



An Optimized Clustering Approach using Tree Seed Algorithm for the Brain MRI Images Segmentation

Ghazi Boumediene Ghaouti^[1], Samia Benyahia^[2], Boudjelal Meftah^[3]

^[1] Computer Science Department, Faculty of Exact Sciences, Mustapha Stambouli University, Mascara, Algeria

^[2] Higher School of Management and Digital Economy, Tipaza, Algeria

^[3] LRSBG Laboratory, Mustapha Stambouli University, Mascara, Algeria

Abstract Clustering algorithms are widely used for segmenting medical images. However, these techniques can be challenging to perform, especially when working with magnetic resonance images (MRI) of the brain. The complexity of the brain tissue's anatomical structure, the inhomogeneity of the pixel intensity in these images, and the effects of partial volume and noise can make clustering difficult, leading the algorithm to fall into local minima. To address this issue, it is recommended to improve clustering algorithms by using optimization techniques to achieve better results. In this study, we propose a developed clustering algorithm and optimize it using a tree seed algorithm (TSA) for segmenting brain MRI images. The algorithms are tested on real brain image datasets, and the experimental results show that our proposed and optimized methods yield satisfactory results according to the Davies-Bouldin Index (DBI), compared to the fuzzy c-mean (FCM) algorithm.

Keywords: Edge detection, Segmentation, Brain MRI image, Tree seed algorithm, Metrics.

1 Introduction

Medical imaging technology is currently experiencing a major revolution, as it plays a crucial role in diagnosing human diseases. In recent years, there has been a notable focus on creating machines and algorithms to simplify the acquisition and analysis of medical images. This advancement allows doctors to make more precise and informed choices concerning the treatment of illnesses. An instance of this is the utilization of MRI to detect and assess the magnitude of brain tumors, a crucial element in treating them effectively. In fact, when brain tumors are detected early, 90% of cases can be successfully treated, including medulloblastoma and lymphoma [1].

To accurately determine the size and extent of tumors, medical images must be segmented to separate the affected regions from healthy ones. This process requires a high degree of precision to avoid losing essential data. Several techniques are available for achieving this goal, such as thresholding methods, clustering methods, edge detection, and machine learning. Clustering methods, for instance, divide the pixels of an image into different groups or clusters based on the similarity and dissimilarity between elements measured.

Clustering algorithms have been widely used in medical image analysis for tasks such as segmentation, classification, and feature extraction. There are several clustering algorithms commonly utilized in medical image analysis, including:

- K-means clustering [2]: K-means is widely used in medical image segmentation for tasks such as tissue classification, tumor segmentation, and brain image analysis. It has been shown to be effective in identifying anatomical regions in medical images.
- Fuzzy C-means (FCM) clustering [3]: FCM is a soft clustering algorithm that assigns a degree of membership to each data point for each cluster, rather than a hard assignment. This algorithm has been widely used in medical image segmentation tasks where a pixel can belong to multiple tissue types.

- Hierarchical clustering [4]: Hierarchical clustering has been used in medical image analysis for tasks such as segmentation and feature extraction. It can provide a hierarchical structure of the data, which can be useful for identifying different tissue types or regions of interest.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [5]: DBSCAN has been used in medical image analysis for tasks such as tumor segmentation and lesion detection. It can identify clusters of points that are close to each other in dense regions of the image.
- Expectation-Maximization (EM) clustering [6]: EM has been used in medical image analysis for tasks such as tissue segmentation and registration. It can model the data as a mixture of probability distributions and estimate the parameters of the distributions using the EM algorithm.

The choice of clustering algorithm depends on the specific task at hand and the characteristics of the medical image being analyzed.

However, when it comes to medical image clustering, these algorithms face two important challenges:

- Firstly, medical images can be complex in nature, and they may contain overlapping regions of interest. Additionally, clustering algorithms rely on local search methods to find the centers of clusters. As a result, there is a significant risk of encountering local minima problems, which may cause the algorithm to choose inappropriate centers that result in suboptimal clustering.
- Secondly, in most clustering techniques, final cluster values are determined using predefined starting centers. This can lead to inaccurate segmentation results, which may not fully capture the underlying characteristics of the medical images being analyzed.

These challenges highlight the need for further research and development of clustering algorithms that can more effectively address the specific requirements of medical image analysis.

To address the first challenge, unsupervised clustering algorithms can be optimized using various techniques, including biologically inspired algorithms. These algorithms simulate natural phenomena or behaviors of living organisms to generate solutions for problems in computer or technological domains. Several biologically-inspired algorithms have been employed for medical image clustering. These algorithms include the Firefly algorithm [7], Bat algorithm [8], Tree Seed Algorithm (TSA) [9], Particle Swarm Optimization [10], and Gray Wolf algorithm [11].

The firefly optimization-based enhanced fuzzy clustering method has been used in CT/MR image segmentation [12]. This paper presents a novel approach for medical image segmentation using a combination of Firefly Optimization (FO) and Fuzzy C-Means (FCM) clustering. The proposed method optimizes clustering parameters using FO to improve the accuracy and robustness of FCM. FO is used to determine the optimal values of FCM parameters, including the number of clusters and the fuzziness coefficient. FCM is then used to segment the image into different regions. The proposed method was evaluated on CT and MR images.

Another study that utilized the firefly algorithm and FCM for clustering brain tissue in magnetic resonance images (MRI) was discussed in [13]. This paper proposes a new approach to segmenting brain tissues in magnetic resonance images using a combination of Chaotic Firefly Algorithm (CFA) and Fuzzy C-Means (FCM) clustering. CFA is used to determine the optimal clustering parameters, including the number of clusters and fuzziness coefficient. FCM is then used to segment the brain tissues into different regions. The proposed method was evaluated on real MRI data.

In a study by Ben Aichouche et al [14], the authors proposed the use of an improved Spatial Fuzzy C-Means (SFCM) algorithm for image segmentation, which incorporates Particle Swarm Optimization (PSO) initialization, Mahalanobis distance, and post-segmentation correction. The proposed method aims to improve the accuracy and robustness of SFCM clustering by optimizing the initialization of the clustering process, distance metric, and post-processing steps. The PSO algorithm is used to initialize the clustering parameters, and the Mahalanobis distance is employed as a distance metric to handle the spatial information of the image. Additionally, a post-segmentation correction technique is applied to refine the segmentation results.

In [15], a population-based hybrid FCM-PSO algorithm for clustering analysis and segmentation of brain images is proposed. This approach combines the strengths of PSO and FCM by using PSO to identify the optimal centers of clusters, which are then used by FCM to perform the clustering analysis and segmentation of the brain images. This population-based hybrid algorithm has the potential to overcome some of the limitations of traditional FCM and PSO algorithms, and provide more accurate and efficient clustering results.

In [16], Romana et al proposed a hybrid method using the k-means clustering algorithm and the firefly algorithm for brain image clustering. This method utilizes the firefly algorithm to find the optimal cluster center and prevent local minima in the k-means algorithm. The proposed hybrid approach was tested on different brain MRI datasets, and the results showed improved performance compared to using the k-means algorithm alone.

Hong Zhu et al [17] proposed a hybrid method using fruit fly optimization and peak density clustering (PDC). This method improves upon the original PDC algorithm by automatically selecting its parameters to improve the quality of segmentation.

Khrissi et al [18] proposed a hybrid method by merging a genetic algorithm and a k-means clustering algorithm. This approach uses k-means and Sobel filter to extract the image region. Subsequently, the genetic algorithm optimizes the final classes by merging two regions if they satisfy the homogeneity criterion.

