We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

Chapter

Segmenting Images Using Hybridization of K-Means and Fuzzy C-Means Algorithms

Raja Kishor Duggirala

Abstract

Image segmentation is an essential technique of image processing for analyzing an image by partitioning it into non-overlapped regions each region referring to a set of pixels. Image segmentation approaches can be divided into four categories. They are thresholding, edge detection, region extraction and clustering. Clustering techniques can be used for partitioning datasets into groups according to the homogeneity of data points. The present research work proposes two algorithms involving hybridization of K-Means (*KM*) and Fuzzy C-Means (*FCM*) techniques as an attempt to achieve better clustering results. Along with the proposed hybrid algorithms, the present work also experiments with the standard K-Means and *FCM* algorithms. All the algorithms are experimented on four images. CPU Time, clustering fitness and sum of squared errors (SSE) are computed for measuring clustering performance of the algorithms. In all the experiments it is observed that the proposed hybrid algorithm *KMandFCM* is consistently producing better clustering results.

Keywords: image segmentation, clustering, K-Means, Fuzzy C-Means, hybridization, sum of squared error, clustering fitness

1. Introduction

Images are often the most important category among the available digital data. In the recent years, image data is increasing and will continue increase in the near future. Since it is difficult to deal with large amount of image data as the available data increases, it becomes crucial to use the automated tools for various purposes in connection to image data. The image processing provides wide range of techniques to deal with the images. By using the image processing techniques, we can make the work much easier not only for now, but also for the future when there will be more data and more work to do on the images.

Image segmentation is an essential image processing technique that analyzes an image by partitioning it into non-overlapped regions each region referring to a set of pixels. The pixels in a region are similar with respect to some characteristic such as color, intensity, or texture [1]. The pixels significantly differ with those in the other regions with respect to the same characteristic [2–4]. Image segmentation plays an important role in a variety of applications such as robot vision, object recognition, medical imaging and etc. [5–7]. Image segmentation approaches can be divided into four categories. They are thresholding, edge detection, region extraction and clustering. Clustering techniques can be used for data segmenting image data as they are used for partitioning large datasets into groups according to the homogeneity of data points.

In clustering, a given population of data is partitioned into groups such that objects are similar to one another within the same group and are dissimilar to the objects in other groups [8, 9]. There are different categories of clustering techniques. These can be partitional (hierarchical and non-hierarchical), like K-means, PAM, CLARA, CLARANs [10, 11]; model-based, like Expectation Maximization, SOM, Mixture model clustering [12, 13]; or fuzzy-based like Fuzzy C-Means [14, 15].

Partitional clustering techniques attempt to break a population of data into some predefined number of clusters such that the partition optimizes a given criterion.

Formally, clusters can be seen as subsets of the given dataset. So, clustering methods can be classified according to whether the subsets are fuzzy or crisp (hard). In hard clustering, an object either does or does not belong to a cluster. These methods partition the data into a specified number of mutually exclusive subsets. However, in fuzzy-based clustering, the objects may belong to several clusters with different degrees of membership [16].

It is studied in the literature that many researchers experimented with the Fuzzy C-Means (*FCM*) algorithm in a wide variety of ways for achieving better image segmentation results [1, 17]. In [18], a penalized *FCM* (*PFCM*) algorithm is presented for image segmentation for handling noise by adjusting a penalty coefficient. The penalty term used here takes the spatial dependence of the objects into consideration, which is modified according to the *FCM* criterion. In [19], a fuzzy rule-based technique is proposed. It employs the rule-based neighborhood enhancement system to impose spatial continuity by post-processing on the clustering results obtained using *FCM* algorithm. In [20], a Geometrically Guided *FCM* (*GG-FCM*) algorithm is proposed, which is based on a semi-supervised *FCM* technique for multivariate image segmentation. In [21], a regularization term was introduced into the standard *FCM* to impose the neighborhood effect. In [22], this regularization term is incorporated into a kernel-based fuzzy clustering algorithm. In [23], this regularization) term is incorporated into the adaptive *FCM* (*AFCM*) algorithm [24] to overcome the noise sensitivity of *AFCM* algorithm.

However, it is found in the literature that a very less attention is paid towards the hybridization of clustering techniques for partitioning the datasets.

The present research work aims at developing hybrid clustering algorithms involving K-Means and Fuzzy C-Means (*FCM*) techniques for achieving better clustering results. As part of hybridization, two algorithms are developed, *KMFCM* and *KMandFCM*. The *KMFCM* algorithm first performs K-Means on the dataset and then performs *FCM* using the results of K-Means. The *KMandFCM* algorithm performs K-Means and *FCM* in the alternative iterations.

All the experiments are carried out using the datasets that are related to four images. For performance evaluation, CPU time, clustering fitness and sum of squared error (SSE) are taken into consideration.

The following sections provide a detailed discussion of K-Means (*KM*), Fuzzy C-Means (*FCM*), *KMFCM* and *KMandFCM* algorithms.

2. The K-Means (*KM*) algorithms

Partitional clustering methods are appropriate for the efficient representation of large datasets [11]. These methods determine k clusters such that the data objects in a cluster are more similar to each other than to the objects in other clusters.

The K-Means is a partitional clustering method, which partitions a given dataset into a pre-specified number, k, of clusters [25]. It is a simple iterative method. The algorithm is initialized by randomly choosing k points from a given dataset as the initial cluster centers, i.e., cluster means. The algorithm iterates through two steps till its convergence:

1. Data assignment: this step partitions the data by assigning each data object to its closest cluster center.

2. Updating the cluster centers: update the center of each cluster based on the objects assigned to that cluster.

The algorithm for K-Means is as follows [26]. Here, k represents the number of clusters, d represents the number of dimensions or attributes, X_i represents the ith data sample, μ_j (j = 1, 2, …, k) represents the mean vector of cluster C_j, t is the iteration number. For termination condition the algorithm computes *percentage change*, Eq. (2). The algorithm terminates when *Percentage change* $< \alpha$. Here, α is assumed to be 3 since it is negligible.

***KM* algorithm**

1. Select k vectors randomly from the dataset as the initial cluster centers, μ_j (j = 1, 2, …, k). Set the current iteration $t = 0$.

2. Assign each vector, X_i, to its closest cluster center using Euclidean distance, Eq. (1).

$$d\left(X_i, \mu_j\right) = \sqrt{\sum_{l=1}^{d}\left(x_{il} - \mu_{jl}\right)^2} \tag{1}$$

3. Update mean vectors μ_j (j = 1, …, k).

4. Compute Percentage change as follows

$$\text{Percentage change} = \frac{|\Psi_t - \Psi_{t+1}|}{\Psi_t} \times 100 \tag{2}$$

where Ψ_t is the number of vectors assigned to new clusters in tth iteration and Ψ_{t+1} is the number of vectors assigned to new clusters in $(t + 1)$th iteration.

5. Stop the process if *Percentage change* $< \alpha$, otherwise set $t = t + 1$ and repeat the steps 2–4 with the updated parameter.

The K-Means uses Euclidean distance as a proximity measure for determining the closest cluster to which a data object is assigned [13]. The algorithm stops when the assignment of data points to the clusters no longer changes or some other criterion is satisfied. The K-Means is a widely used algorithm for clustering and it requires less CPU time. However, it mainly suffers from detecting the natural clusters that have non-spherical shapes or widely different sizes or densities [25].

3. The Fuzzy C-Means (*FCM*) algorithms

Fuzzy-based clustering techniques focus on modeling uncertain and vague information that is found in the real world situations. These techniques deal with the clusters whose boundaries cannot be defined sharply [14, 15]. By fuzzy-based clustering, one can know if data objects fully or partially belong to the clusters based on their memberships in different clusters [27]. Among the fuzzy-based clustering methods, Fuzzy C-Means (*FCM*) is the most well-known algorithm as it has the advantage of robustness for obscure information about the clusters [1, 28].

In *FCM*, a dataset is grouped into k clusters, where every data object may relate to every cluster with some degree of membership to that cluster [16]. The membership of a data object towards a cluster can range between 0 and 1 [29]. The sum of memberships for each data point must be unity.

The *FCM* iterates through two phases for converging to a solution. First, each data object will be associated with a membership value for each cluster, and second, assigning the data object to the cluster with the highest membership value [2].

The algorithm for *FCM* is given below [30]. Here, U is the $k \times N$ membership matrix. While computing the cluster centers and updating the membership matrix at each iteration, the *FCM* uses membership weight, m. For most data $1.5 \leq m \leq 3.0$ gives good results [29]. In all our experiments, we take $m = 1.25$.

***FCM* algorithm**

1. Initialize parameters: select k vectors randomly as cluster means; set initial membership matrix $U^{(0)}_{k \text{ X } N}$, set the current iteration $t = 0$.

2. Assign each data object X_i to clusters using the membership matrix.

3. Compute jth cluster center as follows:

$$\mu_j^{t+1} = \frac{\sum_{i=1}^{N} \left(u_{ji}\right)^m X_i}{\sum_{i=1}^{N} \left(u_{ji}\right)^m} \tag{3}$$

4. Compute new membership matrix using

$$u_{ji}^{t+1} = \left[\sum_{l=1}^{k} \left(\frac{\left\| X_i - \mu_j^{\,t} \right\|^2}{\left\| X_i - \mu_l^{\,t} \right\|^2} \right)^{1/m - 1} \right]^{-1} \tag{4}$$

5. Assign each data object X_i to clusters using the membership matrix.

6. Compute *Percentage change* using Eq. (2).

7. Stop the process if the *Percentage change* is $< \alpha$. Otherwise, set $t = t + 1$ and repeat the steps 3–7 with the updated parameters.

FCM is widely studied and applied in geological shape analysis [31], medical diagnosis [32], automatic target recognition [33], meteorological data [28], pattern recognition, image analysis, image segmentation and image clustering [34–36], agricultural engineering, astronomy, chemistry [37], detection of polluted sites [38] and etc.

4. Hybridization involving K-Means and *FCM* techniques

The partitional [11] and fuzzy-based [16] methods are widely applied clustering techniques in several areas. The partitional clustering methods do hard clustering, where the dataset is partitioned into a specified number of mutually exclusive subsets. The K-Means, as a partitional clustering method is found in the research literature as widely applied technique in a variety of experiments. While clustering the data, the K-Means aims at minimizing the local distortion [39, 40]. However, K-Means is ideal if the data objects are distributed in well-separated groups.

In fuzzy-based clustering, objects are not forced to fully belong to one cluster. Here, an object may belong to many clusters with varying degrees of membership. This membership can range between 0 and 1 indicating the partial belongingness of objects to the clusters [16]. Fuzzy clustering techniques help in understanding if the data objects fully or partially belong to clusters depending on their memberships [27]. In *FCM*, each data object belongs to each cluster with some degree of membership that ranges between 0 and 1 [29]. Here, clusters are treated as fuzzy sets. In general, introducing the fuzzy logic in K-Means is the Fuzzy C-Means algorithm [41].

