Received: 2 June 2025 | Revised: 12 March 2026 | Accepted: 15 May 2026 | Published online: 2 June 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–22
DOI: 10.47852/bonviewAIA62026336

BON VIEW PUBLISHING

Fuzzy Galactic Swarm Optimization Coupled with Superpixel-Based Microscopic Image Segmentation

Debasish Biswas[1], Shouvik Chakraborty[2,*] and Chinmoy Ghorai[1]

[1] *Department of Electronics and Telecommunication Engineering, Jadavpur University, India*

[2] *Department of Computer Science and Technology, Women's Polytechnic, Chandernagore, India*

Abstract: Microscopic imaging is fundamental to modern medical diagnostics, offering detailed structural and morphological insights essential for studying cellular behavior and detecting diseases. However, automated analysis of such images remains challenging due to variability in cell shapes, overlapping structures, noise, and inconsistent staining. These factors complicate accurate interpretation and necessitate advanced computational techniques. A critical step in this process is precise segmentation of cellular structures, which significantly impacts downstream tasks like classification and diagnosis. Effective segmentation enhances visual clarity and improves the reliability of computer-assisted diagnostic systems by identifying well-defined regions of interest. To address these challenges, this paper proposes a novel segmentation framework, SUFGSO (Superpixel-based Fuzzy Galactic Swarm Optimization). The approach integrates type-II fuzzy logic to manage uncertainty and ambiguity, with galactic swarm optimization, a metaheuristic inspired by hierarchical swarm intelligence. Additionally, superpixel techniques are employed to group pixels into meaningful regions, reducing computational complexity and improving spatial coherence, especially in high-resolution images. The proposed method is evaluated using four established cluster validity indices to ensure a comprehensive assessment of segmentation performance. Experimental results demonstrate that SUFGSO achieves improved accuracy and efficiency, indicating its effectiveness as a practical tool for microscopic image analysis in medical applications.

Keywords: microscopic image segmentation, superpixel, galactic swarm optimization, unsupervised clustering, SUFGSO

1. Introduction

As the fundamental units of life, cells hold critical information that aids in understanding biological processes and disease mechanisms. Analyzing cells through microscopic imaging enables researchers and clinicians to identify the underlying causes of various medical conditions and to examine structural details at a granular level [1, 2]. There are multiple types of microscopy techniques available [3–5], such as light microscopy, fluorescence imaging, and atomic force microscopy, each offering distinct advantages for exploring cellular structures [6, 7]. With advancements in imaging technology, modern devices can now capture high-resolution images, further enriching diagnostic precision. However, interpreting these complex images manually can be time-intensive and mentally demanding. In clinical environments where time and accuracy are paramount, automated interpretation tools become essential for expediting and improving microscopic image evaluation. Though numerous algorithms have been introduced in existing research for this purpose, many lack generalizability across diverse image types. Some approaches are domain-specific and may only work effectively on particular datasets. Hence, ongoing research is crucial to enhance segmentation outcomes and ultimately support more accurate diagnostic conclusions [8, 9].

The hippocampus, a critical structure within the brain, is deeply involved in cognitive functions such as memory formation and learning. Damage to this region—caused by trauma, disease, or other neurological events—can lead to significant impairments, including learning difficulties and memory loss. As a result, the hippocampus has remained a focal point of neurological and medical research for decades. Investigating its condition at the cellular scale is crucial for detecting early-stage abnormalities, particularly because untreated cellular degradation in this area may lead to conditions like amnesia. Image segmentation plays a foundational role in such diagnostic processes. However, segmenting microscopic images presents considerable challenges due to their inherent complexity—such as indistinct boundaries, overlapping structures, and low contrast among regions [10]. To address these difficulties, researchers have increasingly turned to metaheuristic optimization techniques. These algorithms—such as genetic algorithms (GAs), ant colony optimization (ACO), cuckoo search, artificial bee colony optimization, and galactic swarm optimization (GSO)—have demonstrated robust performance across

***Corresponding author:** Shouvik Chakraborty, Department of Computer Science and Technology, Women's Polytechnic, Chandernagore, India. Email: shouvikchakraborty@wbscte.ac.in

various problem domains. Their success has motivated their application in the context of automated microscopic image analysis, offering the potential to enhance both accuracy and efficiency.

In recent years, deep learning approaches have become the leading methods for segmenting medical and microscopic images. Convolutional neural networks (CNNs), particularly architectures like U-Net and its variants—including U-Net++, Attention U-Net, and MultiResUNet—have shown excellent performance by capturing both fine-grained details and broader contextual patterns in images [11].

The original U-Net, along with its multiscale extensions, enhances segmentation by incorporating nested skip connections and improved feature fusion, allowing more precise delineation of structures at different scales. Models that combine CNNs with transformer architectures, such as UNETR, TransUNet, and other vision—transformer-based methods, further improve results by modeling long-range dependencies through self-attention mechanisms, often surpassing purely CNN-based approaches in challenging segmentation tasks [12].

More recently, large pre-trained foundation models adapted for microscopy, such as Segment Anything for Microscopy (μSAM) [3], have enabled interactive and automated segmentation across various imaging modalities without the need for retraining. These models leverage powerful backbones and can generate high-quality masks with minimal manual input.

Despite their strong performance, deep learning–based segmentation methods generally depend on large, annotated datasets, significant computational resources, and task-specific fine-tuning to generalize to new image types. In contrast, unsupervised segmentation approaches do not rely on labeled data, making them particularly useful when expert annotation is scarce or costly. The proposed Superpixel-based Fuzzy Galactic Swarm Optimization (SUFGSO) framework combines superpixel representation, type-II fuzzy clustering, and GSO to perform segmentation based on intrinsic image characteristics. This approach is well-suited to data-limited scenarios, accommodates uncertainty, and avoids the need for extensive model retraining or large labeled datasets.

Recent progress in microscopic image analysis has been strongly influenced by advances in deep learning and microfluidic technologies, enabling improved detection and quantification of cellular structures. For instance, Haq et al. [13] introduced a combined microfluidic and deep learning–based system for identifying and analyzing colon cancer–related ascitic cells from bright-field microscopy images. Their approach employs a custom-designed polydimethylsiloxane microfluidic device together with a neural network pipeline that integrates YOLOv8x for cell detection and CSRNet for density-aware cell counting, achieving reliable performance on both laboratory-generated and patient-derived samples. While such data-driven frameworks demonstrate strong potential for diagnostic applications, they typically rely on large volumes of annotated data and task-specific deep models.

By contrast, the proposed SUFGSO framework adopts a fully unsupervised segmentation paradigm that eliminates the need for labeled training data. This characteristic makes it especially suitable for microscopic imaging scenarios where expert annotation is expensive, time-consuming, or impractical. As such, SUFGSO serves as a complementary alternative to deep learning–based approaches, offering robust segmentation capabilities in data—scarce environments while maintaining interpretability and flexibility across varying imaging conditions.

This study introduces a novel approach to microscopic image segmentation that integrates superpixel-based representation with a fuzzy GSO strategy. In this method, the type-II fuzzy logic framework is combined with the GSO algorithm to enhance segmentation accuracy. Superpixels, which group similar pixels into coherent regions, are employed to manage spatial information more effectively and reduce computational complexity. The complete method, referred to as SUFGSO, is designed to handle high-resolution microscopic images with improved efficiency. Experimental evaluation using four established cluster validity indices confirms the practical effectiveness and robustness of the proposed framework.

1.1. Novel contributions of the proposed SUFGSO framework

While fuzzy clustering, GSO, and superpixel-based segmentation have each been independently investigated in prior studies, their straightforward combination is insufficient for addressing the inherent challenges of microscopic medical image segmentation, particularly those arising from uncertainty, noise, and complex cellular morphology. The proposed SUFGSO framework advances the state of the art through the following distinct contributions:

1) Algorithmic contribution: A new SUFGSO optimization scheme is introduced by integrating type-II fuzzy clustering within a hierarchical subswarm–superswarm GSO structure. In contrast to conventional fuzzy clustering techniques that depend on fixed or locally updated membership functions, the proposed method employs GSO as a global optimization mechanism to iteratively refine fuzzy memberships. This design enhances resilience to noise, intensity inhomogeneity, and ambiguity commonly observed in microscopic images.
2) Integration contribution: The framework presents a systematic integration of superpixel representation, type-II fuzzy uncertainty modeling, and GSO into a unified segmentation pipeline. Superpixels serve as region-level primitives that reduce computational complexity while preserving boundary characteristics, enabling the optimization process to operate on meaningful image structures rather than individual pixels. Such a coordinated integration strategy has not been explicitly addressed in earlier segmentation approaches.
3) Performance and robustness contribution: By jointly leveraging uncertainty handling through type-II fuzzy logic and the global search capability of GSO, the SUFGSO framework achieves improved segmentation accuracy and enhanced stability when compared with traditional fuzzy clustering methods, swarm-based techniques, and superpixel-only approaches. Experimental evaluations on challenging microscopic images consistently demonstrate superior performance across multiple quantitative metrics.

These contributions clearly differentiate the proposed SUFGSO framework from existing fuzzy, optimization-based, and superpixel-driven segmentation methods, establishing it as a novel and effective solution for microscopic medical image segmentation.

The remainder of this paper is structured as follows: Section 2 outlines the foundational principles of the GSO algorithm. Section 3 provides an overview of the fuzzy type-II clustering technique. Sections 4 and 5 detail the implementation of the SUFGSO methodology and present the segmentation outcomes. Section 6 discusses some limitations of the proposed work. Finally, Section 7 summarizes the conclusions drawn from this work.

2. A Brief Overview of the Galactic Swarm Optimization

The GSO algorithm is a relatively recent metaheuristic inspired by the large-scale dynamics observed among galaxies, stars, and superclusters. First introduced by Muthiah-Nakarajan and Noel [14] in 2016, GSO models how clusters of stars within galaxies and the distribution of galaxies themselves exhibit non-uniform arrangements in space. In this framework, galaxies are abstracted as point masses whose gravitational interactions guide the search for optimal solutions [15, 16]. The algorithm combines principles from particle swarm optimization (PSO) with hierarchical swarm structures, where promising solutions can attract nearby candidate solutions, forming what is called a "superswarm" comprising the best representatives from multiple subpopulations [17, 18]. The movement of both individual swarms and the overall superswarm can be influenced by other swarm-based strategies to enhance the balance between exploration and exploitation. In the GSO model, each swarm consists of multiple particles that encode candidate solutions in a high-dimensional space [19]. Mathematically, the framework allows the size of each subswarm and the total number of subswarms to be defined flexibly, as shown in Equation (1).

A swarm in the GSO algorithm is represented by a collection X containing d dimensional data that the members of the collection can be represented by $\left(m_q^p \in \Re^d\right)$. Let us assume that the size of each subswarm is S and the number of subswarms is N; then the GSO architecture can be mathematically expressed using Equation (1).

$$\begin{cases} \mathrm{X}_p \in \mathrm{X},\ p = 1, 2, 3, ..., N \\ m_q^p \in \mathrm{X}_p,\ j = 1, 2, 3, ..., S \\ \mathrm{X}_p \cap \mathrm{X}_q = \phi \quad if\ p \neq q \\ \cup_{p=1}^{N} \mathrm{X}_p = \mathrm{X} \end{cases} \tag{1}$$

The members of the set $^{\mathrm{X}}$ can be randomly initialized from the range $[m_{\min}, m_{\max}]^d$. Each particle within a subswarm is initialized randomly within the specified search space bounds. For every particle, a velocity vector and a record of its best-known position (personal best) are maintained. Within each subswarm, the standard PSO mechanism updates these particles by comparing their fitness and moving them toward the current best solutions.

The global best within a given subswarm is determined by identifying the top-performing particle, and its value is continuously refined based on the updated personal bests. Although each subswarm evolves independently, the best solutions from all subswarms collectively form a "galactic best," guiding the superswarm dynamics at a higher level. This setup enables the algorithm to simultaneously explore different regions of the solution space. The particle's velocity and position are recalculated iteratively using Equations (2) and (3), while the inertia weight, which balances exploration and exploitation, is computed as in Equation (4). The random factors used in these updates are uniformly sampled from a defined interval to ensure stochastic variation in the search process. The value of the initial weight σ can be computed using Equation (4), and the random numbers rnd_1 and rnd_2 are uniformly chosen from [−1, +1].

$$\vartheta_q^p = \sigma \cdot \vartheta^p + c_1 \cdot rnd_1 \cdot \left(\psi_q^p - m_q^p\right) + c_2 \cdot rnd_2 \cdot \left(\upsilon^p - m_q^p\right) \tag{2}$$

$$m_q^p = m_q^p + \vartheta_q^p \tag{3}$$

$$\sigma = 1 - \frac{itr}{itr_{\max} + 1}, \quad itr = 0, 1, 2, ..., itr_{\max} \tag{4}$$

The superswarm is constructed by collecting the global best solutions identified by each subswarm, as defined in Equation (5). The update rules for the velocity and position of particles in this higher-level swarm are given by Equations (6) and (7), respectively, with the inertia weight once again computed according to Equation (4). Random values used in these calculations are uniformly drawn to maintain diversity during the search. This process repeats iteratively, with each new iteration building on the best information from the previous one. By combining the strengths of local exploitation within subswarms and global exploration through the superswarm, GSO effectively integrates detailed local knowledge into a broader search strategy. This dual-level mechanism is particularly advantageous for complex problems, such as microscopic image segmentation, where multiple local optima can exist within high-dimensional data. The random numbers rnd_3 and rnd_4 are uniformly chosen from [−1,+1].

$$\tau^k \in \Upsilon,\ k = 1, 2, 3, ..., N\ and\ \tau^k = \upsilon^k \tag{5}$$

$$\vartheta^p = \sigma' \cdot \vartheta^p + c_3 \cdot rnd_3 \cdot (\psi^p - \tau^p) + c_4 \cdot rnd_4 \cdot (\upsilon - \tau^p) \tag{6}$$

$$\tau^p = \tau^p + \vartheta^p \tag{7}$$

At every iteration, the same procedure repeats itself and starts from the point where the previous iteration left off. This approach performs both exploitation and exploration by using the concept of super swarm and the subswarm, and therefore, this approach can effectively incorporate the important local information in the overall optimization process, and this feature is beneficial for the microscopic image analysis [20]. This method also performs well in the case of multimodal microscopic image data due to its searching capability in the local minima.