1.1 Motivation

The questions we need to ask through this paper are:

- What is the most appropriate clustering algorithm to effectively segment medical images, and how can we improve it using biologically inspired algorithms?
- How to determine the ideal number of clusters for which the image should be segmented?

1.2 Contributions

Our research contributes to the clustering of brain images through two approaches:

- The first approach proposes a new clustering algorithm for segmenting brain MRI images. Our algorithm achieves high-quality segmentation compared to other clustering algorithms. It can segment the images into the appropriate number of clusters without any human intervention.
- In the second approach, we use a biologically inspired algorithm, the TSA, to further improve our new algorithm. The TSA is used to select the optimal center of each cluster and to avoid the local minima problem. Additionally, we propose a method to determine the appropriate number of clusters required to segment the brain MRI images.

This paper is organized as follows. In the first section, we will introduce the problem and review the existing solutions proposed in the literature. Section 2 provides the details of our proposed clustering algorithm for segmenting brain MRI images. In Section 3, we present an improved approach using the Tree Seed Algorithm to further enhance the performance of our algorithm. An experimental study of both approaches is described in Section 4. Finally, we conclude with remarks and suggestions for future work in Section 5.

2 Proposed Model for Edge Detection MRI images

K-means clustering and FCM are among the most commonly used clustering methods. These algorithms aim to minimize objective functions to assign clusters to centers. However, their performance is often limited by some drawbacks, including:

- The final cluster values are usually obtained using pre-specified initial centers in most clustering techniques. This can lead to imprecise clustering results, especially when the optimal number of clusters for complex or large datasets such as medical images is unknown.
- Many of these algorithms employ local search methods to determine the location of centers, which can result in getting stuck in local minima.

In order to overcome these two issues, we propose a new clustering algorithm that automatically determines the appropriate number of clusters for clustering the data. At the beginning of the algorithm, we initialize a default number of cluster centers, and during the clustering process, the number and position of the cluster centers are automatically adjusted.

Our approach entails a non-linear optimization strategy aimed at identifying the optimal number of cluster centers for data segmentation, irrespective of their type and precision. Initially, we randomly initialize the cluster centers and threshold value, with their numbers and positions changing as the algorithm progresses. Once we obtain the final list of cluster centers, we assign each data point to its nearest center using the Euclidean distance. Since the centers are randomly initialized for every execution and our algorithm leverages local search for center positioning, it is non-deterministic. Therefore, different initializations used across various experiments will yield disparate results.

We describe our algorithm as follows:

Step 1: We randomly determine the initial cluster's center and threshold value ϑ .

Step 2: For each data point, we compute the distance between it and the center of each cluster using the Euclidean distance formula:

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \tag{1}$$

where y_i is the coordinate of the element in the data and x_i the center's coordinate.

Step3: Based on the minimum distance obtained above:

- If a data point is close to only one cluster center, we assign it to that cluster.
- If a data point is close to at least two cluster centers, we have two options:
 - a. If the minimum distance is less than or equal to the threshold value, we remove the closest centers from the list of centers and consider the data point as a new center. We go back to step 2.
 - b. If the minimum distance exceeds the threshold value, we consider the data point as a new cluster center and return to step 2.

We summarize our algorithm in the following flowchart:

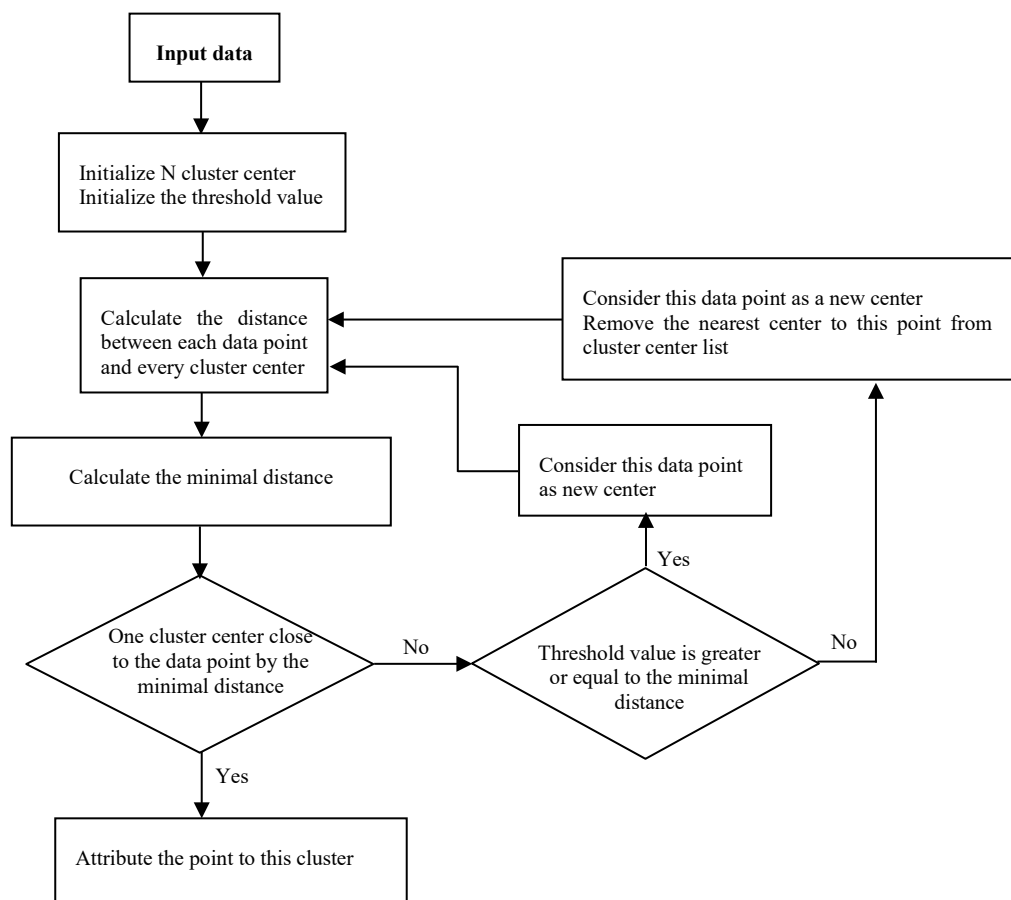


Figure 1. Flowchart of the new clustering algorithm.

The pseudo code for our algorithm is presented as follows:

Algorithm 1:

```

1  Input:
2  Data: Dataset = {x1 , x2, . . . , xN }
3  N : Number of the cluster
4  Nr : Number of the nearest cluster
5  Mindist: Minimal distance
6  Th: threshold value
    
```

```

7   i: Data point
8   Result: clustered data
9   Begin
10  Initialize N random cluster center from the data
11  Initialize the threshold value Th
12  While (i in Data) do
13  Calculate Mindist between I and each center cluster
14  Determine Nr values
15  If (Nr =1) then
16  Assign the data point i to the nearest cluster center
17  Else if (Nr >1 and Mindist <Th)
18  Remove the nearest cluster center from N
19  Consider i as new cluster center
20  Return to the beginning
21  Else if (Nr >1 and Mindist >=Th) then
22  Consider i as new center cluster
23  Return to the beginning
24  End if
25  End while
26  End

```

3 Improved approach using tree-seed algorithm

3.1 Presentation of the tree seed algorithm

Tree Seed Algorithm (TSA) is a type of optimization algorithm inspired by the process of tree growth in nature proposed by Kiran [19]. It is a metaheuristic optimization algorithm that simulates the process of seed germination, growth, and development in plants to generate new solutions to optimization problems. TSA has been applied to a wide range of optimization problems in various fields, including engineering, finance, and biology. Its effectiveness is due in part to its ability to balance exploration and exploitation of the solution space, and to generate diverse and high-quality solutions that are often difficult to find through other optimization methods.

