ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Image Segmentation using Enhanced K-means clustering with divide and Conquer Approach

Babeesh Kumar*[1], Sushila Vikas Maheshkar[2], Ankur Singh Bist[3]
*[1,2] Department of Computer Science and Engineering, ISM Dhanbad, Jharkhand, India
[3] Department of Computer Science and Engineering, QST, Roorkee Uttrakhand, India
babeesh109@gmail.com

Abstract

This paper present image segmentation using Enhanced k-means clustering with divide and conquer approach. First we enhanced the k-means clustering and then segment the image using enhanced approach. K-means is one of the most popular clustering algorithms. The final clustering result of the k-means clustering algorithm greatly depends upon the correctness of the initial centroids, which are selected randomly. Many improvements were already proposed to improve the performance of k-means, but most of these require initial k centroids .In this paper we have proposed a new method enhanced k-means clustering algorithm with divided and conquer algorithm to find the initial centroids with reduced time complexity. In this paper we fixed the initial centroids which gives best result and in the final step we segment the image using enhanced clustering algorithm.

Keywords: K-means Clustering,Image Segmentation,Image processing.

Introduction

Clustering is the process of organizing data objects into a set of disjoint classes called clusters. Clustering is an example of unsupervised classification. Classification refers to a procedure that assigns data objects to a set of classes. Unsupervised means that clustering does not depends on predefined classes and training examples while classifying the data objects. Cluster analysis seeks to partition a given data set into groups based on specified features so that the data points within a group are more similar to each other than the points in different groups Therefore, a cluster is a collection of objects that are similar among themselves and dissimilar to the objects belonging to other clusters. Clustering is a crucial area of research, which finds applications in many fields including bioinformatics, pattern recognition, image processing, marketing, data mining, economics, etc. Cluster analysis is a one of the primary data analysis tool in the data mining. Clustering algorithms are mainly divided into two categories: Hierarchical algorithms and Partition algorithms. A hierarchical clustering algorithm divides the given data set into smaller subsets in hierarchical fashion. A partition clustering algorithm partition the data set into desired number of sets in a single step [9]. Various methods have been proposed to solve clustering problem. One of the most popular clustering methods is k-means clustering algorithm developed by Mac Queen in 1967. The k-means clustering algorithm is a partitioning clustering method that separates data into k groups [1], [2], [4], [5], [7], [9]. The k-means clustering algorithm is more prominent since its intelligence to cluster massive data rapidly and efficiently. However, kmeans algorithm is highly precarious in initial cluster centers Because of the initial cluster centers produced arbitrarily; k-means algorithm does not promise to produce the peculiar clustering results. Efficiency of the original k-means algorithm heavily relies on the initial centroids [2], [5]. Initial centroids also have an influence on the number of iterations required while running the original k-means algorithm. The computational complexity of the original k-means algorithm is very high, specifically for massive data sets [2]. Various methods have been proposed in the literature to enhance the accuracy and efficiency of the k-means clustering algorithm. In this paper a new method is proposed for finding the distance between data points to suitable clusters with reduced time complexity.

To find the distance between two data point we use the closet point –divide and conquer approach and using the closet point –divide and conquer approach we fixed the initial centroid into three parts.

Clustering Techniques

Clustering is a process of dividing data into groups of similar objects [9]. Each group consists of objects that are similar between themselves and dissimilar to objects of other groups. The requirements that should be satisfied by clustering algorithms are scalability, dealing with different types of quality, ability to deal with noise, interpretability and usability. Clustering technique attempts to access the relationship among patterns of the data set by organizing the patterns into groups or clusters such that patterns within a cluster are more similar to each other patterns belonging to different patterns. Clustering refers to the classification of objects into groups according to certain properties of these objects. Clustering techniques can be categorizedinto supervised clustering and unsupervised Clustering. Supervised clustering demands human interaction to decide the clustering criteria and the unsupervised clustering decides the clustering criteria by itself. Supervised clustering includes hierarchical approaches such as relevance feedback techniques [2] and unsupervised clustering includes density based clustering methods. These clustering techniques are done to perform image segmentation. A variety of clustering techniques have been introduced to make the segmentation more effective.

