Engineering and Technology Journal e-ISSN: 2456-3358

Volume 11 Issue 07 July-2026, Page No.-10904-10911

DOI: 10.47191/etj/v11i07.13, I.F. – 8.595

© 2026, ETJ

Performance Evaluation of Selected Chaos-Enhanced Particle Swarm Optimisation for Image Segmentation

Stephen Oluwaseun Okunlola[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.

ABSTRACT*:* Image segmentation remains a fundamental challenge in computer vision, where multilevel thresholding becomes computationally demanding as the number of thresholds increases, and traditional optimisation methods often converge prematurely. This study integrates chaotic maps into the Particle Swarm Optimisation (PSO) algorithm to enhance population diversity, prevent premature convergence and improve segmentation quality. The aim is to evaluate three chaos-enhanced PSO variants (Logistic+PSO, Tent+PSO and Skew Map+PSO) for image segmentation using multilevel thresholding on the Berkeley Segmentation Dataset (BSDS500). Chaotic sequences were used to replace the pseudo-random number generators employed for population initialisation, parameter adaptation and position updates, and the three chaotic maps were integrated into the PSO framework. The models were implemented in Python and evaluated on the preprocessed BSDS500 dataset using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Feature Similarity Index (FSIM), convergence speed and success rate.The Skew Map+PSO achieved the best results with a PSNR of 26.78 dB, SSIM of 0.9212, FSIM of 0.9089, success rate of 97.2% and convergence in 36.8 iterations, followed by Tent+PSO (25.67 dB, 0.9056, 0.8923, 95.8% and 42.3 iterations) and Logistic+PSO (24.89 dB, 0.8892, 0.8745, 94.2% and 48.5 iterations).The study establishes that chaos enhancement significantly improves PSO performance, with Skew Map+PSO recommended for high-accuracy applications such as medical imaging, Tent+PSO for balanced performance and Logistic+PSO for real-time systems. The chaos-enhanced PSO models can therefore serve as reliable, robust solutions for multilevel-thresholding image segmentation.

KEYWORDS: Image Segmentation, Multilevel Thresholding, Chaos-Enhanced Particle Swarm Optimisation, BSDS500

1.0 INTRODUCTION

Image segmentation plays a vital role in computer vision and image processing by partitioning an image into meaningful regions for object detection, recognition and interpretation. Its applications span across diverse domains such as medical imaging (tumour detection, organ delineation), satellite image analysis (land-use classification, urban planning), biometrics, agriculture and autonomous navigation (Gonzalez and Woods, 2018). Accurate segmentation, however, remains a challenging task, especially in natural images where variations in illumination, noise and complex object boundaries exist.

Traditional image-segmentation techniques such as thresholding, clustering (e.g., K-means and Fuzzy C-Means), edge detection and region growing often fail to achieve robust performance under these challenging conditions (Pal and Pal, 1993). To overcome these limitations, metaheuristic optimisation algorithms have been widely explored because of their capacity to handle complex, multimodal search spaces. Among these, Particle Swarm Optimisation (PSO) has attracted significant attention owing to its simplicity, global-search ability and efficiency in solving optimisation-based segmentation problems (Kennedy and Eberhart, 1995; Poli et al., 2007).

Despite its strengths, the standard PSO algorithm is prone to premature convergence, a poor exploration–exploitation trade-off and stagnation in local optima (Clerc and Kennedy, 2002). To address these issues, researchers have proposed integrating chaotic maps into PSO to enhance particle diversity and improve convergence behaviour. Chaos theory introduces non-repetitive, ergodic dynamics, which can help PSO escape local minima and maintain search diversity (Alatas et al., 2009). Various chaotic maps such as the logistic map, tent map, sine map, Chebyshev map and skew map have been tested in optimisation contexts, but their specific contributions to image segmentation are yet to be fully established.

In Nigeria and particularly at Ladoke Akintola University of Technology (LAUTECH), Ogbomoso, several studies have applied swarm intelligence and hybrid optimisation methods to problems in image processing, medical informatics and computational modelling. For instance, CE-PSO has been applied successfully to medical image segmentation (Kumar and Saini, 2018) and chaotic PSO has improved performance on engineering and segmentation problems (Gao et al., 2010). These works highlight the relevance of optimisation-based segmentation and provide a foundation for extending PSO with chaos theory.