The TSA algorithm works by starting with a small number of "seed" solutions, which are then expanded and improved through a process of iterative growth and selection. Each seed solution represents a potential solution to the optimization problem, and is evaluated based on a fitness function that measures its performance or quality. The algorithm then proceeds to iteratively grow and modify the seed solutions, adding new "branches" to the tree by making small random changes to the existing solutions. These modifications are made with the goal of improving the fitness of the solution, and are guided by various factors such as the similarity between the current and new solutions, the distance from the current solution to the optimal solution, and the degree of diversity among the different solutions.

As the TSA algorithm continues to grow and modify the tree, it gradually converges towards a set of optimal solutions that represent the best possible solutions to the optimization problem. The algorithm may also incorporate techniques such as pruning and mutation to remove less promising branches and promote exploration of new areas of the solution space.

The number of seeds for each tree is chosen randomly. Although it should be at least one, it is recommended that it be between 10% and 25% of the total number of trees. These seeds are created based on search trend values, which are used to select the best tree or a random tree during seed development, and using two seed production equations:

$$S_{k,j} = T_{i,j} + a_{i,j} * (B_j - T_{i,j}) \quad (2)$$

$$S_{k,j} = T_{i,j} + a_{i,j} * (T_{i,j} - T_{r,j}) \quad (3)$$

Where $S_{k,j}$ is the j^{th} dimension of the k^{th} seed from the I^{th} tree. $T_{i,j}$ denotes the tree's J^{th} dimension. B_j is the j^{th} dimension of the best tree placement that has been discovered. $T_{r,j}$ denotes the J^{th} dimension of the r^{th} tree, which is chosen randomly from the population. I and J are the scaling factor, which are chosen randomly in the interval $[-1,1]$.

During the search process, selection is made among the seeds of each tree, and the best seed is identified. If the quality of the best seed is superior to the tree, the tree withers and the best seed takes its place. The TSA is run until all stopping conditions have been met.

3.2 Improved algorithm with TSA

As mentioned earlier, a significant challenge with our clustering algorithm, as well as similar approaches, is their dependence on local search to position cluster centers. This can lead to inconsistent cluster centers and varying clustering results even when using identical parameters for multiple runs of the algorithm. To overcome this limitation, we improved our algorithm by using the tree seed algorithm (TSA).

By incorporating TSA, our enhanced algorithm combines the advantages of both methods, effectively avoiding the local minima and producing optimized clustering results.

To improve brain MRI clustering, there are some parameters in the TSA clustering algorithm that may need to be updated to better fit the nature of the images, as follow.

1. Generate randomly N pixels representing the initial trees in the image.
2. Initialize the number of iterations.
3. Initialize Search Tendency (ST) between 0 and 1
4. Generate initial seeds where the number of seeds for each tree is between 10% and 25% of the initial tree population and each seed represents a pixel in our image.
5. Update the intensity of these seeds using the following equations:

$$S_{i,j} = \begin{cases} PT_j + (R - 0.5) * 2 & \text{where random values} < ST \\ PT_j + (PT_j - R) * (R - 0.5) * 2 & \text{otherwise} \end{cases} \quad (4)$$

Where $S_{i,j}$ is the seed j of the tree I and PT_j is the parent tree of the seed I and R presents a random number between 0 and 1.

5. For each tree, calculate the fitness of its seeds defined by the distance between that seed and the nearest tree in the tree's population.
6. If the fitness of a seed is greater than the fitness of the parent tree, we remove the parent from the list of trees and replace it with this seed.
7. Repeat these steps to obtain the terminal condition.

We can summarize the new improved algorithm in the flowchart below:

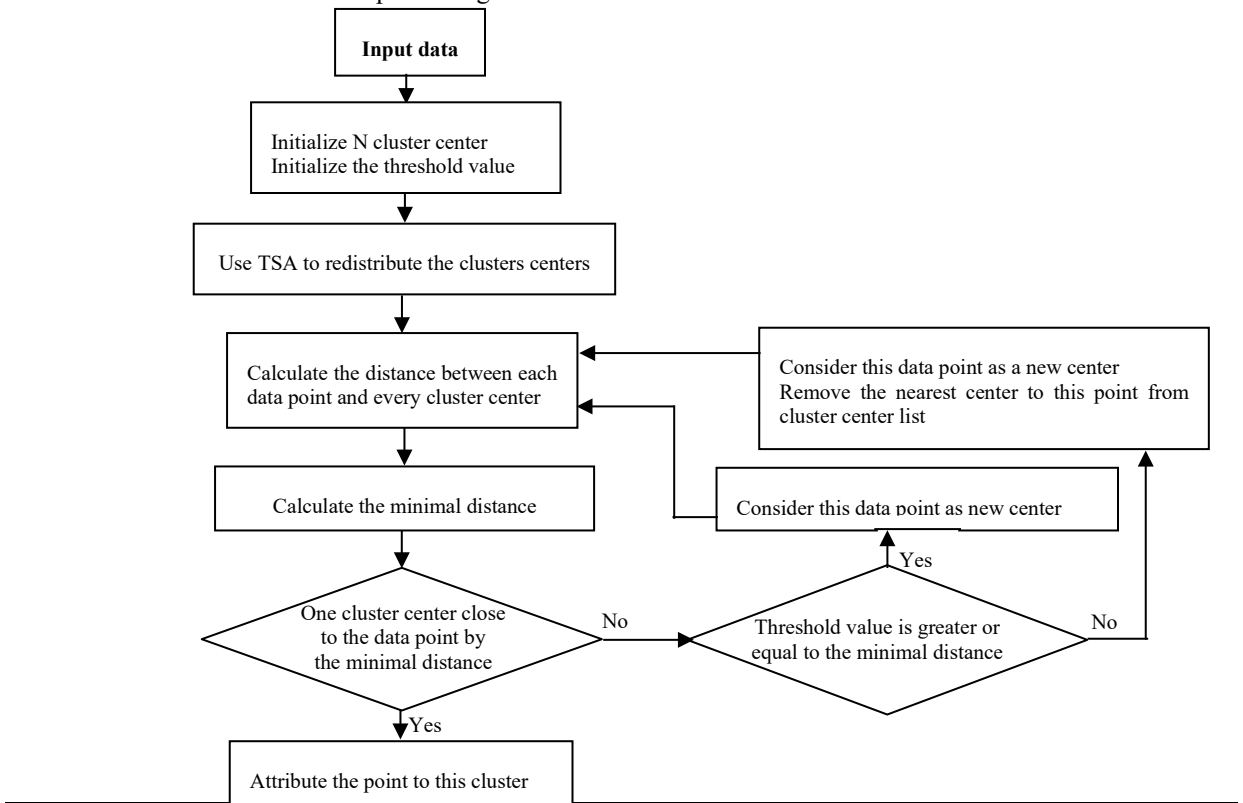


Figure 2: Flowchart of the improved algorithm with TSA

4 Results and discussion

In this section, we evaluate the efficacy of our proposed algorithm and its improved version with TSA in segmenting brain MRI. We conducted our experiment on a dataset consisting of 253 MRI images categorized into two classes: healthy and cancer [18]. All experiments are conducted on a PC with an Intel Core i3-2310M 2.10 GHz processor and 8 GB of RAM, using Python 3.6.

