14.4664**	**17.0808**	**18.3675**	20.3176	21.5725	22.4896	25.3554	28.3497	30.0872	33.0718	34.7081
	FPA	**14.4664**	17.0409	18.2953	20.2155	21.4905	22.2805	24.7820	27.6051	29.4121	32.1529	34.2605
	WOA	**14.4664**	17.0696	18.3579	20.3525	21.6098	22.4598	25.3510	29.2253	31.7159	35.3186	37.5345
61060	HMPA	**16.2867**	**17.1279**	**18.7869**	20.3282	**21.2248**	22.7102	**25.7415**	29.0261	31.4855	35.0411	37.4589
	MPALS	**16.2867**	16.5301	**18.7869**	**20.7489**	21.1836	**22.7211**	25.5405	**29.1335**	**31.7247**	**35.3358**	**37.9970**
	IMPA	**16.2867**	16.5899	**18.7869**	**20.7489**	21.1608	22.6571	25.5977	28.6394	30.9077	33.9779	35.7271
	EO	**16.2867**	16.6497	**18.7869**	19.9676	21.0585	22.0213	25.2043	28.2533	30.0770	32.2269	33.9339
	SMA	**16.2867**	16.5301	18.7867	20.6295	21.0911	22.4372	25.8270	28.7612	30.9514	33.9934	35.5897
	HHA	**16.2867**	16.5301	18.7829	20.3985	21.0996	22.3203	25.6375	28.8090	31.1156	34.0393	35.8146
	ITSA	16.2858	16.5240	18.7386	20.4722	21.1009	22.4833	25.3194	28.3871	30.3330	32.5400	34.6282
	FPA	**16.2867**	16.5073	18.6787	20.6450	20.8716	22.0928	24.7552	27.7593	29.6672	31.8297	34.5425
	WOA	**16.2867**	16.6498	18.7865	20.3303	21.0824	22.3591	25.4131	29.0793	31.4004	34.8459	37.2226
38092	HMPA	**14.0889**	**16.8727**	19.0832	20.6639	21.8331	22.5507	**25.2608**	**29.3674**	**31.8266**	**35.2255**	**37.6994**
	MPALS	**14.0889**	**16.8727**	19.0832	20.6639	21.8331	22.4817	25.2127	29.2490	31.7802	35.1257	37.6970
	IMPA	**14.0889**	**16.8727**	19.0832	20.6639	**21.8343**	22.5197	25.1351	28.7912	31.0273	34.3721	36.7862
	EO	**14.0889**	**16.8727**	**19.0840**	20.6639	21.8333	**22.7682**	25.0484	28.5230	30.5813	33.6575	35.5068
	SMA	**14.0889**	**16.8727**	19.0822	20.6627	21.8203	22.6523	25.2447	29.0130	31.2944	34.1086	35.8762
	HHA	**14.0889**	16.8721	19.0835	20.6663	21.8156	22.4719	25.0838	28.8800	31.1769	33.9901	35.8811
	ITSA	14.0892	16.8701	19.0685	20.6452	21.7865	22.6695	25.0041	28.1805	29.9647	32.5440	34.4988

Table 4 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	FPA	**14.0889**	16.8750	19.0391	20.5574	21.5962	22.5658	24.5601	27.3471	29.5030	32.0048	34.2886
	WOA	**14.0889**	16.8716	19.0839	**20.6664**	21.8195	22.3955	25.2368	29.1929	31.8156	34.4842	36.8215
232038	HMPA	**13.0048**	**15.1266**	17.6735	19.7374	**20.8021**	**21.9461**	24.5608	29.4244	**31.8302**	**35.7076**	**38.3264**
	MPALS	**13.0048**	**15.1266**	18.3971	19.7138	20.4868	21.6560	24.4168	28.4366	31.6141	34.9959	37.8458
	IMPA	**13.0048**	**15.1266**	18.6915	19.5712	20.5223	21.7712	24.4908	28.7895	31.4566	34.7564	37.1517
	EO	**13.0048**	**15.1266**	18.5434	19.7138	20.7919	21.3955	25.1630	**29.4729**	31.8111	35.1984	37.6261
	SMA	**13.0048**	15.1243	**19.5701**	19.7189	20.5650	22.1354	24.6467	29.1071	31.7075	34.6786	36.6348
	HHA	**13.0048**	**15.1266**	19.1214	**19.7566**	20.4483	21.8228	24.4034	29.0830	31.2150	34.2940	36.3549
	ITSA	**13.0048**	15.1154	19.1429	19.4790	20.6487	21.9786	**24.7662**	28.0055	30.2378	33.1431	34.7176
	FPA	**13.0048**	15.1175	19.0884	19.5061	20.3532	21.8932	24.3921	27.4516	29.7525	32.4362	34.7544
	WOA	**13.0048**	15.1175	18.2332	19.6839	20.4912	22.0652	24.2410	28.8736	31.6374	34.6381	37.1100
108082	HMPA	**14.5423**	16.0369	17.1311	**17.8709**	18.9918	19.5365	22.1493	29.2662	**31.9837**	**35.2108**	**37.5987**
	MPALS	**14.5423**	16.0369	17.1311	17.8183	18.9918	19.5180	21.7835	27.3588	31.5110	34.4226	36.4667
	IMPA	**14.5423**	16.0369	17.1311	17.8227	**19.0544**	19.6068	21.4437	29.3494	30.6942	32.8243	35.3342
	EO	**14.5423**	16.0369	**17.1972**	18.0962	19.0759	**19.7447**	23.6272	28.3542	31.1115	34.2267	36.7240
	SMA	**14.5423**	16.0222	17.1331	17.9592	19.0196	19.6199	23.1615	29.0642	31.0705	34.1874	35.6185
	HHA	**14.5423**	16.0369	17.1173	17.8823	19.0179	19.5242	21.8343	29.2973	31.5417	34.1247	36.6478
	ITSA	**14.5423**	16.0564	17.1496	17.8387	18.9827	19.5765	22.8826	27.2873	29.7907	32.7242	34.8073
	FPA	**14.5423**	**16.0592**	17.1865	17.7606	18.9614	19.5815	**23.1759**	26.8195	29.2822	32.3475	34.6654
	WOA	**14.5423**	16.0369	17.1172	17.8426	19.0167	19.5699	21.6444	**29.3911**	31.9492	35.0286	37.5441
148089	HMPA	**15.5498**	**17.5990**	**18.1363**	20.3767	21.8960	23.1969	25.5518	29.0906	**31.7109**	**35.3564**	**37.8787**
	MPALS	**15.5498**	**17.5990**	18.0009	20.3767	21.8960	23.2416	25.5230	29.0197	31.4200	35.0081	37.3717
	IMPA	**15.5498**	**17.5990**	18.0009	**20.3789**	21.9004	23.2908	25.6507	29.3871	31.7509	35.0954	37.4444
	EO	**15.5498**	**17.5990**	18.0009	20.3767	**21.9145**	**23.3506**	**25.8366**	**29.5193**	31.0376	35.2013	37.8245