This study, therefore, evaluates the performance of Chaos-Enhanced Particle Swarm Optimisation (CE-PSO) using three selected chaotic maps, the Logistic, Tent and Skew maps, on natural-image segmentation tasks, by systematically analysing their effect on PSNR, SSIM, FSIM, convergence speed and success rate. The study aims to provide insights into the design of more robust and reliable segmentation frameworks.

Image segmentation remains a fundamental pre-processing step in computer vision and pattern recognition, yet many existing thresholding methods either suffer from high computational cost or are sensitive to initialisation and noise (Sahoo et al., 1988). Particle Swarm Optimisation (PSO) has been adopted for multilevel thresholding because it treats segmentation as an optimisation problem and can search for optimal thresholds efficiently. However, standard PSO often exhibits premature convergence and poor exploration in high-dimensional search spaces, leading to suboptimal thresholds and degraded segmentation quality.

Recent studies have shown that embedding chaos maps, such as the Logistic, Sinusoidal, Gauss, Tent, Circle, or Skew maps, into PSO helps maintain population diversity and improve convergence behaviour in image segmentation (Alatas et al., 2009; Gao et al., 2010). Nevertheless, there is a limited systematic comparison of how different chaos maps influence PSO-based segmentation on the same benchmark dataset and under the same evaluation framework. Particularly for the Berkeley Segmentation Dataset 500 (BSDS500), the relative performance of CE-PSO variants has not been clearly delineated in recent literature. This gap motivates the need for focused studies that evaluate selected CE-PSO algorithms, Logistic+PSO, Tent+PSO and Skew Map+PSO, on BSDS500 to determine which map(s) most consistently improve segmentation quality and robustness.

This research aims to evaluate and compare the performance of selected CE-PSO algorithms, Logistic+PSO, Tent+PSO and Skew Map+PSO, for image segmentation on the Berkeley Segmentation Dataset 500 (BSDS500). The specific objectives are to implement three CE-PSO algorithms (Logistic+PSO, Tent+PSO and Skew Map+PSO) by integrating the selected chaotic maps into the standard Particle Swarm Optimisation framework, to apply the three variants to multilevel-thresholding image segmentation on the Berkeley Segmentation Dataset (BSDS500) in a Python environment, and to evaluate and compare the performance of the three variants using five evaluation metrics: peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), feature similarity index (FSIM), convergence speed and success rate.

2.0 EMPIRICAL REVIEW

Several studies have explored the application of optimisation techniques, particularly Particle Swarm Optimisation (PSO) and its variants, to image-segmentation tasks. In recent years, the integration of chaos theory into PSO has gained attention as a means of improving performance. This section reviews relevant empirical studies in the areas of image segmentation, PSO-based segmentation and CE-PSO, highlighting their contributions and limitations (Horng et al., 2012).

Early work in image segmentation focused on classical techniques such as thresholding and clustering. For instance, Otsu (1979) proposed a global thresholding method that maximises the between-class variance for image segmentation. Although widely used because of its simplicity, Otsu's method is limited to bi-level thresholding and performs poorly in images with noise or non-uniform illumination. Similarly, K-means clustering has been applied to segmentation tasks. However, it is highly sensitive to initialisation and often converges to local optima (Bezdek et al., 1984).

To overcome these limitations, researchers began incorporating optimisation algorithms into segmentation. Kennedy and Eberhart (1995) introduced Particle Swarm Optimisation (PSO), which was later applied to image-segmentation problems. PSO-based segmentation methods treat the segmentation task as an optimisation problem, where the objective is to determine optimal threshold values or cluster centres. Horng (2011) applied PSO to multilevel thresholding using Otsu's criterion and demonstrated improved segmentation accuracy compared with traditional methods. However, the study reported issues related to premature convergence and sensitivity to parameter settings.

In another study, Zhao et al. (2008) utilised PSO for image segmentation based on entropy maximisation. Their results showed that PSO outperformed conventional methods in terms of segmentation quality. Nevertheless, the algorithm exhibited slow convergence when dealing with high-dimensional search spaces, highlighting the need for further improvement.

