EIRA Journal of Multidisciplinary Research and Development (EIRAJMRD)
Journal homepage: https://eirapublisher.com/eirajmrd/
Email: office.eirapublisher@gmail.com
ISSN: 3107-9350 (Online)
Volume 2, Issue 4, (July-August) 2026
DOI: https://doi.org/10.5281/zenodo.21766978

Performance Analysis of Selected Chaos Maps to Improve Particle Swarm Optimization in Image Segmentation Tasks

Olubusola Eniayo John[1], Adebisi Abimbola Baale[2] & Stephen Olatunde Olabiyisi[3]

1,3 Department of Computer Science, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria.
2 Department of Information Systems, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria.

Article history:
Received: 28/05/2026
Accepted: 07/07/2026
Published: 03/08/2026

Keywords: *Image Segmentation, Particle Swarm Optimization, Chaos-Enhanced Metaheuristics, Apple Image Dataset*

***Corresponding Author**:
Olubusola Eniayo John

Abstract **Original Research Article**

Image segmentation remains a fundamental challenge in computer vision, where standard optimization algorithms such as Particle Swarm Optimization (PSO) often suffer from premature convergence and loss of population diversity, which limits segmentation accuracy. This study integrates chaotic maps into the PSO algorithm to improve population diversity, prevent premature convergence, and enhance segmentation quality. The aim is to evaluate three chaos-enhanced PSO (CE-PSO) variants, Logistic+PSO, Sinusoidal+PSO and Circle+PSO, for image segmentation using the Apple Image Dataset obtained from Kaggle. Chaotic sequences were used in place of the pseudo-random number generators employed for swarm initialization, parameter adaptation, and particle-position updates. The models were implemented in Python and evaluated on the preprocessed Apple Image Dataset using PSNR, SSIM, FSIM, convergence speed, and computational time, averaged over 30 independent runs per algorithm. Sinusoidal+PSO achieved the best results, with a PSNR of 27.89 ± 0.98 dB, SSIM of 0.9123 ± 0.018, and FSIM of 0.8967 ± 0.021, followed by Circle+PSO (27.12 ± 1.05 dB, 0.9034 ± 0.019, 0.8856 ± 0.023) and Logistic+PSO (26.78 ± 1.12 dB, 0.8912 ± 0.021, 0.8745 ± 0.025). Sinusoidal+PSO also converged fastest, reaching its solution in an average of 41 iterations, while Circle+PSO required the least computational time (2.38 ± 0.21 s). The study establishes that Sinusoidal+PSO offers the best accuracy for high-precision applications such as agricultural image analysis and automated fruit inspection, Circle+PSO offers the best computational efficiency for real-time, time-critical vision systems, and Logistic+PSO offers a computationally simple, general-purpose baseline.

How to cite this article: John, O. E., Baale, A. A., & Olabiyisi, S. O. (2026). Performance analysis of selected chaos maps to improve particle swarm optimization in image segmentation tasks. *EIRA Journal of Multidisciplinary Research and Development, 2*(4), 55–65.

1.0 INTRODUCTION

Image segmentation, which plays a vital role in computer vision and image processing, refers to the process of partitioning an image into meaningful regions to support object detection, recognition, and interpretation. Despite its importance, achieving accurate segmentation remains challenging because of variations in illumination, image noise, intensity inhomogeneity, and complex object boundaries, challenges that are particularly pronounced in agricultural imaging, where inconsistent lighting and cluttered backgrounds complicate the extraction of fruit or disease regions (Song et al., 2024). Its applications span diverse fields such as medical imaging, agricultural image analysis, industrial inspection, autonomous navigation, and satellite image analysis (Gonzalez and Woods, 2018; Amiriebrahimabadi et al., 2024).

Among metaheuristics, Particle Swarm Optimization (PSO) has gained widespread acceptance for image segmentation because of its simple implementation, fast convergence, and strong global search capability (Kennedy and Eberhart, 1995; Poli et al., 2007). Unfortunately, the standard PSO algorithm suffers from premature convergence, loss of population diversity, and stagnation in local optima when solving complex optimization problems, a weakness that recent convergence analyses confirm persists even in constriction-factor variants of PSO (Clerc and Kennedy, 2002; Tarekegn Nigatu et al., 2024). To address these shortcomings, researchers have introduced chaos theory into the PSO framework; chaotic maps generate deterministic but non-repetitive sequences with ergodic properties that improve particle diversity, enhance exploration of the search space, and reduce the likelihood of premature convergence (Alatas et al., 2009). Several chaotic maps, including the Logistic, Sinusoidal, and Circle maps, have been investigated for enhancing optimization algorithms; however, their comparative effectiveness across different metaheuristics and problem domains remains an active area of research, since the benefit of a given chaotic map has been shown to depend strongly on the algorithm and search space to which it is applied (Cisternas-Caneo et al., 2024).

Traditional image segmentation techniques such as thresholding, clustering, edge detection, and region growing often produce unsatisfactory results when applied to complex images with overlapping intensity distributions and noisy backgrounds, and become computationally expensive as the number of threshold levels increases (Pal and Pal, 1993; Sahoo et al., 1988; Sezgin and Sankur, 2004). Recent studies have shown that integrating chaotic maps into PSO improves population diversity, enhances the balance between exploration and exploitation, and accelerates convergence during optimization (Alatas et al., 2009; Gao et al., 2010). However, most existing studies investigate individual chaotic maps independently, making it difficult to determine which chaotic map provides the best performance under identical experimental conditions, and few studies have provided a systematic comparison of the logistic, sinusoidal, and circle chaotic maps within a unified PSO framework using the same dataset, implementation settings, and evaluation metrics. This study therefore seeks to comparatively evaluate the performance of Logistic+PSO, Sinusoidal+PSO, and Circle+PSO for image segmentation, in order to identify the chaotic map that provides the best segmentation quality, convergence behavior, and optimization performance.

