Mansoura Journal for Computer and Information Sciences (2025), Vol.**20**, No.**1**
doi:10.21608/mjcis.2025.326655.1008

Mansoura Journal for Computer and Information Sciences

ARTICLE

A Novel Newton Raphson Based Optimizer for Tomato Leaf Image Segmentation

Abdelmawla Yousef,[*,1] Ali I. Siam,[1,2] Sherif I. Barakat,[3] and Reham R. Mostafa[4,3]

[1]Department of Embedded Network Systems Technology, Faculty of Artificial Intelligence, Kafr Elsheikh University, Kafr El-Sheikh, Egypt
[2]Sustainable Engineering Asset Management (SEAM) research group, University of Sharjah, Sharjah 27272, United Arab Emirates
[3]Information Systems Department, Faculty of Computers and Information Sciences, Mansoura University, Mansoura 35516, Egypt
[4]Research Institute of Sciences and Engineering (RISE), University of Sharjah, Sharjah 27272, United Arab Emirates
*Corresponding author: abdelmawla_yossef@ai.kfs.edu.eg

(Received: 10 October 2024; Accepted: 3 March 2025; Published: 6 March 2025)

Abstract

Image segmentation plays a pivotal role in computer vision applications, particularly within the agriculture sector, where it plays a critical role in applications such as the early diagnosis of plant diseases. Traditional color image thresholding methods face challenges in determining optimal thresholds, especially for large number of thresholds, that negatively impacts the accuracy of the segmentation process. In this paper, we propose a Newton Raphson-Based Optimizer (NRBO), a novel algorithm for multi-level threshold image segmentation using Kapur's entropy as the objective function. The proposed NRBO incorporates two key components: the Newton Raphson Search Rule (NRSR) to enhance convergence speed and the Trap Avoidance Operator (TAO) to prevent local optima. These components improve the algorithm exploration and exploitation capabilities. The proposed NRBO was applied to segment ten images of tomato leaf diseases. The performance of the proposed NRBO was compared against other optimization algorithms, including Reptile Search Algorithm, Ant Lion Optimizer, Atom Search Optimizer, and Black Widow Optimizer. Experimental results demonstrated that the proposed NRBO consistently achieved superior performance across evaluation metrics, including Feature Similarity Index (FSIM), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM), and CPU time, while maintaining competitive computational efficiency. For instance, NRBO achieved the highest FSIM, PSNR, and SSIM values for the majority of tested images, indicating its robustness in handling multi-level threshold segmentation under various conditions. The results highlight the effectiveness of NRBO in addressing image segmentation challenges, making it a promising solution for diagnosing plant diseases.

Keywords: Image Segmentation, Newton Raphson Based Optimizer (NRBO), Metaheuristic Optimization Algorithms in Agriculture, Tomato Leaves Disease Detection

1. Introduction

The accurate detection and treatment of plant' diseases in the early stages play an essential role in preserving crop production and quality, as the quantity and quality of yield production have a great effect on people's life. The Food and Agriculture Organization (FAO) reported that pests lead to an annual reduction of up to 40% in global crop production, and crop diseases result in economic losses exceeding $220 billion worldwide annually [1]. Therefore, the identification of these diseases must be accurate and timely. Traditional plant disease detection relies on visual inspection by experts, which is labor-intensive, time-consuming, and costly, especially for large-scale farming. In many regions, farmers lack access to expert consultation, further hindering effective disease management. To address these challenges, automatic plant disease detection offers a promising solution. By analyzing images of plant leaves, this technology enables early identification and diagnosis of diseases, reducing reliance on manual inspection and expert intervention. This automated approach also supports advanced agricultural practices, such as machine vision-based control, inspection, and robotic operations [2], [3], [4].

Traditional visual inspection of plants for disease detection is time-consuming, inefficient and unreliable. In contrast, automatic detection techniques offer a more efficient and accurate solution. Common plant diseases, such as brown and yellow spots, early and late scorch, as well as fungal, viral, and bacterial infections, can be identified using image processing techniques. By analyzing digital images of plant leaves, these techniques can quantify the extent of disease damage and differentiate between healthy and diseased tissue based on color variations [5], [2], [6]. A diseased leaf image can be decomposed into three primary regions: the background, the unaffected leaf tissue, and the diseased tissue, which exhibits distinct visual characteristics, including altered color patterns [7]. Multi-threshold image segmentation process was considered an efficient image processing technique for detecting and identifying plant diseases to help farmers take the appropriate actions. This process involves dividing the images into multiple parts based on the properties of the images, like whether it is gray, color, texture or in a geometric shape.

The image segmentation has become a stable and widely used method in recent years [8]. This process has been used in a wide range of applications such as the medical field for diagnosing diseases [9], agriculture [10, 11], satellite images [12] and many other fields. Thresholding is a popular segmentation technique that is characterized by its simplicity, computational efficiency, and robustness to noise and variations in image intensity, it can be classified into two main categories: bi-level and multilevel thresholding [13]. Bi-level thresholding uses only one threshold value to divide an image into two regions: foreground and background. In contrast, multilevel thresholding utilizes multiple threshold values to create multiple distinct regions. While bi-level thresholding is computationally efficient, multilevel thresholding is more computationally demanding, with complexity increasing exponentially as the number of thresholds grows [13]. Multilevel thresholding is more suitable for plant disease detection than binary thresholding, as diseased leaf images typically contain three distinct regions: background, healthy tissue, and diseased tissue.

The Multilevel Thresholding Image Segmentation (MTIS) is extensively employed in medical diagnostics and is regarded as a powerful image processing method [14]. In MTIS technique, it is really important to find the right threshold value to get the best results for detecting and separating objects accurately [15]. The MTIS is considered an NP-hard problem. To find the best thresholds to fix the MTIS issue, Meta-Heuristic algorithms (MHA) are one strategy that has shown promising results [15]. MHAs have traits like being flexible, easy to use, not requiring derivatives, and avoiding Local Optima that rendering them more desirable than conventional optimization methods [16]. As a re-

sult, currently, MHAs have been widely used to address complex optimization problems. MHAs are primarily derived from biological behaviors or inherent natural physical phenomena [17]. MHAs are categorized into three main classes: Swarm-based Algorithms (SAs), Physical based Algorithms (PAs), and Evolution-based Algorithms (EAs), based on the behaviours they mimic from nature [18].

SA is a type of algorithms that uses the actions of many people working together to solve difficult problems. These include some algorithms like Particle Swarm Optimizer (PSO) [19], Moth Flame Optimization (MFO) [20], Slime Mould Algorithm (SMA) [21], Harris Hawks Optimizer (HHO) [22], Firefly Algorithm (FA) [23], Grey Wolf Optimizer [16], Hunger Games Search (HGS) [24], Colony Predation Algorithm (CPA) [25], Cuckoo Search (CS) [26], and Whale Optimization Algorithm (WOA) [27]. On the other hand, PAs imitate fundamental principles of physics and chemical processes, such as Simulated Annealing, RIME optimization algorithm(RIME) [28], Sine Cosine Algorithm (SCA) [29], Weighted MeaN oF VectOrs(INFO) [30], Runge Kutta Optimizer(RUN) [31] and Multi-Verse Optimizer (MVO) [32]. The Genetic Algorithm (GA) [33] and Differential Evolution (DE) [34, 35] also belong to evolutionary algorithms.
The plant diseases were identified using image segmentation techniques. These techniques have some problems that need to be solved using optimization algorithms. The primary contribution of this paper lies in the development and application of a modified version of the Newton Raphson-Based Optimizer (NRBO), specifically tailored for solving the challenging problem of multilevel threshold image segmentation within the agricultural domain. The traditional NRBO algorithm suffers from some drawbacks, such as a shortage in the exploration phase, low convergence rate, and local optima problems. In this paper, two basic rules are employed to solve these problems and improve the performance of the NRBO algorithm. These rules are the Newton Raphson Search Rule (NRSR), which is used for improving the exploration phase and increasing the convergence rate for finding new positions for new solutions, and the Trap Avoidance Operator (TAO), which is used to escape from the local optima. The modified NRBO was used with Kapur's entropy as an objective function for color image segmentation of tomato leaf diseases. The contributions of this work are summarized as follows:

- The NRBO is proposed to solve the image segmentation method for tomato disease detection, utilizing Kapur's objective function.
- The proposed NRBO was combined with NRSR and TAO search rules to balance the exploration and exploitation phases and to avoid local optima.
- Evaluating the efficiency of the proposed algorithm on different threshold values.
- Comparing the performance of the proposed algorithm relative to other well-regarded optimization methods, including Reptile Search Algorithm (RSA), Ant Lion Optimizer (ALO), Atom Search Optimizer (ASO), and Black Widow Optimizer (BWO).

The structure of this paper is as follows: Section 2 presents the related work. Section 3 presents an introduction to the multithreshold image segmentation method. Section 4 discusses the proposed Newton Raphson Based Optimizer. Section 5 presents the dataset and performance measures. Section 6 provides the achieved results and discussion. Section 7 provides the conclusion.

2. Related work

This section reviews existing research works relevant to the proposed work, focusing on the application of metaheuristic optimization algorithms for multilevel threshold image segmentation. Over the last decades, intelligent optimization techniques have showcased their effectiveness in a range of optimization issues. These algorithms have demonstrated promising outcomes in optimizing different problem sets. [36, 37, 38]. Additionally, studies addressing the segmentation of agricultural

images, especially for plant disease diagnosis, are highlighted.

Akay et al. [39] proposed a novel optimization algorithm called Modified Teaching-Learning-Based Artificial Bee Colony (MTLABC). MTLABC is a hybrid approach that combines the strength of TLABC with Levy flights. By incorporating Levy flight's exploration capabilities into TLABC's search equations, MTLABC significantly enhances the optimization performance. For performance evaluation, the proposed algorithm utilized Otsu's function for color image segmentation of plant diseases. Singh and Misra [40] proposed a model for segmenting color images of five common leaf diseases using a genetic algorithm. The fitness function for the algorithm was based on the Euclidean distance between pixels and their assigned clusters. Guo et al. [41] proposed an improvement for Aquila Optimizer (IAO) by incorporating chaotic mapping and elite dimension selection to overcome its drawbacks like local optima and low solution accuracy. For performance evaluation, authors deployed IAO for solving multi-threshold image segmentation problems.

Sushil et al. [42] suggested a modification for particle swarm optimizer (PSO) to mitigate the premature convergence problem inherent, and introduce a novel optimization algorithm that decomposes the high-dimensional search space into multiple one-dimensional subspaces. Applying a specific strategy to avoid premature convergence in each subspace enhances the global exploration and exploitation capabilities of the algorithm. The proposed algorithm utilized a minimum cross-entropy function for segmenting grayscale images. The achieved results indicate that the proposed algorithm outperforms current algorithms in terms of segmentation accuracy. Xing et al. [43] introduced an improved algorithm based on Emperor Penguin Optimization (EPO) that incorporates Gaussian mutation and reverse learning to strike a balance between local and global search. This enhanced algorithm has shown promise in solving image thresholding segmentation issues. The same in [44], Ray et al. developed a hybrid algorithm by combining the WOA with the Eagle strategy, using fuzzy entropy, Kapur's entropy, and cross entropy for assessment. This algorithm effectively segments pathological and threshold color images.

Houssein et al. [45] developed a novel BWO to optimize Kapur's and Otsu's objective functions for multi-level image thresholding. These traditional methods, while effective for bi-level thresholding, become computationally expensive for multi-level thresholding. The proposed BWO is evaluated on various benchmark images and compared against other metaheuristic algorithms, demonstrating superior performance in terms of segmentation quality and computational efficiency. Wang et al. [46] introduced a novel three-stage image repairing method to enhance the performance of current image segmentation algorithms for maize diseases. Initially, a binary model using color features is built to locate potential leaf areas. In the second stage, a random forest classifier is trained on a set of sample points from broken leaf areas to accurately identify repair points. The final stage utilizes a super-pixel based approach to repair broken leaves by extracting their skeletons and aligning corresponding broken edges. In [47], Anitha et al. enhanced the WOA by adjusting the cosine function to control whale positions and introducing a correction factor to manage whale movements effectively. This upgraded algorithm has delivered positive results within five thresholds.