Table 4 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	SMA	**15.5498**	**17.5990**	17.9993	20.3482	21.8654	23.2200	25.5582	28.9096	31.1293	34.2591	36.3460
	HHA	**15.5498**	**17.5990**	17.9906	20.3203	21.8576	23.2134	25.3659	29.0570	31.1320	34.3134	36.4823
	ITSA	**15.5498**	17.6055	17.9870	20.3194	21.7714	23.1200	25.0910	28.1924	29.7129	32.4934	34.7773
	FPA	**15.5498**	17.5745	18.0000	20.2589	21.7950	22.9368	24.9030	27.5831	29.2074	32.0235	33.8759
	WOA	**15.5498**	17.5990	18.0037	20.3555	21.8860	23.2976	25.3381	29.2563	31.5223	35.1979	37.7269
189003	HMPA	**14.5586**	17.0126	19.1787	**20.4719**	21.8177	**23.0326**	**25.6329**	**28.9353**	**31.4806**	**35.1380**	**37.5532**
	MPALS	**14.5586**	17.0126	**19.1809**	20.4711	21.8292	23.0316	25.6260	28.8299	31.2434	34.8429	37.0843
	IMPA	**14.5586**	17.0126	19.1805	20.4713	21.8358	23.0324	25.7002	28.8892	31.3220	34.2056	36.2794
	EO	**14.5586**	17.0140	19.1751	20.4715	**21.8800**	23.0420	25.6328	28.7179	31.1879	34.0964	35.4502
	SMA	**14.5586**	**17.0182**	19.1746	20.4630	21.8202	23.0180	25.6200	28.7440	31.0599	34.1384	35.9329
	HHA	**14.5556**	17.0173	19.1749	20.4656	21.8254	23.0222	25.6300	28.7908	31.1782	34.2853	36.0893
	ITSA	**14.5586**	17.0127	19.1710	20.4418	21.7925	22.9649	25.4341	28.0506	29.6283	32.2608	34.2806
	FPA	**14.5586**	17.0078	19.1383	20.4156	21.7026	22.7326	24.8929	27.4863	29.4045	32.0807	34.3324
	WOA	**14.5586**	**17.0126**	19.1771	20.4670	21.8278	**23.0326**	25.6327	28.9414	31.4765	35.0837	37.5505
227092	HMPA	**15.6460**	18.3668	22.1676	24.5920	25.5721	26.6858	29.7139	33.1146	**35.8689**	**39.5852**	**42.0312**
	MPALS	**15.6460**	18.3668	22.3381	24.7540	25.2780	26.7025	**29.7670**	33.1514	35.8639	39.1879	41.3994
	IMPA	**15.6460**	18.3668	22.1676	24.7475	25.2198	26.7141	29.6789	**33.6072**	35.0020	39.8661	42.0023
	EO	**15.6460**	18.3668	22.3381	24.7540	26.1112	26.7588	29.7819	33.0030	35.3755	38.6554	40.8635
	SMA	**15.6460**	18.3668	22.3381	24.7412	25.5730	**26.8740**	29.7577	33.0151	35.3436	38.4741	40.2057
	HHA	**15.6460**	18.3527	22.2598	**24.7808**	**25.7395**	26.6993	29.3257	32.4273	34.4291	37.6098	39.5880
	ITSA	**15.6460**	**18.3871**	22.2877	24.6811	25.5510	26.6599	29.4714	32.2437	33.6239	36.8020	38.7230
	FPA	**15.6460**	18.3801	22.2330	24.6393	25.3999	26.0011	28.7261	30.7196	32.6772	35.1214	37.4921
	WOA	**15.6460**	18.3704	**22.3950**	24.7595	25.5167	26.7197	29.6993	32.9739	35.5028	38.7427	41.1640
277036	HMPA	15.2290	16.6456	18.8667	20.7566	21.7454	**22.9205**	**25.7304**	**30.0579**	**32.7465**	**36.2037**	**38.8644**

Table 4 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	MPALS	15.2290	15.8882	18.8667	20.7566	21.6355	22.8467	25.7042	29.3066	32.7453	36.1068	38.6984
	IMPA	15.2290	15.9687	18.8667	**20.7567**	21.6161	22.0218	25.6397	29.9015	32.3288	35.8429	37.9036
	EO	15.2290	15.6259	18.8667	20.7566	**21.8448**	22.9279	25.7236	29.5907	31.2485	33.9848	35.5927
	SMA	15.2290	15.6865	18.8912	20.7622	21.7049	22.9871	25.6649	29.4147	32.0428	34.9737	36.7112
	HHA	15.2290	**16.9108**	18.8687	20.7555	21.7143	22.7969	25.5589	29.8806	32.1110	35.1292	37.4408
	ITSA	**15.2297**	15.7160	**19.0130**	20.7219	21.6742	22.8602	25.1740	28.5089	30.3077	33.2220	34.8068
	FPA	15.2290	15.9793	18.9887	20.6342	21.5744	22.5962	25.0183	27.8845	29.8873	32.7984	34.9823
	WOA	15.2290	16.7746	18.8442	20.7490	21.6842	22.8331	25.5381	29.9432	32.6988	36.2081	38.6007
108070	HMPA	**12.8967**	14.2363	**15.7304**	17.2388	21.1568	**22.3775**	25.4960	30.0759	**32.7409**	36.4130	38.5081
	MPALS	**12.8967**	14.2363	**15.7304**	16.8245	20.4843	22.1818	25.6524	29.6496	31.9348	35.4124	37.5079
	IMPA	**12.8967**	14.2363	**15.7304**	16.8258	20.9856	22.3401	25.7956	30.2021	32.6008	35.8996	38.1906
	EO	**12.8967**	14.2363	**15.7304**	18.3165	**21.6110**	22.5770	**27.2898**	**30.8910**	33.4024	**37.0986**	**39.2475**
	SMA	**12.8967**	14.2363	15.7187	17.2666	20.9839	22.3313	25.9540	30.1560	32.3677	34.7148	36.7590
	HHA	**12.8967**	14.2364	15.6517	16.7443	20.9547	22.1834	25.7675	29.8580	31.8108	34.8418	37.0432
	ITSA	**12.8967**	**14.2445**	15.5489	17.2491	20.9722	22.2086	25.4533	28.7819	31.1449	33.7012	35.7406
	FPA	**12.8967**	14.2534	15.6943	**17.6129**	21.1066	21.6043	25.0025	28.0944	29.9532	33.1041	34.8328
	WOA	**12.8967**	14.2363	15.6326	16.8710	21.2967	22.3171	26.0798	29.8002	31.9648	35.2684	37.5089

Bold value shows the best results

Table 5 Performance evaluations using PSNR on all images

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
Barbara	HMPA	**14.9240**	17.3240	19.0009	20.6565	21.9705	23.0243	24.9945	27.5578	31.1921	**34.9093**	**37.5475**
	MPALS	**14.9240**	17.3240	19.0009	20.6529	**21.9707**	**23.0343**	25.0026	27.2661	30.2906	34.3971	37.1629
	IMPA	**14.9240**	17.3240	19.0009	20.6529	21.9511	23.0284	25.0571	**27.9069**	30.8051	34.4800	36.8770
	EO	**14.9240**	17.3240	19.0009	20.6529	21.9507	22.9850	**25.1504**	28.4165	30.8357	33.9630	36.3658
	SMA	**14.9240**	17.3240	19.0029	**20.6893**	21.9456	23.0187	25.0514	27.6701	30.7339	33.8910	36.0041
	HHA	**14.9240**	17.3238	**19.0039**	20.6711	21.9642	23.0251	25.0677	27.1036	30.8170	33.7286	35.9699
	ITSA	**14.9240**	17.3263	19.0016	20.6731	21.9297	22.9582	24.9636	27.2391	29.5860	32.0192	34.1709
	FPA	**14.9240**	**17.3275**	18.9932	20.6021	21.7386	22.7188	24.4337	26.9418	29.0392	32.1906	34.0938
	WOA	**14.9240**	17.3240	19.0028	20.6540	21.9688	23.0322	25.1236	27.4380	**31.3704**	34.5177	37.1461
Airplane	HMPA	**16.0751**	18.8125	20.4488	21.4168	22.0850	24.1641	**27.4491**	30.4306	32.5886	36.1718	38.6647
	MPALS	**16.0751**	18.8125	20.4488	21.4567	22.1102	**24.2119**	27.4413	30.5363	**32.8496**	**36.7006**	**39.4306**
	IMPA	**16.0751**	18.8125	20.4488	21.4567	22.0430	23.9694	27.2074	29.9554	31.3038	34.2763	37.9031
	EO	**16.0751**	18.8125	20.4488	21.3769	22.0345	23.3955	26.8529	29.4341	31.0096	33.0957	34.7722
	SMA	**16.0751**	18.8125	20.4479	21.4704	**22.1347**	24.0732	27.2978	**30.5658**	32.6669	35.6705	37.2157
	HHA	**16.0751**	18.8199	20.4492	21.4453	22.0289	24.0637	27.4101	30.5302	31.6999	35.0707	36.8259
	ITSA	**16.0751**	18.8509	**20.4678**	**21.4998**	22.1174	23.9716	27.2286	29.5944	30.9419	33.1887	35.1232
	FPA	**16.0751**	**18.9080**	20.4164	21.4358	22.0026	23.7785	26.7545	28.9818	30.5672	32.7682	34.7633
	WOA	**16.0751**	18.8125	20.4486	21.4557	22.0525	**24.2119**	27.4476	30.5848	32.6633	36.1629	38.3843
Mandrill	HMPA	**16.0224**	18.6744	**20.4448**	22.1853	23.1789	**24.2564**	26.4394	29.8300	32.4032	36.2305	38.9516
	MPALS	**16.0224**	18.6593	**20.4448**	22.1853	23.0436	**24.2564**	26.4528	29.8249	32.6185	36.4763	39.2159
	IMPA	**16.0224**	18.6593	**20.4448**	22.1853	22.9646	24.1717	26.2799	**30.1739**	**32.7958**	36.4529	**39.2650**
	EO	**16.0224**	18.6845	**20.4448**	22.1853	**23.3156**	24.1418	26.0679	29.6342	31.8335	35.4862	37.8609
	SMA	**16.0224**	18.7349	20.4439	22.2120	23.1538	24.2239	26.3670	29.9145	32.2900	35.5242	37.6226
	HHA	**16.0224**	18.6543	**20.4448**	22.1982	23.2886	24.2322	26.2142	29.5812	31.5313	34.4138	36.5965
	ITSA	**16.0224**	18.7067	20.4437	**22.2373**	23.1951	24.2467	26.3600	29.2519	31.2740	34.0989	35.6996

