Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(6): 9645–9669.
DOI: 10.3934/mbe.2023423
Received: 29 December 2022
Revised: 28 February 2023
Accepted: 06 March 2023
Published: 22 March 2023

Research article

Multi-level thresholding image segmentation for rubber tree secant using improved Otsu's method and snake optimizer

Shenghan Li and Linlin Ye*

Hainan University, Haikou 570228, China

* **Correspondence:** Email:994097@hainanu.edu.cn.

Abstract: The main disease that decreases the manufacturing of natural rubber is tapping panel dryness (TPD). To solve this problem faced by a large number of rubber trees, it is recommended to observe TPD images and make early diagnosis. Multi-level thresholding image segmentation can extract regions of interest from TPD images for improving the diagnosis process and increasing the efficiency. In this study, we investigate TPD image properties and enhance Otsu's approach. For a multi-level thresholding problem, we combine the snake optimizer with the improved Otsu's method and propose SO-Otsu. SO-Otsu is compared with five other methods: fruit fly optimization algorithm, sparrow search algorithm, grey wolf optimizer, whale optimization algorithm, Harris hawks optimization and the original Otsu's method. The performance of the SO-Otsu is measured using detail review and indicator reviews. According to experimental findings, SO-Otsu performs better than the competition in terms of running duration, detail effect and degree of fidelity. SO-Otsu is an efficient image segmentation method for TPD images.

Keywords: Otsu's method; tapping panel dryness; snake optimizer; image segmentation

1. Introduction

Due to the in-situ coagulation of rubber particles, tapping panel dryness (TPD) is a phenomenon that causes a partial or finally complete halt of latex flow upon tapping [1]. The study of TPD is vitally important, encompassing developmental regularity, physiological mechanisms and prevention [2]. But observation of TPD images and accurate diagnosis are the fundamental steps in this process. Around the world, rubber trees are planted over 15 million hectares. Faced with the large amounts of image data and different symptoms, manual observations is time-consuming, inaccurate and prone to misjudgments. It is a good choice to apply image processing technology in observation process. The observation of TPD images will be effective and accurate, and not influenced by subjective factors.

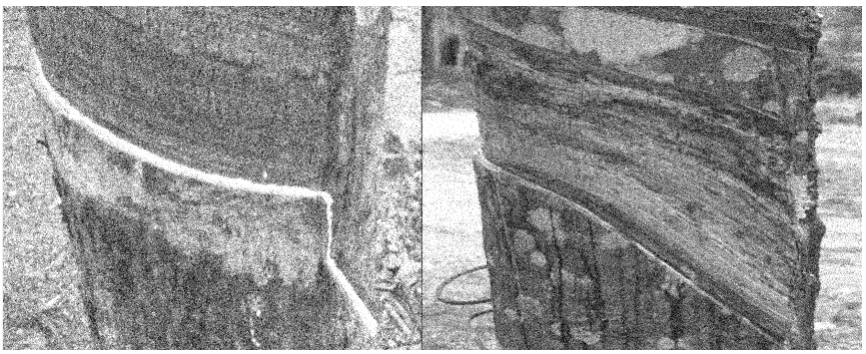

Figure 1. The normal rubber tree (left) and the rubber tree with TPD (right).

Some scholars have made initial attempts in this direction. Sun conducted research using machine vision to recognize and plan tapping trajectories of natural rubber trees [3]. They proposed a combinatorial optimization algorithm that could recognize and plan the next tapping trajectory under natural light conditions, with a success rate of 89.4%. This research could provide valuable technical reference for the development of intelligent tapping robots. To separate the secant and latex and eliminate interference factors, image processing technology was utilized to obtain an accurate secant and latex binary image [4]. In reference [5], an improved 2-D entropy algorithm was applied for rubber tree image segmentation. By using the principle of two-dimensional maximum entropy and synthetically selecting the global search of genetic algorithm and simulated annealing local hill climbing performance, the optimal segmentation threshold was determined. The experimental results showed that this method efficiently extracted the rubber tree cutting marks and separated the rubber latex.

As one of the basic steps of image processing, the purpose of image segmentation is separating the entire image into several specific regions and present the objects under concern [6]. Traditional image segmentation methods include threshold-based segmentation [7], edge-based segmentation [8], region-based segmentation [9] and clustering segmentation [10]. Due to its ease of implementation, minimal storage space needs, high precision and speed, thresholding segmentation is widely used. Thresholding segmentation extract the essential information using the histogram of image under concern. The entire image is divided into light and dark regions, where all pixels in each region has the same intensities [11]. Thanks to the excellent performance, threshold-based image segmentation has been widely used in medical image analysis [12], defect detection [13] and remote sensing image analysis [14]. In our opinion, thresholding segmentation can also be used to obtain the effective results in the observation of TPD and early diagnosis.

Thresholding techniques are frequently used to segment images, with the specific technique chosen based on the image's histogram and a set of threshold values. The goal of threshold-based segmentation is to determine the optimal thresholds for an image using its histogram. A histogram is a graphical representation of the frequency of pixel intensity values within a particular gray-level range of an image [15]. The process of determining the optimal thresholds may involve either bi-level thresholding (BT) [16] or multi-level thresholding (MT) [17].

Furthermore, the primary challenge lies in determining the optimal thresholds that effectively divide an image into multiple segments. In order to address this issue, Otsu proposed a method that selects the optimal thresholds by maximizing the in-between class variance [18]. Kapur's entropy, another classic thresholding approach described in [19], seeks to achieve image segmentation by maximizing

the image entropy. Tsallis entropy, which relies on a new version of Shannon's entropy theory [20], was also proposed for this purpose. In [21], a new thresholding method was proposed that selects the optimal threshold values by minimizing the cross entropy between the original and segmented images. However, this approach can be time-consuming when performing MT segmentation tasks [22].

Otsu's method is a preferred thresholding segmentation approach. There are BT and MT versions. For simple images, both BT and MT Otsu's methods are feasible. However, in the face of images with complex color and region, BT Otsu's method will not complete segmentation, and MT Otsu's method will obtain unsatisfactory results. The computation takes longer as the threshold value rises. When it comes to a large amount of image data, the increase in computation time will be more noticeable. The effectiveness of Otsu's method, therefore, needs to be optimized. Additionally, it is essential to increase segmentation accuracy. The time-honoured Otsu's method is still interesting today because of the many optimized versions proposed.

With the purpose to speed up the image segmentation, scholars have proposed the improved Otsu's methods combining many meta-heuristic algorithms. Meta-heuristic algorithms have gained widespread use in engineering applications, including transportation, communication, image processing and more. This is due to their exceptional performance and their ability to solve optimization problems with ease. Additionally, the No Free Lunch Theorem (NFL) [23] suggests that there is no optimization algorithm that can perform the best across all types of problems. This has kept the research in this area active and encouraged scientists to continually develop new and improved algorithms for better optimization.

Because of higher computing overhead of conventional multilevel image thresholding approaches, the application of meta-heuristic algorithms becomes attractive in order to reduce computing costs.The most used meta-heuristic algorithms include Fruit Fly Optimization Algorithm (FOA) [24], Harris hawks optimization (HHO) [25], Slime mould algorithm (SMA) [26], Monarch butterfly optimization (MBO) [27], Colony Predation Algorithm (CPA) [28], Moth search algorithm (MSA) [29], Hunger games search (HGS) [30], efficient optimization algorithm based on weighted mean of vectors (INFO) [31], Runge Kutta method (RUN) [32], enhanced binary Rat Swarm Optimizer based on local-best concepts of PSO and collaborative crossover operators (BERSOC) [33], hybrid method based on Butterfly Optimization Algorithm and Ant Lion Optimizer (BOAALO) [34], modified whale optimization algorithm with population reduction (mWOAPR) [35], bald eagle search optimization (BES) [36], Binary Horse herd optimization algorithm with crossover operators (BHOA) [37], sparrow search algorithm (SSA) [38], Grey Wolf Optimizer (GWO) [39] and the Whale Optimization Algorithm (WOA) [40].

Many scholars have proposed improved versions of Otsu's method combined with meta-heuristic algorithms. Huang proposed FOA-Otsu by combining FOA [41], which converges faster without sacrificing the segmentation accuracy. Based on a more effective algorithm for artificial bee colonies, Bhandari enhanced Otsu's technique. It performs well when segmenting satellite images [42]. Essam H. Houssein proposed an efficient image segmentation method for skin cancer imaging using improved golden jackal optimization algorithm [43]. Using a Bee Foraging Algorithm, the optimum multilayer threshold of picture segmentation has been presented [44]. The hybrid whale optimization algorithm (HWOA) provided the best threshold value for color picture segmentation [45].

Many scholars also focus on the improvement of strategies of Otsu's method. Using the weighted object variance approach, which has a high detection rate and a low false alarm rate, Yuan enhanced the Otsu's method for defect identification [46]. The use of histogram cumulative moments in the

Improved Otsu method for ore segmentation was proposed in [47]. Mineral pictures with single-peaked or negligible bimodal feature histograms may be efficiently and correctly segmented using this technique. By constraining the gray level search range, Xu improved Otsu's method [48], which has superior performance compared with the original one.