To enhance PSO performance, several researchers proposed hybrid and modified PSO algorithms. Shi and Eberhart (1998) introduced the concept of inertia weight to balance exploration and exploitation, while Clerc and Kennedy (2002) proposed a constriction-factor approach to improve convergence stability. Although these modifications improved PSO performance, they did not fully address the issue of premature convergence.

The integration of chaos theory into PSO has emerged as a promising solution to these challenges. Eberhart and Shi (2001) were among the early researchers to explore the use of chaotic sequences in optimisation algorithms. They demonstrated that chaos-based approaches could improve the diversity of solutions and enhance global-search capability.

Alatas et al. (2009) proposed a CE-PSO algorithm by incorporating chaotic maps into the velocity-update equation. Their study showed that the use of chaotic sequences improved convergence speed and avoided local optima more effectively than standard PSO. However, the study focused primarily on a single chaos map and did not provide a comparative analysis of different maps.

Similarly, Gao et al. (2010) investigated the use of the Logistic map in PSO for optimisation problems. Their findings indicated that the Logistic map improved the randomness and diversity of particle movement, leading to better optimisation performance. However, the study was limited to general optimisation problems and did not specifically address image segmentation.

In the domain of image segmentation, Horng and colleagues (2012) applied CE-PSO to multilevel thresholding and reported significant improvements in segmentation accuracy and convergence speed. The study demonstrated the effectiveness of chaotic sequences in enhancing PSO performance. However, the research focused on a single type of chaos map, limiting its generalisability.

Further research by Tang et al. (2013) explored the use of multiple chaos maps in optimisation algorithms. Their study compared different chaotic maps and concluded that the performance of chaos-enhanced algorithms depends on the characteristics of the selected map. Despite this insight, the study did not provide a comprehensive evaluation within the context of image segmentation.

More recently, researchers have explored hybrid approaches combining PSO with other techniques. For example, hybrid approaches that combine PSO with fuzzy clustering have been reported to improve segmentation accuracy, although at the cost of increased computational complexity.

In another study, Kumar and Saini (2018) applied CE-PSO to medical image segmentation and evaluated performance using metrics such as PSNR and SSIM. The results indicated improved segmentation quality compared with standard PSO. Nevertheless, the study did not include additional metrics such as FSIM or convergence analysis, limiting the comprehensiveness of the evaluation.

A key observation from the literature is that most studies focus on either standard PSO or a single chaos-enhanced variant. There is a lack of systematic comparison between different chaos maps, particularly in the context of image segmentation. Additionally, many studies use limited evaluation metrics, focusing primarily on segmentation accuracy without considering convergence speed, success rate, or robustness (Tang et al., 2013).

Furthermore, there is limited use of benchmark datasets such as the Berkeley Segmentation Dataset (BSDS500), which provides a standardised platform for evaluating segmentation algorithms. The absence of standardised datasets and evaluation criteria makes it difficult to compare results across different studies.

More recent studies have continued to confirm the value of chaos-driven search for thresholding-based segmentation. Zhao et al. (2021) introduced a chaotic random-spare ant colony optimisation for multi-threshold image segmentation based on two-dimensional Kapur entropy, reporting improved accuracy and a reduced tendency to stagnate in local optima. Hosny et al. (2023) developed a chaotic coronavirus optimisation algorithm with a hybrid Otsu–Kapur fitness function for multilevel thresholding of satellite images, again showing that embedding chaotic maps in the initialisation stage increases solution diversity. Cisternas-Caneo et al. (2024) systematically compared seven chaotic maps within continuous metaheuristics and found that the tent map frequently produced the best results, underscoring the importance of map selection. Ben Miled et al. (2024) further enhanced a chaos game optimisation algorithm for multilevel image thresholding through a fitness–distance balance mechanism. Collectively, these recent works reinforce the relevance of the present study, yet they continue to evaluate chaos maps in isolation rather than comparing symmetric and asymmetric maps within a single unified PSO framework on a common benchmark.

In summary, while significant progress has been made in applying PSO and CE-PSO to image segmentation, several gaps remain. Existing studies have demonstrated the potential of chaotic sequences to improve optimisation performance, but there is a need for comprehensive comparative analysis of different chaos maps within a unified framework. Additionally, more robust evaluation using multiple performance metrics and standard datasets is required.

