



# Kahramanmaraş Sütçü İmam University Journal of Engineering Sciences



Geliş Tarihi : 03.01.2024  
Kabul Tarihi : 28.03.2024

Received Date : 03.01.2024  
Accepted Date : 28.03.2024

## MULTILEVEL THRESHOLDING FOR BRAIN MR IMAGE SEGMENTATION USING SWARM-BASED OPTIMIZATION ALGORITHMS

### SÜRÜ ZEKASI TEMELLİ OPTİMİZASYON ALGORİTMALARI KULLANILARAK ÇOK SEVİYELİ EŞİKLEME İLE BEYİN MR GÖRÜNTÜLERİNİN BÖLÜTLENMESİ

Ahmet Nusret TOPRAK<sup>1\*</sup> (ORCID: 0000-0003-4841-9508)

Ömür ŞAHİN<sup>1</sup> (ORCID: 0000-0003-1213-7445)

Rifat KURBAN<sup>2</sup> (ORCID: 0000-0002-0277-2210)

<sup>1</sup> Erciyes Üniversitesi, Bilgisayar Mühendisliği Bölümü, Kayseri, Türkiye

<sup>2</sup> Abdullah Gül Üniversitesi, Bilgisayar Mühendisliği Bölümü, Kayseri, Türkiye

\*Sorumlu Yazar / Corresponding Author: Ahmet Nusret TOPRAK, antoprak@erciyes.edu.tr

#### ABSTRACT

Image segmentation, the process of dividing an image into various sets of pixels called segments, is an essential technique in image processing. Image segmentation reduces the complexity of the image and makes it easier to analyze by dividing the image into segments. One of the simplest yet powerful ways of image segmentation is multilevel thresholding, in which pixels are segmented into multiple regions according to their intensities. This study aims to explore and compare the performance of the well-known swarm-based optimization algorithms on the multilevel thresholding-based image segmentation task using brain MR images. Seven swarm-based optimization algorithms: Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Gray Wolf Optimizer (GWO), Moth-Flame Optimization (MFO), Ant Lion Optimization (ALO), Whale Optimization (WOA), and Jellyfish Search Optimizer (JS) algorithms are compared by applying to brain MR images to determine threshold levels. In the experiments carried out with mentioned algorithms, minimum cross-entropy, and between-class variance objective functions were employed. Extensive experiments show that JS, ABC, and PSO algorithms outperform others.

**Keywords:** Image segmentation, multilevel thresholding, swarm-based optimization, minimum cross-entropy, between-class variance.

#### ÖZET

Bir görüntüyü bölüt adı verilen çeşitli piksel kümelerine ayırma işlemi olan görüntü bölütleme, görüntü işlemede önemli bir tekniktir. Görüntü bölütleme, görüntünün karmaşıklığını azaltmakta ve görüntüyü bölütlere ayırarak analiz edilmesini kolaylaştırmaktadır. Görüntü bölütlemenin en basit ancak etkin yollarından biri, piksellerin değerlerine göre birden çok bölgeye ayrıldığı çok düzeyli eşiklemedir. Bu çalışma, yaygın kullanılan sürü tabanlı optimizasyon algoritmalarının beyin MR görüntülerinde çok düzeyli eşikleme tabanlı görüntü bölütleme performansını araştırmayı ve karşılaştırmayı amaçlamaktadır. Yedi sürü zekâsı temelli optimizasyon algoritması: Parçacık Sürü Optimizasyonu (PSO), Yapay Arı Kolonisi (ABC), Gri Kurt Optimize Edici (GWO), Güve Alevi Optimizasyonu (MFO), Karınca Aslanı Optimizasyonu (ALO), Balina Optimizasyonu (WOA) ve Denizanası Arama Optimizasyon (JS) eşik seviyelerini belirlemek üzere beyin MR görüntülerine uygulanarak karşılaştırılmaktadır. Bahsi geçen algoritmalar ile yapılan deneylerde minimum çapraz entropi ve sınıflar arası varyans amaç fonksiyonları kullanılmıştır. Kapsamlı deneyler, JS, ABC ve PSO algoritmalarının daha iyi performans sergilediğini göstermektedir.

**Anahtar Kelimeler:** Görüntü bölütleme, çok seviyeli eşikleme, sürü zekâsı tabanlı optimizasyon, minimum çapraz entropi, sınıflar arası varyans.

## INTRODUCTION

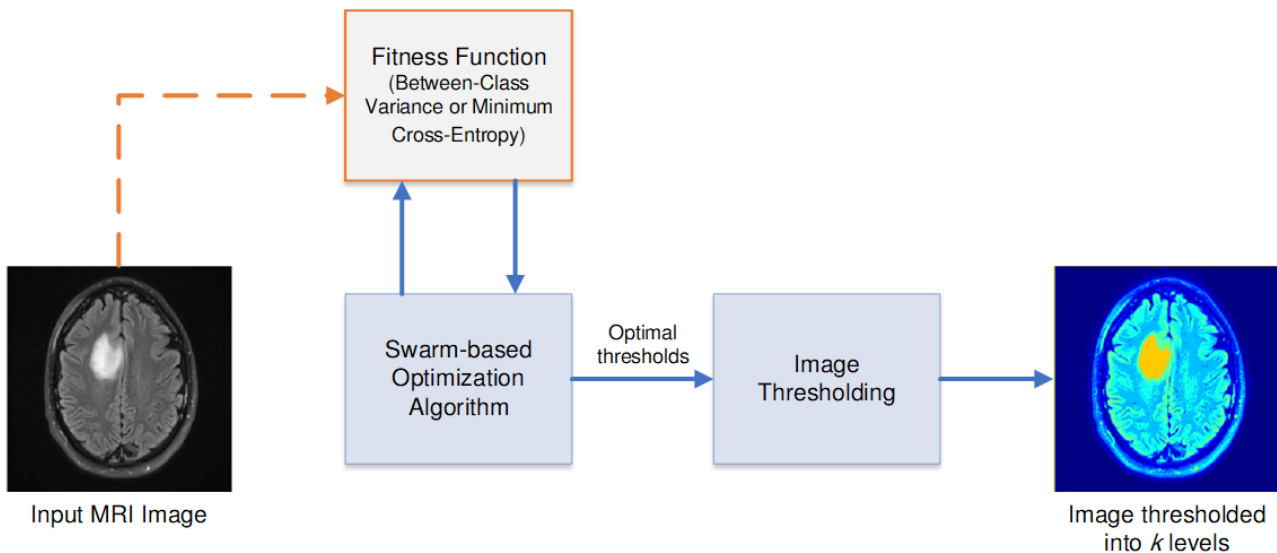
Image segmentation is the process of dividing an image into various sets of pixels called segments based on the characteristics of the pixels. In other words, the image segmentation task can be defined as assigning a label to each pixel in the image so that pixels with certain common characteristics share the same label (Kurban et al., 2021). By dividing an image into segments, the complexity of the image is reduced, and analyzing it becomes easier. Therefore, image segmentation plays an essential role in many areas such as medical diagnosis, object detection, object recognition, remote sensing, and video surveillance (Farshi & Ardabili, 2021).

The proper segmentation of an image is a fundamental and challenging task in the areas mentioned above. In recent years, a considerable amount of literature has been published on image segmentation. These studies can be broadly divided into five categories: thresholding-based methods, clustering-based methods, region-based methods, edge-based methods, and artificial neural network-based methods (Sezgin & Sankur, 2004). Clustering-based image segmentation methods aim to group pixels with similar features like intensity, color, or texture into clusters together and represent them as a single entity. Similarly, region-based segmentation methods aim to obtain segments by dividing the image into sets of pixels with similar properties. These methods are classified into two groups region growing and region splitting. Region-growing methods start with selecting an arbitrary seed pixel in the image and then comparing it with neighboring pixels. Next, similar neighboring pixels are added to the current segment until it cannot grow further. Unlike region-growing methods, region-splitting methods consider the whole image as the region of interest. Then, the image is iteratively divided into regions that share similar characteristics. Edge-based image segmentation techniques aim to detect edges that represent the boundaries of the regions in an image. Edge detection is the process of detecting changes or discontinuities in intensity in an image. Region boundaries and edges in an image are closely related, as region boundaries often have a sharp change in intensity.

One of the simplest yet powerful ways to segment images is thresholding, in which pixels are segmented based on their intensity values. Thresholding methods are divided into two groups based on the number of threshold values: finding a single threshold value (bi-level) or multiple thresholds (multilevel). Bi-level thresholding separates an image into basically two parts; the foreground and background, while multilevel thresholding separates an image into multiple parts according to their intensities (Karakoyun, 2023).

Both bi-level and multilevel thresholding techniques are classified into parametric and nonparametric approaches (Hammouche et al., 2010). Parametric techniques assume that the gray-level distribution of each class is based on a particular distribution, such as a normal distribution. The distribution parameters that best fit the given histogram data are estimated using the least-squares method. Thus, it becomes a nonlinear optimization problem whose solution is more time-consuming and computationally expensive. On the other hand, nonparametric techniques determine the thresholds that optimally divide an image into regions in terms of some statistical criterion or measure of entropy. Compared to parametric methods, nonparametric techniques are more computationally efficient and easier to implement. One of the most well-known statistical criteria-based thresholding algorithms is Otsu's method (Otsu, 1979), in which the threshold is obtained by minimizing within-class variance, or equivalently, by maximizing between-class variance. Some of the most recognized entropy-based thresholding algorithms include Kapur Entropy (Kapur et al., 1985), Tsallis Entropy (Albuquerque et al., 2004), Renyi Entropy (Sahoo et al., 1997), determine the optimal threshold value by maximizing the entropy between classes.

Investigating the optimal threshold values is a challenging task within multilevel thresholding-based segmentation. Since the classical methods rely on extensive search, they can only be effective for two-level thresholding and become computationally costly for multilevel thresholding. However, as powerful search algorithms, metaheuristic optimization algorithms are widely used for multilevel thresholding-based image segmentation applications and offer successful performance (Dhal et al., 2020). For example, the Genetic Algorithm (GA) and modified versions have been applied many times to the multilevel thresholding problem (Hammouche et al., 2008). In addition, Particle Swarm Optimization (PSO) and its variants like fractional-order Darwinian PSO (Ghamisi et al., 2014) and modified PSO (Liu et al., 2015) were also used in the multilevel thresholding field. Cuevas et al. applied the Differential Evolution algorithm to the multilevel thresholding-based segmentation field (Cuevas et al., 2010). The Artificial Bee Colony (ABC) algorithm (Akay, 2013) and its modified versions (Gao et al., 2018) have also been employed and obtained notable performance.



**Figure 1.** The Schematic Diagram of The Brain MRI Segmentation Scheme Using Swarm-Based Optimization Algorithms.

Bakhshali and Shamsi proposed a Bacterial Foraging Optimization (BFO) algorithm-based multilevel thresholding method to segment color lip images (Bakhshali & Shamsi, 2014). Brajevic and Tuba applied Cuckoo Search (CS) and Firefly Algorithm (FA) algorithms to the multilevel thresholding-based image segmentation problem in another work (Brajevic & Tuba, 2014). Ye et al. used a Bat algorithm and fuzzy entropy-based multilevel thresholding method to segment gray-level and infrared images (Ye et al., 2015). El Aziz et al. introduced a model that makes use of the Whale optimization algorithm and Moth-flame optimization algorithms for multilevel thresholding (Aziz et al., 2017). Furthermore, Harmony Search (Oliva et al., 2013), Elephant Herding (Tuba et al., 2017), and Harris Hawk (Rodríguez-Esparza et al., 2020) algorithms have also been used for solving the multilevel image thresholding problem. In another study, Sharma et al. utilized an improved firefly algorithm in multi-level thresholding-based image segmentation (Sharma et al., 2022). Gharehchopogh and Ibrkci proposed a method that uses an African vultures optimization algorithm for a multi-level thresholding determination problem (Gharehchopogh & Ibrkci, 2024). Finally, Guo et al. employed Salp Swarm algorithm for multi-level thresholding on breast pathology images (Guo et al., 2024). In an early work, Kurban et al. conducted a comprehensive study on multilevel thresholding-based color image segmentation using several swarm-based and evolutionary-based optimization algorithms (Kurban et al., 2014). The results of the study revealed that swarm-based algorithms are more accurate and robust than evolutionary algorithms.

In recent decades, there has been an increasing amount of literature on multilevel thresholding-based medical image segmentation. Most of them have been carried out on magnetic resonance (MR) images. For example, Manikandan et al. proposed a Real code Genetic Algorithm based multilevel thresholding approach using Kapur entropy to segment MR brain images (Manikandan et al., 2014). In another study, Kaur et al. presented a multilevel thresholding-based brain tumor MR segmentation method making use of the PSO algorithm and 2D local cross-entropy as an objective criterion (Kaur et al., 2016). Oliva et al. proposed a multilevel thresholding method based on Crow Search Algorithm and Minimum Cross entropy to segment brain MR images (Oliva et al., 2017). Another multilevel thresholding study on brain MR images based on the Adaptive Wind Driven Optimization algorithm was presented by Kotte et al. (Kotte et al., 2018). Tarkhaneh and Shen developed an algorithm called Adaptive Differential Evolution with Lévy Distribution (ALDE) to segment brain MR images and achieve better results than the DE algorithm (Tarkhaneh & Shen, 2019). Jena et al. proposed an approach based on the Attacking Manta-Ray Foraging Optimization algorithm and maximum 3D Tsallis entropy for multilevel thresholding of brain MR images (Jena et al., 2021).

This study aims to explore and compare the performance of the well-known swarm intelligence-based optimization algorithms on the multilevel thresholding-based image segmentation task using a brain MR image dataset. Swarm intelligence-based optimization algorithms are typically composed of individuals interacting with each other and with their environment. They are inspired by the collective behavior of self-organized, decentralized systems (Aslan et al., 2023). The compared swarm-based optimization algorithms include PSO algorithm (Kennedy & Eberhart, 1995),

ABC algorithm (Karaboga & Basturk, 2007), Gray Wolf Optimizer (GWO) algorithm (Mirjalili et al., 2014), Moth-Flame Optimization (MFO) algorithm (Mirjalili, 2015a), Ant Lion Optimization (ALO) algorithm (Mirjalili, 2015b), Whale Optimization algorithm (WOA) (Mirjalili & Lewis, 2016), and Jellyfish Search (JS) Optimizer algorithm (Chou & Truong, 2021).

The contribution of this paper can be summarized as follows:

- We provide a comparison of swarm-based optimization algorithms for the multilevel thresholding problem.
- We compare swarm-based optimization algorithms by using statistical measures: the Mann-Whitney U test and Vargha–Delaney effect size on brain MR image segmentation.

This paper has been divided into five sections, including this introductory section. The next section formulates the multilevel thresholding problem and introduces the minimum cross-entropy and the between-class variance functions. A brief overview of the swarm-based optimization algorithms is provided in the third section. The fourth section gives experimental results and comparisons over a set of brain MR images. Finally, the conclusion section provides a summary and critique of the findings.

## MULTILEVEL IMAGE THRESHOLDING

### *Problem Definition*

Thresholding is dividing an image  $I$  into segments based on the value of a threshold  $t$ . Bi-level thresholding divides an image into two segments. Accordingly, the pixels with a gray level greater than a particular threshold value ( $t$ ) are classified as foreground, and those with less than the threshold are classified as background. Thus, based on the threshold value, bi-level thresholding converts a gray image into a binary image (Hammouche et al., 2010). Let the  $L$  be the number of gray levels; the following equation can express bi-level thresholding:

$$S(x, y) = \begin{cases} 0, & 0 \leq I(x, y) < t \\ 1, & t \leq I(x, y) \leq L \end{cases} \quad (1)$$

where  $x, y$  are the pixel coordinates and  $S$  is the output image. Multilevel thresholding divides an image into multiple segments according to their gray levels by using more than one threshold value as given in the following equation:

$$S(x, y) = \begin{cases} 0, & 0 \leq I(x, y) < t_1 \\ 1, & t_1 \leq I(x, y) < t_2 \\ \vdots & \\ n, & t_n \leq I(x, y) \leq L \end{cases} \quad (2)$$

where  $t_i$  is the  $i$ th threshold value and  $n$  is the number of thresholds.

Calculating a single threshold value for bi-level thresholding can be done computationally efficiently. However, calculating multiple threshold values for multilevel thresholding requires high computational efforts. Notably, the problem becomes more challenging as the number of thresholds increases. The nonparametric techniques determine the thresholds that optimally divide an image into regions in terms of a particular statistical criterion or measure of entropy (Hammouche et al., 2010).

The use of Between-Class Variance (Otsu, 1979) and Minimum Cross-Entropy (Li & Lee, 1993) as two different objective functions in multi-threshold imaging problems is widespread in the literature. Between-Class Variance aims for a clear separation between classes within the image, while Minimum Cross-Entropy attempts to maximize the similarity of the thresholded image to the original. The choice of these methods depends on the requirements and priorities of the application. For example, in medical imaging, where preserving essential anatomical structures is critical, the Minimum Cross-Entropy method may be preferred; however, for a distinct separation of specific features (such as tumors), the Between-Class Variance method is more suitable. Therefore, two of the literature's most widely used functions, minimum cross-entropy and between-class variance, are employed in this study.

### *Minimum Cross-Entropy*

Entropy-based thresholding techniques offer an effective way of segmenting images using the probability distribution of the image histogram. The minimum cross-entropy method is one of the well-known automatic thresholding techniques that select the optimum thresholds based on the cross-entropy of segmented classes (Li & Lee, 1993).

Let  $P = \{p_1, p_2, \dots, p_n\}$  and  $Q = \{q_1, q_2, \dots, q_n\}$  be two probability distributions, the cross entropy measures the distance between two distributions as follows (Li & Lee, 1993):

$$D(P, Q) = \sum_{i=1}^n p_i \log \frac{p_i}{q_i} \quad (3)$$

The minimum cross-entropy method selects the optimum threshold by minimizing the cross-entropy between the original image and the thresholded image. Let  $I$  be the original image, and then the thresholded image  $S$  is obtained as follows:

$$S(x, y) = \begin{cases} \mu_0, & 0 \leq I(x, y) < t \\ \mu_1, & t \leq I(x, y) \leq L \end{cases} \quad (4)$$

where  $t$  is threshold value,  $L$  is the maximum possible pixel value of the  $I$  and  $\mu_0$  and  $\mu_1$  are:

$$\mu_0 = \frac{\sum_{i=1}^{t-1} ih_i}{\sum_{i=1}^{t-1} h_i}, \quad \mu_1 = \frac{\sum_{i=t}^L ih_i}{\sum_{i=t}^L h_i} \quad (5)$$

The cross-entropy is then calculated by the following equation:

$$\eta(t) = \sum_{i=1}^{t-1} ih_i \log \left( \frac{i}{\mu_0(t)} \right) + \sum_{i=t}^L ih_i \log \left( \frac{i}{\mu_1(t)} \right) \quad (6)$$

where  $h_i$  is the corresponding histogram. The optimal threshold can be obtained by minimizing the cross-entropy:

$$t_{opt} = \arg \min(\eta(t)) \quad (7)$$

### **Between-Class Variance**

Another nonparametric thresholding-based segmentation approach is the between-class variance-based technique. The main idea here is to divide an image histogram into two classes with a threshold obtained by minimizing within-class variance or equivalently by maximizing between-class variance. Otsu defined the between-class variance as given in the following equation (Otsu, 1979):

$$\sigma_0 = \omega_0(\mu_0 - \mu_t)^2, \mu_0 = \sum_{i=0}^{t-1} \frac{ip_i}{\omega_0}, \sigma_1 = \omega_1(\mu_1 - \mu_t)^2, \mu_1 = \sum_{i=t}^{L-1} \frac{ip_i}{\omega_1} \quad (8)$$

The optimal threshold can be computed simply by:

$$t_{opt} = \arg \max(\sigma_0 + \sigma_1) \quad (9)$$

Between-class variance-based technique can be extended for multilevel thresholding as given by the following equation:

$$\begin{aligned} \sigma_0 &= \omega_0(\mu_0 - \mu_t)^2, & \mu_0 &= \sum_{i=0}^{t-1} \frac{ip_i}{\omega_0} \\ \sigma_j &= \omega_j(\mu_j - \mu_t)^2, & \mu_j &= \sum_{i=t_j}^{t_{j+1}-1} \frac{ip_i}{\omega_j} \\ \sigma_n &= \omega_n(\mu_n - \mu_t)^2, & \mu_n &= \sum_{i=t_n}^{L-1} \frac{ip_i}{\omega_n} \end{aligned} \quad (10)$$

The optimal thresholds can then be computed by maximizing the following objective function:

$$\vec{t}_{opt} = \arg \max(\sum_{i=0}^n \sigma_i) \quad (11)$$

## **SWARM-BASED OPTIMIZATION TECHNIQUES**

### **Particle Swarm Optimization**

PSO is a population-based meta-heuristic algorithm that was proposed by Kennedy and Eberhart in 1995 to solve continuous and discrete optimization problems (Kennedy & Eberhart, 1995). To find the global optimum, it models the collective behavior of bird flocking. Each solution in the PSO algorithm is represented by agents called particles. Each particle searches for better positions in the search space of the optimization problem by changing its velocity regarding its individual experience and social experience of the swarm.

The PSO algorithm starts by generating random positions of the particles. The velocity of each particle is calculated by considering the current position of the particle, the best position ever found by itself, and the best position ever found by any particle in the neighborhood of the particle as given in the following equation:

$$\vec{v}(t+1) = \omega \vec{v}(t) + \phi_1 \text{rand}(0,1)(\vec{p}(t) - \vec{x}(t)) + \phi_2 \text{rand}(0,1)(\vec{g}(t) - \vec{x}(t)) \quad (12)$$

The velocities and positions of the particles are iteratively updated until a stopping criterion is met.

### **Artificial Bee Colony**

The ABC is a population-based meta-heuristic algorithm that was proposed by Karaboga in 2005 to solve numerical problems (Karaboga, 2005). It is inspired by the intelligent foraging behavior of honeybee colonies. In the ABC algorithm, honeybee colonies are separated into three types of bees: employed, onlooker, and scout. Employed bees are responsible for exploiting the food sources already discovered and sharing information about these food sources' location and nectar quality with the onlooker bees. According to the information given by the worker bees, onlooker bees prefer a food source and try to find richer food sources within the neighborhood of the preferred food source. On the other hand, scout bees leave the hive to search for undiscovered rich food sources randomly.

The ABC algorithm starts the search process by generating random food sources. Then, employed bees search for new food sources locally having more nectar in the neighborhood of the food source in their memory by using the following equation:

$$x'_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \quad (13)$$

The employed bees share information about food sources with onlooker bees that are waiting in the hive, and onlooker bees then select their food source based on this information. An onlooker bee selects a food source based on the probability value computed using the fitness values provided by the employer bees as given in Eq. 14:

$$p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{CS} \text{fitness}_i} \quad (14)$$

The employed bee, whose solution cannot be improved during a predetermined number of trials, called the limit, becomes a scout, and its solution is abandoned. Then, the new scout bee starts looking for new random solutions.

### **Gray Wolf Optimizer**

The GWO, which Mirjalili proposed in 2014, is a swarm-based meta-heuristic algorithm inspired by gray wolves' social hierarchy and hunting strategy (Mirjalili et al., 2014). Gray wolves live in highly organized swarms, with numbers in the range of 5-12. There is a leadership team in the mathematical model to manage the swarm, and this team consists of 3 different managers. This leader team is divided into three different groups: alpha ( $\alpha$ ), beta ( $\beta$ ), and delta ( $\delta$ ), respectively. The strongest members among them are alpha. Members outside the leading team are called omega ( $\omega$ ).