3. Type-II Fuzzy System-Based C-Means Clustering

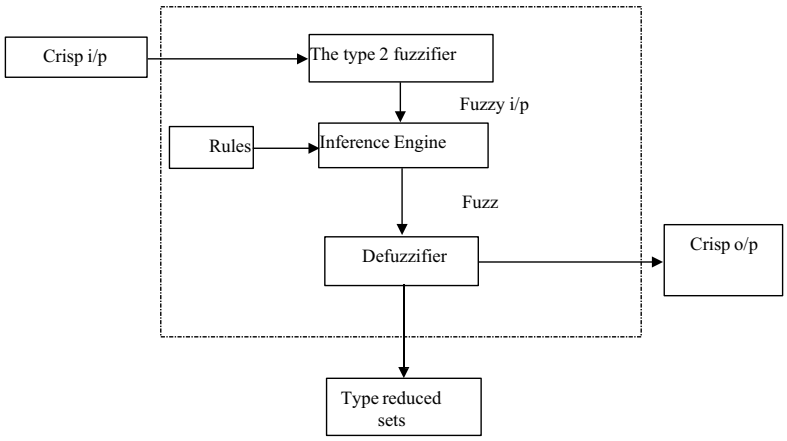
Fuzzy clustering methods enable each data point to simultaneously belong to multiple clusters, with the membership degree quantified by a value between 0 and 1 [21]. The sum of these degrees for any given point must equal 1 [22–24], ensuring that the point's total association is fully distributed across all clusters. The clustering process uses an objective function—defined in Equations (8) and (10)—to iteratively refine cluster centers for optimal partitioning.

$$O_\lambda = \sum_{k=1}^{pCnt} \sum_{l=1}^{cCnt} \mu_{kl}^{\lambda} \left\| x_k - c_l \right\|^2,\ where\ 1 \leq \lambda < \infty \tag{8}$$

The degree of membership for the point x_k to the l^{th} cluster is represented using μ_{kl}^{λ} and can be computed using Equation 9, and λ represents the fuzzifier. The degree of membership can be calculated using Equation 3. As already noted, the total sum of the degree of membership must be 1 [25] for a certain point, that is, $\sum_{l=1}^{cCnt} \mu_{kl} = 1\ for\ k = 1, 2, 3, ..., pCnt$, where $pCnt$ and $cCnt$ represent the number of data points and the count of the number of clusters, respectively.

$$\mu_{kl} = \frac{1}{\sum_{m=1}^{cCnt} \left(\frac{\left\| x_k - c_l \right\|}{\left\| x_k - c_m \right\|} \right)^{\frac{2}{\lambda - 1}}} \tag{9}$$

Figure 1
Function block diagram of the type-II fuzzy system



$$c_l = \frac{\sum_{k=1}^{pCnt} \mu_{kl}^{\lambda} x_k}{\sum_{k=1}^{pCnt} \mu_{kl}^{\phi}} \tag{10}$$

Type-II fuzzy systems extend this concept by better addressing uncertainty within the data [26], offering enhanced modeling capabilities compared to traditional type-I systems [27], [28]. These advanced fuzzy systems allow for additional layers of uncertainty representation, making them suitable for applications where ambiguity is high.

Equations 11 and 12 provide the basis for computing membership values and updating cluster centers within the type-II framework. However, in the SUFGSO approach, the GSO algorithm directly handles the cluster center updates, making Equation 12 unnecessary. Figure 1 illustrates the functional components of a type-II fuzzy system as used in this context [29].

$$\eta_{kl} = \mu_{kl} - \frac{1 - \mu_{kl}}{2} \tag{11}$$

$$\widehat{c}_l = \frac{\sum_{k=1}^{pCnt} \sigma_{kl}^{\lambda} x_k}{\sum_{k=1}^{pCnt} \eta_{kl}^{\lambda}} \tag{12}$$

4. Proposed SUFGSO Method

Advances in microscopic imaging technology continually improve the resolution and quality of acquired images, supporting more accurate diagnostic workflows. However, this technological progress also leads to the generation of extremely large and detailed images, which in turn pose significant computational challenges for automated computer-aided diagnostic systems [30]. Processing such images pixel by pixel demands substantial computational resources and can quickly become impractical without optimization.

To address this challenge, the use of superpixels has proven highly effective for managing spatial information [31]. Superpixels aggregate similar pixels into coherent regions, enabling algorithms to handle groups of pixels rather than individual ones—dramatically reducing computational overhead [32]. Consequently, superpixel segmentation is commonly applied as a preprocessing step in image analysis pipelines. Among the various superpixel generation techniques [33–35], irregular superpixels often yield better segmentation [36–38] performance than regular grid-based methods. SLIC (Simple Linear Iterative Clustering) is a widely used approach that produces regular, grid-like superpixels [39, 40], whereas the Mean Shift and Watershed algorithms generate irregular segments [41]. Although Watershed-based techniques are prone to over-segmentation due to their sensitivity to image noise, the Mean Shift method tends to handle such noise better, albeit at a higher computational cost.

The gradient images are computed [9] first to determine the local minima [42] so that noise can be minimized by considering only important gradient information. Morphological opening (O)- and closing (ϒ)-based reconstruction operations (Λ) are beneficial to process the noisy images, and Equations 13 and 14 can be used for this purpose. Here, IMG and IMG' are considered as the original and the marker images, and IMG' is defined in Equations 15 and 16, where *se* is the structuring element (SE).

$$\Lambda_{IMG}^{O}(IMG') = \Lambda^{\Psi}\left(\Lambda^{\Delta}\right) \tag{13}$$

$$\Lambda_{IMG}^{\Upsilon}(IMG') = \Lambda^{\Delta}\left(\Lambda^{\Psi}\right) \tag{14}$$

$$IMG' = \Psi_{se}(IMG) \tag{15}$$

$$IMG' = \Delta_{se}(IMG) \tag{16}$$

Here, Δ and Ψ denote the morphological erosion and dilation operations, respectively, and it is defined in Equations 17 and 18, respectively.

$$\Delta_{IMG}^{\varsigma}(IMG') = \Delta\left(\Delta^{\varsigma-1}(IMG)\right) \vee IMG' \tag{17}$$

$$\Psi_{IMG}^{\varsigma}(IMG') = \Psi\left(\Psi^{\varsigma-1}(IMG)\right) \wedge IMG' \tag{18}$$

Here, ∨ and ∧ represent the pointwise maximum and the minimum values, respectively. The morphological opening- and closing-based reconstruction approach helps avoid the problem of over-segmentation.

The structuring element *se* has an important impact on the final segmented image. The size of the SEs must be carefully

selected because inappropriate SEs can generate either under-segmented or over-segmented images. The size of the SE is dependent on the image, and therefore, determining the appropriate SE is another challenge. This problem can be solved by applying more than one SE and then considering the pointwise maximum values from the combined gradient image. The number of SEs, which are to be applied, can be customized based on the lower and upper bounds $[\varphi_{low},\ \varphi_{high}] \in \mathbb{N}^+$ of the controlling parameter φ for the SE. Equation 13 can be modified as Equation 19 is useful to implement this concept.

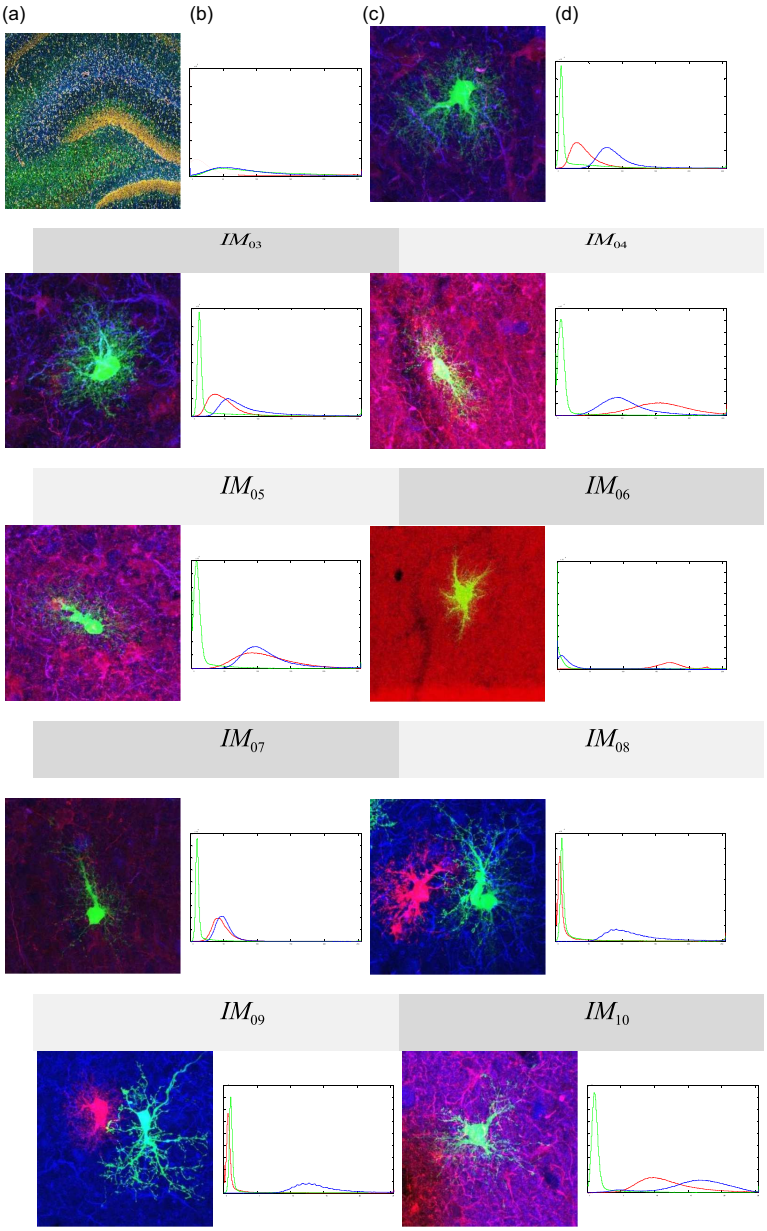
$$\begin{aligned}&\widehat{\Lambda}^{O}_{IMG}\left(IMG', \varphi_{low}, \varphi_{high}\right)\\ &= \max\left\{\Lambda^{O}_{IMG}(IMG')_{se_{\varphi_{low}}}, \Lambda^{O}_{IMG}(IMG')_{se_{\varphi_{low}+1}}, \right.\\ &\left. \ldots\ldots, \Lambda^{O}_{IMG}(IMG')_{se_{\varphi_{high}}}\right\}\end{aligned} \tag{19}$$

In this equation, $\varphi_{low} \le \varphi \le \varphi_{high}$. The error rate can be controlled by determining a small threshold value τ to preserve the important edge information [42, 43], and the upper bound can be determined using the following expression:

$$\left\{\widehat{\Lambda}^{O}_{IMG}\left(IMG', \varphi_{low}, \varphi_{high}\right) - \widehat{\Lambda}^{O}_{IMG}\left(IMG', \varphi_{low}, \varphi_{high}+1\right)\right\} \le \tau \tag{20}$$

The selection of the threshold value is important because a higher threshold value can increase the error rate and the increasing value φ_{high} will increase the computational burden. The image samples are collected from the sectioned tissue of the hippocampus region of the rat and given in Figure 2 (the image IDs are provided at the top of every image for future identification) along with their histograms. Experiments are carried out on 198 sample images. The effect of the different sizes of the SEs on the superpixels can be observed in Figures 3 and 4, where disk and

Figure 2
Test images and their corresponding histograms: (a) and (c) the original images, (b) and (d) the corresponding histogram of the microscopic image (immediately left)



square SEs are used, respectively. Figures 3(i) and 4(i) show that the number of superpixels is gradually decreasing with the increasing size of the superpixel. The image of the hippocampus under consideration is collected from the repository of the Center for Research in Biological Systems [44]. All other test images of the hippocampus region, which are considered in this work, are collected from the same source, and all experiments are performed in the Matlab environment using an Intel® Core i3 processor with 1.8 GHz speed and 4 GB RAM. While the experimental analysis is centered on hippocampal microscopic images, the employed dataset demonstrates substantial variability in cellular structures, illumination conditions, and background characteristics. This inherent diversity enables a rigorous assessment of the proposed segmentation framework under realistic and challenging imaging scenarios.

