Performance Analysis of Deterministic Centroid Initialization Method for Partitional Algorithms in Image Block Clustering

B. Vinoth Kumar*, G. R. Karpagam and N. Vijaya Rekha

Department of CSE, PSG College of Technology, Coimbatore, Tamil Nadu, India; bvk@cse.psgtech.ac.in

Abstract

Image block clustering is important in several exploratory applications such as image segmentation, image pattern classification, image compression and the like. Clustering the blocks of an image into meaningful groups to reveal useful information is a challenging problem. This paper uses Partitional algorithm for image block clustering and also clusters are arranged in a hierarchical structure to explore the different aspects of the data. K Means algorithm is a widely used Partitional algorithm. But due to its gradient descent nature, the result of this algorithm is very sensitive to the initial cluster centroids which do not produce unique clustering results every time for the same input. Many initialization methods have been proposed to address this problem, but it produces results at the cost of high computational time. A Deterministic Centroid Initialization Method (DCIM) was proposed in our earlier work for K means Clustering algorithm which was used to cluster the image blocks for content adaptive image compression. In this paper, we extend the performance analysis part of DCIM in conventional K Means and Fuzzy C Means algorithm. The performance analysis has been done with the measures such as Root Mean Square Error (RMSE), Number of iterations and CPU time. The strength of this DCIM method is that the clustering algorithms require less number of iterations to attain convergence and in producing the unique better clustering result in a single run. Clustering algorithms with DCIM was tested on a variety of images to show its strength. The experimental results show that the clustering algorithm with DCIM outperforms the clustering algorithms with Random Centroid Initialization Method (RCIM) in terms of RMSE and number of iterations. From an average performance view, K Means with DCIM produces a decrease of 20.87% in RMSE with marginal increase of 0.27 seconds at CPU time than K Means with RCIM. The validation results prove that the DCIM guarantees unique better results within a lesser number of iterations with less computational effort.

Keywords: Cluster Centroid Initialization, Fuzzy C Means, Image Block Clustering, K Means, Partitional Clustering

1. Introduction

A digital image is composed of picture elements called pixels which are the smallest sample of an image. It is widely believed that an image is worth more than a thousand words. They are the best way to express the information about positions, sizes and interrelationships between objects1. Due to increase in size of image, image compression plays an important role. Conventional lossy image compression standard such as JPEG which are block transform based schemes give equal importance to all regions (or blocks) of an image and compress it with uniform compression ratio2. But this will not be appropriate when an image contains mixed content such as photographs, text and flat diagrams3. Thus a lot of research has been done to preserve the adaptive image quality of lossy image compression scheme which are generalized into three main categories such as quantization optimization4–6, frequency threshold optimization7,8 and Regions of Interest (ROI) based optimization9.

Clustering plays an important role in the above application. Clustering divides data into groups that are

*Author for correspondence
meaningful, useful, or both. If meaningful groups are the goal, then the cluster should capture the natural structure of the data. In some cases, however, cluster analysis is only a useful starting point for other purposes, such as data summarization. The notion of what constitutes a good cluster depends on the application and there are many methods for finding clusters subject to various criteria. Among clustering formulations that are based on minimizing a formal objective function, the most widely used and studied is k-means clustering which is a Partitional based Clustering. The reasons for preferring K means algorithm by many researchers are: it is simple, fast, requires low memory, flexibility and highly efficient.

Random initialization of initial centroids is a key issue in K means clustering algorithm which does not provide unique results every time and due to this, the algorithm should execute many times which increases computational effort. Emre Celebi et al performed a detailed survey on different initialization methods and they recommended deterministic approach to initialization of cluster centroids for the applications that demand unique clustering results every time. Also a review of different initialization methods for K means clustering done by Harmanpreet Singh et al state that nearly all of the methods surveyed by them are having high computational cost. Based on the above study, we proposed a deterministic method for initialization of centroids in the paper. This proposal made K means algorithm to execute only once and ensures unique results every time. Also the image block clusters were arranged in a hierarchical structure which reduces the computational complexity of various decision making algorithms in pattern classification.

Since the scope of our earlier work was to generate the image adaptive quantization table using knowledge based genetic algorithm approach for content based image compression, hence the analyzing part of the DCIM for K means algorithm was not provided in and also this initialization method can be used for other applications which uses Partitional clustering algorithms. Therefore, the objective of this paper is twofold, first to analyze the DCIM proposed in the paper and to use this initialization method for other Partitional clustering algorithm such as fuzzy c means.