Zhang et al. [10], improved the Aptenodytes Forsteri optimization algorithm by refining the mutation and oscillation processes. The efficacy of the model is validated through tests on CEC 2017 and Berkeley datasets. Hashim et al. [48] developed Snake Optimizer (SO), that inspired by the foraging and mating behavior of snakes. SO has been applied to address various optimization problems, including combination forecasting, production scheduling, feature extraction, and electrical engineering. Numerous enhancements and hybrid algorithms have been developed. Hu et al. [49] introduced the Multi-Strategy Snake Optimizer Algorithm (BEESO), focusing on accelerating the search operation by leveraging insights derived from the best and worst solutions in the updating process to expedite the process. These enhanced algorithms have demonstrated superior results in CEC 2019 and other optimization processes.

Yao et al. [50] enhanced the SO by integrating the reverse learning process and an improved dynamic update algorithm. Benchmark test functions were used to validate the efficiency of the improved algorithm in solving various optimization problems. Wang et al. [51] developed a multi-strategy SO algorithm, with the enhancement being the utilization of an opposition-based learning strategy in the initialization process. This improved algorithm exhibited promising performance in practical solutions. Liu [52] improved the SO using chaotic operations to hasten the convergence speed and enhanced the search ability and prevent the local minima problems using dynamic gaussian distribution. Others utilized enhanced versions for feature selection, for example, Abu Khurma et al. [53] proposed a binary snake optimizer (BSO), that combines greedy crossover operators with SO to enhance the classification process.

The same in [54], Al Shourbaj et al. enhanced the SO algorithm in terms of search strategy, combining the Reptile search strategy with SO for feature selection, demonstrating superiority in feature selection for engineering problems. In [55], Sowmya et al. used the NRBO for solving a continuous optimization problem. It's used for solving some benchmark functions and proved its efficiency due to its high performance of the exploration and exploitation phase. Zhao at al. [56] developed an enhancement for mayfly algorithm by proposing a hybridization with Lévy flight to improve its global exploration and local exploitation abilities. The proposed algorithm is applied to the challenging problem of color image segmentation with multiple threshold levels (K=4,5,6,7). The proposed algorithm outperforms existing methods for the given threshold values. However, its performance may diminish at higher thresholds.

Based on the above analysis, existing image segmentation algorithms face challenges, it is shown that despite being designed for specific segmentation results, these algorithms often suffer from local optima entrapment, slow convergence, and low precision. These challenges may prevent these algorithms from getting the best results, which affect on the quality of the segmented images. Accordingly, this paper proposed a modified NRBO based on NRSR and TAO to balance between the exploration and exploitation phases and to avoid trap into local optima and increase convergence speed.

3. Multi-threshold image segmentation

Kapur's entropy [57] is considered one of the commonly utilized techniques for multithreshold image segmentation based on their shapes. However, the method faces limitations in computational efficiency, primarily due to the suboptimal formulation of the between-class variance, which affects its effectiveness in certain applications.

Through the utilization of a suitable threshold function T, where T is defined as in Eq. (1), while $g(x, y)$ is defined based on $f(x, y)$.

$$g(x,y) = \left\{ \begin{array}{lll} 1 & \text{if} & f(x,y) > T \\ 0 & \text{if} & f(x,y) \leqslant T \end{array} \right\} \tag{1}$$

As T varies across the images, the corresponding thresholding variables can be determined accordingly. If T is neighbourhood dependent of (x, y), then local thresholding can be assessed.

3.1 Bi-level thresholding

Let's suppose an image be represented by L different gray levels. Bi-level thresholding, as specified in Eq. (2), is used. Kapur may be developed for bi-level thresholding as follows:

$$\begin{aligned} M_0 &= \{g(x,y) \in I \mid 0 \leqslant g(x,y) \leqslant t-1\} \\ M_1 &= \{g(x,y) \in I \mid t \leqslant g(x,y) \leqslant L-1\} \end{aligned} \tag{2}$$

3.2 Multilevel thresholding

Thresholding is a process used to convert grayscale images into binary images by assigning pixel values below a specified threshold to zero and those above the threshold to one. In multilevel thresholding, multiple threshold values are used to segment the image into several distinct regions, resulting in an output image composed of multiple parts, as represented in Eq. (3).

$$\begin{aligned} M_0 &= \{g(x,y) \in I \mid 0 \leqslant g(x,y) \leqslant t_1-1\} \\ M_1 &= \{g(x,y) \in I \mid t_1 \leqslant g(x,y) \leqslant t_2-1\} \\ M_i &= \{g(x,y) \in I \mid t_i \leqslant g(x,y) \leqslant t_{i+1}-1\} \\ M_m &= \{g(x,y) \in I \mid t_m \leqslant g(x,y) \leqslant L-1\} \end{aligned} \tag{3}$$

where $t_1(i = 1, ..., m)$, the i th is the threshold value and m is the number of thresholds.

Therefore, the bi-level thresholding is considered much simpler than the multilevel thresholding technique.

3.2.1 Kapur's entropy

Kapur's entropy technique is considered the effective thresholding method for finding the best threshold values for segmenting images [58]. Let the images be divided into different classes based on $[th_1, th_2, \ldots, th_n]$. Then, the objective function for kapur was represented as follows:

$$H(th_1, th_2, \ldots, th_n) = H_0 + H_1 + \ldots + H_n \tag{4}$$

where

$$H_0 = -\sum_{j=0}^{th_1-1} \frac{p_j}{\omega_0} \ln \frac{p_j}{\omega_0}, \omega_0 = \sum_{j=0}^{th_1-1} p_j \tag{5}$$

$$H_1 = -\sum_{j=th_1}^{th_2-1} \frac{p_j}{\omega_1} \ln \frac{p_j}{\omega_1}, \omega_1 = \sum_{j=th_1}^{th_2-1} p_j \tag{6}$$

$$H_n = -\sum_{j=th_n}^{L-1} \frac{p_j}{\omega_n} \ln \frac{p_j}{\omega_n}, \omega_n = \sum_{j=th_n}^{L-1} p_j \tag{7}$$

where $H_0, H_1, ..., H_n$ refers to the entropies of different classes, $w_0, w_1, ..., w_n$ refers to the probability of each class. To obtain the best threshold values, the following equation must be maximized.

$$f_{\text{Kapur}}\ (th_1, th_2, \ldots, th_n) = \arg\max\{H(th_1, th_2, \ldots, th_n)\} \tag{8}$$

The Kapur's entropy may have some drawbacks when used in multilevel thresholding. It is always used with metaheuristic algorithms. In this article, the NRBO is used to support Kapur's objective function.

4. The proposed modified Newton-Raphson-based optimizer

In this section, the concepts of Newton's method will be discussed. Additionally, the main steps of the Newton-Raphson-Based Optimizer (NRBO) will be introduced. The images were collected and the pixel intensity of each image was calculated to obtain the histogram of the three channels for each image. The threshold values were determined and the proposed NRBO was used to optimize and solve the image segmentation problem using Kapur's entropy objective function. Finally, the segmented images were obtained from the determined threshold values. The following figure presents the main steps:

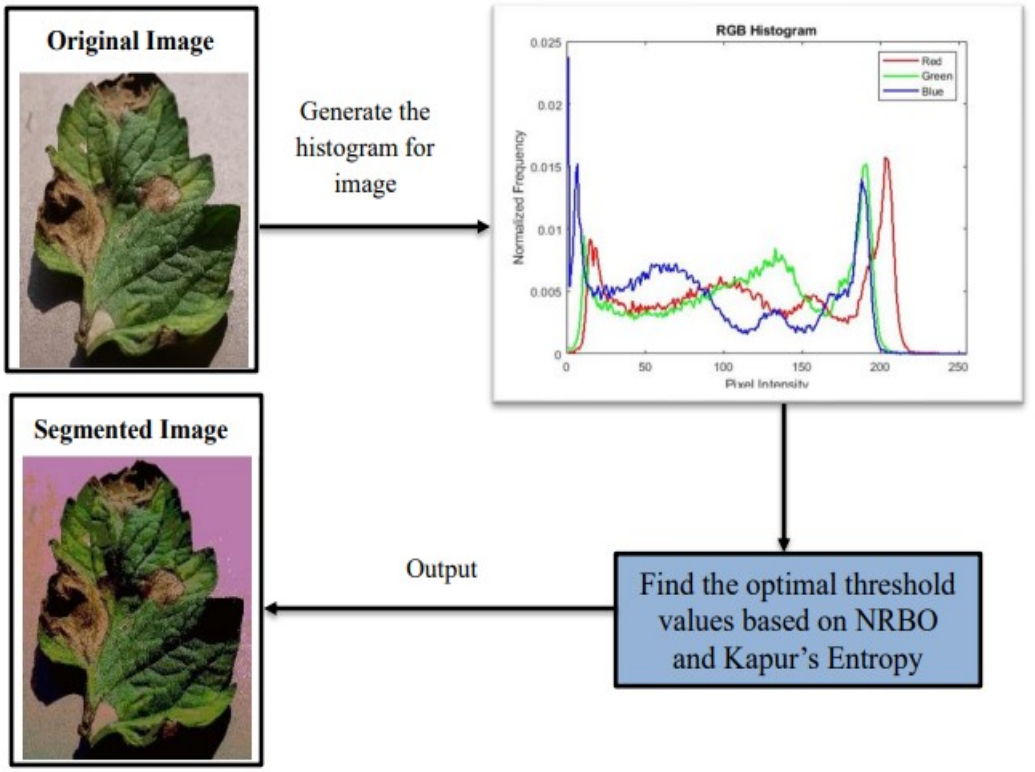


Figure 1. The flowchart of NRBO based MTIS

4.1 Newton-Raphson method

The Newton-Raphson Method (NRM) referred to as Newton's process, is an algorithm used to identify the roots of a function. It achieves this by utilizing the initial terms of the Taylor Series (TS) of the function $f(x)$ in the vicinity of an assumed way to determine the exact location of the root [59, 60]. For a polynomial function f(x), the Newton-Raphson technique (NRM) shares essential similarities with Horner's technique [61]. The NRM starts with a single point (x_0), the NRM uses the TS assessed at x_0 to find another point close to the prior solution. This process is iterated until a suitable answer is discovered. Here we see the TS of $f(x)$ when it comes to the point ($x = x_0 + \epsilon$).

$$f(x_0 + \epsilon) = f(x_0) + f'(x_0)\,\epsilon + \frac{f''(x_0)(\epsilon)^2}{2!} + \frac{f'''(x_0)(\epsilon)^3}{3!} + \cdots \tag{9}$$

Maintaining the second-order parameters,

$$f(x_0 + \epsilon) \approx f'(x_0)\,\epsilon + \frac{f''(x_0)\,(\epsilon)^2}{2} \tag{10}$$

The offset ϵ needed to get closer to the root x_0 from where it originated can be determined from Eq. (10). Suppose $f(x_0 + \epsilon) = 0$, where Eq. (10) was solved when $\epsilon \equiv \epsilon_0$ and resulted,

$$\epsilon_0 = -\frac{f'(x_0)}{f''(x_0)} \tag{11}$$

The second-order adjustment to the root's position is given using Eq. (11), and the following root's position may be computed based on $x_1 = x_0 + \epsilon_0$, and the process will be iterated until the optimal root reached based on Eq. (12).

$$\epsilon_n = -\frac{f'(x_n)}{f''(x_n)} \tag{12}$$

A local maximum or a horizontal asymptote can make this process imbalanced, unfortunately. Even so, starting from the right place each time, the approach can be used to find better approximations.

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}, n = 1, 2, 3, \ldots \tag{13}$$

In NRM, the starting point x0 is approximately zero and guarantees the safety of the algorithm's convergence.

4.2 Newton-Raphson-based optimizer (NRBO)

In NRBO, the search path was defined by using the NRM that was used to detect the search space based on a small number of vector sets and two basic operators like NRSR and TAO for exploring the search space.

4.2.1 Initialization

An optimization problem defined as follows is used for the optimization, which is unconstrained and single-objective.