Hierarchical Clustering

One of the well- known technologies in information retrieval is hierarchical clustering [3], [4]. This technique is based on the use of proximity matrix indicating the similarity between every pair of data points to be clustered. The final result is a tree of clusters representing the nested group of patterns and similarity levels at which groupings transform. There clusters are always created as the internal nodes of the tree. The clustering methods differ in regard to the rules by which two small clusters are merged or a large cluster is split. The two main classification of algorithms used in the hierarchical clustering framework are agglomerative and divisive. Agglomerative algorithms request to merge clusters to be larger by starting with N single point clusters. This algorithm can be divided into three classes.

(i) Single link algorithm
(ii) Complete link algorithm
(iii) Minimum variance algorithm

In single link algorithm, it merges two clusters based on the minimum distance between the data samples from two clusters. In complete link algorithm, it incorporates the maximum distance between the data samples in clusters. In minimum variance algorithm, it combines two clusters to minimize the cost function to form a new cluster. Divisive clustering begins with the same cluster in entire dataset. It follows a reverse splitting of the dataset until the single point clusters are attained on leaf nodes. It follows a reverse clustering approach against agglomerative algorithm.

Partitional Clustering

It uses an iterative optimization procedure that aims at minimizing an objective function which measures the goodness of clustering. The cluster centroids are usually computed based on the optimality criterion such that the objective function is minimized.

Partitioned algorithms are divided into two types.
(i) Partitioning relocation algorithm
(ii) Density based partitioning

Fuzzy Clustering

Clustering methods can be classified as either hard or fuzzy depending on whether a pattern data belongs exclusively to a single cluster or several clusters with different values [8]. In hard clustering, a membership value of zero or one is assigned to each pattern data. In fuzzy clustering, a value between zero and one is assigned to each pattern by a membership function.

Fuzzy clustering methods can be considered to be superior to those of their hard counterparts since they can represent the relationship between the input pattern data and clusters. It is used to minimize the heuristic global cost function. Fuzzy k-means clustering algorithm iteratively updates the cluster centroid and estimates the class membership function by using the gradient descent approach.

K-Means Clustering

It is the simplest method in supervised classification. It does not require training data. It is an iterative procedure. K-means clustering algorithm clusters the data by iteratively computing a mean intensity for each class and segmenting the image by classifying each pixel in the class with the closest mean. Clustering based on the optimization of an overall measure is a fundamental approach explored since the early days of pattern recognition. The most popular method for pattern recognition is K-means clustering. In K-means clustering a centroid vector is computed for every cluster. The centroid must be Chosen such that it should minimize the total distance within the clusters [3].

Algorithm I. k-means clustering
Steps:

1. Choose k initial clusters Z_1(s), Z_2(s)…………Z_k(s).

2. At k^{th} iterative step, distribute the samples X among k clusters using the relation.
$X \in C_i\ (k)\ if\ ||\ X - Z_m\ (k)\ ||\ \leq\ ||\ X - Z_n\ (k)\ ||$

For $m = 1, 2……k, m \neq n$, where $C_i\ (k)$ denotes the set of samples whose cluster center is $Z_n(k)\ N_n$

3. Compute new cluster centres $Z_n(k+1), n = 1, 2……k$. Such that the sum of squared distance from all points in $C_i(k)$ to new Cluster is minimized. The measure which minimizes this is only the sample means of C_i (k), So the new cluster is given by

$$Z_n(k+1) \ = \frac{1}{N_n} \Sigma_X \quad , n = 1, 2……k$$ Where

N_n is the number of samples in C_n(k).