The following sub-section discusses two algorithms that apply hybridization of K-Means (*KM*) and Fuzzy C-Means (*FCM*) clustering techniques [42]. These algorithms are *KMFCM* and *KMandFCM*. The *KMFCM* algorithm first performs K-Means on the given dataset and then performs the *FCM* using the results of K-Means. The *KMandFCM* algorithm performs K-Means and *FCM* in the alternative iterations on the given dataset. The detailed discussion of these hybrid algorithms is presented in the following subsections.

4.1 The *KMFCM* algorithm

The proposed hybrid clustering algorithm *KMFCM* first performs the K-Means (*KM*) technique completely on the given dataset. Using the resulted cluster centers of *KM* as cluster seeds, the *FCM* is performed on the given dataset till termination. Here, to run the first iteration of the *FCM*, the cluster centers and the membership matrix are calculated based on the results of *KM*. The remaining iterations continue as in the *FCM* algorithm.

The algorithm for the *KMFCM* is given below. Here, *KM*-Step is the K-Means step and *FCM*-Step is the Fuzzy C-Means step.

***KMFCM* algorithm**

1. ***KM*-Step**: select k vectors randomly from the dataset as the initial cluster centers μ_j ($j = 1, \ldots, k$). Set the current iteration $t = 0$.
2. Assign each data object X_i to its closest cluster center using Eq. (1).
3. Update cluster centers μ_j ($j = 1, \ldots, k$) and set $t = t + 1$.
4. Compute *Percentage change* using Eq. (2).
5. If *Percentage change* $\geq \alpha$, repeat steps 2–4.
6. ***FCM*-Step**: compute the membership matrix $U^{(t)}_{k \text{ X } N}$ using Eq. (4) based on the results of *KM*-Step.
7. Assign data objects to clusters using membership matrix.
8. For each cluster C_j, compute the center μ_j ($j = 1, \ldots, k$) using Eq. (3)
9. Compute *Percentage change* using Eq. (2).
10. Stop the process if *Percentage change* $< \alpha$. Otherwise, set $t = t + 1$ and repeat steps 6–9.

4.2 The *KMandFCM* algorithm

Clustering in *KMandFCM* is performed by executing K-Means and the *FCM* techniques in alternative iterations on the given dataset till termination. The first iteration is performed using K-Means assuming some randomly selected data points as cluster centers. The second iteration is performed using *FCM* technique. For this iteration the cluster means, covariance matrices and the membership matrix are calculated using the results of first iteration. Third iteration is performed using K-Means technique. This iteration computes cluster means using results obtained from the second iteration. In this way, clustering is performed using K-Means and *FCM* in the alternative iterations till termination.

The algorithm for the proposed *KMandFCM* algorithm is given below. Here, *KM*-Step is the K-Means step and *FCM*-Step is the Fuzzy C-Means step.

***KM and FCM* algorithm**

1.Select k vectors randomly from the dataset as initial cluster centers μ_j ($j = 1, \ldots, k$). Set the current iteration $t = 0$.

2.***KM*-Step**: assign each vector X_i to its closest cluster center using Eq. (1).

3.***FCM*-Step**: set $t = t + 1$.

4.For each cluster C_j, compute the center μ_j using Eq. (3)

5.Compute the new membership matrix $U^{(t)}_{k \text{ X } N}$ using Eq. (4)

6.Assign data objects to clusters using the membership matrix.

7.Compute *Percentage change* using Eq. (2).

8.Stop the process if *Percentage change* $< \alpha$. Otherwise, set $t = t + 1$.

9.***KM*-Step**: For each cluster C_j, compute new center μ_j using Eq. (3).

10.Assign each vector X_i to its closest cluster center using Eq. (1).

11.Compute *Percentage change* using Eq. (2).

12.Stop the process if *Percentage change* $< \alpha$. Otherwise, go to step 3.

For all the algorithms, i.e., *KM*, *FCM*, *KMFCM*, *KMandFCM*, the same termination condition, Eq. (2), is used.

5. Performance evaluation measures

For performance evaluation of algorithms, CPU time in seconds, sum of squared error [12] and clustering fitness [43] are taken into consideration and are calculated for all the algorithms.

5.1 Sum of squared errors

The objective of clustering is to minimize the within-cluster sum of squared error (SSE). The lesser the SSE, the better the goodness of fit is. The sum of squared error [12] for the results of each clustering algorithm is computed using the Eq. (5)

$$SSE = \sum_{j=1}^{k} \sum_{X_i \in C_j} \left(X_i - \mu_j\right)^2 \tag{5}$$

Here, X_i is the ith data object in the dataset, μ_j (j = 1, …, k) is the center of the cluster C_j, and k is the number of clusters.

5.2 Clustering fitness

The main objective of any clustering algorithm is to generate clusters with higher intra-cluster similarity and lower inter-cluster similarity. So, it is also important to consider inter-cluster similarity while evaluating the clustering performance. In the present work, clustering fitness is also considered as a performance criterion, which requires the calculation of both intra-cluster similarity and inter-cluster similarity. The computation of clustering fitness also requires the experiential knowledge, λ. The computation of clustering fitness results in higher value when the inter-cluster similarity is low and results in lower value for when the inter-cluster similarity is high. Also that to make the computation of clustering fitness unbiased, the value of λ is taken as 0.5 [43].

(a) Intra-cluster similarity for the cluster C_j: it can be quantified via a function of the reciprocals of intra-cluster radii within each of the resulting clusters. The intra-cluster similarity [43] of a cluster C_j ($1 = j = k$), denoted as $S_{tra}(C_j)$, is defined in Eq. (6)

$$S_{tra}\left(C_j\right) = \frac{1+n}{1+\sum_1^n dist(I_l, Centroid)} \tag{6}$$

Here, n is the number of items in cluster C_j, I_j ($1 = j = n$) is the jth item in cluster C_j, and dist(I_j, *Centroid*) calculates the distance between I_j and the centroid of C_j, which is the intra-cluster radius of C_j. To smooth the value of $S_{tra}(C_j)$ and allow for possible singleton clusters, 1 is added to the denominator and numerator.

(b) Intra-cluster similarity for one clustering result C: it is denoted as $S_{tra}(C)$. It is defined in Eq. (7), [43]

$$S_{tra}(C) = \frac{\sum_1^k S_{tra}\left(C_j\right)}{k} \tag{7}$$

Here, k is the number of resulting clusters in C and $S_{tra}(C_j)$ is the intra-cluster similarity for the cluster C_j.

(c) Inter-cluster similarity: it can be quantified via a function of the reciprocals of inter-cluster radii of the clustering centroids. The inter-cluster similarity [43] for one of the possible clustering results C, denoted as $S_{ter}(C_j)$, is defined as Eq. (8)

$$S_{ter}(C) = \frac{1+k}{1+\sum_1^k dist\left(Centroid_j, Centroid^2\right)} \tag{8}$$

Here, k is the number of resulting clusters in C, $1 = j = k$, $Centroid_j$ is the centroid of the jth cluster in C, $Centroid^2$ is the centroid of all centroids of clusters in C. We compute inter-cluster radius of $Centroid_j$ by calculating dist($Centroid_j$, $Centroid^2$), which is distance between $Centroid_j$, and $Centroid^2$. To smooth the value of $S_{ter}(C)$

and allow for possible all-inclusive clustering result, 1 is added to the denominator and the numerator.

(d) Clustering fitness: the clustering fitness [43] for one of the possible clustering results C, denoted as CF, is defined as Eq. (9)

$$CF = \lambda \times S_{tra}(C) + \frac{1-\lambda}{S_{ter}(C)} \tag{9}$$

Here, λ $(0 < \lambda < 1)$ is an experiential weight, $S_{tra}(C)$ is the intra-cluster similarity for the clustering result C and $S_{ter}(C)$ is the inter-cluster similarity for the clustering result C. To avoid biasedness in our experiments, λ is assumed to be 0.5.

6. Experiments and results

Experimental work has been carried out on the system with Intel(R) Core(TM) i3-5005U CPU@2.00GHz processor speed, 4GB RAM, Windows 7 OS (64-bit) and using JDK1.7.0_45. Separate modules are written for each of the above discussed methods to observe the CPU time for clustering any dataset by keeping the cluster seeds same for all methods. I/O operations are eliminated and the CPU time observed is strictly for clustering of the data.

Along with the newly developed hybrid algorithms, experiments are also conducted with the algorithms for standard K-Means (*KM*) and Fuzzy C-Means (*FCM*) for performance comparison. All the algorithms are executed using datasets that are related to four images. The details of these images are available in **Table 1**.

SNO	Image	Resolution	No. of points	No. of dimensions
1	Heart	341 × 367	125,147	3
2	Kidneys	473 × 355	167,915	3
3	Baboon	512 × 512	262,144	3
4	Lena	256 × 256	65,536	3

Table 1.
Medical Images.

The medical images used in the present experiment are heart image [44] and kidneys image [45] (**Figures 1** and **2**). The experiments are also carried out using two benchmark images. They are Baboon and Lena images [46] (**Figures 3** and **4**).

Below is the brief description of medical images.

The Heart is a medical image obtained from biology data repository [44]. It is in "jpeg" format. The 'Kidneys' is a colored MRI scan of a coronal section through a human abdomen, showing the front view of healthy kidneys and liver [45]. It is in 'jpeg' format. The Baboon and Lena are benchmark test images that are found frequently in the literature [46]. These are all in uncompressed "tif" format.

All the algorithms for standard K-Means (*KM*), standard Fuzzy C-Means (*FCM*), *KMFCM* and *KMandFCM* are executed on each image data with varying number of clusters (k = 10, 11, 12, 13, 14, 15). For all algorithms, same cluster seeds are used. Same termination condition Eq. (2) is used for all the experiments. The details of CPU time, clustering fitness and SSE of each algorithm for the all images are given in the following sub-sections (**Tables 2–13**). The results are also projected in their respective graphs (**Figures 5–16**).

6.1 Observations with Heart image

K	*KM*	*FCM*	*KMFCM*	*KM and FCM*
10	0.21	0.30	1.36	0.19
11	0.21	0.32	1.48	0.20
12	0.25	0.40	1.61	0.20
13	0.09	0.35	1.58	0.22
14	0.14	0.39	1.73	0.23
15	0.36	0.43	2.15	0.26

Table 2.
CPU time of each clustering technique (Heart image).