The hunting behavior of gray wolves comprises four stages: social hierarchy, encircling prey, hunting, and attacking. To ensure social hierarchy, while the three best solutions are ranked and assigned as  $\alpha$ ,  $\beta$ , and  $\delta$ , respectively, the remaining solutions become  $\omega$ . Thus, the whole search process is carried out with these four different types of gray wolves. Gray wolves encircle during the hunt, and this behavior is modeled as the following equation:

$$\begin{aligned} \vec{D} &= |C \cdot \vec{X}_p(t) - \vec{X}(t)| \\ \vec{X}(t+1) &= \vec{X}_p(t) - A \cdot \vec{D} \end{aligned} \quad (15)$$

Herein, while  $\vec{X}_p$  is the position vector of prey,  $\vec{X}$  is the position vector of a gray wolf. The current iteration of the search is represented by  $t$ .  $A$  and  $C$ , which are calculated in the following equations, are the coefficients:

$$\begin{aligned} A &= 2a \cdot r_1 - a \\ C &= 2r_2 \end{aligned} \quad (16)$$

where  $r_1$  and  $r_2$ , which are selected in the range  $[0, 1]$ , are random numbers, the value of  $a$  is linearly decreased from 2 to 0 during the search. The hunting phase in the GWO is guided by the  $\alpha$ ,  $\beta$ , and  $\delta$  since they have better information than  $\omega$  wolves. This phase is modelled as in Eq. 17 and 18:

$$\begin{aligned} \vec{D}_\alpha &= |C_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta &= |C_1 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta &= |C_1 \cdot \vec{X}_\delta - \vec{X}| \end{aligned} \quad (17)$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - A_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta - A_2 \cdot \vec{D}_\beta \\ \vec{X}_3 &= \vec{X}_\delta - A_3 \cdot \vec{D}_\delta \end{aligned} \quad (18)$$

where  $\vec{X}_\alpha$ ,  $\vec{X}_\beta$  and  $\vec{X}_\delta$  are the position vectors respectively,  $A_1$ ,  $A_2$ ,  $A_3$ ,  $C_1$ ,  $C_2$ , and  $C_3$  are the coefficients. The coefficient values are calculated as in the given Eq. 16. The positions are then updated as in the following equation:

$$\vec{X}(t + 1) = (\vec{X}_1 + \vec{X}_2 + \vec{X}_3) / 3 \quad (19)$$

In the GWO, exploration and exploitation are balanced with the value of  $A$ . If  $A < 1$ , exploitation is performed, otherwise exploration is applied. This process is modeled by linearly decreasing the value of  $a$  from 2 to 0 during the search. Thus, the prey will stop, and the gray wolves will attack to complete the hunting phase.

### Moth-Flame Optimization Algorithm

The MFO (Mirjalili, 2015a), which modeled the natural behavior of moths, is a swarm-based meta-heuristic algorithm that was proposed by Mirjalili in 2015. MFO starts by generating a random population of moths within the search space. Then the fitness value of each moth is calculated, and the best position is tagged as flame. After that, the moths' new positions are calculated according to the spiral movement function given in the following equation:

$$M_i = D_i e^{b\theta} \cos(2\pi\theta) + F_j \quad (20)$$

where  $D_i$  indicates the space between the  $i$ th moth and  $j$ th flame,  $b$  is a constant which define the shape of the logarithmic spiral, and  $t$ , is selected in the range  $[-1, 1]$ , is a random number.

While the MFO focuses on exploration at the beginning of the search, it reduces the number of flames to make exploitation more intense later in the search process. Decreasing the number of flames is applied according to the following equation:

$$FlameNumber = round\left(N - t \cdot \frac{N-1}{T}\right) \quad (21)$$

where  $N$  indicates the maximum number of flames,  $t$ , and  $T$  are the current and maximum number of iterations, respectively.

### Ant Lion Optimization Algorithm

The ALO is a meta-heuristic algorithm that simulates the hunting behavior of antlions (Mirjalili, 2015b). The hunting mechanism of the antlions consists of five main steps, including the random walk of ants, setting traps, entrapment of ants in traps, catching prey, and re-setting traps. The algorithm starts with initializing the population of ants in a search space. Then, the position of each ant is updated utilizing a random walk model as follows:

$$x(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \dots, cumsum(2r(t_n) - 1)] \quad (22)$$

where  $t$  is the iteration index and  $r$  is a random number that can take only the values 0 or 1.

ALO, as a population-based algorithm, employs multiple ants and antlions and stores them in position matrices. In addition, a fitness function is utilized to evaluate each ant and ant lion; and the fitness values are also stored in matrices. In the ALO, ants move within the search space using random walks. The position of the ants is normalized using Eq. 23 to keep the random walks within the search space:

$$X_i^t = \frac{(x_i^t - a_i)(b_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad (23)$$

where  $a_i$  and  $b_i$  are the minimum and maximum random walks of the  $i$ th variable, respectively.  $c_i^t$  and  $d_i^t$  are the minimum and maximum values of the  $i$ th variable at  $t$ th iteration. Traps of the antlions affect the ants, and this assumption is modeled as follows:

$$\begin{aligned} c_i^t &= \text{Antlion}_j^t + c^t \\ d_i^t &= \text{Antlion}_j^t + d^t \end{aligned} \quad (24)$$

where  $c^t$  and  $d^t$  are the minimum and maximum values of all variables at the  $t$ th iteration,  $c_i^t$  and  $d_i^t$  are the minimum and maximum values of all variables for  $i$ th ant.  $\text{Antlion}_j^t$  indicates the position of  $j$ th antlion at the  $t$ th iteration.

In the ALO, two different mechanisms affect the ants. In each iteration, the best antlion so far ( $R_B^t$ ) is saved and made to affect all ants. Also, another antlion ( $R_A^t$ ) is selected by using a roulette wheel. Therefore, each ant randomly walks to the best antlion so far, and a selected antlion by the roulette wheel simultaneously as follows:

$$\text{Ant}_i^t = \frac{R_A^t + R_B^t}{2} \quad (25)$$

### Whale Optimization Algorithm

The WOA is a population-based meta-heuristic algorithm inspired by the bubble-net hunting strategy of humpback whales (Mirjalili & Lewis, 2016). Similar to other population-based algorithms, WOA starts by generating a set of random solutions. It then improves this set over iterations until a predefined criterion is met.

The intelligent foraging behavior of the humpback whales is the main inspiration of the WOA. Humpback whales corral their prey by producing a spiral bubble net. WOA also simulates the encircling behavior of humpback whales. Humpback whales circle their prey to start hunting them using the bubble net. This mechanism is modeled as follows:

$$X(t+1) = \begin{cases} X^*(t) - AD & p < 0.5 \\ D' e^{bl} \cos(2\pi t) + X^*(t) & p \geq 0.5 \end{cases} \quad (26)$$

where  $p$  is a random number in the range of 0 to 1,  $D' = |X^*(t) - X(t)|$  is the distance between  $i$ th whale and the prey,  $b$  is a constant that describes the shape of the spiral,  $l$  is a random number in the range of -1 to 1,  $t$  indicates the current iteration index,  $D = |CX^*(t) - X(t)|$ ,  $A = 2ar - a$ ,  $C = 2r$ ,  $a$  gradually decreases from 2 to 0 over iterations, and  $r$  is a vector of random numbers in the range of 0 to 1. The first element of Eq. 26 mimics the encircling behavior of humpback whales, while the former simulates the bubble net behavior. The random variable  $p$  is used to switch between these elements.

The WOA algorithm starts with a set of random solutions. Then, in each iteration, search agents update their positions regarding either a randomly selected agent or the best solution so far. The parameter  $a$  gradually decreases from 2 to 0 over iterations to guarantee exploration and exploitation. Furthermore, to provide exploration and exploitation, a search agent is randomly selected when  $|A| > 1$  to update the position of the search agents, while the best solution is selected when  $|A| < 1$ .

### Jellyfish Search Optimizer

The JS is a novel meta-heuristic algorithm inspired by the foraging behavior of the jellyfish swarms in the ocean (Chou & Truong, 2021). JS algorithm starts with an exploration stage in which jellyfish follow ocean currents to find the best locations. Then, an exploitation stage is conducted in which a jellyfish swarm is formed, and each jellyfish moves to find a better location in that swarm.

In the JS algorithm, jellyfish have two types of movement: passive and active. Passive movement is the movement of the jellyfish around its own location, and as a result of this movement, its location is updated using the following equation:

$$\vec{X}_i(t+1) = \vec{X}_i(t) + r * \gamma * (u_b - l_b) \quad (27)$$

where  $l_b$  and  $u_b$  are the lower and upper bounds of the search space, respectively;  $r$  is a random number in the range of 0 to 1;  $\gamma > 0$  is the movement coefficient which is related to the length of movement around the current location.



Active movement is the movement of jellyfish toward a better direction to find food in the swarm. For this purpose, a jellyfish other than the one of interest is randomly selected. The direction of the active movement is determined by constructing a vector from the jellyfish of interest to the selected jellyfish. If the amount of food in the location of selected jellyfish is higher, jellyfish of interest move toward selected jellyfish. However, if the amount of food in the location of selected jellyfish is lower, jellyfish of interest move directly away from the location of selected jellyfish. The active movement can be given as follows:

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \vec{r} * \vec{D} \quad (28)$$

where  $\vec{r}$  is a vector of random numbers in the range of 0 to 1 and  $\vec{D}$  is the direction calculated by the following equation:

$$\vec{D} = \begin{cases} \vec{X}_i(t) - \vec{X}_j(t) & f(\vec{X}_i(t)) - f(\vec{X}_j(t)) \\ \vec{X}_j(t) - \vec{X}_i(t) & otherwise \end{cases} \quad (29)$$

where  $j$  is the index of the selected jellyfish, and  $f$  means the fitness function. In the JS algorithm, a time control mechanism is used to regulate the movement of jellyfish and switch among these movements:

$$c(t) = \left(1 - \frac{t}{t_{max}}\right) * (2 * r - 1) \quad (30)$$

where  $t$  is the time specified as the current iteration number,  $t_{max}$  is the maximum number of iterations, and  $r$  is a random number in the range of 0 to 1.

## EXPERIMENTS

### Experimental Setup

In the experiments, ABC, PSO, GWO, WOA, ALO, MFO, and JS algorithms have been investigated to determine performance on the multilevel thresholding of the meta-heuristic algorithms. For a fair comparison, the number of function evaluations is taken into consideration instead of the iteration number as a stopping criterion. The number of function evaluations and the population size of algorithms are selected as 5000 and 30, respectively. Other control parameters of the algorithms are default values, and they are given in Table 1. The initial populations of all algorithms are generated uniformly and randomly among [0, 255].

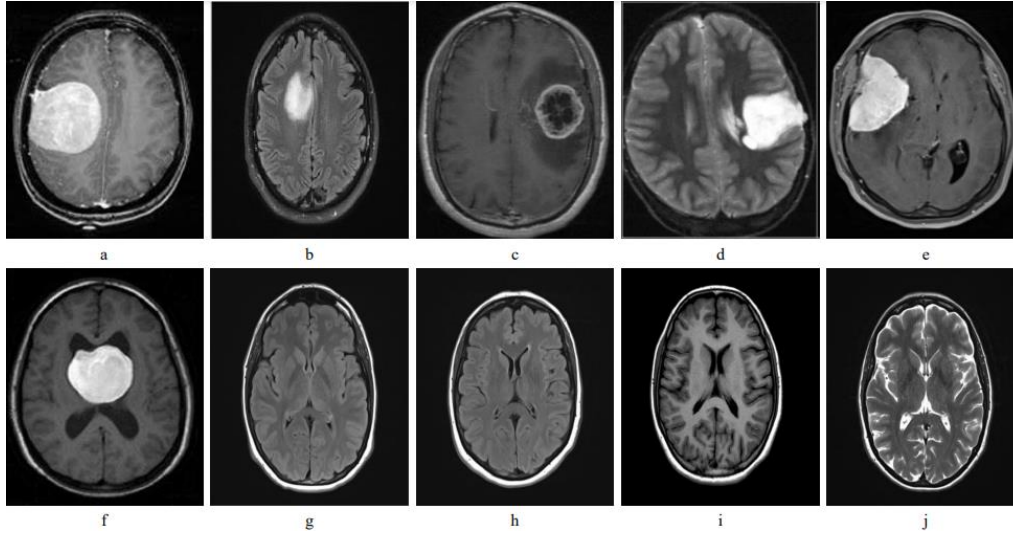
**Table 1.** Control Parameters of The Algorithms Used in Experiments

Algorithm	Control Parameters
ABC	$limit = (NP * D)/2$
PSO	$c_1 = 2, c_2 = 2, w = 0.2-0.9$
GWO	$a = 0-2$
WOA	$a = 0-2$
ALO	-
MFO	-
JS	-

Experiments were carried out for 2, 4, 6, and 8 number of threshold values and each of them was repeated 30 times to perform statistical analysis. Experiments were executed on a computer with an Intel i7 2600 processor clocked at 3.4 GHz and 16 GB of RAM using MATLAB software.

### Image Dataset

Experiments are carried out on brain MR images shown in Fig. 2. For experiments, ten different test images are selected from the "Brain MRI Images for Brain Tumor Detection" dataset (Chakrabarty, 2019). Among them, the first six MR images include tumors while others do not. From the image MR1 to image MR10, the image sizes are as follows: 180×218, 450×446, 230×275, 272×355, 272×331, 238×338, 512×512, 442×442, 728×725, and 442×442.



**Figure 2.** Brain MR Images Used in Experiments: **(a-f).** Images with Tumor **(g-j).** Images with No Tumor

### Objective Quality Metrics

In this study, we employ three different objective image quality assessment metrics to assess the performance of the compared algorithms. These metrics are reference-based Peak Signal-to-Noise Ratio (PSNR) Structural Similarity (SSIM), and non-reference-based Blind/Referenceless Naturalness Image Quality Evaluator (NIQE).

PSNR is the ratio between the maximum possible power of a signal and the power of the distorting noise that affects its quality. PSNR is calculated as a logarithmic quantity in the decibel scale as follows:

$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \quad (31)$$

where  $L$  is the maximum possible pixel value of the image, and MSE is the mean square error calculated as follows:

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (I_A(i, j) - I_B(i, j))^2 \quad (32)$$

The structural similarity (SSIM) is another reference-based metric that measures the similarity between two given images (Wang et al., 2004). The SSIM is calculated on numerous windows of the image as follows:

$$SSIM(I_A, I_B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1) + (\sigma_A^2 + \sigma_B^2 + C_2)} \quad (33)$$

where  $C_1 = (k_1L)^2$  and  $C_2 = (k_2L)^2$ ;  $T$  is the dynamic range of the pixel-values and ( $k_1 = 0.01, k_2 = 0.03$ ).

NIQE is a blind image quality assessment metric that uses measurable differences of statistical regularities in the images. First, the quality features are obtained by making use of a simple natural scene statistic (NSS) model. Then, the quality of a given image is computed as the difference between a multivariate Gaussian (MVG) fit of the NSS features that are derived from the image (Mittal et al., 2013).

### Experimental Results

This section presents the results of the experiments conducted to reveal the multilevel thresholding performance of the different meta-heuristic algorithms using minimum cross-entropy and between-class variance methods. The obtained results and their statistical analyses are given in subsections.

#### Evaluation of the Minimum Cross-Entropy Method

In this section, the multilevel thresholding performance of the different meta-heuristic algorithms that make use of the minimum cross-entropy method on brain MR images is investigated. Tables 2-5 provide the mean and standard deviation (std) of the best objective values, PSNR, SSIM, BRISQUE, and NIQE, values of the 30 independent runs in terms of the different threshold values, respectively. Besides, ranks are given in parentheses next to the mean values. Moreover, to make an overall assessment, average rank, standard deviation of rank, and median rank (in parentheses) values are provided in the last row of the tables.

**Table 2.** Comparison of the Best Objective Value Computed of Swarm-Based Optimization Algorithms Using the Minimum Cross-Entropy Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	3.0916 (4)	0.00	3.0916 (4)	0.00	3.0916 (4)	0.00	3.0916 (4)	0.00	3.0916 (4)	0.00	3.0916 (4)	0.00	3.0916 (4)	0.00
	4	<b>1.0141 (1)</b>	0.00	1.0142 (3)	0.00	1.0145 (7)	0.00	1.0144 (5)	0.00	1.0141 (2)	0.00	1.0145 (6)	0.00	1.0144 (4)	0.00
	6	0.4894 (3)	0.00	0.4921 (4)	0.03	0.4894 (2)	0.00	0.4927 (5)	0.03	0.4999 (7)	0.03	0.4944 (6)	0.03	<b>0.4871 (1)</b>	0.00
	8	0.3314 (4)	0.01	0.3313 (3)	0.02	0.3373 (7)	0.02	0.3322 (6)	0.02	<b>0.3270 (1)</b>	0.01	0.3305 (2)	0.02	0.3314 (5)	0.00
MRI2	2	1.7777 (4)	0.00	1.7777 (4)	0.00	1.7777 (4)	0.00	1.7777 (4)	0.00	1.7777 (4)	0.00	1.7777 (4)	0.00	1.7777 (4)	0.00
	4	<b>0.5300 (1)</b>	0.00	0.5300 (3)	0.00	0.5404 (7)	0.06	0.5300 (6)	0.00	0.5300 (2)	0.00	0.5300 (5)	0.00	0.5300 (4)	0.00
	6	0.2769 (2)	0.00	0.2787 (5)	0.01	0.2826 (6)	0.02	0.2870 (7)	0.03	0.2785 (4)	0.01	0.2770 (3)	0.00	<b>0.2768 (1)</b>	0.00
	8	0.1885 (5)	0.00	0.1857 (2)	0.00	0.1962 (7)	0.02	0.1882 (3)	0.01	0.1882 (4)	0.01	0.1885 (6)	0.01	<b>0.1850 (1)</b>	0.00
MRI3	2	<b>1.3797 (4)</b>	0.00	<b>1.3797 (4)</b>	0.00	<b>1.3797 (4)</b>	0.00	<b>1.3797 (4)</b>	0.00	<b>1.3797 (4)</b>	0.00	<b>1.3797 (4)</b>	0.00	<b>1.3797 (4)</b>	0.00
	4	<b>0.4615 (1)</b>	0.00	0.4686 (5)	0.04	0.4615 (3)	0.00	0.4757 (6)	0.05	0.4827 (7)	0.06	0.4617 (4)	0.00	0.4615 (2)	0.00
	6	0.2543 (2)	0.00	0.2587 (4)	0.02	0.2641 (5)	0.03	0.2644 (6)	0.03	0.2777 (7)	0.04	0.2570 (3)	0.01	<b>0.2541 (1)</b>	0.00
	8	0.1727 (2)	0.01	0.1762 (3)	0.02	0.1881 (7)	0.02	0.1836 (5)	0.02	0.1866 (6)	0.02	0.1771 (4)	0.02	<b>0.1676 (1)</b>	0.00
MRI4	2	<b>3.0400 (4)</b>	0.00	<b>3.0400 (4)</b>	0.00	<b>3.0400 (4)</b>	0.00	<b>3.0400 (4)</b>	0.00	<b>3.0400 (4)</b>	0.00	<b>3.0400 (4)</b>	0.00	<b>3.0400 (4)</b>	0.00
	4	0.8945 (5)	0.00	<b>0.8943 (1)</b>	0.00	0.8945 (4)	0.00	0.8947 (6)	0.00	0.8943 (2)	0.00	0.9011 (7)	0.03	0.8943 (3)	0.00
	6	<b>0.4493 (1.5)</b>	0.00	<b>0.4493 (1.5)</b>	0.00	0.4721 (7)	0.06	0.4698 (6)	0.05	0.4495 (3)	0.00	0.4571 (5)	0.03	0.4495 (4)	0.00
	8	0.3032 (2)	0.01	0.3054 (3)	0.02	0.3133 (6)	0.02	0.3138 (7)	0.03	0.3060 (4)	0.01	0.3060 (5)	0.02	<b>0.3007 (1)</b>	0.00
MRI5	2	<b>2.1409 (4)</b>	0.00	<b>2.1409 (4)</b>	0.00	<b>2.1409 (4)</b>	0.00	<b>2.1409 (4)</b>	0.00	<b>2.1409 (4)</b>	0.00	<b>2.1409 (4)</b>	0.00	<b>2.1409 (4)</b>	0.00
	4	0.7410 (5)	0.00	0.7410 (4)	0.00	0.7410 (3)	0.00	0.7544 (7)	0.07	<b>0.7410 (1.5)</b>	0.00	0.7410 (6)	0.00	<b>0.7410 (1.5)</b>	0.00
	6	0.3905 (2)	0.00	0.3944 (6)	0.02	0.3927 (5)	0.01	0.4097 (7)	0.04	0.3914 (3)	0.01	0.3920 (4)	0.01	<b>0.3904 (1)</b>	0.00
	8	0.2598 (5)	0.01	0.2596 (4)	0.02	0.2557 (3)	0.01	0.2644 (7)	0.03	0.2528 (2)	0.01	0.2611 (6)	0.02	<b>0.2497 (1)</b>	0.00
MRI6	2	<b>2.7510 (4)</b>	0.00	<b>2.7510 (4)</b>	0.00	<b>2.7510 (4)</b>	0.00	<b>2.7510 (4)</b>	0.00	<b>2.7510 (4)</b>	0.00	<b>2.7510 (4)</b>	0.00	<b>2.7510 (4)</b>	0.00
	4	<b>0.8779 (2)</b>	0.00	<b>0.8779 (2)</b>	0.00	0.8780 (6)	0.00	0.8891 (7)	0.03	<b>0.8779 (2)</b>	0.00	0.8780 (5)	0.00	0.8779 (4)	0.00
	6	0.4535 (3)	0.00	<b>0.4523 (1)</b>	0.00	0.4560 (5)	0.00	0.4583 (7)	0.03	0.4569 (6)	0.02	0.4550 (4)	0.01	0.4534 (2)	0.00
	8	0.3158 (7)	0.02	0.2997 (4)	0.03	0.2906 (2)	0.01	<b>0.2895 (1)</b>	0.03	0.3088 (6)	0.03	0.3003 (5)	0.03	0.2932 (3)	0.02
MRI7	2	<b>1.1153 (4)</b>	0.00	<b>1.1153 (4)</b>	0.00	<b>1.1153 (4)</b>	0.00	<b>1.1153 (4)</b>	0.00	<b>1.1153 (4)</b>	0.00	<b>1.1153 (4)</b>	0.00	<b>1.1153 (4)</b>	0.00
	4	<b>0.3453 (2)</b>	0.00	<b>0.3453 (2)</b>	0.00	0.3453 (4)	0.00	0.3533 (7)	0.04	<b>0.3453 (2)</b>	0.00	0.3453 (6)	0.00	0.3453 (5)	0.00
	6	<b>0.1973 (1)</b>	0.00	0.1991 (3)	0.00	0.2034 (6)	0.02	0.2066 (7)	0.03	0.2002 (5)	0.00	0.1999 (4)	0.00	0.1989 (2)	0.00
	8	0.1173 (2)	0.00	<b>0.1172 (1)</b>	0.00	0.1254 (7)	0.01	0.1253 (6)	0.02	0.1192 (4)	0.00	0.1206 (5)	0.01	0.1185 (3)	0.00
MRI8	2	<b>0.9900 (4)</b>	0.00	<b>0.9900 (4)</b>	0.00	<b>0.9900 (4)</b>	0.00	<b>0.9900 (4)</b>	0.00	<b>0.9900 (4)</b>	0.00	<b>0.9900 (4)</b>	0.00	<b>0.9900 (4)</b>	0.00
	4	<b>0.3242 (1.5)</b>	0.00	<b>0.3242 (1.5)</b>	0.00	0.3291 (6.5)	0.03	0.3291 (6.5)	0.03	0.3242 (3)	0.00	0.3242 (5)	0.00	0.3242 (4)	0.00
	6	<b>0.1679 (1)</b>	0.00	0.1698 (5)	0.01	0.1756 (6)	0.02	0.1790 (7)	0.02	0.1681 (3)	0.00	0.1681 (4)	0.00	0.1680 (2)	0.00
	8	0.1086 (3)	0.00	<b>0.1075 (1)</b>	0.00	0.1251 (7)	0.01	0.1211 (6)	0.01	0.1106 (5)	0.00	0.1098 (4)	0.00	0.1078 (2)	0.00
MRI9	2	<b>0.9715 (4)</b>	0.00	<b>0.9715 (4)</b>	0.00	<b>0.9715 (4)</b>	0.00	<b>0.9715 (4)</b>	0.00	<b>0.9715 (4)</b>	0.00	<b>0.9715 (4)</b>	0.00	<b>0.9715 (4)</b>	0.00
	4	0.3227 (2)	0.00	0.3683 (6.5)	0.08	0.3229 (3)	0.00	0.3292 (5)	0.04	0.3683 (6.5)	0.08	0.3237 (4)	0.00	<b>0.3227 (1)</b>	0.00
	6	0.1602 (2)	0.01	0.1992 (6)	0.04	0.1654 (4)	0.02	0.1665 (5)	0.02	0.2118 (7)	0.06	0.1644 (3)	0.02	<b>0.1579 (1)</b>	0.00
	8	0.0989 (2)	0.01	0.1245 (6)	0.04	0.1029 (4)	0.01	0.1061 (5)	0.02	0.1342 (7)	0.03	0.1008 (3)	0.01	<b>0.0914 (1)</b>	0.01
MRI10	2	<b>1.3727 (4)</b>	0.00	<b>1.3727 (4)</b>	0.00	<b>1.3727 (4)</b>	0.00	<b>1.3727 (4)</b>	0.00	<b>1.3727 (4)</b>	0.00	<b>1.3727 (4)</b>	0.00	<b>1.3727 (4)</b>	0.00
	4	0.5341 (2)	0.00	0.5343 (4)	0.00	0.5424 (6)	0.04	0.5425 (7)	0.04	<b>0.5341 (1)</b>	0.00	0.5351 (5)	0.00	0.5343 (3)	0.00
	6	0.2705 (3)	0.00	0.2726 (5)	0.01	0.2766 (6)	0.02	0.2839 (7)	0.03	<b>0.2703 (1)</b>	0.00	0.2708 (4)	0.00	0.2704 (2)	0.00
	8	0.1646 (4)	0.00	<b>0.1637 (1)</b>	0.00	0.1715 (6)	0.02	0.1795 (7)	0.02	0.1639 (2)	0.00	0.1650 (5)	0.00	0.1642 (3)	0.00
Av. Rank		2.98±1.44 (3)		3.51±1.5 (4)		4.94±1.52 (4.5)		5.49±1.46 (6)		3.9±1.83 (4)		4.45±1.04 (4)		<b>2.74±1.36 (3)</b>	

