Comparison Among Image Clustering Algorithms

Khalid Mohammed Saffer

Department of Computer Science,Collage of Science,University of Diyala

Receiving Date: 20-03-2011 **- Accept Date:** 27-06-2011

Abstract

Data clustering is a fundamental operation used in unsupervised images generally clustering involves asset of data (e.g.: image pixels) into specified no of clusters, the motivation behind clustering is to find inherit structure in the data and to expose the structure as asset of groups.

Our search concern with taking image clustering problem using four clustering algorithms named K-mean, K-median, PSO and hybrid of two algorithms, PSO and k-mean. These algorithms applied on three gray brain images then compare the results.

الخلاصة

تعتبر عملية تجميع البيانات عملية جوهرية استخدمت في الصور ذات البيانات غير المرتبة او المجمعة ، بصورة عامة عملية التجميع تتضمن ترتيب البيانات (مثلاً معلومات صورة) الى مجموعة من العناقيد او المصنفات لغرض الحصول على هيكل وراثي للبيانات وعرض الهيكل المذكور كمجاميع مرتبة.

يتضمن البحث مقارنة النتائج التي يتم الحصول عليها من خلال تطبيق 4 خوارزميات للغرض اعلاه هي (K-mean, K-median, PSO and hybrid of two algorithms, PSO and k-mean) على 3 صور للدماغ من نوع gray .

Introduction

There are several clustering algorithms that belong to the unsupervised approach. These algorithms can be categorized into two groups: hierarchical and partitional [1][2]. In hierarchical clustering, the output "is a tree showing a sequence of clustering with each clustering being a partition of the data set"[2]. This type of algorithms has the following advantages:

1- The number of classes need not be specified a priori.

2- They are independent of the initial condition.

However, hierarchical clustering suffers from the following drawbacks:

1- These algorithms are static, i.e. pixels assigned to a cluster cannot move to another cluster.

2- They may fail to separate overlapping clusters due to lack of information about the global shape or size of the clusters [1]. On the other hand, partitional clustering algorithms partition the data set into a specified number of classes. These algorithms try to minimize certain criteria (e.g. a square error function). Therefore, they can be treated as an optimization problem. The advantages of the hierarchical algorithms are the disadvantages of the partitional algorithms and vice versa [2].

The most widely used partitional algorithm is the iterative K-means approach. K-means clustering starts with K cluster centers or centroids, with the initial values set randomly or from a priori information. Each pixel in the image is then assigned to the closest cluster (i.e. closest centroid). Finally, the centroids are recalculated according to the associated pixels. This process is repeated until convergence [3]. On the other hand, a recent clustering algorithm based on PSO that minimizes the initial condition effects has been proposed by the authors [4].

Usually researchers use their own data sets to test the performance of their clustering algorithms [5]. In addition, many researchers create their own synthetic data to test their algorithms. This approach makes the comparison between different clustering algorithms difficult. In order to help in solving this problem we need to create a simple tool, which will help researchers to create synthetic images. Researchers then can apply their clustering algorithm on these images and check the performance of these algorithms. Furthermore, researchers can agree to use some of these synthetic images as benchmarks making comparison between different clustering algorithms easier.

Some of clustering algorithms:

K-Mean algorithm

The basic step of k-means clustering is simple. In the beginning we determine number of cluster K and we assume the centroid or center of these clusters. We can take any random objects as the initial centroids or the first K objects in sequence can also serve as the initial centroids [5].

Here is step by step k-means clustering algorithm:

Algorithm No.(1) k-means clustering

Step 1. Begin with a decision on the value of k = number of clusters.

Step 2. Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:

1 - Take the first k training sample as single-element clusters

2 - Assign each of the remaining (N-k) training samples to the cluster with the nearest centroid. After each assignment, recomputed the centroid of the gaining cluster.

Step 3 . Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Step 4 . Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

If the number of data is less than the number of cluster then we assign each data as the centroid of the cluster. Each centroid will have a cluster number. If the number of data is bigger than the number of cluster, for each data, we calculate the distance to all centroid and get the minimum distance. This data is said belong to the cluster that has minimum distance from this data [5].

Since we are not sure about the location of the centroid, we need to adjust the centroid location based on the current updated data. Then we assign all the data to this new centroid. This process is repeated until no data is moving to another cluster anymore. Mathematically this loop can be proved to be convergent. The convergence will always occur if the following condition satisfied [5]:

1 - Each switch in step 2 the sum of distances from each training sample to that training sample's group centroid is decreased.