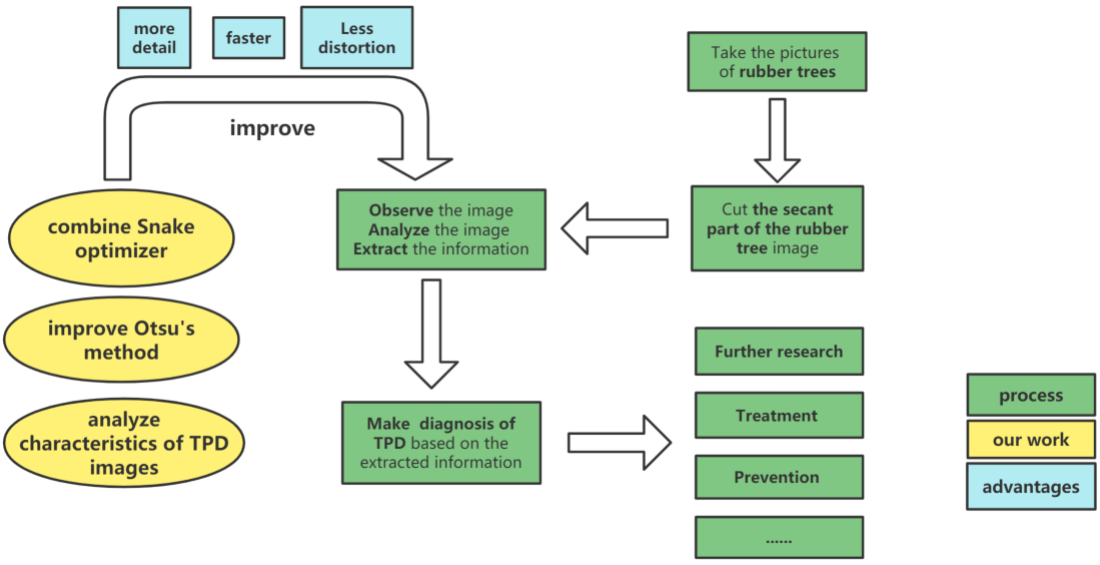

Figure 2. The advantages of SO-Otsu and its value in TPD research process.

The rest of this paper is organized as follows: Section 2 concentrates on Otsu's method and snake optimizer. Section 3 presents an analysis of TPD image features and an improved version of Otsu's method. Based on this, we use Otsu's improved approach in conjunction with the snake optimizer to tackle the multilevel thresholding problem. We refer to the proposed method as SO-Otsu. The experimental data set, evaluation indicators, and parameter tuning of the testing algorithms are presented in Section 4. Section 5 was devoted to a detailed review and reviews of indicators. In this section, the functionality of SO-Otsu is compared with original Otsu's method, FOA-Otsu, SSA-Otsu, GWO-Otsu, WOA-Otsu and HHO-Otsu on three TPD images. In detailed review, we define new evaluation metrics to measure the details of algorithm segmentation. For indicator reviews, We plot the convergence curve of the algorithm to visualize the optimization process. Furthermore, we also performed a statistical test of the experimental results. The outcomes demonstrate that the proposed approach performs comparably better in terms of image detail and processing time than other examined algorithms. Finally, the conclusion and future work are shown in Section 6.

2. Algorithm principle and method

2.1. Otsu's method

Otsu's method, a global adaptive thresholding image segmentation algorithm, was introduced by Nobuyuki Otsu, a Japanese scholar, in 1979 [49]. It is used to select the optimal image thresholds by maximizing the between-class variance, which enables the separation of segmented classes in the

most effective way possible. This technique assumes comparable pixel numbers for each class, which facilitates successful segmentation.

2.1.1. Bi-level thresholding

In Otsu's thresholding, the grayscale image is characterized in L gray levels $(1, 2, \ldots, L)$ having pixels with gray-levels from the integer set $\{0, 1, 2, \ldots, 255\}$ [50]. In the single-level approach, the image is separated by a threshold at a level t into two classes, namely C_0 and C_1. For a bi-level thresholding problem, Otsu's between-class variance can be expressed as:

$$\mu_{\mathrm{T}} = \omega_0\mu_0 + \omega_1\mu_1 \text{ and } \omega_0 + \omega_1 = 1 \tag{2.1}$$

where σ_0 and σ_1 are the variances of C_0 and C_1.

2.1.2. Muti-level thresholding

For multi-level thresholding, assume that m threshold levels $(t_1, t_2, \ldots, t_m)$ split the image into $m + 1$ classes: $C_0, C_1, C_2, ..., C_m$ .the objective function for the segmentation process can be expressed by:

$$\begin{aligned} J(t)_{\max} &= \sigma_0 + \sigma_1 + \cdots + \sigma_m \\ \sigma_0 &= \omega_0 \left(\mu_0 - \mu_t\right)^2 \\ \sigma_1 &= \omega_1 \left(\mu_1 - \mu_t\right)^2 \\ &\cdots \\ \sigma_m &= \omega_m \left(\mu_m - \mu_t\right)^2 \end{aligned} \tag{2.2}$$

2.1.3. Analysis of algorithm

From the above formula, it can be seen that the solution space of the whole algorithm is a $q - 1$ dimensional space. Any $q - 1$ dimensional vector in the solution space needs to be calculated based on the inter-class variance. Parameters, such as the inter-class mean, need to be calculated. Combining these calculations, the time complexity of multi-threshold partitioning according to Otsu's criterion is $O(L^q)$. The time complexity rises exponentially with the quantity of thresholds because of the exhaustive calculation in the $q-1$ dimensional solution space. The multi-level thresholding approach chooses many thresholds as opposed to the single-level thresholding method. The processed image has better detail because there are more thresholds.

2.2. Snake Optimizer

Proposed by Fatma A. Hashim, Snake Optimizer (SO) was compared to other meta-heuristic algorithms, such as MFO, HHO, TEO, GOA, WOA, L-SHADE, and L-SHADE-EpSin using 30 unconstrained benchmark functions from CEC 2017 and 10 functions from CEC 2020, as reported in Reference [51]. The experimental results demonstrate that SO outperforms the other algorithms. Notably, in CEC 2017, SO achieved the best average results in 22 out of 29 functions and the second-best in 3 functions, while ranking third in the remaining 2 functions. In CEC 2020, SO ranked first in 6 functions out of 10, and achieved the second-best results in 3 functions. Given its excellent performance, we decided to use SO to optimize Otsu's method.

It is a brand-new meta-heuristic optimization technique that was motivated by a snake behavior: mating activity only happens when it is cold outside and there is food available. Otherwise, the snakes just search for food or eat the available food. The search method is split into two parts based on this information: exploration and exploitation.

The exploration phase occurs when food is not available, and each individual will explore for food. When the food is available, the snakes will be in exploitation phase. When the temperature is high, the snakes will focus on eating the available food. If the area is cold, the mating process occurs.

During the mating process, the fight mode or mating mode may occur. In the fight mode, each male will fight to get the best female and each female will try to select the best male. In the mating mode, the mating occurs between each pair related to the availability of food quantity. If the mating process occurs in a search space, there is a probability that the female lays eggs and hatch them into new snakes.

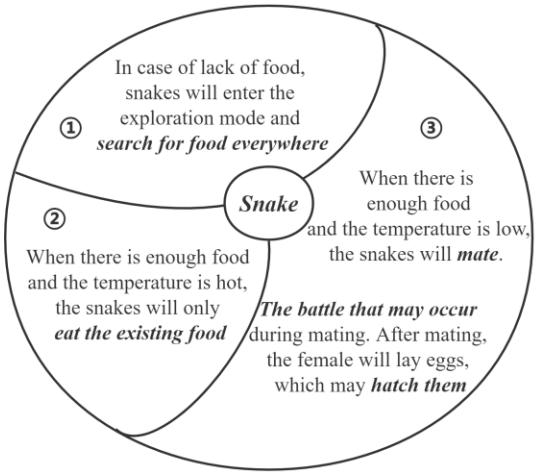

Figure 3. Behavioral patterns of snakes.

The main steps of SO can be described as follows:

Step 1: Initialization

The initial population can be obtained using the following equation:

$$X_i = X_{\min} + rand \times Limit \tag{2.3}$$

where X_i is the position of the i_{th} individual, *rand* is a random number between 0 and 1, and *Limit* is the bounds of the problem.

Then, the population is divided to 2 groups: male group and female one. The best individual in each group is found, including the best male $f_{best,m}$ and the best female $f_{best,f}$ and also the food position f_{food}.

The temperature *temp* and food quantity Q are defined by using following equations:

$$Temp = \exp\left(\frac{-t}{T}\right) \quad and \quad \mathrm{Q} = c_1 \times \exp\left(\frac{t-T}{T}\right) \tag{2.4}$$

where t refers to the current iteration and T refers to the maximum number of iterations.

Step 2: Exploration Phase

When food is not available, the snakes search for food by selecting any random position and update their position.

$$X_i = X_{\text{rand}} \pm c_2 \times A \times (Limit \times rand + X_{\min}) \tag{2.5}$$

where A is the individual ability to search food, c_2 is 0.05, X_{rand} refers to the position of random individual and $X_{\min}$ is the lower bounds of the problem.

Step 3: Exploitation Phase

When the temperature is high, the snakes are in *the ordering mode*. Each individual will move to the current best food location X_{food}.

$$X_i(t+1) = X_{\text{food}} \pm c_3 \times Temp \times rand \times (X_{\text{food}} - X_i(t)) \tag{2.6}$$

where X_i is the position of the individual and c_3 is 2.