It can be seen from the data in Table 2 that the most successful algorithm is the JS in terms of minimum cross-entropy fitness value, and the JS is followed by ABC, PSO, ALO, MFO, GWO, and WOA, respectively. While all the algorithms find the same fitness value for 2-level thresholding, as the threshold levels increase, the results of the algorithms diverge, and the performance of the JS, ABC, and PSO algorithms becomes more prominent.

In Table 3, a comparison of various algorithms based on their PSNR values at different threshold levels is presented. The algorithms are ranked according to their performance from highest to lowest, with ABC being the top-performing algorithm, followed by PSO, JS, MFO, ALO, WOA, and GWO.

**Table 3.** Comparison of PSNR Values of Swarm-Based Optimization Algorithms Using the Minimum Cross-Entropy Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	<b>13.3033 (4)</b>	0.00	<b>13.3033 (4)</b>	0.00	<b>13.3033 (4)</b>	0.00	<b>13.3033 (4)</b>	0.00	<b>13.3033 (4)</b>	0.00	<b>13.3033 (4)</b>	0.00	<b>13.3033 (4)</b>	0.00
	4	<b>19.8454 (1)</b>	0.00	19.8347 (3)	0.04	19.7438 (7)	0.06	19.7875 (4)	0.07	19.8426 (2)	0.02	19.7860 (5)	0.07	19.7813 (6)	0.07
	6	<b>22.4544 (1)</b>	0.24	22.2617 (3)	0.24	22.0851 (7)	0.20	22.1322 (6)	0.24	22.2546 (5)	0.13	22.3367 (2)	0.18	22.2580 (4)	0.12
	8	24.8234 (2)	0.79	24.6213 (3)	0.78	23.3384 (7)	0.29	23.6820 (6)	0.58	24.5427 (5)	0.80	24.5656 (4)	0.76	<b>25.1247 (1)</b>	0.35
MRI2	2	<b>16.6174 (4)</b>	0.00	<b>16.6174 (4)</b>	0.00	<b>16.6174 (4)</b>	0.00	<b>16.6174 (4)</b>	0.00	<b>16.6174 (4)</b>	0.00	<b>16.6174 (4)</b>	0.00	<b>16.6174 (4)</b>	0.00
	4	<b>22.0180 (1)</b>	0.00	22.0162 (2)	0.00	21.9352 (7)	0.35	21.9941 (6)	0.03	22.0147 (3)	0.01	22.0026 (5)	0.03	22.0072 (4)	0.02
	6	<b>24.3532 (1)</b>	0.04	24.2996 (3)	0.12	24.1530 (7)	0.22	24.2150 (6)	0.21	24.2964 (4)	0.08	24.3088 (2)	0.07	24.2847 (5)	0.06
	8	<b>25.7614 (1)</b>	0.27	25.6654 (4)	0.21	25.2960 (7)	0.17	25.4538 (6)	0.18	25.7206 (3)	0.28	25.7227 (2)	0.22	25.5860 (5)	0.21
MRI3	2	<b>17.6085 (4)</b>	0.00	<b>17.6085 (4)</b>	0.00	<b>17.6085 (4)</b>	0.00	<b>17.6085 (4)</b>	0.00	<b>17.6085 (4)</b>	0.00	<b>17.6085 (4)</b>	0.00	<b>17.6085 (4)</b>	0.00
	4	24.1399 (4)	0.01	24.0729 (5)	0.34	24.1467 (3)	0.02	24.0311 (6)	0.47	23.9483 (7)	0.58	24.1492 (2)	0.02	<b>24.1494 (1)</b>	0.02
	6	<b>27.0097 (1)</b>	0.04	26.9138 (3)	0.34	26.7914 (6)	0.42	26.8095 (5)	0.43	26.6397 (7)	0.65	26.8999 (4)	0.28	27.0095 (2)	0.03
	8	28.4459 (4)	0.15	28.4638 (2)	0.30	28.0260 (7)	0.44	28.1363 (5)	0.40	28.0316 (6)	0.51	28.4490 (3)	0.32	<b>28.4824 (1)</b>	0.07
MRI4	2	<b>15.0381 (4)</b>	0.00	<b>15.0381 (4)</b>	0.00	<b>15.0381 (4)</b>	0.00	<b>15.0381 (4)</b>	0.00	<b>15.0381 (4)</b>	0.00	<b>15.0381 (4)</b>	0.00	<b>15.0381 (4)</b>	0.00
	4	19.3399 (7)	0.04	19.3515 (2)	0.00	19.3497 (4)	0.02	19.3483 (5)	0.03	19.3502 (3)	0.01	<b>19.3791 (1)</b>	0.11	19.3469 (6)	0.02
	6	<b>22.9525 (1.5)</b>	0.00	<b>22.9525 (1.5)</b>	0.00	22.6120 (7)	0.73	22.6269 (6)	0.74	22.9090 (4)	0.07	22.7980 (5)	0.51	22.9280 (3)	0.04
	8	<b>24.5359 (1)</b>	0.21	24.4498 (3)	0.26	23.9004 (7)	0.37	24.1681 (6)	0.38	24.4315 (4)	0.28	24.3749 (5)	0.36	24.5015 (2)	0.19
MRI5	2	<b>14.8031 (4)</b>	0.00	<b>14.8031 (4)</b>	0.00	<b>14.8031 (4)</b>	0.00	<b>14.8031 (4)</b>	0.00	<b>14.8031 (4)</b>	0.00	<b>14.8031 (4)</b>	0.00	<b>14.8031 (4)</b>	0.00
	4	20.5866 (2)	0.01	20.5850 (3)	0.01	20.5833 (4)	0.01	20.5203 (7)	0.34	20.5819 (5.5)	0.00	<b>20.5912 (1)</b>	0.03	20.5819 (5.5)	0.00
	6	<b>24.1313 (1)</b>	0.06	24.0363 (4)	0.25	23.9662 (6)	0.18	23.6993 (7)	0.57	24.0484 (3)	0.04	24.0246 (5)	0.22	24.0939 (2)	0.06
	8	<b>26.1605 (1)</b>	0.54	25.8579 (3)	0.47	25.3782 (7)	0.29	25.5232 (6)	0.65	25.7465 (4)	0.22	26.0230 (2)	0.44	25.7395 (5)	0.13
MRI6	2	<b>15.4121 (4)</b>	0.00	<b>15.4121 (4)</b>	0.00	<b>15.4121 (4)</b>	0.00	<b>15.4121 (4)</b>	0.00	<b>15.4121 (4)</b>	0.00	<b>15.4121 (4)</b>	0.00	<b>15.4121 (4)</b>	0.00
	4	20.4431 (4)	0.00	20.4431 (4)	0.00	20.4453 (2)	0.03	20.3062 (7)	0.38	20.4431 (4)	0.00	20.4418 (6)	0.02	<b>20.4573 (1)</b>	0.02
	6	<b>23.6468 (1)</b>	0.08	23.6158 (2)	0.00	23.5060 (6)	0.18	23.5096 (5)	0.50	23.3650 (7)	0.65	23.5383 (4)	0.47	23.5807 (3)	0.08
	8	<b>25.9086 (1)</b>	0.74	25.5847 (2)	0.77	24.6871 (7)	0.31	24.8099 (6)	0.32	25.1442 (5)	0.67	25.5534 (3)	0.82	25.2404 (4)	0.47
MRI7	2	<b>16.4731 (4)</b>	0.00	<b>16.4731 (4)</b>	0.00	<b>16.4731 (4)</b>	0.00	<b>16.4731 (4)</b>	0.00	<b>16.4731 (4)</b>	0.00	<b>16.4731 (4)</b>	0.00	<b>16.4731 (4)</b>	0.00
	4	21.3023 (4)	0.00	21.3023 (4)	0.00	21.3029 (2)	0.01	21.2565 (7)	0.25	21.3023 (4)	0.00	<b>21.3045 (1)</b>	0.01	21.2985 (6)	0.01
	6	22.8542 (6)	0.29	23.1542 (2)	0.36	22.7471 (7)	0.18	23.0530 (4)	0.44	<b>23.2739 (1)</b>	0.30	23.0789 (3)	0.30	22.9788 (5)	0.28
	8	24.4182 (2)	0.04	24.3683 (5)	0.06	24.2912 (7)	0.06	<b>24.4883 (1)</b>	0.57	24.4086 (3)	0.09	24.3701 (4)	0.07	24.3552 (6)	0.11
MRI8	2	<b>16.4765 (4)</b>	0.00	<b>16.4765 (4)</b>	0.00	<b>16.4765 (4)</b>	0.00	<b>16.4765 (4)</b>	0.00	<b>16.4765 (4)</b>	0.00	<b>16.4765 (4)</b>	0.00	<b>16.4765 (4)</b>	0.00
	4	<b>20.9000 (1.5)</b>	0.00	<b>20.9000 (1.5)</b>	0.00	20.8616 (7)	0.21	20.8763 (6)	0.13	20.8965 (3)	0.01	20.8948 (5)	0.02	20.8962 (4)	0.02
	6	<b>23.1611 (1)</b>	0.01	23.1178 (4)	0.25	22.9503 (6)	0.46	22.9368 (7)	0.48	23.1060 (5)	0.05	23.1413 (3)	0.03	23.1460 (2)	0.02
	8	24.0562 (3)	0.06	24.0514 (4)	0.01	23.9615 (7)	0.16	<b>24.1869 (1)</b>	0.65	24.0827 (2)	0.11	24.0354 (6)	0.07	24.0495 (5)	0.03
MRI9	2	<b>25.2276 (4)</b>	0.00	<b>25.2276 (4)</b>	0.00	<b>25.2276 (4)</b>	0.00	<b>25.2276 (4)</b>	0.00	<b>25.2276 (4)</b>	0.00	<b>25.2276 (4)</b>	0.00	<b>25.2276 (4)</b>	0.00
	4	<b>29.7878 (1)</b>	0.02	29.3072 (6.5)	0.87	29.7715 (3)	0.05	29.7128 (5)	0.37	29.3072 (6.5)	0.87	29.7316 (4)	0.09	29.7765 (2)	0.02
	6	<b>32.5910 (1)</b>	0.31	31.7747 (6)	0.80	32.3749 (5)	0.38	32.4648 (3)	0.51	31.4962 (7)	1.03	32.4143 (4)	0.40	32.5824 (2)	0.10
	8	<b>34.7554 (1)</b>	0.47	33.7887 (6)	1.00	33.9625 (5)	0.45	34.1714 (4)	0.63	33.3308 (7)	0.87	34.5463 (3)	0.70	34.5983 (2)	0.21
MRI10	2	<b>16.8109 (4)</b>	0.00	<b>16.8109 (4)</b>	0.00	<b>16.8109 (4)</b>	0.00	<b>16.8109 (4)</b>	0.00	<b>16.8109 (4)</b>	0.00	<b>16.8109 (4)</b>	0.00	<b>16.8109 (4)</b>	0.00
	4	21.0453 (5)	0.02	21.0502 (4)	0.01	20.9813 (7)	0.34	20.9876 (6)	0.34	21.0513 (2)	0.01	<b>21.0663 (1)</b>	0.06	21.0512 (3)	0.01
	6	<b>23.3808 (1)</b>	0.04	23.3466 (4)	0.14	23.2415 (7)	0.19	23.2431 (6)	0.31	23.3594 (3)	0.02	23.3286 (5)	0.05	23.3634 (2)	0.02
	8	24.7237 (4)	0.07	<b>24.7313 (1)</b>	0.02	24.5980 (6)	0.20	24.5962 (7)	0.23	24.7072 (5)	0.04	24.7242 (3)	0.06	24.7266 (2)	0.05
Av. Rank	<b>2.65±1.68 (2)</b>		3.51±1.24 (4)		5.42±1.63 (6)		5.05±1.48 (5)		4.25±1.5 (4)		3.58±1.34 (4)		3.54±1.52 (4)		

Table 4 shows the obtained SSIM results of the compared algorithms in terms of different thresholding levels. Considering the SSIM results, WOA is the best algorithm, GWO is the runner-up algorithm, MFO is the third-best algorithm, and PSO is the poorest algorithm.

At last, Table 5 provides NIQE metric results of the compared algorithms. It can be seen from the table that JS has the best average rank while ALO has the worst. MFO is the runner-up algorithm, followed by ABC, PSO, GWO, and WOA. It can be seen that the results of the PSNR and NIQE metrics are consistent with the optimized fitness function value of the minimum cross-entropy method. On the other hand, SSIM metric results are not correlated with the main objective of optimization algorithms.

**Table 4.** Comparison of SSIM Values of Swarm-Based Optimization Algorithms Using the Minimum Cross-Entropy Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	<b>0.3912 (4)</b>	0.00	<b>0.3912 (4)</b>	0.00	<b>0.3912 (4)</b>	0.00	<b>0.3912 (4)</b>	0.00	<b>0.3912 (4)</b>	0.00	<b>0.3912 (4)</b>	0.00	<b>0.3912 (4)</b>	0.00
	4	0.5893 (5)	0.00	0.5893 (3)	0.00	0.5891 (7)	0.00	0.5892 (6)	0.00	0.5893 (4)	0.00	<b>0.5894 (1)</b>	0.00	0.5893 (2)	0.00
	6	0.7086 (4)	0.00	0.7075 (5)	0.01	<b>0.7112 (1)</b>	0.00	0.7095 (2)	0.01	0.7038 (7)	0.01	0.7060 (6)	0.01	0.7094 (3)	0.00
	8	0.7618 (5)	0.01	0.7607 (6)	0.01	<b>0.8035 (1)</b>	0.02	0.7816 (2)	0.02	0.7638 (4)	0.01	0.7658 (3)	0.01	0.7561 (7)	0.01
MRI2	2	<b>0.2484 (4)</b>	0.00	<b>0.2484 (4)</b>	0.00	<b>0.2484 (4)</b>	0.00	<b>0.2484 (4)</b>	0.00	<b>0.2484 (4)</b>	0.00	<b>0.2484 (4)</b>	0.00	<b>0.2484 (4)</b>	0.00
	4	0.3783 (7)	0.00	0.3787 (2)	0.00	<b>0.3801 (1)</b>	0.01	0.3787 (3)	0.00	0.3784 (6)	0.00	0.3786 (4)	0.00	0.3786 (5)	0.00
	6	0.4440 (7)	0.00	0.4452 (5)	0.00	0.4533 (2)	0.01	<b>0.4723 (1)</b>	0.08	0.4452 (6)	0.00	0.4465 (4)	0.00	0.4472 (3)	0.00
	8	0.4899 (4)	0.02	0.4884 (6)	0.02	<b>0.5388 (1)</b>	0.02	0.5266 (2)	0.07	0.4885 (5)	0.02	0.4840 (7)	0.01	0.4990 (3)	0.02
MRI3	2	0.5267 (4)	0.00	0.5267 (4)	0.00	0.5267 (4)	0.00	0.5267 (4)	0.00	0.5267 (4)	0.00	0.5267 (4)	0.00	0.5267 (4)	0.00
	4	0.6618 (5)	0.00	0.6606 (6)	0.00	0.6623 (4)	0.00	<b>0.6633 (1)</b>	0.00	0.6591 (7)	0.01	0.6628 (2)	0.00	0.6624 (3)	0.00
	6	0.7266 (6)	0.00	0.7268 (4)	0.01	0.7270 (2)	0.00	0.7269 (3)	0.00	0.7180 (7)	0.02	<b>0.7290 (1)</b>	0.01	0.7267 (5)	0.00
	8	0.7865 (4)	0.02	0.7807 (7)	0.02	<b>0.8179 (1)</b>	0.01	0.7999 (2)	0.02	0.7823 (6)	0.02	0.7864 (5)	0.02	0.7893 (3)	0.00
MRI4	2	<b>0.4288 (4)</b>	0.00	<b>0.4288 (4)</b>	0.00	<b>0.4288 (4)</b>	0.00	<b>0.4288 (4)</b>	0.00	<b>0.4288 (4)</b>	0.00	<b>0.4288 (4)</b>	0.00	<b>0.4288 (4)</b>	0.00
	4	0.6281 (4)	0.00	0.6281 (5)	0.00	0.6289 (2)	0.00	<b>0.6290 (1)</b>	0.00	0.6281 (6)	0.00	0.6259 (7)	0.01	0.6282 (3)	0.00
	6	0.6884 (6.5)	0.00	0.6884 (6.5)	0.00	0.6906 (3)	0.01	<b>0.6988 (1)</b>	0.03	0.6887 (5)	0.00	0.6913 (2)	0.01	0.6887 (4)	0.00
	8	0.7503 (7)	0.02	0.7636 (4)	0.03	0.7578 (6)	0.01	<b>0.7768 (1)</b>	0.04	0.7682 (2)	0.04	0.7598 (5)	0.03	0.7670 (3)	0.04
MRI5	2	<b>0.4498 (4)</b>	0.00	<b>0.4498 (4)</b>	0.00	<b>0.4498 (4)</b>	0.00	<b>0.4498 (4)</b>	0.00	<b>0.4498 (4)</b>	0.00	<b>0.4498 (4)</b>	0.00	<b>0.4498 (4)</b>	0.00
	4	0.6232 (6)	0.00	0.6232 (5)	0.00	0.6232 (4)	0.00	<b>0.6253 (1)</b>	0.01	0.6232 (2.5)	0.00	0.6231 (7)	0.00	0.6232 (2.5)	0.00
	6	0.6921 (3)	0.00	0.6907 (7)	0.01	0.6907 (6)	0.00	<b>0.6978 (1)</b>	0.01	0.6910 (5)	0.00	0.6931 (2)	0.01	0.6912 (4)	0.00
	8	0.7500 (7)	0.03	0.7561 (5)	0.03	<b>0.7779 (1)</b>	0.01	0.7748 (2)	0.02	0.7703 (3)	0.01	0.7505 (6)	0.03	0.7699 (4)	0.01
MRI6	2	<b>0.5547 (4)</b>	0.00	<b>0.5547 (4)</b>	0.00	<b>0.5547 (4)</b>	0.00	<b>0.5547 (4)</b>	0.00	<b>0.5547 (4)</b>	0.00	<b>0.5547 (4)</b>	0.00	<b>0.5547 (4)</b>	0.00
	4	0.7320 (6)	0.00	0.7320 (6)	0.00	0.7320 (4)	0.00	<b>0.7360 (1)</b>	0.01	0.7320 (6)	0.00	0.7322 (3)	0.00	0.7322 (2)	0.00
	6	0.8070 (4)	0.00	0.8068 (5)	0.00	<b>0.8090 (1)</b>	0.00	0.8066 (7)	0.00	0.8084 (2)	0.01	0.8077 (3)	0.00	0.8067 (6)	0.00
	8	0.8497 (7)	0.01	0.8607 (5)	0.01	<b>0.8722 (1)</b>	0.01	0.8694 (2)	0.01	0.8532 (6)	0.02	0.8612 (4)	0.01	0.8619 (3)	0.01
MRI7	2	<b>0.3012 (4)</b>	0.00	<b>0.3012 (4)</b>	0.00	<b>0.3012 (4)</b>	0.00	<b>0.3012 (4)</b>	0.00	<b>0.3012 (4)</b>	0.00	<b>0.3012 (4)</b>	0.00	<b>0.3012 (4)</b>	0.00
	4	0.3756 (3)	0.00	0.3756 (3)	0.00	0.3756 (6)	0.00	<b>0.3774 (1)</b>	0.01	0.3756 (3)	0.00	0.3756 (5)	0.00	0.3755 (7)	0.00
	6	0.4146 (2)	0.00	0.4134 (4)	0.00	<b>0.4196 (1)</b>	0.01	0.4144 (3)	0.01	0.4123 (6)	0.00	0.4123 (7)	0.00	0.4128 (5)	0.00
	8	0.4395 (6)	0.00	0.4409 (4)	0.00	0.4553 (2)	0.02	<b>0.4846 (1)</b>	0.11	0.4393 (7)	0.00	0.4487 (3)	0.02	0.4406 (5)	0.00
MRI8	2	<b>0.2784 (4)</b>	0.00	<b>0.2784 (4)</b>	0.00	<b>0.2784 (4)</b>	0.00	<b>0.2784 (4)</b>	0.00	<b>0.2784 (4)</b>	0.00	<b>0.2784 (4)</b>	0.00	<b>0.2784 (4)</b>	0.00
	4	0.3520 (3.5)	0.00	0.3520 (3.5)	0.00	0.3542 (2)	0.01	<b>0.3616 (1)</b>	0.05	0.3520 (5)	0.00	0.3519 (7)	0.00	0.3519 (6)	0.00
	6	0.3947 (5)	0.00	0.3965 (3)	0.01	0.4016 (2)	0.02	<b>0.4210 (1)</b>	0.08	0.3949 (4)	0.00	0.3947 (6)	0.00	0.3945 (7)	0.00
	8	0.4240 (6)	0.00	0.4241 (5)	0.00	0.4522 (2)	0.02	<b>0.4755 (1)</b>	0.11	0.4245 (4)	0.01	0.4275 (3)	0.01	0.4232 (7)	0.00
MRI9	2	<b>0.8342 (4)</b>	0.00	<b>0.8342 (4)</b>	0.00	<b>0.8342 (4)</b>	0.00	<b>0.8342 (4)</b>	0.00	<b>0.8342 (4)</b>	0.00	<b>0.8342 (4)</b>	0.00	<b>0.8342 (4)</b>	0.00
	4	0.9311 (2)	0.00	0.9224 (6.5)	0.02	0.9309 (3)	0.00	0.9299 (5)	0.01	0.9224 (6.5)	0.02	0.9308 (4)	0.00	<b>0.9312 (1)</b>	0.00
	6	0.9590 (2)	0.00	0.9540 (6)	0.01	0.9578 (5)	0.00	0.9589 (3)	0.00	0.9498 (7)	0.01	0.9588 (4)	0.00	<b>0.9593 (1)</b>	0.00
	8	0.9762 (2)	0.00	0.9695 (6)	0.01	0.9718 (5)	0.00	0.9734 (4)	0.01	0.9667 (7)	0.01	0.9753 (3)	0.00	<b>0.9765 (1)</b>	0.00
MRI10	2	<b>0.2830 (4)</b>	0.00	<b>0.2830 (4)</b>	0.00	<b>0.2830 (4)</b>	0.00	<b>0.2830 (4)</b>	0.00	<b>0.2830 (4)</b>	0.00	<b>0.2830 (4)</b>	0.00	<b>0.2830 (4)</b>	0.00
	4	0.3408 (7)	0.00	0.3415 (5)	0.00	0.3430 (3)	0.01	<b>0.3439 (1)</b>	0.01	0.3410 (6)	0.00	0.3436 (2)	0.01	0.3416 (4)	0.00
	6	0.3983 (6)	0.00	0.3982 (7)	0.00	0.4024 (2)	0.02	<b>0.4094 (1)</b>	0.02	0.3987 (3)	0.00	0.3983 (4)	0.00	0.3983 (5)	0.00
	8	0.4383 (7)	0.00	0.4391 (6)	0.00	0.4527 (2)	0.02	<b>0.4757 (1)</b>	0.07	0.4398 (5)	0.00	0.4404 (4)	0.01	0.4408 (3)	0.00
Av Rank		4.72±1.56 (4)		4.79±1.23 (5)		3.08±1.67 (3)		<b>2.55±1.62 (2)</b>		4.83±1.46 (4.5)		4.1±1.6 (4)		3.94±1.56 (4)	