4.1 Metrics used in evaluation

Clustering validation metrics are quantitative measures that are used to assess the quality of clustering results [20]. They provide a way to evaluate how well a clustering algorithm is able to group similar data points together and separate dissimilar data points into different clusters. Clustering validation metrics can help to compare different clustering algorithms, select the best algorithm or parameters for a given dataset, and provide insights into the structure of the data [21, 22].

Clustering validation metrics can be broadly categorized into two types: external and internal metrics. External metrics evaluate the quality of clustering output based on a known ground truth, such as a set of labelled data points, while internal metrics evaluate the quality of clustering output based solely on the intrinsic properties of the data, without reference to external information. Examples of external clustering validation metrics include the Adjusted Rand Index (ARI), Fowlkes-Mallows Index (FMI), and Normalized Mutual Information (NMI). Examples of internal clustering validation metrics include Silhouette Coefficient, Davies-Bouldin Index (DBI), and Calinski-Harabasz Index (CHI).

4.1.1 Davies–Bouldin index

The Davies–Bouldin Index (DBI) is a clustering validation metric used to evaluate the quality of a clustering solution [23]. The index measures the average similarity between each cluster and its most similar cluster, taking into account the size of the clusters. The smaller the index, the better the clustering solution is considered to be.

To compute the DBI, for each cluster, the index calculates the average distance between each point in the cluster and the centroid of the cluster. Then, for each cluster, the index finds the cluster that is most similar in terms of average distance to the centroid. Finally, the index calculates the ratio of the sum of the average distances within each cluster and the distance between the centroids of the two clusters. The process is repeated for all clusters, and the average of the ratios is taken as the final index.

The DBI is defined as follows:

$$DB = \frac{1}{k} \sum_{i=1}^N \max_{i \neq j} R_{i,j} \quad (5)$$

The similarity between two cluster I and J is denoted by $R_{i,j}$ and is defined as:

$$R_{i,j} = \frac{S_i + S_j}{d_{i,j}} \quad (6)$$

Where S_i is the average between each point of cluster I and the centroid of this cluster and $d_{i,j}$ is the distance between cluster centroid I and j.

4.1.2 Silhouette coefficient

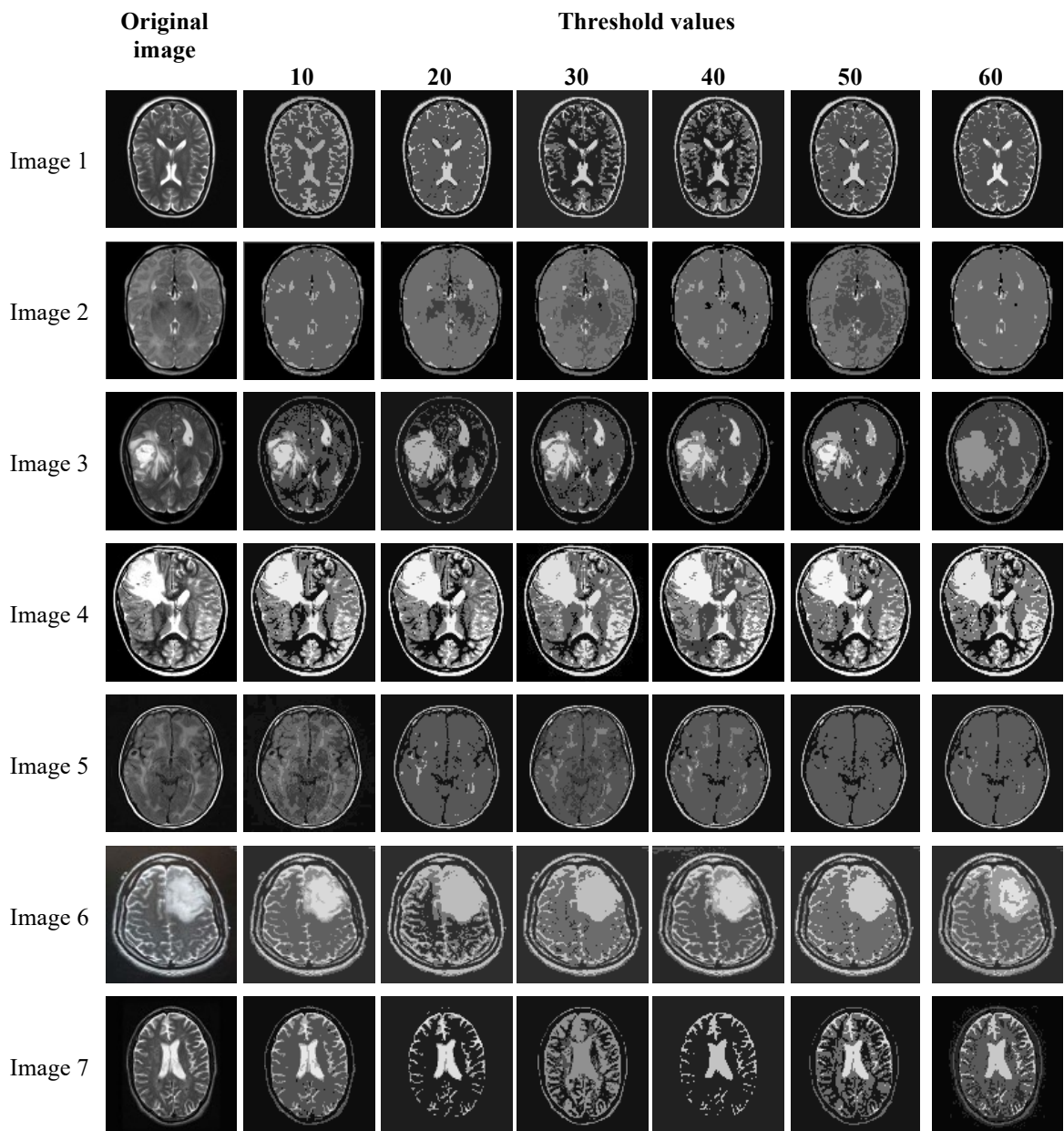
The Silhouette Coefficient is a clustering validation metric used to evaluate the quality of a clustering solution [24]. The coefficient measures how similar an object is to its own cluster compared to other clusters. It takes into account both the average distance between an object and all other points in the same cluster, as well as the average distance between the object and all other points in the nearest neighboring cluster.

To compute the Silhouette Coefficient for a given object, the coefficient first calculates the average distance between the object and all other points within the same cluster. It then calculates the average distance between the object and all other points within the nearest neighboring cluster. The Silhouette Coefficient for the object is defined as the difference between these two values, divided by the maximum of the two values.

The Silhouette Coefficient ranges from -1 to 1, where a value of 1 indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. A value of 0 indicates that the object is equally similar to its own cluster and neighboring clusters, and a value of -1 indicates that the object is poorly matched to its own cluster and well matched to neighboring clusters.