When the temperature is low, the snakes will be in fight mode or mating mode.

In fight mode:

$$X_i(t+1) = X_i(t) + c_3 \times F \times rand \times (Q \times X_{\text{best}} - X_i(t)) \tag{2.7}$$

In mating mode:

$$X_i(t+1) = X_i(t) + c_3 \times M \times rand \times (Q \times X_i(t) - X_i(t)) \tag{2.8}$$

where F is the fighting ability of an individual, and M is its mating ability.

After mating process, the egg may hatch to replace the worst individual.

$$X_{\text{worst}} = X_{\min} + \text{rand} \times Limit \tag{2.9}$$

where X_{worst} is the position of the worst individual.

3. SO-Otsu

3.1. Improvement of Otsu's method

From Subsection 2.1, it is clear that the Otsu technique employs a global adaptive threshold segmentation strategy. The gray value distribution of the entire image's pixels has an impact on the threshold that is chosen. Otsu's method has the strategy of maximizing inter-class variance, which determines that the threshold is selected in the range of a large number of pixels.

Figure 4 is the rubber tree secant image, and Figure 5 is its gray level histogram. we can see that most pixels of the image are in the low gray-level range. While, the rubber tree secant in the image (the objects of interest) occupies a small area, there are a small number of pixels in the high gray-level range. The number of pixels at low gray-level is much larger than the number of pixels at high gray-level. In the calculation process of original Otsu's method, the proportion of pixels in high gray-level range is small. Therefore, the selected threshold is 47, which is in low gray-level range (as shown by the dashed red line).

Algorithm 1: Snake Optimizer Algorithm

Initialize Problem Setting: $Dim, UB, LB, Pop\_Size(N), Max\_Iter(T), Curr\_Iter(t)$
Initialize the population randomly
Divide population N to 2 equal groups N_m and N_f
while $t \leq T$ **do**
    Evaluate each group N_m and N_f
    Find best male $f_{best,m}$
    Find best female $f_{best,f}$
    Define $Temp$ and food quantity Q
    if $(Q \leq 0.25)$ **then**
        Perform exploration
    else
        if $(Q \geq 0.6))$ **then**
            Perform exploitation
        else
            if $(rand \geq 0.6))$ **then**
                Snake in Fight Mode
            else
                Snake in Mating Mode
                Change the worst male and female
            end
        end
    end
end
Return best solution

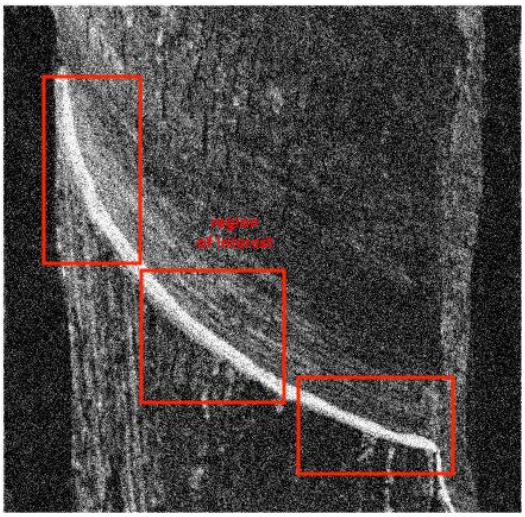

Figure 4. Rubber tree secant.

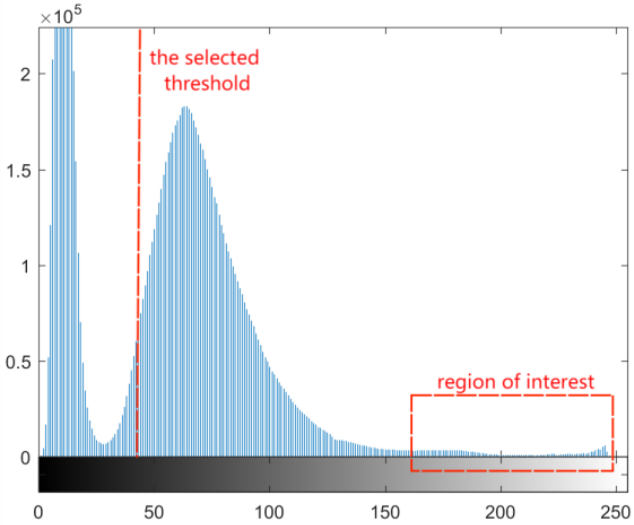

Figure 5. Gray level histogram.

Figure 6. The processed image according to the selected threshold.

The processed image using the selected threshold is shown in Figure 6. The segmentation effect of rubber tree secant is very poor, because the region of interest is not be segmented. If we want to obtain better effect of segmentation, the threshold should be selected in a high gray-level range. So, the paper proposes an improved version of the Otsu's method.

The image under concern is characterized in L gray-levels $(1, 2, \ldots, L)$.

First, set the proper gray-level Th is set as the first threshold. The gray-level of the image is separated into two parts: $[0, Th]$ and $[Th + 1, L - 1]$, according to the selected threshold.

For an image with a $N \times M$ pixel, the number of pixels at with gray-level i is n_i, and the total number of pixels n is:

$$n = M \cdot N = \sum_{i=0}^{L-1} n_i \tag{3.1}$$

The number of pixels in $[0, Th]$ is n_l, and the number of pixels in $[Th + 1, L - 1]$ is n_r.

$$n_l = \sum_{i=0}^{Th} n_i \quad and \quad n_r = \sum_{i=Th+1}^{L-1} n_i \tag{3.2}$$

The probability that a pixel is in $[0, Th]$ is p_{il},and the probability that a pixel is in $[Th + 1, L - 1]$ is p_{ir} .

$$p_{il} = \frac{n_i}{n_l} \quad and \qquad p_{ir} = \frac{n_i}{n_r} \tag{3.3}$$

Setting a gray-level k,the range of $[0, Th]$ is separated into two classes: C_0 and C_1. The range of $[Th + 1, L - 1]$ is the same.

Taking the first part as an example, the gray-level distribution probabilities ω_0 and ω_1 for C_0 and C_1 are expressed as:

$$\omega_0 = \sum_{i=0}^{k} p_{il} \quad and \quad \omega_1 = \sum_{i=k+1}^{Th} p_{il} \tag{3.4}$$

and

$$\omega_0 + \omega_1 = \sum_{i=0}^{Th} p_{il} = \frac{n_l}{n} \tag{3.5}$$

The mean pixel gray-level probabilities for C_0 and C_1 are μ_0 and μ_1

$$\mu_0 = \sum_{i=0}^{k} i \cdot p_{il} \quad and \quad \mu_1 = \sum_{j=k+1}^{Th} i \cdot p_{il} \tag{3.6}$$

The mean intensity μ of range of $[0, Th]$ can be characterized as: $\mu = \sum_{i=0}^{Th} i \cdot p_{il}$

The between-class variance for classes C_0 and C_1 is shown as:

$$\sigma_B^2 = \omega_0 (\mu_0 - \mu)^2 + \omega_1 (\mu_1 - \mu)^2 \tag{3.7}$$

Citing the k , the optimal threshold is k^* , which maximizes $\sigma_B^2(k^*)$:

$$\sigma_B^2(k^*) = \max_{0 \leqslant k \leqslant L-1} \sigma_B^2(k) \tag{3.8}$$

The obtained k^* is the selected threshold in range $[0, Th]$.

In the range of $[Th + 1, L - 1]$, the above process is repeated using n_r and p_{ir}. We will obtain three thresholds.

The improved version treat the range of $[0, Th]$ and $[Th+1, L-1]$ as two separate images. Therefore, during the process of selecting threshold, it can ignore the effect of the numerous pixels in low gray-level range on the region of interest.

What's more, the time complexity of the improved version is $O\left(L^2\right)$, without increasing exponentially with the number of thresholds.

3.2. Optimization of Otsu's method using snake optimizer

To reduce the computation time, we combine the snake optimizer with the improved Otsu's method and propose SO-Otsu. In SO-Otsu, the gray level is regarded as the position of snake and the improved Otsu's method used as objective function. The specific steps are shown in Figure 7.