The complete dataset employed in this study comprises 198 hippocampus microscopic images acquired from rat brain specimens. These images were utilized throughout the development and validation of the proposed SUFGSO-based segmentation framework, including parameter selection, robustness analysis, and qualitative assessment. For clarity of presentation and to avoid visual redundancy, only 10 representative images were selected for detailed visualization and comparative illustration in the Results section. These images were chosen to capture the diversity present in the dataset in terms of cellular density, contrast variations, and structural complexity.

To evaluate robustness and generalizability, quantitative performance metrics were computed over the full set of 198 images. Consequently, the reported results represent aggregated performance statistics rather than outcomes from a limited subset,

Figure 3
Effect of *disk structuring element* of different sizes on superpixel: (a)–(h) superpixel images obtained after applying structuring elements of size 3–10, respectively, (i) relation between the size of the structuring elements and the number of superpixels

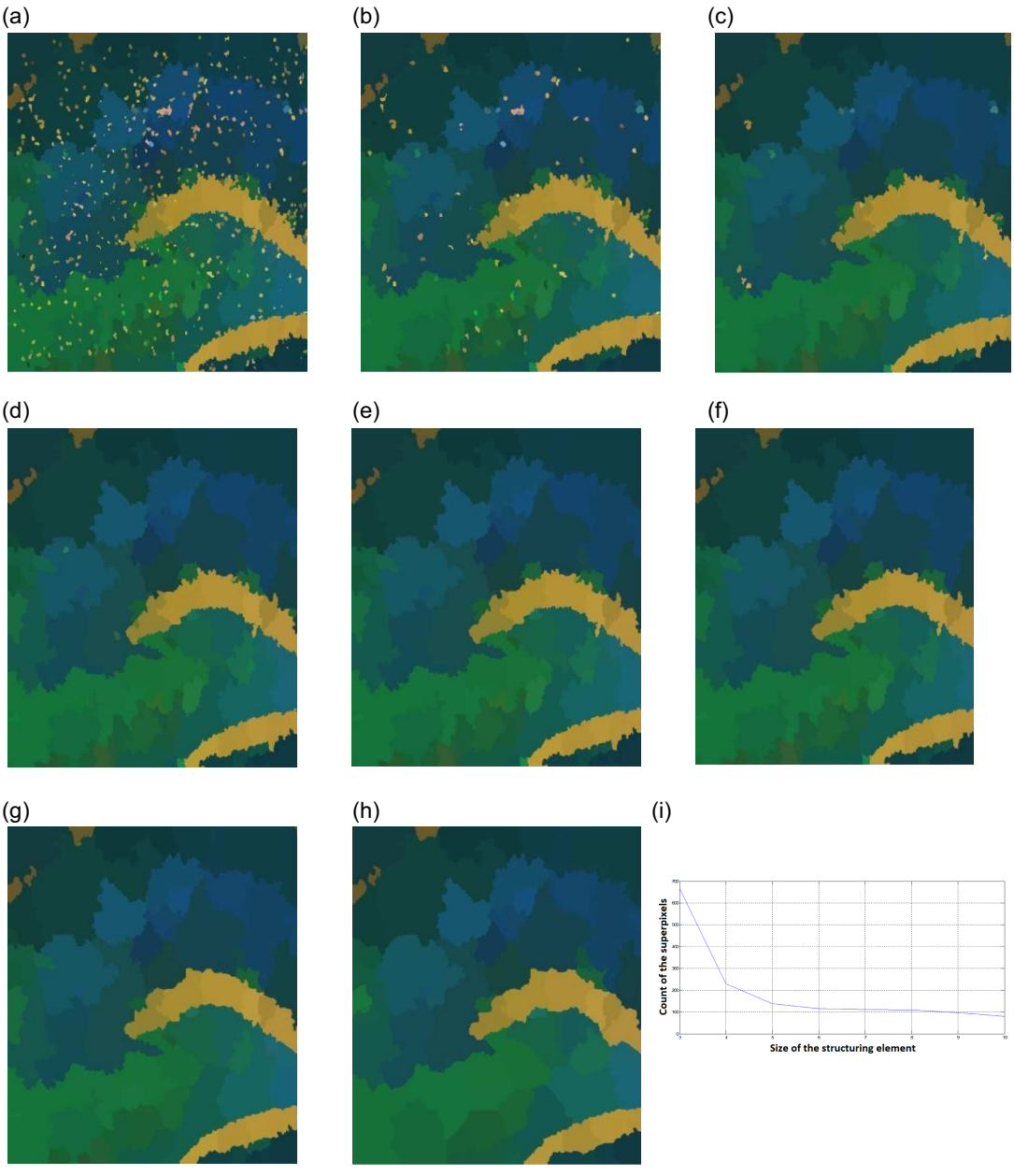
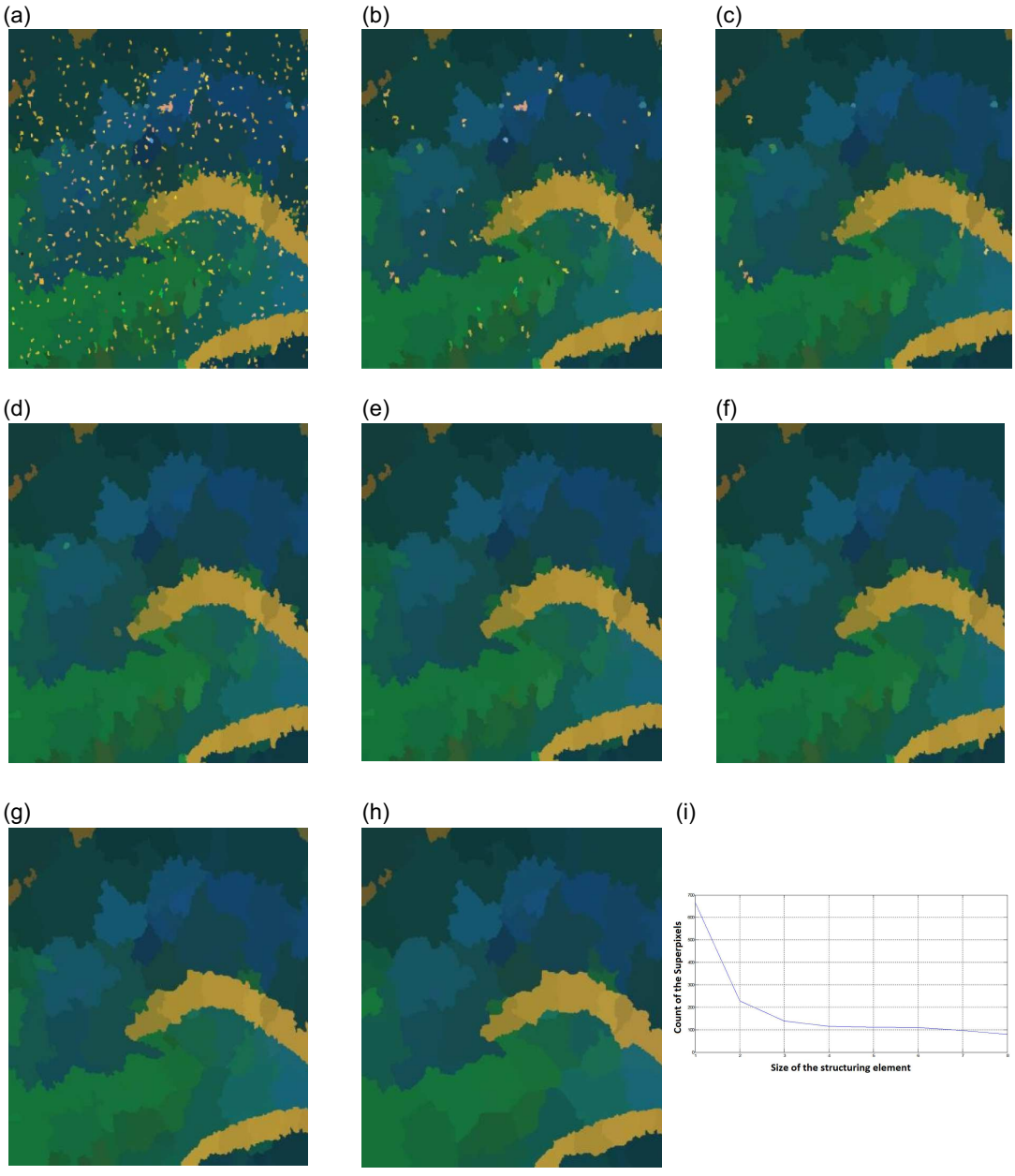


Figure 4
Effect of *disk structuring element* of different sizes on superpixel: (a)–(h) superpixel images obtained after applying structuring elements of size 3–10, respectively, (i) relation between the size of the structuring elements and the number of superpixels



ensuring that the evaluation reflects overall method behavior and is not biased toward individual image instances.

The fuzzy clustering objective function is modified depending on the superpixel-based method. It is essential to incorporate the fuzzy type-II clustering with the superpixel-based approach so that the advantage of both methods can be exploited [45]. In the modified objective function (which is given in Equation 22), every superpixel is represented by a representative value that can be computed using Equation 21.

$$\upsilon_q = \frac{1}{PxCnt_q} \sum_{w \in R_q} px_w \tag{21}$$

$$O_\lambda = \sum_{k=1}^{pCnt} \sum_{l=1}^{cCnt} PxCnt_k \mu_{kl}^{\lambda} \|\upsilon_k - c_l\|^2, \; where \; 1 \le \lambda < \infty \tag{22}$$

Here, $PxCnt_k$ is the number of pixels in the k^{th} region R_k. The membership value μ_{kl} can be computed using Equation 23, and the type-II fuzzy membership value can be computed using Equation 24. The cluster centers will be updated by the GSO algorithm.

$$\mu_{kl} = \frac{1}{\sum_{w=1}^{cCnt} \left(\frac{\|\upsilon_k - c_k\|}{\|\upsilon_k - c_w\|} \right)^{\frac{2}{\lambda - 1}}} \tag{23}$$

$$\eta_{kl} = \mu_{kl} - \frac{1 - \mu_{kl}}{2} \tag{24}$$

Equation 12 is not required in this context because the cluster centers can be guided and updated using the GSO algorithm. The proposed SUFGSO method is illustrated in Algorithm 1.

Algorithm 1:

The proposed SUFGSO approach
Input: The input image
Output: The segmented output
1: Find the gradient image using the method described by Hore et al. [9].
2: Use Equations (13) and (14) to find the superpixel image.
3: Randomly initialize the parameters for the subswarm level (m^p, ϑ^p, ψ^p, υ^p) and for the superswarm level (ϑ^p, ψ^p, υ).
4: Assign the fuzzy type-II membership values randomly to the superpixels using Equation (23).
5: *cntItr* ←1
6: Repeat until *cntItr* > *gIterationCnt*
7: Apply the PSO algorithm in the subswarm level.
8: Compute velocity and the new position of the particles using Equations (2) and (3).
9: Initialize the superswarm using Equation (5).
10: Apply PSO in the superswarm level.
11: Compute velocity and the new position of the particles using Equations (6) and (7).
end until
12: Construct the segmented image using optimal cluster centers by assigning the superpixels to their nearest cluster centers.
13: Return the segmented image.

The control parameters of the SUFGSO framework, such as swarm size, inertia weight, and fuzzy clustering settings, were selected in accordance with established guidelines reported in the literature and were kept constant throughout all experimental evaluations. Adopting a fixed-parameter configuration highlights the robustness of the proposed approach and eliminates the need for image-dependent parameter adjustment.

4.1. Computational complexity analysis

Let N represent the total number of image pixels and S denote the number of extracted superpixels, with $S \ll N$. The superpixel generation and associated preprocessing stages exhibit linear time complexity, that is, $O(N)$. Since fuzzy membership evaluation is carried out at the superpixel level, its computational cost is reduced to $O(S{\cdot}C)$, where C denotes the number of clusters. The primary computational overhead is attributed to the GSO stage. Considering P particles in each subswarm, K subswarms, and a total of T optimization iterations, the overall optimization complexity can be expressed as $O(T{\cdot}K{\cdot}P{\cdot}C)$. Importantly, as the optimization process operates on superpixel representations rather than individual pixels, the effective computational load is substantially lower than pixel-level approaches. Consequently, the proposed SUFGSO framework achieves a favorable balance between computational efficiency and segmentation accuracy.

4.2. Superpixel generation and parameter justification

In the proposed SUFGSO framework, superpixel representation is adopted to reduce computational overhead while retaining salient structural information in microscopic images. By aggregating perceptually homogeneous pixels into compact regions, superpixels facilitate region-level analysis that is inherently more robust to noise and local intensity fluctuations than pixel-level processing.