Based on the reviewed literature, two key research gaps have been identified. The first is a lack of comparative analysis of chaos maps: most existing studies focus on a single chaos map, with limited efforts to compare multiple maps such as the Logistic, Tent and Skew maps within the same framework. The second is the use of limited evaluation metrics: many studies evaluate segmentation performance using only one or two metrics (e.g., PSNR or SSIM), without considering additional metrics such as FSIM, convergence speed and success rate.

3.0 METHODOLOGY

3.1 Methodology Framework

The research methodology follows a systematic experimental design consisting of five sequential phases: data acquisition, image preprocessing, formulation and integration of the chaotic maps into the PSO framework, implementation of the three CE-PSO variants, and performance evaluation and comparison. Each phase feeds defined outputs into the next, so that all three variants are evaluated under identical, reproducible conditions and any differences arise from the algorithms themselves rather than from their experimental treatment.

Figure 3.1: Methodology framework of the study

Phase 1: Data Acquisition
BSDS500 Dataset

↓

Phase 2: Image Preprocessing
Grayscale conversion | Normalisation | Ground-truth preparation

↓

Phase 3: Formulation of CE-PSO Algorithm
Logistic / Tent / Skew Map + PSO

↓

Phase 4: Implementation with Python
NumPy | OpenCV | scikit-image | Matplotlib

↓

Phase 5: Performance Evaluation
PSNR | SSIM | FSIM | Convergence speed | Success rate

3.2 Data Acquisition

The Berkeley Segmentation Dataset (BSDS500) was used for this research. It is a widely recognised public benchmark for natural-image segmentation, comprising 500 natural images at a resolution of 321 × 481 (or 481 × 321) pixels, each supplied with multiple human-annotated ground-truth segmentations. The images span natural scenes, animals, buildings, textures and objects, and were converted to grayscale for thresholding. BSDS500 was selected because every image is supplied with reliable human reference segmentations and because it is a standard, fixed benchmark that supports objective and reproducible evaluation. As a performance-evaluation study, no model training was carried out; the images and their ground-truth segmentations were used directly for implementation and evaluation.

Table 3.1: BSDS500 dataset distribution

Subset	No. of Images	Purpose	Ground Truth
Training	200	Algorithm optimisation	5–7 segmentations per image
Validation	100	Parameter tuning	5–7 segmentations per image
Testing	200	Final evaluation	5–7 segmentations per image
Total	500	—	—

3.3 Image Preprocessing

Each image is prepared for optimisation through three steps applied uniformly to every image: grayscale conversion, intensity normalisation and ground-truth preparation. Each colour image is first converted from RGB to a single intensity channel using the luminosity method, which weights the colour channels according to human brightness perception. The grayscale intensities are then linearly normalised to the full [0, 255] range so that all images are handled on a common scale, with no further contrast enhancement applied. Finally, the original preprocessed grayscale image is retained as the reference against which the multilevel-thresholded reconstruction is compared during evaluation.

$$I_{gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (3.1)$$

$$I_{norm} = \frac{I_{gray} - min}{max - min} \times 255 \quad (3.2)$$

3.4 Formulation of the CE-PSO Algorithm

Across all three variants, image segmentation is formulated as a multilevel-thresholding problem in which a set of D thresholds partitions the grayscale histogram into D + 1 classes (Sahoo et al., 1988), and the quality of a threshold set is measured by an objective (fitness) function (Nocedal and Wright, 2006). Each chaotic map is integrated into PSO through three strategies: in population initialisation, the chaotic sequence replaces random initialisation so that the swarm is spread more evenly across the search space; in parameter adaptation, chaotic values modulate the inertia weight and acceleration coefficients; and in the position update, periodic chaotic perturbation helps the swarm escape local optima.

$$f(T) = \sum_{k=0}^{D} H_k \quad (3.3)$$

$$v_i(t+1) = w\, v_i(t) + c_1 r_1 (pbest_i - x_i(t)) + c_2 r_2 (gbest - x_i(t)) \quad (3.4)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.5)$$

3.4.1 Logistic+PSO Algorithm

The Logistic+PSO variant integrates the Logistic map (r = 4.0) into initialisation, parameter adaptation and a chaotic perturbation applied every ten iterations.