The Silhouette Coefficient (SC) index is defined as follows:

$$SC = \frac{a - b}{\max(b, a)} \tag{7}$$



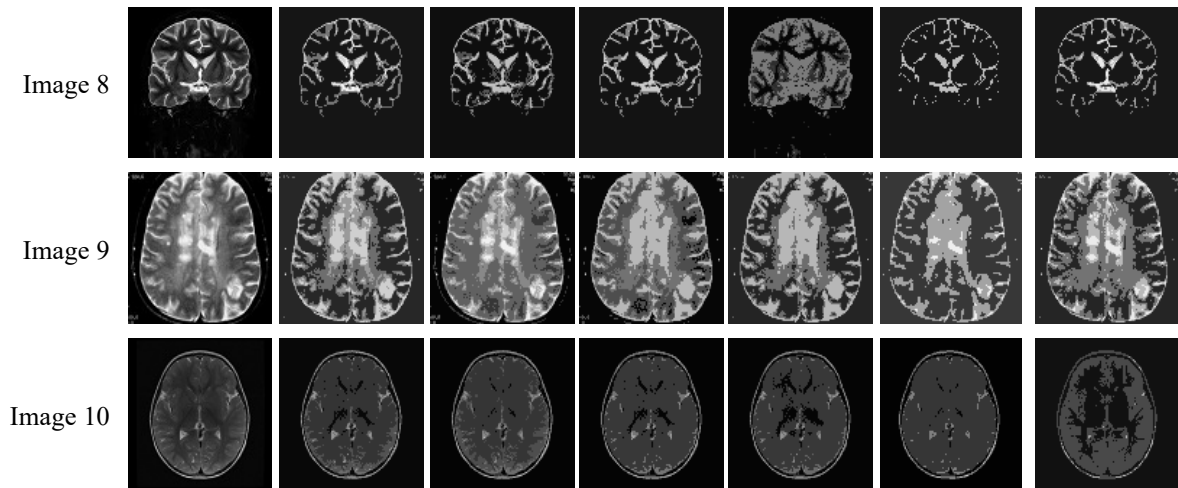


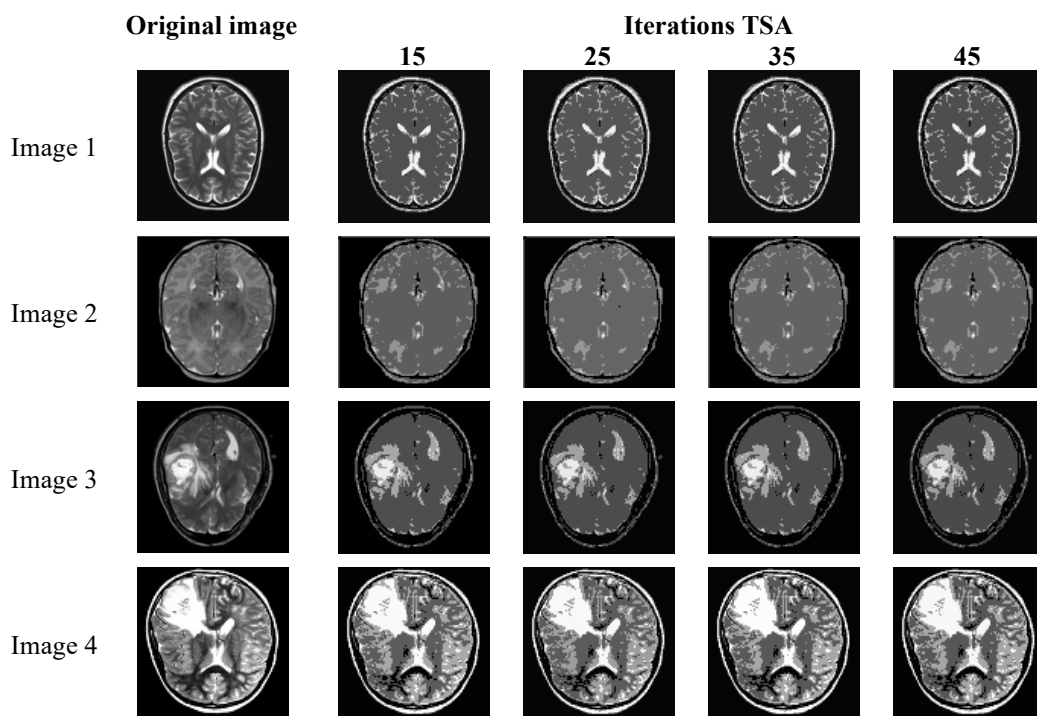
Figure 3. Segmentation results of the proposed algorithm.

Table 1: Performance Results on Brain MRI for the Proposed Algorithm with Different Metrics.

Original image	Index	Threshold values					
		10	20	30	40	50	60
Image 1	DBI	0.51	0.47	0.49	0.48	0.41	0.38
	SC	0.77	0.714	0.55	0.56	0.77	0.80
Image 2	DBI	0.51	0.46	0.460	0.45	0.47	0.38
	SC	0.68	0.68	0.65	0.73	0.67	0.74
Image 3	DBI	0.48	0.52	0.438	0.39	0.41	0.45
	SC	0.62	0.527	0.713	0.78	0.76	0.76
Image 4	DBI	0.45	0.44	0.70	0.46	0.42	0.42
	SC	0.613	0.60	0.52	0.71	0.69	0.68
Image 5	DBI	0.66	0.45	0.50	0.44	0.35	0.41
	SC	0.389	0.76	0.670	0.68	0.76	0.76
Image 6	DBI	0.50	0.49	0.45	0.49	0.46	0.44
	SC	0.609	0.524	0.63	0.58	0.64	0.57
Image 7	DBI	0.286	0.43	0.64	0.41	0.61	0.61
	SC	0.694	0.58	0.47	0.69	0.62	0.62
Image 8	DBI	0.38	0.49	0.46	0.63	0.44	0.44
	SC	0.69	0.63	0.72	0.58	0.72	0.72
Image 9	DBI	0.57	0.47	0.47	0.58	0.53	0.53
	SC	0.39	0.58	0.60	0.41	0.45	0.47
Image 10	DBI	0.46	0.45	0.44	0.44	0.36	0.44
	SC	0.72	0.74	0.75	0.72	0.76	0.61
Image 11	DBI	0.54	0.76	0.52	0.47	0.44	0.41
	SC	0.55	0.32	0.45	0.60	0.71	0.710
Image 12	DBI	0.45	0.56	0.47	0.42	0.37	0.57
	SC	0.78	0.73	0.69	0.790	0.74	0.795
Image 13	DBI	0.49	0.44	0.43	0.45	0.46	0.41
	SC	0.69	0.67	0.67	0.68	0.59	0.74
Image 14	DBI	0.53	0.56	0.35	0.32	0.50	0.65
	SC	0.46	0.62	0.68	0.72	0.58	0.30
Image 15	DBI	0.43	0.43	0.51	0.55	0.42	0.51
	SC	0.67	0.672	0.49	0.37	0.64	0.52
Image 16	DBI	0.45	0.48	0.52	0.41	0.41	0.33
	SC	0.66	0.62	0.67	0.75	0.70	0.76
Image 17	DBI	0.56	0.44	0.50	0.49	0.36	0.55
	SC	0.57	0.71	0.44	0.49	0.72	0.66
Image 18	DBI	0.61	0.43	0.61	0.48	0.59	0.47
	SC	0.44	0.62	0.30	0.59	0.53	0.55