The remainder of this paper is organized as follows. In the next section, the overviews of Partitional clustering methods are given. Image block clustering is illustrated in section 3. In section 4, the definition of performance criteria is given. The experimental setup and results are shown in section 5. Final thoughts and future work is summarized in the concluding section 6.

2. Partitional Clustering Algorithm

Partitional Clustering directly decomposes the dataset into a set of disjoint clusters. The clusters are formed to optimize a certain criterion function such as dissimilarity function based on distance, so that the objects within a cluster are “similar” to one another and “dissimilar” to objects in other clusters in terms of the data set attributes. The most widely used Partitional algorithms are K Means and Fuzzy C Means which is also called as hard and soft clustering respectively.

2.1 K Means Clustering

K-means (KM) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The “K” in its name refers to the fact that the algorithm looks for a fixed number of clusters which are defined in terms of proximity of data points to each other. Generally in k means clustering, initial centroids are chosen randomly. But randomly chosen centroids do not guarantee the uniqueness of the cluster results every time, hence user specified centroids are chosen for better results. These centroids should be selected in an appropriate way because different location causes different result. Hence they should be placed as possible as far away from each other. The points are grouped to the closest centroid by finding the distance between the point and the centroid using Euclidean or Manhattan distance formula. K means is formally described in algorithm 1 as given below.

Algorithm 1: Basic K-Means Algorithm
Input: a set of N data vectors (Data set), K (Number of Clusters)
Output: K Clusters

- Select K points randomly as initial cluster centroids.
- Assign each point in the data vector to the closest cluster based upon the Euclidean distance between each point and each cluster centroid.
- Each cluster centroid is recomputed as the average of the point in that cluster.
- Repeat steps 2 and 3 until the centroids no longer move.
2.2 Fuzzy C Means Clustering

Fuzzy C Means (FCM) is an extension of K-Means. It is one of the most widely used methods in fuzzy clustering. It uses fuzzy models in the clustering data, allowing all data to be a member of any cluster\textsuperscript{21}. Since it is an objective function based clustering algorithm, its main objective is to minimize the function shown in equation 1.

\[
J(U < V) = \sum_{i=1}^{n} \sum_{j=1}^{c} (\mu_{ij})^m ||x_i - v_j||^2
\]

(1)

Fuzzy C Means assigns a feature vector to a centroid with a certain degree of membership and features tend to overlap. Initial centroids are chosen randomly. Therefore different initial centroids produce different results which make FCM to be sensitive to initial centroids\textsuperscript{22}. This algorithm works by assigning membership to each data point corresponding to each cluster centroid on the basis of distance between the cluster centroid and the data point. More the data is near to the cluster centroid more is its membership towards the particular cluster centroid. Clearly, summation of the membership of each data point should be equal to one. The objective function value can be obtained through an iterative process where membership and cluster centroids are updated according to the equations 2 and 3.

\[
\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{d_{ij}}{d_{ik}} \right)^{2/m}}
\]

(2)

\[
v_j = \frac{\sum_{i=1}^{n} (\mu_{ij})^m x_i}{\sum_{i=1}^{n} (\mu_{ij})^m} \quad \forall j = 1, 2, ..., c
\]

(3)

where,
- ‘n’ is the number of data points
- ‘v’ represents the jth cluster center
- ‘m’ is the fuzziness index \(m \in [1, \infty]\)
- ‘c’ represents the number of cluster center
- ‘\mu’ represents the membership of i\textsuperscript{th} data to j\textsuperscript{th} cluster center.
- ‘d’ represents the Euclidean distance between i\textsuperscript{th} data and j\textsuperscript{th} cluster center.

Algorithm 2: Basic Fuzzy C-Means Algorithm

Input: a set of N data vectors (Data set), C (Number of Clusters)
Output: C Clusters

- Initialize membership value for data point.
- Compute the fuzzy centroid.
- Update the fuzzy membership value.
- If updated membership value is less than previous membership value stop, else return to step 2.
- Repeat until the centroid no longer move.