4. If $Z_n(k+1), n = 1, 2…K$, the algorithm has converged and the procedure is terminated. If not go to step 2.

Weakness of k-means clustering

1. The number of clusters must be determined before the algorithm is executed
2. The algorithm is sensitive to initial conditions. It produces different results for different initial conditions.
3. The K-means algorithm may be trapped in the local optimum. As a result, the trapped clusters would rep-resent wrong group of data.
4. Data which are far away from the centers may pull the centers away from the optimum location, leading to poor representation of data.

We can reduce the weakness of k-means clustering using closet pair –divide and conquer approach. The closet pairs –divide and conquer approach find the best initial centroid.

Proposed Approach

1. *Closet pairs-divide and conquer approach*

Each recursive invocation of the algorithm takes as input a subset $P \subseteq Q$ and arrays X and Y, each of which contains all the points of the input subset P. The points in array X are sorted so that their x-coordinates are monotonically increasing. Similarly, array Y is sorted by monotonically increasing y-coordinate. Note that in order to attain the $O(n\ \lg n)$ time bound, we cannot afford to sort in each recursive call; if we did, the recurrence for the running time would be $T(n) = 2T(n/2) + O(n\ \lg n)$, whose solution is
$T\ (n) = O\ (n\ \lg 2n)$.
A given recursive invocation with inputs P, X, and Y first checks whether $|P| \leq 3$. If so, $\overset{|p|}{C}{}_2$ the invocation simply performs the brute-force method described above: try all $\overset{|p|}{C}{}_2$ pairs of points and return the closest pair. If $|P| > 3$, the recursive invocation carries out the divide-and conquer paradigm as follows.

2. Divide:

It finds a vertical line l that bisects the point set Pinto two sets $P_L$and $P_R$such
That $|P_L| = \lceil |P|/2 \rceil$, $|P_R| = \lfloor |P|/2 \rfloor$, all points in $P_L$are on or to the left of line l, and all points in PR are on or to the right of l. The array X is divided into arrays $X_L$and X_R, which contain the points of $P_L$and $P_R$respectively, sorted by monotonically increasing x-coordinate. Similarly, the array Y is divided into arrays $Y_L$and$Y_R$which contain the points of $P_L$and $P_R$respectively, sorted by monotonically increasing y-coordinate.

3. Conquer:

Having divided P into $P_L$andP_R, it makes two recursive calls, one to find the closest pair of points in $P_L$and the other to find the closest pair of points inP_R. Theinputs to the first call are the subset$P_L$and arrays $X_L$and Y_L; the second call receives the inputsP_R , X_R, and Y_R. Let the closest-pair distances returned for $P_L$and $P_R$be $\delta_L$andδ_R, respectively, and let $\delta = \min\ (\delta_L, \delta_R)$.

4. Combine:

The closest pair is either the pair with distance δ found by one of the recursive calls, or it is a pair of points with one point in $P_L$and the other in P_R. The algorithm determines if there is such a pair whose distance is less than δ. Observe that if there is a pair of points with distance less than δ, both points of the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

pair must be within δ units of line l. Thus, as Figure (a) shows, they both must reside in the 2δ-wide vertical strip centered at line l. To find such a pair, if one exists, the algorithm does the following

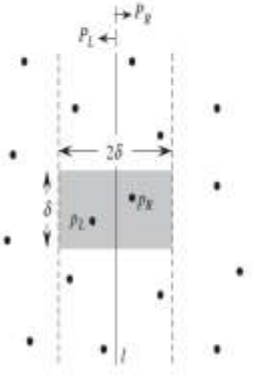

Figure (a)

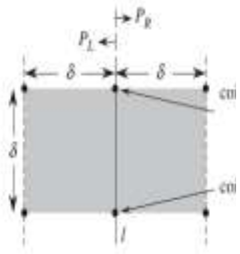

Figure (b)

Key concepts in the proof that the closest-pair algorithm needs to check only 7 points following each point in the array Y′. figure*(a)* If $p_L \subseteq P_L$and $p_R \subseteq P_R$ are less than δ units apart, they must reside within a δ × 2δ rectangle centered at line *l*. figure*(b)* How 4 points that are pair wise at least δ units apart can all reside within a δ × δ square. On the left are 4 points inP_L, and on the right are 4 points in P_R. There can be 8 points in the δ × 2δ rectangle if the points shown on line l are actually pairs of coincident points with one point in $P_L$and one in P_R.