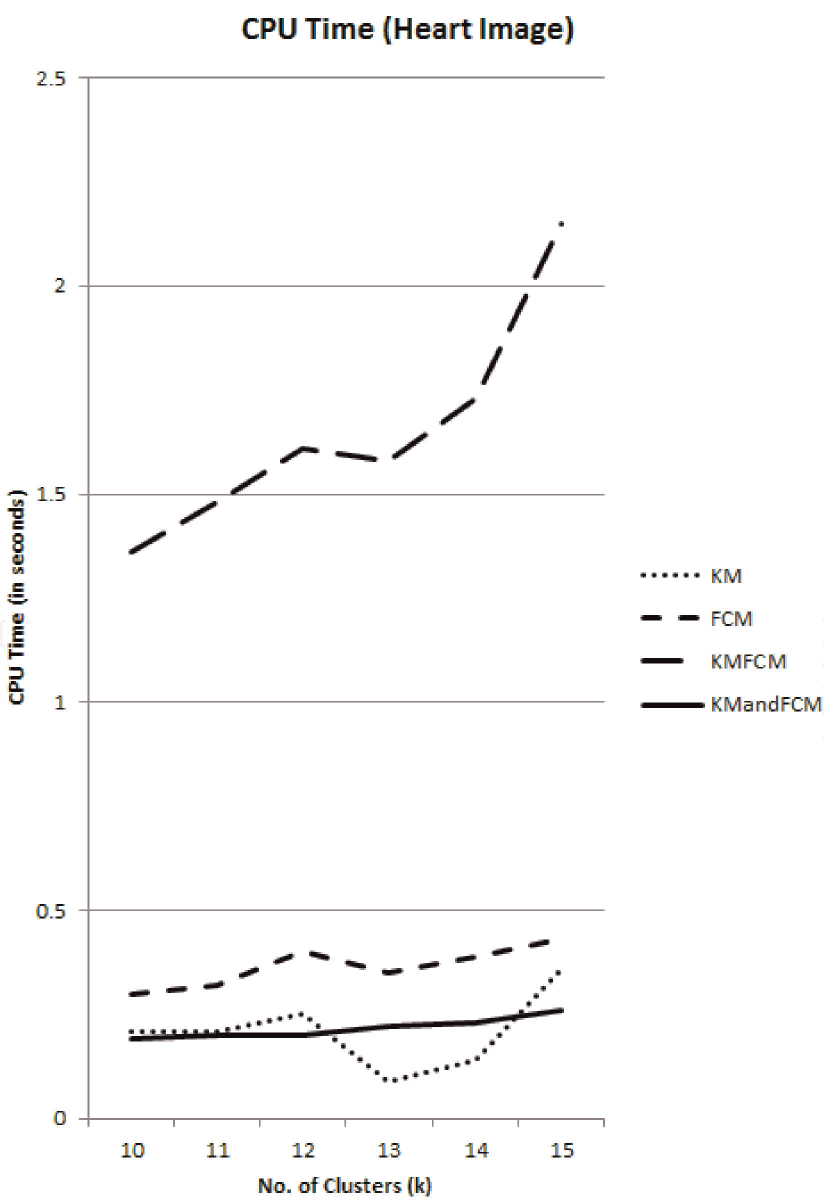

Figure 1.
CPU time (Heart image).

K	*KM*	*FCM*	*KMFCM*	*KM and FCM*
10	51.20	56.62	58.51	64.78
11	49.79	55.73	55.40	62.14
12	42.27	55.80	61.16	65.97
13	34.88	47.54	41.08	58.46
14	48.34	55.22	56.62	60.35
15	47.54	57.96	48.24	59.22

Table 3.
Clustering fitness of each clustering technique (Heart image).

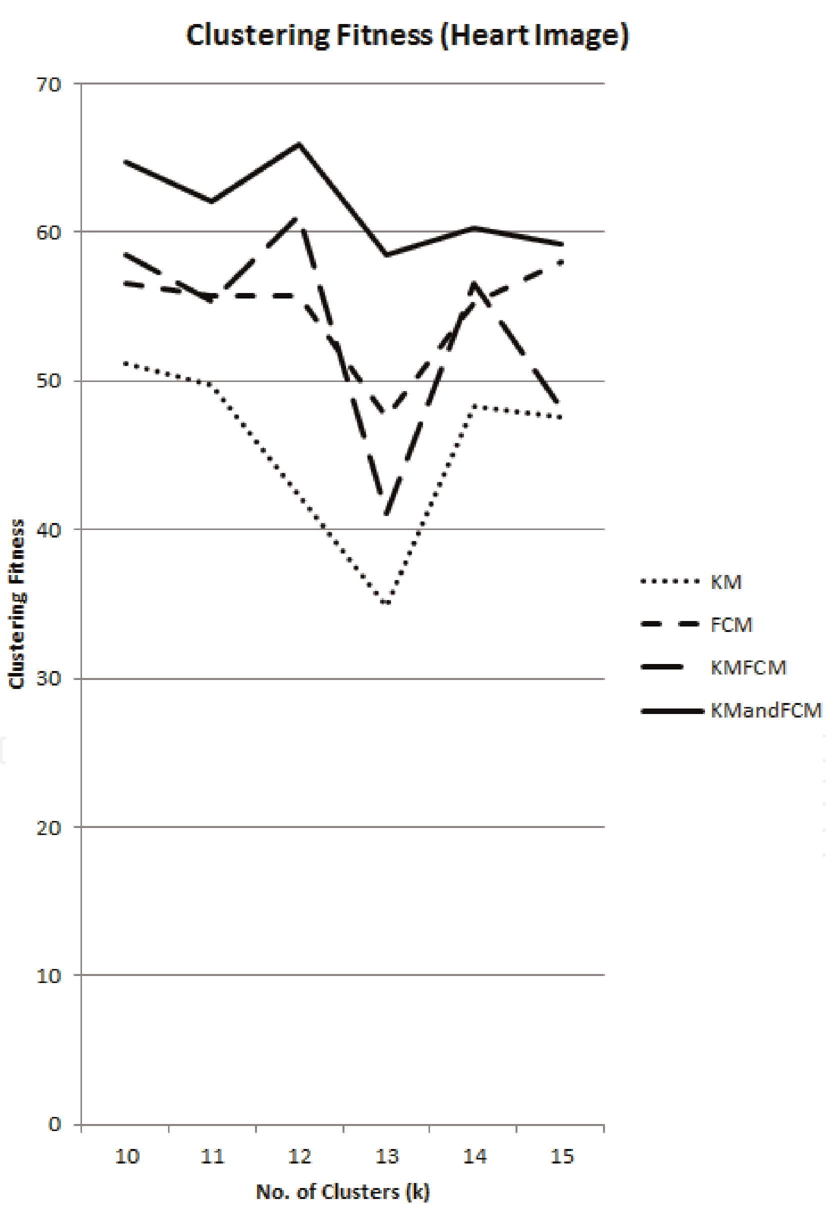

Figure 2.
Clustering Fitness (Heart image).

K	*KM*	*FCM*	*KMFCM*	*KM and FCM*
10	0.0163	0.0152	0.0148	0.0041
11	0.0150	0.0145	0.0074	0.0036
12	0.0173	0.0163	0.0059	0.0031
13	0.0185	0.0171	0.0285	0.0037
14	0.0142	0.0139	0.0113	0.0028
15	0.0138	0.0114	0.0241	0.0024

Table 4.
SSE of each clustering technique (Heart image).

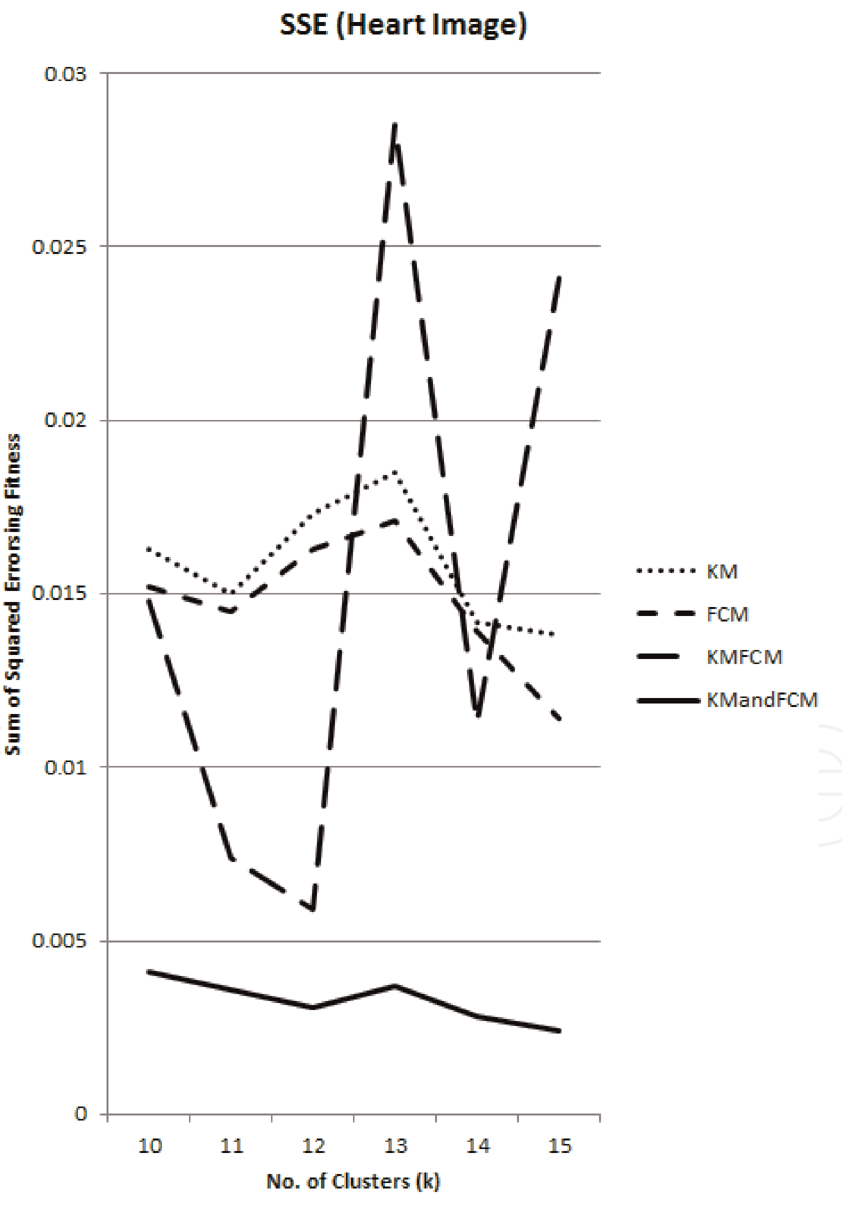

Figure 3.
Sum of squared errors (Heart image).

6.2 Observations with Kidneys image

K	*KM*	*FCM*	*KMFCM*	*KM and FCM*
10	0.09	0.68	1.58	0.55
11	0.13	0.41	1.83	0.26
12	0.81	0.58	2.64	0.46
13	0.08	0.47	2.07	0.30
14	0.24	0.60	2.40	0.31
15	0.65	1.78	2.22	1.06

Table 5.
CPU time of each clustering technique (Kidneys image).