Image 19	DBI	0.40	0.54	0.50	0.67	0.45	0.46
	SC	0.59	0.59	0.58	0.34	0.64	0.68
Image 20	DBI	0.46	0.47	0.64	0.45	0.50	0.31
	SC	0.59	0.71	0.28	0.70	0.67	0.72
Image 21	DBI	0.51	0.50	0.52	0.51	0.44	0.49
	SC	0.59	0.46	0.49	0.57	0.65	0.53
Image 22	DBI	0.43	0.49	0.45	0.38	0.44	0.44
	SC	0.57	0.67	0.56	0.62	0.57	0.57
Image 23	DBI	0.49	0.47	0.48	0.42	0.41	0.55
	SC	0.64	0.56	0.52	0.62	0.60	0.24
Image 24	DBI	0.54	0.46	0.47	0.49	0.46	0.48
	SC	0.35	0.62	0.65	0.55	0.68	0.71
Image 25	DBI	0.45	0.49	0.44	0.50	0.32	0.35
	SC	0.66	0.62	0.52	0.69	0.73	0.71
Image 26	DBI	0.42	0.53	0.55	0.48	0.41	0.39
	SC	0.71	0.61	0.55	0.71	0.80	0.78
Image 27	DBI	0.49	0.48	0.39	0.48	0.51	0.44
	SC	0.63	0.63	0.66	0.65	0.66	0.67
Image 28	DBI	0.49	0.50	0.49	0.46	0.49	0.49
	SC	0.64	0.61	0.62	0.59	0.65	0.61
Image 29	DBI	0.44	0.49	0.48	0.46	0.38	0.45
	SC	0.69	0.69	0.69	0.75	0.75	0.55
Image 30	DBI	0.45	0.38	0.37	0.48	0.38	0.39
	SC	0.76	0.81	0.81	0.72	0.72	0.76
Mean metrics result	Mean DBI	0.481	0.485	0.492	0.472	0.440	0.455
	Mean SC	0.614	0.629	0.588	0.632	0.673	0.644

The second experiment aims to test the improved algorithm by the tree-seeding algorithm. We conducted this experiment by setting the number of initial clusters to four, and testing four different values of iterations, using the optimal threshold value obtained in our previous experiment. We perform our experiment in a set of 30 images, but for simplification only the first 10 images are displayed in Figure 4. The effectiveness of our approach was evaluated using the DBI and SC metrics, and the results are shown in Table 2 with different TSA iterations.



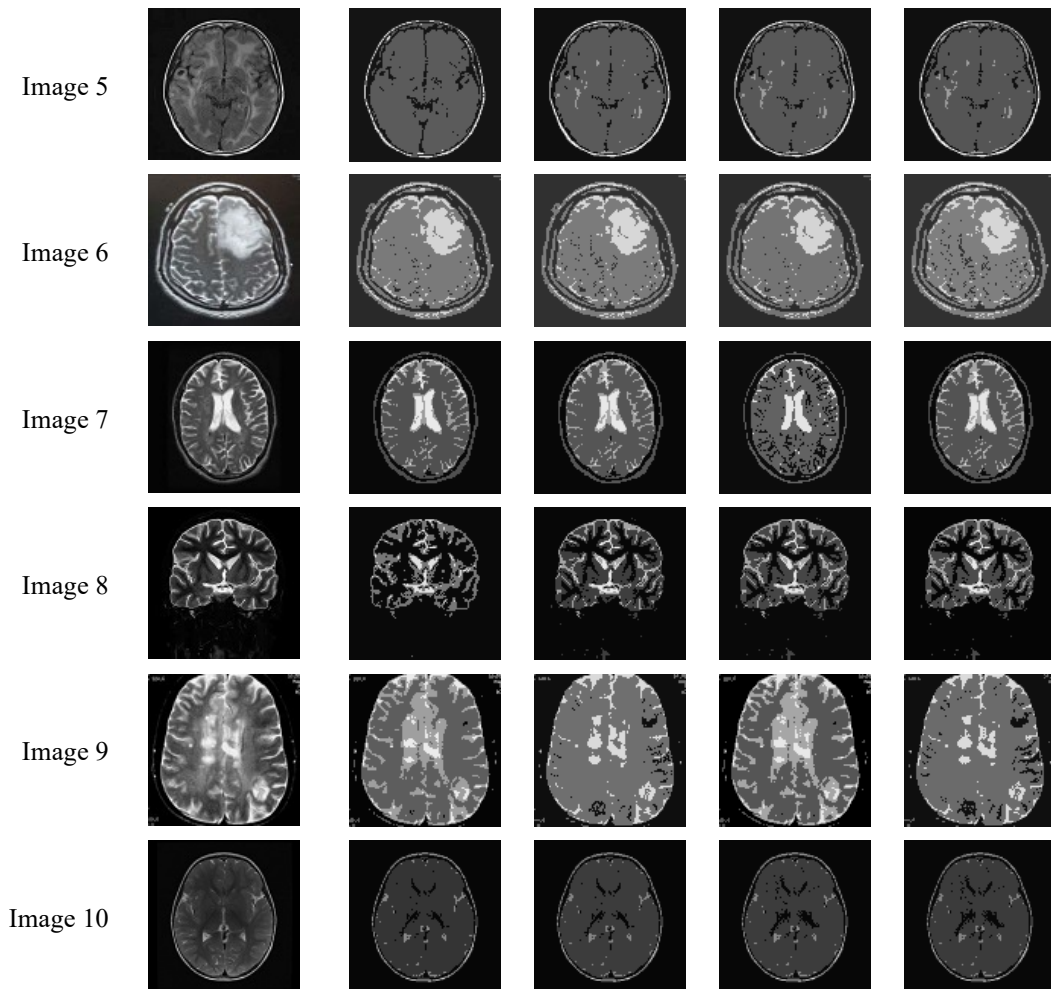


Figure 4. Segmentation results for the improved TSA algorithm.

Table 2: Performance Results on the Brain MRI for the Hybrid Algorithm with different TSA Iterations.

Original image	Index	Number of TSA Iteration			
		15	25	35	45
Image 1	DBI	0.37	0.37	0.37	0.37
	SC	0.82	0.81	0.81	0.81
Image 2	DBI	0.42	0.42	0.42	0.42
	SC	0.70	0.70	0.70	0.70
Image 3	DBI	0.41	0.41	0.41	0.41
	SC	0.76	0.76	0.76	0.76
Image 4	DBI	0.44	0.44	0.44	0.44
	SC	0.72	0.72	0.72	0.72
Image 5	DBI	0.29	0.41	0.41	0.41
	SC	0.77	0.76	0.76	0.76
Image 6	DBI	0.49	0.49	0.49	0.50
	SC	0.56	0.55	0.55	0.54
Image 7	DBI	0.40	0.39	0.38	0.39
	SC	0.78	0.78	0.70	0.78
Image 8	DBI	0.47	0.42	0.42	0.42
	SC	0.72	0.74	0.74	0.74
Image 9	DBI	0.40	0.40	0.41	0.41

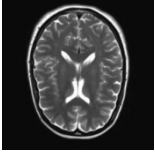

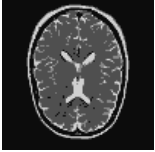
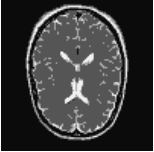
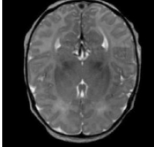
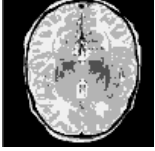
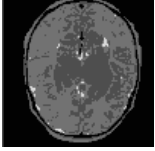
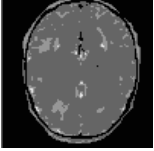
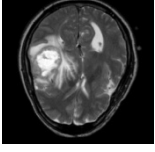
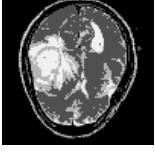
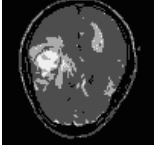
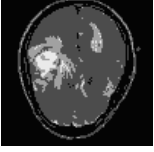