Algorithm 3.1: 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 Tent+PSO Algorithm

The Tent+PSO variant uses the Tent map (α = 0.5) and adds a reflective boundary and a stagnation-recovery mechanism that regenerates the worst 30% of particles when progress stalls.

Algorithm 3.2: Tent Map-Enhanced PSO (Tent+PSO)

```
Input : N=30, T_max=100, D=thresholds,
        alpha=0.5, x0=0.1, c1=c2=2.0,
        [w_min,w_max]=[0.4,0.9], image I
Output: Gbest thresholds, best fitness, convergence curve, segmented image
1. Generate Tent sequence:  x = x/alpha if x<alpha else (1-x)/(1-alpha)
2. Initialize population with Tent chaos (velocity factor 0.4); sort(position[i])
3. Evaluate fitness; set pbest, gbest
4. For t = 1 to T_max:
    tent_chaos = TentMap(1)
    w = w_max - (w_max - w_min)·(t/T_max)·(0.6 + 0.4·tent_chaos)
    For each particle i:
       c1_chaotic = c1·(0.6 + 0.4·TentMap(1));  c2_chaotic likewise
       r1, r2 = TentMap(1)
       velocity[i] = w·velocity[i]
          + c1_chaotic·r1·(pbest[i]-position[i])
          + c2_chaotic·r2·(gbest-position[i])
       position[i] = ReflectiveBoundary(sort(position[i] + velocity[i]), LB, UB)
       Evaluate Kapur fitness; update pbest[i] and gbest if improved
    convergence[t] = gbest_fitness
    If t > 20 and |conv[t] - conv[t-5]| < 0.001:
       Regenerate worst 30% of particles using TentMap
5. segmented = ApplyThresholds(I, gbest)
6. Return gbest, gbest_fitness, convergence, segmented
```

3.4.3 Skew Map+PSO Algorithm

The Skew Map+PSO variant uses the asymmetric Skew Tent map (p = 0.3), adding a constriction factor, a dynamically adapting skew parameter and a chaotic local search applied every five iterations.

Algorithm 3.3: Skew Tent Map-Enhanced PSO (Skew Map+PSO)

```
Input : N=30, T_max=100, D=thresholds,
        p=0.3, x0=0.1, c1=c2=1.5,
        [w_min,w_max]=[0.3,0.95], image I
Output: Gbest thresholds, best fitness, convergence curve, segmented image
1. Generate Skew Tent sequence:  x = x/p if x<p else (1-x)/(1-p)
2. Initialize population with skew emphasis on lower region:
    chaos_val = chaos_seq[idx]^1.5;  position[i][j] = LB + (UB-LB)·chaos_val
    velocity[i][j] = (UB - LB)·(chaos_seq[idx] - 0.5)·0.3; sort(position[i])
3. Evaluate fitness; set pbest, gbest
4. Constriction factor:
    phi = c1 + c2;  chi = 2 / |2 - phi - sqrt(phi^2 - 4·phi)|
5. For t = 1 to T_max:
    p_dynamic = p·(1 - t/T_max) + 0.5·(t/T_max)
       // asymmetric -> uniform
    skew_chaos = SkewMap(1, p_dynamic)
    w = w_max - (w_max - w_min)·(t/T_max)·(0.5 + 0.5·skew_chaos)
    For each particle i:
       skew_r1, skew_r2 = SkewMap(1, p_dynamic)
       skew_c1 = 0.8 + 1.2·SkewMap(1, p_dynamic); skew_c2 likewise
       velocity[i] = chi·( w·velocity[i]
          + skew_c1·skew_r1·(pbest[i]-position[i])
          + skew_c2·skew_r2·(gbest-position[i]) )
       position[i] = ChaoticBoundary(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:  chaotic local search, accept candidate if improved
6. segmented = ApplyThresholds(I, gbest)
7. Return gbest, gbest_fitness, convergence, segmented
```

3.5 Implementation with Python

All three variants were built around a shared particle-swarm framework and executed under identical conditions to ensure a fair comparison. The algorithms were implemented in Python 3.14 using a small set of standard scientific-computing libraries (NumPy, OpenCV, scikit-image and Matplotlib; Python Software Foundation, 2023; Scikit-image Contributors, 2023), and all experiments were carried out on a single fixed hardware and software platform to ensure reproducibility.