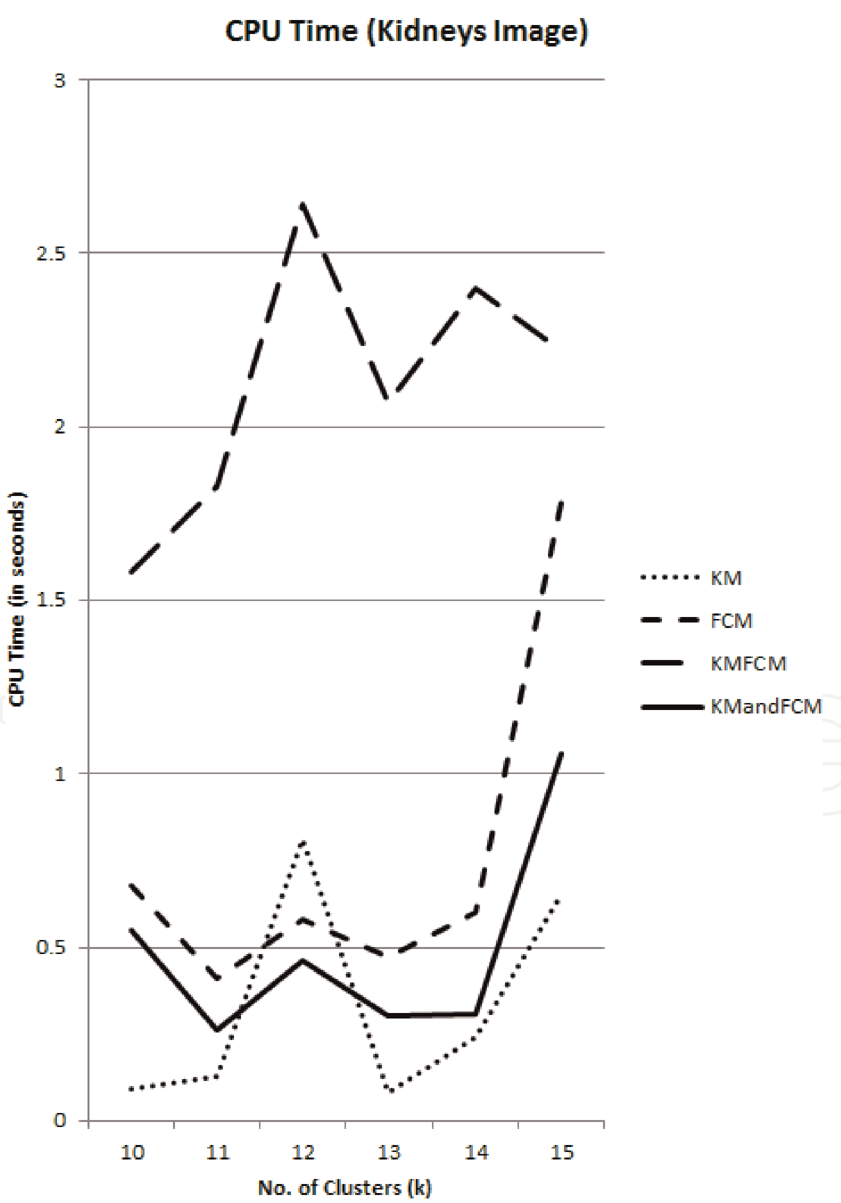

Figure 4.
CPU time (Kidneys image).

K	*KM*	*FCM*	*KMFCM*	*KM and FCM*
10	38.40	47.15	54.76	61.48
11	42.11	49.43	57.86	65.84
12	52.41	61.03	60.00	65.41
13	41.20	51.04	48.73	56.79
14	57.49	64.85	64.88	71.59
15	53.10	61.40	62.85	66.42

Table 6.
Clustering fitness of each clustering technique (Kidneys image).

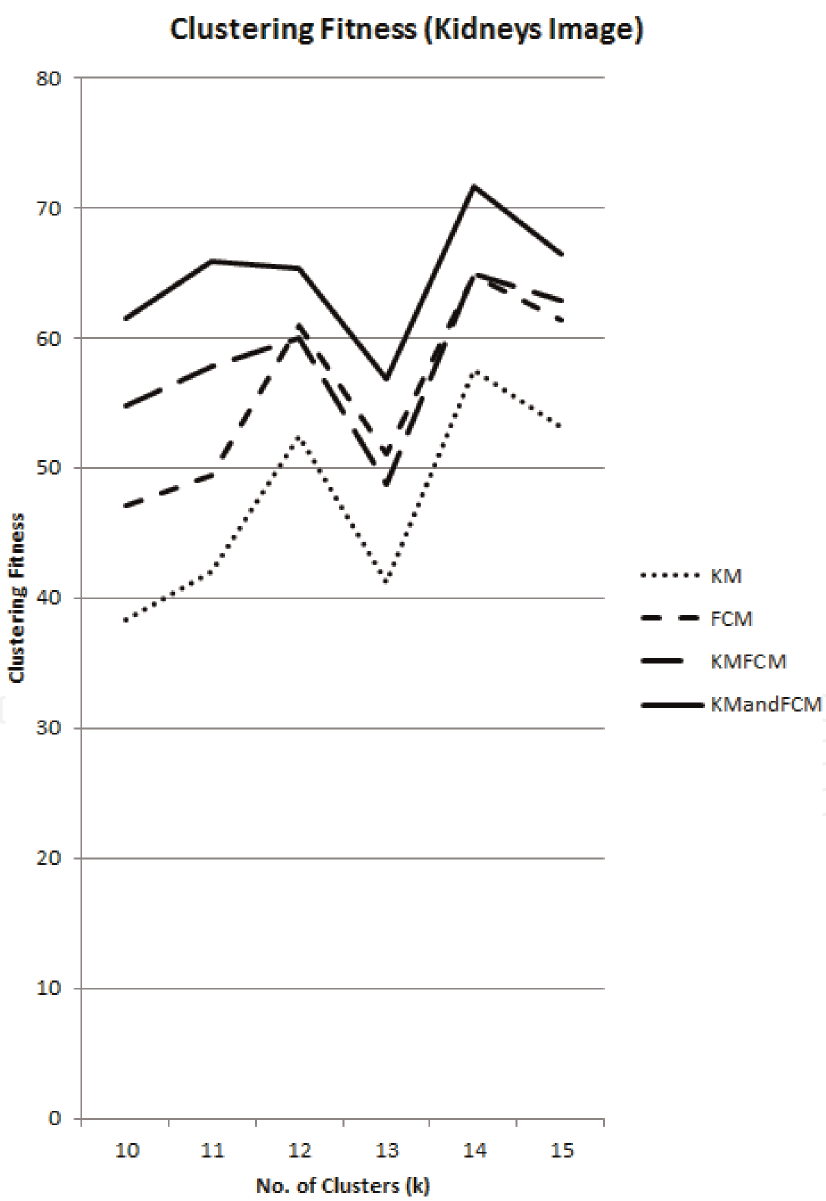

Figure 5.
Clustering Fitness (Kidneys image).

K	*KM*	*FCM*	*KMFCM*	*KM and FCM*
10	0.0281	0.0215	0.0129	0.0075
11	0.0265	0.0172	0.0114	0.0054
12	0.0249	0.0109	0.0140	0.0029
13	0.0123	0.0109	0.0191	0.0112
14	0.0144	0.0090	0.0067	0.0037
15	0.0115	0.0045	0.0028	0.0011

Table 7.
SSE of each clustering technique (Kidneys image).

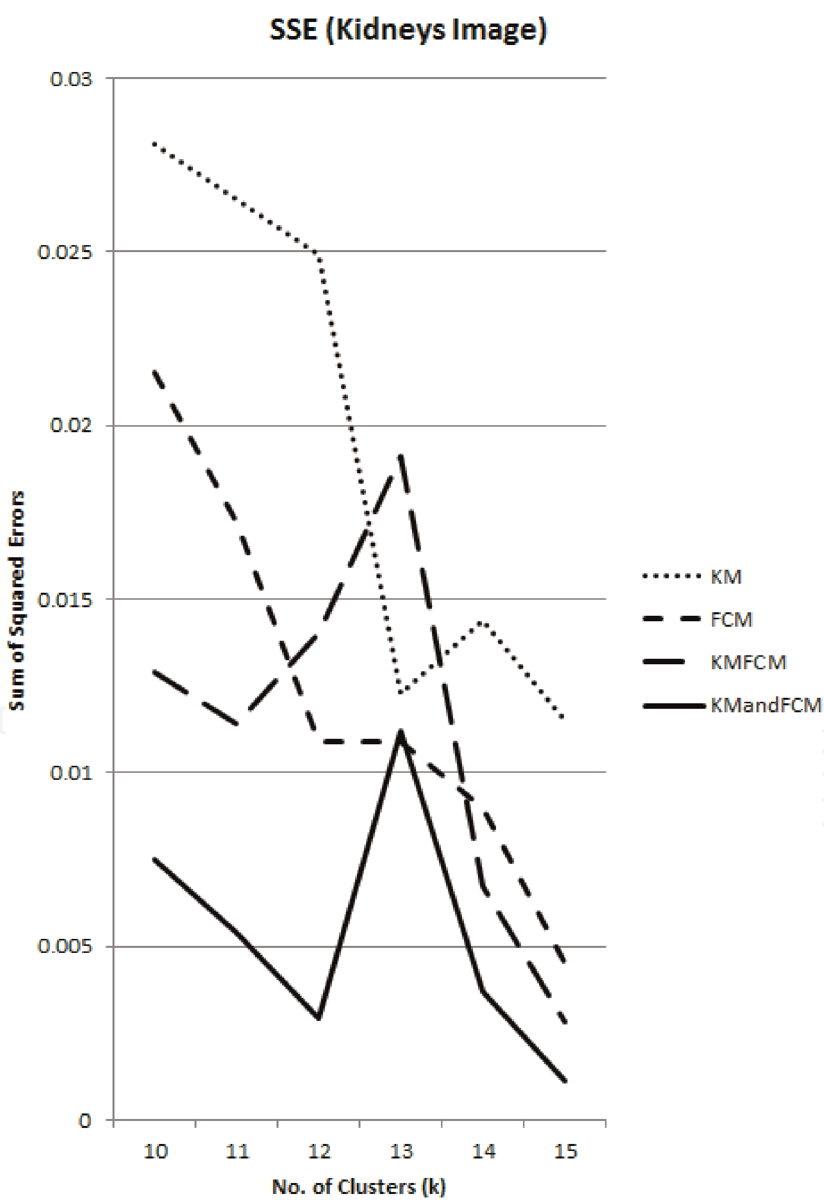

Figure 6.
Sum of squared errors (Kidneys image).

6.3 Observations with Baboon image

K	*KM*	*FCM*	*KMFCM*	*KM and FCM*
10	0.14	0.79	2.16	0.62
11	0.16	0.86	2.37	0.63
12	0.29	0.91	2.68	0.63
13	0.31	1.01	2.91	0.50
14	0.36	0.72	3.14	0.78
15	0.48	1.10	3.24	0.55

Table 8.
CPU time of each clustering method (Baboon image).