3. Image Block Clustering

The motivation of this paper is to cluster the image blocks in an image. Thus it is important to extract the features to differentiate the image blocks. In the literature, different features are extracted from both spatial domain\textsuperscript{23–27} and frequency domain\textsuperscript{28–30}. Transform coding is a widely used lossy image compression technique and hence, the features are extracted in the transformed domain to reduce the computational complexity. Though many image transforms are used for compression, Discrete Cosine Transform is used in the majority of the image compression standards such as JPEG and MPEG. Discrete Cosine Transformation (DCT) is used to transform the image from the spatial domain to frequency domain. The spatial domain shows the amplitude of the gray level when it is moving through space. The frequency domain shows how quickly the amplitude of the gray scale shade is changing from one pixel to the next in an image file. The frequency domain is a better representation for the data because it’s separate out the information that is not very important to human perception. DCT also preserves the properties such as energy compaction, and image data correlation\textsuperscript{31}. In frequency mapping, the low-order or “DC” term represents the average luminance values in each block, that generally dominates the content within the block, while AC coefficients reflects the variance of luminance. The higher-order “AC” terms represent the strength of rapid changes across the width or height of the block. The highest AC term represents the strength of a cosine wave alternating from maximum to minimum at adjacent pixels. Therefore DC coefficient and AC coefficients of a block gives the nature of the block and it is evident to employ DCT coefficients as feature for content based image processing applications\textsuperscript{32–36}.

Based on our experience, it is identified that if the DC coefficients and standard deviation of AC coefficients of two blocks differ slightly, then the pixels of those blocks will almost have the same values. Therefore the DC coefficient and standard deviation of AC coefficients are chosen as feature vectors and also they are used as representatives...
for their corresponding blocks. The clusters are formed as main (level 1) and sub clusters (level 2) in order to form two level structure of clusters where clusters in level 2 are nested in clusters in level 1. Here levels of hierarchical structure are decided based upon the dimension of feature vectors. Partitional clustering algorithm is applied over the extracted features at each level and as a result image blocks are clustered into a hierarchical structure. Main clusters are formed by grouping the DC coefficients and Sub clusters are formed by grouping the standard deviation of AC coefficients in each corresponding main cluster.

The drawbacks of choosing initial centroids as random in Partitional algorithm are described in introduction part. A simple deterministic centroid initialization method was proposed in the paper\(^4\) to choose the initial centroids for the Partitional algorithms. In this method, the given data vector is arranged in ascending order and then it is split into K bins randomly. The data which is occurring most frequently in each bin is considered as an initial centroid. The Deterministic Centroid initialization method and procedure for image block clustering are given in algorithm 3 and 4 respectively. The conceptual view of the image block clustering is also shown in Figure 1.

**Algorithm 3**: Deterministic Centroid Initialization Method

**Input**: a set of N data vectors (Data set), K (Number of Clusters)

**Output**: Initial Cluster Centroids

- Sort the data vectors in ascending order.
- Split the ordered vectors into K bins randomly.
- Calculate the mode for each bin
- Mode of each bin is considered as initial centroids.

**Algorithm 4**: Image Block Clustering procedure

**Input**: Gray Scale Image, M (Number of Main Clusters), N (Number of Sub Clusters)

**Output**: M x N Number of Image Block Clusters

- Get an image.
- An image is split into 8x8 non overlapping blocks.
- Transform each 8x8 block of the image to a frequency-domain representation, using a normalized, two-dimensional type –II Discrete Cosine Transform (DCT)
- Extract DC coefficients and Standard deviation of AC coefficients as features from each DCT block.
- Store them together as a vector.
- Group the DC coefficients into M clusters (Main Clusters) using Partitional algorithm. Choose the initial centroids using DCIM.
- Based on DC coefficient clusters, the corresponding image blocks are grouped into M clusters.
- Group the Standard deviation of AC coefficients in the corresponding clusters into N clusters (Sub Clusters) using Partitional algorithm. Choose the initial centroids using DCIM.

![Figure 1. Conceptual view of image block clustering.](image-url)
Based on Standard deviation of AC coefficients clusters, the corresponding image blocks are grouped into \( N \) clusters.

### 4. Performance Criteria

The main objective of the DCIM is to improve the effectiveness (quality) and efficiency (speed) of the Partitional algorithm. To analyze the effectiveness of the algorithm, the following measure is used.

i) **Root Mean Square Error** \( \text{RMSE} \); is given by the differences between the instances of each cluster and their cluster centroid. Smaller values indicate clusters of high quality.