2 -There are only finitely many partitions of the training examples into k clusters.

K-Median algorithm

k-median algorithm is similar to k-mean algorithm in that it take initial random centroid and then refine other centroids from this initial centroid, the main difference is that k-median arrange the clusters for each information of the centroid either in ascending or descending order then take the median value(s) to change the source values of centroid other than calculate the sum and the average of appropriate cluster to change the information of the centroid.

The process above will repeat number of times until reached the optimal solution [5].

Particle Swarm Optimization (PSO)

In this algorithm, each individual solution is referred to as a *particle* and the parametric values to be optimized are encoded in an array known as the *position vector* (x). The flight of these particles is achieved by iteratively changing their position vectors in the search space. To achieve this, PSO maintains a *velocity vector* (v), for each particle, containing the instantaneous velocity of its particle. At each iteration, the instantaneous velocity of a particle is added to its position vector to *Ay* it to its next position. To achieve this, every particle maintains the fittest position it has personally found in a vector referred to as the *personal best* (or pbest), as its .personal memory. Additionally, all particles maintain the memory of the fittest position the population has collectively found in a vector known as the *global best* (or gbest). The values in the velocity vector are calculated using a heuristic which attempts to find fitter solutions in the direction where fit solutions have already been found by each particle personally, and by the entire population, globally [6].

Hybrid PSO and K-Means Clustering Algorithm

Merwe's research [7] indicates that utilizing the PSO algorithm's optimal ability, if given enough time, the PSO clustering algorithm could generate more compact clustering results from the low dimensional dataset than the traditional K-means clustering algorithm. However, when clustering large document datasets, the slow shift from the global searching stage to the local refining stage causes the PSO clustering algorithm to require many more iterations to converge to the optima in the refining stage than the K-means algorithm requiring. Although the PSO algorithm is inherently parallel and can be implemented using parallel hardware, such as a computer cluster, the computation requirement for clustering large document dataset is still high. In our experiments, it needs more than 500 iterations for the PSO algorithm to converge to the optimal result for a document dataset that includes 800 documents. The K-means algorithm only requires 10 to 20 iterations. Although the PSO algorithm generates much better clustering result than the K-means algorithm does, in terms of execution time, the K-means algorithm is more efficient for large datasets [1]. For this reason, we present a hybrid PSO approach that uses K-means algorithm to replace the refining stage in the PSO algorithm. In the hybrid PSO algorithm, the algorithm includes two modules, the PSO module and the K-means module.

The global searching stage and local refine stage are accomplished by those two modules, respectively. In the initial stage, the PSO module is executed for a short period (50 to 100 iterations) to discover the vicinity of the optimal solution by a global search and at the same time to avoid consuming high computation. The result from the PSO module is used as the initial seed of the K-means module. The K-means algorithm will be applied for refining and generating the final result. The whole algorithm can be summarized as:

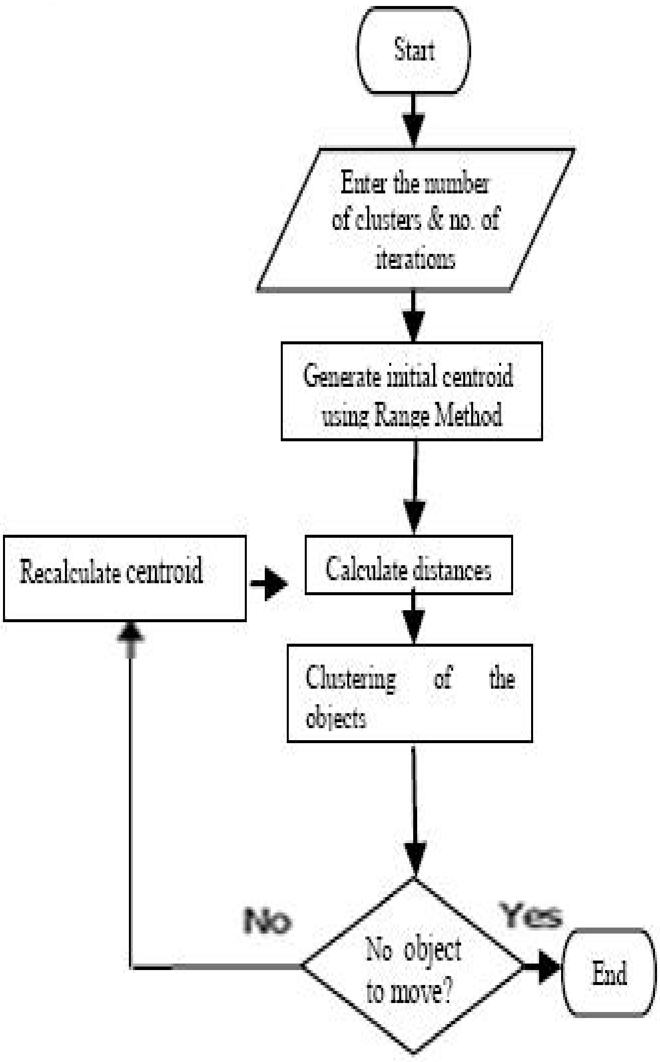
Algorithm No.(2) Hybrid PSO and K-means

(1)Start the PSO clustering process until the maximum number of iterations is exceeded.
(2) Inherit clustering result from PSO as the initial centroid vectors of K-means module.

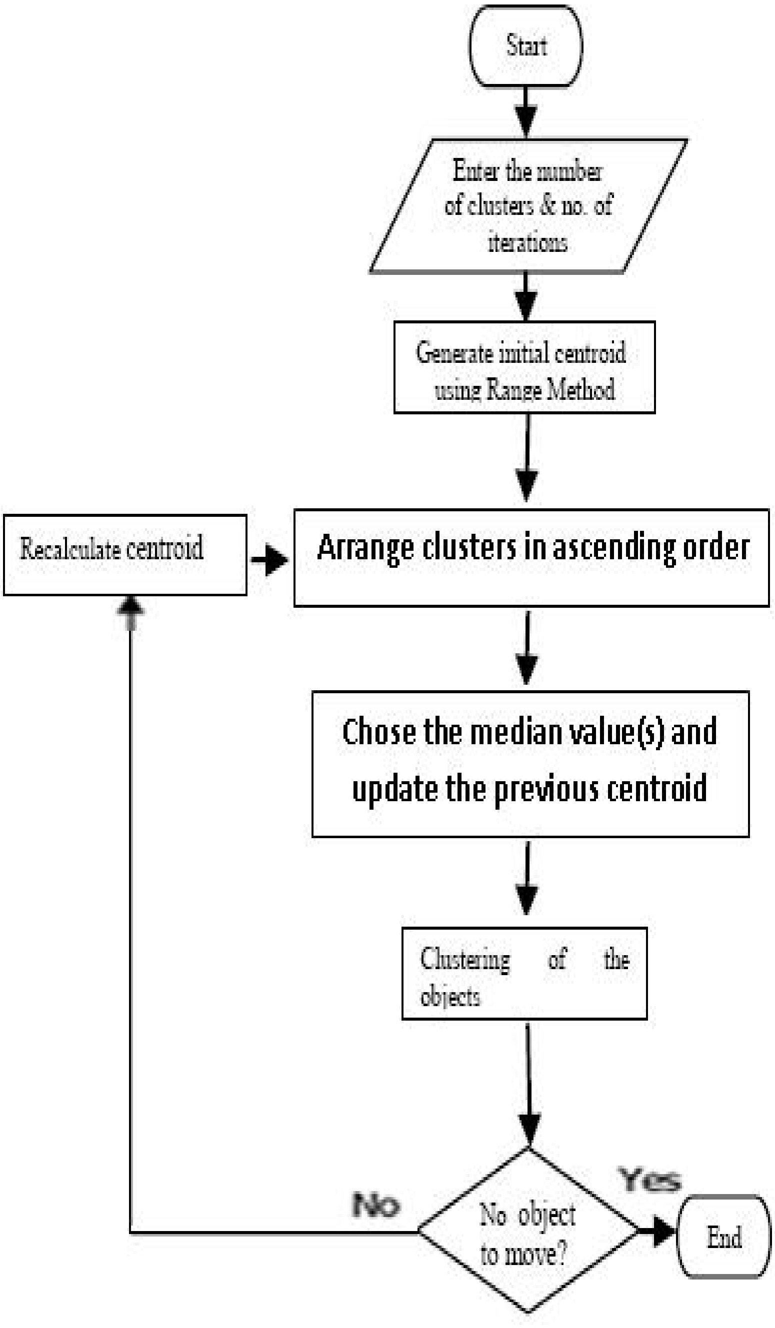
(3)Start K-means process until maximum number of iterations is reached.