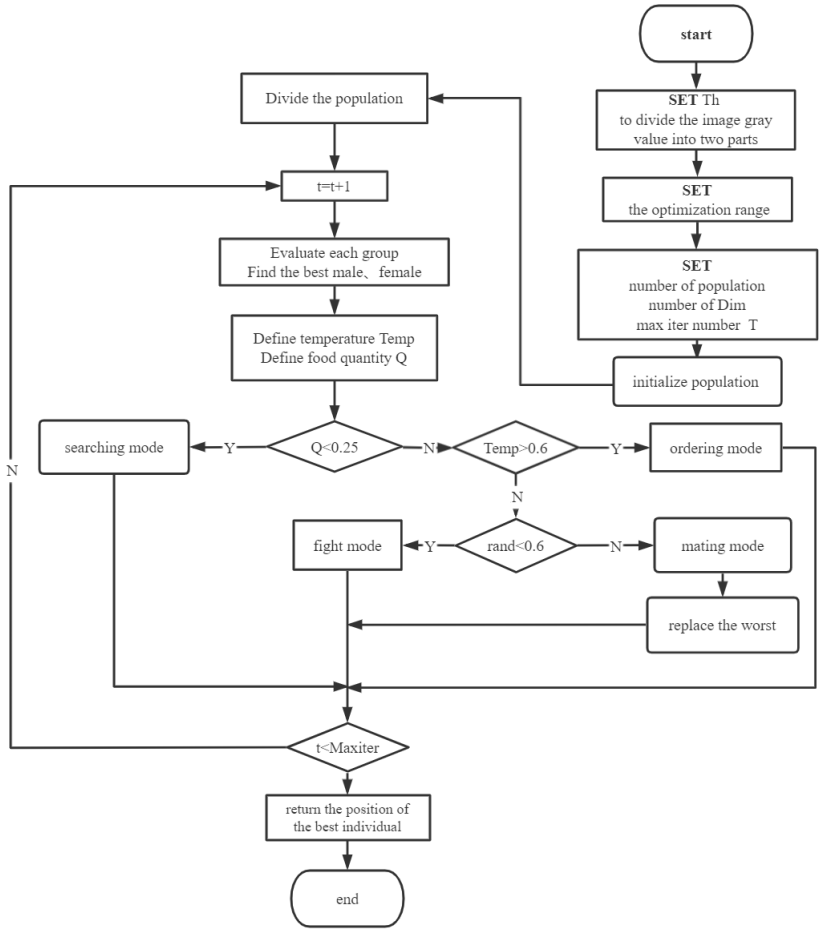

Figure 7. Flow chart of the snake optimization algorithm.

4. Preliminaries

4.1. Experimental data set

In order to verify the effectiveness of the algorithm, three rubber tree images with tapping panel dryness were simulated and generated. The three images with different TPD symptoms are numbered as A, B and C. There are different characteristics in rubber tree secant.

In Figure 8, we can see that the rubber fluid has the spotty distribution feature. In Figure 9, the boundary between the region of interest and the background is clear. The situation in Figure 10 is between the previous two figures: the latex presents a lumpy distribution but there is a fuzzy boundary between the target area and the background.

4.2. Evaluation indicators

Evaluation of a segmented image is a critical step to measure whether the process of segmentation has been done accurately [52]. For evaluating the performance and quality of segmented images at different levels of thresholding, several techniques are used to measure the segmentation goodness. Peak

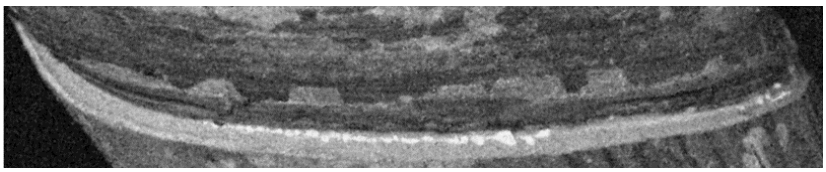

Figure 8. A: latex distributed in a point pattern.

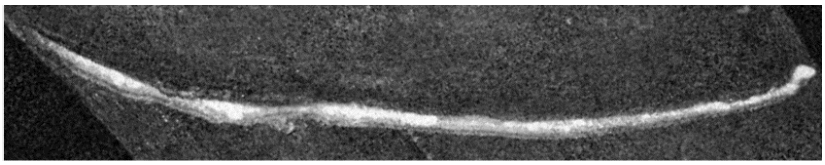

Figure 9. B: clear boundary between the region of interest and the background.

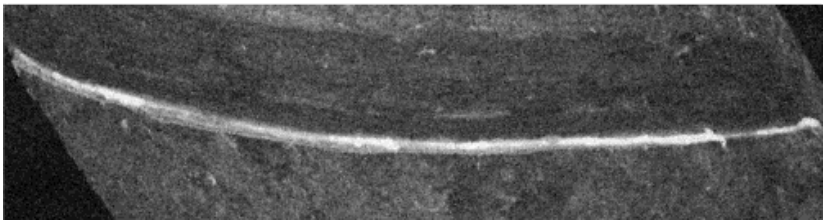

Figure 10. C: clear boundary between the region of interest and the background.

Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Feature Similarity Index Measure are the three metrics employed in this paper (FSIM).

4.2.1. PSNR

PSNR is a quality validation metric used to find the quality difference among the original image and segmented one [53]. PSNR can be described as follows:

$$PSNR = 20\log_{10}\frac{255}{RMSE} \tag{4.1}$$

where $RMSE$ stands for the Root Mean Squared Error, which can be described as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left((I(i,j) - \mathrm{Seg}(i,j))^2\right)}{M \times N}} \tag{4.2}$$

4.2.2. FSIM

FSIM is a measure used to map the features to measure the similarity between the original and the segmented images [54].

FSIM depends on two major criteria: phase consistency (PC) and gradient magnitude (GM). To ease the process of features detection, PC approach is used. To compute the image gradient, GM approach is used. At first, the similarity among two images can be determined as follows:

$$S_{PC} = \frac{2PC_1PC_2 + T_1}{PC_1^2 + PC_2^2 + T_1} \tag{4.3}$$

where PC_1 and PC_2 are the Phase Congruence (PC) for original and segmented images, respectively. To increase the stability of S_{PC}, T_1 is used.

Then, S_G is calculated.

$$S_G = \frac{2G_1G_2 + T_2}{G_1^2 + G_2^2 + T_2} \tag{4.4}$$

where G_1 represents the gradient of the first image (original), and G_2 represents the gradient of the second image (segmented), and T_2 is a positive constant number. From Eqs (4.3) and (4.4), the similarity is calculated as follows:

$$S_L(x) = [S_{PC}(x)]^{\alpha} [S_G(x)]^{\beta} \tag{4.5}$$

where α and β are used to adjust the importance of $PC$and GM.

4.2.3. SSIM

SSIM is a measure that is used to compute the similarity among the original and segmented images [55], SSIM can be described as follows:

$$\text{SSIM}(I, Seg) = \frac{\left(2\mu_1\mu_{\text{Seg}} + c_1\right)\left(2\sigma_{1,\text{Seg}} + c_2\right)}{\left(\mu_I^2 + \mu_{\text{Seg}}^2 + c1\right)\left(\sigma_I^1 + \sigma_{\text{Seg}}^2 + c2\right)} \tag{4.6}$$

The SSIM value ranges from 0 to 1, and the larger it is the more similar the image is. If the two images are identical, the SSIM value is 1.

4.3. Parameter setting of the testing algorithm

To evaluate the performance of the proposed SO-Otsu, we used three pictures from Subsection 4.1 as the benchmark images. Due to the stochastic nature of the meta-heuristics algorithm, the results change in every execution. Here, the experiments have been performed 30 times for each algorithm. The performance of proposed approach was compared with six algorithms: the original Otsu's method, FOA-Otsu, SSA-Otsu, GWO-Otsu, WOA-Otsu and HHO-Otsu. Each algorithm's parameter settings are displayed in Table 1.

All these experiments have been performed using MATLAB R2016a on 10 Windows Operating System with 16 GB RAM memory and AMD Ryzen 7 4800U CPU.

Table 1. Parameter setting of the testing algorithm.

Algorithm	Parameter	Setting
FOA-Otsu	maxgen	1000
	popsize	100
SSA-Otsu	SearchAgents_no	100
	Max_iteration	1000
GWO-Otsu	MaxIter	1000
	pop	100
WOA-Otsu	SearchAgents_no	100
	Max_iteration	1000
HHO-Otsu	E_0	$[-1, 1]$
SO-Otsu	C1	0.5
	C2	0.05
	C3	2
	Max_iteration_no	1000
	SearchAgents_no	100
	Th	161

5. Analysis of results

5.1. Detail review

The selected thresholds are shown as Table 2. The entire image is split into four areas based on the thresholds. The value of each pixel is determined by the region to which it belongs. These particular values in this study are 0, 100, 190 and 255.

The full process of the detailed evaluation is shown as Figure 11. In order to better display and evaluate the details, we cut out the rubber tree secant from Figures 8–10. What's more, a dotted line (shown as the blue dotted lines in Figures 12–14) is selected at the proper region, which contains more detail information of rubber tree secant.

According to the dotted line, the statistical information of gray level is obtained and it is shown in Figures 15–17. In each sub-graph, the horizontal axis represents the position of each pixel on the sampling line, while the vertical axis represents the gray value of the corresponding pixel.

In these figures, the first seven curves correspond to the statistics of each method. The last one is the statistics of the original after gray processing. By comparing how similar each curve is to the final one in every group, each method can be evaluated on the performance in segmenting details.