Minimize :

$$f(x_1, x_2, \ldots, x_n)\; lb \le x_j \le \text{ub}, j = 1, 2, \ldots, \text{dim} \tag{14}$$

where $f(x)$ was the fitness function aimed to maximize, x_i, *dim*, *lb* and *ub* refer to the decision vector, problem's dimension, lower bounds and upper bounds respectively. Similar to other metaheuristic algorithms, NRBO starts by generating random initial solutions within the limits of potential solutions. Given the existence of a certain number of populations, denoted as Np, it can be observed that each population is composed of decision variables or vectors with a certain dimensionality. Hence, the random population is initialized via the Eq. (15).

$$x_j^n = lb + \text{rand} \times (ub - lb), n = 1, 2, \ldots, N_p \text{ and } j = 1, 2, \ldots, \text{dim} \tag{15}$$

where x_j^i indicates where the $j^t h$ dimension of the $n^t h$ population is located and the *rand* indicates to a random numbers between (0, 1). The population matrix, which provides a comprehensive depiction of populations across all dimensions was depicted as in the following Eq. (16).

$$\boldsymbol{X_n} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{dim}^1 \\ x_1^2 & x_2^2 & \cdots & x_{dim}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{N_v} & x_2^{N_p} & \cdots & x_{dim}^{N_v} \end{bmatrix}_{N_p \times \dim} \tag{16}$$

4.2.2 Newton-Raphson Search Rule (NRSR)

NRSR controls the vectors, enabling a more precise exploration of the feasible region and achieving superior placements. The NRSR is predicated on the idea that the NRM is introduced to accelerate convergence and enhance the exploration inclination. In circumstances when many optimisation strategies lack differentiability, a mathematical NRM is employed to substitute the explicit formulation of the function. The NRM moves to the next place with a specific direction based on the starting of the initial solution. To obtain the NRSR from Eq. (13), it is necessary to calculate the second-order derivative using the TS, where the TS of $f(x - \Delta x)$ and $f(x + \Delta x)$ is discussed as the following equations:

$$f(x + \Delta x) = f(x) + f'(x_0)\Delta x + \frac{1}{2!}f''(x_0)\Delta x^2 + \frac{1}{3!}f'''(x_0)\Delta x^3 + \cdots \tag{17}$$

and

$$f(x - \Delta x) = f(x) - f'(x_0)\Delta x + \frac{1}{2!}f''(x_0)\Delta x^2 - \frac{1}{3!}f'''(x_0)\Delta x^3 + \cdots \tag{18}$$

The formulas $f'(x)$ and $f''(x)$ are determined by subtracting Eq. 17 from Eq. 18.

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \tag{19}$$

and

$$f''(x) = \frac{f(x + \Delta x) + f(x - \Delta x) - 2 \times f(x)}{\Delta x^2} \tag{20}$$

By substituting Eq. (19) and Eq. (20) into Eq. (13), the new position of the root is expressed as following.

$$x_{n+1} = x_n - \frac{(f(x_n + \Delta x) - f(x_n - \Delta x)) \times \Delta x}{2 \times (f(x_n + \Delta x) + f(x_n - \Delta x) - 2 \times f(x_n))} \tag{21}$$

The NRSR considered the first component in NRBO, hence, some modifications must be taken to manage the search of population. From the results of Eq. (21), the x's positions are adjacent and indicated by $x_n + \Delta x$ and $x - \Delta x$, respectively. NRBO generates two additional solutions using the neighboring locations.

$$\text{NRSR} = \text{randn} \times \frac{(X_w - X_b) \times \Delta x}{2 \times (X_w + X_b - 2 \times x_n)} \tag{22}$$

where *rand* is a standard normal random variable with a mean of 0 and a variance of 1, X_w and X_b refer to the worst and best positions respectively.

The suggested method must strike a balance of diversification and intensification to effectively find the best possible outcomes in the search space and subsequently converge to the global result.

The method may be improved by using a flexible coefficient known as δ. The equation for δ is given in Eq.23.

$$\delta = \left(1 - \left(\frac{2 \times IT}{Max\_IT}\right)\right)^5 \tag{23}$$

where *IT* refers to the current iteration, *MaxIT* refers to the upper bound for iterations. For better balancing between the both exploration and exploitation phases. The parameter δ adjusts automatically according to iterations.

In addition to the adaptive parameter δ, the proposed Non-Randomized Stochastic Ranking (NRSR) method enhances the Non-Randomized Binary Optimisation (NRBO) by incorporating random actions into the optimisation process. This leads to more diversification and prevents the algorithm from getting stuck in local optima. Furthermore, the NRSR method achieves a substantial reduction in the number of iterations required. The Δx can be computed as following:

$$\Delta x = \text{rand}(1, \text{dim}) \times \left| X_b - X_n^{IT} \right| \tag{24}$$

where the best obtained solution is represented by the X_b and $rand(1 : dim)$ refers to the random number with a *dim* variables. Then, the Eq. (21) was modified based on the NRSR as following:

$$x_{n+1} = x_n - NRSR \tag{25}$$

The enhancement of the suggested NRBO is achieved by incorporating an additional parameter, referred to as ρ, which effectively directs the population to the desired location. The formula for it is provided as follows.

$$\rho = a \times \left(X_b - X_n^{IT} \right) + b \times \left(X_{r_1}^{IT} - X_{r_2}^{IT} \right) \tag{26}$$

where the random numbers between $(0, 1)$ was represented based on two variables called a and b, there are two not equal numbers that selected from the population called r_1 and r_2. In the following equation, the current vector's position has been updated.

$$\begin{aligned} X1_n^{IT} = x_n^{IT} - \left(\text{randn} \times \frac{(X_w - X_b) \times \Delta x}{2 \times (X_{\text{w}} + X_b - 2 \times X_n)} \right) \\ + \left(a \times \left(X_b - X_n^{IT} \right) + b \times \left(X_{r_1}^{IT} - X_{r_2}^{IT} \right) \right) \end{aligned} \tag{27}$$

where $X1_n^{IT}$ is the obtained new vector position. The NRSR was enhanced NRM that suggested [62, 63] and the Eq. (21) was changed as following:

$$\text{NRSR} = \text{rand}\, n \times \frac{\left(y_\Psi - y_b \right) \times \Delta x}{2 \times \left(y_\Psi + y_b - 2 \times x_n \right)} \tag{28}$$

$$y_w = r_1 \times (\text{Mean}\, (Z_{n+1} + x_n) + r_1 \times \Delta x) \tag{29}$$

$$y_b = r_1 \times (\text{Mean}\, (Z_{n+1} + x_n) - r_1 \times \Delta x) \tag{30}$$

$$Z_{n+1} = x_n - \text{rand}\, n \times \frac{(X_w - X_b) \times \Delta x}{2 \times (X_w + X_b - 2 \times x_n)} \tag{31}$$

where the vector's two locations that created based on $Z_n + 1$ and X_n represented by y_w and y_b and r_1 refers to a random number in the range of $(0, 1)$. The newely improved version of the NRSR is shown in Eq. (28). Based on Eq. (28), the Eq. (27) was changed as following:

$$\begin{aligned} X1_n^{IT} = x_n^{IT} - \left(\text{rand}\, n \times \frac{(y_w - y_b) \times \Delta x}{2 \times (y_w + y_b - 2 \times x_n)} \right) \\ + \left(a \times \left(X_b - X_n^{IT} \right) + b \times \left(X_{r_1}^{IT} - X_{r_2}^{IT} \right) \right) \end{aligned} \tag{32}$$

The new vector $X2_n^{IT}$ must be constructed by exchanging the best vector's location known as X_b based on the current vector's position as in Eq. (32).

$$\begin{aligned} X2_n^{IT} &= x_b - \left(\text{rand } n \times \frac{(y_w - y_b) \times \Delta x}{2 \times (y_w + y_b - 2 \times x_n)}\right) \\ &+ \left(a \times \left(X_b - X_n^{IT}\right) + b \times \left(X_{r_1}^{IT} - X_{r_2}^{IT}\right)\right) \end{aligned} \tag{33}$$

The exploitation phase is considered the main part of the search process. Regarding local search, the search method suggested in Eq. (33) is effective, but it has constraints when applied to global search. Conversely, the search strategy outlined in Eq. (32) is effective for global search but has restrictions for local search. Nevertheless, the NRBO utilizes both Eq. (32) and Eq. (33) to enhance both the diversification and intensification stages. The subsequent iteration represents the updated position vector, as indicated by Eq. (34).

$$X_n^{IT+1} = r_2 \times (r_2 \times X1_n^{IT} + (1 - r_2) \times X2_n^{IT}) + (1 - r_2) \times X3_n^{IT} \tag{34}$$

$$X3_n^{IT} = X_n^{IT} - \delta \times (X2_n^{IT} - X1_n^{IT}) \tag{35}$$

where r_2 indicates to a random number in the range of $(0, 1)$.

4.3 Trap Avoidance Operator (TAO)

The inclusion of the TAO aims to enhance the efficiency of the suggested NRBO in addressing practical problems. The TAO is a modified and improved operator that derived from [64]. It used to change the position of X_n^{IT+1}. The produced solution has a good quality X_{TAO}^{IT} by fusing the optimal position X_b with the current vector position X_n^{IT}. The X_{TAO}^{IT} is resulted when the rand's value is less than DF based on the following equations.

$$\begin{cases} X_{TAO}^{IT} = X_n^{IT+1} + \theta_1 \times \left(\mu_1 \times x_b - \mu_2 \times X_n^{IT}\right) + \theta_2 \times \delta \\ \times \left(\mu_1 \times \text{Mean}\left(X^{IT}\right) - \mu_2 \times X_n^{IT}\right), \text{ if } \mu_1 < 0.5 \\ X_{TAO}^{IT} = x_b + \theta_1 \times \left(\mu_1 \times x_b - \mu_2 \times X_n^{IT}\right) + \theta_2 \times \delta \\ \times \left(\mu_1 \times \text{Mean}\left(X^{IT}\right) - \mu_2 \times X_n^{IT}\right), \text{ Otherwise} \end{cases} \tag{36}$$

$$X_n^{IT+1} = X_{TAO}^{IT} \tag{37}$$

where *rand* refers to a random number between $(0, 1)$, the random numbers between $(-1, 1)$ and $(-0.5, 0.5)$ were represented by θ_1 and θ_2, respectively. The NRBO's performance may be controlled based on a deciding factor, μ_1 and μ_2 are also random numbers that created based on the following equations.

$$\mu_1 = \begin{cases} 3 \times \text{ rand }, & \text{if } \Delta < 0.5 \\ 1, & \text{Otherwise} \end{cases} \tag{38}$$

$$\mu_1 = \begin{cases} \text{rand }, & \text{if } \Delta < 0.5 \\ 1, & \text{Otherwise} \end{cases} \tag{39}$$

where the *rand* indicates a random number and Δx refers to a random number. The Eqs. (38) and (39) were streamlined based on the following equations.

$$\mu_1 = \beta \times 3 \times \text{ rand } + (1 - \beta) \tag{40}$$

$$\mu_1 = \beta \times \text{ rand } + (1 - \beta) \tag{41}$$

where β indicates a binary number that may be 0 or 1. The $\beta's$ value will be 0 if the $\Delta's$ value is greater than or equal to 0.5. Alternatively, the value will be 1. As a result of the random selection of μ_1 and μ_2 parameters, the population has more differentiation and avoids local solutions. The following **Algorithm** introduces the pseudocode of NRBO and Figure 2 presents the flowchart of the proposed NRBO.

Algorithm 1 Pseudocode of NRBO

Require: Population size: N, Max iterations: $MaxIT$, current iteration: IT, lower bound: LB, upper bound: UB, Dimension: dim.
Ensure: The optimal solution
Initialize population positions X_i for $i = 1, 2, \ldots, N$ randomly.
Evaluate fitness of each X_i and determine best (X_b) and worst (X_w) solutions.
for $IT = 1$ to Max_IT **do**
  for $n = 1$ to N **do**
    for $j = 1$ to dim **do**
      Select random integers $r_1, r_2 \in \{1, 2, \ldots, N\}$ where $r_1 \neq r_2 \neq N$.
      Calculate x_n^{IT+1} using Equation 34.
    end for
    ———-**Trap Avoidance Operator (TAO):**———-
    if $rand() < DF$ **then**
      Calculate X_{TAO}^{IT} using Equation 36.
      $x_n^{IT+1} = X_{TAO}^{IT}$.
    else
      $x_n^{IT+1} = x_n^{IT+1}$ (Select the solution using Eq. 34).
    end if
    Update X_b and X_w based on the new x_n^{IT+1}.
  end for
end for
Termination: Return the best solution found, X_b.