The aim of this research is therefore to evaluate the performance of three selected chaos maps, the logistic, sinusoidal, and circle maps, in improving Particle Swarm Optimization for image segmentation tasks. To achieve this aim, the study sets out to integrate the selected chaos maps into the PSO algorithm to improve its exploration and exploitation balance, to implement the resulting chaos-enhanced PSO variants, namely Logistic+PSO, Sinusoidal+PSO, and Circle+PSO, using the Python programming language, and to evaluate and compare the performance of the three variants using the Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity Index (SSIM), the Feature Similarity Index (FSIM), convergence speed, and computational efficiency. In line with this aim and these objectives, the study seeks to answer two research questions: which of the selected chaotic maps, Logistic+PSO, Sinusoidal+PSO, or Circle+PSO, produces the best segmentation performance on the Apple Image Dataset, and how the three CE-PSO variants compare with one another across the five evaluation metrics of PSNR, SSIM, FSIM, convergence speed, and computational efficiency. By isolating the contribution of three widely used chaotic dynamics inside a common PSO backbone, this work clarifies how chaos influences swarm diversity, parameter adaptation, and convergence trajectories in segmentation objectives, and provides practically relevant guidance for algorithm selection in agricultural image analysis and other computer-vision optimization tasks.

2.0 RELATED WORKS

Early segmentation approaches, such as Otsu's thresholding method (Otsu, 1979) and Kapur's entropy-based thresholding technique (Kapur et al., 1985), laid the foundation for automatic image segmentation; these methods are computationally efficient for simple images with distinct intensity distributions, but their performance deteriorates considerably in images affected by noise, uneven illumination, or overlapping intensity values. The introduction of Particle Swarm Optimization by Kennedy and Eberhart (1995) provided an alternative by formulating image segmentation as an optimization problem, and Horng (2011) applied PSO to multilevel thresholding using Otsu's criterion, reporting improved segmentation accuracy with reduced computational cost compared with exhaustive search techniques, while Zhao et al. (2008) employed PSO for entropy-based image segmentation and demonstrated better segmentation quality than conventional thresholding methods. To address the premature convergence reported in these studies, Shi and Eberhart (1998) introduced the inertia-weight strategy and Clerc and Kennedy (2002) developed the constriction-factor approach; more recently, Mohamed (2022) integrated PSO with K-means clustering to improve centroid initialization, Suganthi and Ramakrishnan (2021) combined PSO with Fuzzy C-Means clustering for greater robustness in noisy images, Ma et al. (2024) proposed a complementary inertia-weight PSO for medical image segmentation, Hu et al. (2024) enhanced PSO using a Cauchy mutation strategy on urban image datasets, Qiao et al. (2024) hybridized the Arithmetic Optimization Algorithm with the Harris Hawks Optimizer for multilevel thresholding, and Song et al. (2024) applied a modified Snake Optimizer to multilevel color-image thresholding for agricultural disease detection, demonstrating the growing relevance of swarm-based segmentation to fruit and crop imaging tasks similar to the Apple Image Dataset used in this study. Although these modifications improved optimization performance, they did not eliminate the tendency of PSO to become trapped in local optima.

Chaos-enhanced PSO has since emerged as a promising response to this limitation. Alatas et al. (2009) were among the earliest researchers to demonstrate that replacing conventional random-number generation with chaotic sequences significantly enhanced convergence speed and reduced premature convergence, while Gao et al. (2010) showed that the Logistic map improved the exploration capability of PSO by producing more uniformly distributed search trajectories. More recently, Mostafa et al. (2024) and Hosny et al. (2023) reported that chaos-enhanced swarm algorithms achieved superior segmentation performance on medical and satellite images by improving population diversity during optimization, and Mahmoud et al. (2025) demonstrated that a hybridized, Lévy-enhanced metaheuristic improved both segmentation accuracy and convergence speed in multilevel thresholding applications, reinforcing the value of diversity-preserving mechanisms such as chaos. Although numerous studies have therefore demonstrated the effectiveness of chaos-enhanced PSO, comparatively fewer have investigated the relative performance of different chaotic maps under identical conditions: Cisternas-Caneo et al. (2024) compared several chaotic maps within continuous metaheuristic algorithms and concluded that the effectiveness of chaos-enhanced optimization depends strongly on the characteristics of the selected chaotic map rather than on the optimization algorithm alone, underscoring the need for further comparative studies focused on specific application domains such as image segmentation.

Taken together, the literature shows that significant progress has been made in image segmentation through deep learning, PSO, and chaos-enhanced optimization, yet several important gaps remain. Deep-learning-based methods achieve high accuracy but require large labelled datasets and substantial computational resources, limiting their applicability to unsupervised optimization problems, while most improvements to PSO itself rely on adaptive parameter adjustment, mutation operators, or hybrid strategies, leaving the underlying problem of premature convergence and loss of population diversity inadequately addressed. Existing studies on chaos-enhanced PSO also primarily investigate a single chaotic map at a time, making it difficult to determine which chaotic mechanism provides the most effective balance between exploration, exploitation, convergence speed, and segmentation accuracy; this pattern persists even in the most recent hybrid metaheuristics for thresholding-based segmentation (Qiao et al., 2024; Mahmoud et al., 2025), which incorporate a single chaotic or stochastic enhancement mechanism rather than systematically comparing multiple chaotic maps. In particular, comparative studies involving the Logistic, Sinusoidal, and Circle chaotic maps within the same PSO framework, under identical experimental conditions and using a fruit-image dataset, remain scarce, so there is no clear consensus on the most suitable chaotic map for improving PSO performance in image segmentation.