Table 5 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	FPA	**16.0224**	**18.7406**	20.3874	22.0470	22.8431	23.7555	25.7358	28.5234	29.9231	32.6412	34.9721
	WOA	**16.0224**	18.6845	20.4448	22.1853	23.2334	24.2377	**26.5001**	29.8506	31.6659	35.4391	38.0163
Lena	HMPA	**14.5905**	17.2956	**19.0590**	19.9951	20.8807	**21.4687**	26.2804	30.0561	32.5912	36.2249	**38.9853**
	MPALS	**14.5905**	17.2956	**19.0590**	19.9951	20.8807	21.0678	26.3008	29.9871	32.6643	**36.4160**	38.9121
	IMPA	**14.5905**	17.2956	**19.0590**	19.9951	**20.8866**	21.0621	26.7188	**30.4612**	**32.9366**	36.2528	38.8436
	EO	**14.5905**	17.2956	19.0163	19.9951	20.8829	22.0402	**26.7955**	30.1535	32.3900	35.9136	38.0418
	SMA	**14.5905**	17.2956	19.0150	20.0009	20.8757	21.3079	25.9504	29.8573	32.4553	35.4793	37.1676
	HHA	**14.5905**	**17.2959**	19.0501	**20.0045**	20.8730	21.0615	26.2220	29.7627	31.9222	34.6143	36.3550
	ITSA	14.5894	17.2786	19.0480	19.9475	20.8641	21.1903	25.6903	29.0530	31.1515	33.3735	36.0050
	FPA	**14.5905**	17.2821	18.9990	19.8287	20.7826	21.3547	24.5859	27.9321	30.0111	33.0308	34.8313
	WOA	**14.5905**	17.2944	**19.0590**	20.0045	20.8762	20.9505	26.4470	29.6796	32.1200	35.1233	36.7990

Bold value shows the best results

Table 6 Performance evaluations using SSIM on all images

ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
12003	HMPA	**0.7954**	0.8857	**0.9148**	0.9419	0.9542	0.9622	0.9754	**0.9856**	**0.9881**	**0.9900**	**0.9906**
	MPALS	**0.7954**	0.8857	**0.9148**	0.9425	0.9540	0.9621	0.9754	0.9849	0.9879	0.9898	0.9905
	IMPA	**0.7954**	0.8857	**0.9148**	0.9425	0.9548	0.9623	0.9767	0.9854	0.9880	0.9898	0.9904
	EO	**0.7954**	0.8857	**0.9148**	**0.9427**	**0.9556**	**0.9636**	**0.9775**	0.9854	0.9879	0.9898	0.9904
	SMA	**0.7954**	0.8854	**0.9148**	0.9420	0.9546	0.9627	0.9759	0.9849	0.9876	0.9895	0.9901
	HHA	**0.7954**	0.8857	0.9147	0.9426	0.9548	0.9621	0.9757	0.9845	0.9871	0.9892	0.9901
	ITSA	**0.7954**	**0.8863**	0.9148	0.9423	0.9543	0.9616	0.9755	0.9830	0.9855	0.9884	0.9892
	FPA	**0.7954**	0.8855	0.9140	0.9407	0.9536	0.9595	0.9723	0.9808	0.9843	0.9875	0.9890
	WOA	**0.7954**	0.8859	0.9146	0.9426	0.9548	0.9621	0.9757	0.9853	0.9879	0.9899	0.9905
61060	HMPA	**0.8475**	**0.8785**	**0.9413**	0.9507	**0.9576**	0.9737	**0.9847**	0.9914	0.9940	0.9960	0.9967
	MPALS	**0.8475**	0.8521	**0.9413**	0.9534	0.9571	**0.9742**	0.9841	**0.9916**	**0.9942**	**0.9961**	**0.9968**
	IMPA	**0.8475**	0.8548	**0.9413**	0.9534	0.9570	0.9739	0.9839	0.9903	0.9930	0.9951	0.9955
	EO	**0.8475**	0.8574	**0.9413**	0.9483	0.9555	0.9652	0.9828	0.9899	0.9922	0.9933	0.9945
	SMA	**0.8475**	0.8521	**0.9413**	**0.9526**	0.9561	0.9705	0.9844	0.9907	0.9930	0.9951	0.9956
	HHA	**0.8475**	0.8521	**0.9413**	0.9511	0.9560	0.9692	0.9840	0.9908	0.9934	0.9952	0.9959
	ITSA	**0.8475**	0.8519	0.9410	0.9514	0.9560	0.9716	0.9825	0.9896	0.9917	0.9936	0.9951
	FPA	**0.8475**	0.8513	0.9400	0.9518	0.9542	0.9666	0.9794	0.9869	0.9905	0.9929	0.9950
	WOA	**0.8475**	0.8574	**0.9413**	0.9506	0.9560	0.9687	0.9836	0.9915	0.9940	0.9960	0.9967
38092	HMPA	**0.8728**	**0.9323**	**0.9568**	**0.9692**	**0.9761**	0.9791	**0.9880**	**0.9938**	**0.9955**	**0.9968**	**0.9972**
	MPALS	**0.8728**	**0.9323**	**0.9568**	**0.9692**	**0.9761**	0.9789	0.9878	**0.9938**	**0.9955**	**0.9968**	**0.9972**
	IMPA	**0.8728**	**0.9323**	**0.9568**	**0.9692**	**0.9761**	0.9790	0.9875	0.9932	0.9950	0.9965	0.9970
	EO	**0.8728**	**0.9323**	**0.9568**	**0.9692**	**0.9761**	**0.9798**	0.9871	0.9930	0.9946	0.9961	0.9966
	SMA	**0.8728**	**0.9323**	**0.9568**	0.9691	0.9760	0.9794	0.9879	0.9935	0.9952	0.9964	0.9967
	HHA	**0.8728**	**0.9323**	**0.9568**	0.9692	0.9759	0.9787	0.9873	0.9933	0.9951	0.9963	0.9968
	ITSA	**0.8728**	**0.9323**	**0.9568**	0.9690	0.9757	0.9794	0.9868	0.9922	0.9938	0.9954	0.9962

Table 6 (continued)

ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
		Threshold level (T)										
	FPA	**0.8728**	**0.9323**	**0.9568**	0.9684	0.9744	0.9787	0.9850	0.9907	0.9934	0.9951	0.9962
	WOA	**0.8728**	**0.9323**	**0.9568**	**0.9692**	0.9760	0.9785	0.9879	0.9937	0.9955	0.9966	0.9971
232038	HMPA	**0.6517**	0.7972	0.8740	**0.9322**	0.9427	0.9535	0.9751	**0.9909**	**0.9939**	**0.9963**	**0.9970**
	MPALS	**0.6517**	0.7972	0.8941	0.9319	0.9416	0.9549	0.9743	0.9881	0.9936	0.9957	0.9968
	IMPA	**0.6517**	0.7972	0.9026	0.9280	0.9427	0.9558	0.9752	0.9890	0.9930	0.9956	0.9965
	EO	**0.6517**	0.7972	0.8982	0.9319	**0.9457**	**0.9617**	**0.9786**	**0.9909**	**0.9939**	0.9958	0.9966
	SMA	**0.6517**	0.7972	**0.9273**	0.9321	0.9432	0.9595	0.9754	0.9900	0.9936	0.9956	0.9963
	HHA	**0.6517**	0.7972	0.9142	**0.9322**	0.9413	0.9562	0.9743	0.9898	0.9929	0.9952	0.9962
	ITSA	**0.6517**	0.7970	0.9167	0.9264	0.9437	0.9569	0.9757	0.9867	0.9906	0.9941	0.9951
	FPA	**0.6517**	**0.7983**	0.9154	0.9274	0.9402	0.9566	0.9734	0.9851	0.9903	0.9937	0.9953
	WOA	**0.6517**	0.7971	0.8892	0.9311	0.9415	0.9588	0.9734	0.9894	0.9936	0.9956	0.9965
108082	HMPA	**0.5317**	0.6703	0.7613	0.7974	0.8468	0.8692	0.9213	**0.9850**	**0.9905**	**0.9946**	**0.9959**
	MPALS	**0.5317**	0.6703	0.7613	0.7938	0.8468	0.8691	0.9172	0.9718	0.9904	0.9939	0.9952
	IMPA	**0.5317**	0.6703	0.7613	0.7938	0.8510	0.8705	0.9177	0.9841	0.9873	0.9918	0.9942
	EO	**0.5317**	0.6703	**0.7616**	0.8130	**0.8550**	**0.8798**	**0.9439**	0.9821	0.9888	0.9935	0.9953
	SMA	**0.5317**	0.6676	0.7613	0.8036	0.8483	0.8713	0.9246	0.9842	0.9895	0.9933	0.9946
	HHA	**0.5317**	0.6703	0.7587	0.7990	0.8484	0.8691	0.9189	0.9849	0.9901	0.9934	0.9954
	ITSA	**0.5317**	0.6749	0.7572	**0.8006**	0.8478	0.8693	0.9357	0.9805	0.9859	0.9918	0.9935
	FPA	**0.5317**	**0.6780**	0.7579	0.7917	0.8462	0.8691	0.9369	0.9706	0.9829	0.9906	0.9936
	WOA	**0.5317**	0.6703	0.7587	0.7978	0.8480	0.8703	0.9174	0.9854	0.9902	0.9944	0.9959
148089	HMPA	**0.8300**	**0.8975**	**0.9080**	**0.9460**	0.9611	0.9716	0.9822	0.9912	**0.9947**	**0.9963**	**0.9969**
	MPALS	**0.8300**	**0.8975**	0.9055	**0.9460**	0.9611	0.9720	0.9821	0.9911	0.9939	0.9961	0.9968
	IMPA	**0.8300**	**0.8975**	0.9055	**0.9460**	0.9611	0.9722	0.9831	0.9919	0.9943	0.9961	0.9968
	EO	**0.8300**	**0.8975**	0.9055	**0.9460**	**0.9612**	**0.9725**	**0.9841**	**0.9921**	**0.9947**	**0.9963**	**0.9969**

Table 6 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	SMA	**0.8300**	**0.8975**	0.9055	0.9457	0.9610	0.9720	0.9827	0.9909	0.9936	0.9957	0.9965
	HHA	**0.8300**	**0.8975**	0.9055	0.9456	0.9609	0.9719	0.9820	0.9911	0.9936	0.9958	0.9966
	ITSA	**0.8300**	**0.8975**	0.9052	0.9450	0.9605	0.9712	0.9806	0.9894	0.9917	0.9945	0.9959
	FPA	**0.8300**	0.8962	0.9057	0.9443	0.9606	0.9698	0.9796	0.9880	0.9909	0.9941	0.9954
	WOA	**0.8300**	**0.8975**	0.9055	0.9458	0.9610	0.9724	0.9819	0.9915	0.9940	0.9961	0.9969
189003	HMPA	**0.8357**	**0.9067**	**0.9420**	**0.9557**	**0.9656**	**0.9714**	0.9807	**0.9866**	**0.9887**	**0.9903**	**0.9908**
	MPALS	**0.8357**	**0.9067**	**0.9420**	**0.9557**	**0.9656**	**0.9714**	0.9807	**0.9866**	**0.9887**	**0.9903**	**0.9908**
	IMPA	**0.8357**	**0.9067**	**0.9420**	**0.9557**	**0.9656**	**0.9714**	0.9807	0.9865	0.9886	0.9900	0.9906
	EO	**0.8357**	**0.9067**	**0.9420**	0.9556	0.9655	**0.9714**	**0.9811**	0.9864	0.9884	0.9899	0.9903
	SMA	**0.8357**	**0.9066**	**0.9420**	**0.9557**	**0.9656**	**0.9714**	0.9807	0.9864	0.9884	0.9900	0.9905
	HHA	**0.8357**	**0.9066**	**0.9420**	**0.9557**	**0.9656**	**0.9714**	0.9806	0.9865	0.9886	0.9901	0.9906
	ITSA	**0.8357**	0.9065	**0.9420**	0.9556	0.9653	0.9712	0.9803	0.9853	0.9870	0.9890	0.9899
	FPA	**0.8357**	0.9065	0.9415	0.9549	0.9645	0.9702	0.9786	0.9842	0.9867	0.9889	0.9899
	WOA	**0.8357**	**0.9066**	**0.9420**	**0.9557**	**0.9656**	**0.9714**	0.9806	**0.9866**	**0.9887**	**0.9903**	**0.9908**
227092	HMPA	**0.7616**	0.8453	0.9198	0.9501	0.9561	0.9621	**0.9738**	**0.9816**	**0.9841**	**0.9857**	**0.9862**
	MPALS	**0.7616**	0.8453	0.9237	**0.9530**	0.9546	0.9623	**0.9738**	0.9806	0.9836	0.9853	0.9859
	IMPA	**0.7616**	0.8453	0.9201	0.9527	0.9543	0.9624	**0.9738**	**0.9816**	**0.9841**	**0.9857**	**0.9862**
	EO	**0.7616**	0.8453	0.9237	0.9530	**0.9590**	0.9625	0.9743	0.9810	0.9835	0.9853	0.9858
	SMA	**0.7616**	0.8453	0.9237	0.9528	0.9563	**0.9632**	0.9737	0.9803	0.9830	0.9849	0.9855
	HHA	**0.7616**	0.8451	0.9223	0.9527	0.9569	0.9621	0.9714	0.9791	0.9814	0.9841	0.9852
	ITSA	**0.7616**	0.8468	0.9234	0.9516	0.9557	0.9619	0.9725	0.9783	0.9802	0.9835	0.9846
	FPA	**0.7616**	**0.8469**	0.9225	0.9493	0.9535	0.9559	0.9685	0.9736	0.9780	0.9812	0.9837
	WOA	**0.7616**	0.8455	**0.9249**	0.9526	0.9557	0.9625	**0.9738**	0.9806	0.9833	0.9852	0.9859
277036	HMPA	**0.8642**	0.8920	0.9388	**0.9549**	0.9601	0.9662	0.9774	**0.9867**	**0.9890**	**0.9904**	**0.9909**

Table 6 (continued)