.

5. Experimental setup

5.1 Test dataset

The benchmarked dataset is publicly available, and it can be accessed from: https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset. This dataset includes about 38 classes of disease types for different plants. In our work, we selected ten random images of different tomato leaf diseases and their histogram to analyze the distribution of pixel intensities and to understand the nature of image content. Figure 3 shows samples of randomly selected tomato leaves diseases.

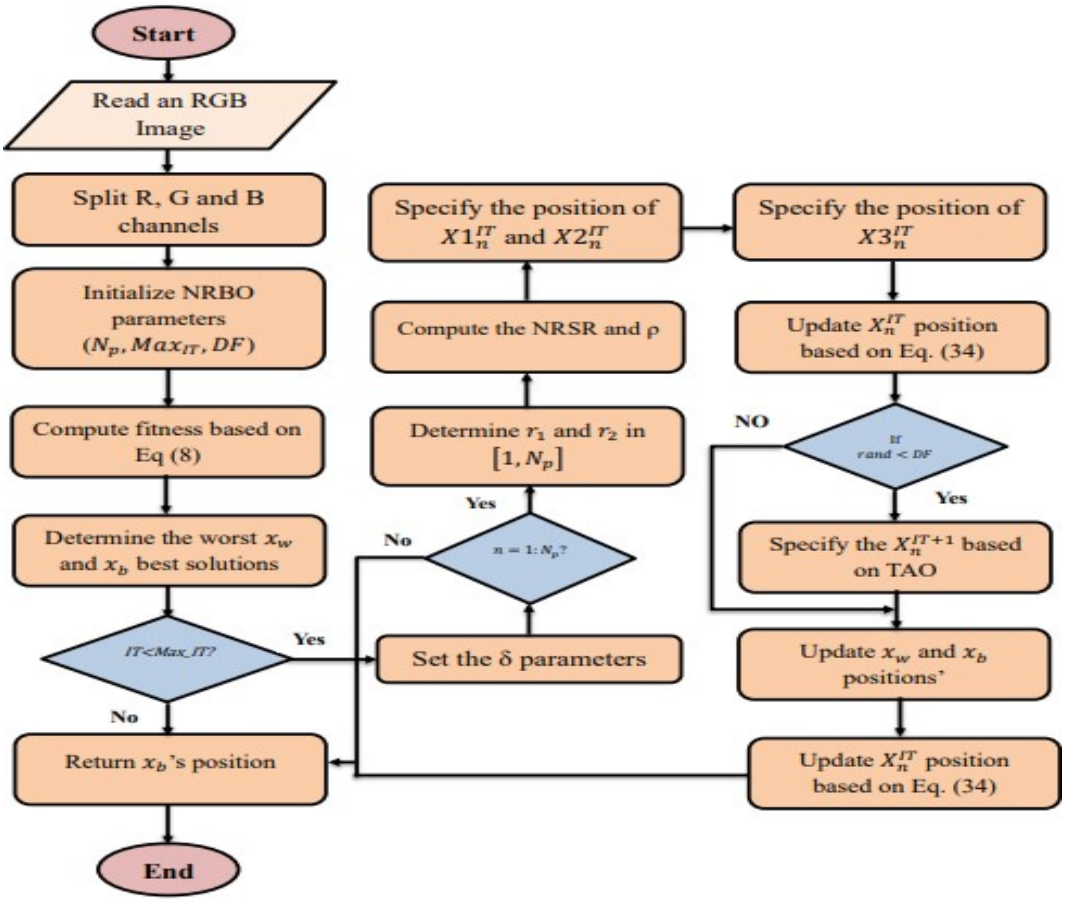


Figure 2. Flowchart of the proposed algorithm for image segmentation

5.2 Environmental setup

Experiments have been implemented on a Lambda RTX 2080Ti server (12 cores 2.1 GHz) and 256 GB RAM.

5.3 Performance measures

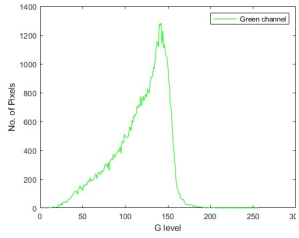
For evaluation of multilevel thresholding of the proposed approach and other metaheuristic optimization algorithms, some metrics have been employed including peak signal to noise ratio (PSNR), structural similarity index measure (SSIM), feature Similarity (FSIM), standard deviation (Std), fitness value, and the CPU time. These metrics are described as follows:

The PSNR value is computed as follows:

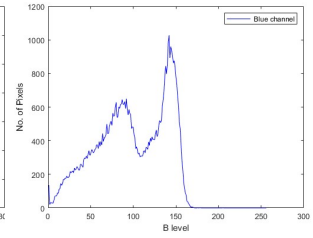
$$PSNR = 10\log_{10}\left(\frac{255}{MSE}\right), \text{ with mean squared error} \tag{42}$$

$$MSE = \frac{1}{M \times N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[I(x,y) - I'(x,y)]^2 \tag{43}$$

Image 1 R component

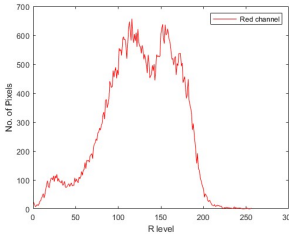


G component

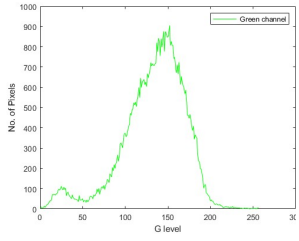


B component

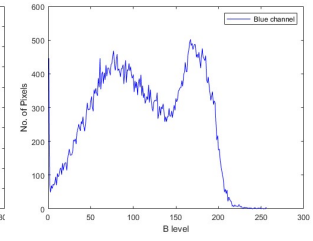
Image 2



R component

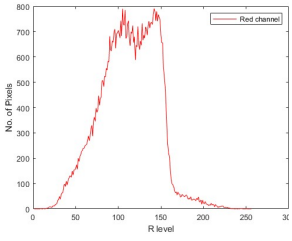


G component

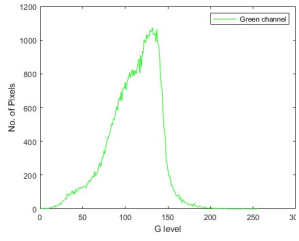


B component

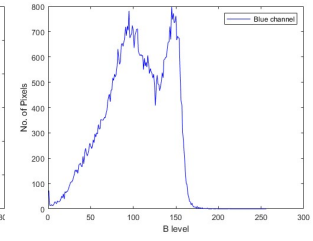
Image 3



R component

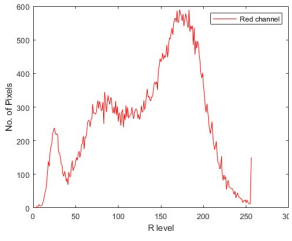


G component

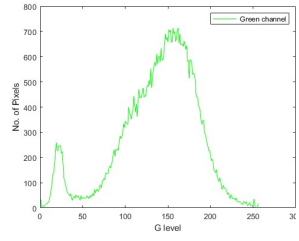


B component

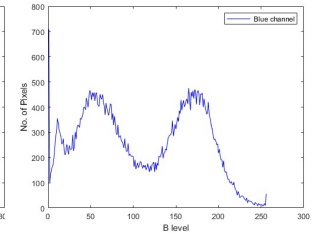
Image 4



R component



G component



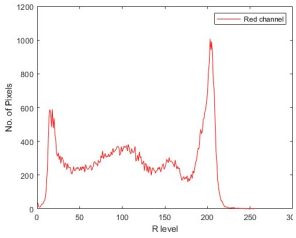
B component

where M and N refer to the height and width of the image, respectively, I(x,y) is the pixel's value of the input image that located at (x,y), and $I'(x, y)$ refers to the segmented image value at (x,y).

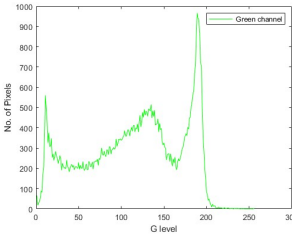
The SSIM value is computed as follows:

$$\mathrm{SSIM}(I, \mathrm{Seg}) = \frac{\left(2\mu_I \mu_{\mathrm{seg}} + c_1\right)\left(2\sigma_{ISeg} + c_2\right)}{\left(\mu_I^2 + \mu_{\mathrm{Seg}^2} + c_1\right)\left(\sigma_I^2 + \sigma_{\mathrm{Seg}^2 + c_2}\right)} \tag{44}$$

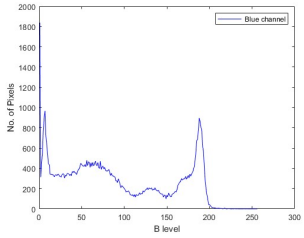
Image 5



R component

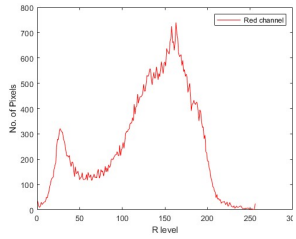


G component

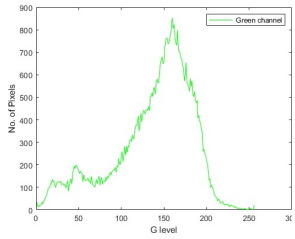


B component

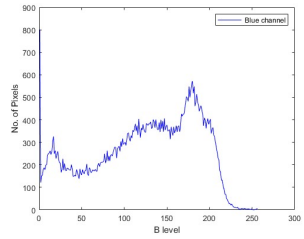
Image 6



R component

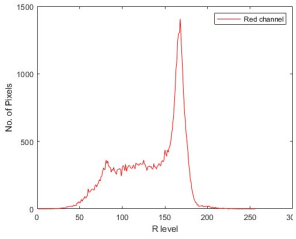


G component

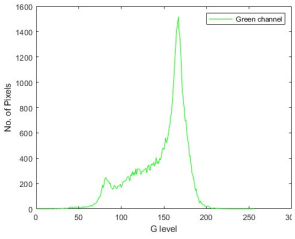


B component

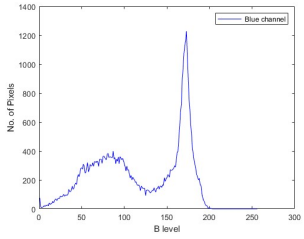
Image 7



R component

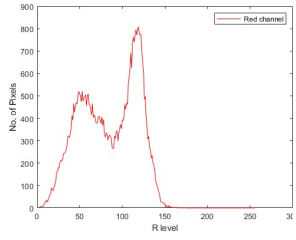


G component

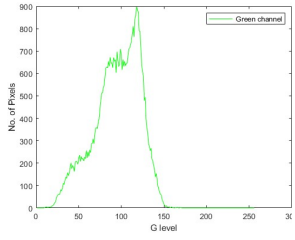


B component

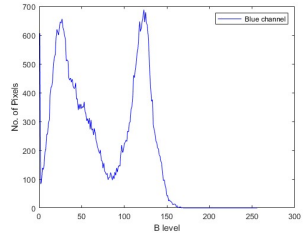
Image 8



R component



G component



B component

where μ_{Seg} and μ_I refer to the average gray value for the segmented and original images, respectively, and σ_{Seg} and σ_I refers to the standard deviations of the segmented and original images, respectively. The constants c_1 and c_2 are regularization constants, and computed as follows:

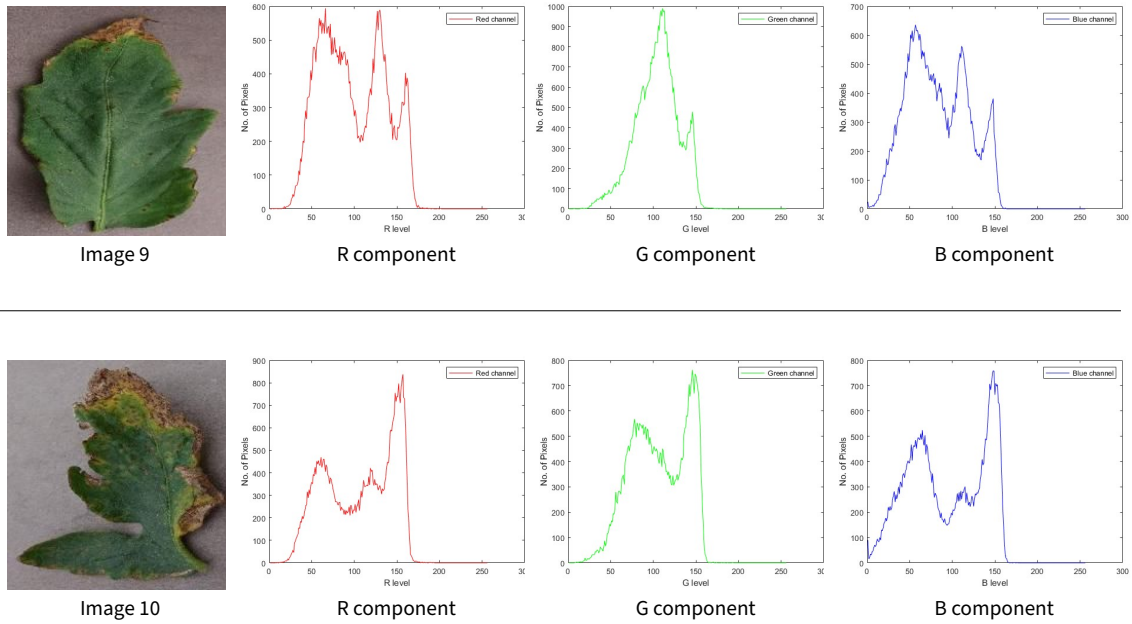


Figure 3: A set of benchmarked images from the dataset and their histograms

$$c_1 = (0.01^* \text{ L})^2 , c_2 = (0.03^* \text{ L})^2 \tag{45}$$

where L is the dynamic range value (255 in our case). This results in the values of c_1=6.5025 and c_2=58.52252.

The FSIM is calculated by:

$$FSIM = \frac{\sum_{w\in\Omega} S_L(w)PC_m(w)}{\sum_{w\in\Omega} PC_m(w)} \tag{46}$$

where Ω refers to the entire domain of the image. Additionally, for computing the mean of the fitness value, the following equation is used:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{47}$$

where N refers to execution runs, and x_i refers to the fitness value in $i-th$ iterations.

The standard deviation (Std) is calculated as follows:

$$std = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{48}$$

where μ is the mean value obtained over runs of the algorithm.

The CPU time provides the average CPU time required by each algorithm to segment all images, and is computed by the tic-toc function in Matlab

6. Results and discussion

6.1 Kapur's entropy results

In this section, the results obtained using the proposed algorithm and other benchmarked algorithms across various multithreshold metrics are presented. For performance evaluation, each algorithm was executed 25 times per image, with 350 iterations per run, and a population size of 40 for all algorithms. The values of FSIM, PSNR, and SSIM metrics are summarized in Tables 1, 2, and 3, respectively. The proposed algorithm achieved the best results because it has two search rules NRSR and TAO that helped in getting the optimal results. The optimal results were determined when the mean value of each evaluation metric increased and the std values decreased. The average fitness values for each algorithm are illustrated in Figure 4, where the proposed NRBO outperformed the compared algorithms, achieving the highest fitness values.

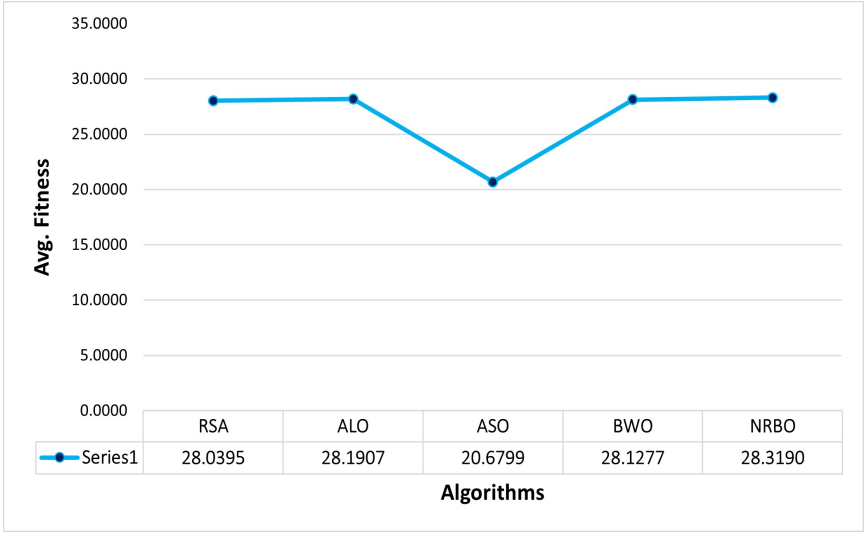


Figure 4. Comparison among various algorithms in terms of average fitness values

The compared metaheuristic algorithms showed higher results than the proposed one in some cases because of the selected threshold values where these values were optimal with the compared algorithms than with the proposed algorithm with some images, but our algorithm has the best results in most cases as shown in the results table in bold format. The FSIM's, PSNR's, SSIM's metric results were introduced in the table 1, 2, 3 respectively.

Table 1 presents the FSIM metric results, demonstrating that the proposed algorithm achieved the highest values three times for Image 1 with threshold values k= 5, 6, and 7. However, at k= 8, the BWO algorithm obtained the highest value, positioning the proposed algorithm as the overall top performer for this image. Similarly, for Image 2, the proposed algorithm ranked first, achieving the best performance at k=5, 7, and 8, while BWO outperformed at k=6.

For image 3, the best results were achieved using the proposed algorithm with all used threshold values. For image 4, the proposed NRBO algorithm achieved the highest results when k=5, 6, and 7, while the ALO algorithm achieved the best result when k=8. For Image 5, the best achieved results were obtained with the proposed algorithm three times when k=5, 6, and 8, while the BWO algorithm achieved the best result when k=7. For Image 6, the NRBO and RSA algorithms performed equally, each achieving the best results twice. Specifically, NRBO demonstrated superior performance at k=6 and 7, while RSA achieved the highest values at k=5 and 8. For Image 7, the proposed NRBO achieved

the best results two times when k=7 and 8 while the BWO algorithm achieved the best result when k=5 and the RSA algorithm achieved the best when k=6. For Image 8, the NRBO and RSA algorithms performed equally, where each algorithm achieved the best results two times. Specifically, NRBO demonstrated superior performance at k=5 and 6, while RSA achieved the highest values at k=7 and 8. For Image 9, the proposed NRBO came with the first rank where it achieved the best results three times when k=5, 6 and 7, while the RSA algorithm achieved the best result only one time when k=5. Similarily, For Image 10, the proposed NRBO came with the first rank where it achieved the best results three times when k=5, 6 and 7, while the RSA algorithm achieved the best result only one time when k=5. Based on the discussion of FSIM metrics, it is evident that the proposed NRBO delivered the best results for multilevel threshold segmentation across the majority of tested scenarios.

Table 2 presents the PSNR results which demonstrates that the proposed NRBO algorithm achieved the highest values three times for Image 1 at k= 5, 6, and 8, while the BWO algorithm outperformed at k= 7, positioning NRBO as the overall top algorithm. Similarly, for Image 2, the proposed algorithm achieved the best results at k=5, 7, and 8, while BWO obtained the highest value at k=6.

For Image 3, NRBO delivered superior results across all threshold values. For Image 4, NRBO achieved the highest PSNR values at k=6 and 7, whereas RSA and ALO performed best at k=5 and 8, respectively. For Image 5, the proposed algorithm produced the best results at k=5, 6, and 8, while BWO outperformed at k=7. For Image 6, NRBO achieved the best PSNR values for all tested threshold values.

For Image 7, the proposed NRBO achieved the best PSNR results with k=7 and 8, and the BWO and RSA achieved the best when k=5 and 6 respectively. For Image 8, the proposed NRBO achieved the best PSNR results with all threshold levels except when k=7, the RSA is considered the best.

For Image 9, the proposed NRBO achieved the best results with k=5, 6, and 7 while the RSA achieves the best only one time when k=8. For Image 10, the proposed NRBO achieved the best PSNR results with k=5, 6, and 8 while the RSA achieves the best only one time when k=7. This analysis of PSNR metrics shows that the proposed NRBO algorithm outperformed the compared algorithms, demonstrating its robustness and effectiveness in multilevel threshold segmentation.

Table 3 presents the SSIM results for different images, where the proposed algorithm achieved the highest values with all threshold values for images: Image 1, Image 2, Image 5, and Image 6. For Image 3, the proposed algorithm achieved the best results for three thresholds k=6, 7, and 8, while the RSA algorithm achieved the highest SSIM value at k=5. For image 4, the proposed NRBO algorithm achieved the best results for k=5 and 7, while BWO and ALO achieved the highest values at k=6 and 8, respectively. Based on the analysis of the SSIM metric, it is evident that the proposed algorithm demonstrated superior results in multi-threshold image segmentation, achieving best results across the majority of evaluated scenarios. For image 7, the NRBO achieved the best results when k=6, 7, and 8 while the BWO achieved the best only when k=5.

For image 8, the proposed NRBO algorithm achieved the highest best results with all threshold levels except only when k=7, the RSA is the best. For image 9, the proposed NRBO has the highest results when k=5, 6, and 7 and when k=8, the RSA achieved the highest one. For image 10, the proposed NRBO algorithm achieved the best results with all threshold levels.

From the above discussion and the results presented in the tables, it is evident that the proposed algorithm demonstrates superior performance in solving the image segmentation problem for color images of leaf diseases compared to other algorithms. The enhanced performance of the algorithm can be attributed to the incorporation of two key mechanisms: NRSR and TAO. These mechanisms significantly improve the algorithm by enhancing both the exploration and exploitation phases, resulting in higher accuracy and robustness in multilevel threshold segmentation.