The findings of Olatunji *et al*. (2025) align with broader research trends in chaos-enhanced optimization algorithms for image and handwriting recognition tasks. Also, Okunlola *et al* (2026) integrated chaotic maps into the Particle Swarm Optimisation (PSO) algorithm to enhance population diversity, prevent premature convergence and improve segmentation quality and Moronkeji *et al*. (2026) evaluated three chaos-enhanced PSO (CE-PSO) variants, Logistic+PSO, Sinusoidal+PSO and Gaussian+PSO, for image segmentation using multilevel thresholding on the Berkeley Segmentation Dataset (BSDS500). Amusan *et al*. (2026) evaluated three chaos-enhanced PSO (CE-PSO) variants, namely Tent+PSO, Sinusoidal+PSO, and Circle+PSO, for image segmentation through multilevel thresholding on the Berkeley Segmentation Dataset (BSDS500). Furthermore, Olabiyisi *et al*. (2026) evaluated the performance of three chaos-enhanced CSO variants (Chebyshev+CSO, Lorenz+CSO, and Henon+CSO) for HDI using the Institut für Informatik und Angewandte Mathematik (IAM) dataset.

The present study addresses this gap by developing and comparatively evaluating three chaos-enhanced PSO variants based on the Logistic, Sinusoidal, and Circle chaotic maps, using the Apple Image Dataset under identical parameter settings and a common set of established segmentation-quality metrics, namely PSNR, SSIM, FSIM, convergence speed, and computational efficiency, to identify the chaotic map that provides the best segmentation accuracy, convergence performance, and optimization efficiency.

3.0 METHODOLOGY

This section presents the methodology adopted for the comparative study of selected chaotic maps to improve Particle Swarm Optimization (PSO) in image segmentation tasks. It covers the design, implementation, and evaluation of three chaos-enhanced PSO variants, Logistic+PSO, Sinusoidal+PSO, and Circle+PSO, and details the research framework, dataset description, image-preprocessing pipeline, chaotic-map integration, algorithmic design, implementation procedures, and evaluation metrics adopted for multilevel-threshold image segmentation using the Apple Image Dataset.

3.1 Methodology Framework

The overall research methodology follows a systematic experimental design consisting of five sequential phases as shown in Fig 1: data acquisition, image preprocessing, formulation of the CE-PSO algorithms, implementation with Python, and performance evaluation. Each phase produces outputs that serve as inputs to the subsequent phase, ensuring that the three chaos-enhanced PSO variants are evaluated under identical, reproducible conditions. Presenting the methodology as an explicit pipeline makes the dependencies between stages clear and underlines that the comparison is conducted under controlled conditions from beginning to end. Each phase, from data acquisition and image preprocessing through formulation of the CE-PSO algorithm, implementation in Python, and performance evaluation, feeds defined outputs into the next, so that the same inputs and procedures apply to all three variants, and any differences reported later arise from the algorithms themselves rather than from inconsistencies in how they were applied