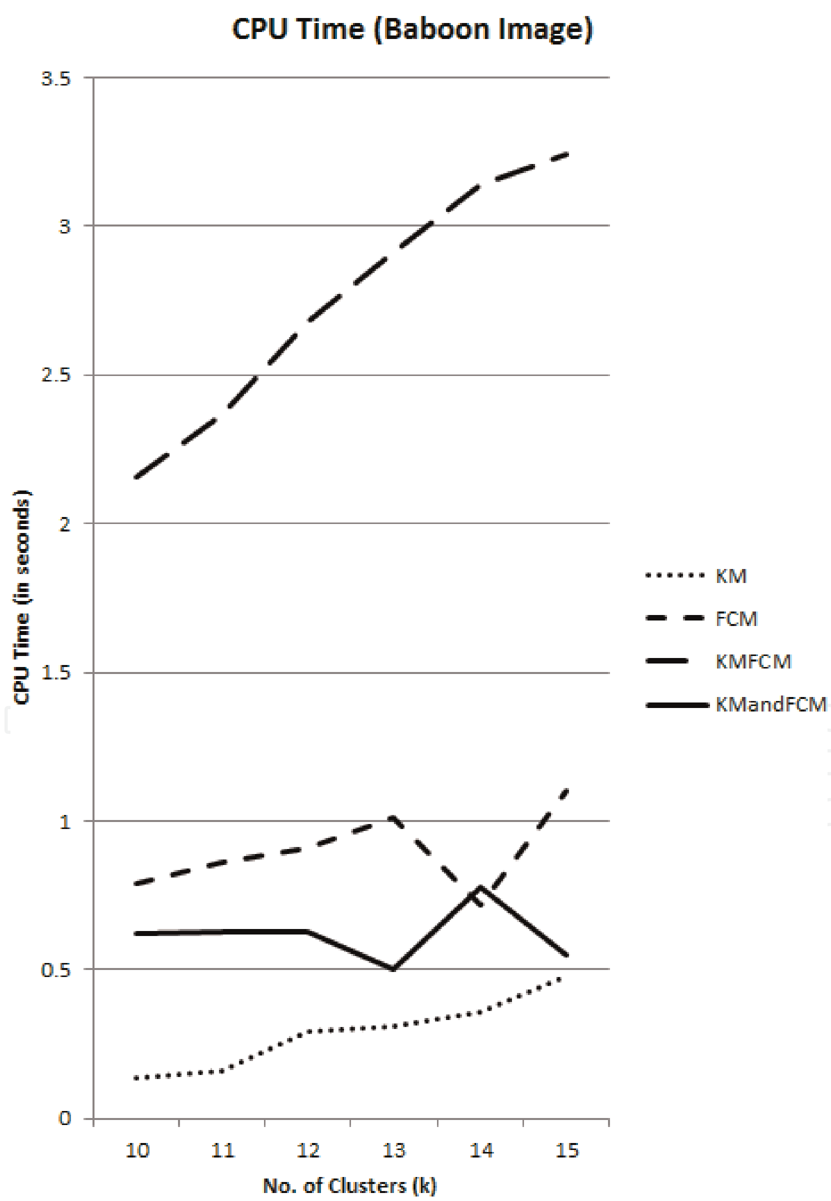

Figure 7.
CPU time (Baboon image).

K	KM	FCM	KMFCM	KM and FCM
10	30.22	32.17	36.02	39.07
11	22.28	29.71	37.36	39.49
12	28.70	32.63	35.13	39.57
13	31.28	33.47	40.39	42.28
14	25.92	29.49	37.77	39.81
15	36.48	38.16	34.43	39.98

Table 9.
Clustering fitness of each clustering method (Baboon image).

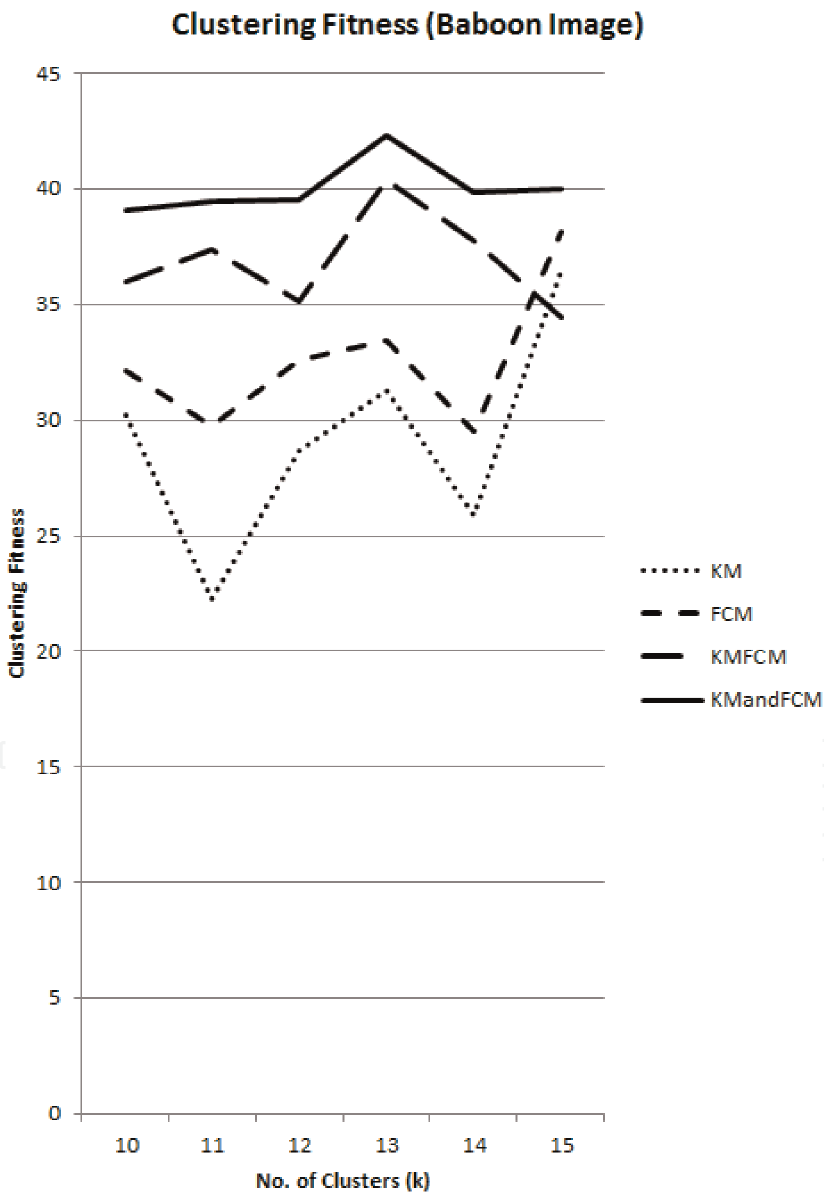

Figure 8.
Clustering fitness (Baboon image).

K	*KM*	*FCM*	*KMFCM*	*KM and FCM*
10	0.0080	0.0063	0.0059	0.0030
11	0.0073	0.0068	0.0037	0.0024
12	0.0099	0.0071	0.0053	0.0029
13	0.0065	0.0058	0.0070	0.0025
14	0.0087	0.0070	0.0041	0.0022
15	0.0069	0.0056	0.0027	0.0019

Table 10.
SSE of each clustering method (Baboon image).

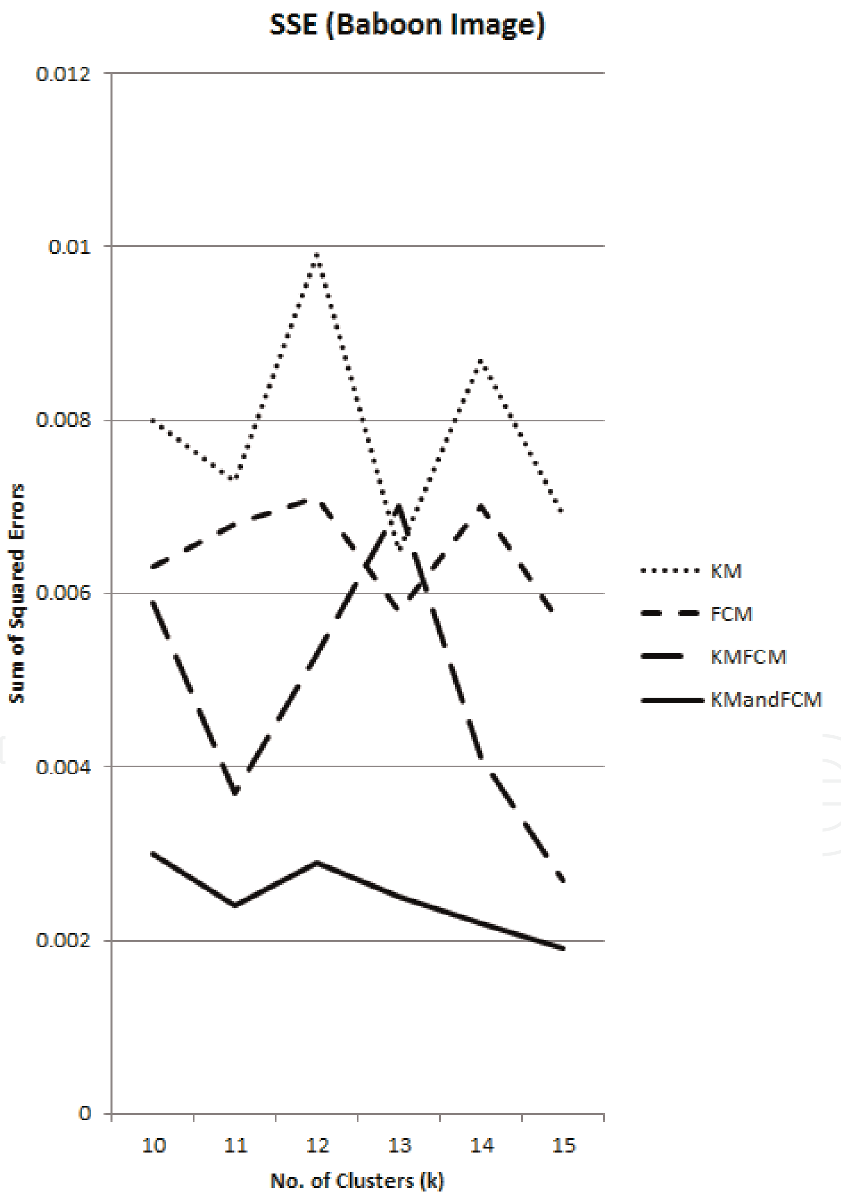

Figure 9.
Sum of squared errors (Baboon image).

6.4 Observations with Lena image

K	*KM*	*FCM*	*KMFCM*	*KMandFCM*
10	0.08	0.15	0.66	0.09
11	0.13	0.44	0.76	0.32
12	0.06	0.17	0.77	0.11
13	0.09	0.40	0.84	0.32
14	0.05	0.20	0.92	0.13
15	0.21	0.24	1.09	0.14

Table 11.
CPU time of each clustering method (Lena image).