ID	Algorithm	Threshold level (T) 2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	MPALS	**0.8642**	0.8727	0.9388	**0.9549**	0.9586	0.9649	0.9771	0.9845	0.9889	0.9903	0.9908
	IMPA	**0.8642**	0.8749	0.9388	**0.9549**	0.9583	0.9656	0.9771	0.9867	0.9888	0.9902	0.9906
	EO	**0.8642**	0.8662	0.9388	**0.9549**	**0.9609**	**0.9668**	**0.9785**	**0.9867**	0.9880	0.9895	0.9899
	SMA	**0.8642**	0.8680	0.9391	**0.9549**	0.9591	0.9658	0.9773	0.9855	0.9884	0.9898	0.9903
	HHA	**0.8642**	**0.9004**	0.9389	**0.9549**	0.9594	0.9651	0.9766	0.9866	0.9885	0.9898	0.9904
	ITSA	**0.8642**	0.8688	**0.9402**	0.9546	0.9589	0.9658	0.9752	0.9836	0.9859	0.9885	0.9894
	FPA	**0.8642**	0.8776	0.9399	0.9540	0.9588	0.9636	0.9748	0.9823	0.9855	0.9885	0.9896
	WOA	**0.8642**	0.8962	0.9386	**0.9549**	0.9594	0.9653	0.9765	0.9865	**0.9890**	**0.9904**	0.9908
108070	HMPA	**0.3956**	0.5708	0.6920	0.7794	0.8979	**0.9217**	0.9552	0.9795	**0.9859**	**0.9888**	**0.9899**
	MPALS	0.3955	0.5708	0.6920	0.7649	0.8845	0.9186	0.9562	0.9785	0.9837	0.9880	0.9891
	IMPA	**0.3956**	0.5708	0.6920	0.7648	0.8946	0.9207	0.9575	0.9802	0.9848	0.9883	0.9895
	EO	**0.3956**	0.5708	0.6920	**0.8151**	0.9068	0.9230	**0.9691**	**0.9816**	**0.9859**	**0.9888**	**0.9899**
	SMA	**0.3956**	0.5708	0.6915	0.7804	0.8938	0.9199	0.9588	0.9796	0.9842	0.9868	0.9885
	HHA	**0.3956**	0.5708	0.6883	0.7596	0.8935	0.9178	0.9571	0.9788	0.9833	0.9870	0.9886
	ITSA	0.3955	0.5718	0.6836	0.7759	0.8928	0.9175	0.9537	0.9732	0.9809	0.9851	0.9874
	FPA	**0.3956**	**0.5724**	**0.6933**	0.7922	0.8953	0.9024	0.9486	0.9701	0.9774	0.9839	0.9866
	WOA	**0.3956**	0.5708	0.6868	0.7647	**0.9005**	0.9204	0.9602	0.9787	0.9838	0.9875	0.9890

Bold value shows the best results

Table 7 Performance evaluations using SSIM on all images

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
Barbara	HMPA	**0.8834**	**0.9356**	**0.9574**	0.9710	**0.9778**	0.9818	0.9875	0.9919	**0.9954**	**0.9969**	**0.9974**
	MPALS	**0.8834**	**0.9356**	**0.9574**	0.9710	**0.9778**	0.9818	0.9876	0.9915	0.9947	0.9967	0.9973
	IMPA	**0.8834**	**0.9356**	**0.9574**	0.9710	**0.9778**	**0.9819**	0.9878	0.9924	0.9951	0.9967	0.9973
	EO	**0.8834**	**0.9356**	**0.9574**	0.9710	**0.9778**	**0.9819**	**0.9880**	**0.9932**	0.9952	0.9966	0.9972
	SMA	**0.8834**	**0.9356**	**0.9574**	**0.9711**	**0.9778**	0.9818	0.9877	0.9921	0.9950	0.9965	0.9971
	HHA	**0.8834**	**0.9356**	**0.9574**	**0.9711**	**0.9778**	0.9818	0.9878	0.9912	0.9951	0.9966	0.9971
	ITSA	**0.8834**	**0.9356**	**0.9574**	0.9710	0.9777	0.9816	0.9875	0.9913	0.9939	0.9955	0.9965
	FPA	**0.8834**	**0.9356**	0.9572	0.9702	0.9766	0.9808	0.9858	0.9906	0.9933	0.9956	0.9964
	WOA	**0.8834**	**0.9356**	**0.9574**	0.9710	**0.9778**	0.9818	0.9879	0.9917	0.9953	0.9968	0.9973
Airplane	HMPA	**0.9019**	0.9486	**0.9613**	0.9667	0.9710	0.9798	**0.9885**	**0.9932**	0.9951	0.9966	0.9971
	MPALS	**0.9019**	0.9486	**0.9613**	0.9668	0.9710	**0.9800**	**0.9885**	**0.9932**	**0.9952**	**0.9967**	**0.9972**
	IMPA	**0.9019**	0.9486	**0.9613**	0.9668	0.9708	0.9789	0.9880	0.9926	0.9937	0.9955	0.9968
	EO	**0.9019**	0.9486	**0.9613**	0.9665	0.9710	0.9764	0.9874	0.9919	0.9936	0.9948	0.9957
	SMA	**0.9019**	0.9486	**0.9613**	0.9669	0.9712	0.9793	0.9880	**0.9932**	0.9949	0.9961	0.9966
	HHA	**0.9019**	0.9486	**0.9613**	0.9669	0.9707	0.9793	0.9883	**0.9932**	0.9941	0.9959	0.9964
	ITSA	**0.9019**	0.9488	0.9612	**0.9671**	0.9710	0.9787	0.9876	0.9914	0.9920	0.9942	0.9953
	FPA	**0.9019**	**0.9489**	0.9608	0.9665	0.9691	0.9765	0.9856	0.9898	0.9915	0.9939	0.9949
	WOA	**0.9019**	0.9486	**0.9613**	0.9667	0.9709	0.9799	**0.9885**	0.9932	0.9950	0.9964	0.9970
Mandrill	HMPA	**0.8637**	0.9183	**0.9405**	0.9613	**0.9678**	**0.9752**	0.9838	0.9913	0.9943	**0.9964**	**0.9971**
	MPALS	**0.8637**	0.9182	**0.9405**	0.9613	0.9673	**0.9752**	**0.9839**	0.9913	**0.9944**	**0.9964**	**0.9971**
	IMPA	**0.8637**	0.9182	**0.9405**	0.9613	0.9668	0.9742	0.9832	**0.9917**	**0.9944**	**0.9964**	**0.9971**
	EO	**0.8637**	0.9184	**0.9405**	0.9613	0.9681	0.9739	0.9821	0.9908	0.9935	0.9959	0.9967
	SMA	**0.8637**	0.9189	**0.9405**	**0.9615**	0.9674	0.9747	0.9833	0.9912	0.9938	0.9958	0.9965
	HHA	**0.8637**	0.9182	**0.9405**	0.9614	0.9682	0.9748	0.9825	0.9903	0.9927	0.9948	0.9960
	ITSA	**0.8637**	0.9186	**0.9405**	**0.9615**	0.9678	0.9746	0.9830	0.9894	0.9919	0.9945	0.9954

Table 7 (continued)

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
	FPA	**0.8637**	**0.9191**	0.9398	0.9594	0.9648	0.9702	0.9788	0.9870	0.9894	0.9930	0.9950
	WOA	**0.8637**	0.9184	**0.9405**	0.9613	0.9679	0.9749	**0.9839**	0.9912	0.9931	0.9959	0.9968
Lena	HMPA	**0.8295**	**0.9017**	**0.9307**	0.9400	0.9473	**0.9514**	0.9825	**0.9923**	0.9947	**0.9964**	**0.9970**
	MPALS	**0.8295**	**0.9017**	**0.9307**	0.9400	0.9473	0.9482	0.9826	0.9918	0.9948	**0.9964**	**0.9970**
	IMPA	**0.8295**	**0.9017**	**0.9307**	0.9400	**0.9474**	0.9487	0.9851	**0.9923**	**0.9949**	0.9963	0.9969
	EO	**0.8295**	**0.9017**	0.9300	0.9400	0.9473	0.9560	**0.9853**	**0.9923**	0.9943	0.9961	0.9968
	SMA	**0.8295**	**0.9017**	0.9300	**0.9401**	0.9473	0.9503	0.9801	0.9912	0.9944	0.9959	0.9965
	HHA	**0.8295**	**0.9017**	0.9306	**0.9401**	0.9473	0.9484	0.9822	0.9913	0.9935	0.9954	0.9961
	ITSA	**0.8295**	0.9013	0.9306	0.9395	0.9469	0.9490	0.9786	0.9886	0.9924	0.9940	0.9959
	FPA	**0.8295**	0.9016	0.9292	0.9375	0.9459	0.9504	0.9720	0.9858	0.9905	0.9941	0.9952
	WOA	**0.8295**	0.9016	**0.9307**	**0.9401**	0.9473	0.9472	0.9837	0.9913	0.9940	0.9957	0.9964