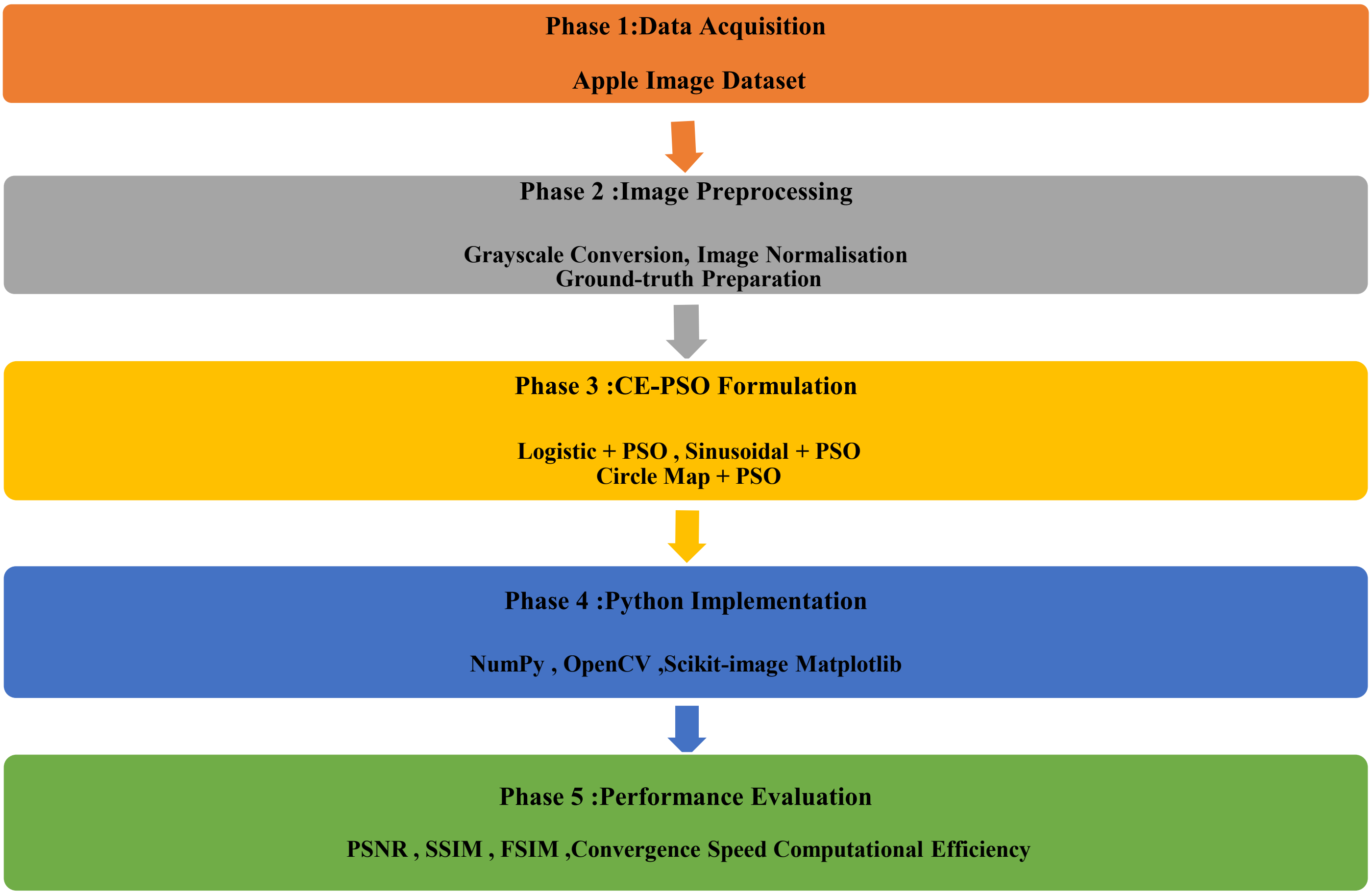

Figure 1: Methodology framework of the study

3.2 Data Acquisition

The first phase of the study involves acquiring the Apple Image Dataset, obtained from Kaggle, which serves as the benchmark for evaluating the proposed chaos-enhanced PSO algorithms. The dataset comprises apple images captured under varying imaging conditions, with different levels of segmentation complexity arising from variations in illumination, background composition, object orientation, and the presence of stems and leaves. The dataset was selected because it provides images with varying levels of segmentation difficulty, making it suitable for evaluating the robustness of multilevel-threshold segmentation algorithms; its basic characteristics are shown in Table 1.

Table 1: Apple Image dataset characteristics	
Attribute	**Value**
Source	Kaggle Apple Image Dataset
Total images	50 apple images
Image resolution	256 × 256 pixels
Color space	RGB (converted to grayscale for thresholding)
Ground truth	Manual segmentation masks (apple, stem/leaf, background)
Complexity levels	Low (30%), Medium (50%), High (20%)

3.3 Image Preprocessing

Image preprocessing is an essential step that prepares the input images for effective segmentation by enhancing image quality and ensuring consistency in data representation. In this study, preprocessing operations, namely grayscale conversion, intensity normalization, and ground-truth preparation, were performed to reduce computational complexity and improve the accuracy of the subsequent segmentation process.

Each color image is first converted from RGB to a single intensity channel using the luminosity method, which weights the color channels according to human brightness perception:

$$I\_gray = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (1)$$

where I_gray is the grayscale intensity value and R, G, and B are the red, green, and blue channel values respectively. The grayscale intensities are then normalized to the full [0, 255] range so that all images are handled on a common scale:

$$I\_norm = ((I\_gray - min) / (max - min)) \times 255 \quad (2)$$

where I_norm is the normalized intensity value and min and max are the minimum and maximum grayscale values in the image. This linear rescaling is the only intensity adjustment applied to the images, so that all algorithms operate on identically preprocessed inputs. The reference image used to compute the segmentation-quality metrics is this preprocessed grayscale image; during evaluation, the multilevel-thresholded reconstruction produced by each algorithm is compared against it so that PSNR, SSIM, and FSIM quantify how faithfully the segmented output preserves the information content and perceptual characteristics of the original image.

3.4 Formulation of the CE-PSO Algorithms

While standard PSO has demonstrated effectiveness in solving a wide range of optimization problems, its performance may deteriorate due to premature convergence and insufficient exploration of the search space. To address these limitations, chaotic maps can be incorporated into the PSO framework to improve population diversity and enhance the balance between exploration and exploitation. In this study, the Logistic, Sinusoidal, and Circle chaotic maps are integrated into the PSO algorithm to generate three chaos-enhanced variants for image segmentation, with the development, mathematical formulation, and implementation of each variant presented below.

3.4.1 Logistic+PSO Algorithm

The Logistic+PSO variant integrates the Logistic map (r = 4.0) into initialization, parameter adaptation, and a chaotic perturbation applied every ten iterations. The full procedure is given in Algorithm 1.