1. It creates an array Y', which is the array Y with all points not in the 2δ-wide vertical Strip removed. The array Y' is sorted by y-coordinate, just as Y is.

2. for each point p in the array Y', the algorithm tries to find points in Y' that are within δ units of p. As we shall see shortly, only the 7 points in Y' that follow p need be considered. The algorithm computes the distance from p to each of these 7 points and keeps track of the closest-pair distance δ' found over all pairs of points in Y'.

3. If $\delta' < \delta$, then the vertical strip does indeed contain a closer pair than was found by the recursive calls. This pair and its distance δ' are returned. Otherwise, the closest pair and its distance δ found by the recursive calls are returned. The above description omits some implementation details that are necessary to achieve the O (n lg$n$) running time. After proving the correctness of the algorithm, we shall show how to implement the algorithm to achieve the desired time bound.

5. *Implementation and running time:*

As we have noted, our goal is to have the recurrence for the running time be T (n) = 2T (n/2) + o (n), where T (n) is the running time for a set of n points. The main difficulty comes from ensuring that the arrays $X_L, X_R, Y_L,$ and Y_R, which arepassed to recursive calls, are sorted by the proper coordinate and also that thearray Y' is sorted by y-coordinate. (Note that if the array X that is received by is cursive call is already sorted, then we can easily divide set Pinto $P_L$and P_R in linear time.)The key observation is that in each call, we wish to form a sorted subset of a sorted array. For example, a particular invocation receives the subset P and the array Y, sorted by y-coordinate. Having partitioned P into $P_L$and P_R, it needs to form the arrays Y_L and Y_R, which are sorted by y-coordinate, in linear time. We can view the method as the opposite of the MERGE procedure from merge sort .we are splitting a sorted array into two sorted arrays. The following pseudo code gives the idea.

Algorithm II: Divide and conquer Approach

```
Let $Y_L[1....Y.\ length]$ and $Y_R[1....Y.\ length]$ be new arrays.
$Y_L.\ length = Y_R\ .\ length = 0$
For $i = 1$ to $Y.length$
    If $Y[i]$ is in $P_L$
        $Y_L.length = Y_R.length + 1$
        $Y_L\ [\ Y_L.\ length\ ] = Y[i]$
    else $Y_R.length = Y_R.length + 1$
        $Y_R[\ Y_R.length] = Y[i]$
```

We simply examine the points in array Y in order. If a point Y[i] is in P_L, we append it to the end of array Y_L; otherwise, we append it to the end of array Y_R. Similar pseudo code works for forming arrays X_L, X_R, and Y’. The only remaining question is how to get the points sorted in the first place. We ***presort*** them; that is, we sort them once and for all *before* the first recursive call. We pass these sorted arrays into the first recursive call, and from there we whittle them down through the recursive calls as necessary. Presorting adds an additional O (n lgn) term to the running time, but now each step of the recursion takes linear time exclusive of the recursive calls. Thus, if we let T (n) be the running time of each recursive step and T' (n) be the running time of the entire algorithm, we get T' (n) = T (n) + O (n lgn) and

$T(n) = 2T(n/2) + O(n)$ if $n > 3$
$O(1)$ if $n \leq 3$

Thus, T (n) = O (n lgn) and T' (n) = O (n lgn).