Bold value shows the best results

Table 8 Comparison between HMPA and each competing algorithm under the p-value

Threshold level (T)												
ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
277036	EO	NaN	**6.1E−05**	NaN	NaN	**2.1E-02**	**4.4E-02**	**7.3E-04**	**1.3E-08**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	SMA	NaN	**6.1E-05**	**4.2E-02**	8.1E-02	2.9E-01	6.9E-02	**3.0E-02**	**1.2E-06**	**2.4E-10**	**3.3E-11**	**3.0E-11**
	ITSA	NaN	3.6E-01	**4.6E-12**	**1.2E-12**	**2.5E-05**	**1.0E-05**	**2.7E-09**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	HHA	NaN	1.4E-01	**5.7E-09**	**1.9E-09**	**2.9E-03**	**2.1E-03**	**2.8E-03**	**5.5E-10**	**3.7E-11**	**3.0E-11**	**3.0E-11**
	FPA	NaN	2.9E-01	**1.2E-12**	**1.2E-12**	**1.6E-10**	**3.5E-10**	**2.5E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	IMPA	NaN	**3.8E-03**	NaN	NaN	1.3E-01	8.9E-02	3.0E-01	**8.3E-06**	**1.5E-09**	**2.2E-09**	**2.2E-10**
	WOA	NaN	**9.2E-03**	**1.6E-08**	**6.5E-11**	**3.9E-12**	**8.6E-12**	**4.0E-13**	**1.4E-13**	**3.8E-13**	**6.8E-14**	**2.0E-14**
	HMPA	NaN	6.1E-05	NaN	NaN	4.1E-02	8.2E-01	1.8E-01	5.9E-01	3.5E-01	1.5E-01	3.2E-01
108070	EO	NaN	NaN	NaN	2.8E-02	**2.8E-03**	1.6E-02	1.0E-01	**5.5E-11**	**7.4E-11**	**3.0E-11**	**3.0E-11**
	SMA	NaN	NaN	3.3E-01	1.4E-01	**4.7E-06**	**3.9E-03**	**9.3E-04**	**4.2E-09**	**1.4E-09**	**3.0E-11**	**3.0E-11**
	ITSA	NaN	**3.3E-07**	**4.6E-12**	**9.7E-04**	**2.6E-11**	**2.8E-07**	**2.2E-10**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	HHA	NaN	**2.2E-02**	**4.5E-08**	**1.9E-02**	**5.9E-11**	**2.1E-05**	**6.2E-08**	**3.0E-11**	**3.3E-11**	**3.0E-11**	**3.0E-11**
	FPA	NaN	**5.8E-09**	**1.2E-12**	**1.5E-10**	**2.6E-11**	**8.0E-11**	**2.8E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	IMPA	NaN	NaN	NaN	4.4E-01	1.3E-01	2.0E-01	2.2E-01	**3.1E-08**	**2.7E-06**	**8.1E-10**	**3.3E-11**
	WOA	**7.7E-12**	**2.4E-05**	**3.8E-02**	9.7E-01	1.4E-01	**1.5E-03**	**1.0E-05**	**4.7E-08**	**3.2E-11**	**5.9E-14**	**6.7E-17**
	HMPA	NaN	NaN	NaN	1.7E-01	7.7E-01	3.0E-01	7.0E-01	8.5E-01	2.5E-01	1.3E-03	**3.3E-02**
Barbara	EO	NaN	NaN	NaN	3.3E-01	**4.0E-10**	**9.4E-05**	**1.8E-06**	**3.0E-11**	**7.4E-11**	**3.0E-11**	**3.0E-11**
	SMA	NaN	NaN	**1.1E-02**	**2.9E-06**	**1.4E-09**	**1.5E-07**	**3.0E-06**	**3.8E-09**	**2.7E-09**	**3.0E-11**	**3.0E-11**
	ITSA	NaN	**8.6E-07**	**1.9E-10**	**1.7E-12**	**9.0E-12**	**2.5E-11**	**3.1E-10**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	HHA	NaN	**8.1E-02**	**5.5E-03**	**5.9E-08**	**1.4E-09**	**3.0E-08**	**8.0E-07**	**7.2E-11**	**3.7E-11**	**3.0E-11**	**3.0E-11**
	FPA	NaN	**5.7E-11**	**1.2E-12**	**1.7E-12**	**9.0E-12**	**2.5E-11**	**2.9E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	IMPA	NaN	NaN	NaN	3.3E-01	**3.0E-04**	7.8E-01	4.2E-05	**3.4E-09**	**3.1E-08**	**6.1E-11**	**4.5E-11**
	WOA	**3.7E-13**	**4.5E-07**	**2.8E-03**	2.8E-01	4.6E-01	**2.1E-02**	**1.4E-04**	**3.4E-07**	**4.8E-11**	**2.9E-14**	**1.3E-18**
	HMPA	NaN	NaN	NaN	3.3E-01	6.2E-01	1.0E+00	8.1E-01	3.2E-01	**1.4E-02**	**6.5E-04**	**4.1E-05**
Airplane	EO	NaN	NaN	NaN	7.6E-01	**1.2E-03**	**5.0E-11**	**1.2E-10**	**4.3E-08**	**4.5E-11**	**3.0E-11**	**3.0E-11**

Table 8 (continued)

ID	Algorithm	2-T	3-T	4-T	5-T	6-T	7-T	10-T	15-T	20-T	30-T	40-T
Threshold level (T)												
	SMA	NaN	NaN	4.2E-02	3.0E-01	**1.5E-03**	**1.4E-10**	**3.1E-10**	**7.7E-02**	**5.5E-11**	**3.0E-11**	**3.0E-11**
	ITSA	**NaN**	**2.0E-06**	**5.7E-11**	**2.6E-04**	**1.4E-11**	**1.1E-11**	**1.9E-11**	**5.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	HHA	**NaN**	**2.1E-02**	**8.7E-07**	**8.6E-03**	**1.4E-05**	**6.1E-10**	**9.7E-10**	**2.7E-04**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	FPA	**NaN**	**5.8E-09**	**1.2E-12**	**7.3E-05**	**1.4E-11**	**1.1E-11**	**1.9E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	IMPA	NaN	NaN	NaN	**5.5E-03**	**5.8E-03**	**7.6E-06**	**1.9E-06**	7.1E-01	**3.2E-09**	**1.1E-09**	**6.1E-10**
	WOA	**1.2E-19**	**1.4E-10**	**5.5E-07**	**2.2E-03**	2.1E-01	6.0E-01	**3.8E-02**	**1.6E-04**	**7.9E-07**	**1.4E-10**	**1.2E-14**
	HMPA	NaN	NaN	NaN	**5.5E-03**	7.0E-01	7.4E-01	4.1E-01	9.7E-01	2.7E-01	**4.7E-04**	1.2E-01
Mandrill	EO	NaN	**2.5E-02**	NaN	NaN	2.6E-01	**9.0E-12**	**6.3E-10**	**3.0E-11**	**4.1E-11**	**3.0E-11**	**3.0E-11**
	SMA	NaN	**6.2E-03**	8.2E-02	**2.1E-09**	**4.7E-03**	**9.5E-11**	**1.2E-09**	**1.7E-09**	**1.1E-09**	**3.0E-11**	**3.0E-11**
	ITSA	NaN	**4.6E-05**	**1.9E-10**	**1.2E-12**	**1.1E-11**	**1.7E-12**	**2.8E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	HHA	NaN	3.3E-01	8.1E-02	**1.3E-05**	7.4E-01	**1.9E-12**	**2.8E-11**	**3.0E-11**	**9.9E-11**	**3.0E-11**	**3.0E-11**
	FPA	**NaN**	**1.4E-09**	**1.2E-12**	**1.2E-12**	**1.1E-11**	**1.7E-12**	**2.8E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	IMPA	NaN	3.3E-01	NaN	NaN	**2.9E-03**	**6.5E-08**	**1.5E-01**	**3.1E-08**	**8.9E-10**	**4.1E-11**	**3.0E-11**
	WOA	**7.5E-17**	**1.5E-11**	**1.9E-07**	**3.4E-03**	4.6E-01	4.3E-01	**1.5E-02**	**1.2E-05**	**5.0E-09**	**2.7E-13**	**4.3E-18**
	HMPA	NaN	3.3E-01	NaN	NaN	2.4E-03	3.3E-01	8.2E-01	9.6E-01	7.0E-01	**1.9E-04**	**1.2E-05**
Lena	EO	NaN	NaN	1.6E-01	NaN	**6.5E-05**	**1.5E-04**	**6.1E-10**	**3.5E-10**	**9.9E-11**	**3.0E-11**	**3.0E-11**
	SMA	NaN	NaN	**6.5E-05**	**1.9E-06**	**5.4E-06**	**8.3E-02**	**2.1E-08**	**4.4E-06**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	ITSA	NaN	**2.8E-05**	**1.6E-11**	1.2E-12	**1.2E-12**	3.7E-04	2.6E-10	**3.0E-11**	**3.0E-11**	3.0E-11	3.0E-11
	HHA	NaN	**1.3E-03**	**1.3E-07**	**4.6E-08**	**5.7E-11**	**6.3E-01**	**5.0E-10**	**1.2E-09**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	FPA	NaN	**5.5E-11**	**1.2E-12**	**4.6E-12**	**1.2E-12**	**2.3E-10**	**1.9E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**	**3.0E-11**
	IMPA	NaN	NaN	NaN	NaN	1.4E-04	5.3E-01	**1.4E-09**	**6.0E-05**	**2.9E-10**	**1.5E-10**	**6.1E-10**
	WOA	**1.9E-16**	**4.0E-11**	**8.8E-08**	**3.2E-04**	7.1E-02	9.9E-01	1.2E-01	**9.6E-04**	**9.8E-07**	**1.8E-10**	**3.6E-15**
	HMPA	**NaN**	NaN	NaN	NaN	NaN	1.0E-01	**3.0E-02**	**6.5E-02**	1.6E-01	**4.7E-04**	**5.6E-04**

Bold value shows the best results

of the competing algorithms and similar to the others. Note that, NaN value in this table indicates that the outcomes of each pair of algorithms are the same.