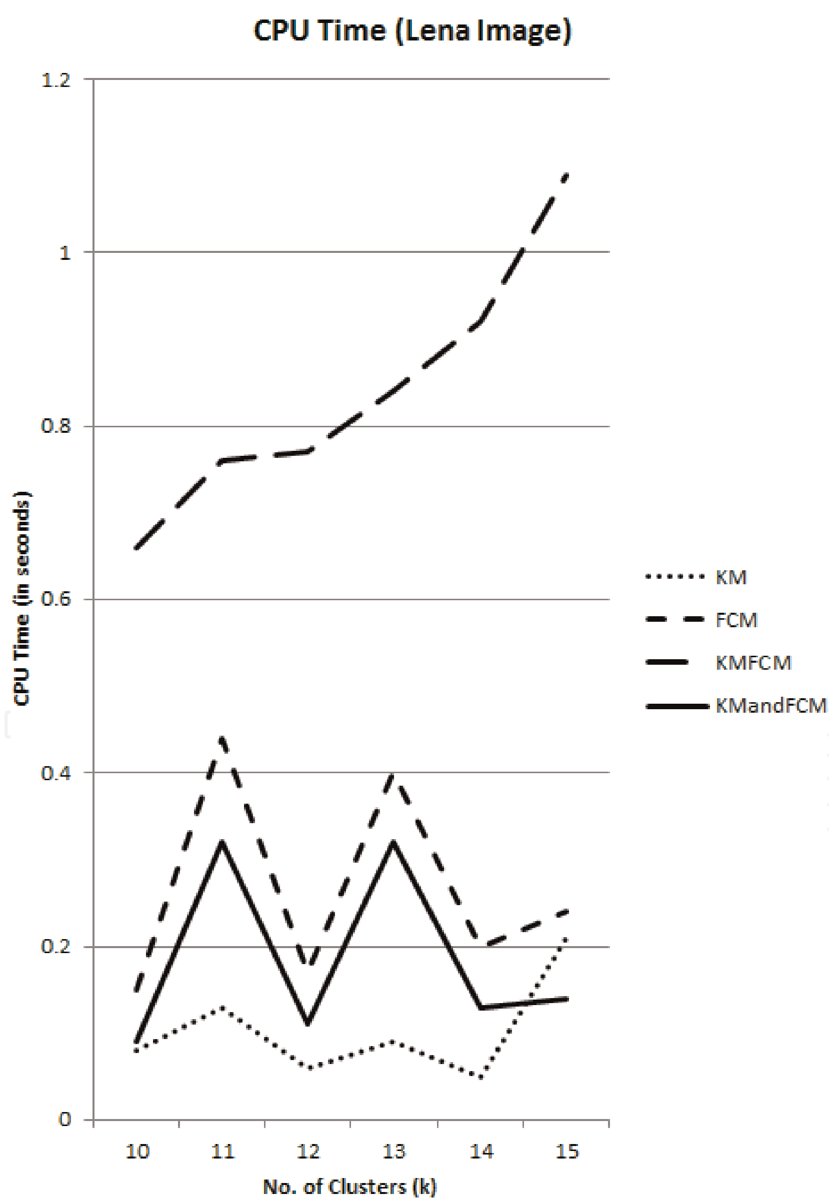

Figure 10.
CPU time (Lena image).

K	KM	FCM	KMFCM	KM and FCM
10	25.50	28.80	30.61	32.79
11	22.97	25.52	27.95	31.08
12	20.22	23.38	25.44	29.97
13	28.71	30.13	32.74	34.26
14	26.75	29.83	31.05	33.27
15	23.70	30.19	32.79	34.60

Table 12.
Clustering fitness of each clustering method (Lena image).

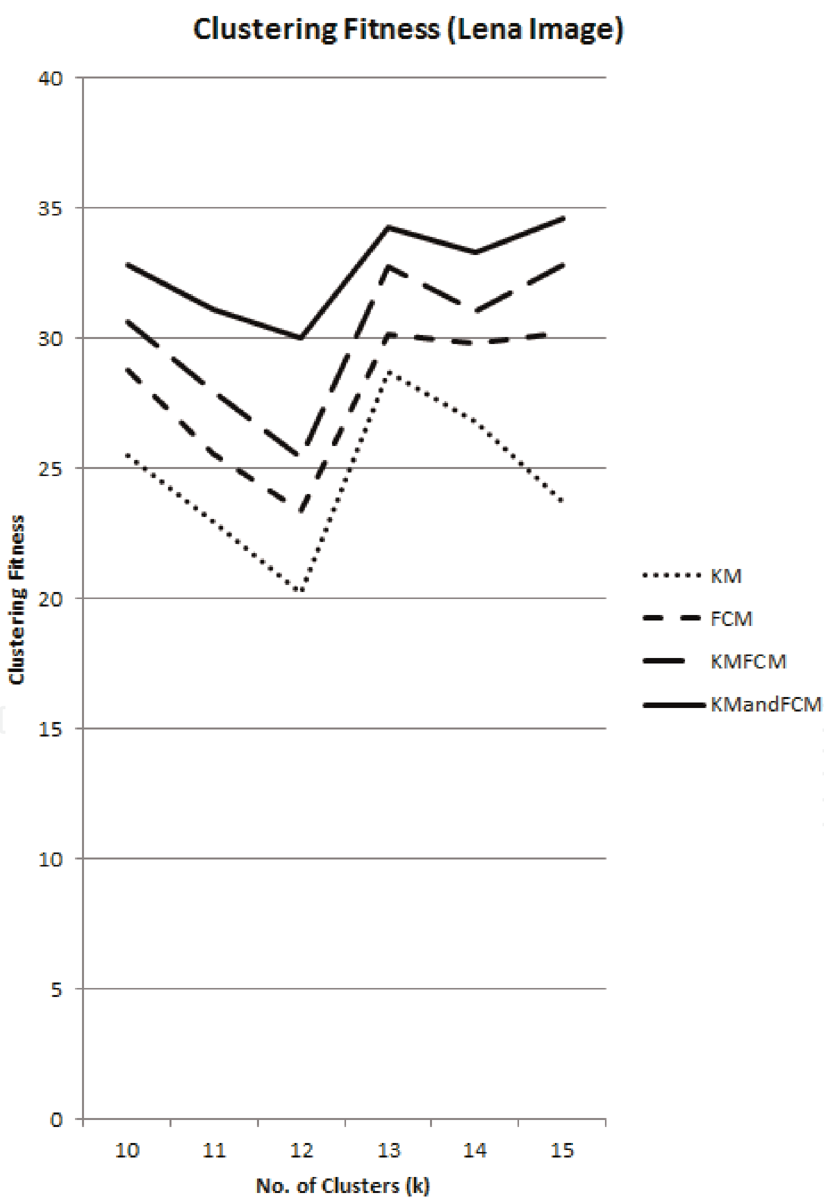

Figure 11.
Clustering fitness (Lena image).

K	KM	FCM	KMFCM	KM and FCM
10	0.0147	0.0127	0.0093	0.0034
11	0.0245	0.0218	0.0099	0.0041
12	0.0246	0.0178	0.0077	0.0034
13	0.0144	0.0106	0.0060	0.0027
14	0.0135	0.0110	0.0062	0.0024
15	0.0130	0.0100	0.0049	0.0022

Table 13.
SSE of each clustering method (Lena image).

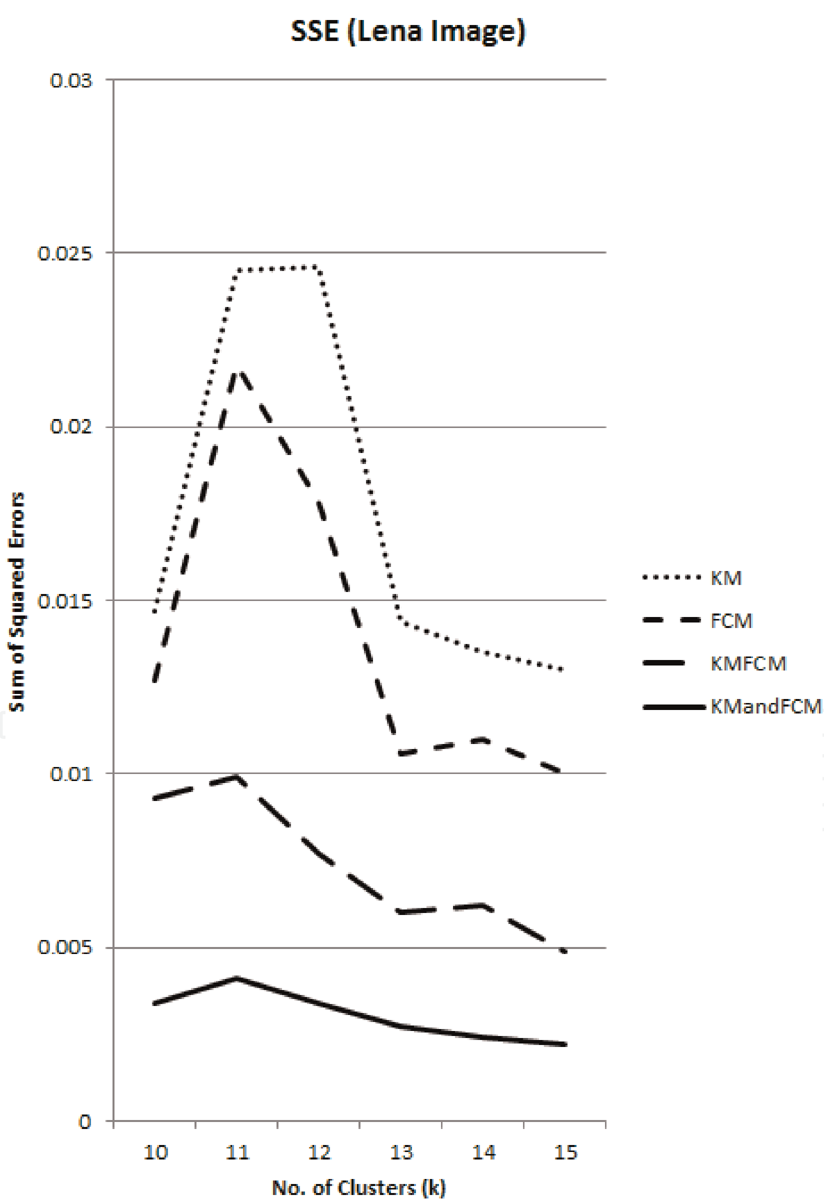

Figure 12.
Sum of squared errors (Lena image).