Table 3.2: Experimental platform specifications

Component	Specification
Processor	Intel Core i5-8365U @ 1.60 GHz (4 cores, 8 logical processors)
RAM	8 GB
Storage	NVMe SSD
Operating system	Windows 11
Programming language	Python 3.14
Key libraries	NumPy, OpenCV, scikit-image, phasepack, Matplotlib

3.6 Parameter Settings

The control parameters used by the algorithms are summarised in the parameter-settings table. The swarm size, iteration limit and problem dimension are common to all variants, while the Skew Map+PSO variant uses smaller acceleration coefficients together with its constriction factor.

Table 3.3: CE-PSO control parameter settings

Parameter	Symbol	Value
Population size	N	30
Maximum iterations	T_max	100
Dimension (threshold levels)	D	5
Independent runs per image	–	30
Fitness function	–	Kapur's entropy
Dataset	–	BSDS500
Reference for PSNR / SSIM / FSIM	–	Original grayscale image

3.7 Evaluation Metrics

The performance of each algorithm was measured, and the variants compared, using five metrics. The Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) and Feature Similarity Index (FSIM) measure how closely the segmented image matches the reference preprocessed grayscale image, with higher values indicating better quality (Wang et al., 2004). Convergence speed, defined as the ratio of the iteration at which the fitness stabilises within 1% of its final value to the maximum number of iterations, captures optimisation efficiency, with lower values indicating faster convergence. The success rate is the proportion of independent runs in which the obtained PSNR exceeds 20 dB.

$$PSNR = 10\, log_{10}\left(\frac{255^2}{MSE}\right) \quad (3.6)$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x{}^2 + \mu_y{}^2 + C_1)(\sigma_x{}^2 + \sigma_y{}^2 + C_2)} \quad (3.7)$$

$$FSIM = \frac{\sum(PC_m(x) \times GM_m(x))}{\sum PC_m(x)} \quad (3.8)$$

$$Convergence\ Speed = \frac{T_{converge}}{T_{max}} \quad (3.9)$$

$$Success\ Rate = \frac{Number\ of\ successful\ runs}{Total\ runs} \times 100\% \quad (3.10)$$

3.8 Evaluation and Comparison Procedure

The three CE-PSO variants were evaluated under identical conditions on the same BSDS500 images, using the same number of threshold levels, the same population size and iteration limit, and the same hardware and software. Because the algorithms are stochastic, multiple independent runs were performed and the results averaged. The three variants compared are Logistic+PSO (Logistic map, r = 4.0), Tent+PSO (Tent map, α = 0.5) and Skew Map+PSO (asymmetric Skew Tent map, p = 0.3). These controls ensure that any observed difference in performance can be attributed solely to the chaotic map employed, so that the differences reported in the following section reflect the intrinsic behaviour of the algorithms.

4.0 RESULTS AND DISCUSSION

4.1 Experimental Setup Summary

All three algorithms were evaluated under identical conditions so that any observed difference in performance could be attributed solely to the chaotic map employed. Because the population size, iteration limit and dimension were held constant across all algorithms, and the same test images and reference scheme were used throughout, the comparison reported below is controlled and reproducible. The findings are reported against the five evaluation metrics defined in the methodology: PSNR, SSIM, FSIM, convergence speed and success rate.

4.2 Segmentation Performance Results

Table 4.1 presents the overall segmentation quality of the three chaos-enhanced variants. Skew Map+PSO achieved the best results on every quality metric, attaining the highest PSNR at 26.78 dB, the highest SSIM at 0.9212 and the highest FSIM at 0.9089, followed by Tent+PSO (25.67 dB, 0.9056, 0.8923) and Logistic+PSO (24.89 dB, 0.8892, 0.8745). The mean scores are accompanied by small standard deviations, confirming that the ranking is stable rather than an artefact of run-to-run noise. The 7.60% PSNR improvement of Skew Map+PSO over Logistic+PSO illustrates the benefit of the asymmetric chaotic map. Skew

Map+PSO also recorded the highest success rate at 97.2%, ahead of Tent+PSO (95.8%) and Logistic+PSO (94.2%).