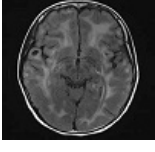
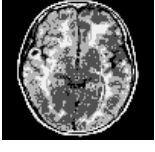
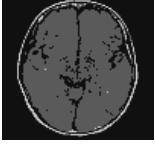
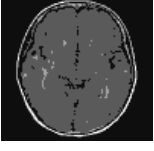
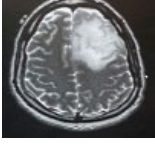

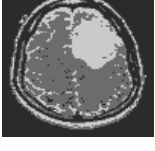
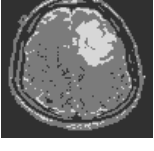




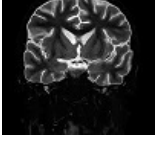


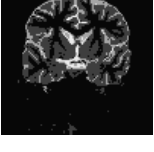
	SC	0.60	0.60	0.58	0.58
Image 10	DBI	0.35	0.35	0.35	0.35
	SC	0.75	0.75	0.74	0.74
Image 11	DBI	0.37	0.35	0.43	0.40
	SC	0.70	0.67	0.71	0.70
Image 12	DBI	0.39	0.39	0.39	0.39
	SC	0.82	0.82	0.82	0.82
Image 13	DBI	0.49	0.50	0.50	0.49
	SC	0.69	0.68	0.68	0.68
Image 14	DBI	0.53	0.53	0.52	0.52
	SC	0.72	0.72	0.72	0.72
Image 15	DBI	0.43	0.42	0.42	0.42
	SC	0.67	0.68	0.68	0.68
Image 16	DBI	0.29	0.29	0.29	0.29
	SC	0.76	0.76	0.76	0.76
Image 17	DBI	0.40	0.41	0.38	0.41
	SC	0.73	0.73	0.68	0.73
Image 18	DBI	0.34	0.36	0.39	0.46
	SC	0.67	0.65	0.63	0.70
Image 19	DBI	0.41	0.41	0.38	0.41
	SC	0.64	0.64	0.65	0.64
Image 20	DBI	0.30	0.48	0.48	0.48
	SC	0.71	0.72	0.72	0.72
Image 21	DBI	0.45	0.46	0.46	0.46
	SC	0.59	0.59	0.59	0.59
Image 22	DBI	0.32	0.32	0.45	0.32
	SC	0.74	0.73	0.55	0.73
Image 23	DBI	0.50	0.50	0.50	0.49
	SC	0.65	0.65	0.64	0.66
Image 24	DBI	0.44	0.43	0.43	0.44
	SC	0.73	0.74	0.71	0.75
Image 25	DBI	0.49	0.51	0.49	0.51
	SC	0.71	0.69	0.68	0.70
Image 26	DBI	0.33	0.33	0.33	0.33
	SC	0.81	0.81	0.81	0.81
Image 27	DBI	0.43	0.43	0.23	0.42
	SC	0.72	0.67	0.75	0.68
Image 28	DBI	0.47	0.50	0.49	0.49
	SC	0.65	0.64	0.64	0.64
Image 29	DBI	0.37	0.38	0.40	0.38
	SC	0.70	0.75	0.66	0.75
Image 30	DBI	0.38	0.38	0.37	0.35
	SC	0.80	0.80	0.80	0.81
Mean result of metrics	Mean DBI	0.406	0.416	0.414	0.419
	Mean SC	0.713	0.710	0.698	0.713

4.2 Discussion

The first experiment aimed to determine the optimal threshold value for image segmentation. The results obtained indicated that a threshold value of 50 produced the best result, with an average DBI of 0.440 and a silhouette coefficient of 0.673. Additionally, we observed that the number of clusters obtained after segmenting each image fell within the range of three to four.

On the other hand, the second experiment aimed to test the improved algorithm by the tree-seeding algorithm. The results obtained indicated that a running of 15 iteration of TSA produced the best result, with an average DBI of 0.406 and a silhouette coefficient of 0.713.

Table 3 presents a comparison between the outcomes of our improved algorithm and the FCM algorithm, including the segmenting results of FCM with four clusters and a fuzziness degree of 2, as well as 100 iterations. By comparing the proposed method with the FCM algorithm, we observed that our algorithm achieves better results in terms of the DBI metric, which indicates the average similarity measure for each cluster with their closest cluster. This finding suggests that our improved algorithm with TSA is more effective in separating the final clusters from the image. We obtain an average DBI of 0.418 and a silhouette coefficient of 0.715.

	Original image	FCM	Proposed algorithm	Improved algorithm with TSA
Image 1				
Image 2				
Image 3				
Image 4				
Image 5				
Image 6				
Image 7				
Image 8				

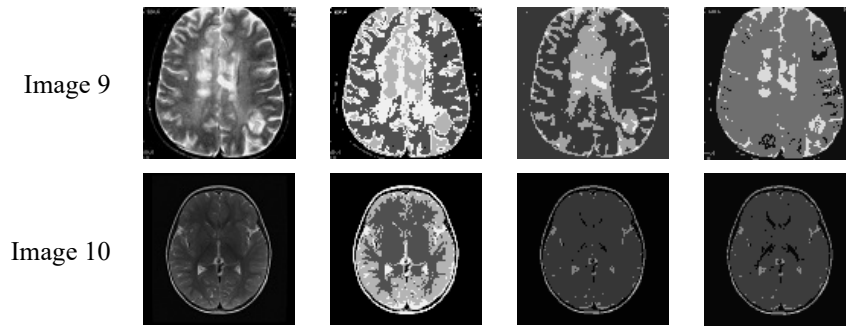


Figure 5. Qualitative comparison between the results of three methods (FCM, Proposed and improved approach).

Table 3: Comparison of performance results on brain MRI for three methods with different TSA iterations

Original image	Index	FCM	Proposed algorithm	Improved algorithm with TSA
Image1	DBI	0.43	0.41	0.37
	SC	0.80	0.77	0.81
Image2	DBI	0.52	0.47	0.42
	SC	0.70	0.67	0.70
Image 3	DBI	0.45	0.41	0.41
	SC	0.76	0.76	0.76
Image 4	DBI	0.44	0.42	0.44
	SC	0.72	0.69	0.72
Image 5	DBI	0.62	0.35	0.41
	SI	0.65	0.76	0.76
Image 6	DBI	0.46	0.46	0.49
	SC	0.63	0.64	0.55
Image 7	DBI	0.44	0.61	0.39
	SC	0.77	0.62	0.78
Image 8	DBI	0.53	0.44	0.42
	SC	0.73	0.72	0.74
Image 9	DBI	0.46	0.53	0.40
	SC	0.65	0.45	0.60
Image 10	DBI	0.48	0.36	0.35
	SC	0.73	0.76	0.75
Image 11	DBI	0.44	0.44	0.40
	SC	0.80	0.71	0.70
Image 12	DBI	0.52	0.37	0.39
	SC	0.70	0.74	0.82
Image 13	DBI	0.45	0.46	0.49
	SC	0.76	0.59	0.68
Image 14	DBI	0.41	0.50	0.52
	SI	0.70	0.58	0.72
Image 15	DBI	0.65	0.42	0.42
	SC	0.65	0.64	0.68
Image 16	DBI	0.47	0.41	0.29
	SC	0.63	0.70	0.76
Image 17	DBI	0.44	0.36	0.41
	SC	0.78	0.72	0.73
Image 18	DBI	0.53	0.59	0.46
	SC	0.72	0.53	0.70
Image 19	DBI	0.45	0.45	0.41
	SC	0.65	0.64	0.64
Image 20	DBI	0.49	0.50	0.48