Implementation of Proposed Algorithms

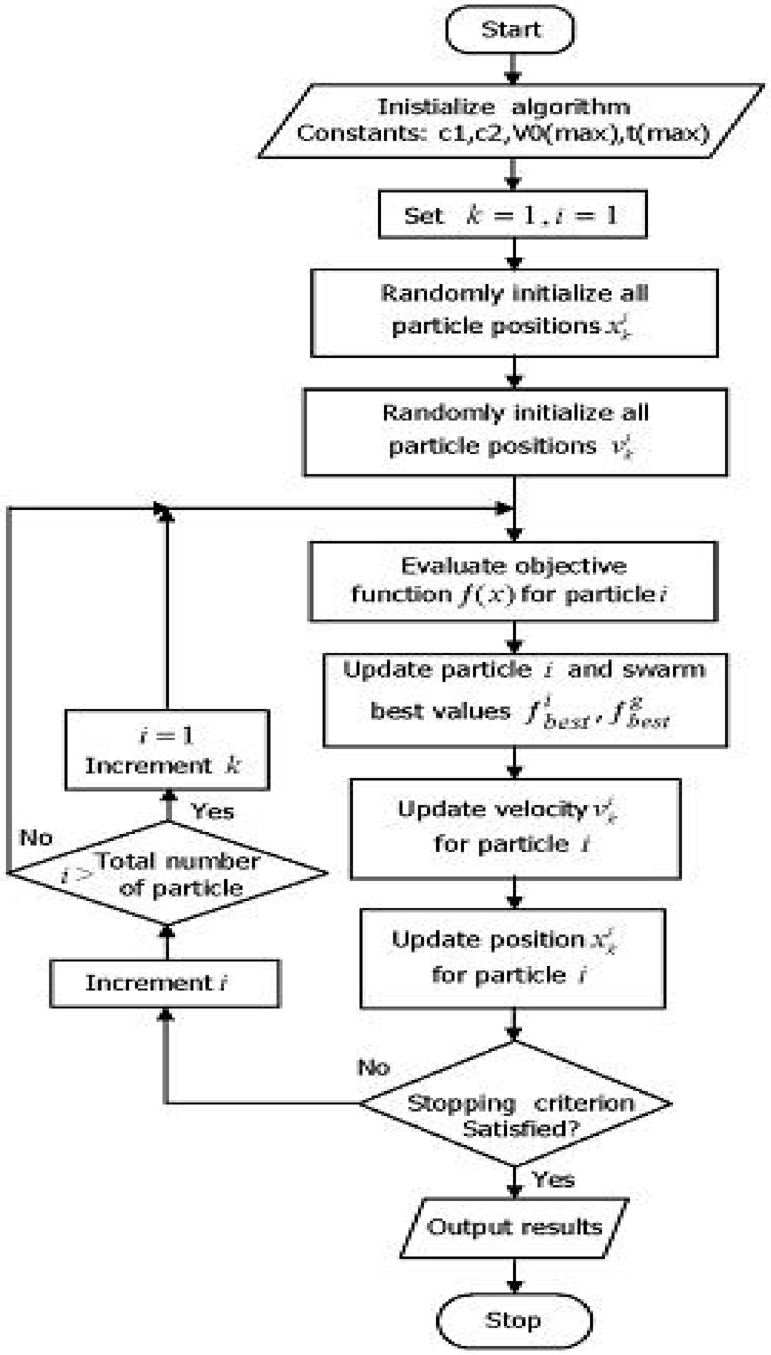
K-mean clustering:



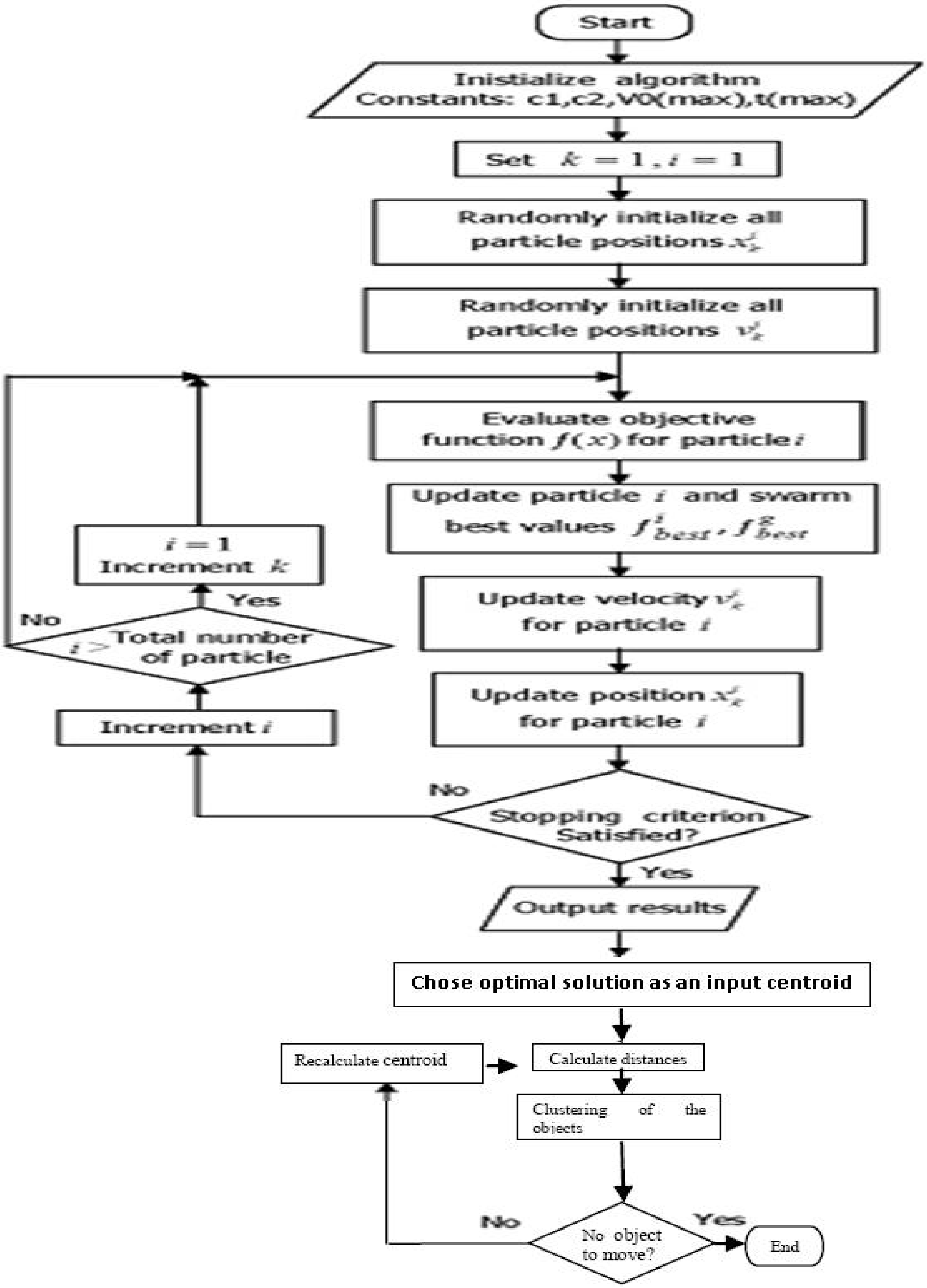
K-median clustering:



PSO clustering:



Hybrid PSO and K-mean:



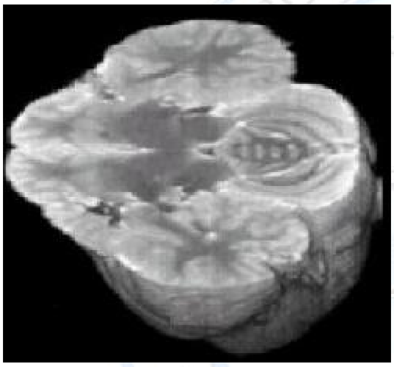
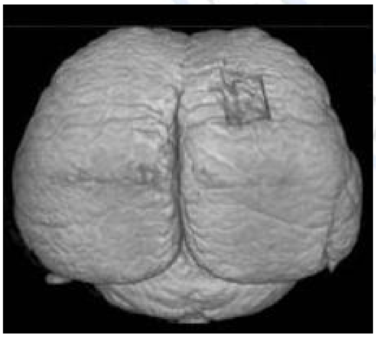
DIYALA JOURNAL FOR PURE SCIENCES

Experiments and Results

Experiment

Three different gray brain images are used to compare the performance of each of four clustering algorithms mentioned early;

K-means, K-median, PSO and Hybrid PSO and K-means. These algorithms are executed by using the Visual Basic programming language (VB6) .The attributes of those images are shown in Fig. (1).