Morphological operations used during the superpixel preprocessing stage require careful selection of the SE size. In this study, SE sizes ranging from 3 × 3 to 7 × 7 were experimentally evaluated. Smaller SEs preserve fine structural details but may lead to over-segmentation and fragmented regions, whereas larger elements tend to oversmooth boundaries and suppress important features. Based on empirical evaluation across all datasets, an intermediate SE size of 5 × 5 was selected, as it consistently achieved a favorable trade-off between boundary preservation and noise reduction.

Thresholds employed during the superpixel refinement stage were determined using the statistical distribution of image intensities. To avoid image-dependent tuning, fixed threshold values within a normalized range were applied uniformly across all experiments. This strategy enhances reproducibility and minimizes the risk of overfitting the segmentation process to specific image characteristics.

The robustness of the SUFGSO framework with respect to parameter selection was further examined by analyzing the effect of varying the SE size and threshold values on segmentation performance. Experimental results show that moderate parameter variations lead to only minor changes in segmentation accuracy. This limited sensitivity is attributed to the subsequent type-II fuzzy clustering and GSO stages, which effectively compensate for small perturbations introduced during superpixel generation.

5. Results of the Simulation

The proposed SUFGSO method is tested on the 10 microscopic images of the hippocampus region, which were collected from the rats. The efficiency and the practical applicability of the proposed work are established by evaluating and comparing the SUFGSO approach quantitatively with the help of four well-known and frequently used cluster validity indices, which are discussed below. The performance of the proposed SUFGSO-based segmentation framework was primarily assessed using established cluster validity indices, namely, the Davies–Bouldin (DB), Xie–Beni (XB), Dunn, and β indices. These metrics are widely used in unsupervised segmentation and clustering studies because they quantitatively measure intra-cluster compactness and inter-cluster separation without requiring pixel-level ground truth annotations.

In this study, consistent ground truth segmentation masks were not available for the entire hippocampus microscopic image dataset. Consequently, overlap-based accuracy measures such as the Dice coefficient and Jaccard index could not be reliably computed for all images. To ensure a fair, uniform, and unbiased evaluation across the full dataset, cluster validity indices were therefore adopted as the primary performance measures.

Where pixel-level annotations are available, future extensions of this work will incorporate overlap-based metrics to enable a more direct assessment of segmentation accuracy. Such an evaluation will complement the unsupervised validity analysis and further strengthen quantitative validation. In addition to the main evaluation, overlap-based metrics (Dice and Jaccard) were employed in a superpixel parameter sensitivity analysis conducted on a limited subset of images for which ground truth annotations

were available. This localized analysis was intended to examine the influence of superpixel-related parameters on segmentation behavior rather than to provide a comprehensive accuracy comparison. Owing to the lack of consistent ground truth across the entire dataset, overlap-based metrics were not used in the primary comparative evaluation.

5.1. Cluster validity indices

1) **Davies–Bouldin index (*DBIndex*):** The ratio of the intra-cluster and inter-cluster distances is measured by the DB index [46]. Equation (25) can be used to determine the value of this index. The total number of clusters is represented by ε and $g \neq m$, $1 \leq g \leq \varepsilon$.

$$DBIndex = \frac{1}{\varepsilon}\sum_{g=1}^{\varepsilon}\max\left(\frac{d_w(a_g)+d_w(a_m)}{d_b(a_g,a_m)}\right) \tag{25}$$

2) **Xie–Beni index (*XBIndex*):** This index computes the ratio between the intra-cluster bonding and the separation. Equation (26) can be used to compute the value of the XB index [47].

$$XBIndex = \frac{\sum_{k=1}^{\varepsilon}\sum_{l=1}^{n} U_{kl}^2 \|V_k - X_l\|^2}{d_{\min}\|V_k - V_l\|^2} \tag{26}$$

3) **Dunn index (DI_n):** Equation (27) measures the value of the inter-cluster distance $dist\ (c_k,c_l\ )$ and the mean distance between each couple of clusters Υ_i [48].

$$DI_n = \min_{1\leq k\leq n}\left(\min_{1\leq l\leq n, j\neq i}\left(\frac{dist(c_k,c_l)}{\max_{1\leq i\leq m}\Upsilon_i}\right)\right) \tag{27}$$

4) **β index:** Equation (28) can be used to determine the value of this index that measures the ratio of the overall variation and the intra-cluster variation [49]. In this equation, $nPix_r$ is the pixel count in the r^{th} cluster, the ξ_{pq} is the pixel intensity, and $x_p = \frac{1}{nPix_r}\sum_{e=1}^{nPix_r} I_{er}$

$$\beta = \frac{\sum_{p=1}^{\varepsilon}\sum_{q=1}^{nPix_r}(I_{pq}-x)^2}{\sum_{p=1}^{\varepsilon}\sum_{q=1}^{nPix_r}(I_{pq}-x_p)^2} \tag{28}$$

5.2. Results of the experiments

The proposed SUFGSO method is evaluated in both qualitative and quantitative manners. A comparative study is performed with the Beam-ACO [50], adaptive PSO [51], efficient GA [52], and the modified cuckoo search (MCS) method [53]. Figure 5 gives a comparative overview of using the test image IM01. The final segmentation results after applying the proposed SUFGSO method, for the rest of the 9 images (i.e., IM_{02}–IM_{10}), are given in Figure 6. Tables 1, 2, 3, and 4 report the quantitative comparative results based on the *DBIndex, XBIndex, DI_n,* and β index, respectively, and the acceptable values are highlighted in boldface. The convergence of the proposed method is evaluated and compared for different numbers of clusters, and the convergence analysis is discussed in the next section.

To comprehensively evaluate the effectiveness and competitiveness of the proposed SUFGSO approach, a comparative analysis is conducted with both traditional metaheuristic-based segmentation algorithms and recent state-of-the-art methods. These include widely used optimization techniques such as Beam-ACO, MCS, adaptive PSO, and efficient GA, as well as more recent segmentation frameworks such as U-Net, Attention U-Net, DeepLabv3+, Fully Convolutional Networks (FCNs), and unsupervised clustering variants like Multi-Scale Fuzzy C-Means (clustering) (MCFCM) and K-means with SLIC superpixels.

Table 5 summarizes the average performance across all test images using four standard cluster validity indices: DB index, XB index, Dunn index, and the β index. While supervised deep learning methods show strong results, they rely heavily on large, annotated training datasets. In contrast, the proposed SUFGSO method consistently performs competitively or better, particularly in unsupervised contexts, demonstrating robustness and practical value for biomedical imaging applications where annotated data is scarce or unavailable.

5.3. Analysis of the convergence rate

Convergence analysis is a major part of the current topic of discussion. It is essential to understand the convergence of the proposed method, and it is an important parameter to be compared with other standard methods. This analysis gives an overview of the performance of the SUFGSO method for different numbers of clusters. A graphical analysis of the rate of convergence is presented in this subsection for the image ID IM_{01} using the Dunn index in Figure 7. The graphical comparison shows that the proposed SUFGSO method outperforms some of the standard methods for a certain number of clusters.

To enhance clarity and ensure reproducibility, Table 6 provides a comprehensive summary of all parameters employed in the proposed SUFGSO framework, covering optimization-related variables, type-II fuzzy clustering parameters, and image processing settings. The parameter values were chosen within well-established ranges commonly reported in the literature and were held constant for all experiments to guarantee consistent evaluation and fair comparison across datasets.

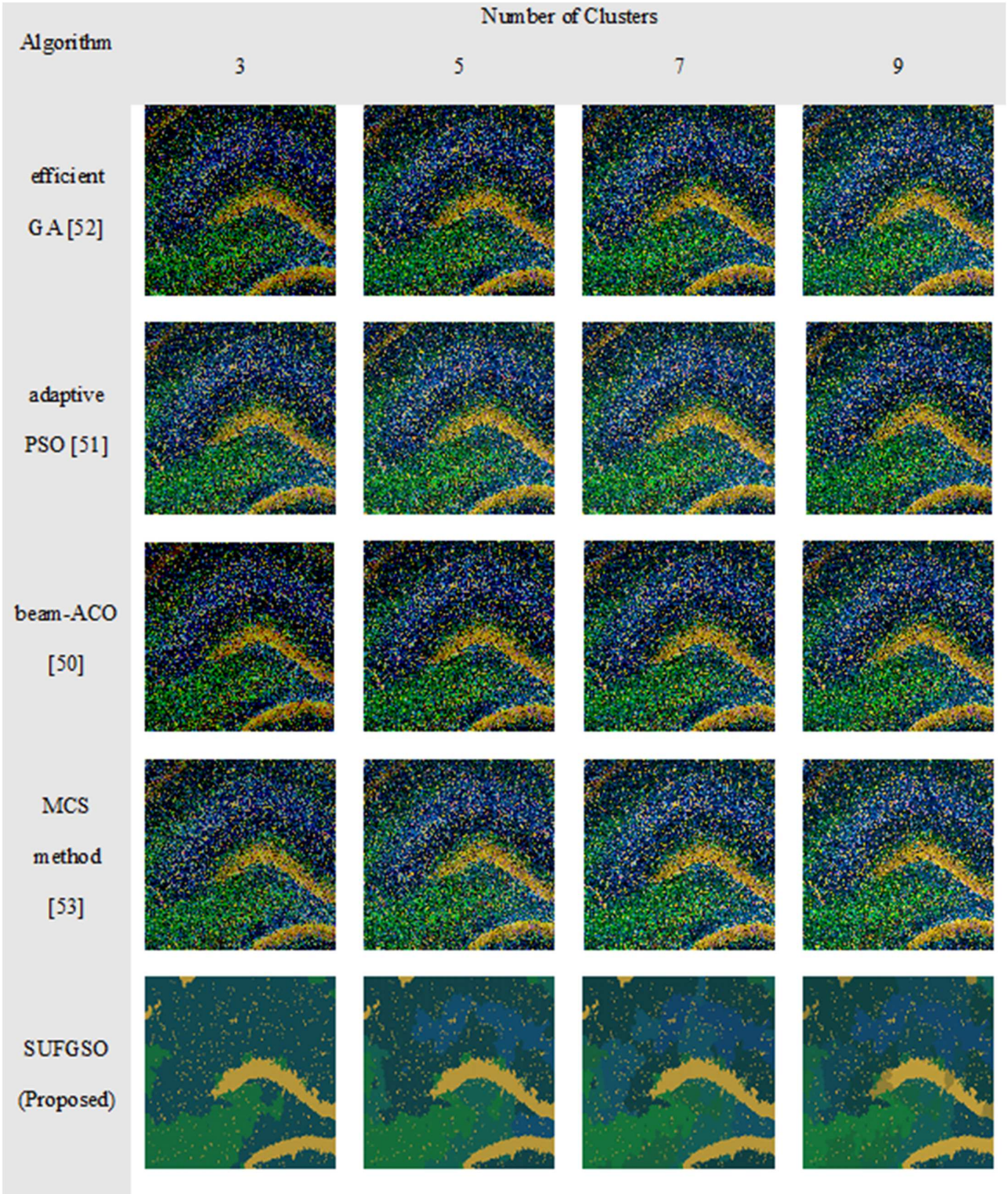
The robustness of the proposed method with respect to variations in superpixel parameters is further supported by the sensitivity analysis presented in Table 7.

Figure 8 presents a boxplot of Dice coefficients computed across all 198 hippocampus images, illustrating the median performance, interquartile range, and variability of the proposed SUFGSO framework, thereby demonstrating its robustness and generalizability.

The comparative analysis in this study is centered on optimization-based and nature-inspired segmentation methods, including GA, PSO, ACO, and MCS. These approaches were selected because they pursue similar unsupervised optimization objectives and operate under comparable assumptions regarding parameter control, search strategy, and computational complexity. This alignment ensures a fair and meaningful comparison with the proposed SUFGSO framework.