### Evaluation of the Between-Class Variance Method

This section is concerned with the multilevel thresholding performance of the different meta-heuristic algorithms that use the between-class variance method on brain MR images. As in the previous section, Tables 6-9 provide the mean and standard deviation (std) of the best objective values, PSNR, SSIM, and NIQE values of the 30 independent runs in terms of the different numbers of the threshold value, respectively. Besides, ranks are provided in parenthesis next to the mean values. The average rank, the standard deviation, and median rank (in parenthesis) values are also given in the last row of the tables to make an overall assessment.

**Table 5.** Comparison of NIQE Values of Swarm-Based Optimization Algorithms Using the Minimum Cross-Entropy Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	<b>18.8075 (4)</b>	0.00	<b>18.8075 (4)</b>	0.00	<b>18.8075 (4)</b>	0.00	<b>18.8075 (4)</b>	0.00	<b>18.8075 (4)</b>	0.00	<b>18.8075 (4)</b>	0.00	<b>18.8075 (4)</b>	0.00
	4	22.1770 (6)	0.00	22.1267 (5)	0.19	<b>21.6894 (1)</b>	0.33	21.8711 (3)	0.37	22.1773 (7)	0.00	21.8497 (2)	0.37	21.8890 (4)	0.38
	6	16.6906 (5)	0.36	16.5746 (2)	0.49	16.9314 (7)	0.47	16.5793 (3)	0.36	16.7286 (6)	0.90	16.6615 (4)	0.59	<b>16.5172 (1)</b>	0.31
	8	13.9120 (6)	0.81	13.8393 (4)	1.01	<b>12.3188 (1)</b>	0.73	13.1437 (2)	0.81	13.8813 (5)	0.90	13.7714 (3)	1.03	14.4095 (7)	0.61
MRI2	2	<b>18.6679 (4)</b>	0.00	<b>18.6679 (4)</b>	0.00	<b>18.6679 (4)</b>	0.00	<b>18.6679 (4)</b>	0.00	<b>18.6679 (4)</b>	0.00	<b>18.6679 (4)</b>	0.00	<b>18.6679 (4)</b>	0.00
	4	<b>10.0320 (1)</b>	0.00	10.4825 (3)	0.49	11.0323 (7)	1.10	10.8852 (6)	0.34	10.2271 (2)	0.40	10.6859 (5)	0.47	10.5847 (4)	0.49
	6	8.5589 (2)	0.20	8.5943 (4)	0.49	8.8709 (6)	0.48	8.8552 (5)	0.50	8.5823 (3)	0.53	<b>8.5256 (1)</b>	0.41	8.8787 (7)	0.46
	8	7.0903 (6)	0.57	6.8136 (3)	0.52	6.9321 (4)	0.52	<b>6.5630 (1)</b>	0.47	7.1542 (7)	0.60	7.0305 (5)	0.57	6.6493 (2)	0.51
MRI3	2	<b>14.5763 (4)</b>	0.00	<b>14.5763 (4)</b>	0.00	<b>14.5763 (4)</b>	0.00	<b>14.5763 (4)</b>	0.00	<b>14.5763 (4)</b>	0.00	<b>14.5763 (4)</b>	0.00	<b>14.5763 (4)</b>	0.00
	4	17.6408 (5)	0.48	17.9087 (6)	0.46	17.4112 (3)	0.58	17.4615 (4)	0.92	18.0154 (7)	0.75	<b>17.2517 (1)</b>	0.60	17.3768 (2)	0.58
	6	<b>14.5579 (1)</b>	0.17	14.5866 (2)	0.28	15.0568 (7)	0.24	14.9339 (6)	0.41	14.7995 (5)	0.42	14.7145 (4)	0.43	14.5904 (3)	0.25
	8	13.4668 (3)	0.78	13.4467 (2)	0.77	14.0497 (5)	1.08	14.1287 (6)	0.67	14.2033 (7)	0.71	<b>13.3043 (1)</b>	0.90	13.8558 (4)	0.56
MRI4	2	<b>19.7873 (4)</b>	0.00	<b>19.7873 (4)</b>	0.00	<b>19.7873 (4)</b>	0.00	<b>19.7873 (4)</b>	0.00	<b>19.7873 (4)</b>	0.00	<b>19.7873 (4)</b>	0.00	<b>19.7873 (4)</b>	0.00
	4	23.6020 (2)	0.55	23.7152 (5)	0.00	23.6658 (3)	0.09	<b>23.5995 (1)</b>	0.08	23.7176 (6)	0.01	23.7214 (7)	0.05	23.7069 (4)	0.05
	6	18.6250 (6.5)	0.00	18.6250 (6.5)	0.00	<b>18.1005 (1)</b>	0.42	18.4243 (4)	0.47	18.4588 (5)	0.35	18.2616 (2)	0.53	18.3380 (3)	0.40
	8	<b>13.1761 (1)</b>	0.84	13.5536 (3)	1.16	15.0710 (7)	1.60	14.4973 (6)	1.58	13.7684 (5)	0.99	13.6007 (4)	1.70	13.3609 (2)	0.81
MRI5	2	<b>21.5962 (4)</b>	0.00	<b>21.5962 (4)</b>	0.00	<b>21.5962 (4)</b>	0.00	<b>21.5962 (4)</b>	0.00	<b>21.5962 (4)</b>	0.00	<b>21.5962 (4)</b>	0.00	<b>21.5962 (4)</b>	0.00
	4	16.5536 (2)	0.08	16.5626 (3)	0.07	16.5732 (4)	0.04	16.6810 (7)	0.55	16.5807 (5.5)	0.00	<b>16.5072 (1)</b>	0.12	16.5807 (5.5)	0.00
	6	<b>13.4603 (1)</b>	0.32	13.7202 (3)	0.10	14.0513 (7)	0.46	13.9181 (6)	0.41	13.8884 (5)	0.56	13.8790 (4)	0.54	13.6382 (2)	0.27
	8	<b>13.6289 (1)</b>	0.60	13.7269 (3)	0.63	13.7239 (2)	0.76	13.9994 (5)	0.79	14.4368 (7)	0.97	13.8006 (4)	0.67	14.1127 (6)	0.49
MRI6	2	<b>19.8033 (4)</b>	0.00	<b>19.8033 (4)</b>	0.00	<b>19.8033 (4)</b>	0.00	<b>19.8033 (4)</b>	0.00	<b>19.8033 (4)</b>	0.00	<b>19.8033 (4)</b>	0.00	<b>19.8033 (4)</b>	0.00
	4	12.6333 (3)	0.00	12.6333 (3)	0.00	12.7785 (6)	0.38	13.2078 (7)	1.21	12.6333 (3)	0.00	12.7468 (5)	0.49	<b>12.5719 (1)</b>	0.20
	6	11.6670 (5)	0.41	11.8339 (7)	0.00	11.3310 (2)	0.52	11.5660 (3)	0.31	11.6253 (4)	0.39	11.7028 (6)	0.32	<b>11.2639 (1)</b>	0.57
	8	11.7600 (7)	1.36	10.0941 (4)	1.45	9.8245 (2)	0.83	<b>9.3537 (1)</b>	0.94	11.1585 (6)	1.52	10.3974 (5)	1.50	9.9588 (3)	1.36
MRI7	2	<b>15.5863 (4)</b>	0.00	<b>15.5863 (4)</b>	0.00	<b>15.5863 (4)</b>	0.00	<b>15.5863 (4)</b>	0.00	<b>15.5863 (4)</b>	0.00	<b>15.5863 (4)</b>	0.00	<b>15.5863 (4)</b>	0.00
	4	10.9610 (5)	0.00	10.9610 (5)	0.00	10.9249 (2)	0.20	10.9922 (7)	0.15	10.9610 (5)	0.00	<b>10.9170 (1)</b>	0.23	10.9532 (3)	0.04
	6	9.2573 (6)	0.15	9.1016 (3)	0.14	9.3458 (7)	0.18	9.1940 (5)	0.40	<b>9.0425 (1)</b>	0.12	9.0596 (2)	0.20	9.1491 (4)	0.19
	8	8.3286 (2)	0.13	<b>8.2452 (1)</b>	0.24	8.7049 (7)	0.17	8.4345 (4)	0.32	8.4057 (3)	0.24	8.4836 (6)	0.26	8.4834 (5)	0.22
MRI8	2	<b>16.2836 (4)</b>	0.00	<b>16.2836 (4)</b>	0.00	<b>16.2836 (4)</b>	0.00	<b>16.2836 (4)</b>	0.00	<b>16.2836 (4)</b>	0.00	<b>16.2836 (4)</b>	0.00	<b>16.2836 (4)</b>	0.00
	4	11.1797 (2.5)	0.00	11.1797 (2.5)	0.00	11.2431 (7)	0.35	11.2163 (6)	0.20	11.1819 (4)	0.01	11.1913 (5)	0.17	<b>11.1699 (1)</b>	0.06
	6	10.1225 (4)	0.10	10.1420 (6)	0.05	9.8400 (2)	0.46	10.1371 (5)	0.15	10.1706 (7)	0.16	<b>9.7771 (1)</b>	0.40	9.9456 (3)	0.33
	8	<b>8.8158 (1)</b>	0.20	8.8206 (2)	0.11	9.2290 (7)	0.43	9.2237 (6)	0.49	8.8442 (4)	0.20	8.8220 (3)	0.22	8.8515 (5)	0.14
MRI9	2	<b>10.7207 (4)</b>	0.00	<b>10.7207 (4)</b>	0.00	<b>10.7207 (4)</b>	0.00	<b>10.7207 (4)</b>	0.00	<b>10.7207 (4)</b>	0.00	<b>10.7207 (4)</b>	0.00	<b>10.7207 (4)</b>	0.00
	4	9.3499 (4)	0.02	9.7127 (6.5)	0.68	<b>9.3326 (1)</b>	0.04	9.3946 (5)	0.29	9.7127 (6.5)	0.68	9.3357 (2)	0.12	9.3382 (3)	0.02
	6	7.9418 (3)	0.39	<b>7.5956 (1)</b>	0.50	8.0299 (6)	0.37	7.8576 (2)	0.24	8.6688 (7)	0.93	7.9881 (5)	0.33	7.9755 (4)	0.13
	8	7.4405 (5)	0.28	7.4493 (6)	0.32	7.2682 (2)	0.20	7.3448 (4)	0.38	7.5887 (7)	0.34	7.3311 (3)	0.27	<b>7.2416 (1)</b>	0.27
MRI10	2	<b>13.2830 (4)</b>	0.00	<b>13.2830 (4)</b>	0.00	<b>13.2830 (4)</b>	0.00	<b>13.2830 (4)</b>	0.00	<b>13.2830 (4)</b>	0.00	<b>13.2830 (4)</b>	0.00	<b>13.2830 (4)</b>	0.00
	4	<b>11.0146 (1)</b>	0.17	11.1431 (4)	0.31	11.4571 (6)	0.34	11.1923 (5)	0.41	11.0368 (2)	0.03	11.4824 (7)	0.46	11.1297 (3)	0.30
	6	9.8947 (4)	0.09	9.9641 (6)	0.25	9.8862 (2)	0.36	10.1061 (7)	0.43	9.9282 (5)	0.04	<b>9.8473 (1)</b>	0.12	9.8936 (3)	0.08
	8	8.4139 (5)	0.31	8.4355 (6)	0.29	<b>8.0493 (1)</b>	0.44	8.6544 (7)	0.42	8.3178 (3)	0.32	8.3784 (4)	0.29	8.2105 (2)	0.21
Av. Rank		3.65±1.73 (4)		3.94±1.51 (4)		4.12±2.1 (4)		4.47±1.66 (4)		4.75±1.59 (4.5)		3.6±1.66 (4)		<b>3.46±1.5 (4)</b>	

As can be seen from Table 6, the most successful algorithm is the JS regarding between-class variance fitness value, and the JS is followed by ABC, PSO, ALO, MFO, WOA, and GWO, respectively. As with the minimum cross-entropy method, as the threshold levels increase, the results of the algorithms differ in the between-class variance method. While all the algorithms find the same fitness value for 2-level thresholding, JS, ABC, and PSO algorithms converge to the best fitness values for higher threshold levels.

The results in Table 7 display the PSNR scores acquired from varying threshold levels among the compared algorithms. After analyzing the results, it was concluded that ABC performed the best, with JS following closely as the runner-up. PSO was ranked third, while GWO was found to be the least effective algorithm.

**Table 6.** Comparison of the Best Objective Value Computed of Swarm-Based Optimization Algorithms Using Between-Class Variance Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	<b>4485.95 (4)</b>	0.00	<b>4485.95 (4)</b>	0.00	<b>4485.95 (4)</b>	0.00	<b>4485.95 (4)</b>	0.00	<b>4485.95 (4)</b>	0.00	<b>4485.95 (4)</b>	0.00	<b>4485.95 (4)</b>	0.00
	4	4721.62 (3)	0.00	<b>4721.62 (1.5)</b>	0.00	4721.60 (6)	0.03	4721.58 (7)	0.05	<b>4721.62 (1.5)</b>	0.00	4721.61 (5)	0.03	4721.61 (4)	0.01
	6	<b>4766.89 (1)</b>	0.05	4766.38 (3)	1.64	4764.33 (7)	5.05	4765.29 (5)	4.76	4765.10 (6)	2.56	4766.08 (4)	1.84	4766.89 (2)	0.03
	8	<b>4789.15 (1)</b>	0.58	4788.94 (2)	0.97	4785.08 (7)	3.68	4787.70 (6)	4.41	4788.73 (3)	1.11	4788.30 (5)	1.18	4788.71 (4)	1.04
MRI2	2	<b>2320.92 (4)</b>	0.00	<b>2320.92 (4)</b>	0.00	<b>2320.92 (4)</b>	0.00	<b>2320.92 (4)</b>	0.00	<b>2320.92 (4)</b>	0.00	<b>2320.92 (4)</b>	0.00	<b>2320.92 (4)</b>	0.00
	4	2458.56 (3)	0.00	<b>2458.56 (1.5)</b>	0.00	2458.56 (6)	0.01	2458.56 (7)	0.01	<b>2458.56 (1.5)</b>	0.00	2458.56 (5)	0.00	2458.56 (4)	0.00
	6	2488.15 (3)	0.51	2487.87 (4)	1.29	<b>2488.38 (1)</b>	0.07	2487.38 (7)	3.12	2487.45 (6)	1.03	2487.83 (5)	0.93	2488.18 (2)	0.69
	8	2504.87 (2)	0.07	2504.70 (4)	0.56	2502.05 (7)	3.06	2503.32 (6)	2.84	2504.82 (3)	0.49	2504.49 (5)	0.46	<b>2504.88 (1)</b>	0.10
MRI3	2	<b>1099.11 (4)</b>	0.00	<b>1099.11 (4)</b>	0.00	<b>1099.11 (4)</b>	0.00	<b>1099.11 (4)</b>	0.00	<b>1099.11 (4)</b>	0.00	<b>1099.11 (4)</b>	0.00	<b>1099.11 (4)</b>	0.00
	4	<b>1197.88 (1)</b>	0.00	1197.79 (5)	0.20	1197.87 (3)	0.10	1197.85 (4)	0.14	1195.05 (7)	8.33	1197.69 (6)	0.27	1197.88 (2)	0.00
	6	1228.94 (3)	0.55	1227.16 (6)	3.17	1229.13 (2)	0.27	1228.40 (4)	2.74	1226.13 (7)	3.92	1227.98 (5)	2.18	<b>1229.13 (1)</b>	0.42
	8	1240.39 (2)	0.48	1240.15 (3)	1.62	1239.05 (5)	1.94	1238.82 (6)	2.76	1237.77 (7)	4.21	1239.85 (4)	2.70	<b>1241.03 (1)</b>	0.21
MRI4	2	<b>2690.52 (4)</b>	0.00	<b>2690.52 (4)</b>	0.00	<b>2690.52 (4)</b>	0.00	<b>2690.52 (4)</b>	0.00	<b>2690.52 (4)</b>	0.00	<b>2690.52 (4)</b>	0.00	<b>2690.52 (4)</b>	0.00
	4	<b>2997.98 (2)</b>	0.00	<b>2997.98 (2)</b>	0.00	2997.96 (6)	0.02	2997.94 (7)	0.04	2997.98 (4)	0.00	2997.97 (5)	0.02	<b>2997.98 (2)</b>	0.00
	6	3063.99 (2)	0.14	3063.67 (4)	1.30	3061.72 (7)	5.95	3063.80 (3)	0.93	3061.94 (6)	2.57	3062.81 (5)	2.21	<b>3064.00 (1)</b>	0.04
	8	3087.13 (2)	0.08	3086.84 (5)	1.34	3083.79 (7)	4.46	3085.45 (6)	2.74	3087.12 (3)	0.07	3087.06 (4)	0.19	<b>3087.14 (1)</b>	0.03
MRI5	2	<b>2806.54 (3.5)</b>	0.00	<b>2806.54 (3.5)</b>	0.00	<b>2806.54 (3.5)</b>	0.00	2806.54 (7)	0.00	<b>2806.54 (3.5)</b>	0.00	<b>2806.54 (3.5)</b>	0.00	<b>2806.54 (3.5)</b>	0.00
	4	2963.70 (3)	0.00	<b>2963.70 (1.5)</b>	0.00	2963.68 (6)	0.01	2961.73 (7)	10.72	<b>2963.70 (1.5)</b>	0.00	2963.68 (5)	0.02	2963.69 (4)	0.01
	6	3011.23 (4)	0.07	3011.28 (2)	0.02	3010.10 (7)	3.15	3010.13 (6)	4.41	3011.28 (3)	0.01	3011.04 (5)	0.47	<b>3011.28 (1)</b>	0.01
	8	<b>3031.20 (1)</b>	0.10	3030.70 (4)	1.35	3028.80 (7)	2.29	3030.27 (6)	2.31	3030.93 (3)	0.49	3030.36 (5)	1.94	3031.19 (2)	0.21
MRI6	2	<b>2876.23 (4)</b>	0.00	<b>2876.23 (4)</b>	0.00	<b>2876.23 (4)</b>	0.00	<b>2876.23 (4)</b>	0.00	<b>2876.23 (4)</b>	0.00	<b>2876.23 (4)</b>	0.00	<b>2876.23 (4)</b>	0.00
	4	<b>3018.08 (1)</b>	0.00	3018.08 (2)	0.00	3018.07 (6)	0.00	3018.07 (5)	0.00	3017.48 (7)	1.37	3018.07 (4)	0.00	3018.08 (3)	0.00
	6	<b>3067.90 (1)</b>	0.00	3067.56 (3)	0.68	3066.06 (7)	2.87	3067.31 (4)	1.29	3067.13 (6)	0.81	3067.29 (5)	0.98	3067.89 (2)	0.02
	8	<b>3084.59 (1)</b>	0.24	3084.36 (4)	0.62	3081.61 (7)	2.01	3083.13 (6)	1.57	3084.40 (3)	0.41	3083.96 (5)	0.68	3084.53 (2)	0.36
MRI7	2	<b>2407.76 (4)</b>	0.00	<b>2407.76 (4)</b>	0.00	<b>2407.76 (4)</b>	0.00	<b>2407.76 (4)</b>	0.00	<b>2407.76 (4)</b>	0.00	<b>2407.76 (4)</b>	0.00	<b>2407.76 (4)</b>	0.00
	4	2510.58 (5)	0.01	2510.58 (2)	0.00	2510.57 (7)	0.01	2510.58 (3)	0.00	<b>2510.58 (1)</b>	0.00	2510.57 (6)	0.01	2510.58 (4)	0.01
	6	2544.64 (2)	0.01	<b>2544.65 (1)</b>	0.00	2544.12 (6)	2.36	2542.91 (7)	4.48	2544.64 (3)	0.01	2544.60 (5)	0.07	2544.63 (4)	0.01
	8	<b>2555.33 (1)</b>	0.05	2555.20 (3)	0.83	2553.74 (6)	2.14	2553.20 (7)	3.16	2555.18 (4)	0.83	2555.11 (5)	0.82	2555.28 (2)	0.06
MRI8	2	<b>2717.90 (4)</b>	0.00	<b>2717.90 (4)</b>	0.00	<b>2717.90 (4)</b>	0.00	<b>2717.90 (4)</b>	0.00	<b>2717.90 (4)</b>	0.00	<b>2717.90 (4)</b>	0.00	<b>2717.90 (4)</b>	0.00
	4	2803.34 (2)	0.01	<b>2803.35 (1)</b>	0.00	2803.33 (3)	0.02	2802.99 (5)	1.25	2801.52 (7)	2.43	2802.84 (6)	1.51	2803.01 (4)	1.26
	6	2832.01 (2)	0.04	2831.97 (4)	0.28	2830.40 (7)	3.47	2830.78 (6)	3.20	2832.01 (3)	0.03	2831.70 (5)	0.64	<b>2832.02 (1)</b>	0.01
	8	2842.73 (3)	0.05	2842.69 (4)	0.29	2840.59 (6)	2.51	2840.10 (7)	2.66	2842.74 (2)	0.03	2842.22 (5)	1.27	<b>2842.75 (1)</b>	0.03
MRI9	2	<b>487.88 (4)</b>	0.00	<b>487.88 (4)</b>	0.00	<b>487.88 (4)</b>	0.00	<b>487.88 (4)</b>	0.00	<b>487.88 (4)</b>	0.00	<b>487.88 (4)</b>	0.00	<b>487.88 (4)</b>	0.00
	4	<b>525.59 (1)</b>	0.01	523.45 (6)	4.88	525.20 (4.5)	1.51	525.20 (4.5)	1.51	523.45 (7)	4.88	525.35 (3)	1.16	525.59 (2)	0.00
	6	536.51 (3)	0.98	535.19 (6)	2.53	535.80 (5)	1.15	536.59 (2)	0.50	533.51 (7)	4.02	536.22 (4)	1.20	<b>536.81 (1)</b>	0.01
	8	539.99 (2)	0.52	538.64 (6)	2.00	539.67 (5)	0.40	539.68 (4)	1.00	538.49 (7)	1.72	539.78 (3)	0.73	<b>540.24 (1)</b>	0.43
MRI10	2	<b>2976.97 (4)</b>	0.00	<b>2976.97 (4)</b>	0.00	<b>2976.97 (4)</b>	0.00	<b>2976.97 (4)</b>	0.00	<b>2976.97 (4)</b>	0.00	<b>2976.97 (4)</b>	0.00	<b>2976.97 (4)</b>	0.00
	4	<b>3162.65 (2.5)</b>	0.00	<b>3162.65 (2.5)</b>	0.00	3162.64 (6)	0.02	3162.65 (5)	0.01	<b>3162.65 (2.5)</b>	0.00	3162.64 (7)	0.02	<b>3162.65 (2.5)</b>	0.00
	6	3201.84 (4)	0.03	3201.87 (2)	0.01	3201.70 (6)	0.11	3200.26 (7)	4.88	<b>3201.87 (1)</b>	0.01	3201.83 (5)	0.07	3201.87 (3)	0.02
	8	3219.40 (5)	0.12	<b>3219.50 (1)</b>	0.04	3216.84 (7)	3.12	3218.06 (6)	2.92	3219.48 (3)	0.02	3219.43 (4)	0.10	3219.48 (2)	0.03
Av Rank		2.7±1.25 (3)		3.39±1.43 (4)		5.3±1.59 (6)		5.21±1.41 (5)		4.14±1.89 (4)		4.61±0.82 (5)		<b>2.65±1.26 (2.25)</b>	

Table 8 compares SSIM metric results of the algorithms. As can be seen from the table, GWO has the best average rank while ABC has the worst, surprisingly. The other algorithms' rankings based on their SSIM metric scores were WOA, MFO, PSO, JS, and ALO.