Source Images	Attributes
 Image 1	Size on disk : 170,430 bytes Resolution : 102x102 dpi Width : 248 Height : 229
 Image 2	Size on disk : 138,222 bytes Resolution : 102x102 dpi Width : 228 Height : 202

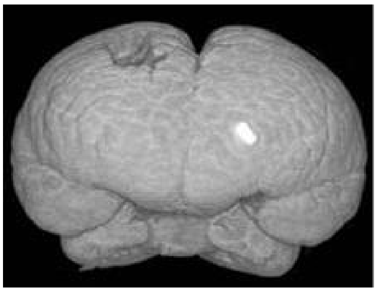
 Image 3	Size on disk : 117,014 bytes Resolution : 102x102 dpi Width : 226 Height : 172

Fig. (1): Description of the test datasets.

The measurements of the Research:

In order to trace the progressive in each algorithm, there must be use several measurements to measure the optimality of each one [2], there are two measures had been used in this research named compactness and separation measures.

Compactness measure:

This measures works locally among the cluster of each centroid value, it measures the rang convergence of cluster's information's (values) ,any algorithm work optimal if the cluster values converging, so initial centroid start at convergence value, when this value is increase then algorithms work well and vice versa.

Separation measures:

This measures starts at a value that must be decreased as soon as the algorithm work well otherwise the algorithm doesn't reach optimal solution, since separation measure using among the values of each centroid (particle)

Results

Tables (1, 2, and 3) show the results at runtime for each algorithm:

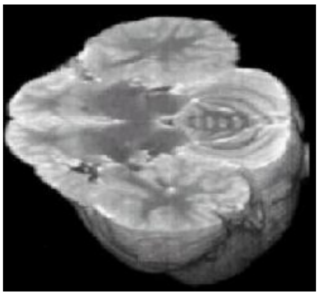
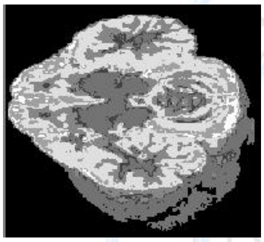
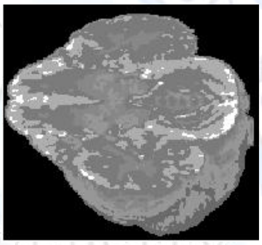
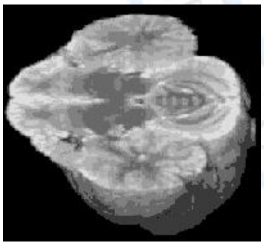
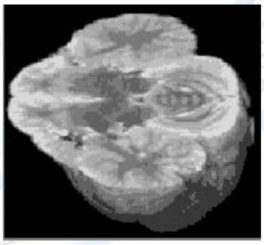
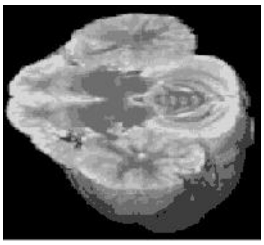
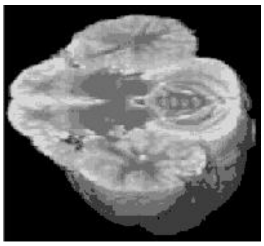
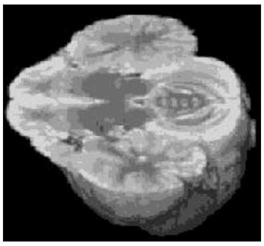
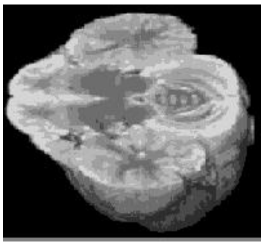


Fig. (2) First Original image

Table (1) Comparison Results for original image of Fig. (2)

algorithm	**Compactness measure**	**Details**	**Separation measure**	**Details**
PSO		**Iteration no.: 150** **Swarm no. : 50** **Area no. : 8** **Start fitness :** **9.02672129190723** **End fitness :** **6.77559269400728**		**Iteration no.: 150** **Swarm no. : 50** **Area no. : 8** **Start fitness :** **14.3229141235352** **End fitness :** **21.7763739038534**
K-Mean		**Iteration no.: 30** **Swarm no. :** **Area no. : 8** **Start fitness :** **12.2975141364076** **End fitness :** **6.40954757310694**		**Iteration no.: 30** **Swarm no. :** **Area no. : 8** **Start fitness :** **16.0892028808594** **End fitness :** **20.3636932373047**
K-Median		**Iteration no.: 30** **Swarm no. :** **Area no. : 8** **Start fitness :** **11.1251016001554** **End fitness :** **6.06521632348395**		**Iteration no.: 30** **Swarm no. :** **Area no. : 8** **Start fitness : 21** **End fitness : 17**