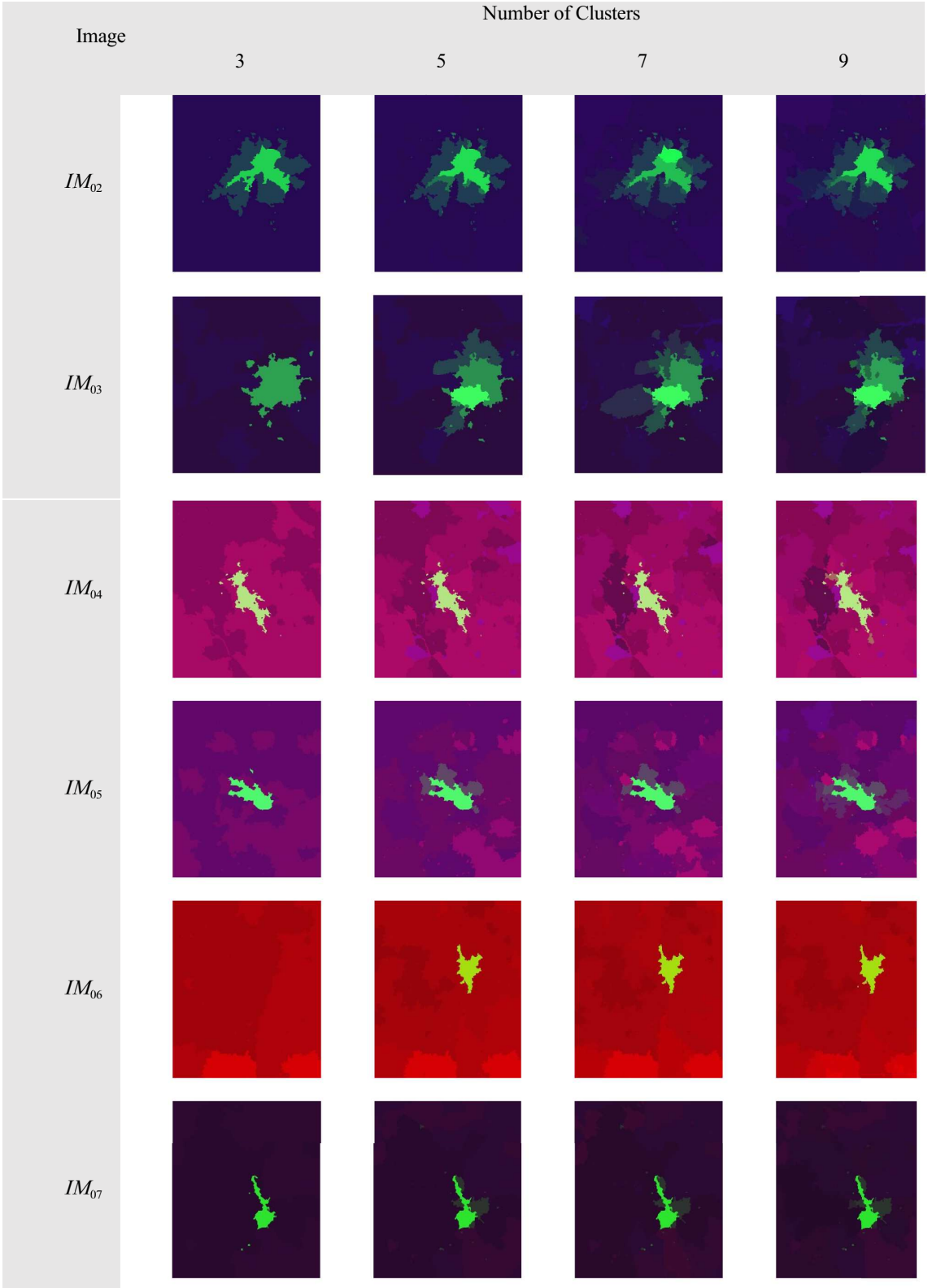
Although recent deep learning–based segmentation techniques, such as CNN-driven and superpixel-assisted neural models, have reported strong performance in various biomedical imaging applications, they typically depend on large volumes of annotated

Figure 5
A comparative study using IM_{01} for different numbers of clusters



data, extensive training pipelines, and substantial computational resources. Such requirements may be impractical in specialized microscopic imaging scenarios where expert annotations are limited or costly to obtain. In contrast, the proposed SUFGSO framework is entirely unsupervised and does not require labeled training data or prior model learning, making it well-suited to data-scarce environments. Accordingly, the selected baseline methods provide an appropriate evaluation context for assessing the effectiveness of SUFGSO within the class of unsupervised optimization-based segmentation techniques. Deep learning–based approaches are therefore discussed as complementary solutions rather than as directly comparable baselines.

Figure 6
The SUFGSO approach based on segmented output



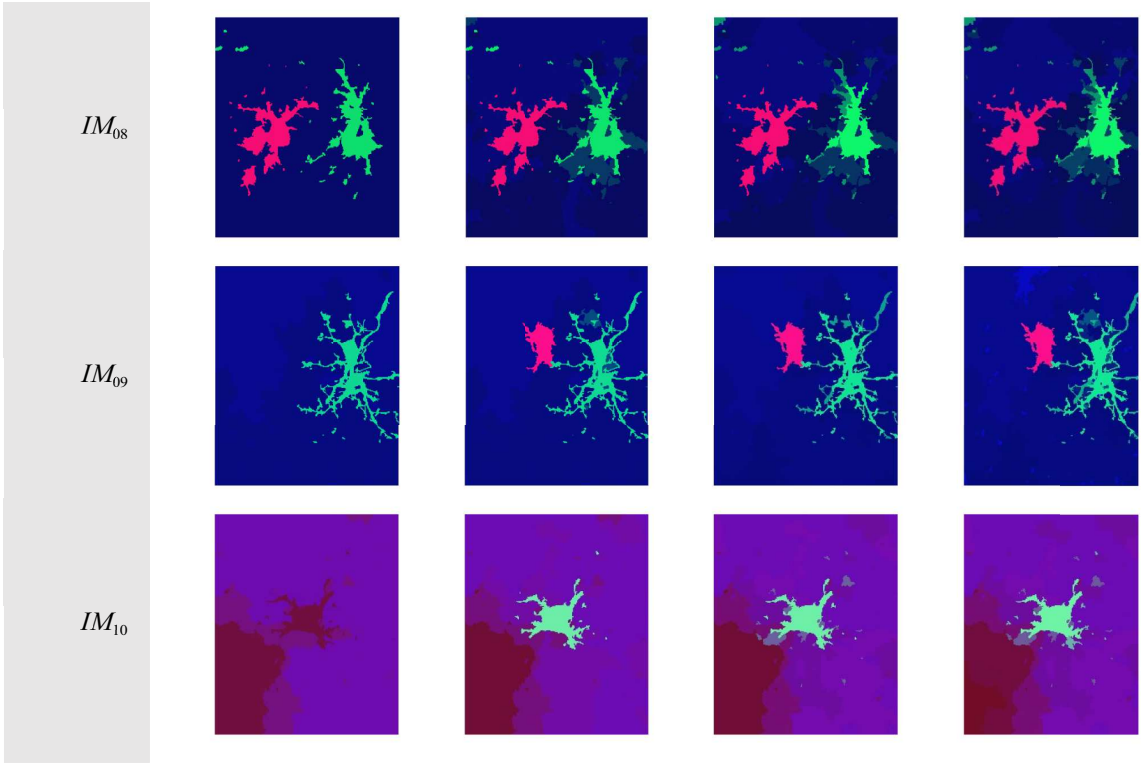


Table 1
Quantitative comparison of some standard methods using the Davies–Bouldin index for different numbers of clusters (highlighted values represent the acceptable values)

Image ID	Algorithm	No. of clusters			
		3	5	7	9
	Efficient genetic algorithm [52]	1.19767659	1.84866269	2.89843	1.435701044
	Adaptive particle swarm optimization [51]	2.07200534	1.80807727	2.30973804	2.340613771
IM_{01}	Beam-ant colony optimization [50]	0.83215941	0.6927213	0.60537095	1.186205142
	Modified cuckoo search method [53]	1.56193174	1.83319809	0.64315198	1.793002977
	SUFGSO (proposed)	1.28987669	1.2207349	0.60447801	0.950601811
	Efficient genetic algorithm [52]	1.07198351	0.76653188	1.83921626	1.287533643
	Adaptive particle swarm optimization [51]	1.42455298	2.62563653	2.3576109	1.702672906
IM_{02}	Beam-ant colony optimization [50]	3.29543036	2.44091584	1.36696859	2.049065715
	Modified cuckoo search method [53]	1.56869044	2.3298641	2.21990596	2.247272222
	SUFGSO (proposed)	1.28404631	0.50529826	1.6966998	3.837301973
	Efficient genetic algorithm [52]	1.52272589	1.36747697	0.99797831	1.323944348
	Adaptive particle swarm optimization [51]	1.49942365	1.22052383	1.14389777	1.833580077
$^{IM}03$	Beam-ant colony optimization [50]	0.97092456	1.96539491	2.76629682	1.79356845
	Modified cuckoo search method [53]	0.98295511	1.02868739	2.25656379	1.55563453
	SUFGSO (proposed)	2.10105967	2.06653225	0.68642029	0.679494729
	Efficient genetic algorithm [52]	0.95409252	1.85485665	2.48115536	1.808671083
	Adaptive particle swarm optimization [51]	2.48881896	2.3301901	1.63587525	1.780929203
IM_{04}	Beam-ant colony optimization [50]	1.22905618	0.88120473	1.90761593	1.528918223

(*Continued*)

Table 1
(*Continued*)

Image ID	Algorithm	No. of clusters			
	Modified cuckoo search method [53]	2.66121385	0.75221579	1.17090734	1.487581222
	SUFGSO (proposed)	2.3494863	1.46494873	1.05076411	2.843409042
	Efficient genetic algorithm [52]	2.17358931	2.17492377	0.99303958	1.016599747
	Adaptive particle swarm optimization [51]	0.87717168	1.34164074	1.21095781	1.498079833
IM_{05}	Beam-ant colony optimization [50]	1.41889364	2.66146666	1.39022295	2.834124109
	Modified cuckoo search method [53]	1.65145165	3.66741263	2.91531181	1.75990858
	SUFGSO (proposed)	2.62827009	1.87307884	0.72925014	2.011664713
	Efficient genetic algorithm [52]	0.96372479	1.58464905	1.17246859	1.132231458
	Adaptive particle swarm optimization [51]	1.17824629	0.80486385	1.4616967	2.955162124
IM_{06}	Beam-ant colony optimization [50]	0.93290309	1.20904924	1.58982334	2.004688296
	Modified cuckoo search method [53]	1.20942496	1.04602263	0.43895552	0.456756222
	SUFGSO (proposed)	0.43773739	0.93022601	1.53774741	1.47408747
	Efficient genetic algorithm [52]	0.99732078	1.81329654	1.24311841	1.646032775
$IM07$	Adaptive particle swarm optimization [51]	0.99850287	1.56049627	1.54453129	1.830764475
	Beam-ant colony optimization [50]	2.16562262	1.80382499	2.83471402	0.980476705
	Modified cuckoo search method [53]	2.3373178	2.50054794	0.94190857	2.653373269
	SUFGSO (proposed)	0.42775151	2.44023948	1.18163572	1.163755439
	Efficient genetic algorithm [52]	3.13068626	3.16540975	2.66298699	2.694763032
	Adaptive particle swarm optimization [51]	1.60409828	1.7866609	2.12990264	1.198148305
IM_{08}	Beam-ant colony optimization [50]	0.73912051	2.99581354	1.8696097	1.295045013
	Modified cuckoo search method [53]	2.02185563	1.85235664	1.92139577	1.374818127
	SUFGSO (proposed)	0.69681774	1.8752761	1.83455956	1.396932145
	Efficient genetic algorithm [52]	0.87357376	2.2919851	1.39462181	3.274354801
	Adaptive particle swarm optimization [51]	1.36185156	1.73340791	0.53818721	3.186937675
IM_{09}	Beam-ant colony optimization [50]	0.88054391	1.77246245	2.28229698	1.598217703
	Modified cuckoo search method [53]	1.78360445	1.63559231	0.50005748	2.565307323
	SUFGSO (proposed)	0.44718438	0.41852698	1.55620497	2.384480766
	Efficient genetic algorithm [52]	1.32707252	1.15304945	3.16126158	2.159250904
	Adaptive particle swarm optimization [51]	1.60656264	1.31367086	1.87127733	1.643180276
IM_{10}	Beam-ant colony optimization [50]	0.82418003	1.47866895	1.44271337	1.14578388
	Modified cuckoo search method [53]	1.41394883	1.12630018	0.19292111	2.50022123
	SUFGSO (proposed)	2.03807595	0.73799092	0.9297664	1.128679299

Table 2
Quantitative comparison of some standard methods using Xie–Beni index for different numbers of clusters (highlighted values represent the acceptable values)

Image ID	Algorithm	No. of clusters			
		3	5	7	9
	Efficient genetic algorithm [52]	3.0361998	1.79595054	1.2045761	1.836324775
	Adaptive particle swarm optimization [51]	2.35171627	1.63134726	1.63921589	1.723451571
IM_{01}	Beam-ant colony optimization [50]	1.07734619	1.61075425	0.88974142	2.664424439
	Modified cuckoo search method [53]	1.31253601	2.17001438	1.17711956	1.662697022
	SUFGSO (proposed)	3.11535055	0.7526194	0.58502058	0.410551105
	Efficient genetic algorithm [52]	2.63061363	2.80038591	2.54128087	2.477282211
	Adaptive particle swarm optimization [51]	2.20540323	2.22826706	2.16712106	2.611136552
IM_{02}	Beam-ant colony optimization [50]	2.34577539	3.30787898	3.24716274	2.632724596

(*Continued*)

Table 2
(*Continued*)