Lastly, Table 9 presents the NIQE results obtained from compared algorithms. As can be seen from the table, PSO has the best average rank, followed by ALO, GWO, JS, MFO, ABC, and WOA. The results obtained from the objective analysis show that there is a significant correlation between best fitness values and the PSNR metric.

**Table 7.** Comparison of PSNR Values of Swarm-Based Optimization Algorithms Using Between-Class Variance Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	<b>15.6116 (4)</b>	0.00	<b>15.6116 (4)</b>	0.00	<b>15.6116 (4)</b>	0.00	<b>15.6116 (4)</b>	0.00	<b>15.6116 (4)</b>	0.00	<b>15.6116 (4)</b>	0.00	<b>15.6116 (4)</b>	0.00
	4	21.3435 (3)	0.02	21.3389 (6.5)	0.00	21.3400 (5)	0.04	21.3455 (2)	0.04	21.3389 (6.5)	0.00	21.3404 (4)	0.02	<b>21.3538 (1)</b>	0.03
	6	<b>24.4712 (1)</b>	0.02	24.3868 (3)	0.25	24.0801 (7)	0.54	24.2975 (5)	0.39	24.1730 (6)	0.42	24.3350 (4)	0.28	24.4588 (2)	0.03
	8	<b>26.4686 (1)</b>	0.10	26.4441 (2)	0.12	25.9543 (7)	0.42	26.2828 (6)	0.43	26.4042 (3)	0.14	26.3646 (5)	0.15	26.3931 (4)	0.15
MRI2	2	<b>18.0510 (4)</b>	0.00	<b>18.0510 (4)</b>	0.00	<b>18.0510 (4)</b>	0.00	<b>18.0510 (4)</b>	0.00	<b>18.0510 (4)</b>	0.00	<b>18.0510 (4)</b>	0.00	<b>18.0510 (4)</b>	0.00
	4	22.6375 (7)	0.01	22.6384 (4.5)	0.00	22.6399 (3)	0.01	<b>22.6480 (1)</b>	0.02	22.6384 (4.5)	0.00	22.6425 (2)	0.02	22.6383 (6)	0.01
	6	<b>24.7767 (1)</b>	0.11	24.7061 (5)	0.16	24.7600 (2)	0.03	24.6897 (6)	0.20	24.6280 (7)	0.14	24.7150 (4)	0.13	24.7509 (3)	0.11
	8	26.1470 (4)	0.03	26.1657 (2)	0.05	26.0497 (7)	0.26	26.0654 (6)	0.17	26.1512 (3)	0.04	<b>26.1753 (1)</b>	0.05	26.1442 (5)	0.03
MRI3	2	<b>19.0335 (4)</b>	0.00	<b>19.0335 (4)</b>	0.00	<b>19.0335 (4)</b>	0.00	<b>19.0335 (4)</b>	0.00	<b>19.0335 (4)</b>	0.00	<b>19.0335 (4)</b>	0.00	<b>19.0335 (4)</b>	0.00
	4	23.7497 (7)	0.00	23.8767 (2)	0.29	23.7751 (4)	0.14	23.8038 (3)	0.21	23.7670 (5)	0.47	<b>23.9817 (1)</b>	0.36	23.7498 (6)	0.00
	6	27.0210 (4)	0.36	26.5550 (6)	0.52	27.1145 (2)	0.16	27.0614 (3)	0.41	26.4759 (7)	0.61	26.6671 (5)	0.48	<b>27.1246 (1)</b>	0.26
	8	28.6660 (2)	0.21	28.6532 (3)	0.35	28.6034 (5)	0.35	28.5395 (6)	0.39	28.2388 (7)	0.79	28.6237 (4)	0.53	<b>28.8731 (1)</b>	0.10
MRI4	2	<b>15.7430 (4)</b>	0.00	<b>15.7430 (4)</b>	0.00	<b>15.7430 (4)</b>	0.00	<b>15.7430 (4)</b>	0.00	<b>15.7430 (4)</b>	0.00	<b>15.7430 (4)</b>	0.00	<b>15.7430 (4)</b>	0.00
	4	<b>20.1590 (2)</b>	0.00	<b>20.1590 (2)</b>	0.00	20.1493 (6)	0.01	20.1368 (7)	0.03	20.1584 (4)	0.00	20.1537 (5)	0.01	<b>20.1590 (2)</b>	0.00
	6	23.3243 (2)	0.05	23.2853 (4)	0.21	23.1195 (6)	0.42	23.3042 (3)	0.16	22.9829 (7)	0.39	23.1457 (5)	0.34	<b>23.3355 (1)</b>	0.04
	8	25.1495 (2)	0.15	25.0665 (5)	0.22	24.6788 (7)	0.39	24.9732 (6)	0.34	25.1111 (3)	0.14	25.0800 (4)	0.15	<b>25.1673 (1)</b>	0.12
MRI5	2	16.7743 (4.5)	0.00	16.7743 (4.5)	0.00	16.7743 (4.5)	0.00	<b>16.7947 (1)</b>	0.05	16.7743 (4.5)	0.00	16.7743 (4.5)	0.00	16.7743 (4.5)	0.00
	4	22.0478 (3)	0.00	22.0469 (4.5)	0.00	22.0529 (2)	0.02	21.9634 (7)	0.42	22.0469 (4.5)	0.00	22.0426 (6)	0.01	<b>22.0609 (1)</b>	0.01
	6	<b>24.9289 (1)</b>	0.04	24.9221 (4)	0.03	24.7740 (7)	0.23	24.8399 (6)	0.31	24.8915 (5)	0.02	24.9240 (2)	0.06	24.9231 (3)	0.02
	8	27.1206 (3)	0.05	<b>27.1299 (1)</b>	0.17	26.8581 (7)	0.31	27.0185 (6)	0.25	27.1245 (2)	0.06	27.0200 (5)	0.23	27.0839 (4)	0.05
MRI6	2	<b>17.7496 (4)</b>	0.00	<b>17.7496 (4)</b>	0.00	<b>17.7496 (4)</b>	0.00	<b>17.7496 (4)</b>	0.00	<b>17.7496 (4)</b>	0.00	<b>17.7496 (4)</b>	0.00	<b>17.7496 (4)</b>	0.00
	4	<b>22.3002 (1)</b>	0.00	22.2950 (2)	0.02	22.2270 (6)	0.03	22.2293 (5)	0.02	22.1587 (7)	0.28	22.2507 (4)	0.04	22.2924 (3)	0.02
	6	25.1784 (3)	0.02	25.1659 (4)	0.03	24.9281 (7)	0.30	25.0872 (6)	0.07	<b>25.1882 (1)</b>	0.08	25.1183 (5)	0.09	25.1855 (2)	0.05
	8	<b>27.4799 (1)</b>	0.07	27.4086 (3)	0.13	26.8837 (7)	0.35	27.1698 (6)	0.28	27.3637 (4)	0.09	27.3354 (5)	0.16	27.4700 (2)	0.09
MRI7	2	<b>17.6542 (4)</b>	0.00	<b>17.6542 (4)</b>	0.00	<b>17.6542 (4)</b>	0.00	<b>17.6542 (4)</b>	0.00	<b>17.6542 (4)</b>	0.00	<b>17.6542 (4)</b>	0.00	<b>17.6542 (4)</b>	0.00
	4	<b>21.7429 (1)</b>	0.01	21.7395 (2)	0.00	21.7340 (7)	0.01	21.7371 (6)	0.01	21.7388 (3)	0.00	21.7379 (4)	0.01	21.7374 (5)	0.01
	6	23.6104 (3)	0.02	<b>23.6171 (1)</b>	0.01	23.5873 (6)	0.12	23.5670 (7)	0.24	23.6056 (5)	0.01	23.6125 (2)	0.02	23.6070 (4)	0.02
	8	24.4896 (6)	0.04	24.4947 (5)	0.04	24.5452 (2)	0.23	<b>24.7809 (1)</b>	0.69	24.5128 (4)	0.04	24.4890 (7)	0.04	24.5370 (3)	0.02
MRI8	2	<b>17.6676 (4)</b>	0.00	<b>17.6676 (4)</b>	0.00	<b>17.6676 (4)</b>	0.00	<b>17.6676 (4)</b>	0.00	<b>17.6676 (4)</b>	0.00	<b>17.6676 (4)</b>	0.00	<b>17.6676 (4)</b>	0.00
	4	21.3700 (2)	0.01	21.3698 (3)	0.00	21.3523 (6)	0.02	21.3462 (7)	0.03	<b>21.3883 (1)</b>	0.03	21.3641 (5)	0.03	21.3667 (4)	0.02
	6	23.4601 (2)	0.01	<b>23.4617 (1)</b>	0.03	23.3590 (7)	0.23	23.4329 (6)	0.15	23.4447 (4)	0.03	23.4392 (5)	0.05	23.4581 (3)	0.02
	8	24.3569 (2)	0.02	24.3529 (3)	0.02	24.3218 (7)	0.12	<b>24.4183 (1)</b>	0.58	24.3285 (6)	0.02	24.3409 (5)	0.07	24.3496 (4)	0.02
MRI9	2	<b>26.7904 (4)</b>	0.00	<b>26.7904 (4)</b>	0.00	<b>26.7904 (4)</b>	0.00	<b>26.7904 (4)</b>	0.00	<b>26.7904 (4)</b>	0.00	<b>26.7904 (4)</b>	0.00	<b>26.7904 (4)</b>	0.00
	4	30.2211 (2)	0.00	30.0000 (6)	0.50	30.1132 (4.5)	0.41	30.1132 (4.5)	0.41	29.9996 (7)	0.50	30.1764 (3)	0.15	<b>30.2216 (1)</b>	0.00
	6	33.4431 (2)	0.39	32.9065 (5)	0.86	32.7985 (6)	0.47	33.3789 (3)	0.36	32.3833 (7)	1.18	33.2089 (4)	0.51	<b>33.5528 (1)</b>	0.06
	8	35.4188 (2)	0.38	34.6197 (6)	1.08	35.0520 (5)	0.30	35.1217 (4)	0.69	34.4655 (7)	0.92	35.2031 (3)	0.56	<b>35.6011 (1)</b>	0.32
MRI10	2	<b>17.8175 (4)</b>	0.00	<b>17.8175 (4)</b>	0.00	<b>17.8175 (4)</b>	0.00	<b>17.8175 (4)</b>	0.00	<b>17.8175 (4)</b>	0.00	<b>17.8175 (4)</b>	0.00	<b>17.8175 (4)</b>	0.00
	4	21.1933 (4.5)	0.00	21.1933 (4.5)	0.00	21.1896 (7)	0.02	21.1959 (2)	0.01	21.1933 (4.5)	0.00	<b>21.1993 (1)</b>	0.02	21.1933 (4.5)	0.00
	6	23.8599 (3)	0.01	23.8642 (2)	0.00	23.8309 (6)	0.04	23.7615 (7)	0.32	23.8590 (4)	0.00	23.8582 (5)	0.02	<b>23.8642 (1)</b>	0.01
	8	25.0718 (4)	0.02	25.0775 (3)	0.01	24.9930 (7)	0.19	<b>25.0869 (1)</b>	0.21	25.0678 (5)	0.01	25.0662 (6)	0.03	25.0777 (2)	0.02
Av Rank		<b>3.05±1.56 (3)</b>		3.64±1.42 (4)		5.2±1.66 (5.5)		4.41±1.9 (4)		4.61±1.62 (4)		4.04±1.33 (4)		3.05±1.51 (3.5)	

The data presented in the tables demonstrates that the performance of algorithms varies significantly depending on the type of fitness function employed. This variation occurs due to the distinct behavior that each algorithm exhibits across diverse fitness landscapes. Since each fitness function has its unique characteristics, it results in a diverse landscape that can impact the algorithm's performance based on its exploration/exploitation ability. It is thus crucial to consider the specific characteristics of each fitness function when selecting an algorithm for a given task. This approach will ensure that the most suitable algorithm is used, leading to better results and improved efficiency.



**Table 8.** Comparison of SSIM Values of Swarm-Based Optimization Algorithms Using Between-Class Variance Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	<b>0.4537 (4)</b>	0.00	<b>0.4537 (4)</b>	0.00	<b>0.4537 (4)</b>	0.00	<b>0.4537 (4)</b>	0.00	<b>0.4537 (4)</b>	0.00	<b>0.4537 (4)</b>	0.00	<b>0.4537 (4)</b>	0.00
	4	0.6024 (4)	0.00	0.6023 (6.5)	0.00	0.6024 (2)	0.00	0.6024 (5)	0.00	0.6023 (6.5)	0.00	0.6024 (3)	0.00	<b>0.6025 (1)</b>	0.00
	6	0.6551 (7)	0.00	0.6576 (5)	0.01	<b>0.6668 (1)</b>	0.02	0.6603 (3)	0.02	0.6637 (2)	0.01	0.6590 (4)	0.01	0.6552 (6)	0.00
	8	0.7168 (3)	0.00	0.7153 (5)	0.01	<b>0.7619 (1)</b>	0.02	0.7230 (2)	0.02	0.7138 (6)	0.01	0.7112 (7)	0.01	0.7162 (4)	0.01
MRI2	2	<b>0.2604 (4)</b>	0.00	<b>0.2604 (4)</b>	0.00	<b>0.2604 (4)</b>	0.00	<b>0.2604 (4)</b>	0.00	<b>0.2604 (4)</b>	0.00	<b>0.2604 (4)</b>	0.00	<b>0.2604 (4)</b>	0.00
	4	0.3787 (6)	0.00	0.3787 (3.5)	0.00	0.3786 (7)	0.00	<b>0.3787 (1)</b>	0.00	0.3787 (3.5)	0.00	0.3787 (2)	0.00	0.3787 (5)	0.00
	6	0.4348 (3)	0.01	0.4311 (5)	0.01	<b>0.4401 (1)</b>	0.00	0.4377 (2)	0.02	0.4232 (7)	0.02	0.4300 (6)	0.01	0.4340 (4)	0.01
	8	0.4537 (7)	0.00	0.4570 (4)	0.00	<b>0.4887 (1)</b>	0.03	0.4719 (2)	0.03	0.4557 (5)	0.00	0.4609 (3)	0.01	0.4548 (6)	0.00
MRI3	2	<b>0.5710 (4)</b>	0.00	<b>0.5710 (4)</b>	0.00	<b>0.5710 (4)</b>	0.00	<b>0.5710 (4)</b>	0.00	<b>0.5710 (4)</b>	0.00	<b>0.5710 (4)</b>	0.00	<b>0.5710 (4)</b>	0.00
	4	0.6526 (5)	0.00	0.6552 (2)	0.01	0.6532 (4)	0.00	0.6538 (3)	0.00	0.6514 (7)	0.01	<b>0.6581 (1)</b>	0.01	0.6526 (6)	0.00
	6	0.7215 (4)	0.01	0.7094 (6)	0.01	0.7278 (2)	0.01	<b>0.7285 (1)</b>	0.00	0.7085 (7)	0.02	0.7124 (5)	0.01	0.7250 (3)	0.01
	8	0.7470 (6)	0.01	0.7489 (5)	0.01	<b>0.7734 (1)</b>	0.04	0.7661 (2)	0.03	0.7423 (7)	0.02	0.7504 (4)	0.01	0.7535 (3)	0.00
MRI4	2	<b>0.3540 (4)</b>	0.00	<b>0.3540 (4)</b>	0.00	<b>0.3540 (4)</b>	0.00	<b>0.3540 (4)</b>	0.00	<b>0.3540 (4)</b>	0.00	<b>0.3540 (4)</b>	0.00	<b>0.3540 (4)</b>	0.00
	4	0.5535 (6)	0.00	0.5535 (6)	0.00	<b>0.5536 (1)</b>	0.00	0.5535 (4)	0.00	0.5535 (3)	0.00	0.5536 (2)	0.00	0.5535 (6)	0.00
	6	<b>0.6828 (1)</b>	0.00	0.6797 (4)	0.01	0.6780 (5)	0.02	0.6815 (3)	0.01	0.6620 (7)	0.03	0.6708 (6)	0.02	0.6826 (2)	0.00
	8	0.7202 (6)	0.01	0.7216 (4)	0.01	<b>0.7337 (1)</b>	0.01	0.7282 (2)	0.02	0.7212 (5)	0.01	0.7222 (3)	0.01	0.7191 (7)	0.00
MRI5	2	0.5216 (4.5)	0.00	0.5216 (4.5)	0.00	0.5216 (4.5)	0.00	<b>0.5221 (1)</b>	0.00	0.5216 (4.5)	0.00	0.5216 (4.5)	0.00	0.5216 (4.5)	0.00
	4	0.6298 (5)	0.00	0.6298 (6.5)	0.00	<b>0.6312 (1)</b>	0.00	0.6304 (3)	0.00	0.6298 (6.5)	0.00	0.6310 (2)	0.00	0.6299 (4)	0.00
	6	0.6726 (7)	0.00	0.6729 (4)	0.00	0.6759 (2)	0.01	<b>0.6775 (1)</b>	0.02	0.6728 (5)	0.00	0.6751 (3)	0.00	0.6727 (6)	0.00
	8	0.7101 (7)	0.00	0.7140 (3)	0.01	<b>0.7274 (1)</b>	0.03	0.7196 (2)	0.02	0.7137 (4)	0.01	0.7129 (5)	0.02	0.7104 (6)	0.00
MRI6	2	<b>0.6140 (4)</b>	0.00	<b>0.6140 (4)</b>	0.00	<b>0.6140 (4)</b>	0.00	<b>0.6140 (4)</b>	0.00	<b>0.6140 (4)</b>	0.00	<b>0.6140 (4)</b>	0.00	<b>0.6140 (4)</b>	0.00
	4	0.7201 (7)	0.00	0.7203 (6)	0.00	0.7228 (3)	0.00	0.7228 (2)	0.00	<b>0.7238 (1)</b>	0.01	0.7220 (4)	0.00	0.7204 (5)	0.00
	6	0.7606 (6)	0.00	0.7663 (4)	0.01	<b>0.7749 (1)</b>	0.02	0.7642 (5)	0.01	0.7736 (2)	0.02	0.7690 (3)	0.02	0.7606 (7)	0.00
	8	0.7989 (7)	0.00	0.8033 (4)	0.01	<b>0.8272 (1)</b>	0.02	0.8108 (2)	0.01	0.8018 (5)	0.00	0.8075 (3)	0.01	0.8006 (6)	0.00
MRI7	2	<b>0.3208 (4)</b>	0.00	<b>0.3208 (4)</b>	0.00	<b>0.3208 (4)</b>	0.00	<b>0.3208 (4)</b>	0.00	<b>0.3208 (4)</b>	0.00	<b>0.3208 (4)</b>	0.00	<b>0.3208 (4)</b>	0.00
	4	0.3801 (5)	0.00	<b>0.3801 (1)</b>	0.00	0.3801 (7)	0.00	0.3801 (3)	0.00	0.3801 (2)	0.00	0.3801 (6)	0.00	0.3801 (4)	0.00
	6	0.4078 (6)	0.00	0.4081 (4)	0.00	0.4094 (2)	0.01	<b>0.4229 (1)</b>	0.06	0.4080 (5)	0.00	0.4083 (3)	0.00	0.4074 (7)	0.00
	8	0.4313 (7)	0.01	0.4327 (5)	0.01	0.4572 (2)	0.05	<b>0.5018 (1)</b>	0.13	0.4345 (3)	0.01	0.4333 (4)	0.01	0.4320 (6)	0.00
MRI8	2	<b>0.2979 (4)</b>	0.00	<b>0.2979 (4)</b>	0.00	<b>0.2979 (4)</b>	0.00	<b>0.2979 (4)</b>	0.00	<b>0.2979 (4)</b>	0.00	<b>0.2979 (4)</b>	0.00	<b>0.2979 (4)</b>	0.00
	4	0.3612 (5)	0.00	0.3612 (4)	0.00	0.3608 (6)	0.00	0.3608 (7)	0.00	<b>0.3632 (1)</b>	0.00	0.3615 (2)	0.00	0.3614 (3)	0.00
	6	0.3996 (3)	0.00	0.3992 (6)	0.00	0.4121 (2)	0.05	<b>0.4171 (1)</b>	0.05	0.3995 (4)	0.00	0.3971 (7)	0.01	0.3995 (5)	0.00
	8	0.4151 (6)	0.00	0.4166 (4)	0.00	0.4472 (2)	0.05	<b>0.4676 (1)</b>	0.10	0.4156 (5)	0.00	0.4197 (3)	0.01	0.4151 (7)	0.00
MRI9	2	<b>0.8608 (4)</b>	0.00	<b>0.8608 (4)</b>	0.00	<b>0.8608 (4)</b>	0.00	<b>0.8608 (4)</b>	0.00	<b>0.8608 (4)</b>	0.00	<b>0.8608 (4)</b>	0.00	<b>0.8608 (4)</b>	0.00
	4	0.9212 (2)	0.00	0.9186 (4)	0.01	0.9179 (6.5)	0.01	0.9179 (6.5)	0.01	0.9186 (5)	0.01	0.9207 (3)	0.00	<b>0.9213 (1)</b>	0.00
	6	0.9606 (2)	0.00	0.9539 (5)	0.01	0.9508 (6)	0.01	0.9589 (3)	0.01	0.9479 (7)	0.01	0.9573 (4)	0.01	<b>0.9617 (1)</b>	0.00
	8	0.9743 (2)	0.00	0.9686 (7)	0.01	0.9740 (3)	0.00	0.9714 (5)	0.01	0.9690 (6)	0.01	0.9724 (4)	0.00	<b>0.9754 (1)</b>	0.00
MRI10	2	<b>0.2939 (4)</b>	0.00	<b>0.2939 (4)</b>	0.00	<b>0.2939 (4)</b>	0.00	<b>0.2939 (4)</b>	0.00	<b>0.2939 (4)</b>	0.00	<b>0.2939 (4)</b>	0.00	<b>0.2939 (4)</b>	0.00
	4	0.3480 (4.5)	0.00	0.3480 (4.5)	0.00	0.3479 (7)	0.00	0.3480 (2)	0.00	0.3480 (4.5)	0.00	<b>0.3480 (1)</b>	0.00	0.3480 (4.5)	0.00
	6	0.3916 (7)	0.00	0.3922 (4)	0.00	0.3954 (2)	0.00	<b>0.4059 (1)</b>	0.06	0.3921 (6)	0.00	0.3928 (3)	0.00	0.3921 (5)	0.00
	8	0.4180 (7)	0.00	0.4187 (4)	0.00	0.4468 (2)	0.05	<b>0.4474 (1)</b>	0.08	0.4190 (3)	0.00	0.4186 (5)	0.00	0.4183 (6)	0.00
Av Rank		4.85±1.67 (4.75)		4.41±1.15 (4)		<b>3.1±1.94 (2.25)</b>		2.86±1.59 (3)		4.54±1.66 (4.25)		3.79±1.39 (4)		4.45±1.69 (4)	

### Statistical Analysis

In the third part of the experiments, PSO, ABC, GWO, MFO, ALO, WOA, and JS algorithms are compared on ten brain MR images using statistical tests. Since the randomness of the algorithms can affect their performance, each experiment is repeated 30 times with different random seeds, and the Mann-Whitney U test is applied to analyze their fitness values.