Algorithm 1: Logistic Map-Enhanced PSO (Logistic+PSO)

```
Input : N=30, T_max=100, D=thresholds, r=4.0,
    x0=0.1, [w_min, w_max] = [0.4,0.9],
    c1=c2=2.0, image I

Output: Gbest thresholds, best fitness, convergence curve, segmented image
1. Generate Logistic chaotic sequence:  x = r·x· (1 - x), N×D values
2. Initialize population with chaos:
   position[i][j] = LB + (UB - LB)·chaos_seq[idx]
   velocity[i][j] = (UB - LB)·(chaos_seq[idx] - 0.5)·0.5;  sort(position[i])
3. Evaluate fitness; set pbest = position, gbest = best
4. For t = 1 to T_max:
   chaos_w, chaos_c1, chaos_c2 = LogisticMap(1)
   w = w_max - (w_max - w_min)·(t/T_max)·(0.5 + 0.5·chaos_w)
   For each particle i:
      c1_adapt = c1·(0.5 + chaos_c1);  c2_adapt = c2·(0.5 + chaos_c2)
      r1, r2 = LogisticMap(1)
      velocity[i] = w·velocity[i]
        + c1_adapt·r1·(pbest[i]-position[i])
        + c2_adapt·r2·(gbest-position[i])
      position[i] = clamp(sort(position[i] + velocity[i]), LB, UB)
      Evaluate Kapur fitness; update pbest[i] and gbest if improved
   convergence[t] = gbest_fitness
   If t mod 10 == 0:  apply chaotic perturbation to selected particles
5. segmented = ApplyThresholds(I, gbest)
6. Return gbest, gbest_fitness, convergence, segmented
```

3.4.2 Sinusoidal+PSO Algorithm

The Sinusoidal+PSO algorithm integrates the sinusoidal chaotic map into the conventional PSO framework to improve exploration, maintain population diversity, and enhance convergence toward optimal threshold values. By utilizing sinusoidal chaotic sequences for particle initialization, parameter adaptation, and local search, the algorithm seeks to overcome premature convergence and achieve superior image-segmentation performance. The pseudocode of the proposed Sinusoidal+PSO algorithm is presented in Algorithm 2.

Algorithm 2: Sinusoidal Map-Enhanced PSO (Sinusoidal+PSO)

```
Input: N=30, T_max=100, D=thresholds, a=1.0,
       x0=0.1, [w_min, w_max] = [0.3,0.95], c1=c2=1.8, image I
Output: Gbest thresholds, best fitness, convergence curve, segmented image

1. Generate Sinusoidal chaotic sequence: x = a·sin(π·x), N×D values
2. Initialize population with chaos:
   position[i][j] = LB + (UB - LB)·chaos_seq[idx]^1.2
   velocity[i][j] = (UB - LB) ·(chaos_seq[idx] - 0.5) ·0.35; sort(position[i])
3. Evaluate fitness; set pbest = position, gbest = best
4. Compute constriction factor: φ = c1+c2; χ = 2/|2-φ-√(φ²-4φ) |
5. For t = 1 to T_max:
     a_dynamic = 0.8 + 0.4· (1 - t/T_max)
     sin_chaos = Sinusoidal Map(1, a_dynamic)
     w = w_max - (w_max - w_min) ·(t/T_max) · (0.5 + 0.3·sin_chaos)
     For each particle i:
     sin_c1 = 0.5 + 1.5·SinusoidalMap (1, a_dynamic)
     sin_c2 = 0.5 + 1.5·SinusoidalMap (1, a_dynamic)
     sin_r1, sin_r2 = Sinusoidal Map (1, a_dynamic)
     velocity[i] = χ·(velocity[i]
     + sin_c1·sin_r1·(pbest[i]-position[i])
     + sin_c2·sin_r2·(gbest-position[i]))
     position[i] = clamp(sort(position[i] + velocity[i]), LB, UB)
     Evaluate Kapur fitness; update pbest[i] and gbest if improved
     convergence[t] = gbest_fitness
     If t mod 5 == 0:  greedy chaotic local search around each particle;
     accept candidate only if fitness improves pbest/gbest
6.  segmented = Apply Thresholds(I, gbest)
7.  Return gbest, gbest_fitness, convergence, segmented
```

3.4.3 Circle+PSO Algorithm

The Circle+PSO algorithm integrates the Gaussian chaotic map into the conventional PSO framework to enhance exploration, maintain population diversity, and prevent premature convergence. By exploiting the strong mixing characteristics of Gaussian chaos, the algorithm improves the search for optimal threshold values and enhances image-segmentation performance. The pseudocode of the proposed Circle algorithm is presented in Algorithm 3.

Algorithm 3: Circle Map-Enhanced PSO (Circle+PSO)
Input: N=30, T_max=100, D=thresholds, x0=0.1 (≠0),   **[w_min, w_max] = [0.35,0.92], c1=c2=1.7, image I**   **Output: Gbest thresholds, best fitness, convergence curve, segmented image**   **1. Generate Circle sequence: x_(n+1) = (x_n + b − (a/2π) sin(2πx_n)) mod 1**   **2. Initialize population with chaos:**   **position[i][j] = LB + (UB - LB)·chaos_seq[idx]^0.8**   **velocity[i][j] = (UB - LB) ·(chaos_seq[idx] - 0.5) ·0.3; sort(position[i])**   **3. Evaluate fitness; set pbest = position, gbest = best**   **4. Compute constriction factor: φ = c1+c2; χ = 2/|2-φ-√(φ²-4φ) |**   **5. For t = 1 to T_max: circle_chaos = CircleMap(1); w = w_max − (w_max−w_min)·(t/T_max)·(0.55+0.3·circle_chaos); for each particle i: circle_c1 = 0.7+1.3×CircleMap(1), circle_c2 = 0.7+1.3×CircleMap(1); r1, r2 = CircleMap(1); velocity[i] = χ·(w·velocity[i] + circle_c1·r1·(pbest[i]−position[i]) + circle_c2·r2·(gbest−position[i])); position[i] = clamp(sort(position[i]+velocity[i]), LB, UB); evaluate Kapur fitness; update pbest[i] and gbest if improved; convergence[t] = gbest_fitness; if t mod 4 == 0: apply Circle chaotic local search; update pbest and gbest if improved**   **6. segmented = ApplyThresholds(I, gbest)**   **7. Return gbest, gbest_fitness, convergence, segmented**