	SC	0.73	0.67	0.72
Image 21	DBI	0.45	0.44	0.46
	SC	0.75	0.65	0.59
Image 22	DBI	0.50	0.44	0.32
	SC	0.70	0.57	0.73
Image 23	DBI	0.45	0.41	0.49
	SC	0.76	0.60	0.66
Image 24	DBI	0.46	0.46	0.44
	SC	0.72	0.68	0.75
Image 25	DBI	0.63	0.32	0.51
	SC	0.61	0.73	0.70
Image 26	DBI	0.46	0.41	0.33
	SC	0.63	0.80	0.81
Image 27	DBI	0.44	0.51	0.42
	SC	0.74	0.66	0.68
Image 28	DBI	0.53	0.49	0.49
	SC	0.73	0.65	0.64
Image 29	DBI	0.46	0.38	0.38
	SC	0.64	0.75	0.75
Image 30	DBI	0.46	0.38	0.35
	SC	0.73	0.72	0.81
Mean result	DBI	0.484	0.440	0.418
	SC	0.709	0.673	0.715

5 Conclusion and Futur Work

This paper explores two clustering methods for segmenting brain MRI images. The first method is a novel clustering algorithm that can effectively segment any image into the appropriate number of clusters. The second method involves combining the first method with the TSA, a biologically inspired algorithm, to address the issue of random cluster center selection and achieve better outcomes. The TSA is leveraged to redistribute the cluster centers and minimize the objective function to avoid the problem of local minima. After comparing our approach with the FCM using two clustering metrics Davies–Bouldin index and Silhouette coefficient, we found that our improved algorithm outperforms the FCM algorithm. The proposed approach offers three benefits: edge detection, which plays a critical role in patient monitoring, segmentation, which partitions the image into homogeneous regions to extract essential information for later use, and interpretation, which helps to determine the normality of the acquired image. Ensuring the efficiency and reliability of our tool is essential, as it can significantly aid doctors in their diagnoses. To further improve this algorithm, future studies may explore the integration of other meta-heuristic methods, such as PSO, the bat-algorithm, or the firefly algorithm.

References

- [1] A. Mahmoud Khaled, A. I. Awad, A. M. Khalaf and F. A. Hamed. A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned. *Magnetic resonance imaging*, 61: 300-318, 2019.
- [2] S. Saha and S. Mukhopadhyay, A comparative study of k-means and fuzzy C-means clustering algorithms for brain tumor segmentation in MRI, *Journal of Medical Systems*, 44 (1): 1-11, 2020.
- [3] S. Gopalakrishnan, and K. Manimegalai. A novel approach for brain tumor segmentation using fuzzy c-means clustering algorithm. *Computer Methods and Programs in Biomedicine*, 178: 139-150, 2019.
- [4] S. Chandra and R. Srivastava. Automated breast ultrasound image segmentation using hierarchical clustering and watershed transform. *Journal of Medical Imaging and Health Informatics*, 10(10), 2297-2306, 2020.
- [5] G. Y., Chen, and H. Zhang. An unsupervised clustering algorithm with DBSCAN for liver tumor segmentation in CT images. *Journal of Medical Imaging and Health Informatics*, 12(1), 122-128, 2022.

- [6] Y.A. Tareq Alharbi and M. Al-Tarifi. The use of expectation-maximization algorithm in nuclear medicine imaging: A review. *International Journal of Radiation Research*, 19 (1): 1-8, 2021.
- [7] S. Zhang, S. Li, J. Li, H. Li, and Q. Li, Medical image segmentation based on adaptive Firefly algorithm, *Journal of Medical Systems*, 44 (10): 1-11, 2020.
- [8] Y. Zhang, X. Chen, F. Shi, and L. Guo, Medical image segmentation using a hybrid Bat algorithm, *Journal of Ambient Intelligence and Humanized Computing*, 11(4): 1299-1310, 2020.
- [9] M. Meenakshi, K. Latha. Medical Image Segmentation using Tree Seed Algorithm. *International Journal of Engineering Research & Technology*, 8(12):169-174, 2019.
- [10] S. Das and C.A. Chakraborty. A hybrid approach using fuzzy c-means and particle swarm optimization for segmentation of brain magnetic resonance images. *Multimedia Tools and Applications*, 80(5): 6305-6332, 2021.
- [11] S. R. Daraei, S. Khademi, and H. Rabbani. An efficient optimization-based method for segmentation of knee bone structures in X-ray images using gray wolf optimization. *Computer Methods and Programs in Biomedicine*. 203: 1-13, 2021.
- [12] S. N. Kumar, A. L. Fred, H. A. Kumar and P. S. Varghese. Firefly optimization based improved fuzzy clustering for ct/mr image segmentation, *Intelligent Systems Reference Library, pages 1-28, 2018*.
- [13] parth Ghosh, K. Mali, S. K. Das, Chaotic firefly algorithm-based fuzzy c-means algorithm for segmentation of brain tissues in magnetic resonance images, *J. Vis. Commun. Image Represent.* 54: 63–79, 2018.
- [14] A. N. Benaichouche, H. Oulhadj and P. Siarry. Improved spatial fuzzy c-means clustering for image segmentation using pso initialization, mahalnobis distance and post-segmentation correction, *Digit. Signal Process.* 23: 1390–1400, 2013.
- [15] H. Verma, D. Verma and P. K. Tiwari. A population based hybrid FCM-PSO algorithm for clustering analysis and segmentation of brain image, *Expert Syst. Appl.* 167: 114-121. 2021.
- [16] R. C. Hroik, E. Tuba, E. Dolicanin, R. Jovanovic and M. Tuba. Brain image segmentation based on firefly algorithm combined with k-means clustering, *Studies in Informatics and Control.* 28(2): 167-176. 2019.
- [17] H. Zhu, H. He, J. Xu, Q. Fang and W. Wang. Medical image segmentation using fruit fly optimization and density peaks clustering. *Computational and Mathematical Methods in Medicine*, 2018.
- [18] L. Khrissi, N. E. Akkad, H. Satori and K. Satori, Image segmentation based on k-means and genetic algorithms, 2020.
- [19] M. S. Kiran, Tsa: Tree-seed algorithm for continuous optimization. *Expert Syst. Appl.* 42 : 6686–6698, 2015.
- [20] Ren, J., Yang, Y., Yu, H., Liu, H., Liu, X., & Li, Q. (2021). Performance evaluation of clustering algorithms: A comprehensive study. *Knowledge-Based Systems*, 229, 107120.
- [21] A. Al Mamun, P. P. Em and J. Hossen. An efficient encode-decode deep learning network for lane markings instant segmentation. *International Journal of Electrical and Computer Engineering.* 11(6): 4982-4990, 2021.
- [22] A. Al Mamun, P. P. Em, Md. J. Hossen, A. Tahabilder and B. Jahan, Efficient lane marking detection using deep learning technique with differential and cross-entropy loss, *International Journal of Electrical and Computer Engineering.* 12(4): 4206-4216, 2022.
- [23] I. Ali, N. Tarihoran, U. Supriatna, A. Tohir, S. Suwarni and R. Rahim. Distance analysis measuring for clustering using k-means and Davies Bouldin index algorithm.” *TEM Journal.* 11(4): 1871-1876, 2022.
- [24] R. Hana. Different approaches to the silhouette coefficient calculation in cluster evaluation. *In Proc. of the 21st International Scientific Conference AMSE Applications of Mathematics and Statistics in Economics*, Kutná Hora, Czech Republic, 2018.
-