Table 1
The mean and standard deviation of FSIM results over all test images

Image	k	RSA		ALO		ASO		BWO		NRBO (Proposed)	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	5	0.6496	0.0625	0.6667	0.0509	0.6235	0.0664	0.6395	0.0333	**0.6818**	0.0513
	6	0.6809	0.0621	0.6763	0.0555	0.6564	0.0763	0.6620	0.0665	**0.6839**	0.0449
	7	0.7111	0.0509	0.6984	0.0761	0.6665	0.0526	**0.7240**	0.0482	0.7170	0.0573
	8	0.7100	0.0583	0.7283	0.0515	0.7038	0.0753	0.7273	0.0709	**0.7354**	0.0538
2	5	0.7203	0.0939	0.7040	0.0894	0.7203	0.0939	0.7135	0.0629	**0.7401**	0.0718
	6	0.7274	0.0993	0.7591	0.0910	0.6889	0.1298	**0.7791**	0.0843	0.7716	0.0888
	7	0.7837	0.0819	0.7974	0.0751	0.7371	0.1078	0.7670	0.0847	**0.8009**	0.0765
	8	0.8108	0.0673	0.7958	0.0613	0.7796	0.0954	0.7812	0.0566	**0.8264**	0.0677
3	5	0.6671	0.0793	0.6667	0.0763	0.6323	0.0977	0.6195	0.0663	**0.6701**	0.0659
	6	0.6721	0.0829	0.6955	0.0886	0.6649	0.0965	0.6199	0.1325	**0.7257**	0.0564
	7	0.7132	0.0927	0.6849	0.0984	0.6891	0.0950	0.6693	0.0995	**0.7533**	0.0745
	8	0.7284	0.0745	0.7171	0.0818	0.7103	0.0765	0.7502	0.1565	**0.7561**	0.0684
4	5	0.6640	0.0425	0.7398	0.1006	0.7469	0.1055	0.7056	0.0956	**0.7703**	0.0747
	6	0.7316	0.0685	0.7573	0.0877	0.7620	0.0871	0.7832	0.0833	**0.7822**	0.0565
	7	0.8068	0.1059	0.8063	0.0449	0.8037	0.0658	0.7378	0.1063	**0.8103**	0.0619
	8	0.7388	0.1031	**0.8292**	0.0708	0.8083	0.0730	0.8125	0.0420	0.8255	0.0544
5	5	0.7312	0.0235	0.7367	0.0841	0.7397	0.0624	0.7467	0.0750	**0.7517**	0.0741
	6	0.7447	0.0835	0.7662	0.0807	0.7484	0.0694	0.7732	0.0498	**0.7798**	0.0571
	7	0.7871	0.0036	0.7850	0.0691	0.7539	0.0679	**0.7966**	0.0950	0.7892	0.0898
	8	0.7421	0.0998	0.7988	0.0866	0.7434	0.0834	0.8090	0.0579	**0.8183**	0.0496
6	5	**0.7509**	0.1076	0.7562	0.0894	0.7366	0.1149	0.7594	0.0867	0.7696	0.0930
	6	0.7791	0.0711	0.7596	0.0864	0.7737	0.0744	0.7773	0.0901	**0.8081**	0.0501
	7	0.7748	0.0794	0.8058	0.0818	0.7758	0.0914	0.8083	0.1065	**0.8220**	0.0732
	8	**0.8482**	0.0365	0.8368	0.0544	0.7774	0.0860	0.8258	0.0947	0.8475	0.0577
7	5	0.6562	0.0656	0.6625	0.0702	0.6552	0.0612	**0.6790**	0.0791	0.6743	0.0700
	6	**0.6965**	0.0634	0.6565	0.0888	0.6613	0.0845	0.6757	0.0713	0.6903	0.0814
	7	0.6918	0.0642	0.6981	0.0804	0.6607	0.0709	0.6956	0.0733	**0.7129**	0.0731
	8	0.7151	0.0275	0.7044	0.0616	0.7069	0.0778	0.7170	0.0767	**0.7292**	0.0692
8	5	0.6462	0.0800	0.6436	0.0684	0.6087	0.0789	0.6467	0.0673	**0.6607**	0.0550
	6	0.6699	0.0675	0.6680	0.0523	0.6268	0.0659	0.6604	0.0766	**0.7110**	0.0592
	7	**0.7083**	0.0674	0.6828	0.0502	0.6203	0.0780	0.6954	0.0682	0.6940	0.0578
	8	**0.7313**	0.0727	0.7172	0.0977	0.6656	0.0847	0.7067	0.0652	0.7434	0.0592
9	5	0.6394	0.0495	0.6360	0.0539	0.6260	0.0526	0.6390	0.0006	**0.6464**	0.0635
	6	0.6506	0.0446	0.6512	0.0475	0.6411	0.0695	0.6625	0.0613	**0.6732**	0.0504
	7	0.6886	0.0566	0.6895	0.0482	0.6524	0.0651	0.6578	0.0453	**0.6967**	0.0465
	8	**0.7246**	0.0542	0.6779	0.0598	0.6625	0.0547	0.6798	0.0716	0.7040	0.0453
10	5	0.6685	0.0509	0.6606	0.0605	0.6417	0.0672	0.6594	0.0615	**0.6752**	0.0459
	6	0.6755	0.0594	0.6869	0.0645	0.6459	0.0863	0.6788	0.0692	**0.6966**	0.0629
	7	0.7228	0.0657	0.7070	0.0568	0.6948	0.0605	0.6951	0.0705	**0.7277**	0.0658
	8	**0.7407**	0.0654	0.7085	0.0546	0.7078	0.0602	0.6951	0.0624	0.7368	0.0671

Table 2
The mean and standard deviation of PSNR results over all test images

Image	K	RSA		ALO		ASO		BWO		NRBO (Proposed)	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	5	15.8390	2.9182	16.7726	2.0457	14.3775	3.3338	15.7698	1.2557	**17.1656**	2.5169
	6	17.0580	2.5441	17.0464	2.5261	15.9255	3.5612	16.5249	1.9953	**17.5311**	2.1431
	7	18.4400	2.2429	17.8422	2.9980	16.3973	2.5155	**18.8863**	2.1691	18.8617	2.4464
	8	18.1168	2.7595	19.2674	2.2769	17.8901	3.2808	19.0957	2.9835	**19.3311**	2.3217
2	5	14.6264	2.4133	14.2775	2.2800	14.8432	2.7797	14.4662	2.5565	**15.1441**	1.8671
	6	14.9054	2.5753	15.9882	2.4831	13.9194	3.3342	**16.2948**	2.4500	16.2073	2.5400
	7	16.6964	2.3105	17.0067	2.1253	15.3678	2.9236	16.2475	2.2472	**17.1528**	2.2867
	8	17.5038	1.9962	16.8797	2.0664	16.7921	2.8840	16.7668	2.1445	**18.0038**	2.2451
3	5	16.4215	3.9705	16.5373	2.7056	14.9146	3.9205	15.3704	1.9605	**16.5377**	2.1007
	6	16.4602	2.9154	17.3690	2.9401	15.8917	3.5471	14.4999	5.1191	**18.4132**	1.9501
	7	17.8739	3.2988	17.5768	2.5663	16.5308	3.6740	16.8990	0.9879	**19.0527**	2.7211
	8	18.5181	2.2353	17.9382	2.7990	17.6451	2.9743	19.0130	4.8959	**19.3244**	2.2119
4	5	**16.2992**	1.7587	15.1442	2.6093	15.3352	2.9860	14.5908	1.1613	16.0876	2.0527
	6	14.9050	1.5569	15.6947	2.5335	15.7949	2.8356	15.9750	2.7322	**16.3481**	1.8910
	7	16.9712	3.8925	17.3003	1.5982	16.9675	2.3563	14.5820	2.7114	**17.4813**	2.1181
	8	15.0243	3.0348	**18.1653**	2.6194	17.2238	2.4710	16.7132	2.1165	18.1496	1.8620
5	5	16.1940	3.0070	16.5352	2.5152	16.5997	1.9814	16.4782	2.7558	**16.9765**	2.3873
	6	17.1355	2.5249	17.5621	2.4454	16.8416	2.2741	17.6103	1.9368	**17.8596**	2.5231
	7	17.9342	0.7738	18.0640	2.1881	17.0463	2.4691	**18.3603**	3.5438	18.2725	2.7810
	8	17.4438	3.9152	19.0500	2.9165	16.9803	2.7441	18.9199	1.3775	**19.5736**	2.3398
6	5	15.0167	2.4665	14.8278	2.4557	14.3814	3.0783	14.7666	2.3318	**15.1978**	2.4599
	6	15.1970	2.5493	15.0327	2.4230	15.2383	2.3475	15.4518	2.4827	**16.2370**	1.8539
	7	15.3116	2.6048	16.4015	2.5138	15.8279	2.7581	16.2555	4.5435	**16.9061**	2.2577
	8	17.5774	1.5731	17.3502	1.9591	15.7334	2.5256	17.2168	2.8940	**17.7416**	2.2487
7	5	14.9472	2.0731	14.8995	2.3946	14.1798	2.3268	**15.3987**	2.7716	15.2050	2.8029
	6	**16.1768**	1.9455	15.0941	2.7767	14.3759	2.7595	15.6701	2.5900	15.7389	2.6943
	7	15.6682	2.2546	15.9818	2.6859	14.7908	2.5632	16.1340	2.3388	**16.7246**	2.5748
	8	15.4659	1.3563	16.6578	2.2376	16.2111	2.9759	16.7032	2.3344	**17.3330**	2.0117
8	5	15.4870	2.5277	16.1436	1.8688	14.7622	2.9863	15.7587	2.2922	**16.2415**	2.0061
	6	16.7668	2.3563	16.4418	1.4884	14.7468	2.4096	16.1320	2.3583	**18.0113**	1.9593
	7	**18.0993**	2.2541	17.2662	2.0261	15.0899	2.8871	17.4554	2.3502	17.5181	1.8102
	8	18.5322	2.2332	18.2096	2.9094	16.5932	2.8058	17.9366	1.9888	**18.8935**	1.5997
9	5	15.2982	2.4125	14.9788	2.6990	14.0287	2.5619	15.3229	1.0013	**15.4835**	2.9347
	6	15.8547	2.1095	15.8716	2.0822	14.2842	3.2004	16.0364	2.7847	**16.6107**	2.3395
	7	17.3868	2.0893	17.0829	1.9909	15.3753	3.0314	16.0698	2.4778	**17.4531**	1.7816
	8	**18.3833**	1.7673	16.6403	2.5108	15.8357	2.3443	16.6642	2.6531	17.6690	1.8630
10	5	16.1444	2.4322	16.0671	2.9133	14.9300	3.0977	16.1369	2.9713	**16.5139**	2.4425
	6	16.9759	2.4372	17.2824	2.5963	15.8561	3.6522	17.2786	2.9112	**17.7636**	2.8482
	7	**19.1532**	2.5036	18.2747	2.3783	17.5200	2.6049	17.6453	3.0373	18.8527	2.7134
	8	19.1363	2.1846	18.2972	2.0859	17.9754	2.5474	17.8162	2.3929	**19.6116**	2.5741

Table 3
The mean and standard deviation of SSIM results over all test images

Image	K	RSA		ALO		ASO		BWO		NRBO (Proposed)	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	5	0.5347	0.1071	0.5739	0.0720	0.4513	0.1515	0.5325	0.0898	**0.5884**	0.0689
	6	0.5880	0.0938	0.5759	0.0912	0.5166	0.1648	0.5833	0.1032	**0.5906**	0.0617
	7	0.6330	0.0691	0.6072	0.1018	0.5392	0.1030	0.6429	0.0594	**0.6438**	0.0720
	8	0.6212	0.0895	0.6540	0.0636	0.5894	0.1398	0.6437	0.0941	**0.6608**	0.0624
2	5	0.5615	0.1303	0.5386	0.1251	0.5589	0.1543	0.5528	0.0852	**0.5861**	0.0960
	6	0.5691	0.1398	0.6146	0.1268	0.5151	0.1905	0.6394	0.1193	**0.6358**	0.1272
	7	0.6506	0.1161	0.6721	0.1030	0.5861	0.1615	0.6302	0.1189	**0.6775**	0.1046
	8	0.6932	0.0933	0.6666	0.0915	0.6515	0.1369	0.6470	0.0807	**0.7166**	0.1016
3	5	**0.5527**	0.1414	0.5395	0.1102	0.4447	0.1778	0.5512	0.1200	0.5364	0.0906
	6	0.5354	0.1183	0.5750	0.1127	0.4956	0.1618	0.4374	0.1835	**0.6090**	0.0769
	7	0.5879	0.1336	0.5681	0.0825	0.5327	0.1637	0.5430	0.0641	**0.6415**	0.1021
	8	0.6147	0.1193	0.5965	0.1031	0.5798	0.1222	0.6611	0.1869	**0.6474**	0.0856
4	5	0.5619	0.0698	0.5960	0.1365	0.5888	0.1535	0.5461	0.0180	**0.6312**	0.0995
	6	0.5924	0.0860	0.6299	0.1160	0.6154	0.1308	**0.6808**	0.1205	0.6613	0.0796
	7	0.6912	0.1819	0.7023	0.0690	0.6812	0.0910	0.6194	0.1731	**0.7104**	0.0912
	8	0.6188	0.1294	**0.7421**	0.0939	0.6951	0.1083	0.7191	0.0618	0.7336	0.0847
5	5	0.5660	0.1026	0.5673	0.1106	0.5631	0.1062	0.5834	0.1003	**0.6018**	0.0989
	6	0.5855	0.1656	0.6166	0.1166	0.5587	0.1081	0.6271	0.0640	**0.6379**	0.0829
	7	0.6597	0.2100	0.6470	0.0891	0.5740	0.1063	0.6423	0.1222	**0.6594**	0.1185
	8	0.6208	0.1494	0.6584	0.1866	0.5528	0.1130	0.6693	0.1008	**0.7031**	0.0730
6	5	0.5644	0.1757	0.5961	0.1228	0.5622	0.1651	0.5929	0.1189	**0.6170**	0.1291
	6	0.6169	0.0955	0.6092	0.1149	0.6052	0.1095	0.6219	0.1176	**0.6663**	0.0807
	7	0.6026	0.1301	0.6644	0.1153	0.6354	0.1274	0.6544	0.1914	**0.6917**	0.1055
	8	0.7253	0.0454	0.7207	0.0740	0.6354	0.1157	0.7024	0.1354	**0.7300**	0.0928
7	5	0.5687	0.1162	0.5875	0.0965	0.5370	0.1155	**0.6105**	0.1200	0.5979	0.0999
	6	0.6168	0.0841	0.5644	0.1205	0.5461	0.1373	0.5968	0.1029	**0.6169**	0.1061
	7	0.6137	0.0899	0.6329	0.1013	0.5655	0.1053	0.6283	0.0883	**0.6394**	0.0959
	8	0.6578	0.0130	0.6411	0.0739	0.6214	0.1220	0.6537	0.0922	**0.6635**	0.0755
8	5	0.4648	0.1492	0.4771	0.1237	0.3891	0.1846	0.4846	0.1147	**0.5106**	0.0967
	6	0.5301	0.1083	0.5228	0.0827	0.4095	0.1484	0.5016	0.1425	**0.5888**	0.0865
	7	**0.5835**	0.1001	0.5531	0.0754	0.3827	0.1738	0.5711	0.1157	0.5647	0.0779
	8	0.6081	0.1047	0.5886	0.1401	0.4844	0.1628	0.5731	0.0969	**0.6326**	0.0723
9	5	0.4621	0.1308	0.4463	0.1288	0.3775	0.1485	0.4209	0.0360	**0.4837**	0.1462
	6	0.5037	0.1010	0.5033	0.0917	0.3954	0.1933	0.5179	0.1229	**0.5269**	0.1063
	7	0.5591	0.0935	0.5703	0.0888	0.4514	0.1744	0.5149	0.1238	**0.5927**	0.0665
	8	**0.6151**	0.0915	0.5359	0.1093	0.4746	0.1219	0.5546	0.1252	0.5918	0.0658
10	5	0.5461	0.0860	0.5321	0.0994	0.4552	0.1572	0.5348	0.1261	**0.5601**	0.0835
	6	0.5701	0.1085	0.5724	0.0980	0.4811	0.1926	0.5690	0.1249	**0.5903**	0.0988
	7	0.6311	0.0948	0.6226	0.0872	0.5526	0.1066	0.5844	0.1104	**0.6454**	0.0828
	8	0.6586	0.0871	0.6115	0.0966	0.5868	0.1173	0.6029	0.0930	**0.6599**	0.0847