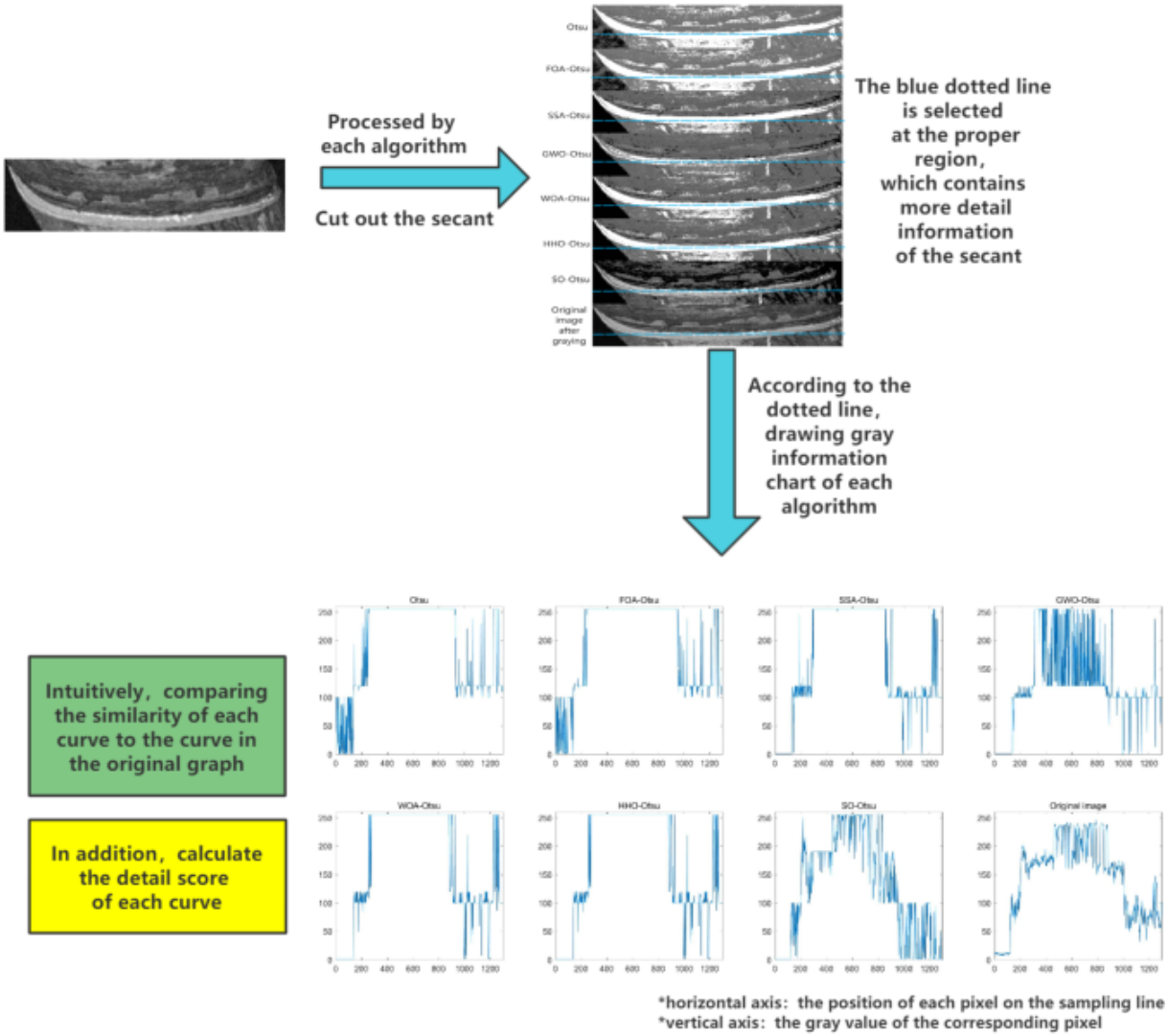

Figure 11. The full process of the detailed evaluation.

In addition, we set a score for curve similarity:

$$SCORE = \sum_{i=1}^{L} \text{score}(i)$$
$$\text{score}(i) = \begin{cases} 1, & |a(i) - b(i)| \leq 20 \\ -1, & |a(i) - b(i)| > 20 \end{cases} \tag{5.1}$$

where i represents the horizontal axis coordinate of the curve, L is the maximum length of the horizontal axis, $a(i)$ represents the specific value of the processed curve at i, $b(i)$ represents the specific value of the original curve at i.

When the value difference between the two curves at a point is small, it is recorded as one point. If the difference is large, one point is deducted as a penalty. Finally, adding up the $score(i)$ to get

SCORE. By comparing the scores in Table 2, we can see how well the algorithm is segmenting in detail.

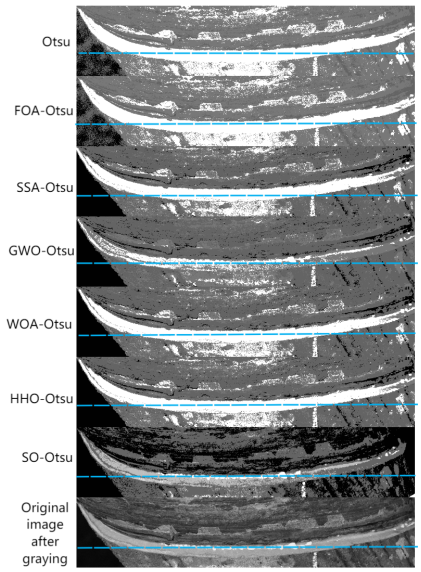

Figure 12. Group A.

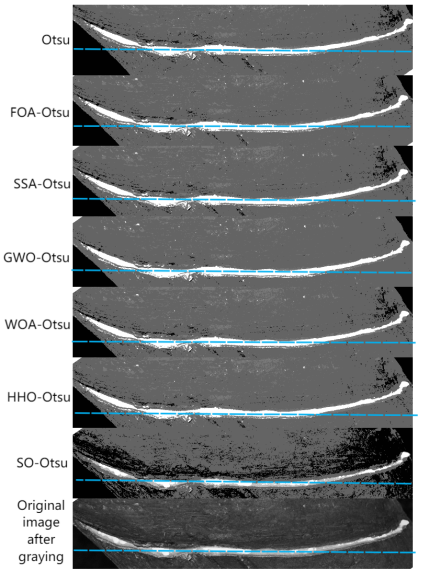

Figure 13. Group B.

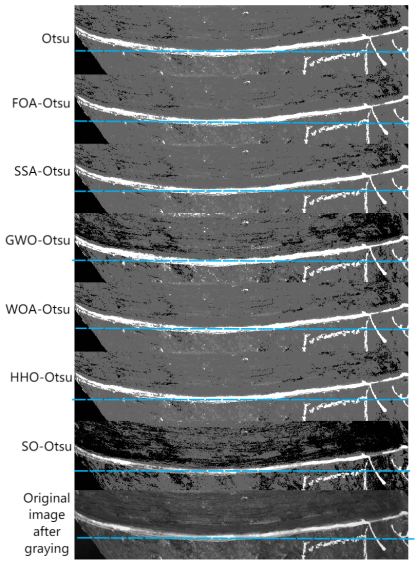

Figure 14. Group C.

Table 2. Detail score results.

Group	Otsu	FOA-Otsu	SSA-Otsu	GWO-Otsu	WOA-Otsu	HHO-Otsu	SO-Otsu
A	126	27	593	1025	473	473	1096
B	-163	426	-154	-311	-261	-261	883
C	583	562	619	392	583	583	909
Mean	182	348	353	369	265	265	962

In each group, SO-Otsu achieved the highest detail score, which proves that the results of SO-Otsu processing are more consistent with the details of the original image than other algorithms.

In Figures 15–17, on the intuitive comparison of curve similarity, we have made the following analysis.

Group A: As shown in Figure 15, in the range of $[0, 100]$, the statistics of each method are very similar to the last one, except for FOA-Otsu. In the range of $[101, 255]$, SO-Otsu provides the best results.The results of SO-Otsu are more similar to those of the final one. It reflects the fluctuation of gray levels in the range of $[101, 255]$. For the other five methods, the pixels with different gray levels are classified into the same category. In summary, SO-Otsu can distinguish the rubber tree secant from the rubber fluid above it. SO-Otsu has a better performance in segmenting details.

Group B: As shown in Figure 16, in the whole range,the results of SO-Otsu are very similar to those of the final one. The curve of the original image ebbs and flows. In contrast, the other five methods cannot reflect the statistical trend of the original image, because the curve of the original image is mapped to a straight line. SO-Otsu clearly reflects the changing trend of the original image. Obviously, SO-Otsu performs better than the other methods in segmenting details.

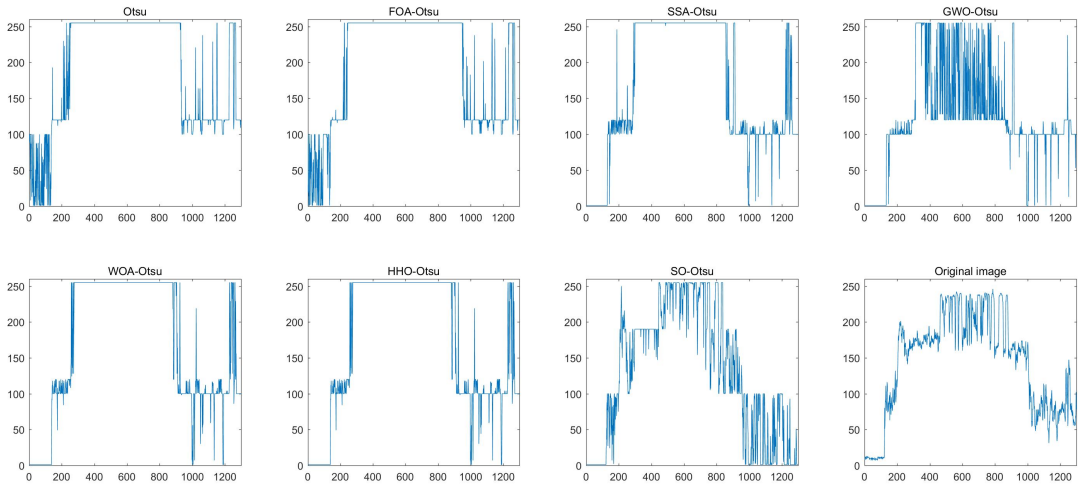

Figure 15. Statistical results of gray-level of group A.