3.5 Implementation with Python

The proposed chaos-enhanced PSO (CE-PSO) algorithms were implemented using the Python programming language due to its flexibility, extensive scientific-computing libraries, and suitability for image-processing and optimization tasks. The implementation utilized NumPy for numerical computations and array operations, OpenCV and scikit-image for image processing and evaluation metrics, phasepack for computing the phase congruency component of FSIM, and Matplotlib for plotting convergence curves and segmentation results. All three variants were developed within a common PSO framework and executed under identical experimental conditions, ensuring that the selected chaotic map remained the only distinguishing factor among the algorithms.

Establishing a consistent experimental environment ensures fair comparison, reproducibility, and reliability of the obtained results. The hardware and software environment used in this study is presented in Table 2.

Table 2: Experimental platform specifications

Component	**Specification**
Processor	Intel Core i7-10750H @ 2.60 GHz (6 cores, 12 threads)
RAM	16 GB
Storage	NVMe SSD
Operating system	Windows 11
Programming language	Python 3.9+
Key libraries	NumPy, OpenCV, scikit-image, phasepack, Matplotlib

3.6 Parameter Settings

The control parameters adopted for the experiments are presented in Table 3. The three CE-PSO variants share common values for the swarm size, maximum number of iterations, and threshold dimension; however, each algorithm uses its own chaotic-map parameter, inertia-weight bounds, and acceleration coefficients, corresponding to the characteristics of the Logistic, Sinusoidal, and Circle maps.

Table 3: CE-PSO control parameter settings		
Parameter	**Symbol**	**Value**
Population size	N	30
Maximum iterations	T_max	100
Dimension (threshold levels)	D	4
Independent runs per image	–	30
Fitness function	–	Kapur’s entropy
Dataset	–	Apple Image
Reference for PSNR / SSIM / FSIM	–	Original grayscale image

3.7 Evaluation Metrics

Evaluation metrics provide a quantitative basis for assessing the performance of image-segmentation algorithms. In this study, the reference image is the original preprocessed grayscale image, while the image under evaluation is the multilevel-thresholded reconstruction produced by the algorithm. PSNR, SSIM, and FSIM measure image-quality preservation, structural similarity, and feature consistency between the segmented image and the reference image, while convergence speed and computational efficiency evaluate the optimization performance of the algorithms. These metrics are given below.

$$PSNR = 10\, log_{10}\left(\frac{255^2}{MSE}\right) \quad (3)$$

where $MSE = (1/(M \times N))\, \Sigma\Sigma\, [I(i,j) - S(i,j)]^2$; $MAXI = 255$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x{}^2 + \mu_y{}^2 + C_1)(\sigma_x{}^2 + \sigma_y{}^2 + C_2)} \quad (4)$$

$$FSIM = \frac{\Sigma(PC_m(x) \times GM_m(x))}{\Sigma\, PC_m(x)} \quad (5)$$

$$Convergence\ Speed = \frac{T_{converge}}{T_{max}} \quad (6)$$

$$Computational\ Efficiency = \mathrm{T}_{\_execution} \quad (7)$$

where M and N are the number of rows and columns of the image, I(i, j) and S(i,j) are the reference and segmented pixel values, μ and σ denote means, variances and covariance of the compared images, PCm and GMm are the phase congruency and gradient magnitude used by FSIM, T_converge is the iteration at which fitness stabilises within 1% of the final value and T_execution is the total execution time of the algorithm, from initialization to completion, in seconds. All three CE-PSO variants were evaluated under identical conditions, on the same Apple image dataset images, using the same population size, iteration limit, and hardware and software, with results averaged over 30 independent runs so that any observed differences in performance can be attributed to the chaotic map employed rather than to inconsistencies in experimental treatment.

4.0 RESULTS AND DISCUSSION

This section presents the experimental results obtained from evaluating the three chaos-enhanced PSO (CE-PSO) algorithms, Logistic+PSO, Sinusoidal+PSO, and Circle+PSO, for multilevel-threshold image segmentation using the Apple Image Dataset. All algorithms were evaluated under identical conditions, using the same population size, iteration limit, dimension, dataset, and reference scheme, so that the comparison reported below is controlled and reproducible; the findings are reported against the five evaluation metrics defined in the methodology, namely PSNR, SSIM, FSIM, convergence speed, and computational efficiency, averaged over 30 independent runs per algorithm.

This section presents the experimental results obtained from evaluating the three chaos-enhanced PSO (CE-PSO) algorithms, Logistic+PSO, Sinusoidal+PSO, and Circle+PSO, for multilevel-threshold image segmentation using the Apple Image Dataset. All algorithms were evaluated under identical conditions, using the same population size, iteration limit, dimension, dataset, and reference scheme, so that the comparison reported below is controlled and reproducible; the findings are reported against the five evaluation metrics defined in the methodology, namely PSNR, SSIM, FSIM, convergence speed, and computational efficiency, averaged over 30 independent runs per algorithm.