To analyze the efficiency of the algorithm, the following measures are used.

ii) **Number of iterations**; gives the number of iterations that the algorithm requires to reach the convergence after initialization.

iii) **CPU time**; gives the total CPU time taken by the initialization and clustering phases.

### 5. Experimental Result and Discussion

Image block clustering algorithm is implemented on a dual core processor of 2.2 GHz each, with 3 GB of RAM. The algorithm is coded using MATLAB R2007b and we have used variety of test images shown in Figure 2 which are of size 256X256 and digitized to 256 gray levels. Figures 3, 4 and 5 show the graphical view of the image block clustering process. Figure 3 shows all the 8 X 8 blocks in a Lena image in DCT domain represented by their corresponding DC coefficient and standard deviation of AC coefficients. Image blocks are grouped into 5 main clusters based on the DC coefficient and it is shown in Figure 4. Then each main cluster block is grouped into 5 sub clusters based on the standard deviation of AC coefficients of the corresponding cluster. One of the sub clusters is shown in Figure 5. Different shapes are used to represent different clusters for better understanding.

Our objective is to analyze the performance of DCIM for Partitional algorithms such as K means and Fuzzy C means clustering algorithms. To evaluate the performance of the DCIM, it is compared with RCIM by employing three performance measures; RMSE, Number of iterations and CPU time as mentioned in section 6. Tables 1, 2, 3 and 4 show the practical clustering performance of the K Means with RCIM (KM-RCIM), K Means with DCIM (KM-DCIM), Fuzzy C Means with RCIM (FCM-RCIM) and Fuzzy C Means with DCIM (FCM-DCIM) respectively. The results shown for RCIM in both the Tables 1 and 3 are the best results from 20 independent runs with different random seeds whereas the results shown in Tables 2 and 4 for DCIM are unique clustering results in a single run.

Table 5 shows the average value of all performance measures for different clustering methods. Based upon the Table 5, the Figures 6, 7 and 8 are drawn where K means with RCIM and DCIM is denoted as a solid line and dashed line respectively while Fuzzy C Means with RCIM and DCIM is denoted as dash dot line and dotted line respectively. The following points are observed from the Figures 6, 7 and 8.

- It is noticed that with a respective of initialization methods, both K means and Fuzzy C Means achieve almost same RMSE value. K Means need more number of iterations for convergence than Fuzzy C Means. Although Fuzzy C means require less number of iterations, it takes more CPU time which shows the computational complexity of the algorithm.

- It is noticed that with irrespective of Clustering algorithms, DCIM outperforms RCIM in terms of RMSE and number of iterations. But due to computational complexity introduced by DCIM, CPU time is slightly larger than RCIM.

- If RCIM is used then with irrespective of clustering methods, number of iterations is not proportional to the number of clusters for both the clustering methods. But RMSE is inversely proportional to the number of clusters whereas CPU time is directly proportional to the number of clusters.

- If DCIM is used then with irrespective of clustering methods, both number of iterations and CPU time are directly proportional to the number of clusters while RMSE is inversely proportional to the number of clusters.

- The results presented for clustering methods with RCIM are best among 20 independent runs. Therefore these methods require more computational effort.

- Clustering methods with DCIM yields better results than RCIM in a single run and hence, it does not require more computational effort.
Figure 2. Test images (a) lena (b) camera man (c) barbara (d) montage (e) bridge (f) fruits (g) peppers (h) aerial.
Even though the new deterministic approach to find the initial centroids would increase the computational complexity, it will be compensated by yielding very less iterations for convergence; in turn CPU time of clustering algorithm using DCIM is slightly higher than the clustering algorithm using RCIM.

When the number of clusters is more, K Means with DCIM takes equal CPU time as K Means with RCIM.

From the above observations, it can be clearly understood that DCIM outperforms the RCIM and also K Means clustering algorithm performs well than Fuzzy C means algorithm. Therefore, it can be concluded that K means with DCIM produces better unique clustering results than other clustering algorithms. In particular, it achieves a decrease of 20.87% in RMSE with an increase of 0.27 seconds at CPU time than K Means with RCIM.