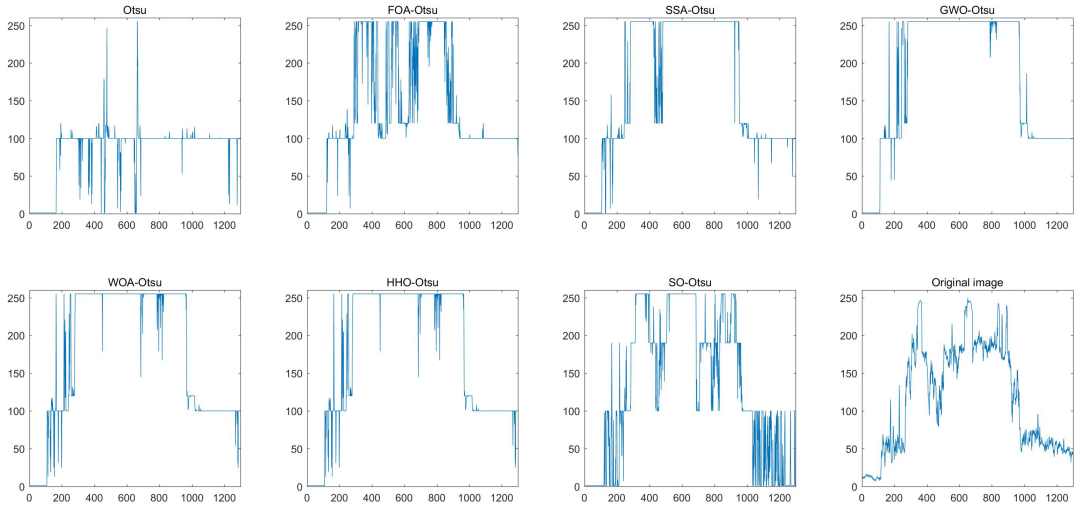

Figure 16. Statistical results of gray-level of group B.

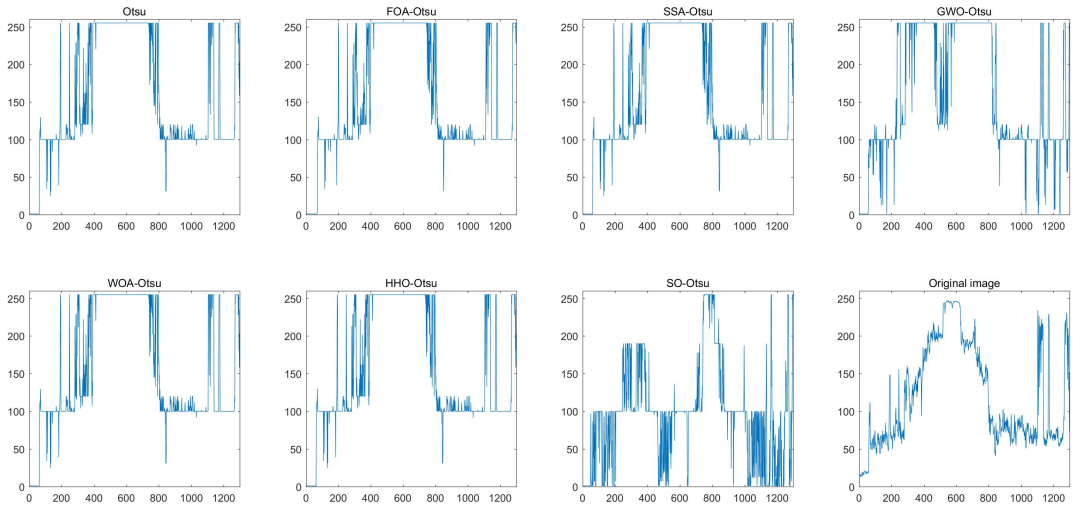

Figure 17. Statistical results of gray-level of group C.

Group C: As shown in Figure 17, in the range of [0, 100], the statistical trend of SO-Otsu is similar to that of the final one. In the range of [101, 255], each method performs poorly, except for SO-Otsu. The results of the other five methods cannot show the trend of gray levels changing.

In conclusion, SO-Otsu can distinguish the rubber tree secant and the rubber fluid. It has a better performance in segmenting details.

5.2. Indicator reviews

In this section, SO-Otsu is compared with six other algorithms: FOA-Otsu, SSA-Otsu, GWO-Otsu, WOA-Otsu, HHO-Otsu and the original Otsu's method. The indicators include: running time, PSNR, FSIM and SSIM. The running time is related to the real-time environment and is an important objective criterion. The operation results are shown in Table 3.

Table 3. Test results of seven algorithms.

Group	Algorithm	Time	Threshold	PSNR	SSIM	FSIM
A	Otsu	35.814	10, 53, 113	10.702	0.9967	0.6765
	FOA-Otsu	1.031	10, 60, 111	13.308	0.9990	0.6116
	SSA-Otsu	0.944	47, 93, 131	15.828	**0.9992**	0.7773
	GWO-Otsu	0.929	61, 98, 148	**20.851**	0.9949	0.7794
	WOA-Otsu	1.071	45, 94, 135	16.039	0.9990	**0.7796**
	HHO-Otsu	1.465	45, 94, 135	16.039	0.9990	**0.7796**
	SO-Otsu	**0.383**	78, 161, 194	17.620	0.9955	0.7490
B	Otsu	41.302	36, 79, 139	17.785	**0.9999**	0.8542
	FOA-Otsu	1.216	34, 79, 137	17.691	**0.9999**	**0.8572**
	SSA-Otsu	0.971	36, 79, 138	17.758	**0.9999**	0.8551
	GWO-Otsu	1.192	35, 86, 140	17.371	**0.9999**	0.8501
	WOA-Otsu	1.167	36, 79, 139	17.785	**0.9999**	0.8542
	HHO-Otsu	1.507	36, 79, 139	17.785	**0.9999**	0.8542
	SO-Otsu	**0.457**	53, 161, 201	**19.228**	0.9997	0.6942
C	Otsu	41.536	43, 88, 141	18.350	**0.9995**	0.8374
	FOA-Otsu	1.229	42, 89, 144	18.581	0.9994	**0.8384**
	SSA-Otsu	0.930	43, 88, 141	18.601	0.9994	0.8368
	GWO-Otsu	1.137	24, 108, 171	16.807	**0.9995**	0.7788
	WOA-Otsu	1.192	43, 88, 141	18.350	0.9989	0.8374
	HHO-Otsu	1.484	43, 88, 141	18.350	0.9989	0.8374
	SO-Otsu	**0.480**	64, 161, 203	**18.834**	0.9993	0.7317

* The bold shows the best score for a certain indicator.

In order to avoid the chance of experimental results, we recorded the fitness obtained in 30 experiments. The statistical results are then shown in Table 4, where Min represents the minimum value, Max represents the maximum value, Avg represents the average value and Std represents the standard deviation.

Table 4. The comparison results of all algorithms.

Group	Algorithm	Min	Max	Avg	Std
A	FOA-Otsu	-2097.4761	-2073.2134	-2083.5962	17.1563
	SSA-Otsu	-2090.2649	-2085.2374	-2087.1733	2.2106
	GWO-Otsu	-2028.5177	-2001.1134	-2042.9591	35.1368
	WOA-Otsu	-2099.3886	-2085.9177	-2092.6531	7.0998
	HHO-Otsu	-2099.3886	-2085.9177	-2092.6531	7.0998
	SO-Otsu	-1738.0143	-1738.0143	-1738.0143	0
B	FOA-Otsu	-1343.7825	-1330.6645	-1337.4671	7.3039
	SSA-Otsu	-1355.2036	-1357.6066	-1356.6169	1.1588
	GWO-Otsu	-1324.0171	-1265.4254	-1297.5898	28.4313
	WOA-Otsu	-1357.6066	-1357.6066	-1357.6066	0
	HHO-Otsu	-1357.6066	-1357.6066	-1357.6066	0
	SO-Otsu	-842.1048	-840.5693	-841.0466	0.7275
C	FOA-Otsu	-1478.6927	-1417.8281	-1472.1903	17.3877
	SSA-Otsu	-1527.7611	-1518.7764	-1526.0685	4.2035
	GWO-Otsu	-1450.9134	-1523.8241	-1494.3500	31.5665
	WOA-Otsu	-1527.6912	-1527.6912	-1527.6912	0
	HHO-Otsu	-1527.6912	-1527.6912	-1527.6912	0
	SO-Otsu	-1194.0617	-1190.9742	-1192.7038	1.3736

We plot convergence curves (shown in Figure 18) for the optimization algorithm used to optimize the threshold in the experiments to make the comparison results more intuitive. In each sub-graph, the horizontal axis represents the number of iterations, while the vertical axis represents the best fitness obtained so far.