6.2 CPU time results

This section presents the average CPU time required by each algorithm to segment all images. As illustrated in Figure 5, the proposed NRBO algorithm demonstrated superior efficiency, completing the segmentation process with an average time less than all other compared algorithms.

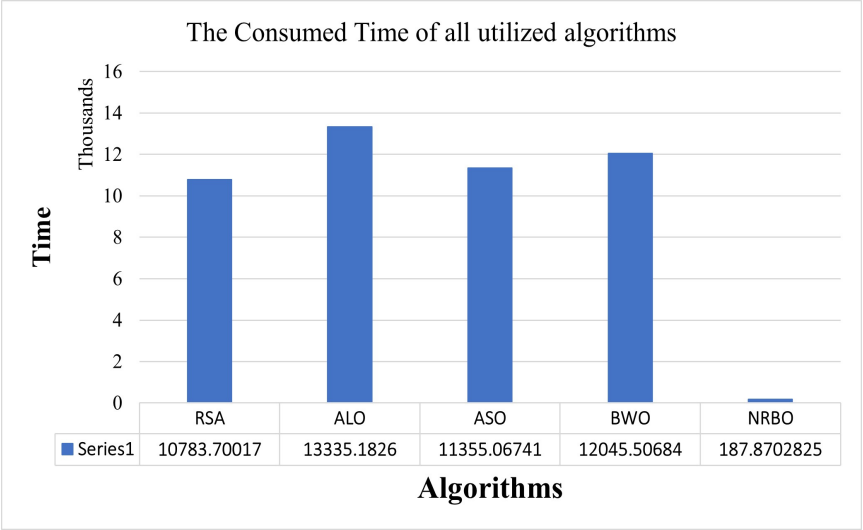


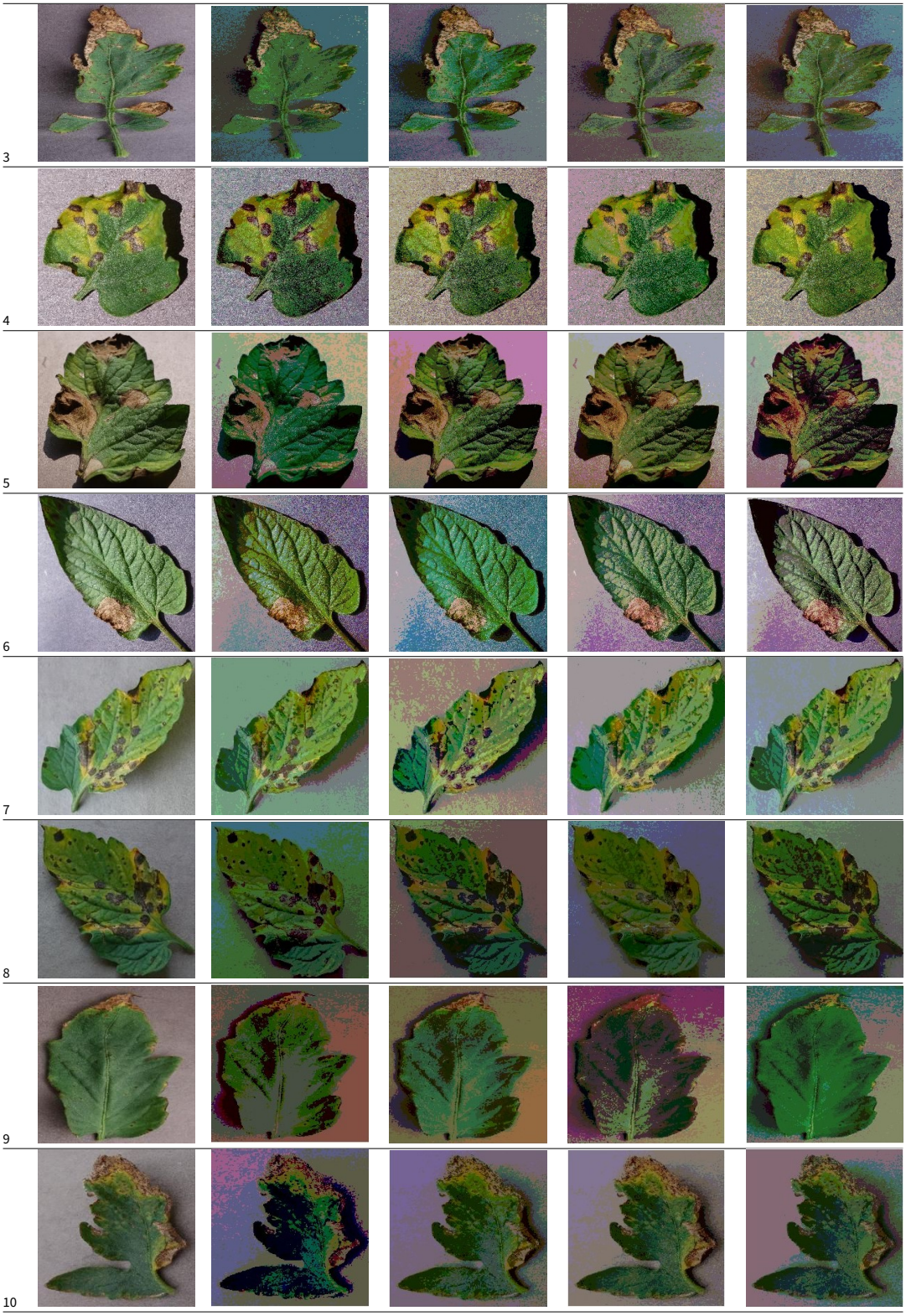
Figure 5. The average consumed time of all compared algorithms

6.3 Segmented images using Kapur

In Table 4, the segmentation results for used images with different threshold values (k) are presented. The nature of the segmented images changes for different threshold values, where increasing the utilized threshold values will present more details about each segmented image.

Table 4
Results after applying NRBO-Kapur to the benchmarked images

Image	Original Image	K=5	K=6	K=7	K=8
1					
2					


3
4
5
6
7
8
9
10

7. Conclusion

Image segmentation is a fundamental step in diagnosing plant diseases, and effective segmentation methods are crucial for ensuring the timely and accurate identification of diseased regions. This paper presented an enhanced version of the Newton Raphson-Based Optimizer (NRBO), a novel metaheuristic algorithm enhanced by two key mechanisms: the Newton Raphson Search Rule (NRSR) and the Trap Avoidance Operator (TAO). These mechanisms significantly enhance the traditional NRBO by addressing its limitations, like slow convergence and local optima, thereby improving its overall performance. The modified NRBO algorithm is deployed to solve widely recognized multilevel threshold image segmentation problem, which poses challenges in determining optimal threshold values, especially for color images. In this study, the modified NRBO utilizes Kapur's entropy as the objective function for segmenting tomato leaf diseases images, enabling the identification of affected areas. Experiments and comparisons against common algorithms, including RSA, ALO, ASO, and BWO, demonstrated the superior performance of the proposed algorithm in terms of metrics such as FSIM, PSNR, SSIM, fitness value, and CPU time. Future work will focus on extending the algorithm's capabilities to other domains, exploring alternative objective functions, and integrating additional optimization mechanisms to further enhance its efficiency and generalizability.

Open data statement

The datasets analyzed during the current study are available in https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset.

References

[1] Joshua Benjamin, Oluwadamilola Idowu, Oreoluwa Khadijat Babalola, Emmanuel Victor Oziegbe, David Olayinka Oyedokun, Aanuoluwapo Mike Akinyemi, and Aminat Adebayo. "Cereal production in Africa: the threat of certain pests and weeds in a changing climate—a review". In: *Agriculture & Food Security* 13.1 (2024), p. 18.

[2] Prof Sanjay B Dhaygude and Mr Nitin P Kumbhar. "Agricultural plant leaf diseases detection image processing". In: *paper by in IJAREEIE* 2.1 (2013).

[3] Sai Arivazhagan, R Newlin Shebiah, S Ananthi, and S Vishnu Varthini. "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features". In: *Agricultural Engineering International: CIGR Journal* 15.1 (2013), pp. 211–217.

[4] Anand H Kulkarni and Ashwin Patil. "Applying image processing technique to detect plant diseases". In: *International Journal of Modern Engineering Research* 2.5 (2012), pp. 3661–3664.

[5] Savita N Ghaiwat and Parul Arora. "Detection and classification of plant leaf diseases using image processing techniques: a review". In: *International Journal of Recent Advances in Engineering & Technology* 2.3 (2014), pp. 1–7.

[6] Sabah Bashir and Navdeep Sharma. "Remote area plant disease detection using image processing". In: *IOSR Journal of Electronics and Communication Engineering* 2.6 (2012), pp. 31–34.

[7] Shanwen Zhang, Haoxiang Wang, Wenzhun Huang, and Zhuhong You. "Plant diseased leaf segmentation and recognition by fusion of superpixel, K-means and PHOG". In: *Optik* 157 (2018), pp. 866–872.

[8] Songwei Zhao, Pengjun Wang, Ali Asghar Heidari, Huiling Chen, Hamza Turabieh, Majdi Mafarja, and Chengye Li. "Multilevel threshold image segmentation with diffusion association slime mould algorithm and Renyi's entropy for chronic obstructive pulmonary disease". In: *Computers in Biology and Medicine* 134 (2021), p. 104427.

[9] Mohamed Abdel-Basset, Victor Chang, and Reda Mohamed. "HSMA_WOA: A hybrid novel Slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images". In: *Applied soft computing* 95 (2020), p. 106642.

[10] Dong-Yan Zhang, Han-Sen Luo, Tao Cheng, Wei-Feng Li, Xin-Gen Zhou, Chun-Yan Gu, Zhihua Diao, et al. "Enhancing wheat Fusarium head blight detection using rotation Yolo wheat detection network and simple spatial attention network". In: *Computers and Electronics in Agriculture* 211 (2023), p. 107968.