**Table 9.** Comparison of NIQE Values of Swarm-Based Optimization Algorithms Using Between-Class Variance Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	<b>21.2081 (4)</b>	0.00	<b>21.2081 (4)</b>	0.00	<b>21.2081 (4)</b>	0.00	<b>21.2081 (4)</b>	0.00	<b>21.2081 (4)</b>	0.00	<b>21.2081 (4)</b>	0.00	<b>21.2081 (4)</b>	0.00
	4	21.2356 (4)	0.07	<b>21.2181 (1.5)</b>	0.00	21.2414 (5)	0.10	21.2717 (7)	0.23	<b>21.2181 (1.5)</b>	0.00	21.2231 (3)	0.15	21.2619 (6)	0.10
	6	17.5876 (5)	0.09	17.3452 (2)	0.58	17.5318 (4)	1.03	17.7325 (7)	0.59	<b>16.9066 (1)</b>	1.01	17.4528 (3)	0.80	17.6060 (6)	0.06
	8	14.0716 (2)	0.44	14.2975 (4)	0.66	<b>13.8657 (1)</b>	0.29	14.1532 (3)	0.58	14.4390 (6)	0.73	14.6523 (7)	0.77	14.4306 (5)	0.71
MRI2	2	<b>15.2363 (4)</b>	0.00	<b>15.2363 (4)</b>	0.00	<b>15.2363 (4)</b>	0.00	<b>15.2363 (4)</b>	0.00	<b>15.2363 (4)</b>	0.00	<b>15.2363 (4)</b>	0.00	<b>15.2363 (4)</b>	0.00
	4	10.8733 (3)	0.00	<b>10.8733 (1.5)</b>	0.00	10.8836 (5)	0.06	10.9019 (7)	0.07	<b>10.8733 (1.5)</b>	0.00	10.8913 (6)	0.07	10.8739 (4)	0.00
	6	8.9155 (5)	0.52	8.8674 (4)	0.60	<b>8.5750 (1)</b>	0.12	8.7634 (3)	0.45	9.1141 (7)	0.55	8.9291 (6)	0.67	8.6842 (2)	0.47
	8	9.0110 (7)	0.20	8.6051 (3)	0.62	8.5637 (2)	1.08	8.8442 (6)	0.41	8.8353 (5)	0.31	<b>8.2152 (1)</b>	0.66	8.8030 (4)	0.27
MRI3	2	<b>13.4194 (4)</b>	0.00	<b>13.4194 (4)</b>	0.00	<b>13.4194 (4)</b>	0.00	<b>13.4194 (4)</b>	0.00	<b>13.4194 (4)</b>	0.00	<b>13.4194 (4)</b>	0.00	<b>13.4194 (4)</b>	0.00
	4	17.3680 (6)	0.02	<b>17.2992 (1)</b>	0.23	17.3583 (4)	0.07	17.3522 (3)	0.07	17.5649 (7)	0.80	17.3267 (2)	0.26	17.3645 (5)	0.03
	6	14.3002 (3)	0.63	14.9687 (5)	0.77	14.3877 (4)	0.39	14.2206 (2)	0.68	15.1224 (6)	0.93	15.1909 (7)	1.10	<b>14.2037 (1)</b>	<b>0.59</b>
	8	12.1432 (4)	0.84	11.9230 (2)	0.87	12.6953 (7)	1.03	12.4391 (5)	1.21	12.6309 (6)	1.43	11.9469 (3)	1.16	<b>11.5643 (1)</b>	<b>0.39</b>
MRI4	2	<b>25.0862 (4)</b>	0.00	<b>25.0862 (4)</b>	0.00	<b>25.0862 (4)</b>	0.00	<b>25.0862 (4)</b>	0.00	<b>25.0862 (4)</b>	0.00	<b>25.0862 (4)</b>	0.00	<b>25.0862 (4)</b>	<b>0.00</b>
	4	24.2593 (4)	0.00	24.2593 (4)	0.00	24.3224 (7)	0.07	<b>22.4736 (1)</b>	2.08	24.2629 (6)	0.02	23.9954 (2)	1.05	24.2593 (4)	0.00
	6	18.7888 (6)	0.16	18.6749 (3)	0.50	18.7931 (7)	0.86	18.7609 (5)	0.44	<b>18.0345 (1)</b>	1.02	18.3507 (2)	0.87	18.7560 (4)	0.11
	8	13.7054 (5)	1.29	13.4737 (3)	1.45	13.3548 (2)	1.68	14.3580 (7)	1.29	13.6477 (4)	1.23	<b>13.3156 (1)</b>	1.25	13.8451 (6)	1.07
MRI5	2	19.1506 (4.5)	0.00	19.1506 (4.5)	0.00	19.1506 (4.5)	0.00	<b>18.7490 (1)</b>	0.91	19.1506 (4.5)	0.00	19.1506 (4.5)	0.00	19.1506 (4.5)	0.00
	4	16.1328 (4)	0.01	16.1345 (5.5)	0.00	<b>15.9234 (1)</b>	0.23	16.1468 (7)	0.17	16.1345 (5.5)	0.00	15.9945 (2)	0.24	16.1004 (3)	0.04
	6	15.9617 (7)	0.24	15.8573 (6)	0.15	15.8075 (4)	0.70	15.7555 (3)	0.35	15.7501 (2)	0.09	<b>15.5528 (1)</b>	0.62	15.8441 (5)	0.05
	8	13.4312 (7)	0.42	<b>12.8870 (1)</b>	0.52	12.9886 (4)	0.71	12.9085 (2)	0.43	12.9821 (3)	0.34	13.2050 (5)	0.59	13.2796 (6)	0.46
MRI6	2	<b>20.5786 (4)</b>	0.00	<b>20.5786 (4)</b>	0.00	<b>20.5786 (4)</b>	0.00	<b>20.5786 (4)</b>	0.00	<b>20.5786 (4)</b>	0.00	<b>20.5786 (4)</b>	0.00	<b>20.5786 (4)</b>	0.00
	4	<b>12.9651 (1)</b>	0.00	13.0305 (2)	0.25	13.8314 (6)	0.30	13.8422 (7)	0.30	13.5975 (5)	0.99	13.5864 (4)	0.48	13.0632 (3)	0.30
	6	14.2514 (7)	0.05	13.7896 (4)	1.03	13.2469 (2)	1.73	14.1096 (6)	1.38	<b>12.7555 (1)</b>	1.21	13.3422 (3)	1.17	14.0806 (5)	0.57
	8	14.3462 (6)	0.88	13.8621 (4)	1.61	<b>12.2415 (1)</b>	1.14	12.5439 (2)	1.81	14.0717 (5)	1.03	13.5710 (3)	1.65	14.4006 (7)	1.05
MRI7	2	<b>14.6675 (4)</b>	0.00	<b>14.6675 (4)</b>	0.00	<b>14.6675 (4)</b>	0.00	<b>14.6675 (4)</b>	0.00	<b>14.6675 (4)</b>	0.00	<b>14.6675 (4)</b>	0.00	<b>14.6675 (4)</b>	0.00
	4	<b>10.0330 (1)</b>	0.04	10.0421 (4)	0.01	10.0829 (7)	0.18	10.0636 (6)	0.12	10.0411 (3)	0.00	10.0601 (5)	0.13	10.0335 (2)	0.03
	6	9.1112 (4)	0.11	9.0810 (2)	0.04	9.1528 (6)	0.09	9.1612 (7)	0.27	<b>9.0658 (1)</b>	0.01	9.1222 (5)	0.08	9.0836 (3)	0.07
	8	8.6253 (4)	0.16	8.7620 (7)	0.17	8.5820 (2)	0.27	8.7401 (6)	0.19	<b>8.5655 (1)</b>	0.23	8.6761 (5)	0.21	8.6082 (3)	0.10
MRI8	2	<b>14.9483 (4)</b>	0.00	<b>14.9483 (4)</b>	0.00	<b>14.9483 (4)</b>	0.00	<b>14.9483 (4)</b>	0.00	<b>14.9483 (4)</b>	0.00	<b>14.9483 (4)</b>	0.00	<b>14.9483 (4)</b>	0.00
	4	10.9603 (2)	0.17	<b>10.9281 (1)</b>	0.00	11.1307 (4)	0.43	11.5714 (7)	0.55	11.3839 (6)	0.61	11.2105 (5)	0.48	11.1074 (3)	0.41
	6	10.4503 (3)	0.19	10.4880 (5)	0.04	<b>10.3906 (1)</b>	0.62	10.5022 (7)	0.38	10.4485 (2)	0.11	10.4904 (6)	0.17	10.4700 (4)	0.03
	8	9.2952 (2)	0.15	<b>9.2870 (1)</b>	0.15	9.3406 (4)	0.25	9.4011 (7)	0.38	9.3057 (3)	0.07	9.3536 (6)	0.18	9.3496 (5)	0.07
MRI9	2	<b>12.3297 (4)</b>	0.00	<b>12.3297 (4)</b>	0.00	<b>12.3297 (4)</b>	0.00	<b>12.3297 (4)</b>	0.00	<b>12.3297 (4)</b>	0.00	<b>12.3297 (4)</b>	0.00	<b>12.3297 (4)</b>	0.00
	4	10.5058 (3)	0.01	10.4803 (2)	0.05	10.6089 (6.5)	0.40	10.6089 (6.5)	0.40	<b>10.4796 (1)</b>	0.05	10.5561 (5)	0.23	10.5067 (4)	0.02
	6	8.2523 (4)	0.55	8.3172 (5)	0.72	8.5990 (6)	0.53	7.9635 (2)	0.51	8.6672 (7)	0.95	8.1972 (3)	0.53	<b>7.7472 (1)</b>	0.30
	8	7.7824 (4)	0.21	7.8376 (6)	0.32	7.7107 (2)	0.31	<b>7.7085 (1)</b>	0.34	7.8231 (5)	0.25	7.8498 (7)	0.24	7.7787 (3)	0.20
MRI10	2	<b>13.1825 (4)</b>	0.00	<b>13.1825 (4)</b>	0.00	<b>13.1825 (4)</b>	0.00	<b>13.1825 (4)</b>	0.00	<b>13.1825 (4)</b>	0.00	<b>13.1825 (4)</b>	0.00	<b>13.1825 (4)</b>	0.00
	4	11.5423 (5.5)	0.00	11.5423 (5.5)	0.00	<b>11.5261 (1)</b>	0.10	11.5421 (3)	0.00	11.5423 (5.5)	0.00	11.5399 (2)	0.02	11.5423 (5.5)	0.00
	6	9.5368 (4)	0.15	9.4989 (3)	0.07	9.9460 (7)	0.25	9.6231 (6)	0.29	<b>9.4638 (1)</b>	0.04	9.6114 (5)	0.26	9.4976 (2)	0.11
	8	9.5933 (6)	0.25	9.4862 (3)	0.17	<b>9.2972 (1)</b>	0.50	9.6256 (7)	0.21	9.4223 (2)	0.13	9.5232 (4)	0.21	9.5751 (5)	0.15
Av Rank		4.22±1.51 (4)		<b>3.54±1.52 (4)</b>		3.88±1.95 (4)		4.64±2.02 (4)		3.79±1.92 (4)		3.99±1.63 (4)		3.95±1.42 (4)	

The Mann-Whitney U test was employed in this study due to its robustness and suitability for analyzing non-parametric data (Mann & Whitney, 1947). By utilizing this test, we aimed to accurately evaluate any differences or similarities between the groups under investigation without making assumptions about the underlying distribution of the data, thus ensuring the validity and reliability of our findings. Tables 10 and 11 present Mann-Whitney U-tests result ( $p < 0.05$ ) for each optimization algorithm and its corresponding Vargha-Delaney effect size  $\widehat{A}_{12}$  result (in parentheses) that given in the "Better Than" column.

Table 10 presents the statistical analysis of compared algorithms for eight threshold levels using the minimum cross-entropy method. It can be seen from the table that for the MRI1 image, ABC and JS exhibit the same characteristics,

and both are statistically better than WOA according to mean fitness value. However, Vargha–Delaney effect size  $\widehat{A}_{12}$  results show that WOA is superior to ABC and JS algorithms in most of the runs. This indicates that the WOA algorithm performs quite poorly in some runs and is not robust. In other words, WOA has higher standard deviation than the ABC in 30 runs. On the other hand, PSO and ALO are statistically better than both ABC and JS, while GWO is not better than any algorithms.

For the MRI2 image in the same table, JS and WOA outperform ABC, GWO, ALO, and MFO algorithms, while PSO is better than ABC, ALO, and MFO. For the MRI3 image, JS algorithm is statistically better than all algorithms except PSO, and PSO is better than GWO, WOA, and ALO. The results for the MRI4 image show that both PSO and JS algorithms perform better than GWO and WOA, while ABC, ALO, and MFO are better than just GWO. For the MRI5 image, JS and ALO algorithms are superior to ABC and GWO, while GWO is better than PSO and WOA. Herein WOA has the same characteristics as in the MRI1 image, showing that it is not robust.

From the results of the MRI6 image in Table 10, it can be seen that WOA statistically performs better than ABC, GWO, ALO, and JS. It must be noted that the ABC algorithm has the worst performance. For the images MRI7 and MRI8, the PSO algorithm statistically outperforms other algorithms, while GWO is worse than the others. It can be seen from the results of MRI9; that the JS algorithm is statistically more successful than other algorithms except for WOA. On the other hand, the ALO algorithm has the worst performance. For the MRI10 image, the PSO algorithm is statistically better than all the other algorithms, while GWO is worse than the others except for WOA.

Table 11 provides the statistical analysis of the compared algorithms for 8-levels of threshold using the between-class variance method. As can be seen from the table for the MRI1 image, the PSO and ALO algorithms are statistically more successful than GWO, MFO, and JS. On the other hand, GWO shows the poorest performance. For the MRI2 image, ABC is statistically better than GWO, ALO, and MFO, while JS is better than ABC, GWO, and MFO. Again, as in the MRI1 image, GWO has the poorest performance. From the results of the MRI3 image, the JS algorithm is statistically more successful than other algorithms except for PSO. Besides, there is no statistically significant difference between the GWO, WOA, and ALO algorithms.

For the MRI4 image, ABC, ALO, MFO, and JS algorithms statistically have a better performance than PSO and GWO algorithms. On the other hand, the GWO algorithm has the worst performance. The results of the MRI5 image in the same table show that while all other algorithms outperform the GWO algorithm, the JS algorithm surpasses the MFO algorithm along with GWO.

For the MRI6 image, ABC, ALO, and JS algorithms are statistically better than PSO, GWO, WOA, and MFO. Besides, GWO shows the poorest performance. It can be seen from the results of the MRI7 image that ABC is superior to GWO, WOA, MFO, and JS, whereas PSO surpasses GWO, WOA, MFO, and ALO algorithms. From the results of the MRI8 image, it can be said that both ALO and JS algorithms are statistically better than PSO, GWO, WOA, and MFO algorithms. According to Vargha–Delaney effect size  $\widehat{A}_{12}$  results, PSO is better than ABC, ALO, and JS algorithms in most of the runs, which reveals that the PSO algorithm performs poorly in some runs, and it is not robust. For the MRI9 image, the JS algorithm is statistically more successful than all other algorithms. Besides, ABC and WOA have better performance than PSO, GWO, and ALO algorithms. Finally, for the MRI10 image, the PSO algorithm is statistically better than other algorithms except for WOA, while the GWO algorithm shows the worst performance.

Overall, these results indicate that GWO performs poorly, while ABC, PSO, and JS algorithms perform better than others. However, the performances of the algorithms significantly differ according to the different images and threshold levels. This can be explained by the no-free-lunch theorem (Wolpert & Macready, 1997).

**Table 10.** Mann–Whitney U-Test Results ( $p < 0.05$ ) of Swarm-Based Optimization Algorithms and Its Corresponding Vargha–Delaney Effect Size  $\hat{A}_{12}$  Results (in Parentheses) That Given in the "Better Than" Column For 8-Level Threshold and Minimum Cross-Entropy Objective Function.

Image	Algorithm	Fitness	Better than
MRI1	ABC	0.3314	WOA (0.2489)
	PSO	0.3313	ABC (0.7556) JS (0.8217)
	GWO	0.3373	
	WOA	0.3322	GWO (0.6889)
	ALO	0.3270	ABC (0.6533) JS (0.6517)
	MFO	0.3305	JS (0.6500)
	JS	0.3314	WOA (0.2244)
MRI2	ABC	0.1885	
	PSO	0.1857	ABC (0.7394) ALO (0.6694) MFO (0.6939)
	GWO	0.1962	
	WOA	0.1882	ABC (0.7789) GWO (0.7689) ALO (0.6994) MFO (0.7889)
	ALO	0.1882	
	MFO	0.1885	
	JS	0.1850	ABC (0.7700) GWO (0.6633) ALO (0.6700) MFO (0.7417)
MRI3	ABC	0.1727	PSO (0.3211) GWO (0.6822)
	PSO	0.1762	GWO (0.7183) WOA (0.6800) ALO (0.6783)
	GWO	0.1881	
	WOA	0.1836	
	ALO	0.1866	
	MFO	0.1771	GWO (0.6733)
	JS	0.1676	ABC (0.7861) GWO (0.7933) WOA (0.7083) ALO (0.7328) MFO (0.6678)
MRI4	ABC	0.3032	GWO (0.8900)
	PSO	0.3054	GWO (0.8667) WOA (0.6639)
	GWO	0.3133	WOA (0.2322)
	WOA	0.3138	
	ALO	0.3060	GWO (0.8356)
	MFO	0.3060	GWO (0.8211)
	JS	0.3007	GWO (0.9622) WOA (0.6778)
MRI5	ABC	0.2598	WOA (0.2789)
	PSO	0.2596	ABC (0.7111)
	GWO	0.2557	PSO (0.3367) WOA (0.2822)
	WOA	0.2644	
	ALO	0.2528	ABC (0.7739) GWO (0.7756)
	MFO	0.2611	
	JS	0.2497	ABC (0.8822) GWO (0.9267)
MRI6	ABC	0.3158	
	PSO	0.2997	ABC (0.7333) ALO (0.6494)
	GWO	0.2906	ABC (0.8167)
	WOA	0.2895	ABC (0.8167) GWO (0.7567) ALO (0.7528) JS (0.6567)
	ALO	0.3088	ABC (0.6478)
	MFO	0.3003	ABC (0.7000)
	JS	0.2932	ABC (0.8267) ALO (0.6833)
MRI7	ABC	0.1173	GWO (0.9311) MFO (0.6511) JS (0.6756)
	PSO	0.1172	ABC (0.7933) GWO (0.9389) WOA (0.7950) ALO (0.8306) MFO (0.8167) JS (0.8478)
	GWO	0.1254	
	WOA	0.1253	GWO (0.7067)
	ALO	0.1192	GWO (0.7733)
	MFO	0.1206	GWO (0.7400)
	JS	0.1185	GWO (0.8044)
MRI8	ABC	0.1086	GWO (0.8583) WOA (0.7011)
	PSO	0.1075	ABC (0.9211) GWO (1.0000) WOA (0.9061) ALO (0.8456) MFO (0.8333) JS (0.8428)
	GWO	0.1251	
	WOA	0.1211	GWO (0.7006)
	ALO	0.1106	GWO (0.8233) WOA (0.7172)
	MFO	0.1098	GWO (0.8122) WOA (0.6967)
	JS	0.1078	ABC (0.6950) GWO (0.9511) WOA (0.7933)
MRI9	ABC	0.0989	PSO (0.6833) GWO (0.6822) ALO (0.8394)
	PSO	0.1245	ALO (0.6567)
	GWO	0.1029	ALO (0.7656)
	WOA	0.1061	PSO (0.6989) ALO (0.8222)
	ALO	0.1342	
	MFO	0.1008	ALO (0.7950)
	JS	0.0914	ABC (0.9133) PSO (0.7944) GWO (0.9611) ALO (0.9500) MFO (0.8772)
MRI10	ABC	0.1646	GWO (0.7411)
	PSO	0.1637	ABC (0.9206) GWO (0.9956) WOA (0.8344) ALO (0.8167) MFO (0.7961) JS (0.9178)
	GWO	0.1715	WOA (0.3222)
	WOA	0.1795	
	ALO	0.1639	ABC (0.8056) GWO (0.9778) JS (0.7728)
	MFO	0.1650	GWO (0.7822)
	JS	0.1642	GWO (0.8656)

**Table 11.** Mann–Whitney U-Test Results ( $p < 0.05$ ) of Swarm-Based Optimization Algorithms and Its Corresponding Vargha–Delaney Effect Size  $\hat{A}_{12}$  Results (in Parentheses) That Given in the "Better Than" Column For 8-level Threshold and Between-Class Variance Objective Function.