The convergence diagram indicates that each group undergoes a similar change throughout the iterations: the best fitness obtained gradually converges as the iteration progresses. Additionally, it can be observed that curve SO exhibits a rapid convergence rate, which can be attributed to the optimizing effect of the SO algorithm. However, it is evident that the SO curve does not converge to the identical value as the other algorithms. The reason for this discrepancy is that the optimization process employed by the SO-Otsu algorithm involves an enhanced version of the Otsu's method proposed by us, while the optimization employed by the other algorithms utilizes the original Otsu's method.

In terms of time, compared with the original Otsu's method, SO-Otsu has a huge improvement in running time. What's more, SO-Otsu reduces the running time by 50% compared with the other four methods.

Thanks to the strategy of the SO-Otsu, the selected thresholds are more proper. Because they are not limited to the range with more pixels (just like the other six methods), which means that the interested region with less pixels could get more attention. The proper thresholds lead to better image processing results. This is the reason why SO-Otsu performs best in detail reviews.

For the PSNR mean values, SO-Otsu is the first. In group A, the results of SO-Otsu are ranked in the second, GWO-Otsu provides the highest PSNR. However, compared with the original Otsu's method, the results of SO-Otsu have a 70% increase. In group B, SO-Otsu provides the highest PSNR.

Compared with the other five methods, the results of SO-Otsu have a 10% increase in group B. And in group C, the results obtained by the six methods show little difference, but SO-Otsu provides the highest PSNR. In summary, SO-Otsu has a better performance in PSNR, which means that it has less distortion.

In terms of SSIM mean values, the variation between the six methods is negligible, with all SSIM values exceeding 0.9990. This suggests that all methods perform well in preserving structural similarity.

Regarding FSIM mean values, the performance of the SO-Otsu algorithm is not particularly strong. In group A, it ranks fourth, while in groups B and C, it ranks last. This indicates that the algorithm is not effective at preserving Feature Similarity.

Based on the statistical data presented in Table 4, it is evident that the convergence range of SO-Otsu differs from that of other algorithms, as verified by Figure 18. Further analysis of the mean value and standard deviation indicates that the outcomes yielded by SO-Otsu exhibit a higher level of consistency. Furthermore, it can be observed that the WOA-Otsu and HHO-Otsu algorithms yield identical outcomes. Moreover, both Group B and Group C exhibit a standard deviation of 0, implying that these two algorithms are highly stable. Conversely, GWO-Otsu demonstrates a considerably large standard deviation in each group, indicating that GWO-Otsu has a poor algorithmic stability.

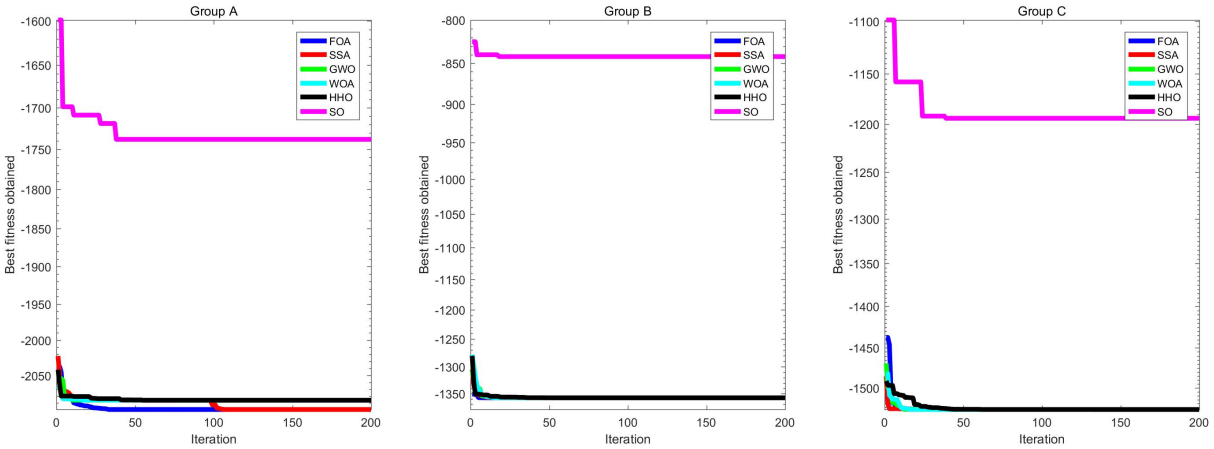

Figure 18. Convergence curve of every group for all algorithm.

In conclusion, SO-Otsu obtains excellent results in the test of PSNR , which means that SO-Otsu has a great performance in reducing distortion. Based on the statistical findings, it can be concluded that SO-Otsu exhibits high stability. Moreover, SO-Otsu has the ability to preserve additional details of the original image during segmentation, resulting in improved performance in capturing finer segmentation details. Additionally, the optimization strategy employed in SO-Otsu results in reduced computational complexity compared to alternative algorithms, enabling it to perform well in terms of execution time evaluation.

On the one hand, it is regrettable that SO-Otsu did not achieve the top score in the SSIM evaluation. However, the scores of several algorithms were only slightly different, within a margin of one thousandth, which we consider acceptable. On the other hand, in the FSIM evaluation, SO-Otsu did not perform as well, indicating that it lacks an advantage in preserving feature stability.

6. Summary

In order to strengthen the ability of segmenting details, we improve the Otsu's method. To solve the multilevel thresholding problem, snake optimizer is used as the optimization function. Combining the above two aspects, SO-Otsu is proposed for segmenting rubber tree secant. In the detail review, SO-Otsu provides excellent segmenting results. SO-Otsu outperforms the other five methods. In the indicator reviews, SO-Otsu has the best performance in running time and PSNR. Therefore, SO-Otsu has Faster computation speed and less distortion. In terms of FSIM, SO-Otsu obtains a poor grade. However, it is worth sacrificing FSIM score for faster computation speed and better segmentation details.

For future works, it is recommended to apply SO-Otsu in the situation where higher computational efficiency is sought. Also, SO-Otsu is likely to perform excellently when the region of interest is smaller and its gray levels are different from other areas.

Acknowledgments

This work was supported by Hainan Provincial Natural Science Foundation of China (120QN178). This work was also supported by Hainan University Scientific Research Startup Funding Project (KYQD(ZR)-21082) and Hainan University Scientific Research Startup Funding Project (kyqd(sk)2022013).

Conflict of interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

References

1. C. Nayanakantha, Tapping panel dryness: The killer affecting the productivity of rubber plantations, **19** (2021), 26–29.
2. R. Putranto, E. Herlinawati, M. Rio, J. Leclercq, P. Piyatrakul, E. Gohet, et al., Involvement of ethylene in the latex metabolism and tapping panel dryness of hevea brasiliensis, *Int. J. Mol. Sci.*, **16** (2015), 17885–17908. https://doi.org/10.3390/ijms160817885
3. Z. Sun, J. Xing, H. Hun, X. Zhang, X. Dong, Y. Deng, Research on recognition and planning of tapping trajectory of natural rubber tree based on machine vision, *J. Chin. Agric. Machanization*, **43** (2022), 102–108. https://doi.org/10.13733/j.jcam.issn.20955553.2022.05.015
4. J. Zhang, Y. Liu, H. Xing, Application of improved 2-d entropy algorithm in rubber tree image segmentation, in *2019 2nd International Conference on Safety Produce Informatization (IICSPI)*, (2019), 311–314. https://doi.org/10.1109/IICSPI48186.2019.9096014
5. S. Li, J. Zhang, J. Zhang, L. Sun, Y. Liu, Study on the secant segmentation algorithm of rubber tree, *J. Phys. Conf. Ser.*, **1004** (2018), 012033. https://doi.org/10.1088/1742-6596/1004/1/012033

6. P. Parvati, B. Rao, M. Das, Image segmentation using gray-scale morphology and marker-controlled watershed transformation, *Discrete Dyn. Nat. Soc.*, **2008** (2008). https://doi.org/10.1155/2008/384346
7. P. Sathya, R. Kalyani, V. Sakthivel, Color image segmentation using kapur, otsu and minimum cross entropy functions based on exchange market algorithm, *Expert Syst. Appl.*, **172** (2021), 114636. https://doi.org/10.1016/j.eswa.2021.114636
8. X. Wang, S. Wang, Y. Guo, K. Hu, W. Wang, Coal gangue image segmentation method based on edge detection theory of star algorithm, *Int. J. Coal Prep. Util.*, **43** (2023), 119–134. https://doi.org/10.1080/19392699.2021.2024173
9. H. Yu, P. Sun, F. He, Z. Hu, A weighted region-based level set method for image segmentation with intensity inhomogeneity, *PLoS One*, **16** (2021), e0255948. https://doi.org/10.1371/journal.pone.0255948
10. D. Wei, Z. Wang, L. Si, C. Tan, X. Lu, An image segmentation method based on a modified local-information weighted intuitionistic fuzzy c-means clustering and gold-panning algorithm, *Eng. Appl. Artif. Intell.*, **101** (2021), 104209. https://doi.org/10.1016/j.engappai.2021.104209
11. P. Ghamisi, M. Couceiro, J. Benediktsson, N. Ferreira, An efficient method for segmentation of image based on fractional calculus and natural selection, *Expert Syst. Appl.*, **39** (2012), 12407–12417. https://doi.org/10.1016/j.eswa.2012.04.078
12. K. Kumar, K. Venkatalakshmi, K. Krishnan, Lung cancer detection using image segmentation by means of various evolutionary algorithms, *Comput. Math. Methods Med.*, **2019** (2019), 1–16. https://doi.org/10.1155/2019/4909846
13. Z. Wakaf, H. Jalab, Defect detection based on extreme edge of defective region histogram, *J. King Saud Univ. Comput. Inf. Sci.*, **30** (2018), 33–40. https://doi.org/10.1016/j.jksuci.2016.11.001
14. L. Zhang, A. Li, X. Li, S. Xu, X. Yang, Remote sensing image segmentation based on an improved 2-d gradient histogram and mmad model, *IEEE Geosci. Remote Sens. Lett.*, **12** (2015), 58–62. https://doi.org/10.1109/LGRS.2014.2326008
15. Y. Xie, L. Ning, M. Wang, C. Li, Image enhancement based on histogram equalization, *J. Phys. Conf. Ser.*, **1314** (2019), 012161. https://doi.org/10.1088/1742-6596/1314/1/012161
16. M. Abd Elaziz, A. A. Ewees, D. Oliva, Hyper-heuristic method for multi-level thresholding image segmentation, *Expert Syst. Appl.*, **146** (2020), 113201. https://doi.org/10.1016/j.eswa.2020.113201
17. S. Aja-Fernández, A. H. Curiale, G. Vegas-Sánchez-Ferrero, A local fuzzy thresholding methodology for multiregion image segmentation, *Knowl.-Based Syst.* , **83** (2015), 1–12. https://doi.org/10.1016/j.knosys.2015.02.029
18. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.*, **9** (1979), 62–66. https://doi.org/10.1109/TSMC.1979.4310076
19. J. N. Kapur, P. K. Sahoo, A. K. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, *Comput. Vision, Graphics, Image Process.*, **29** (1985), 273–285. https://doi.org/10.1016/0734-189X(85)90125-2