7 Conclusion and future work

In this paper, a new multilevel thresholding image segmentation algorithm based on the improved marine predators algorithm (MPA) is developed, this algorithm is improved with a novel strategy called the linearly increased worst solutions improvement strategy (LIS) to accelerate the convergence by steering gradually a number np of the worst solutions in the right direction of the best-so-far solution, and randomly within the search space of the problem to help in avoiding stuck into local minima. this LIS is integrated with the standard MPA to propose a new variant, namely MPALS, for the ISP. Further, another strategy called ranking-based updating strategy (RUS) has been here proposed and employed to strengthen the exploration and exploitation capability of MPALS for reaching other regions which are intractable by MPALS. This improved MPALS was abbreviated as HMPA. The two proposed variants: HMPA and MPALS have been validated on 14 test images and compared with seven state-of-the-arts meta-heuristic algorithms. The experimental results show the superiority of HMPA for all performance metrics: PSNR, F-value, SSIM, and SD except CPU time as its main limitation addressed in future work. Broadly speaking, the HMPA could occupy the first rank because it was competitive with some algorithms for the small threshold levels and superior for the high threshold levels; meanwhile, MPALS, WOA, IMPA, EO, SMA, HHA, ITSA, and FPA are respectively ranked from the second-best one to the worst for the fitness values and PSNR. Unfortunately, MPALS come in the seventh rank in term of the SSIM which were considered as its main limitation addressed using the RUS in the second variant: HMPA. Our future work includes investigating the performance of the marine predator algorithm when tackling DNA fragment assembly problems, feature selection problems, and flow shop scheduling problems, in addition to tackling the CPU time problem related to the proposed algorithms.

Funding This research has no funding source.

Declarations

Conflict of interest The authors declare that there is no conflict of interest about the research.

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

References

Abd El Aziz M, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Syst Appl 83:242–256

Abdel-Basset M, Manogaran G, El-Shahat D, Mirjalili S (2018) A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem. Futur Gener Comput Syst 85:129–145

Abdel-Basset M, Mohamed R, Elhoseny M, Chakrabortty RK, Ryan M (2020) A hybrid Covid-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy. IEEE Access 8:79521–79540
Abdel-Basset M, Mohamed R, Mirjalili S, Chakrabortty RK, Ryan MJ (2020) Solar photovoltaic parameter estimation using an improved equilibrium optimizer. Sol Energy 209:694–708
Abdel-Basset M, Chang V, Mohamed R (2020) A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. Neural Comput Appl 1–34
Abdel-Basset M, Mohamed R, Mirjalili S, Chakrabortty RK, Ryan M (2021) An efficient marine predators algorithm for solving multi-objective optimization problems: analysis and validations. IEEE Access 9:42817–42844
Abouhawwash M, Alessio AM (2021) Multi-objective evolutionary algorithm for pet image reconstruction: Concept. IEEE Trans Med Imaging 40(8):2142–2151
Agrawal S, Panda R, Bhuyan S, Panigrahi BK (2013) Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. Swarm Evol Comput 11:16–30
Aksac A, Ozyer T, Alhajj R (2017) Complex networks driven salient region detection based on superpixel segmentation. Pattern Recognit 66:268–279
Arora S, Acharya J, Verma A, Panigrahi PK (2008) Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. Pattern Recognit Lett 29(2):119–125
Bao X, Jia H, Lang C (2019) A novel hybrid Harris Hawks optimization for color image multilevel thresholding segmentation. IEEE Access 7:76529–76546
Barman R, Ehrmann M, Clematide S, Oliveira SA, Kaplan F (2020) Combining visual and textual features for semantic segmentation of historical newspapers. arXiv preprint arXiv:2002.06144
Bhandari AK, Kumar A, Singh GK (2015) Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. Expert Syst Appl 42(3):1573–1601
Chakraborty F, Nandi D, Roy PK (2019) Oppositional symbiotic organisms search optimization for multilevel thresholding of color image. Appl Soft Comput 82:105577
Chen K, Zhou Y, Zhang Z, Dai M, Chao Y, Shi J (2016) Multilevel image segmentation based on an improved firefly algorithm. Math Probl Eng 2016
Chouksey M, Jha RK, Sharma R (2020) A fast technique for image segmentation based on two meta-heuristic algorithms. Multimedia Tools Appl 1–53
Cuevas E, Fausto F, González A (2020) Locust search algorithm applied to multi-threshold segmentation. In: New advancements in swarm algorithms: operators and applications. Springer, pp 211–240
Di Martino F, Sessa S (2020) PSO image thresholding on images compressed via fuzzy transforms. Inf Sci 506:308–324
Diab AAZ, Tolba MA, El-Magd AGA, Zaky MM, El-Rifaie AM (2020) Fuel cell parameters estimation via marine predators and political optimizers. IEEE Access 8:166998–167018
Durmus A (2021) The concentric elliptical antenna array patterns synthesis using marine predators algorithm. Arab J Sci Eng 1–11
Elsayed SM, Sarker RA, Essam DL (2014) A new genetic algorithm for solving optimization problems. Eng Appl Artif Intell 27:57–69
Elsayed AM, Shaheen AM, Alharthi MM, Ghoneim SS, El-Sehiemy RA (2021) Adequate operation of hybrid AC/MT-HVDC power systems using an improved multi-objective marine predators optimizer. IEEE Access 9:51065–51087
Erdmann H, Wachs-Lopes G, Gallao C, Ribeiro M, Rodrigues P (2015) A study of a firefly meta-heuristics for multithreshold image segmentation. In: Developments in medical image processing and computational vision. Springer, pp 279–295
Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377
Ghoneimy M, Hassan HA, Nabil E (2021) A new hybrid clustering method of binary differential evolution and marine predators algorithm for multi-omics datasets
Guo C, Li H (2007) Multilevel thresholding method for image segmentation based on an adaptive particle swarm optimization algorithm. In: Australasian joint conference on artificial intelligence. Springer, pp 654–658
Han J, Yang C, Zhou X, Gui W (2017) A new multi-threshold image segmentation approach using state transition algorithm. Appl Math Model 44:588–601
Hassanzadeh T, Essam D, Sarker R (2020) An evolutionary denseres deep convolutional neural network for medical image segmentation. IEEE Access, vol. 8, pp 212 298–212 314
Hore A, Ziou D (2010) Image quality metrics: PSNR vs. SSIM. In: 20th international conference on pattern recognition. IEEE 2010:2366–2369