Image	Algorithm	Fitness	Better than
MRI1	ABC	4789.1538	PSO (0.6522) GWO (0.0744)
	PSO	4788.9420	GWO (0.1044) MFO (0.2689) JS (0.3139)
	GWO	4785.0889	
	WOA	4787.7008	GWO (0.1611)
	ALO	4788.7323	GWO (0.1444) MFO (0.3211) JS (0.3350)
	MFO	4788.3045	GWO (0.2167)
	JS	4788.7195	GWO (0.1478)
MRI2	ABC	2504.8737	GWO (0.0078) ALO (0.7444) MFO (0.1922)
	PSO	2504.7017	GWO (0.0833) MFO (0.2506)
	GWO	2502.0579	
	WOA	2503.3205	GWO (0.1767)
	ALO	2504.8229	GWO (0.0211) MFO (0.1483)
	MFO	2504.4935	GWO (0.2067)
	JS	2504.8855	ABC (0.3506) GWO (0.0078) MFO (0.1578)
MRI3	ABC	1240.3945	PSO (0.6933) MFO (0.6622)
	PSO	1240.1530	GWO (0.2356) WOA (0.3256) ALO (0.2578)
	GWO	1239.0510	
	WOA	1238.8258	
	ALO	1237.7775	
	MFO	1239.8529	GWO (0.2556)
	JS	1241.0314	ABC (0.0817) GWO (0.0522) WOA (0.3306) ALO (0.2867) MFO (0.3239)
MRI4	ABC	3087.1361	PSO (0.7439) GWO (0.0056)
	PSO	3086.8467	GWO (0.0444) WOA (0.2261)
	GWO	3083.7930	
	WOA	3085.4550	GWO (0.2044)
	ALO	3087.1285	PSO (0.7889) GWO (0.0056)
	MFO	3087.0622	PSO (0.7344) GWO (0.0322)
	JS	3087.1427	PSO (0.8317) GWO (0.0000)
MRI5	ABC	3031.2010	GWO (0.0000)
	PSO	3030.7024	GWO (0.0900)
	GWO	3028.8021	
	WOA	3030.2742	GWO (0.1322)
	ALO	3030.9397	GWO (0.0600)
	MFO	3030.3694	GWO (0.1367)
	JS	3031.1965	GWO (0.0067) MFO (0.3089)
MRI6	ABC	3084.5953	PSO (0.6922) GWO (0.0044) WOA (0.1611) MFO (0.1922)
	PSO	3084.3652	GWO (0.0356) WOA (0.1550) MFO (0.2433)
	GWO	3081.6150	
	WOA	3083.1328	GWO (0.2189)
	ALO	3084.4014	PSO (0.7094) GWO (0.0389) WOA (0.1922) MFO (0.2844)
	MFO	3083.9624	GWO (0.1022) WOA (0.3489)
	JS	3084.5361	PSO (0.6833) GWO (0.0133) WOA (0.1533) MFO (0.2111)
MRI7	ABC	2555.3314	GWO (0.0289) WOA (0.2922) MFO (0.3517) JS (0.2667)
	PSO	2555.2039	GWO (0.0233) WOA (0.2111) ALO (0.3372) MFO (0.2700)
	GWO	2553.7412	WOA (0.6756)
	WOA	2553.2087	
	ALO	2555.1843	GWO (0.0322) WOA (0.3122)
	MFO	2555.1172	GWO (0.1433)
	JS	2555.2831	PSO (0.8728) GWO (0.0900) ALO (0.7100)
MRI8	ABC	2842.7372	PSO (0.8017) GWO (0.0189) WOA (0.1489)
	PSO	2842.6980	GWO (0.0333) WOA (0.0628) MFO (0.1956)
	GWO	2840.5944	
	WOA	2840.1082	
	ALO	2842.7446	PSO (0.7606) GWO (0.0078) WOA (0.1456) MFO (0.3417)
	MFO	2842.2292	GWO (0.1433) WOA (0.2461)
	JS	2842.7542	PSO (0.7611) GWO (0.0044) WOA (0.1106) MFO (0.3022)
MRI9	ABC	539.9905	PSO (0.3389) GWO (0.2456) ALO (0.2583)
	PSO	538.6416	
	GWO	539.6719	ALO (0.2844)
	WOA	539.6833	PSO (0.3372) GWO (0.3144) ALO (0.2528)
	ALO	538.4992	
	MFO	539.7868	GWO (0.3244) ALO (0.2828)
	JS	540.2493	ABC (0.2228) PSO (0.2439) GWO (0.1056) WOA (0.3033) ALO (0.1072) MFO (0.2067)
MRI10	ABC	3219.4032	GWO (0.0844) WOA (0.6922)
	PSO	3219.5013	ABC (0.1356) GWO (0.0078) ALO (0.2006) MFO (0.2750) JS (0.2717)
	GWO	3216.8426	
	WOA	3218.0689	GWO (0.1522)
	ALO	3219.4860	ABC (0.2572) GWO (0.0067)
	MFO	3219.4321	GWO (0.0656)
	JS	3219.4869	ABC (0.2444) GWO (0.0089)

### Computational Time Analysis

This section provides a computational time analysis to examine the time complexity of the compared algorithms. Tables 12 and 13 compare the computational times of the algorithms using minimum cross-entropy and between-class variance objective functions, respectively. According to Table 12, WOA is the fastest algorithm in all images and threshold values, while ALO and PSO are the slowest algorithms, respectively. In terms of overall rank values, GWO is the second-fastest algorithm, followed by MFO, JS, and ABC. When Table 13 is examined, as in the minimum cross-entropy metric, WOA is the fastest algorithm while ALO is the slowest one. GWO is the second-fastest algorithm, followed by MFO, JS, ABC, and PSO.

Since all algorithms are compared by a fixed number of evaluations instead of the maximum number of iterations, it is expected that the average CPU times of the algorithms will be similar. According to the results, WOA, GWO, MFO, JS, and ABC algorithms obtained results at a similar time, while ALO and PSO algorithms are slower compared to others. Therefore, it turns out that WOA, GWO, MFO, JS, and ABC have effective internal mechanisms.

**Table 12.** Comparison of CPU Time Consumption (in Seconds) of Swarm-Based Optimization Algorithms Using a Minimum Cross-Entropy Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	0.2958 (5)	0.04	0.5006 (6)	0.02	0.2327 (2)	0.01	<b>0.2286 (1)</b>	0.01	0.5131 (7)	0.05	0.2367 (3)	0.01	0.2493 (4)	0.00
	4	0.2956 (5)	0.00	0.5050 (6)	0.00	0.2449 (2)	0.00	<b>0.2363 (1)</b>	0.00	0.7035 (7)	0.00	0.2464 (3)	0.00	0.2570 (4)	0.00
	6	0.3040 (5)	0.01	0.5121 (6)	0.00	0.2589 (3)	0.01	<b>0.2452 (1)</b>	0.00	0.8816 (7)	0.01	0.2586 (2)	0.00	0.2636 (4)	0.00
	8	0.3093 (5)	0.00	0.5188 (6)	0.00	0.2697 (2)	0.00	<b>0.2533 (1)</b>	0.00	1.1035 (7)	0.01	0.2710 (3)	0.00	0.2710 (4)	0.00
MRI2	2	0.2886 (5)	0.00	0.4968 (6)	0.00	0.2314 (2)	0.00	<b>0.2269 (1)</b>	0.00	0.5042 (7)	0.00	0.2337 (3)	0.00	0.2475 (4)	0.00
	4	0.2953 (5)	0.00	0.5038 (6)	0.01	0.2433 (2)	0.00	<b>0.2350 (1)</b>	0.00	0.7025 (7)	0.00	0.2460 (3)	0.00	0.2553 (4)	0.00
	6	0.3011 (5)	0.00	0.5105 (6)	0.00	0.2556 (2)	0.00	<b>0.2435 (1)</b>	0.00	0.8769 (7)	0.00	0.2565 (3)	0.00	0.2623 (4)	0.00
	8	0.3075 (5)	0.00	0.5206 (6)	0.01	0.2689 (2)	0.00	<b>0.2532 (1)</b>	0.00	1.1018 (7)	0.01	0.2700 (3)	0.01	0.2705 (4)	0.01
MRI3	2	0.2884 (5)	0.00	0.4962 (6)	0.00	0.2309 (2)	0.00	<b>0.2271 (1)</b>	0.00	0.5022 (7)	0.00	0.2331 (3)	0.00	0.2474 (4)	0.00
	4	0.2947 (5)	0.00	0.5032 (6)	0.00	0.2435 (2)	0.00	<b>0.2357 (1)</b>	0.00	0.7030 (7)	0.01	0.2453 (3)	0.00	0.2550 (4)	0.00
	6	0.3018 (5)	0.00	0.5109 (6)	0.00	0.2569 (2)	0.00	<b>0.2442 (1)</b>	0.00	0.8761 (7)	0.01	0.2572 (3)	0.00	0.2630 (4)	0.00
	8	0.3076 (5)	0.00	0.5178 (6)	0.00	0.2688 (2)	0.00	<b>0.2536 (1)</b>	0.00	1.1022 (7)	0.01	0.2702 (4)	0.00	0.2698 (3)	0.00
MRI4	2	0.2884 (5)	0.01	0.4963 (6)	0.01	0.2308 (2)	0.00	<b>0.2267 (1)</b>	0.00	0.5043 (7)	0.00	0.2333 (3)	0.00	0.2485 (4)	0.00
	4	0.2943 (5)	0.00	0.5030 (6)	0.00	0.2428 (2)	0.00	<b>0.2348 (1)</b>	0.00	0.7017 (7)	0.00	0.2451 (3)	0.00	0.2556 (4)	0.00
	6	0.3015 (5)	0.00	0.5104 (6)	0.00	0.2555 (2)	0.00	<b>0.2433 (1)</b>	0.00	0.8787 (7)	0.01	0.2583 (3)	0.01	0.2636 (4)	0.00
	8	0.3072 (5)	0.00	0.5189 (6)	0.00	0.2693 (2)	0.00	<b>0.2535 (1)</b>	0.00	1.1018 (7)	0.01	0.2700 (3)	0.00	0.2703 (4)	0.00
MRI5	2	0.2883 (5)	0.00	0.4970 (6)	0.00	0.2310 (2)	0.00	<b>0.2266 (1)</b>	0.00	0.5036 (7)	0.00	0.2337 (3)	0.00	0.2491 (4)	0.00
	4	0.2949 (5)	0.00	0.5036 (6)	0.00	0.2429 (2)	0.00	<b>0.2358 (1)</b>	0.00	0.7032 (7)	0.01	0.2450 (3)	0.00	0.2557 (4)	0.00
	6	0.3008 (5)	0.00	0.5111 (6)	0.00	0.2562 (2)	0.00	<b>0.2447 (1)</b>	0.00	0.8804 (7)	0.01	0.2592 (3)	0.01	0.2630 (4)	0.00
	8	0.3078 (5)	0.00	0.5184 (6)	0.01	0.2691 (2)	0.00	<b>0.2527 (1)</b>	0.00	1.1015 (7)	0.01	0.2699 (3)	0.00	0.2702 (4)	0.00
MRI6	2	0.2879 (5)	0.00	0.4968 (6)	0.00	0.2308 (2)	0.00	<b>0.2273 (1)</b>	0.00	0.5039 (7)	0.00	0.2338 (3)	0.00	0.2487 (4)	0.00
	4	0.2947 (5)	0.00	0.5040 (6)	0.00	0.2433 (2)	0.00	<b>0.2357 (1)</b>	0.00	0.7036 (7)	0.01	0.2449 (3)	0.00	0.2560 (4)	0.00
	6	0.3008 (5)	0.00	0.5108 (6)	0.00	0.2557 (2)	0.00	<b>0.2444 (1)</b>	0.00	0.8786 (7)	0.01	0.2577 (3)	0.00	0.2627 (4)	0.00
	8	0.3081 (5)	0.00	0.5213 (6)	0.01	0.2697 (2)	0.00	<b>0.2528 (1)</b>	0.00	1.1032 (7)	0.01	0.2701 (3)	0.00	0.2702 (4)	0.00
MRI7	2	0.2861 (5)	0.00	0.4934 (6)	0.00	0.2287 (2)	0.00	<b>0.2244 (1)</b>	0.00	0.5016 (7)	0.00	0.2314 (3)	0.00	0.2464 (4)	0.00
	4	0.2932 (5)	0.00	0.5011 (6)	0.00	0.2414 (2)	0.00	<b>0.2339 (1)</b>	0.00	0.7006 (7)	0.00	0.2438 (3)	0.00	0.2534 (4)	0.00
	6	0.2989 (5)	0.00	0.5098 (6)	0.00	0.2539 (2)	0.00	<b>0.2422 (1)</b>	0.00	0.8765 (7)	0.01	0.2551 (3)	0.00	0.2606 (4)	0.00
	8	0.3056 (5)	0.00	0.5171 (6)	0.00	0.2674 (2)	0.00	<b>0.2514 (1)</b>	0.00	1.1019 (7)	0.01	0.2679 (3)	0.00	0.2680 (4)	0.00
MRI8	2	0.2876 (5)	0.00	0.4958 (6)	0.00	0.2301 (2)	0.00	<b>0.2262 (1)</b>	0.00	0.5029 (7)	0.00	0.2326 (3)	0.00	0.2473 (4)	0.00
	4	0.2943 (5)	0.00	0.5029 (6)	0.00	0.2434 (2)	0.00	<b>0.2355 (1)</b>	0.00	0.7040 (7)	0.00	0.2447 (3)	0.00	0.2556 (4)	0.00
	6	0.3028 (5)	0.01	0.5136 (6)	0.01	0.2568 (2)	0.01	<b>0.2442 (1)</b>	0.00	0.8771 (7)	0.01	0.2571 (3)	0.00	0.2629 (4)	0.00
	8	0.3094 (5)	0.01	0.5250 (6)	0.04	0.2703 (3)	0.01	<b>0.2541 (1)</b>	0.00	1.1044 (7)	0.01	0.2699 (2)	0.00	0.2714 (4)	0.01
MRI9	2	0.2872 (5)	0.00	0.4947 (6)	0.00	0.2291 (2)	0.00	<b>0.2251 (1)</b>	0.00	0.5020 (7)	0.00	0.2326 (3)	0.00	0.2480 (4)	0.00
	4	0.2947 (5)	0.00	0.5052 (6)	0.00	0.2431 (2)	0.00	<b>0.2345 (1)</b>	0.00	0.7046 (7)	0.00	0.2444 (3)	0.00	0.2556 (4)	0.00
	6	0.3017 (5)	0.00	0.5133 (6)	0.01	0.2552 (2)	0.00	<b>0.2440 (1)</b>	0.00	0.8802 (7)	0.00	0.2571 (3)	0.00	0.2629 (4)	0.00
	8	0.3093 (5)	0.00	0.5204 (6)	0.00	0.2693 (2)	0.00	<b>0.2531 (1)</b>	0.00	1.1092 (7)	0.02	0.2706 (4)	0.00	0.2699 (3)	0.00
MRI10	2	0.2874 (5)	0.00	0.4949 (6)	0.00	0.2297 (2)	0.00	<b>0.2255 (1)</b>	0.00	0.5022 (7)	0.00	0.2322 (3)	0.00	0.2479 (4)	0.00
	4	0.2944 (5)	0.00	0.5028 (6)	0.00	0.2426 (2)	0.00	<b>0.2350 (1)</b>	0.00	0.7030 (7)	0.00	0.2446 (3)	0.00	0.2547 (4)	0.00
	6	0.3005 (5)	0.00	0.5106 (6)	0.00	0.2560 (2)	0.00	<b>0.2440 (1)</b>	0.00	0.8778 (7)	0.00	0.2570 (3)	0.00	0.2625 (4)	0.00
	8	0.3070 (5)	0.00	0.5181 (6)	0.00	0.2689 (3)	0.00	<b>0.2530 (1)</b>	0.00	1.0993 (7)	0.01	0.2688 (2)	0.00	0.2693 (4)	0.00
Av Rank		5±0(5)		6±0(6)		2.08±0.27(2)		<b>1±0(1)</b>		7±0 (7)		2.98±0.36 (3)		3.95±0.22 (4)	

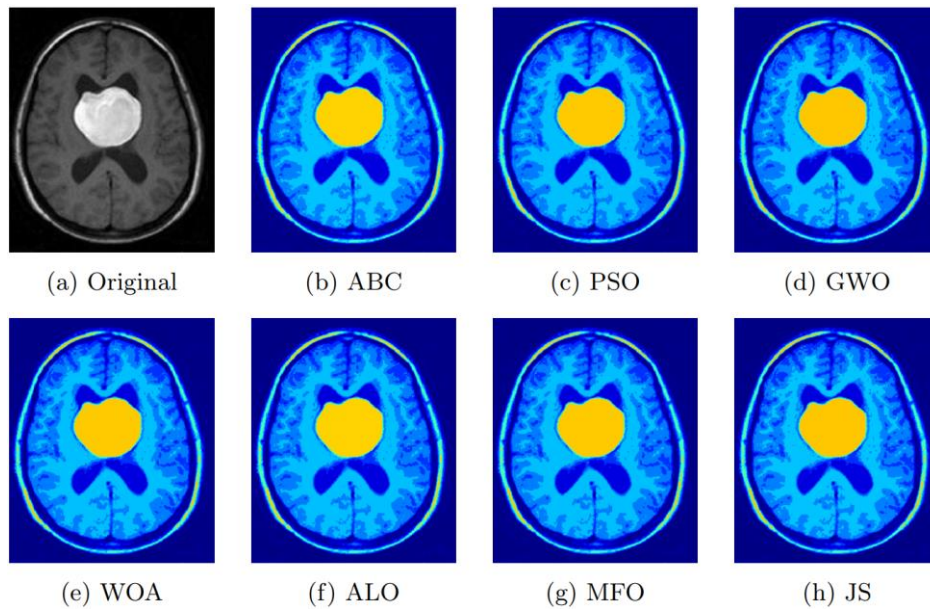
**Table 13.** Comparison of CPU Time Consumption (in Seconds) of Swarm-Based Optimization Algorithms Using Between-Class Variance Objective Function.

Image	nTh	ABC		PSO		GWO		WOA		ALO		MFO		JS	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
MRI1	2	0.9970 (5)	0.05	1.2084 (7)	0.02	0.9306 (2)	0.01	<b>0.9291 (1)</b>	0.01	1.2071 (6)	0.02	0.9326 (3)	0.01	0.9433 (4)	0.02
	4	1.0383 (5)	0.00	1.2556 (6)	0.00	0.9858 (3)	0.00	<b>0.9794 (1)</b>	0.00	1.4485 (7)	0.01	0.9851 (2)	0.00	0.9926 (4)	0.00
	6	1.0837 (5)	0.00	1.3047 (6)	0.00	1.0405 (3)	0.00	<b>1.0335 (1)</b>	0.00	1.6668 (7)	0.01	1.0381 (2)	0.00	1.0425 (4)	0.01
	8	1.1319 (5)	0.00	1.3574 (6)	0.01	1.0965 (4)	0.00	<b>1.0889 (1)</b>	0.01	1.9322 (7)	0.01	1.0921 (3)	0.00	1.0910 (2)	0.01
MRI2	2	2.0002 (5)	0.00	2.2291 (7)	0.00	1.9297 (2)	0.01	<b>1.9252 (1)</b>	0.01	2.2281 (6)	0.01	1.9304 (3)	0.01	1.9448 (4)	0.01
	4	2.0578 (5)	0.00	2.2889 (6)	0.01	1.9970 (2)	0.00	<b>1.9958 (1)</b>	0.01	2.4920 (7)	0.01	1.9988 (3)	0.01	2.0109 (4)	0.01
	6	2.1084 (5)	0.01	2.3425 (6)	0.00	2.0608 (3)	0.01	<b>2.0532 (1)</b>	0.01	2.7158 (7)	0.01	2.0598 (2)	0.01	2.0658 (4)	0.01
	8	2.1579 (5)	0.01	2.3940 (6)	0.01	2.1204 (4)	0.01	<b>2.1109 (1)</b>	0.00	2.9839 (7)	0.01	2.1148 (2)	0.00	2.1177 (3)	0.01
MRI3	2	1.2588 (5)	0.04	1.5709 (6)	0.00	1.1829 (2)	0.04	<b>1.1777 (1)</b>	0.03	1.5748 (7)	0.01	1.1899 (3)	0.04	1.2121 (4)	0.04
	4	1.3707 (5)	0.03	1.6324 (6)	0.01	<b>1.3096 (1)</b>	0.03	1.3148 (2)	0.03	1.8573 (7)	0.02	1.3282 (3)	0.01	1.3351 (4)	0.02
	6	1.4421 (5)	0.01	1.7004 (6)	0.03	1.3934 (2)	0.01	<b>1.3833 (1)</b>	0.02	2.0963 (7)	0.02	1.3954 (3)	0.01	1.4059 (4)	0.00
	8	1.5029 (5)	0.02	1.7595 (6)	0.03	1.4577 (2)	0.01	<b>1.4474 (1)</b>	0.01	2.3831 (7)	0.04	1.4697 (4)	0.03	1.4627 (3)	0.01
MRI4	2	1.4540 (5)	0.02	1.7062 (6)	0.01	1.3855 (2)	0.03	<b>1.3805 (1)</b>	0.03	1.7110 (7)	0.01	1.3888 (3)	0.03	1.4062 (4)	0.02
	4	1.5150 (5)	0.01	1.7645 (6)	0.01	1.4592 (2)	0.02	<b>1.4555 (1)</b>	0.01	1.9805 (7)	0.01	1.4619 (3)	0.01	1.4693 (4)	0.00
	6	1.5719 (5)	0.01	1.8209 (6)	0.01	1.5258 (4)	0.02	<b>1.5146 (1)</b>	0.01	2.2083 (7)	0.01	1.5188 (2)	0.01	1.5232 (3)	0.01
	8	1.6349 (5)	0.08	1.8870 (6)	0.06	1.5887 (4)	0.03	1.5798 (2)	0.04	2.4857 (7)	0.05	1.5805 (3)	0.01	<b>1.5796 (1)</b>	0.02
MRI5	2	1.3911 (5)	0.04	1.6750 (6)	0.01	1.3244 (3)	0.04	<b>1.3110 (1)</b>	0.04	1.6783 (7)	0.03	1.3221 (2)	0.03	1.3373 (4)	0.04
	4	1.4662 (5)	0.03	1.7333 (6)	0.01	1.4248 (3)	0.01	<b>1.4117 (1)</b>	0.02	1.9512 (7)	0.01	1.4228 (2)	0.01	1.4273 (4)	0.02
	6	1.5371 (5)	0.01	1.7899 (6)	0.01	1.4911 (3)	0.00	<b>1.4829 (1)</b>	0.01	2.1800 (7)	0.01	1.4878 (2)	0.01	1.4934 (4)	0.00
	8	1.5906 (5)	0.01	1.8451 (6)	0.01	1.5546 (4)	0.01	<b>1.5420 (1)</b>	0.01	2.4506 (7)	0.01	1.5496 (3)	0.01	1.5483 (2)	0.01
MRI6	2	1.4115 (5)	0.04	1.7008 (7)	0.01	1.3540 (3)	0.03	1.3527 (2)	0.04	1.7007 (6)	0.01	<b>1.3488 (1)</b>	0.04	1.3715 (4)	0.03
	4	1.4992 (5)	0.03	1.7854 (6)	0.15	1.4580 (2)	0.11	1.4620 (4)	0.09	1.9809 (7)	0.04	<b>1.4546 (1)</b>	0.05	1.4615 (3)	0.04
	6	1.5627 (5)	0.01	1.8176 (6)	0.02	1.5152 (3)	0.02	<b>1.5075 (1)</b>	0.01	2.2025 (7)	0.01	1.5134 (2)	0.01	1.5177 (4)	0.01
	8	1.6141 (5)	0.01	1.8655 (6)	0.01	1.5760 (4)	0.00	<b>1.5680 (1)</b>	0.01	2.4710 (7)	0.01	1.5725 (3)	0.01	1.5702 (2)	0.00
MRI7	2	2.2578 (5)	0.03	2.4826 (6)	0.03	<b>2.1952 (1)</b>	0.04	2.2000 (2)	0.04	2.4933 (7)	0.04	2.2011 (3)	0.03	2.2111 (4)	0.03
	4	2.3066 (5)	0.02	2.5380 (6)	0.03	2.2475 (2)	0.01	<b>2.2468 (1)</b>	0.01	2.7445 (7)	0.03	2.2504 (3)	0.02	2.2638 (4)	0.02
	6	2.3633 (5)	0.03	2.5921 (6)	0.03	2.3132 (2)	0.02	<b>2.3093 (1)</b>	0.03	2.9778 (7)	0.05	2.3181 (3)	0.04	2.3273 (4)	0.04
	8	2.4131 (5)	0.02	2.6547 (6)	0.03	2.3790 (4)	0.03	2.3745 (2)	0.03	3.2413 (7)	0.06	<b>2.3723 (1)</b>	0.04	2.3754 (3)	0.03
MRI8	2	2.0051 (5)	0.01	2.2355 (7)	0.00	1.9355 (2)	0.00	<b>1.9324 (1)</b>	0.01	2.2329 (6)	0.01	1.9359 (3)	0.00	1.9497 (4)	0.01
	4	2.0615 (5)	0.00	2.2907 (6)	0.01	2.0014 (3)	0.00	<b>1.9999 (1)</b>	0.01	2.4923 (7)	0.01	2.0009 (2)	0.00	2.0123 (4)	0.01
	6	2.1149 (5)	0.01	2.3481 (6)	0.01	2.0757 (2)	0.02	<b>2.0669 (1)</b>	0.02	2.7214 (7)	0.01	2.1059 (4)	0.23	2.0827 (3)	0.07
	8	2.1631 (5)	0.00	2.4021 (6)	0.01	2.1455 (4)	0.03	2.1407 (3)	0.03	2.9858 (7)	0.01	<b>2.1237 (1)</b>	0.01	2.1251 (2)	0.01
MRI9	2	3.6566 (5)	0.01	3.9315 (7)	0.02	3.5861 (2)	0.01	<b>3.5840 (1)</b>	0.01	3.9314 (6)	0.02	3.5877 (3)	0.01	3.6080 (4)	0.01
	4	3.7256 (5)	0.01	4.0011 (6)	0.02	3.6511 (2)	0.01	<b>3.6507 (1)</b>	0.00	4.2105 (7)	0.02	3.6576 (3)	0.01	3.6730 (4)	0.01
	6	3.8118 (5)	0.02	4.0728 (6)	0.04	3.7170 (2)	0.01	<b>3.7162 (1)</b>	0.02	4.4596 (7)	0.04	3.7358 (3)	0.03	3.7448 (4)	0.01
	8	3.9358 (5)	0.02	4.1637 (6)	0.05	<b>3.7880 (1)</b>	0.02	3.7953 (2)	0.04	4.7524 (7)	0.06	3.8243 (3)	0.03	3.8245 (4)	0.02
MRI10	2	2.0042 (5)	0.00	2.2340 (7)	0.01	1.9326 (2)	0.01	<b>1.9311 (1)</b>	0.00	2.2287 (6)	0.01	1.9327 (3)	0.01	1.9483 (4)	0.00
	4	2.0588 (5)	0.00	2.2887 (6)	0.00	2.0004 (3)	0.00	1.9985 (2)	0.00	2.4913 (7)	0.01	<b>1.9980 (1)</b>	0.00	2.0085 (4)	0.00
	6	2.1120 (5)	0.01	2.3444 (6)	0.01	2.0649 (3)	0.01	2.0631 (2)	0.02	2.7173 (7)	0.02	<b>2.0597 (1)</b>	0.01	2.0667 (4)	0.01
	8	2.1593 (5)	0.00	2.4004 (6)	0.03	2.1349 (4)	0.02	2.1232 (3)	0.02	2.9830 (7)	0.01	2.1180 (2)	0.00	<b>2.1179 (1)</b>	0.00
Av Rank		5±0(5)		6.15±0.36(6)		2.65±0.92(2.5)		<b>1.38±0.7(1)</b>		6.85±0.36(7)		2.48±0.82(3)		3.5±0.88(4)	