20. S. Agrawal, R. Panda, S. Bhuyan, B. K. Panigrahi, Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm, *Swarm Evol. Comput.*, **11** (2013), 16–30. https://doi.org/10.1016/j.swevo.2013.02.001
21. C. H. Li, C. Lee, Minimum cross entropy thresholding, *Pattern Recognit.*, **26** (1993), 617–625. https://doi.org/10.1016/0031-3203(93)90115-D
22. P. Y. Yin, Multilevel minimum cross entropy threshold selection based on particle swarm optimization, *Appl. Math. Comput.*, **184** (2007), 503–513. https://doi.org/10.1016/j.amc.2006.06.057
23. D. Wolpert, W. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, **1** (1997), 67–82. https://doi.org/10.1109/4235.585893
24. W. T. Pan, A new fruit fly optimization algorithm: Taking the financial distress model as an example, *Knowl.-Based Syst.*, **26** (2012), 69–74. https://doi.org/10.1016/j.knosys.2011.07.001
25. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
26. S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.*, **111** (2020), 300–323. https://doi.org/10.1016/j.future.2020.03.055
27. Y. Feng, S. Deb, G. G. Wang, A. H. Alavi, Monarch butterfly optimization: A comprehensive review, *Expert Syst. Appl.*, **168** (2021), 114418. https://doi.org/10.1016/j.eswa.2020.114418
28. J. Tu, H. Chen, M. Wang, A. Gandomi, The colony predation algorithm, *J. Bionic Eng.*, **18** (2021), 674–710. https://doi.org/10.1007/s42235-021-0050-y
29. G. G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, *Memet. Comput.*, **10** (2018), 151–164. https://doi.org/10.1007/s12293-016-0212-3
30. Y. Yang, H. Chen, A. A. Heidari, A. H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.*, **177** (2021), 114864. https://doi.org/10.1016/j.eswa.2021.114864
31. I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. Chen, A. H. Gandomi, Info: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.*, **195** (2022), 116516. https://doi.org/10.1016/j.eswa.2022.116516
32. I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, H. Chen, Run beyond the metaphor: An efficient optimization algorithm based on runge kutta method, *Expert Syst. Appl.*, **181** (2021), 115079. https://doi.org/10.1016/j.eswa.2021.115079
33. M. A. Awadallah, M. A. Al-Betar, M. S. Braik, A. I. Hammouri, I. A. Doush, R. A. Zitar, An enhanced binary rat swarm optimizer based on local-best concepts of pso and collaborative crossover operators for feature selection, *Comput. Biol. Med.*, **147** (2022), 105675. https://doi.org/10.1016/j.compbiomed.2022.105675
34. S. Thawkar, S. Sharma, M. Khanna, L. kumar Singh, Breast cancer prediction using a hybrid method based on butterfly optimization algorithm and ant lion optimizer, *Comput. Biol. Med.*, **139** (2021), 104968. https://doi.org/10.1016/j.compbiomed.2021.104968

35. S. Chakraborty, A. K. Saha, S. Nama, S. Debnath, Covid-19 x-ray image segmentation by modified whale optimization algorithm with population reduction, *Comput. Biol. Med.*, **139** (2021), 104984. https://doi.org/10.1016/j.compbiomed.2021.104984
36. G. I. Sayed, M. M. Soliman, A. E. Hassanien, A novel melanoma prediction model for imbalanced data using optimized squeezenet by bald eagle search optimization, *Comput. Biol. Med.*, **136** (2021), 104712. https://doi.org/10.1016/j.compbiomed.2021.104712
37. M. A. Awadallah, A. I. Hammouri, M. A. Al-Betar, M. S. Braik, M. A. Elaziz, Binary horse herd optimization algorithm with crossover operators for feature selection, *Comput. Biol. Med.*, **141** (2022), 105152. https://doi.org/10.1016/j.compbiomed.2021.105152
38. J. K. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Syst. Sci. Control Eng.*, **8** (2020), 22–34. https://doi.org/10.1080/21642583.2019.1708830
39. S. Mirjalili, S. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
40. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
41. C. Huang, X. Li, Y. Wen, An otsu image segmentation based on fruitfly optimization algorithm, *Alexandria Eng. J.*, **60** (2021), 183–188. https://doi.org/10.1016/j.aej.2020.06.054
42. A. Bhandari, A. Kumar, G. Singh, Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, otsu and tsallis functions, *Expert Syst. Appl.*, **42** (2015), 1573–1601. https://doi.org/10.1016/j.eswa.2014.09.049
43. E. Houssein, D. Abdelkareem, M. Emam, M. Hameed, M. Younan, An efficient image segmentation method for skin cancer imaging using improved golden jackal optimization algorithm, *Comput. Biol. Med.*, **149** (2022), 106075. https://doi.org/10.1016/j.compbiomed.2022.106075
44. Z. Zhang, J. Yin, Bee foraging algorithm based multi-level thresholding for image segmentation, *IEEE Access*, **8** (2020), 16269–16280. https://doi.org/10.1109/ACCESS.2020.2966665
45. M. Abdel-Basset, R. Mohamed, N. AbdelAziz, M. Abouhawwash, Hwoa: A hybrid whale optimization algorithm with a novel local minima avoidance method for multi-level thresholding color image segmentation, *Expert Syst. Appl.*, **190** (2021), 116145. https://doi.org/10.1016/j.eswa.2021.116145
46. G. Kang, S. Gao, L. Yu, D. Zhang, Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning, *IEEE Trans. Instrum. Meas.*, **68** (2018), 2679–2690. https://doi.org/10.1109/TIM.2018.2868490
47. Y. Zhan, G. Zhang, An improved otsu algorithm using histogram accumulation moment for ore segmentation, *Symmetry*, **11** (2019), 431. https://doi.org/10.3390/sym11030431
48. X. Xu, S. Xu, L. Jin, E. Song, Characteristic analysis of otsu threshold and its applications, *Pattern Recognit. Lett.*, **32** (2011), 956–961. https://doi.org/10.1016/j.patrec.2011.01.021
49. S. Tripathi, K. Kumar, B. Singh, R. Singh, Image segmentation: A review, *Int. J. Comput. Sci. Manage. Res.*, **1** (2012), 838–843.
50. P. Sathya, R. Kayalvizhi, Optimal multilevel thresholding using bacterial foraging algorithm, *Expert Syst. Appl.*, **38** (2011), 15549–15564. https://doi.org/10.1016/j.eswa.2011.06.004

51. F. A.Hashim, A. Hussien, Snake optimizer: A novel meta-heuristic optimization algorithm, *Knowl.-Based Syst.*, **242** (2022), 108320. https://doi.org/10.1016/j.knosys.2022.108320

52. L. Qingge, R. Zheng, X. Zhao, S. Wei, P. Yang, An improved otsu threshold segmentation algorithm, *Int. J. Comput. Sci. Eng.*, **22** (2020), 146–153. https://doi.org/10.1504/IJCSE.2020.107266

53. M. H. Horng, Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation, *Expert Syst. Appl.*, **38** (2011), 13785–13791. https://doi.org/10.1016/j.eswa.2011.04.180

54. S. Sarkar, S. Das, S. Chaudhuri, A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution, *Pattern Recognit. Lett.*, **54** (2015), 27–35. https://doi.org/10.1016/j.patrec.2014.11.009

55. D. Oliva, V. Osuna-Enciso, E. Cuevas, G. Pajares, M. Cisneros, D. Zaldivar, Improving segmentation velocity using an evolutionary method, *Expert Syst. Appl.*, **42** (2015), 5874–5886. https://doi.org/10.1016/j.eswa.2015.03.028

AIMS Press