Horng M-H (2010) Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. Expert Syst Appl 37(6):4580–4592
Houssein EH, Helmy BE-D, Elngar AA, Abdelminaam DS, Shaban H (2021) An improved tunicate swarm algorithm for global optimization and image segmentation. IEEE Access 9:56066–56092
Huo F, Sun X, Ren W (2020) Multilevel image threshold segmentation using an improved bloch quantum artificial bee colony algorithm. Multimedia Tools Appl 79(3):2447–2471
Kandhway P, Bhandari AK (2019) Spatial context cross entropy function based multilevel image segmentation using multi-verse optimizer. Multimedia Tools Appl 78(16):22613–22641
Kapur JN, Sahoo PK, Wong AK (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vis Graph Image Process 29(3):273–285
Karydas CG (2020) Optimization of multi-scale segmentation of satellite imagery using fractal geometry. Int J Remote Sens 41(8):2905–2933
Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput
Kuruvilla J, Sukumaran D, Sankar A, Joy SP (2016) A review on image processing and image segmentation. In: International conference on data mining and advanced computing (SAPIENCE). IEEE 2016:198–203
Lam F, Longnecker M (1983) A modified wilcoxon rank sum test for paired data. Biometrika 70(2):510–513
Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. Futur Gener Comput Syst 111:300–323
Li W, Lin Q, Wang K, Cai K (2021) Improving medical image fusion method using fuzzy entropy and nonsubsampling contourlet transform. Int J Imaging Syst Technol 31(1):204–214
Liu X, Yang D (2021) Color constancy computation for dyed fabrics via improved marine predators algorithm optimized random vector functional-link network. Color Res Appl
Liu Y, Mu C, Kou W, Liu J (2015) Modified particle swarm optimization-based multilevel thresholding for image segmentation. Soft Comput 19(5):1311–1327
Ma L, Cheng S, Shi Y (2020) Enhancing learning efficiency of brain storm optimization via orthogonal learning design. IEEE Trans Syst Man Cybern Syst. https://doi.org/10.1109/TSMC.2020.2963943
Ma L, Huang M, Yang S, Wang R, Wang X (2021) An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2020.3041212
Mahajan S, Mittal N, Pandit AK (2021) Image segmentation using multilevel thresholding based on type ii fuzzy entropy and marine predators algorithm. Multimedia Tools Appl 80(13):19335–19359
Maitra M, Chatterjee A (2008) A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. Expert Syst Appl 34(2):1341–1350
Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
Mittal M, Arora M, Pandey T, Goyal LM (2020) Image segmentation using deep learning techniques in medical images. In: Advancement of machine intelligence in interactive medical image analysis. Springer, pp 41–63
Mokhtari SY, Kimour MT (2019) A novel improved bat algorithm based image multi-thresholding. Int J Electr Eng Inf 11(2)
Naji Alwerfali HS, Al-qaness MA, Abd Elaziz M, Ewees AA, Oliva D, Lu S (2020) Multi-level image thresholding based on modified spherical search optimizer and fuzzy entropy. Entropy 22(3):328
Naoum A, Nothman J, Curran J (2019) Article segmentation in digitised newspapers with a 2d markov model. In: 2019 international conference on document analysis and recognition (ICDAR). IEEE, pp 1007–1014
Narayanan BN, Hardie RC, Kebede TM, Sprague MJ (2019) Optimized feature selection-based clustering approach for computer-aided detection of lung nodules in different modalities. Pattern Anal Appl 22(2):559–571
Oliva D, Cuevas E, Pajares G, Zaldivar D, Osuna V (2014) A multilevel thresholding algorithm using electromagnetism optimization. Neurocomputing 139:357–381
Oliva D, Hinojosa S, Cuevas E, Pajares G, Avalos O, Gálvez J (2017) Cross entropy based thresholding for magnetic resonance brain images using crow search algorithm. Expert Syst Appl 79:164–180
Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
Panagant N, Yıldız M, Pholdee N, Yıldız AR, Bureerat S, Sait SM (2021) A novel hybrid marine predators-Nelder-Mead optimization algorithm for the optimal design of engineering problems. Mater Test 63(5):453–457
Prathusha P, Jyothi S (2018) A novel edge detection algorithm for fast and efficient image segmentation. In: Data engineering and intelligent computing. Springer, pp 283–291

Ramezani M, Bahmanyar D, Razmjooy N (2021) A new improved model of marine predator algorithm for optimization problems. Arab J Sci Eng 1–24
Riad N, Anis W, Elkassas A, Hassan AE-W (2021) Three-phase multilevel inverter using selective harmonic elimination with marine predator algorithm. Electronics 10(4):374
Ridha HM (2020) Parameters extraction of single and double diodes photovoltaic models using marine predators algorithm and lambert W function. Sol Energy 209:674–693
Sanyal N, Chatterjee A, Munshi S (2011) An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation. Expert Syst Appl 38(12):15489–15498
Saremi S, Mirjalili SZ, Mirjalili SM (2015) Evolutionary population dynamics and grey wolf optimizer. Neural Comput Appl 26(5):1257–1263
Sayed GI, Hassanien AE, Azar AT (2019) Feature selection via a novel chaotic crow search algorithm. Neural Comput Appl 31(1):171–188
Shahabi F, Pourahangarian F, Beheshti H (2019) A multilevel image thresholding approach based on crow search algorithm and Otsu method
Shaheen MA, Yousri D, Fathy A, Hasanien HM, Alkuhayli A, Muyeen S (2020) A novel application of improved marine predators algorithm and particle swarm optimization for solving the ORPD problem. Energies 13(21):5679
Shaheen AM, Elsayed AM, El-Sehiemy RA, Kamel S, Ghoneim SS (2021) A modified marine predators optimization algorithm for simultaneous network reconfiguration and distributed generator allocation in distribution systems under different loading conditions. Eng Optim 1–22
Soliman MA, Hasanien HM, Alkuhayli A (2020) Marine predators algorithm for parameters identification of triple-diode photovoltaic models. IEEE Access 8:155832–155842
Sultana F, Sufian A, Dutta P (2020) Evolution of image segmentation using deep convolutional neural network: a survey. Knowl-Based Syst 201:106062
Swief RA, Hassan NM, Hasanien HM, Abdelaziz AY, Kamh MZ (2021) Multi-regional optimal power flow using marine predators algorithm considering load and generation variability. IEEE Access
Tang K, Xiao X, Wu J, Yang J, Luo L (2017) An improved multilevel thresholding approach based modified bacterial foraging optimization. Appl Intell 46(1):214–226
University of California. www2.eecs.berkeley.edu/research/projects/cs/vision/grouping/resources.html
Wang R, Zhou Y, Zhao C, Wu H (2015) A hybrid flower pollination algorithm based modified randomized location for multi-threshold medical image segmentation. Bio-Med Mater Eng 26(s1):S1345–S1351
Wang X, Wang X, Wilkes DM (2020) An efficient image segmentation algorithm for object recognition using spectral clustering. In: Machine learning-based natural scene recognition for mobile robot localization in an unknown environment. Springer, pp 215–234
Wang Z, Wang Q, Zhang Z, Razmjooy N (2021) A new configuration of autonomous CHP system based on improved version of marine predators algorithm: a case study. Int Trans Electr Energy Syst 31(4):e12806
Xiong L, Tang G, Chen Y-C, Hu Y-X, Chen R-S (2020) Color disease spot image segmentation algorithm based on chaotic particle swarm optimization and FCM. J Supercomput 1–15
Xu L, Jia H, Lang C, Peng X, Sun K (2019) A novel method for multilevel color image segmentation based on dragonfly algorithm and differential evolution. IEEE Access 7:19502–19538
Yan Z, Zhang J, Tang J (2020) Modified water wave optimization algorithm for underwater multilevel thresholding image segmentation. Multimedia Tools Appl 1–34
Yao X, Li Z, Liu L, Cheng X (2019) Multi-threshold image segmentation based on improved grey wolf optimization algorithm. In: IOP conference series: earth and environmental science, vol. 252, no. 4. IOP Publishing, p 042105
Yang X-S (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation. Springer, pp 240–249
Yu G, Meng Z, Ma H, Liu L (2021) An adaptive marine predators algorithm for optimizing a hybrid PV/DG/battery system for a remote area in China. Energy Rep 7:398–412
Zhang Z, Wu C, Coleman S, Kerr D (2020) Dense-inception u-net for medical image segmentation. Comput Methods Programs Biomed 192:105395

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

Authors and Affiliations

Mohamed Abdel-Basset[1] · Reda Mohamed[1] · Mohamed Abouhawwash[2,3]

Mohamed Abdel-Basset
mohamedbasset@ieee.org

Reda Mohamed
redamoh@zu.edu.eg

[1] Zagazig Univesitry, Shaibet an Nakareyah, Zagazig 2, Ash Sharqia Governorate 44519, Egypt

[2] Department of Mathematics Faculty of Science, Mansoura University, Mansoura 35516, Egypt

[3] Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI 48824, USA