6. Conclusion

In this paper, we extended the performance analysis of DCIM proposed in our earlier work for hierarchical structured Partitional clustering algorithms. This deterministic method enhances the potentials of the Partitional clustering algorithms in producing the unique better clustering result in a single run and minimizes the number of iterations to reach the convergence. The reduced number of iterations shows that initial cluster centroids computed by DCIM are found to be very close to the desired cluster centroids. A comparative analysis has been made...
Table 3. Performance measures of Fuzzy C Means with RCIM for different number of clusters

| No. of Clusters | Measures | Lena | Camera Man | Barbara | Montage | Bridge | Fruits | Peppers | Aerial |
|-----------------|----------|------|------------|---------|---------|--------|--------|---------|--------|
| 50              | No. of iterations | 85   | 127        | 70      | 65      | 101    | 153    | 76      | 88     |
|                 | RMSE     | 1.9153 | 1.9754 | 1.1989 | 1.7986 | 1.5696 | 1.809  | 1.6972  | 1.6558 |
|                 | CPU time | 3.9023 | 4.56     | 2.9604 | 3.0852 | 3.7292 | 3.3045 | 3.553   | 3.0251 |
| 100             | No. of iterations | 174  | 156       | 97      | 81      | 136    | 165    | 121     | 101    |
|                 | RMSE     | 1.6265 | 1.5121 | 0.9832 | 1.3927 | 1.3682 | 1.7072 | 1.6574  | 1.3539 |
|                 | CPU time | 4.937  | 4.8286   | 3.5289 | 4.0982 | 4.4722 | 4.605  | 4.2438  | 3.0827 |
| 150             | No. of iterations | 107  | 216       | 107     | 78      | 148    | 190    | 93      | 144    |
|                 | RMSE     | 0.9924 | 0.6934  | 0.3808 | 0.4708 | 0.5626 | 0.697  | 0.5626  | 0.7688 |
|                 | CPU time | 5.0393 | 5.3498   | 3.6673 | 4.1632 | 4.5915 | 4.605  | 4.4753  | 3.2244 |
| 200             | No. of iterations | 82   | 119       | 97      | 87      | 145    | 85     | 114     | 113    |
|                 | RMSE     | 0.6866 | 0.6195  | 0.3808 | 0.4708 | 0.5626 | 0.697  | 0.5626  | 0.6282 |
|                 | CPU time | 5.0393 | 5.3498   | 3.6673 | 4.1632 | 4.5915 | 4.605  | 4.4753  | 3.2244 |

Table 4. Performance measures of Fuzzy C Means with DCIM for different number of clusters

| No. of Clusters | Measures | Lena | Camera Man | Barbara | Montage | Bridge | Fruits | Peppers | Aerial |
|-----------------|----------|------|------------|---------|---------|--------|--------|---------|--------|
| 50              | No. of iterations | 53   | 39         | 49      | 48      | 55     | 47     | 49      | 59     |
|                 | RMSE     | 1.6058 | 1.8444 | 1.1524 | 1.4969 | 1.4792 | 1.6814 | 1.6234  | 1.4413 |
|                 | CPU time | 4.3585 | 4.6437   | 3.4393 | 3.487 | 4.3799 | 3.813  | 4.1653  | 3.0835 |
| 100             | No. of iterations | 58   | 56         | 68      | 53      | 84     | 64     | 65      | 63     |
|                 | RMSE     | 1.2382 | 1.1407 | 0.9171 | 1.0507 | 1.1342 | 1.0355 | 1.3815  | 1.0697 |
|                 | CPU time | 5.7344 | 5.4194   | 3.5537 | 4.5607 | 5.5406 | 5.0934 | 5.2279  | 3.3201 |
| 150             | No. of iterations | 53   | 67         | 67      | 52      | 77     | 81     | 60      | 64     |
|                 | RMSE     | 0.6859 | 0.5203  | 0.5453 | 0.5274 | 0.6387 | 0.6925 | 0.7973  | 0.5924 |
|                 | CPU time | 5.744  | 5.5529   | 3.6789 | 4.6808 | 5.6142 | 5.3513 | 5.275   | 3.4996 |
| 200             | No. of iterations | 71   | 86         | 64      | 48      | 66     | 67     | 72      | 65     |
|                 | RMSE     | 0.4384 | 0.2769  | 0.3709 | 0.3338 | 0.4232 | 0.4617 | 0.4675  | 0.3949 |
|                 | CPU time | 5.942  | 5.6137   | 3.875  | 5.0932 | 5.639  | 5.5269 | 5.3982  | 3.6131 |