6.5 Original images used for present experimentation

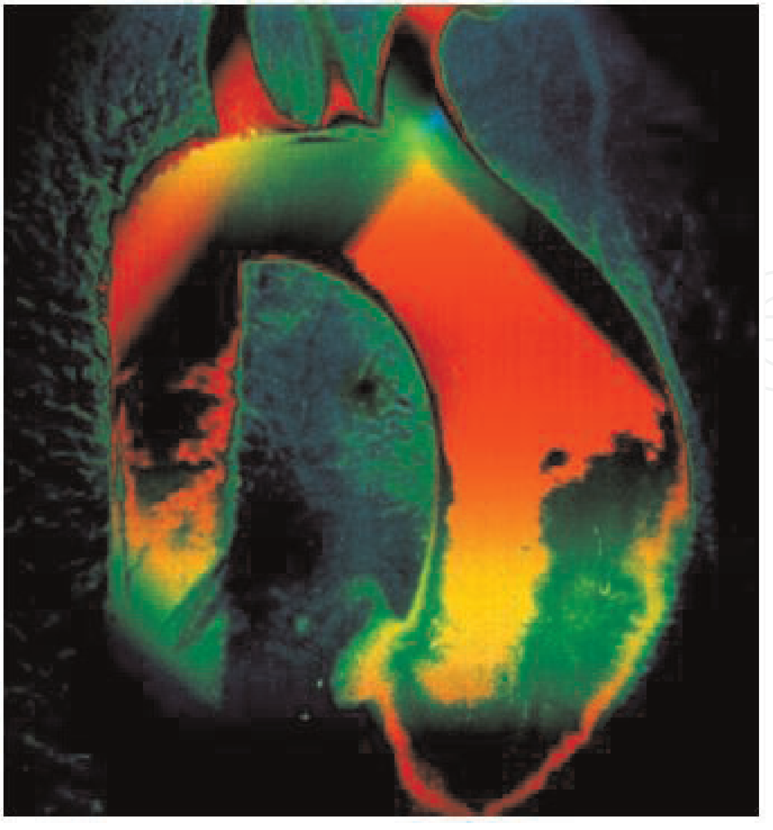

Figure 13.
Heart image.

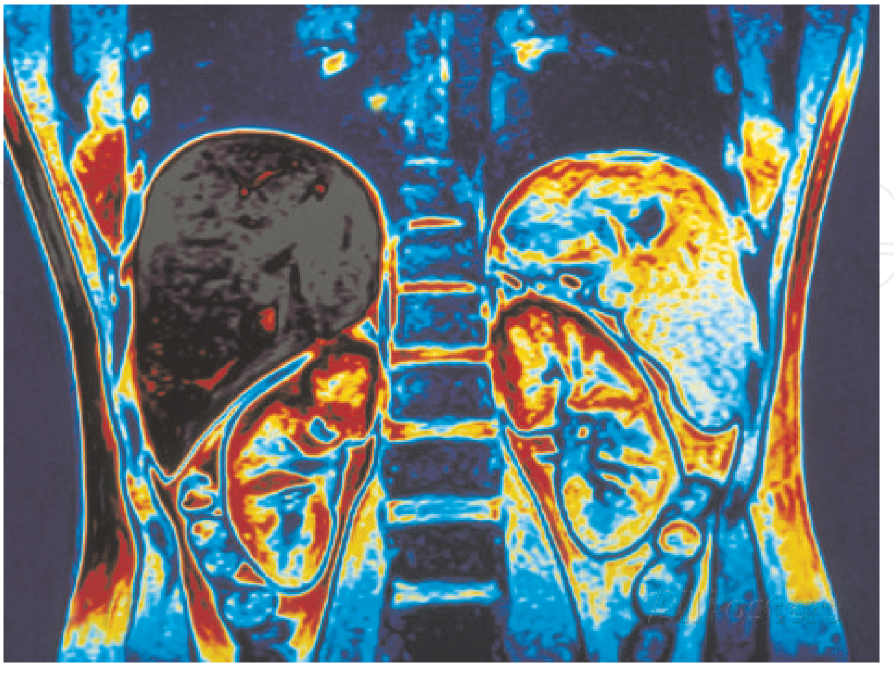

Figure 14.
Kidneys image.

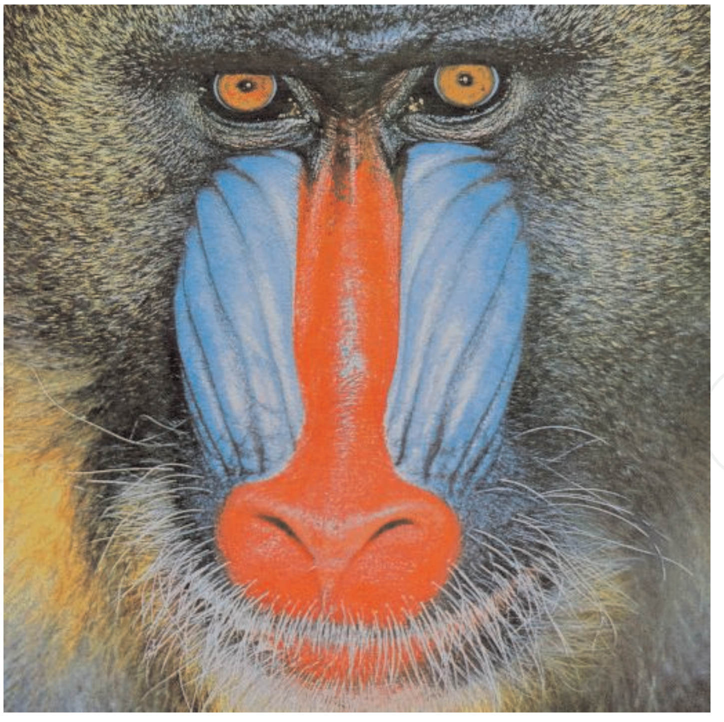

Figure 15.
Baboon image.

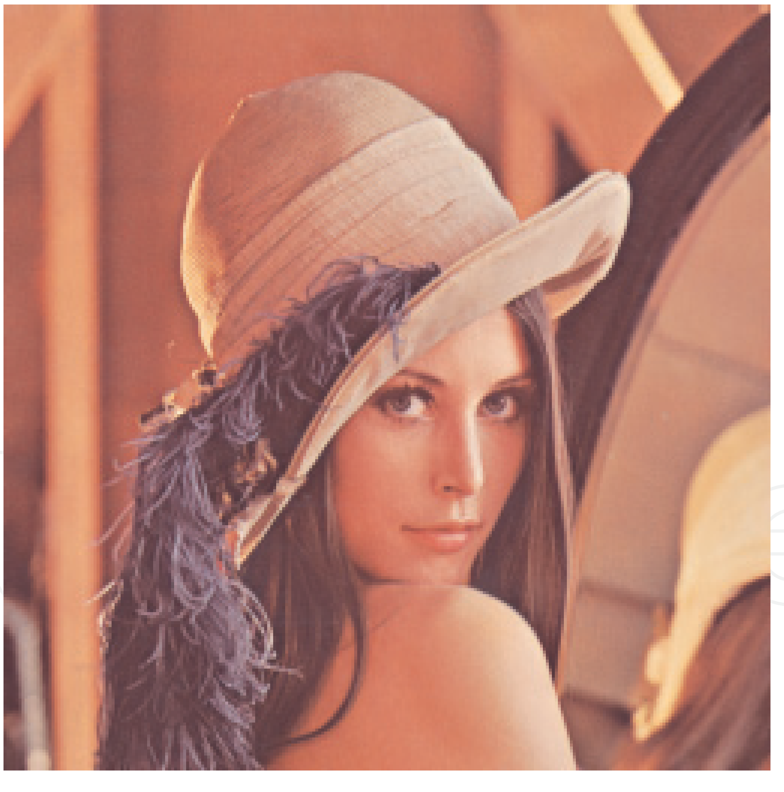

Figure 16.
Lena image.

6.6 Comparison of segmentation results on Baboon image

As an example of the present experiments for image segmentation, segmentation results for Baboon image for 10 clusters are presented here. These results are generated by the above proposed hybrid clustering algorithms along with the standard K-Means and standard *FCM* algorithms.

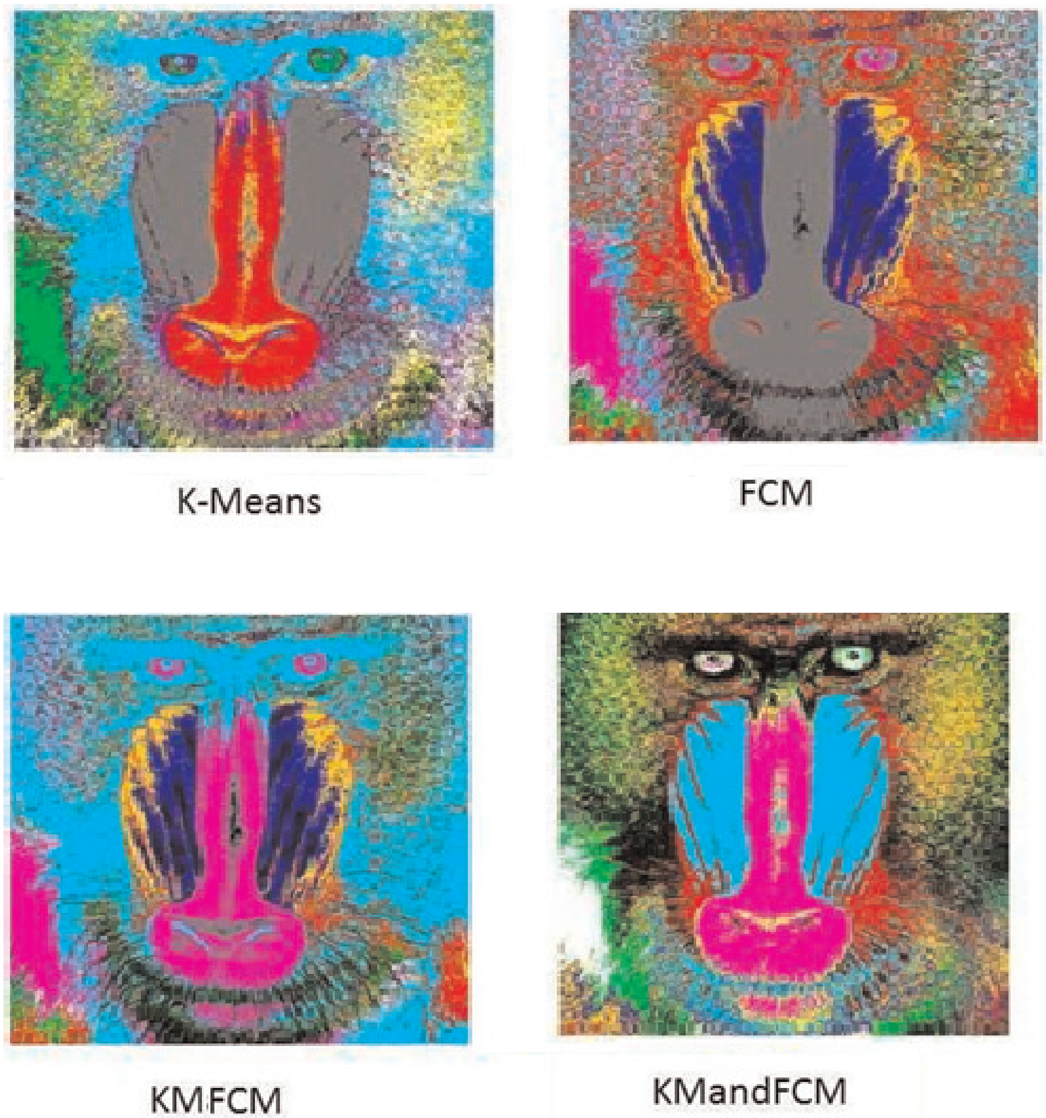

Figure 17.
Image segmentation results for Baboon image (for 10 Clusters).