Hybrid PSO + K-Mean		**Iteration no.: 100** **Swarm no. : 50** **Area no. : 8** **Start fitness :** **11.2441949760093** **End fitness :** **6.47008810178553**		**Iteration no.: 100** **Swarm no. : 50** **Area no. : 8** **Start fitness :** **2.2725787** **End fitness :** **22.49153**

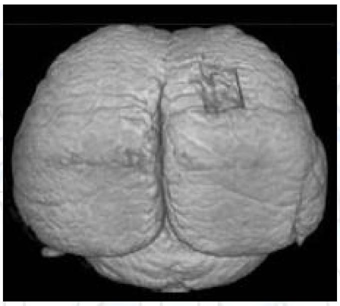
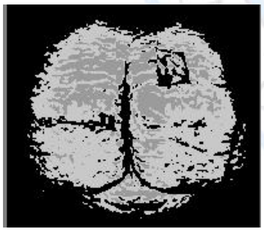
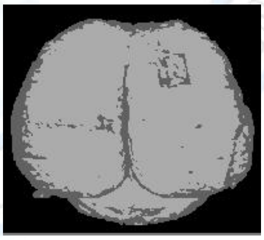
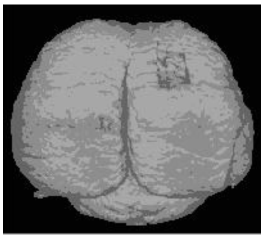
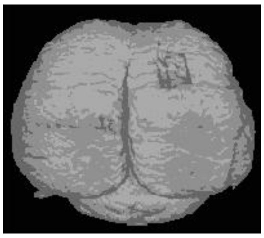
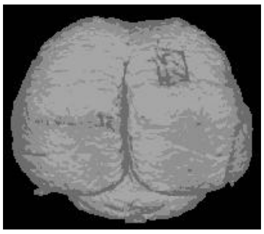
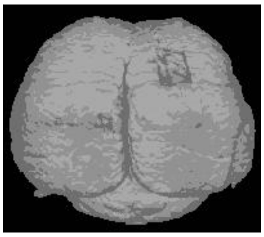
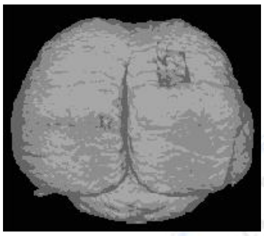
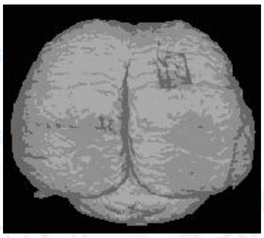


Fig. (3) Second Original image

Table (2) Comparison Results for original image of Fig. (3)

algorithm	**Compactness measure**	**Details**	**Separation measure**	**Details**
PSO		**Iteration no.: 150** **Swarm no. : 50** **Area no. : 4** **Start fitness :** **15.0432943468062** **End fitness :** **5.86298632858805**		**Iteration no.: 150** **Swarm no. : 50** **Area no. : 4** **Start fitness :** **57.2054901123047** **End fitness :** **84.8252361446975**
K-Mean		**Iteration no.: 30** **Swarm no. :** **Area no. : 4** **Start fitness :** **34.1009780129963** **End fitness :** **7.76180326814502**		**Iteration no.: 30** **Swarm no. :** **Area no. : 4** **Start fitness :** **21.8429260253906** **End fitness :** **25.8887939453125**

K-Median		**Iteration no.: 30 Swarm no. : Area no. : 4 Start fitness : 21.7312107598608 End fitness : 7.27632497625044**		**Iteration no.: 30 Swarm no. : Area no. : 4 Start fitness : 40 End fitness : 22**
Hybrid PSO + K-Mean		**Iteration no.: 100 Swarm no. : 50 Area no. : 4 Start fitness : 11.7616941544175 End fitness : 7.76180326814502**		**Iteration no.: 100 Swarm no. : 50 Area no. : 4 Start fitness : 45.75548 End fitness : 25.88879**