Image ID	Algorithm	No. of clusters			
	Modified cuckoo search method [53]	3.69179397	2.45441187	1.44446135	1.593892063
	SUFGSO (proposed)	0.9635441	2.09074046	1.74527833	1.62468822
	Efficient genetic algorithm [52]	4.29005674	3.20301628	2.40693231	3.181281993
	Adaptive particle swarm optimization [51]	2.99172481	4.32214245	3.12809412	3.380633058
IM_{03}	Beam-ant colony optimization [50]	3.67001368	4.01067491	2.1550199	3.032183764
	Modified cuckoo search method [53]	3.221136	1.91575054	3.54420513	3.097224296
	SUFGSO (proposed)	2.06066938	1.79664973	3.64599791	1.213170794
	Efficient genetic algorithm [52]	2.06199241	2.48652806	3.0115882	1.921529919
	Adaptive particle swarm optimization [51]	1.34599817	1.36433841	1.86006857	2.448293801
IM_{04}	Beam-ant colony optimization [50]	1.03181064	0.93501194	2.01326336	1.631491863
	Modified cuckoo search method [53]	2.26037689	1.00794168	1.92420813	1.419872091
	SUFGSO (proposed)	1.10985226	1.59534362	2.24690351	2.100503531
	Efficient genetic algorithm [52]	3.19965262	2.25389519	1.6781744	1.717644535
	Adaptive particle swarm optimization [51]	2.26597829	1.25139468	1.07162663	2.743600451
IM_{05}	Beam-ant colony optimization [50]	2.90158611	1.89674818	2.00724016	2.505351996
	Modified cuckoo search method [53]	2.23348778	1.63361274	1.57935419	0.642324595
	SUFGSO (proposed)	1.51743298	0.96956652	1.01660002	1.017512065
	Efficient genetic algorithm [52]	2.38725909	1.56154275	0.91327071	2.535793225
	Adaptive particle swarm optimization [51]	0.8999061	0.93384946	1.469079	1.760517042
IM_{06}	Beam-ant colony optimization [50]	1.6393879	0.9297595	2.34977583	1.777317935
	Modified cuckoo search method [53]	1.34073301	1.33413076	0.84601374	1.076130539
	SUFGSO (proposed)	1.28739741	1.16210287	1.40787883	0.581152542
	Efficient genetic algorithm [52]	3.80928836	5.04680837	3.11874903	4.249505616
IM_{07}	Adaptive particle swarm optimization [51]	3.80493063	3.65652575	2.44201061	1.961169188
	Beam-ant colony optimization [50]	2.67745485	3.44861913	2.67121659	2.605521415
	Modified cuckoo search method [53]	3.04316628	1.96760189	3.9035894	2.683038517
	SUFGSO (proposed)	2.3446003	3.54890798	1.34646377	3.148449214
	Efficient genetic algorithm [52]	2.95474213	2.28993552	1.00247191	2.166330669
	Adaptive particle swarm optimization [51]	1.09337097	2.22266819	3.73887534	3.437054057
IM_{08}	Beam-ant colony optimization [50]	1.22668019	1.1329084	0.64254461	2.630924266
	Modified cuckoo search method [53]	2.46353818	2.7467644	2.86601178	2.025558436
	SUFGSO (proposed)	2.43240071	0.89902576	1.6622046	1.84327027
	Efficient genetic algorithm [52]	1.91355315	1.03157297	0.81918361	2.993876219
	Adaptive particle swarm optimization [51]	3.17329791	1.30869767	1.90724839	2.606692539
IM_{09}	Beam-ant colony optimization [50]	2.05485213	2.72516598	2.2190952	2.950597731
	Modified cuckoo search method [53]	0.97215878	1.76615122	1.76300386	1.200670581
	SUFGSO (proposed)	1.13722801	2.09210184	1.57355279	1.713777777
	Efficient genetic algorithm [52]	2.70718109	2.54431224	1.25242849	1.40686934
	Adaptive particle swarm optimization [51]	1.55090756	2.02959244	0.81320774	1.887138557
IM_{10}	Beam-ant colony optimization [50]	1.31165848	0.90239621	1.01065354	2.547875548
	Modified cuckoo search method [53]	0.97159556	1.98130869	0.80321158	2.411030319
	SUFGSO (proposed)	2.53788369	1.09127054	0.71312897	0.972404383

Table 3
Quantitative comparison of some standard methods using the Dunn index for different numbers of clusters (highlighted values represent the acceptable values)

Image ID	Algorithm	No. of clusters			
		3	5	7	9
	Efficient genetic algorithm [52]	1.30959986	2.08885942	3.23387709	1.706250744
	Adaptive particle swarm optimization [51]	3.2565953	4.47794141	3.79411727	2.558002808
IM_{01}	Beam-ant colony optimization [50]	3.68941034	2.60399067	2.64408552	3.127708545
	Modified cuckoo search method [53]	2.55712234	3.50423133	1.93685475	3.426235408
	SUFGSO (proposed)	2.97008601	4.77794576	3.30081197	2.643300841
	Efficient genetic algorithm [52]	1.41399197	0.30605217	0.6640814	0.261848281
	Adaptive particle swarm optimization [51]	1.6025736	0.1925035	2.93080924	0.71224029
IM_{02}	Beam-ant colony optimization [50]	0.49494106	0.81751033	1.67577928	1.476196264
	Modified cuckoo search method [53]	0.80274966	0.97088161	1.5802531	1.905712022
	SUFGSO (proposed)	1.30603339	2.49499195	2.97256143	1.26470563
	Efficient genetic algorithm [52]	0.57715667	1.12810571	2.6285921	2.118886458
	Adaptive particle swarm optimization [51]	1.07707018	0.79317075	2.36340569	0.61038888
IM_{03}	Beam-ant colony optimization [50]	0.94414332	1.22804251	1.6719936	0.4366487
	Modified cuckoo search method [53]	1.77582141	0.71924061	0.89574164	3.019947512
	SUFGSO (proposed)	1.5371799	0.68340654	1.28983813	2.954250119
	Efficient genetic algorithm [52]	0.85057398	1.12650091	0.79786922	2.133366942
	Adaptive particle swarm optimization [51]	0.65110923	0.78799958	2.99586142	3.222811587
IM_{04}	Beam-ant colony optimization [50]	2.5135739	0.80067189	1.28150041	0.742909021
	Modified cuckoo search method [53]	1.32107243	1.50296469	2.0007755	2.245130095
	SUFGSO (proposed)	2.976198	1.0162849	0.39613292	1.648928459
	Efficient genetic algorithm [52]	1.24497183	1.72597118	0.93092451	1.098027664
	Adaptive particle swarm optimization [51]	0.7370837	0.76346649	2.19260747	0.440437231
IM_{05}	Beam-ant colony optimization [50]	1.50142219	0.31640665	1.99943697	2.264939528
	Modified cuckoo search method [53]	1.61131036	0.42605694	2.37207627	1.120896641
	SUFGSO (proposed)	2.45579187	2.21720243	1.83067493	1.823524287
	Efficient genetic algorithm [52]	1.0133718	2.03869787	1.8386312	1.100155169
	Adaptive particle swarm optimization [51]	0.26526712	2.44478578	2.38930664	2.683326201
IM_{06}	Beam-ant colony optimization [50]	3.2980254	2.45394297	1.64264634	2.744321435
	Modified cuckoo search method [53]	1.32983747	-0.0705115	2.07254542	1.393110254
	SUFGSO (proposed)	1.54337316	1.37852562	3.62203962	2.158475777
	Efficient genetic algorithm [52]	0.9912406	0.98099455	2.30763838	2.870097192
	Adaptive particle swarm optimization [51]	1.87859119	1.88873161	1.8733476	0.96538965
IM_{07}	Beam-ant colony optimization [50]	1.10151464	0.94302963	0.48438997	0.413279307
	Modified cuckoo search method [53]	1.58860326	1.9170295	0.03116362	3.327746487
	SUFGSO (proposed)	2.53269834	1.91057363	1.05831828	3.415732284
	Efficient genetic algorithm [52]	1.04061173	0.0874821	0.37609632	2.453488921
	Adaptive particle swarm optimization [51]	2.57237674	1.30081597	1.22429687	0.391366907
IM_{08}	Beam-ant colony optimization [50]	1.24657936	3.20154623	0.0841101	0.933206586
	Modified cuckoo search method [53]	0.88524465	1.59093715	0.57928694	0.425103489
	SUFGSO (proposed)	3.37451701	3.62651381	2.07757219	1.998636609
	Efficient genetic algorithm [52]	2.23428227	1.54973375	1.54347483	2.683212247
	Adaptive particle swarm optimization [51]	3.81466967	1.55939525	1.35818696	1.427481488
IM_{09}	Beam-ant colony optimization [50]	2.2346634	3.8055466	3.06938318	1.652482124

(*Continued*)

Table 3
(*Continued*)

Image ID	Algorithm	No. of clusters			
	Modified cuckoo search method [53]	4.00148195	3.61608928	3.40931632	4.029169249
	SUFGSO (proposed)	3.97215046	3.65276502	3.07438982	1.937605302
	Efficient genetic algorithm [52]	0.95861672	2.21763258	2.31682409	1.854681042
	Adaptive particle swarm optimization [51]	4.05532617	4.01835802	4.53050063	3.519291521
IM_{10}	Beam-ant colony optimization [50]	3.7978737	2.24081643	1.80069182	3.589180072
	Modified cuckoo search method [53]	3.05590854	3.69246064	2.2359491	3.00956479
	SUFGSO (proposed)	2.5481343	4.72442038	2.92258409	2.015598984

Table 4
Quantitative comparison of some standard methods using β index for different numbers of clusters (highlighted values represent the acceptable values)

Image ID	Algorithm	No. of clusters			
		3	5	7	9
	Efficient genetic algorithm [52]	1.22959204	2.28186677	2.50260438	1.732484833
	Adaptive particle swarm optimization [51]	0.74061825	1.97834625	2.94414723	3.061315455
IM_{01}	Beam-ant colony optimization [50]	1.15560027	1.85116502	1.34414274	2.268929405
	Modified cuckoo search method [53]	3.10144569	1.67122783	2.11421683	2.07025988
	SUFGSO (proposed)	1.25040555	1.7896598	3.55263153	2.100113774
	Efficient genetic algorithm [52]	1.83000535	3.04697937	2.27680926	0.619186358
	Adaptive particle swarm optimization [51]	2.88541722	1.18914389	1.81536107	0.572882349
IM_{02}	Beam-ant colony optimization [50]	0.52088903	2.49219617	1.28295488	3.20396946
	Modified cuckoo search method [53]	2.67922041	2.48893169	2.10404811	2.693810866
	SUFGSO (proposed)	1.291212	1.97709191	3.96014289	1.844907629
	Efficient genetic algorithm [52]	1.33482466	0.89268452	2.17687105	2.362862568
	Adaptive particle swarm optimization [51]	0.32393808	2.51231759	2.08082761	2.606107261
IM_{03}	Beam-ant colony optimization [50]	0.55433715	0.66134161	1.63505247	1.563036186
	Modified cuckoo search method [53]	1.80614236	1.39205549	1.17867044	2.16182576
	SUFGSO (proposed)	1.10817966	3.29115642	2.29060567	2.595609198
	Efficient genetic algorithm [52]	1.2924449	0.99284807	2.5433069	2.255263433
	Adaptive particle swarm optimization [51]	2.06132899	2.41576087	3.11647134	2.710377465
IM_{04}	Beam-ant colony optimization [50]	3.39115542	2.11734252	2.3865507	1.268453298
	Modified cuckoo search method [53]	1.95056538	1.90134887	2.05594298	1.417500746
	SUFGSO (proposed)	1.72119221	2.7380142	4.82938737	1.390624862
	Efficient genetic algorithm [52]	1.2254123	1.84359249	1.236993	1.264044802
	Adaptive particle swarm optimization [51]	2.23589998	3.93938749	3.66057964	1.424905782
IM_{05}	Beam-ant colony optimization [50]	2.49418252	2.25556364	2.13398079	1.718883718
	Modified cuckoo search method [53]	2.33120612	2.39103657	3.77197853	3.082412638
	SUFGSO (proposed)	3.13809604	3.61992723	3.04370925	2.05252124
	Efficient genetic algorithm [52]	2.23352864	2.79999674	1.79103759	2.579146914
	Adaptive particle swarm optimization [51]	2.25775738	2.85521345	1.69384428	3.379436617
IM_{06}	Beam-ant colony optimization [50]	0.49515545	1.51910226	1.89811887	3.18678836
	Modified cuckoo search method [53]	3.09196039	2.99991357	2.35369506	2.890530494
	SUFGSO (proposed)	0.53389638	2.40378705	4.22406492	1.587738339
	Efficient genetic algorithm [52]	1.44824363	2.2974353	1.19663929	3.038215289
IM_{07}	Adaptive particle swarm optimization [51]	2.20079666	1.8023298	3.52784954	4.042539385

(*Continued*)

Table 4
(*Continued*)

Image ID	Algorithm	No. of clusters			
	Beam-ant colony optimization [50]	0.65882837	1.23303759	2.38522355	1.545504745
	Modified cuckoo search method [53]	3.77821674	4.01099851	2.41682084	0.686906657
	SUFGSO (proposed)	1.33492213	3.94046743	2.08528065	1.810174374
	Efficient genetic algorithm [52]	2.4557476	3.04822538	2.09847134	2.20883189
	Adaptive particle swarm optimization [51]	0.9763061	1.49378468	1.84892925	0.908132713
IM_{08}	Beam-ant colony optimization [50]	1.86103839	2.47491656	2.01847797	1.650601127
	Modified cuckoo search method [53]	3.04560643	3.85605121	3.74380896	2.433786589
	SUFGSO (proposed)	2.89299047	4.28323918	2.15554789	2.416974931
	Efficient genetic algorithm [52]	1.14593221	1.30549378	2.41054919	2.087767291
	Adaptive particle swarm optimization [51]	2.19165077	0.76590265	1.88302031	1.324633819
IM_{09}	Beam-ant colony optimization [50]	3.02859717	2.78791415	2.43858593	1.55123079
	Modified cuckoo search method [53]	2.36370813	1.44110428	2.27331822	2.592134075
	SUFGSO (proposed)	2.74624879	2.16594671	4.68458712	3.273175555
	Efficient genetic algorithm [52]	0.79846551	1.88953312	2.81850528	2.403462968
	Adaptive particle swarm optimization [51]	1.12519978	2.78074757	2.73755037	3.768010343
IM_{10}	Beam-ant colony optimization [50]	1.02951762	2.17469727	2.66767931	2.618995573
	Modified cuckoo search method [53]	2.50879479	1.70994984	1.96940483	1.498769566
	SUFGSO (proposed)	0.78262303	2.05180114	3.90755286	1.964268443