Table 4: Performance metrics comparison

Algorithm	**PSNR (dB)**	**SSIM**	**FSIM**	**Computational Efficiency**
Logistic+PSO	26.78 ± 1.12	0.8912 ± 0.021	0.8745 ± 0.025	2.45 ± 0.23
Sinusoidal+PSO	27.89 ± 0.98	0.9123 ± 0.018	0.8967±0.021.	2.56 ± 0.22
Circle+PSO	27.12 ± 1.05	0.9034 ± 0.019	0.8856 ±0.023	2.38 ± 0.21

The three CE-PSO variants also differ in how quickly they reach their solutions. Sinusoidal+PSO exhibited the fastest convergence, reaching a stable solution after an average of 41 iterations, compared with 46 iterations for Circle+PSO and 52 iterations for Logistic+PSO, an 11% improvement over Circle+PSO and a 21% improvement over Logistic+PSO. In terms of computational efficiency, Circle+PSO recorded the shortest average execution time of 2.38 ± 0.21 s, followed by Logistic+PSO (2.45 ± 0.23 s) and Sinusoidal+PSO (2.56 ± 0.22 s), indicating that although Sinusoidal+PSO achieved the best segmentation quality and fastest convergence, Circle+PSO offers the most computationally efficient implementation among the three variants.

4.1 Discussion of Findings

The superior performance of Sinusoidal+PSO can be attributed to several factors. First, the sinusoidal map's smooth nonlinear oscillatory behavior generates well-distributed chaotic sequences that promote effective exploration of the search space while maintaining sufficient exploitation around promising solutions, enabling the swarm to avoid premature convergence and locate higher-quality threshold values than the Logistic and Circle maps. Second, the dynamic adjustment of the sinusoidal parameter maintains population diversity during the early stages of optimization while gradually encouraging convergence as the search progresses, improving both segmentation quality and convergence speed. Third, the incorporated constriction factor in the velocity-update equation stabilizes particle movement and enhances convergence towards the global optimum, while the periodic sinusoidal chaotic local search further refines the best solution without significantly increasing computational cost. These mechanisms collectively enable Sinusoidal+PSO to achieve the highest segmentation quality while converging more rapidly than the other variants.

When the three selected chaotic maps are compared on the basis of their influence on optimization performance, convergence behavior, and solution quality, chaotic maps with better diversity and exploration capability are found to improve PSO performance more consistently, with the sinusoidal map achieving superior convergence, stability, and optimization accuracy compared with the other two maps. In answer to the research questions posed at the outset of this study, Sinusoidal+PSO provides the best segmentation quality among the three chaotic maps, achieving the highest PSNR, SSIM, and FSIM values while also converging fastest; across the five evaluation metrics, Circle+PSO attained the highest computational efficiency, while Logistic+PSO consistently produced the lowest segmentation quality and slowest convergence. Sinusoidal+PSO is therefore recommended for high-accuracy applications such as agricultural image analysis and automated fruit inspection, Circle+PSO for time-critical applications where computational efficiency is prioritized, and Logistic+PSO as a computationally simple reference implementation for baseline comparison.

5.0 CONCLUSION

This study evaluated and compared three chaos-enhanced Particle Swarm Optimization (CE-PSO) variants, Logistic+PSO, Sinusoidal+PSO, and Circle+PSO, for image segmentation by multilevel thresholding on the Apple Image Dataset, addressing the limitations of conventional PSO, particularly premature convergence and inadequate exploration of the search space, by integrating chaotic sequences into population initialization, inertia-weight adaptation, acceleration-coefficient adjustment, and particle perturbation. The results demonstrate that integrating chaotic maps into PSO improves both segmentation quality and optimization performance: Sinusoidal+PSO achieved the best overall performance, with a PSNR of 27.89 ± 0.98 dB, SSIM of 0.9123 ± 0.018, and FSIM of 0.8967 ± 0.021, followed by Circle+PSO (27.12 ± 1.05 dB, 0.9034 ± 0.019, 0.8856 ± 0.023) and Logistic+PSO (26.78 ± 1.12 dB, 0.8912 ± 0.021, 0.8745 ± 0.025). Sinusoidal+PSO also converged fastest, reaching its solution in an average of 41 iterations, while Circle+PSO achieved the highest computational efficiency, with the lowest average execution time of 2.38 ± 0.21 s.

On the strength of these results, the study recommends the adoption of Sinusoidal+PSO for image-segmentation applications where maximum accuracy and improved optimization performance are required, given its superior segmentation quality, convergence behavior, and solution stability; Circle+PSO is recommended for time-critical applications in which computational efficiency is a priority, given its shortest average execution time; and Logistic+PSO is recommended as a suitable reference method for baseline comparison, given its simplicity of implementation. This work provides a systematic comparison of three chaotic maps, the Logistic map with its parabolic distribution, the Sinusoidal map with its smooth oscillatory behavior, and the Circle map with its rotational dynamics, within the same PSO optimization framework for image segmentation, establishing the superiority of the sinusoidal map for this application and demonstrating that the chaos-enhanced PSO framework provides a robust and reliable approach to multilevel-threshold image-segmentation tasks. Future work may extend this framework to additional chaotic maps, deep-learning integration, multi-objective optimization, and color or three-dimensional image segmentation.

References

1. Alatas, B., Akin, E., & Ozer, M. (2009). Chaos-based particle swarm optimization algorithms. *Applied Mathematics and Computation*, 214(1), 76–89.
2. Amusan, O. F., Olabiyisi, S. O., Ismaila, W. O., & Adeyemo, I. A. (2026). Performance evaluation of particle swarm optimization algorithm with selected

chaos maps for image segmentation. *UKR Journal of Multidisciplinary Studies*, 2(7), 128-136.