1. Enhanced k-means with divide and conquer Approach

In this section, we proposed an enhanced method for enhancing the performance of k-means clustering algorithm. In the paper [1] authors proposed an enhanced method to improve the efficiency of the k-means clustering algorithm. But in this method the initial centroids are selected randomly. So this method is very sensitive to the initial starting points and it does not promise to produce the unique clustering results. In the paper [2] authors proposed an enhanced algorithm to improve the accuracy and efficiency of the k-means clustering algorithm. In this algorithm two methods are used, one method for finding the better initial centroids. And another method for an efficient way for assigning data points to appropriate clusters. In the paper [2] the method used for finding the initial centroids computationally expensive. In this paper we proposed a new approach for finding the better initial centroids with reduced time complexity. The initial centroids are fixed for this we used divide and conquer approach, ClusterId of a data point denotes the cluster to which it belongs, for assigning the data points we follows the paper [1], [2]. The pseudo code for the proposed algorithm is outlined as Algorithm 2.

Algorithm III: The Enhanced Method
Require: *D = {d1, d2, d3……….....dn}* // Set of n data points.
K // Number of desired clusters.
Ensure: A set of *k* clusters.
Steps:

```
1. In the first step apply divide and conquer Approach which is define above.
2. Find the distance of each closer pair for $P_L$= {$\delta_{L1},\delta_{L1},\delta_{L2},\delta_{L3}, \dots \dots \dots \dots . \delta_{Ln}$}.
3. Find the distance of each closer pair for $P_R$= {$\delta_{R1},\delta_{R1},\delta_{R2},\delta_{R3}, \dots \dots \dots \dots . \delta_{Rn}$}
4. Then find the distance of each closer pair for 2$\delta$ strip
       $P_M = \{ \delta_1,\delta_2,\delta_3 \dots\dots\dots\dots\dots\dots, \delta_n \}$.
5. Sort the data points of $P_L$ , $P_R$ , $P_M$ accordance with the distances.
6. In each set $P_L$ , $P_R$ , $P_M$, take the middle point as the initial centroid.
7. Compute the distance between each data point to all the initial centroids ( $P_L$ , $P_R$ , $P_M$).
8. For each data point find the closest centroid ( $P_L$ , $P_R$ , $P_M$) and assign to cluster j.
9. Set Clustered[i] = j. // j: Id of the closest cluster.
```

Image segmentation using enhanced algorithm

Image segmentation refers to the process of partitioning an image into group of pixels which are homogeneous with respect to some criterion. Different groups must not intersect with each other. And adjacent group must be heterogeneous. Segmentation algorithms are area oriented instead of pixel-oriented. The result of segmentation is the splitting up of the image into connected areas. Thus segmentation is concerned with dividing an image into meaningful regions. Image segmentation was, is and will be a major research topic for many image processing researchers. The reasons are obvious and applications endless: most computer vision and image analysis problems require a segmentation stage in order to detect objects or divide the image into regions which can be considered homogeneous according to a given criterion, such as color, motion, texture, etc.

There are many application of image segmentation such that in medical imaging locate tumors and other pathologies, measure tissue volumes, diagnosis study of anatomical structure. In object detection like pedestrian detection, face detection, brake light detection, and locate objects in satellite images (roads, forest, forests, crops, etc.). In recognition tasks like face recognition, fingerprint recognition, lris recognition. In traffic control systems like video surveillance. The purpose of this paper is to detect the infected area in medical database colour image and detect the disease easily.

Algorithm IV: color Image Segmentation Using Enhanced Algorithm

The basic aim is to segment colors in an automated fashion using the L*a*b* color space and enhanced

K-means clustering. The entire process can be summarized in following steps.

Step 1. Read the image from source which is in .JPEG format.
Step 2: Convert Image from RGB Color Space to L*a*b* Color Space. The L*a*b* color space (also known as CIELAB or CIE L*a*b*) enables us to quantify these visual differences. The L*a*b* space consists of a luminosity layer 'L*', chromaticity-layer 'a*' indicating where color falls along the red-green axis, and chromaticity-layer 'b*' indicating where the color falls along the blue-yellow axis.