Table 5
Average performance comparison with SUFGSO and new methods

Method	Supervision	DB index ↓	Xie–Beni ↓	Dunn ↑	β–index ↑	Remarks
SUFGSO (proposed)	Unsupervised	1.32	1.64	2.51	2.65	No supervision
Beam-ACO	Unsupervised	1.78	2.07	2.05	2.24	Bio-inspired heuristic
MCS	Unsupervised	1.91	2.02	1.88	2.13	Cuckoo search optimized
Adaptive PSO	Unsupervised	1.95	2.31	1.67	2.20	Fast convergence
Efficient GA	Unsupervised	1.87	2.48	1.54	2.22	Genetic operator strength
U-Net	Supervised	1.25	1.52	2.40	2.55	Best DL model, but needs labels
Attention U-Net	Supervised	1.21	1.48	2.42	2.58	Strong in complex structures
DeepLabv3+	Supervised	1.18	1.50	2.43	2.54	Very accurate boundary detection
FCN	Supervised	1.35	1.71	2.31	2.47	Legacy DL model
MCFCM	Unsupervised	1.46	1.80	2.29	2.35	No training, fuzzy enhanced
K-means + SLIC	Unsupervised	1.58	1.89	2.00	2.10	Simple but effective baseline

Figure 7

Analysis and comparison of the rate of convergence for image ID IM01 using various methods and for different numbers of clusters using the Dunn index. The plotted curves using (a) efficient GA, (b) adaptive PSO, (c) Beam-ACO, (d) MCS, and (e) Superpixel-based Fuzzy Electromagnetism-like Optimization (SUFEMO) (proposed)

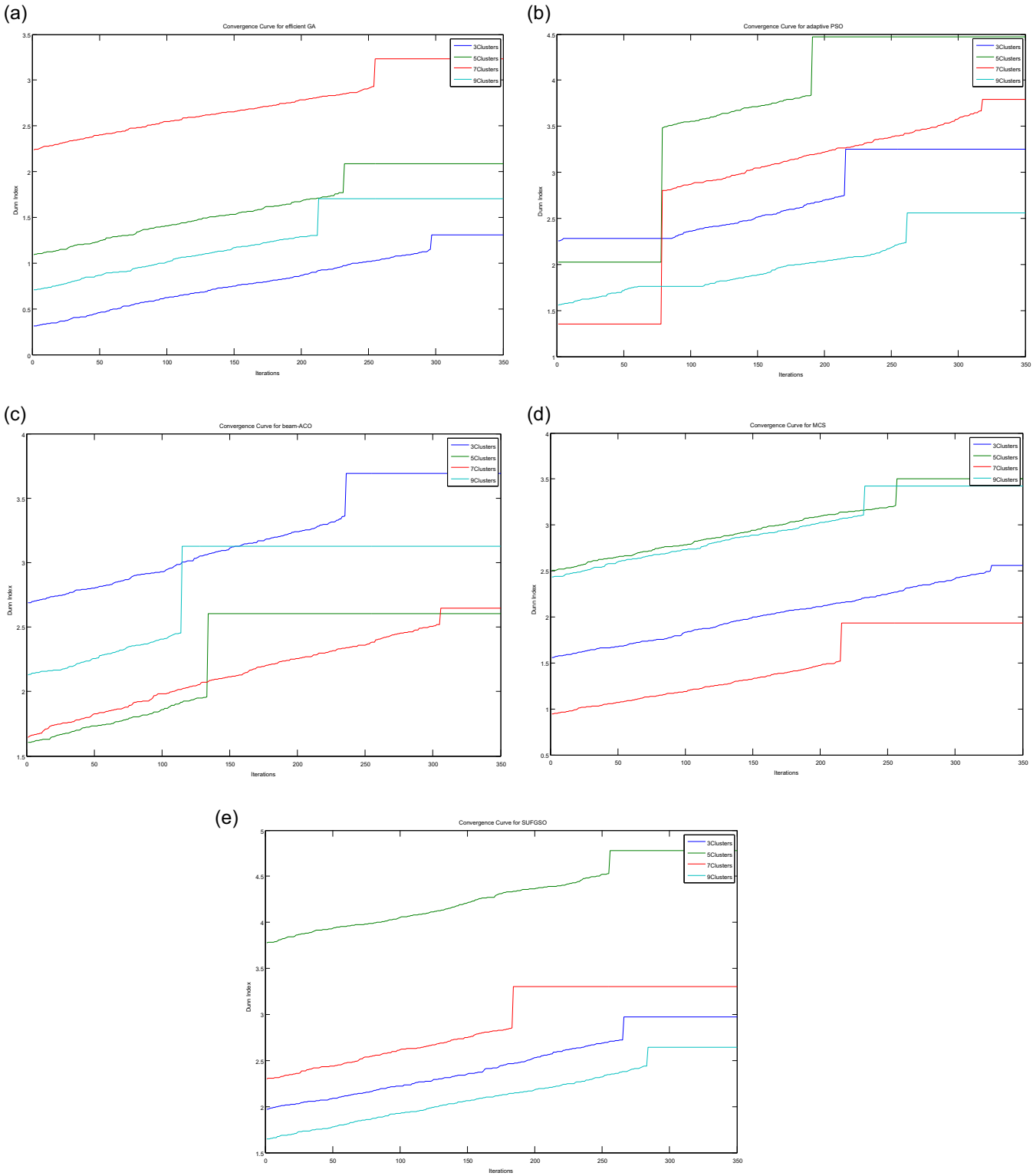


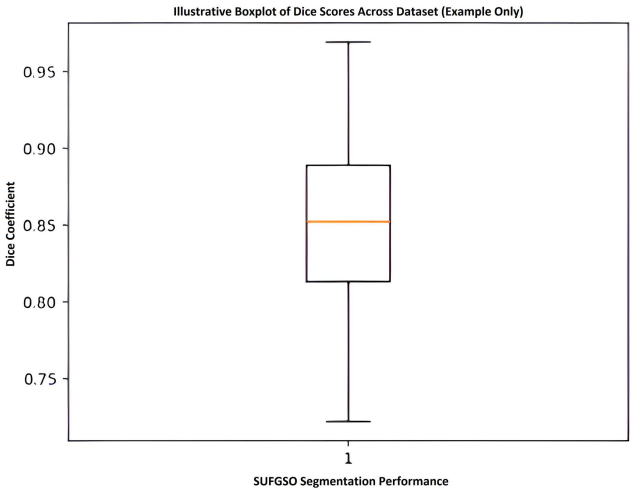
Table 6
Summary of parameters used in the proposed SUFGSO framework

Symbol/parameter	Description	Typical range/value	Value used
C	Number of clusters	2–6	3
S	Number of superpixels	100–500	200
ϕlow	Lower bound of GSO attraction coefficient	(0, 1)	0.2
ϕhigh	Upper bound of GSO attraction coefficient	(0, 1)	0.9
τ	Convergence threshold	10^{-4}	10^{-3}
P	Number of particles per subswarm	10–40	20
K	Number of subswarms	2–6	4
Tmax	Maximum number of iterations	50–300	150
m	Fuzzy membership exponent	1.5–3.0	2.0
α\	Type-II uncertainty control parameter	0–1	0.5
SE	Structuring element size (morphology)	$3 \times 3 - 7 \times 7$	5×5
ϵ	Small constant for numerical stability	10^{-6}	10^{-6}

Table 7
Sensitivity analysis of superpixel parameters

SE size	Threshold	Dice score	Jaccard index
3×3	0.45	0.882	0.793
5×5	0.45	0.901	0.821
7×7	0.45	0.894	0.813
5×5	0.40	0.897	0.818
5×5	0.50	0.899	0.819

Figure 8
A boxplot of Dice coefficients computed across all 198 hippocampus images



6. Limitations of the Study

Despite its promising performance, this study has several limitations that merit discussion. First, the experimental validation was carried out on a relatively small dataset comprising 10 microscopic hippocampus images. This constraint primarily stems from the challenges associated with acquiring high-quality microscopic data and obtaining reliable expert annotations, both of which are time-consuming and resource-intensive in biomedical imaging applications. Although the selected images demonstrate notable variability in cellular morphology, contrast, and noise levels, the limited sample size may reduce the statistical generalizability of the results to other imaging conditions, anatomical regions, or microscopy modalities.

Furthermore, the proposed SUFGSO framework requires the number of clusters to be defined in advance and involves multiple control parameters associated with superpixel generation, type-II fuzzy clustering, and GSO. While sensitivity analysis suggests that the framework is relatively stable under moderate parameter variations, full automation—such as automatic determination of the optimal cluster number and adaptive parameter tuning—has not yet been realized. Future work will therefore focus on evaluating the proposed approach on larger and more diverse datasets and on incorporating adaptive mechanisms to enable fully automated segmentation, thereby improving robustness and generalizability.

7. Conclusion

This study introduced a novel SUFGSO-based segmentation framework for microscopic biomedical image analysis by synergistically combining superpixel representation, type-II fuzzy clustering, and GSO. By operating at the region level and explicitly modeling uncertainty, the proposed approach achieves reliable and accurate segmentation without relying on annotated training data. Experimental evaluation on hippocampus microscopy images shows that the method consistently surpasses conventional clustering- and optimization-based segmentation techniques across multiple performance metrics, demonstrating strong resilience to noise, intensity non-uniformity, and complex cellular morphology. Notwithstanding these encouraging outcomes, certain limitations must be recognized. The experimental analysis was performed on a relatively small dataset comprising 10 hippocampus images, largely due to the inherent difficulty of obtaining high-quality microscopic data and dependable expert annotations. Although the dataset captures meaningful variability in structural and intensity characteristics, the limited sample size may constrain the statistical generalizability of the reported results. Moreover, the current implementation of the SUFGSO framework requires prior specification of the number of clusters and involves several control parameters associated with

superpixel formation, fuzzy clustering, and swarm-based optimization, which may necessitate limited user intervention.

From an application standpoint, the proposed segmentation framework holds significant potential for supporting biomedical research and clinical workflows, particularly as a preprocessing module for cellular structure analysis, region-of-interest extraction, and quantitative morphometric assessment. Its unsupervised design and transparent decision-making process make it well-suited to data-scarce environments and applications where interpretability is essential, such as microscopy-based research studies and computer-assisted pathology. Future research will focus on extensive validation using larger and more diverse biomedical image datasets, extending the framework to additional microscopy modalities, and developing adaptive strategies for automatic cluster estimation and parameter self-optimization. These enhancements are expected to further improve the robustness, scalability, and real-world applicability of the proposed SUFGSO framework for biomedical image analysis.

Acknowledgment

The authors would like to express their gratitude and thank the editors, anonymous reviewers, and referees for their valuable comments and suggestions, which are helpful in further improvement of this research work.

Ethical Statement

This study does not involve direct experimentation on animals. The microscopic images of rat hippocampal tissue used in this work were obtained from a public dataset [44]. As no live animals were handled or subjected to experimental procedures by the authors, formal ethical approval was not required. All data usage complied with the relevant guidelines and source permissions.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in the Center for Research in Biological Systems at https://www.crbs.ucsd.edu/crbs-projects/highlighted-research-projects?highlight=94.

Author Contribution Statement

Debasish Biswas: Methodology, Software, Formal analysis. **Shouvik Chakraborty:** Conceptualization, Writing – original draft, Supervision, Project administration. **Chinmoy Ghorai:** Validation, Writing – review & editing, Supervision, Resources.