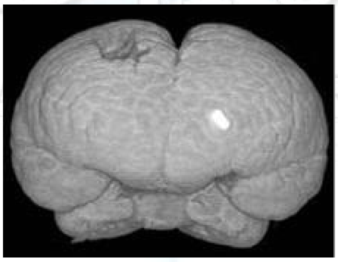
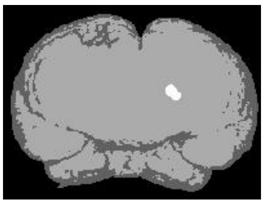
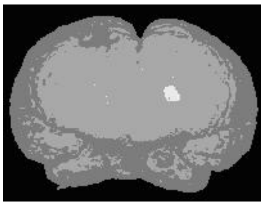
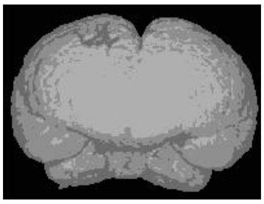
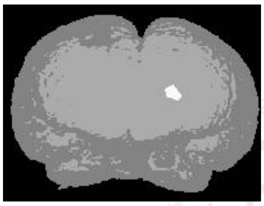
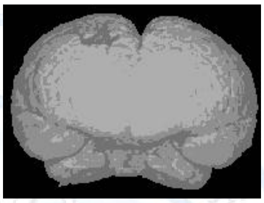
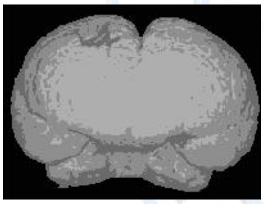
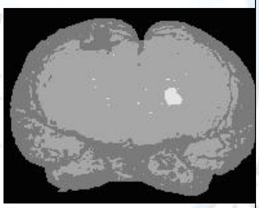


Fig. (4) Third Original image

Table (3) Comparison Results for original image of Fig. (4)

algorithm	**Compactness measure**	**Details**	**Separation measure**	**Details**
PSO		**Iteration no.: 150 Swarm no. : 50 Area no. : 4 Start fitness : 17.1874551970742 End fitness : 50,5.95855029264967**		**Iteration no.: 150 Swarm no. : 50 Area no. : 4 Start fitness : 57.2054901123047 End fitness : 84.8252361446975**

K-Mean		**Iteration no.: 30** **Swarm no. :** **Area no. : 4** **Start fitness :** **16.074946119312** **End fitness :** **11.5252177524449**		**Iteration no.: 30** **Swarm no. :** **Area no. : 4** **Start fitness :** **24.6542510986328** **End fitness :** **31.6297454833984**
K-Median		**Iteration no.: 30** **Swarm no. :** **Area no. : 4** **Start fitness :** **20.3202503595261** **End fitness :** **7.03899641711034**		**Iteration no.: 30** **Swarm no. :** **Area no. : 4** **Start fitness : 11** **End fitness : 30**
Hybrid PSO + K-Mean		**Iteration no.: 100** **Swarm no. : 50** **Area no. : 4** **Start fitness :** **26.4075892857143** **End fitness :** **7.47302451657229**		**Iteration no.: 100** **Swarm no. : 50** **Area no. : 4** **Start fitness :** **47.79905** **End fitness :** **44.24078**

Conclusions

This work investigate clustering algorithms applied on gray medicine images .However, some points can be inferred such as:

- Images with larger number of areas reached to optimal solutions better than the others(less number of areas)
- Since the PSO and the Hybrid success with images that having more details.
- In general the compactness measures had been successes for all algorithms.

- The separation measures record some of failure in images with less details and that applied on an algorithms with communicated calculations (PSO).
- The work can applied on color images, not only gray images.

References

1. Frigui, H., and Krishnapuram, R., A Robust Competitive Clustering Algorithm with Applications in Computer Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, pp 450-465, 1999.
2. Leung, Y., Zhang, J., and Xu, Z., Clustering by Space-Space Filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, pp 1396-1410, 2000.
3. Forgy, E., Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification. Biometrics, Vol. 21, pp 768–769, 1965.
4. Omran, M., Engelbrecht, A., and Salman, A., Particle Swarm Optimization Method for Image Clustering. Submitted to the International Journal on Pattern Recognition and Artificial Intelligence of World Scientific Press, 2003.
5. Puzicha, J., Hofmann, T., and Buhmann, J., Histogram Clustering for Unsupervised Image Segmentation. IEEE Proceedings of the Computer Vision and Pattern Recognition, Vol. 2, pp 602-608, 2000.
6. J Kennedy, RC Eberhart, “Particle Swarm Optimization”, Proceedings of the IEEE International Joint Conference on Neural Networks, Vol. 4, pp 1942–1948, 1995.
7. Merwe V. D. and Engelbrecht, A. P., 2003. Data clustering using particle swarm optimization. Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003), Canbella, Australia. pp. 215-220.