Step 3: Classify the Colors in 'a*b*' Space Using enhanced K-Means Clustering.
Step 4: Label Every Pixel in the Image Using the Results from enhanced K-MEANS For every object in our input, enhanced K-means returns an index corresponding to a cluster. Label every pixel in the image with its cluster index.
Step 5: Create Images that Segment the Image by Color.
Using pixel labels, we have to separate objects in image by color, which will result in three images.
Step 6: Segment the image into a Separate Image.
Then programmatically determine the index of the cluster containing the blue objects

Experimental result

We have apply enhanced k-means clustering on various images like brain tumer ,brain cancer and more other which gives the best result as compaire to k-means clustering.

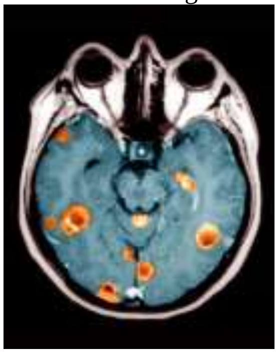

Fig1(a). Original brain cancer Image

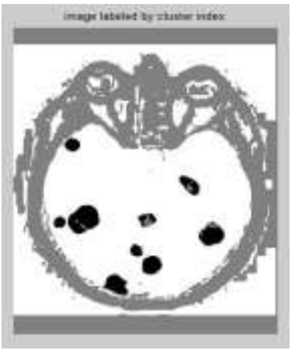

Fig1(b). Labeled Image

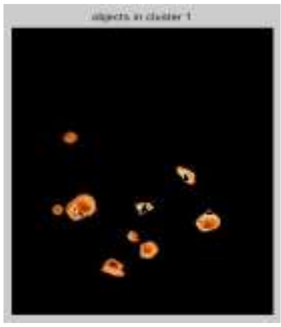

Fig1 (c) Object in cluster 1

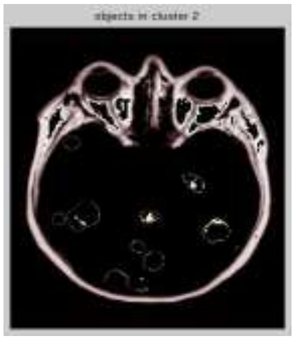

fig1 (d) Object in cluster 2

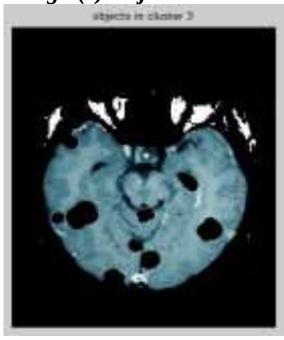

Fig1(e). Object in cluster 3 image fig1. (f).Segmented fig1. (f).Segmented

The fig1 (a) show the original image of Brain cancer. After apply the enhanced k-means clustering we find the cluster index and then we labeled the pixel values based on the clustered index and we find the labeled image fig1 (b) .the fig1(c), fig1 (d), fig1(e) are object in cluster 1, 2, 3.
The fig1 (f) show the segmented image of object in cluster 3.
Same process we apply on Brain tumor and insect image which result are show below.