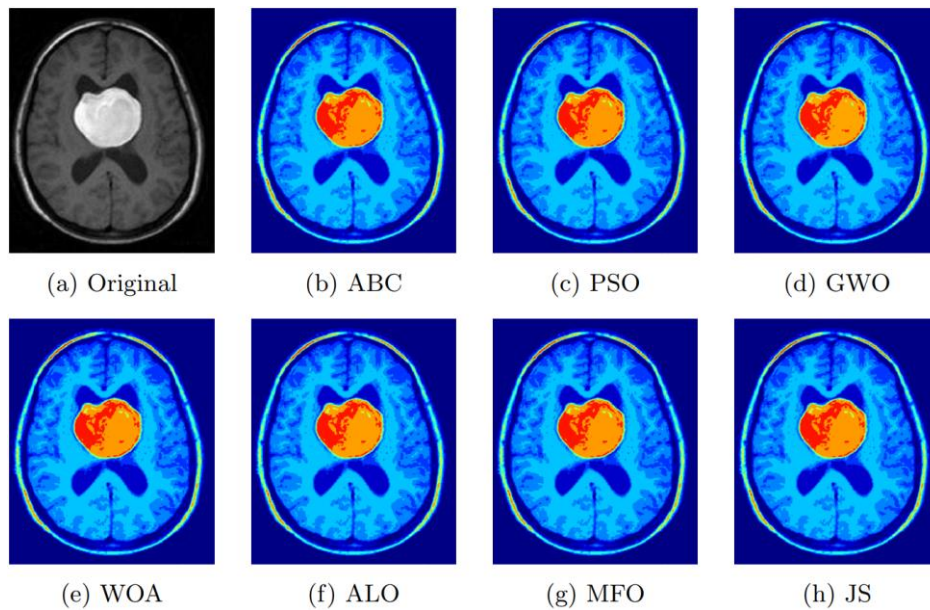
### Visual Evaluation

In this section, the multilevel thresholding performance of the compared swarm-based optimization algorithms is evaluated visually. Considering the article's length, the visual results of only two images (MRI6, MRI8) and one threshold level (8-level threshold) are given.

Figs. 3 and 4 illustrate the thresholded images of the MRI6 image for the best 8-level threshold values of the compared algorithms using between-class variance and minimum cross-entropy objective functions, respectively. For a more precise comparison, thresholded images are colorized. Since the MRI6 image contains a tumor, it is crucial that algorithms should accurately segment this tumor. When Figs. 3 and 4 are examined; it can be seen that algorithms using minimum cross-entropy segment the tumor in the image into one region. Additionally, it is seen that the best threshold values of the algorithms using the same objective function obtain very similar visual results.



**Fig. 3** The Thresholded Images with 8-level Using Minimum Cross-Entropy for MRI6 Image.



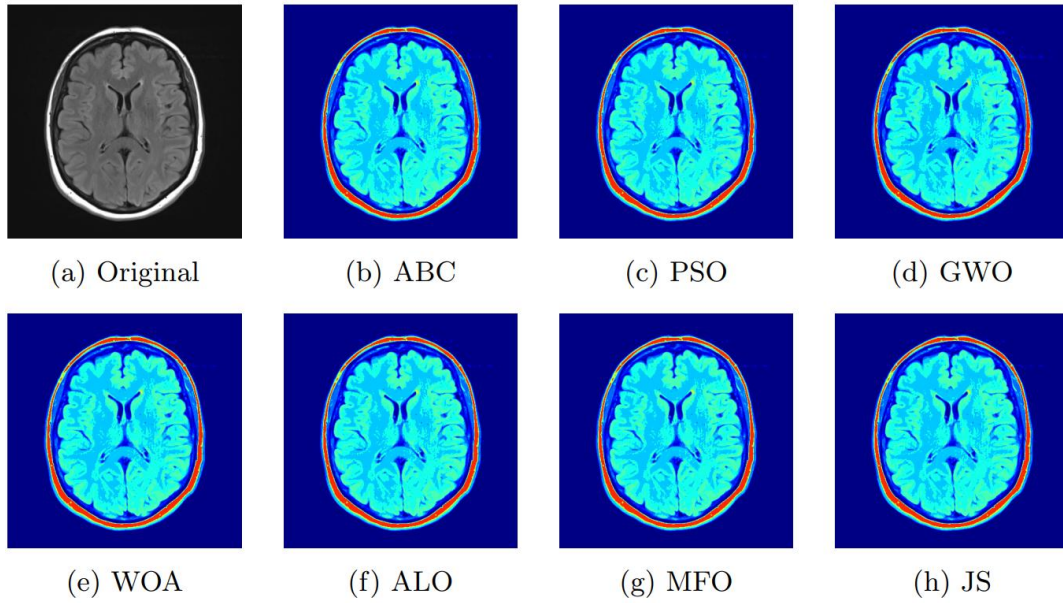
**Fig. 4** The Thresholded Images with 8-level Using Between-Class Variance for MRI6 image.

Figs. 5 and 6 show the thresholded images of the MRI8 image for the best 8-level threshold values of the swarm-based optimization algorithms using minimum cross-entropy and between-class variance objective functions, respectively. From the figures, it is seen that thresholded images of the algorithms produced using the best threshold values are very similar.

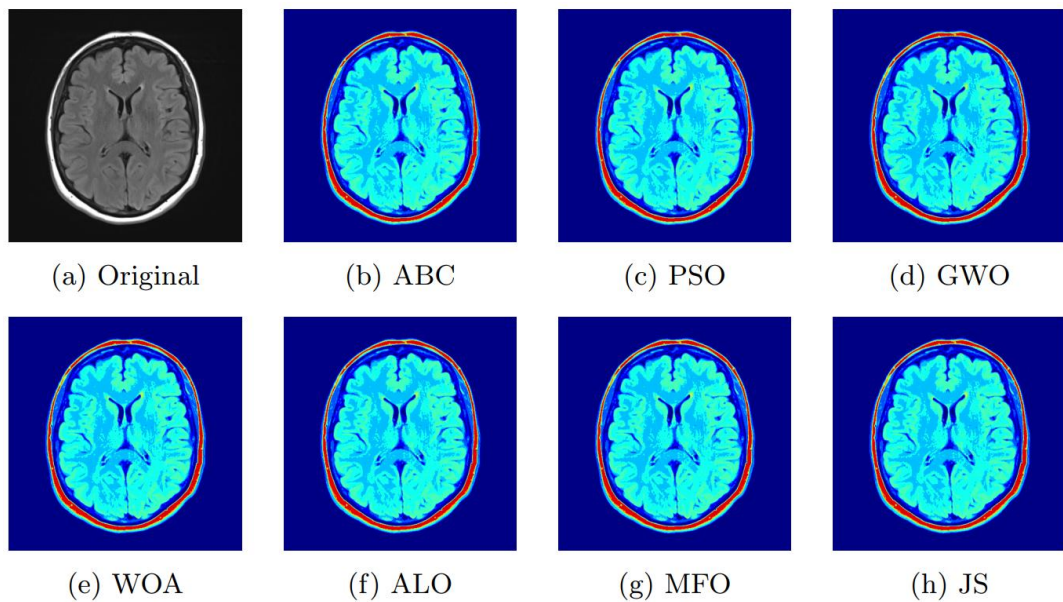
## CONCLUSION

In this study, the performances of the seven popular swarm-based optimization algorithms on the multilevel image thresholding problem, which is a widely used technique in many image processing applications for image segmentation, have been investigated in detail. The main contribution of this study is to objectively compare the performance of PSO, ABC, GWO, MFO, ALO, WOA, and JS algorithms for multilevel thresholding in brain MR image segmentation. Additionally, two popular objective functions for optimal multilevel thresholding-based image segmentation were compared: minimum cross-entropy and between-class variance. The performance of the optimization algorithms and objective functions were compared both quantitatively and qualitatively.





**Fig. 5** The Thresholded Images with 8-level Using Minimum Cross-Entropy for MRI8 Image.



**Fig. 6** The Thresholded Images with 8-level Using Between-Class Variance for MRI8 Image.

The performance of the algorithms was compared quantitatively using reference-based PSNR and SSIM and non-reference-based BRISQUE and NIQE metrics. In terms of fitness and PSNR metric values, JS, ABC, and PSO show the best performance. The results reveal that there is a positive correlation between fitness values and the PSNR metric. On the other hand, GWO, WOA, and MFO are superior to other algorithms considering SSIM. According to BRISQUE and NIQE metrics, different algorithms produced the best results for different image and objective functions. Experimental results show that BRISQUE and NIQE metrics are surprisingly inconsistent and not correlated with fitness values. This is because BRISQUE and NIQE are non-reference-based metrics.

The Mann-Whitney U test was also applied to the fitness values of the optimization algorithms to verify their numerical metric results. From the results, the JS, PSO, and ABC algorithms were statistically more successful than other algorithms, while GWO showed the poorest performance. It must also be noted that Vargha–Delaney effect size  $\widehat{A}_{12}$  results showed WOA performed better in most of the runs for some images and threshold values, where it performed poorly according to the mean fitness value. This indicates that the WOA algorithm performs relatively poorly in some runs and is not robust.

In the experiments, we also examined the time complexity of the algorithms considering their CPU time consumption under the same number of function evaluation criteria. The results of computational time analysis set out that WOA is the most efficient algorithm, followed by GWO, MFO, JS, and ABC, while ALO and PSO were the slowest.

## REFERENCES

- Akay, B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing Journal*, 13(6). <https://doi.org/10.1016/j.asoc.2012.03.072>
- Aslan, S., Demirci, S., Oktay, T., & Yesilbas, E. (2023). Percentile-Based Adaptive Immune Plasma Algorithm and Its Application to Engineering Optimization. *Biomimetics*, 8(6), 486.
- Aziz, M. A. El, Ewees, A. A., & Hassanien, A. E. (2017). Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation. *Expert Systems with Applications*, 83. <https://doi.org/10.1016/j.eswa.2017.04.023>
- Bakhshali, M. A., & Shamsi, M. (2014). Segmentation of color lip images by optimal thresholding using bacterial foraging optimization (BFO). *Journal of Computational Science*, 5(2). <https://doi.org/10.1016/j.jocs.2013.07.001>
- Brajevic, I., & Tuba, M. (2014). Cuckoo search and firefly algorithm applied to multilevel image thresholding. *Studies in Computational Intelligence*, 516. [https://doi.org/10.1007/978-3-319-02141-6\\_6](https://doi.org/10.1007/978-3-319-02141-6_6)
- Chakrabarty, N. (2019). Brain MRI Images for Brain Tumor Detection. Retrieved from <https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection>
- Chou, J.-S., & Truong, D.-N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, 389, 125535. <https://doi.org/10.1016/j.amc.2020.125535>
- Cuevas, E., Zaldivar, D., & Pérez-Cisneros, M. (2010). A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert Systems with Applications*, 37(7). <https://doi.org/10.1016/j.eswa.2010.01.013>
- Dhal, K. G., Das, A., Ray, S., Gálvez, J., & Das, S. (2020). Nature-Inspired Optimization Algorithms and Their Application in Multi-Thresholding Image Segmentation. In *Archives of Computational Methods in Engineering* (Vol. 27). Springer Netherlands. <https://doi.org/10.1007/s11831-019-09334-y>
- Gao, H., Fu, Z., Pun, C. M., Hu, H., & Lan, R. (2018). A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm. *Computers and Electrical Engineering*, 70. <https://doi.org/10.1016/j.compeleceng.2017.12.037>
- Ghamisi, P., Couceiro, M. S., Martins, F. M. L., & Benediktsson, J. A. (2014). Multilevel image segmentation based on fractional-order darwinian particle swarm optimization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5). <https://doi.org/10.1109/TGRS.2013.2260552>
- Gharehchopogh, F. S., & Ibrikci, T. (2024). An improved African vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation. *Multimedia Tools and Applications*, 83(6). <https://doi.org/10.1007/s11042-023-16300-1>
- Guo, H., Li, M., Liu, H., Chen, X., Cheng, Z., Li, X., ... He, Q. (2024). Multi-threshold Image Segmentation based on an improved Salp Swarm Algorithm: Case study of breast cancer pathology images. *Computers in Biology and Medicine*, 168(August 2023), 107769. <https://doi.org/10.1016/j.combiomed.2023.107769>
- Hammouche, K., Diaf, M., & Siarry, P. (2008). A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Computer Vision and Image Understanding*, 109(2). <https://doi.org/10.1016/j.cviu.2007.09.001>

Hammouche, K., Diaf, M., & Siarry, P. (2010). A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem. *Engineering Applications of Artificial Intelligence*, 23(5), 676–688. <https://doi.org/10.1016/j.engappai.2009.09.011>

Jena, B., Naik, M. K., Panda, R., & Abraham, A. (2021). Maximum 3D Tsallis entropy based multilevel thresholding of brain MR image using attacking Manta Ray foraging optimization. *Engineering Applications of Artificial Intelligence*, 103(April), 104293. <https://doi.org/10.1016/j.engappai.2021.104293>

Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3), 273–285. [https://doi.org/10.1016/0734-189X\(85\)90125-2](https://doi.org/10.1016/0734-189X(85)90125-2)

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471. <https://doi.org/10.1007/s10898-007-9149-x>

Karakoyun, M. (2023). The Comparison Of The Effects Of Thresholding Methods On Segmentation Using The Moth Flame Optimization Algorithm. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 26(2). <https://doi.org/10.17780/ksujes.1222041>

Kaur, T., Saini, B. S., & Gupta, S. (2016). Optimized multi threshold brain tumor image segmentation using two dimensional minimum cross entropy based on co-occurrence matrix. In *Studies in Computational Intelligence* (Vol. 651). [https://doi.org/10.1007/978-3-319-33793-7\\_20](https://doi.org/10.1007/978-3-319-33793-7_20)

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942–1948. IEEE. <https://doi.org/10.1109/ICNN.1995.488968>

Kotte, S., Pullakura, R. K., & Injeti, S. K. (2018). Optimal multilevel thresholding selection for brain MRI image segmentation based on adaptive wind driven optimization. *Measurement*, 130, 340–361. <https://doi.org/10.1016/j.measurement.2018.08.007>

Kurban, R., Durmus, A., & Karakose, E. (2021). A comparison of novel metaheuristic algorithms on color aerial image multilevel thresholding. *Engineering Applications of Artificial Intelligence*, 105(July), 104410. <https://doi.org/10.1016/j.engappai.2021.104410>

Kurban, T., Civicioglu, P., Kurban, R., & Besdok, E. (2014). Comparison of evolutionary and swarm based computational techniques for multilevel color image thresholding. *Applied Soft Computing Journal*, 23. <https://doi.org/10.1016/j.asoc.2014.05.037>

Li, C. H., & Lee, C. K. (1993). Minimum cross entropy thresholding. *Pattern Recognition*, 26(4), 617–625. [https://doi.org/10.1016/0031-3203\(93\)90115-D](https://doi.org/10.1016/0031-3203(93)90115-D)

Liu, Y., Mu, C., Kou, W., & Liu, J. (2015). Modified particle swarm optimization-based multilevel thresholding for image segmentation. *Soft Computing*, 19(5). <https://doi.org/10.1007/s00500-014-1345-2>

Manikandan, S., Ramar, K., Willjuice Iruthayarajan, M., & Srinivasagan, K. G. (2014). Multilevel thresholding for segmentation of medical brain images using real coded genetic algorithm. *Measurement: Journal of the International Measurement Confederation*, 47(1). <https://doi.org/10.1016/j.measurement.2013.09.031>

Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 50–60.

- Mirjalili, S. (2015a). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, 228–249. <https://doi.org/10.1016/j.knosys.2015.07.006>
- Mirjalili, S. (2015b). The Ant Lion Optimizer. *Advances in Engineering Software*, 83, 80–98. <https://doi.org/10.1016/j.advengsoft.2015.01.010>
- Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Mittal, A., Soundararajan, R., & Bovik, A. C. (2013). Making a “completely blind” image quality analyzer. *IEEE Signal Processing Letters*, 20(3). <https://doi.org/10.1109/LSP.2012.2227726>
- Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., & Perez-Cisneros, M. (2013). Multilevel thresholding segmentation based on harmony search optimization. *Journal of Applied Mathematics*, 2013. <https://doi.org/10.1155/2013/575414>
- 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 Systems with Applications*, 79, 164–180. <https://doi.org/10.1016/j.eswa.2017.02.042>
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66.
- Portes de Albuquerque, M., Esquef, I. A., Gesualdi Mello, A. R., & Portes de Albuquerque, M. (2004). Image thresholding using Tsallis entropy. *Pattern Recognition Letters*, 25(9), 1059–1065. <https://doi.org/10.1016/j.patrec.2004.03.003>
- Rahkar Farshi, T., & K. Ardabili, A. (2021). A hybrid firefly and particle swarm optimization algorithm applied to multilevel image thresholding. *Multimedia Systems*, 27(1), 125–142. <https://doi.org/10.1007/s00530-020-00716-y>
- Rodríguez-Esparza, E., Zanella-Calzada, L. A., Oliva, D., Heidari, A. A., Zaldivar, D., Pérez-Cisneros, M., & Foong, L. K. (2020). An efficient Harris hawks-inspired image segmentation method. *Expert Systems with Applications*, 155. <https://doi.org/10.1016/j.eswa.2020.113428>
- Sahoo, P., Wilkins, C., & Yeager, J. (1997). Threshold selection using Renyi’s entropy. *Pattern Recognition*, 30(1), 71–84. [https://doi.org/10.1016/S0031-3203\(96\)00065-9](https://doi.org/10.1016/S0031-3203(96)00065-9)
- Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1), 146–165. <https://doi.org/10.1117/1.1631316>
- Sharma, A., Chaturvedi, R., & Bhargava, A. (2022). A novel opposition based improved firefly algorithm for multilevel image segmentation. *Multimedia Tools and Applications*, 81(11). <https://doi.org/10.1007/s11042-022-12303-6>
- Tarkhaneh, O., & Shen, H. (2019). An adaptive differential evolution algorithm to optimal multi-level thresholding for MRI brain image segmentation. *Expert Systems with Applications*, 138. <https://doi.org/10.1016/j.eswa.2019.07.037>
- Tuba, E., Alihodzic, A., & Tuba, M. (2017). Multilevel image thresholding using elephant herding optimization algorithm. *2017 14th International Conference on Engineering of Modern Electric Systems, EMES 2017*. <https://doi.org/10.1109/EMES.2017.7980424>
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4). <https://doi.org/10.1109/TIP.2003.819861>

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>

Ye, Z. W., Wang, M. W., Liu, W., & Chen, S. Bin. (2015). Fuzzy entropy based optimal thresholding using bat algorithm. *Applied Soft Computing Journal*, 31. <https://doi.org/10.1016/j.asoc.2015.02.012>