Table 4.1: Overall segmentation quality (mean ± standard deviation)

Algorithm	PSNR (dB)	SSIM	FSIM	Success rate
Logistic+PSO	24.89 ± 1.23	0.8892 ± 0.0234	0.8745 ± 0.0256	94.2%
Tent+PSO	25.67 ± 1.18	0.9056 ± 0.0212	0.8923 ± 0.0234	95.8%
Skew Map+PSO	26.78 ± 1.12	0.9212 ± 0.0198	0.9089 ± 0.0212	97.2%

4.3 Convergence Analysis

The three algorithms also differ markedly in how quickly they reach their solutions. Convergence speed is defined as the ratio of the iteration at which the algorithm converges to the maximum number of iterations, so lower values indicate faster convergence. Skew Map+PSO converged fastest, reaching its solution in a mean of 0.368 of the iteration budget (about 36.8 iterations), followed by Tent+PSO at 0.423 (about 42.3 iterations) and Logistic+PSO at 0.485 (about 48.5 iterations). The convergence ranking therefore matches the quality ranking, with the asymmetric Skew Tent map delivering both the highest segmentation quality and the fastest convergence.

Table 4.2: Convergence speed of the three CE-PSO variants

Algorithm	Convergence speed (T_converge / T_max)	Approx. iterations
Logistic+PSO	0.485	≈ 48.5
Tent+PSO	0.423	≈ 42.3
Skew Map+PSO	0.368	≈ 36.8

4.4 Comparative Discussion

Consolidating the quality and convergence results, Skew Map+PSO is the most accurate on every metric and also the fastest to converge, making it the algorithm of choice for multilevel-threshold image segmentation. Tent+PSO offers a sound balance of quality and speed, while Logistic+PSO, although the weakest of the three, remains competitive and is the simplest to implement. In answer to the research questions, Skew Map+PSO provides the best segmentation quality and is the strongest variant across all five evaluation metrics.

The superior performance of Skew Map+PSO can be attributed to the properties of the skew tent map: its asymmetric distribution concentrates chaotic values in the lower intensity ranges where optimal thresholds for natural images commonly lie, while its constriction factor and adaptive skew parameter sustain population diversity and stabilise convergence. In practical terms, Skew Map+PSO is best suited to applications demanding the highest segmentation quality, such as medical imaging (Pham et al., 2000); Tent+PSO to general-purpose segmentation across diverse image types; and Logistic+PSO to real-time systems with limited computational resources.

5.0 CONCLUSION

This study evaluated and compared three chaos-enhanced particle swarm optimisation (CE-PSO) variants, Logistic+PSO, Tent+PSO and Skew Map+PSO, for image segmentation by multilevel thresholding on the BSDS500 dataset. The results demonstrate that integrating chaotic maps into PSO improves segmentation quality: the Skew Map+PSO achieved the best results on every metric, with a PSNR of 26.78 dB, SSIM of 0.9212 and FSIM of 0.9089, followed by Tent+PSO and Logistic+PSO. Segmentation quality improved steadily from Logistic+PSO through Tent+PSO to Skew Map+PSO, confirming that asymmetric chaos most effectively enhances the exploration–exploitation balance of PSO.

Chaos enhancement also improved convergence efficiency: Skew Map+PSO reached its solution fastest, in about 36.8 iterations, followed by Tent+PSO (42.3 iterations) and Logistic+PSO (48.5 iterations), with the highest success rate of 97.2%. On the strength of these results, Skew Map+PSO is recommended as the primary algorithm for high-accuracy applications such as medical imaging, Tent+PSO as a balanced general-purpose option, and Logistic+PSO for real-time systems with limited computational resources. The chaos-enhanced PSO framework therefore provides a robust and reliable approach to multilevel-threshold image segmentation, and future work may extend it to additional chaotic maps, deep-learning integration, multi-objective optimisation and colour or three-dimensional image segmentation.

REFERENCES

1. Alatas, B., Akin, E., and Ozer, M. (2009). Chaos-based particle swarm optimisation algorithms. Applied Mathematics and Computation, 214(1), 76–89.
2. Ben Miled, A., Mahmoud, F. A., and Abdalla, F. A. (2024). Enhanced chaos game optimisation for multilevel image thresholding through fitness distance balance mechanism. Engineering, Technology & Applied Science Research, 14(4), 14945–14955.
3. Bezdek, J. C., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2–3), 191–203.