Table 5. Summary table of performance measures for different clustering methods

| S.No | Method   | Number of Clusters | Number of Iterations | RMSE  | CPU Time |
|------|----------|--------------------|----------------------|-------|----------|
| 1    | KM –RCIM | 50                 | 138.63               | 1.67  | 2.46     |
|      |          | 100                | 167.13               | 1.35  | 2.57     |
|      |          | 150                | 153.5                | 0.83  | 2.62     |
|      |          | 200                | 134.38               | 0.55  | 3.05     |
| 2    | KM – DCIM| 50                 | 64.75                | 1.52  | 2.82     |
|      |          | 100                | 86.75                | 1.12  | 2.89     |
|      |          | 150                | 92.75                | 0.61  | 2.97     |
|      |          | 200                | 97.63                | 0.39  | 3.08     |
| 3    | FCM-RCIM | 50                 | 95.63                | 1.7   | 3.51     |
|      |          | 100                | 128.81               | 1.4   | 4.22     |
|      |          | 150                | 135.38               | 0.83  | 4.31     |
|      |          | 200                | 105.25               | 0.59  | 4.48     |
| 4    | FCM – DCIM| 50                 | 49.88                | 1.54  | 3.92     |
|      |          | 100                | 63.88                | 1.12  | 4.81     |
|      |          | 150                | 65.13                | 0.62  | 4.92     |
|      |          | 200                | 67.38                | 0.4   | 5.09     |
Figure 3. DCT blocks.

Figure 4. Main clusters.

Figure 5. Sub clusters.

Figure 6. Number of iterations vs. number of clusters.

Figure 7. RMSE vs. number of clusters.

Figure 8. CPU time vs. number of clusters.
between DCIM and RCIM in K Means and Fuzzy C Means with RMSE, Number of iterations and CPU time as the performance measures. This analysis shows that K means with DCIM has a faster convergence rate with a better RMSE value, but it takes a slight increase in CPU time than K Means with RCIM. As a future scope, the Partitional clustering algorithm with DCIM can be used in the application which needs unique clustering results every time.