For segmentation, here, each algorithm is executed using Baboon image data assuming that the number of clusters is 10, i.e., k = 10. Each segment is represented by each cluster. Separate color code is assigned to each cluster. The color codes are red, yellow, green, blue, orange, black, white, gray, cyan and magenta. The projections of all segmentation results generated by the algorithms are shown in **Figure 17**. The original Baboon image also shown in the figure.

In all the experiments, it is observed that hybrid clustering algorithm *KMandFCM* is showing better performance in terms of CPU, clustering fitness and SSE than the other algorithms.

7. Conclusion

The present chapter notably includes the study of hybridization of popular clustering algorithms, K-Means and *FCM*, and identifies the best hybridization strategy. All experiments are carried out for segmenting four images, which include two medical images also. For all the algorithms CPU time, clustering fitness and sum of squared error (SSE) are taken into consideration while carrying out their performance evaluation. In all the experiments that are conducted, the proposed hybrid algorithm *KMandFCM* is exhibiting better performance in terms of CPU time, Clustering Fitness (CF) and SSE.

In all experiments, it is also observed that the proposed hybrid clustering algorithms are showing better performance than the standard K-Means and *FCM* algorithms. Especially the *KMandFCM* algorithm has good results when compared to all other algorithms. Thus, it could be concluded that the hybrid clustering algorithm *KMandFCM* will have good application in other fields too.

Author details

Raja Kishor Duggirala
Department of CSE, Dr. L. Bullayya College of Engineering (for Women), Visakhapatnam, A.P., India

*Address all correspondence to: rajakisgor@gmail.com

References

[1] Kaltri K, Mahjoub M. Image segmentation by gaussian mixture models and modified *FCM* algorithm. The International Arab Journal of Information Technology. 2014;**11**(1): 11-18

[2] Wang S, Geng Z, Zhang J, Chen Y, Wang J. A fuzzy C-means model based on the spatial structural information for brain MRI segmentation. International Journal of Signal Processing, Image Processing and Pattern Recognition. 2014;**7**(1):313-322

[3] Khan AM, Ravi S. Image segmentation methods: A comparative study. International Journal of Soft Computing and Engineering (IJSCE). 2013;**3**(4):84-92. ISSN: 2231-2307

[4] Fwu J-K, Djuric PM. EM algorithm for image segmentation initialized by a tree structure scheme. IEEE Transactions on Image Processing. 1997;**6**(2):349-352

[5] Bezdek JC, Hall LO, Clarke LP. Review of MR image segmentation techniques using pattern recognition. Medical Physics. 1993;**20**:1033-1048

[6] Pham DL, Xu CY, Prince JL. A survey of current methods in medical image segmentation. Annual Review of Biomedical Engineering. 2000;**2**:315-337

[7] Wells WM, LGrimson WE, Kikinis R, Arrdrige SR. Adative segmentation of MRI data. IEEE Transactions on Medical Imaging. 1996;**15**:429-442

[8] Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association. 2002; **97**(458):611, ABI/INFORM Global

[9] Hannah IH, Selva KS. Hybrid TRS-FA clustering approach for Web2.0 social tagging system. International Journal of Rough Sets and Data Analysis (IJRSDA). 2015;**2**(1):70-87. DOI: 10.4018/ijrsda.2015010105

[10] Nizar Banu PK, Andrews S. Performance analysis of hard and soft clustering approaches for gene expression data. International Journal of Rough Sets and Data Analysis (IJRSDA). 2015;**2**(1):58-69. DOI: 10.4018/ijrsda.2015010104

[11] Ayramo S, Karkkainen T. Introduction to partitioning-based clustering methods with a robust example (Tech. Rep. No. C. 1/2006). Agora, Finland: University of Jyvaskyla, Department of Mathematical Information Technology; 2006

[12] Han J, Kamber M. Data Mining Concepts and Techniques, 2/e. New Delhi, India: Elsevier Inc; 2007

[13] Tan P-N, Steinbach M, Kumar V. Introduction to Data Mining, 1/e. New Delhi, India: Pearson Education; 2007

[14] Zadeh LA. Fuzzy sets. Information and Control. 1965;**8**(3):338-353

[15] Lemiare J. Fuzzy insurance. Astin Bulletin. 1990;**20**(1):33-55

[16] Chen S, Zhang D. Robust image segmentation using *FCM* with spatial constraints based on new kernel-induced distance measure. IEEE Transactions on Systems, Man, and Cybernetics. 1998;**34**:1907-1916

[17] Tsai A, Zhang J, Willsky AS. Expectation-maximization algorithms for image processing using multiscale models and mean-field theory, with applications to laser radar range profiling and segmentation. Optical Engineering. 2001;**40**(7):1287-1301

[18] Yang Y. Image segmentation by fuzzy C-means clustering algorithm

with a novel penalty term. Computing and Informatics. 2007;**26**:17-31

[19] Tolias YA, Panas SM. On applying spatial constraints in fuzzy image clustering using a fuzzy rule-based system. IEEE Signal Processing Letters. 1998;**5**:245-247

[20] Noordam JC, Van Den Broek WHAM, Buydens LMC. Geometrically guided Fuzzy C-Means clustering for multivariate image segmentation. In: Proc. International Conference on Pattern Recognition. Vol. 1. 2000. pp. 462-465

[21] Ahmed MN, Yamany SM, Mohamed N, Farag AA, Moriarty T. A modified fuzzy C-means algorithm for Bias field estimation and segmentation of MRI data. IEEE Transactions on Medical Imaging. 2002;**21**:193-199

[22] Zhang DQ, Chen SC, Pan ZS, Tan KR. Kernel-based Fuzzy clustering incorporating spatial constraints for image segmentation. In: Proceedings of International Conference on Machine Learning and Cybernetics; Vol. 4. 2003. pp. 2189-2192

[23] Li X, Li L, Lu H, Chen D, Liang Z. Inhomogeneity correction for magnetic resonance images with Fuzzy C-Mean algorithm. In: Proc. SPIE Int. Soc. Opt. Eng. Vol. 5032. 2003. pp. 995-1005

[24] Pham DL, Prince JL. Adaptive fuzzy segmentation of magnetic resonance images. IEEE Transactions on Medical Imaging. 1999;**18**:737-752

[25] Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, et al. Top 10 algorithms in data mining. Knowledge and Information Systems. 2007;**14**:1-37 (survey paper: DIO 10.1007/s10115-007-0114-2)

[26] Aggarwal N, Aggarwal K. A mid-point based k-mean clustering algorithm for data mining. International Journal on Computer Science and Engineering. 2012;**4**(6):1174-1180

[27] Das S. Pattern recognition using fuzzy c-means technique, international journal of energy. Information and Communications. 2013;**4**(1):1-14

[28] Lu Y, Ma T, Yin C, Xie X, Tian W, Zhong SM. Implementation of the fuzzy C-means clustering algorithm in meteorological data. International Journal of Database Theory and Application. 2013;**6**(6):1-18

[29] Bezdek JC, Ehrlich R, Full W. *FCM*: The fuzzy c-means clustering algorithm. Computers & Geosciences. 1984;**10** (2–3):191-203

[30] Klir GJ, Yuan B. Fuzzy Sets and Fuzzy Logic: Theory and Applications. Upper Saddle River, New Jersey: Prentice Hall of India Private Limited; 2005

[31] Bezdek JC, Ehrlich R, Full W. FCM: The Fuzzy c-Means Clustering Algorithm. Computers & Geosciences. 1984;**10**(2-3):191-203

[32] Bezdek JC. Feature selection for binary data-Medical diagnosis with fuzzy sets. In: Proc. Nat. Comput. Conf. AFIPS Press; 1972. pp. 1057-1068

[33] Cannon RL, Dave JV, Bezdek JC. Efficient implementation of fuzzy c-means clustering algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986;**8**(2):248-255

[34] Cannon RL, Jacobs C. Multispectral pixel classification with fuzzy objective functions. Cen. for Automation Res., Univ. Maryland, College Park, Tech. Rep. CAR-TR-51. 1984

[35] Gong M, Liang Y, Ma W, Ma J. Fuzzy C-means clustering with local information and kernel metric for image segmentation. 2013;**22**(2):573-584

[36] Krinidis S, Krinidis M, Chatzis V. Fast and robust fuzzy active contours. IEEE Transactions on Image Processing. 2010;**19**(5):1328-1337

[37] Yong Y, Chongxun Z, Pan L. A novel fuzzy C-means clustering algorithm for image thresholding. Measurement Science Review. 2004; **4**(1):11-19

[38] Hanesch M, Scholger R, Dekkers MJ. The application of fuzzy C-means cluster analysis and non-linear mapping to a soil data set for the detection of polluted sites. Physics and Chemistry of the Earth (Part A). 2001;**26**(11–12): 885-891. DOI: S1464-1895(01)00137-5

[39] Adebisi AA, Olusayo OE, Olatunde OS. An exploratory study of K-Means and expectation maximization algorithms. British Journal of Mathematics and Computer Science. 2012;**2**(2):62-71

[40] Kearns M, Mansour Y, Andrew YNg. An Information-theoretic analysis of hard and soft assignment methods for clustering. In: Uncertainty in Artificial Intelligence, 1997. pp. 282-293

[41] Ghosh S, Dubey SK. Comparative analysis of K-means and Fuzzy c-means algorithms. International Journal of Advanced Computer Science and Applications. 2013;**4**(4):35-39

[42] Raja Kishor D, Venkateswarlu NB. A Behavioral study of some widely employed partitional and model-based clustering algorithms and their hybridizations. In: Proceedings of the International Conference on Data Engineering and Communication Technology, Advances in Intelligent Systems and Computing 469; Springer Nature; 2017. DOI 10.1007/978-981-10-1678-3_56

[43] Han X, Zhao T. Auto-K dynamic clustering algorithm. Journal of Animal and Veterinary Advances. 2005;**4**(5): 535-539

[44] http://www.biologyreference.com/Bl-Ce/Cardiovascular-Diseases.html

[45] http://www.allposters.com/-sp/Colour-MRI-Scan-of-Abdomen-Showing-Kidneys-Liver-Posters_i10024816_.htm?UPI=AP10024816_PC14258389_FI0_SV6_IT1_VRV1

[46] http://www.imageprocessingplace.com/root_files_V3/image_databases.htm