3. Amiriebrahimabadi, M., Rouhi, Z., & Mansouri, N. (2024). A comprehensive survey of multi-level thresholding segmentation methods for image processing. *Archives of Computational Methods in Engineering*, 31(6), 3647–3697.

4. Cisternas-Caneo, F., Crawford, B., Soto, R., Giachetti, G., Paz, Á., & Peña Fritz, A. (2024). Chaotic binarization schemes for solving combinatorial optimization problems using continuous metaheuristics. *Mathematics*, 12(2), 262.

5. Clerc, M., & Kennedy, J. (2002). The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73.

6. Gao, F., Zhang, Y., & Luo, X. (2010). Chaos particle swarm optimization algorithm and its application in engineering optimization. *Expert Systems with Applications*, 37(12), 8608–8615.

7. Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (4th ed.). Pearson.

8. Horng, M. F. (2011). A hybrid particle swarm optimization for multilevel thresholding. *Expert Systems with Applications*, 38(5), 5865–5873.

9. Hosny, K. M., Khalid, A. M., Hamza, H. M., & Mirjalili, S. (2023). Multilevel thresholding satellite image segmentation using chaotic coronavirus optimisation algorithm with hybrid fitness function. *Neural Computing and Applications*, 35(1), 855–886.

10. Hu, X., et al. (2024). Dual-attention guided network architecture for promptable multi-source image segmentation. *IEEE Access*, 12, 10341–10355.

11. Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3), 273–285.

12. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN'95)* (pp. 1942–1948).

13. Ma, J., Wang, Y., An, X., Liu, C., Gao, Z., Dong, Q., ... & Wang, B. (2024). Segment anything in medical images. *Nature Communications*, 15, Article 701.

14. Mahmoud, E., Alkhalaf, S., Senjyu, T., et al. (2025). GAAOA-Lévy: a hybrid metaheuristic for optimized multilevel thresholding in image segmentation. *Scientific Reports*, 15, 27232.

15. Mohamed, A. W. (2022). Multi-threshold image segmentation for melanoma based on an enhanced ant colony optimization for continuous domains. *Biomedical Signal Processing and Control*, 78, Article 103985.

16. Moronkeji, I. O., Olabiyisi, S. O., & Adedeji, O. T. (2026). A comparative study of selected chaos maps to improve particle swarm optimisation in image segmentation tasks. *EIRA Journal of Multidisciplinary Research and Development*, 2(4), 17-26.

17. Mostafa, R. R., Houssein, E. H., Hussien, A. G., Singh, B., & Emam, M. M. (2024). An enhanced chameleon swarm algorithm for global optimization and multilevel thresholding medical image segmentation. *Neural Computing and Applications*, 36(15), 8775–8823.

18. Okunlola, S. O., Baale, A. A., & Olabiyisi, S. O. (2026). Performance Evaluation of Selected Chaos-Enhanced Particle Swarm Optimisation for Image Segmentation. *Engineering and Technology Journal*, 11(7), 10904–10911. https://doi.org/10.47191/etj/v11i07.13

19. Olabiyisi, S. O., Babarinde, T. A., & Oladimeji, I. W. (2026). Performance Evaluation of Selected Chaos-Enhanced Chicken Swarm Optimization for Handwritten Document Identification. *Engineering And Technology Journal*, 11(07), 10965–10974. https://doi.org/10.47191/etj/v11i07.20

20. Olatunji, Babatunde Lekan, Olabiyisi, Stephen Olatunde, Oyeleye, Christopher Akinwale and Omotade, Adedotun Lawrence (2025). "An Enhanced Chicken Swarm Optimization Algorithm Using Gaussian and Tent Chaotic Map Functions," *International Journal of Research and Innovation in Applied Science (IJRIAS)*, 10(7), 653-664.

21. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66.

22. Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, 26(9), 1277–1294.

23. Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. *Swarm Intelligence*, 1(1), 33–57.

24. Qiao, L., Liu, K., Xue, Y., Tang, W., & Salehnia, T. (2024). A multi-level thresholding image segmentation method using hybrid arithmetic optimization and Harris Hawks optimizer algorithms. *Expert Systems with Applications*, 241, 122316.

25. Sahoo, P. K., Soltani, S., Wong, A. K. C., & Chen, Y. C. (1988). A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41(2), 233–260.

26. Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1), 146–165.

27. Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)* (pp. 69–73).

28. Song, H., Wang, J., Bei, J., & Wang, M. (2024). Modified snake optimizer based multi-level thresholding for color image segmentation of agricultural diseases. *Expert Systems with Applications*, 255, 124624.

29. Suganthi, S., & Ramakrishnan, K. (2021). Enhanced segmentation of inflamed ROI to improve the efficacy of breast thermography. *Journal of Ambient Intelligence and Humanized Computing*, 12(5), 5185–5195.

30. Tarekegn Nigatu, D., Gemechu Dinka, T., & Luleseged Tilahun, S. (2024). Convergence analysis of particle swarm optimization algorithms for different constriction factors. *Frontiers in Applied Mathematics and Statistics*, 10, 1304268.

31. Zhao, J., Zheng, J., & Zhu, W. (2008). Image segmentation using particle swarm optimization based on entropy. *Pattern Recognition Letters*, 29(9), 1294–1302.