7. References

1. Ahmed AM, Sharadqueh. Linear model of resolution and quality of digital images. Contemporary Engineering Sciences. 2012; 5 (6):273–9.
2. Kumar BV, Karpagam GR. An empirical analysis of requantization errors for recompressed JPEG images. IJEST. 2011; 3(12):8519–27.
3. Wong A, Bishop W. A flexible content based approach to adaptive image compression. IEEE International Conference on Multimedia and Expo; 2006 July 9-12; Toronto: IEEE; 2006. p. 713–6.
4. Kumar BV, Karpagam GR. Knowledge based genetic algorithm approach to quantization table generation for the JPEG baseline algorithm. Turk J Electr Eng Co. In press.
5. Huizhu M, Quiju Z. Research on cultural-based multi-objective particle swarm optimization in image compression quality assessment. Optik. 2013; 124:957–61.
6. Beatrice L, Francesco M, Massimo V. A multi-objective evolutionary approach to image quality/compression ratio trade-off in JPEG baseline algorithm. Appl Soft Comput. 2010; 10(14):548–61.
7. Abu NA, Ernawan F, Suryana N. A generic psycho visual error threshold for the quantization table generation on JPEG image compression. 9th International Colloquium of Signal Processing and its Applications; 2013 March 8-10; Kuala Lumpur, Malaysia. New York: IEEE; 2013. p. 39–43.
8. Bonyadi MR, Dehghani E, Moghaddam ME. A non-uniform image compression using genetic algorithm. International conference on systems, Signals and Image Processing; 2008 Jun 25-28; Bratislava, Slovak Republic. New York: IEEE; 2008. p. 315–8.
9. Deng C, Lin W, Cai J. Content-based image compression for arbitrary-resolution display devices. IEEE T Multimedia. 2012; 14(4):1127–39.
10. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An efficient k-means clustering algorithm analysis and implementation. IEEE T Pattern Anal. 2002; 24:881–92.
11. Velmurugan T. Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data. Appl Soft Comput. 2014; 19:134–46.
12. Zhang C, Fang Z. An improved K-means clustering algorithm. J Inform Comput Sci. 2013; 10(1):193–9.
13. Dhanabal S, Chandramathi S. An efficient K-Means initialization using Minimum-Average-Maximum (MAM) method. Asian J Inform Tech. 2013; 12(2):77–82.
14. Devi DMR, Thambidurai P. Similarity measurement in recent biased time series databases using different clustering methods. Indian Journal of Science and Technology. 2014; 7(2):189–98.
15. Celebi ME, Kingravi HA, Vela PA. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl. 2013; 40:200–10.
16. Singh H, Kaur K. Review of existing methods for finding initial clusters in K-means algorithm. Int J Comput Appl. 2013: 68(14):24–8.
17. Kumar R, Arthanaree AM. Performance evaluation and comparative analysis of proposed image segmentation algorithm. Indian Journal of Science and Technology. 2014; 7(1):39–47.
18. Vivekanandan K, Krishnakumari P. Discrete wavelet transformation of an image based on genetic-algorithm clustering. Indian Journal of Science and Technology. 2008; 1(3):1–5.
19. Han J, Kamber M. Data mining: concepts and techniques. 2nd ed. Boston: Morgan Kaufmann Publisher Elsevier Inc; 2006.
20. Pena JM, Lozano JM, Larranaga P. An empirical comparison of four initialization methods for the K-Means algorithm. Pattern Recog Lett. 1999; 20:1027–40.
21. Sheshasayee A, Sharmila P. Comparative study of fuzzy C means and K Means algorithm for requirements clustering. Indian Journal of Science and Technology; 2014: 7(6):853–7.
22. Fan J, Han M, Wang J. Single point iterative weighted fuzzy C-means clustering algorithm for remote sensing image segmentation. Pattern Recog. 2009; 42:2527–40.
23. Alexandropoulos T, Loumos V, Kayafas E. A block-based clustering technique for realtime objects detection on a static background. Second IEEE International Conference on Intelligent Systems; 2004 Jun 22–24; New York: IEEE; 2004. p. 169–73.
24. Sekeh MA, Maarof MA, Rohani MF, Motiei M. Sequential straightforward clustering for local image block matching. World Academy of Science, Engineering and Technology. 2011; 50:693–7.
25. Sekeh MA, Maarof MA, Rohani MF, Mahdian B. Efficient image duplicated region detection model using sequential block clustering. Digit Investig. 2013; 10:73–84.
26. Radke RJ, Andra S, Al-Kofahi O, Roysam B. Image change detection algorithms a systematic survey. IEEE T Image Process. 2005; 14(3):294–307.
27. Wenping M, Jiao L, Gong M, Li C. Image change detection based on an improved rough fuzzy c-means clustering algorithm. Int J Mach Learn & Cyber. 2013; 5:369–77.
28. Won CS. Improved block based image segmentation. International Conference on Image Processing; 1999 Oct 24-28; Kobe. New York: IEEE; 1999. p. 329–32.
29. Nezamabadi-pour H, Saryazdi S. Object-based image indexing and retrieval in DCT domain using clustering techniques. World Academy of Science, Engineering and Technology. 2005; 1(3):768–71.
30. Jiang J, Armstrong A, Feng GC. Direct content access and extraction from JPEG compressed images. Pattern Recogn. 2002; 35:2511–9.
31. Singh PK. Unsupervised segmentation of HRCT lung images using FDK clustering. International Workshop on Biomedical Circuits and systems; 2004 Dec 1-3; New York: IEEE; 2004. p. 3–8.
32. Ngo C-W, Pong T-C, Chin RT. Exploiting image indexing techniques in DCT domain. Pattern Recogn. 2001; 34:1841–51.
33. Lu Z-M, Li S-Z, Burkhardt H. A content based image retrieval scheme in JPEG compressed domain. International Journal of Innovative Computing, Information and Control. 2006; 2 (4):831–9.
34. Feng G, Jiang J. JPEG compressed image retrieval via statistical features. Pattern Recogn. 2003; 36:977–85.
35. Climer S, Sanjiv KB. Image database indexing using JPEG coefficients. Pattern Recogn. 2002; 35:2479–88.
36. Zhao M, Kneepkens REJ, Hofman PM, de Haan G. Content adaptive image de-blocking. International Symposium of Consumer Electronics; 2004 Sep 1-3; New York: IEEE; 2004. p. 299–304.