[11] Panli Zhang, Jingnan Yang, Fanfan Lou, Jiquan Wang, and Xiaobo Sun. "Aptenodytes Forsteri optimization algorithm based on adaptive perturbation of oscillation and mutation operation for image multi-threshold segmentation". In: *Expert Systems with Applications* 224 (2023), p. 120058.

[12] Luo Chaoxi, He Lifang, Huang Songwei, Huang Bin, Yang Changzhou, and Du Lingpan. "An improved bald eagle algorithm based on Tent map and Levy flight for color satellite image segmentation". In: *Signal, Image and Video Processing* 17.5 (2023), pp. 2005–2013.

[13] Seyed Jalaleddin Mousavirad and Hossein Ebrahimpour-Komleh. "Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms". In: *Evolutionary Intelligence* 10.1 (2017), pp. 45–75.

[14] Leila Esmaeili, Seyed Jalaleddin Mousavirad, and Ali Shahidinejad. "An efficient method to minimize cross-entropy for selecting multi-level threshold values using an improved human mental search algorithm". In: *Expert Systems with Applications* 182 (2021), p. 115106.

[15] R Srikanth and K Bikshalu. "Multilevel thresholding image segmentation based on energy curve with harmony Search Algorithm". In: *Ain Shams Engineering Journal* 12.1 (2021), pp. 1–20.

[16] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer". In: *Advances in engineering software* 69 (2014), pp. 46–61.

[17] Hongliang Guo, Mingyang Li, Hanbo Liu, Xiao Chen, Zhiqiang Cheng, Xiaohua Li, Helong Yu, and Qiuxiang He. "Multi-threshold Image Segmentation based on an improved Salp Swarm Algorithm: Case study of breast cancer pathology images". In: *Computers in Biology and Medicine* 168 (2024), p. 107769.

[18] Hossam Faris, Seyedali Mirjalili, Ibrahim Aljarah, Majdi Mafarja, and Ali Asghar Heidari. "Salp swarm algorithm: theory, literature review, and application in extreme learning machines". In: *Nature-inspired optimizers: theories, literature reviews and applications* (2020), pp. 185–199.

[19] James Kennedy and Russell Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95-international conference on neural networks*. Vol. 4. ieee. 1995, pp. 1942–1948.

[20] Seyedali Mirjalili. "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm". In: *Knowledge-based systems* 89 (2015), pp. 228–249.

[21] Shimin Li, Huiling Chen, Mingjing Wang, Ali Asghar Heidari, and Seyedali Mirjalili. "Slime mould algorithm: A new method for stochastic optimization". In: *Future generation computer systems* 111 (2020), pp. 300–323.

[22] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen. "Harris hawks optimization: Algorithm and applications". In: *Future generation computer systems* 97 (2019), pp. 849–872.

[23] Xin-She Yang. "Firefly algorithms for multimodal optimization". In: *International symposium on stochastic algorithms*. Springer. 2009, pp. 169–178.

[24] Yutao Yang, Huiling Chen, Ali Asghar Heidari, and Amir H Gandomi. "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts". In: *Expert Systems with Applications* 177 (2021), p. 114864.

[25] Jiaze Tu, Huiling Chen, Mingjing Wang, and Amir H Gandomi. "The colony predation algorithm". In: *Journal of Bionic Engineering* 18 (2021), pp. 674–710.

[26] Xin-She Yang and Suash Deb. "Cuckoo search via Lévy flights". In: *2009 World congress on nature & biologically inspired computing (NaBIC)*. Ieee. 2009, pp. 210–214.
[27] Seyedali Mirjalili and Andrew Lewis. "The whale optimization algorithm". In: *Advances in engineering software* 95 (2016), pp. 51–67.
[28] Hang Su, Dong Zhao, Ali Asghar Heidari, Lei Liu, Xiaoqin Zhang, Majdi Mafarja, and Huiling Chen. "RIME: A physics-based optimization". In: *Neurocomputing* 532 (2023), pp. 183–214.
[29] Seyedali Mirjalili. "SCA: a sine cosine algorithm for solving optimization problems". In: *Knowledge-based systems* 96 (2016), pp. 120–133.
[30] Iman Ahmadianfar, Ali Asghar Heidari, Saeed Noshadian, Huiling Chen, and Amir H Gandomi. "INFO: An efficient optimization algorithm based on weighted mean of vectors". In: *Expert Systems with Applications* 195 (2022), p. 116516.
[31] Iman Ahmadianfar, Ali Asghar Heidari, Amir H Gandomi, Xuefeng Chu, and Huiling Chen. "RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method". In: *Expert Systems with Applications* 181 (2021), p. 115079.
[32] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Abdolreza Hatamlou. "Multi-verse optimizer: a nature-inspired algorithm for global optimization". In: *Neural Computing and Applications* 27 (2016), pp. 495–513.
[33] Peter Fortini and Richard Barakat. "An algorithm for gene frequency changes for linked autosomal loci based on genetic algebras". In: *Journal of Mathematical Analysis and Applications* 83.1 (1981), pp. 135–143.
[34] Rainer Storn and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces". In: *Journal of global optimization* 11 (1997), pp. 341–359.
[35] Xiaopeng Ma, Kai Zhang, Liming Zhang, Chuanjin Yao, Jun Yao, Haochen Wang, Wang Jian, and Yongfei Yan. "Data-driven niching differential evolution with adaptive parameters control for history matching and uncertainty quantification". In: *Spe Journal* 26.02 (2021), pp. 993–1010.
[36] Shweta Bondre and Dipti Patil. "Crop disease identification segmentation algorithm based on Mask-RCNN". In: *Agronomy Journal* 116.3 (2024), pp. 1088–1098.
[37] Shalini Sharma, E Rajesh, Baskar Kasi, and Sakthi Govindaraju. "The Classification Model for Plant Disease Detection with Segmentation and GLCM". In: *Proceedings of the 1st International Conference on Artificial Intelligence, Communication, IoT, Data Engineering and Security, IACIDS 2023, 23-25 November 2023, Lavasa, Pune, India*. 2024.
[38] Pablo Mesejo, Oscar Ibáñez, Oscar Cordón, and Stefano Cagnoni. "A survey on image segmentation using metaheuristic-based deformable models: state of the art and critical analysis". In: *Applied Soft Computing* 44 (2016), pp. 1–29.
[39] Rustu Akay, Radhwan AA Saleh, Shawqi MO Farea, and Muzaffer Kanaan. "Multilevel thresholding segmentation of color plant disease images using metaheuristic optimization algorithms". In: *Neural Computing and Applications* 34.2 (2022), pp. 1161–1179.
[40] Vijai Singh and Ak K Misra. "Detection of plant leaf diseases using image segmentation and soft computing techniques". In: *Information processing in Agriculture* 4.1 (2017), pp. 41–49.
[41] Hairu Guo, Jin'ge Wang, and Yongli Liu. "Multi-threshold image segmentation algorithm based on Aquila optimization". In: *The Visual Computer* 40.4 (2024), pp. 2905–2932.
[42] Rupak Chakraborty, Rama Sushil, and Madan Lal Garg. "An improved PSO-based multilevel image segmentation technique using minimum cross-entropy thresholding". In: *Arabian Journal for Science and Engineering* 44 (2019), pp. 3005–3020.
[43] Zhikai Xing. "An improved emperor penguin optimization based multilevel thresholding for color image segmentation". In: *Knowledge-Based Systems* 194 (2020), p. 105570.

[44] Swarnajit Ray, Arunita Das, Krishna Gopal Dhal, Jorge Gálvez, and Prabir Kumar Naskar. "Cauchy with whale optimizer based eagle strategy for multi-level color hematology image segmentation". In: *Neural Computing and Applications* 33.11 (2021), pp. 5917–5949.

[45] Essam H Houssein, Bahaa El-din Helmy, Diego Oliva, Ahmed A Elngar, and Hassan Shaban. "A novel black widow optimization algorithm for multilevel thresholding image segmentation". In: *Expert Systems with Applications* 167 (2021), p. 114159.

[46] Ping Wang, Yan Zhang, Boran Jiang, and Jinyi Hou. "An maize leaf segmentation algorithm based on image repairing technology". In: *Computers and electronics in agriculture* 172 (2020), p. 105349.

[47] J Anitha, S Immanuel Alex Pandian, and S Akila Agnes. "An efficient multilevel color image thresholding based on modified whale optimization algorithm". In: *Expert Systems with Applications* 178 (2021), p. 115003.

[48] Fatma A Hashim and Abdelazim G Hussien. "Snake Optimizer: A novel meta-heuristic optimization algorithm". In: *Knowledge-Based Systems* 242 (2022), p. 108320.

[49] Gang Hu, Rui Yang, Muhammad Abbas, and Guo Wei. "BEESO: Multi-strategy boosted snake-inspired optimizer for engineering applications". In: *Journal of Bionic Engineering* 20.4 (2023), pp. 1791–1827.

[50] Liguo Yao, Panliang Yuan, Chieh-Yuan Tsai, Taihua Zhang, Yao Lu, and Shilin Ding. "ESO: An enhanced snake optimizer for real-world engineering problems". In: *Expert Systems with Applications* 230 (2023), p. 120594.

[51] Jiquan Wang, Jinling Bei, Haohao Song, Hongyu Zhang, and Panli Zhang. "A whale optimization algorithm with combined mutation and removing similarity for global optimization and multilevel thresholding image segmentation". In: *Applied Soft Computing* 137 (2023), p. 110130.

[52] Xiang Liu, Min Tian, Jie Zhou, and Jinyan Liang. "An efficient coverage method for SEMWSNs based on adaptive chaotic Gaussian variant snake optimization algorithm." In: *Mathematical Biosciences and Engineering: MBE* 20.2 (2022), pp. 3191–3215.

[53] Ruba Abu Khurma, Dheeb Albashish, Malik Braik, Abdullah Alzaqebah, Ashwaq Qasem, and Omar Adwan. "An augmented Snake Optimizer for diseases and COVID-19 diagnosis". In: *Biomedical Signal Processing and Control* 84 (2023), p. 104718.

[54] Ibrahim Al-Shourbaji, Pramod H Kachare, Samah Alshathri, Salahaldeen Duraibi, Bushra Elnaim, and Mohamed Abd Elaziz. "An efficient parallel reptile search algorithm and snake optimizer approach for feature selection". In: *Mathematics* 10.13 (2022), p. 2351.

[55] Ravichandran Sowmya, Manoharan Premkumar, and Pradeep Jangir. "Newton-Raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems". In: *Engineering Applications of Artificial Intelligence* 128 (2024), p. 107532.

[56] Xiaohan Zhao, Liangkuan Zhu, and Bowen Wu. "An improved mayfly algorithm based on Kapur entropy for multilevel thresholding color image segmentation". In: *Journal of Intelligent & Fuzzy Systems* 44.1 (2023), pp. 365–380.

[57] Bahriye Akay. "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding". In: *Applied Soft Computing* 13.6 (2013), pp. 3066–3091.

[58] Jagat Narain Kapur, Prasanna K Sahoo, and Andrew KC Wong. "A new method for gray-level picture thresholding using the entropy of the histogram". In: *Computer vision, graphics, and image processing* 29.3 (1985), pp. 273–285.

[59] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming: theory and algorithms*. John wiley & sons, 2006.

[60] Bahman Kalantari. "Generalization of Taylor's theorem and Newton's method via a new family of determinantal interpolation formulas and its applications". In: *Journal of Computational and Applied Mathematics* 126.1-2 (2000), pp. 287–318.

[61] Tian-Xiao He. "Generalized Stirling numbers and generalized Stirling functions". In: *arXiv preprint arXiv:1106.5251* (2011).

[62] Sunethra Weerakoon and TGI1791767 Fernando. "A variant of Newton's method with accelerated third-order convergence". In: *Applied mathematics letters* 13.8 (2000), pp. 87–93.
[63] Á Alberto Magreñán and Ioannis K Argyros. "Newton's method". In: (2018).
[64] Iman Ahmadianfar, Omid Bozorg-Haddad, and Xuefeng Chu. "Gradient-based optimizer: A new metaheuristic optimization algorithm". In: *Information Sciences* 540 (2020), pp. 131–159.