References

[1] Melanthota, S. K., Gopal, D., Chakrabarti, S., Kashyap, A. A., Radhakrishnan, R., & Mazumder, N. (2022). Deep learning-based image processing in optical microscopy. *Biophysical Reviews*, *14*(2), 463–481. https://doi.org/10.1007/s12551-022-00949-3

[2] Abdou, M. A. (2022). Literature review: Efficient deep neural networks techniques for medical image analysis. *Neural Computing and Applications*, *34*(8), 5791–5812. https://doi.org/10.1007/s00521-022-06960-9

[3] Archit, A., Freckmann, L., Nair, S., Khalid, N., Hilt, P., Rajashekar, V., . . . , & Pape, C. (2025). Segment anything for microscopy. *Nature Methods*, *22*(3), 579–591. https://doi.org/10.1038/s41592-024-02580-4

[4] Ma, J., Hu, C., Zhou, P., Jin, F., Wang, X., & Huang, H. (2023). Review of image augmentation used in deep learning-based material microscopic image segmentation. *Applied Sciences*, *13*(11), 6478. https://doi.org/10.3390/app13116478

[5] Liu, R., Dai, W., Wu, T., Wang, M., Wan, S., & Liu, J. (2022). AIMIC: Deep learning for microscopic image classification. *Computer Methods and Programs in Biomedicine*, *226*, 107162. https://doi.org/10.1016/j.cmpb.2022.107162

[6] Ma, J., & Wang, B. (2023). Towards foundation models of biological image segmentation. *Nature Methods*, *20*(7), 953–955. https://doi.org/10.1038/s41592-023-01885-0

[7] Malciu, A. M., Lupu, M., & Voiculescu, V. M. (2022). Artificial intelligence-based approaches to reflectance confocal microscopy image analysis in dermatology. *Journal of Clinical Medicine*, *11*(2), 429. https://doi.org/10.3390/jcm11020429

[8] Zhao, L., Ma, J., Shao, Y., Jia, C., Zhao, J., & Yuan, H. (2022). MM-UNet: A multimodality brain tumor segmentation network in MRI images. *Frontiers in Oncology*, *12*, 950706. https://doi.org/10.3389/fonc.2022.950706

[9] Zhang, Y., & Yan, L. (2022). A fast face recognition based on image gradient compensation for feature description. *Multimedia Tools and Applications*, *81*(18), 26015–26034. https://doi.org/10.1007/s11042-022-12804-4

[10] Shanthi, P. B., & Hareesha, K. S. (2022). Automated detection and classification of cervical cancer using pap smear microscopic images: A comprehensive review and future perspectives. *Engineered Science*, *19*(4), 20–41. http://dx.doi.org/10.30919/es8d633

[11] Liu, S., Lin, Y., Liu, D., Wang, P., Zhou, B., & Si, F. (2025). Frequency-enhanced lightweight vision Mamba network for medical image segmentation. *IEEE Transactions on Instrumentation and Measurement*, *74*, 1–12. https://doi.org/10.1109/TIM.2025.3527526

[12] Bargagna, F., de Santi, L. A., Martini, N., Genovesi, D., Favilli, B., Vergaro, G., . . . , & Santarelli, M. F. (2023). Bayesian convolutional neural networks in medical imaging classification: A promising solution for deep learning limits in data scarcity scenarios. *Journal of Digital Imaging*, *36*(6), 2567–2577. https://doi.org/10.1007/s10278-023-00897-8

[13] Haq, I., He, L., Li, J., Chen, B., Yang, X., & Li, Y. (2025). Integrated microfluidics and deep learning for detection and interpretation of colon cancer-associated ascitic cells using bright-field microscopy. *Physica Scripta*, *100*(8), 085005. https://doi.org/10.1088/1402-4896/adedce

[14] Muthiah-Nakarajan, V., & Noel, M. M. (2016). Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion. *Applied Soft Computing*, *38*, 771–787. https://doi.org/10.1016/j.asoc.2015.10.034

[15] Escorcia-Gutierrez, J., Gamarra, M., Ariza-Colpas, P. P., Roncallo, G. B., Leal, N., Soto-Diaz, R., & Mansour, R. F. (2022). Galactic swarm optimization with deep transfer learning driven colorectal cancer classification for image guided intervention. *Computers and Electrical Engineering*, *104*, 108462. https://doi.org/10.1016/j.compeleceng.2022.108462

[16] Karthick, S., & Gomathi, N. (2022). Galactic swarm-improved whale optimization algorithm-based resource management in Internet of things. *International Journal of Communication Systems*, *35*(3), e5006. https://doi.org/10.1002/dac.5006

[17] Pandey, D., & Pandey, B. K. (2022). An efficient deep neural network with adaptive galactic swarm optimization for complex image text extraction. In V. Yadav, A. K. Dubey, H. P. Singh, G. Dubey, & E. Suryani (Eds.), *Process mining techniques for pattern recognition: Concepts, theory, and practice* (pp. 121–137). CRC Press. https://doi.org/10.1201/9781003169550-10

[18] Uymaz, O., Turkoglu, B., Kaya, E., & Asuroglu, T. (2024). A novel diversity guided galactic swarm optimization with feedback mechanism. *IEEE Access*, *12*, 108154–108175. https://doi.org/10.1109/ACCESS.2024.3438104

[19] Wang, J. (2024). Fuzzy logic with galactic swarm optimization based higher vocational english teaching system in cloud computing. In *2024 International Conference on Data Science and Network Security*, 1–4. https://doi.org/10.1109/ICDSNS62112.2024.10691307

[20] Patgiri, C., & Ganguly, A. (2021). Adaptive thresholding technique based classification of red blood cell and sickle cell using Naïve Bayes Classifier and K-nearest neighbor classifier. *Biomedical Signal Processing and Control*, *68*, 102745. https://doi.org/10.1016/j.bspc.2021.102745

[21] Sharma, R., & Ravinder, M. (2023). Remote sensing image segmentation using feature based fusion on FCM clustering algorithm. *Complex & Intelligent Systems*, *9*(6), 7423–7437. https://doi.org/10.1007/s40747-023-01129-w

[22] Lin, Y.-C., & Chen, T.-C. T. (2022). Type-II fuzzy approach with explainable artificial intelligence for nature-based leisure travel destination selection amid the COVID-19 pandemic. *Digital Health*, *8*, 205520762211063. https://doi.org/10.1177/20552076221106322

[23] Dahiya, S., & Gosain, A. (2023). A novel type-II intuitionistic fuzzy clustering algorithm for mammograms segmentation. *Journal of Ambient Intelligence and Humanized Computing*, *14*(4), 3793–3808. https://doi.org/10.1007/s12652-022-04022-5

[24] Köklü, A., Güven, Y., & Kumbasar, T. (2024). Enhancing Interval Type-2 Fuzzy Logic Systems: Learning for precision and prediction intervals. In *2024 IEEE International Conference on Fuzzy Systems*, 1–7. https://doi.org/10.1109/FUZZ-IEEE60900.2024.10612062

[25] Atacak, İ., Çıtlak, O., & Doğru, İ. A. (2023). Application of interval type-2 fuzzy logic and type-1 fuzzy logic-based approaches to social networks for spam detection with combined feature capabilities. *PeerJ Computer Science*, *9*, e1316. https://doi.org/10.7717/peerj-cs.1316

[26] Cuevas, F., Castillo, O., & Cortes, P. (2022). Optimal setting of membership functions for interval type-2 fuzzy tracking controllers using a shark smell metaheuristic algorithm. *International Journal of Fuzzy Systems*, *24*(2), 799–822. https://doi.org/10.1007/s40815-021-01136-4

[27] Guerrero, M., Valdez, F., & Castillo, O. (2022). Comparative study between type-1 and interval type-2 fuzzy systems in parameter adaptation for the cuckoo search algorithm. *Symmetry*, *14*(11), 2289. https://doi.org/10.3390/sym14112289

[28] Alsebai, M. D., & Pedapenki, K. K. (2024). Transformation of type I fuzzy logic into interval type II fuzzy logic-based unit voltage template with controlled load. *Periodica Polytechnica Electrical Engineering and Computer Science*, *68*(1), 37–53. https://doi.org/10.3311/PPee.22465

[29] Özlü, Ş., & Karaaslan, F. (2022). Correlation coefficient of T-spherical type-2 hesitant fuzzy sets and their applications in clustering analysis. *Journal of Ambient Intelligence and Humanized Computing*, *13*(1), 329–357. https://doi.org/10.1007/s12652-021-02904-8

[30] Zhao, X., & Xu, W. (2024). NFMPAtt-Unet: Neighborhood fuzzy c-means multi-scale pyramid hybrid attention unet for medical image segmentation. *Neural Networks*, *178*, 106489. https://doi.org/10.1016/j.neunet.2024.106489

[31] Sasmal, B., Das, A., Dhal, K. G., & Ray, S. (2023). Aquila-particle swarm based cooperative search optimizer with superpixel techniques for epithelial layer segmentation. *Applied Soft Computing*, *149*, 110947. https://doi.org/10.1016/j.asoc.2023.110947

[32] Huang, Q., Zhou, Y., Tao, L., Yu, W., Zhang, Y., Huo, L., & He, Z. (2021). A Chan-Vese model based on the Markov chain for unsupervised medical image segmentation. *Tsinghua Science and Technology*, *26*(6), 833–844. https://doi.org/10.26599/TST.2020.9010042

[33] Barcelos, I. B., Belem, F. D. C., Joao, L. D. M., Patrocínio Jr, Z. K. D., Falcao, A. X., & Guimarães, S. J. F. (2024). A comprehensive review and new taxonomy on superpixel segmentation. *ACM Computing Surveys*, *56*(8), 1–39. https://doi.org/10.1145/3652509

[34] Xu, S., Wei, S., Ruan, T., & Zhao, Y. (2023). ESNet: An efficient framework for superpixel segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, *34*(7), 5389–5399. https://doi.org/10.1109/TCSVT.2023.3347402

[35] Prakash, J., & Kumar, B. V. (2023). An extensive survey on superpixel segmentation: A research perspective. *Archives of Computational Methods in Engineering*, *30*(6), 3749–3767. https://doi.org/10.1007/s11831-023-09919-8

[36] Pan, Y. J., Wen, C., Zhao, X. L., Ding, M., Lin, J., & Fan, Y. R. (2023). Irregular tensor representation for superpixel-guided hyperspectral image denoising. *IEEE Geoscience and Remote Sensing Letters*, *20*, 1–5. https://doi.org/10.1109/LGRS.2023.3329936

[37] Hammoudi, K., Cabani, A., Slika, B., Benhabiles, H., Dornaika, F., & Melkemi, M. (2022). Superpixelgridmasks data augmentation: Application to precision health and other real-world data. *Journal of Healthcare Informatics Research*, *6*(4), 442–460. https://doi.org/10.1007/s41666-022-00122-1

[38] Zeng, X., Wu, W., Tian, G., Li, F., & Liu, Y. (2021). Deep superpixel convolutional network for image recognition. *IEEE Signal Processing Letters*, *28*, 922–926. https://doi.org/10.1109/LSP.2021.3075605

[39] Ng, T. C., Choy, S. K., Lam, S. Y., & Yu, K. W. (2023). Fuzzy superpixel-based image segmentation. *Pattern Recognition*, *134*, 109045. https://doi.org/10.1016/j.patcog.2022.109045

[40] Sasmal, B., & Dhal, K. G. (2023). A survey on the utilization of Superpixel image for clustering based image segmentation. *Multimedia Tools and Applications*, *82*(23), 35493–35555. https://doi.org/10.1007/s11042-023-14861-9

[41] Gauen, K., & Chan, S. (2024). Soft superpixel neighborhood attention. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 3712.

[42] Shekar, B. H., Shetty, P. R., Bhat, S. S., & Mestetsky, L. (2022). Complex gradient function based image descriptor: Shekar et al. *SN Computer Science*, *4*(1), 42. https://doi.org/10.1007/s42979-022-01436-y

[43] Sun, R., Lei, T., Chen, Q., Wang, Z., Du, X., Zhao, W., & Nandi, A. K. (2022). Survey of image edge detection. *Frontiers in Signal Processing*, *2*, 826967. https://doi.org/10.3389/frsip.2022.826967

[44] The Cell Image Library. (2022). https://www.cellimagelibrary.org/home

[45] Yadav, S., & Vishwakarma, V. P. (2024). Robust face recognition using quaternion interval type II fuzzy logic–based feature extraction on colour images. *Medical & Biological Engineering & Computing*, *62*(5), 1503–1518. https://doi.org/10.1007/s11517-024-03015-0

[46] Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 224–227. https://doi.org/10.1109/TPAMI.1979.4766909 *PAMI-1*(2)

[47] Xie, X. L., & Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(8), 841–847. https://doi.org/10.1109/34.85677

[48] Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, *4*(1), 95–104. https://doi.org/10.1080/01969727408546059

[49] Pal, S. K., Ghosh, A., & Shankar, B. U. (2000). Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation. *International Journal of Remote Sensing*, *21*(11), 2269–2300. https://doi.org/10.1080/01431160050029567

[50] Blum, C. (2005). Beam-ACO—hybridizing ant colony optimization with beam search: An application to open shop scheduling. *Computers & Operations Research*, *32*(6), 1565–1591. https://doi.org/10.1016/j.cor.2003.11.018

[51] Taherkhani, M., & Safabakhsh, R. (2016). A novel stability-based adaptive inertia weight for particle swarm optimization. *Applied Soft Computing*, *38*, 281–295. https://doi.org/10.1016/j.asoc.2015.10.004

[52] Kadri, R. L., & Boctor, F. F. (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*, *265*(2), 454–462. https://doi.org/10.1016/j.ejor.2017.07.027

[53] Chakraborty, S., Chatterjee, S., Dey, N., Ashour, A. S., Ashour, A. S., Shi, F., & Mali, K. (2017). Modified cuckoo search algorithm in microscopic image segmentation of hippocampus. *Microscopy Research and Technique*, *80*(10), 1051–1072. https://doi.org/10.1002/jemt.22900

How to Cite: Biswas, D., Chakraborty, S., & Ghorai, C. (2026). Fuzzy Galactic Swarm Optimization Coupled with Superpixel-Based Microscopic Image Segmentation. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62026336