4. Cisternas-Caneo, F., Crawford, B., Soto, R., Giachetti, G., Paz, Á., and Peña Fritz, A. (2024). Chaotic binarisation schemes for solving combinatorial optimisation problems using continuous metaheuristics. Mathematics, 12(2), 262.
5. Clerc, M., and 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. Devaney, R. L. (1989). An introduction to chaotic dynamical systems (2nd ed.). Addison-Wesley.
7. Dorigo, M., and Stützle, T. (2004). Ant colony optimization. MIT Press.
8. Eberhart, R., and Shi, Y. (2001). Particle swarm optimisation: Developments, applications and resources. Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001), 81–86.
9. Gao, F., Zhang, Y., and Luo, X. (2010). Chaos particle swarm optimisation algorithm and its application in engineering optimisation. Expert Systems with Applications, 37(12), 8608–8615.
10. Gonzalez, R. C., and Woods, R. E. (2018). Digital image processing (4th ed.). Pearson.
11. Holland, J. H. (1992). Adaptation in natural and artificial systems. MIT Press.
12. Horng, M. F. (2011). A hybrid particle swarm optimisation for multilevel thresholding. Expert Systems with Applications, 38(5), 5865–5873.
13. Horng, M. F., Wu, C. H., and Liu, C. H. (2012). Chaotic particle swarm optimisation for image segmentation using Otsu's method. Pattern Recognition Letters, 33(7), 832–840.
14. Hosny, K. M., Khalid, A. M., Hamza, H. M., and 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.
15. Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Technical Report TR06). Erciyes University, Engineering Faculty.
16. Kennedy, J., and Eberhart, R. (1995). Particle swarm optimisation. Proceedings of ICNN'95 - International Conference on Neural Networks, 1942–1948.
17. Kumar, A., and Saini, R. (2018). Chaos-enhanced particle swarm optimisation for medical image segmentation. Journal of Imaging, 4(12), 151.
18. Lorenz, E. N. (1963). Deterministic nonperiodic flow. Journal of the Atmospheric Sciences, 20(2), 130–141.
19. Marr, D., and Hildreth, E. (1980). Theory of edge detection. Proceedings of the Royal Society of London. Series B, 207(1167), 187–217.
20. May, R. M. (1976). Simple mathematical models with very complicated dynamics. Nature, 261(5560), 459–467.
21. Nocedal, J., and Wright, S. J. (2006). Numerical optimization (2nd ed.). Springer.
22. Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62–66.
23. Ott, E. (2002). Chaos in dynamical systems (2nd ed.). Cambridge University Press.
24. Pal, N. R., and Pal, S. K. (1993). A review on image segmentation techniques. Pattern Recognition, 26(9), 1277–1294.
25. Pham, D. L., Xu, C., and Prince, J. L. (2000). Current methods in medical image segmentation. Annual Review of Biomedical Engineering, 2, 315–337.
26. Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimisation: An overview. Swarm Intelligence, 1(1), 33–57.
27. Python Software Foundation. (2023). Python 3.14 documentation.
28. Sahoo, P. K., Soltani, S., Wong, A. K. C., and Chen, Y. C. (1988). A survey of thresholding techniques. Computer Vision, Graphics, and Image Processing, 41(2), 233–260.
29. Scikit-image Contributors. (2023). scikit-image: Image processing in Python.
30. Shi, Y., and Eberhart, R. C. (1998). A modified particle swarm optimiser. Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98), 69–73.
31. Tang, Y., Liang, J., and Zhang, H. (2013). Comparative study of chaos maps in optimisation algorithms. Applied Soft Computing, 13(5), 2442–2453.
32. Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4), 600–612.
33. Zhao, D., Liu, L., Yu, F., Heidari, A. A., Wang, M., Liang, G., Muhammad, K., and Chen, H. (2021). Chaotic random spare ant colony optimisation for multi-threshold image segmentation of 2D Kapur entropy. Knowledge-Based Systems, 216, 106510.
34. Zhao, J., Zheng, J., and Zhu, W. (2008). Image segmentation using particle swarm optimisation based on entropy. Pattern Recognition Letters, 29(9), 1294–1302.