Fig2 (a) Original image of Brain tumor

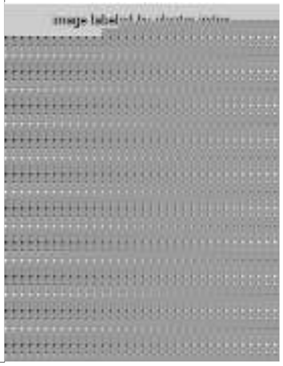

fig2 (b) Image labeled by cluster index

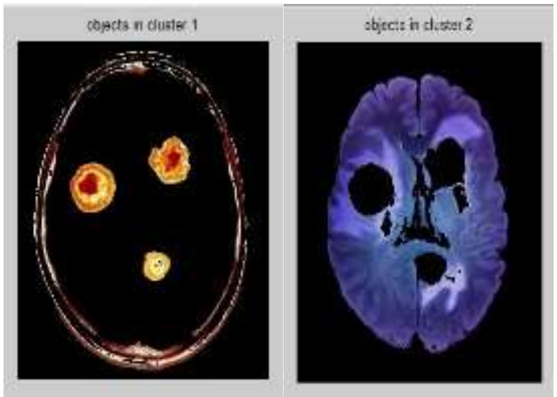

Fig2(c) Object in cluster 1 fig2 (d) object in cluster 2

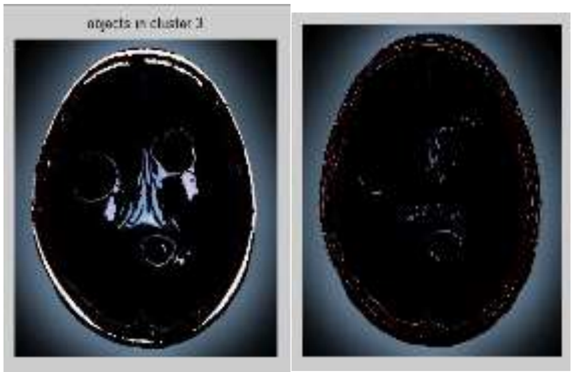

fig2 (e) Object in cluster 3 fig2 (f) Segmented image of Cluster 3

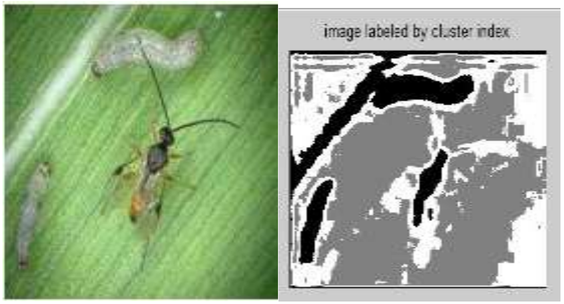

Fig3 (a) Original inset image fig3 (b) labeled image

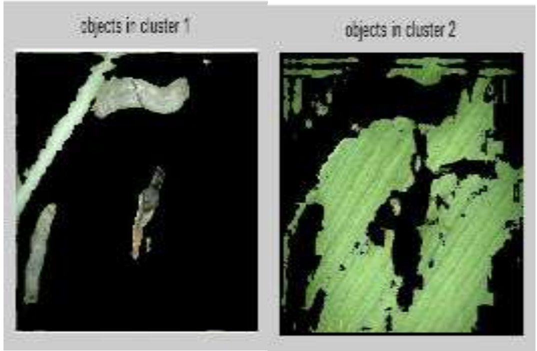

Fig3(c) Cluster image 1 fig3 (d) Cluster image 2

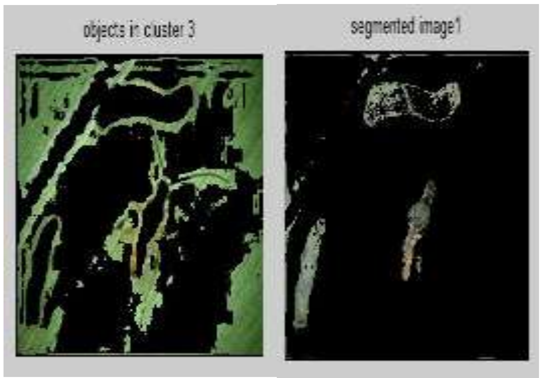

Fig3 (e) Cluster image 3 fig3 (f) Segment image of Cluster 3

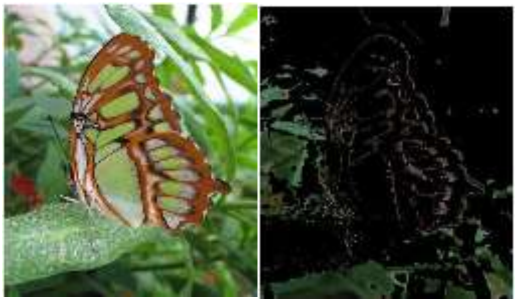

Fig4 (a) original insect image fig4 (b) segmented image

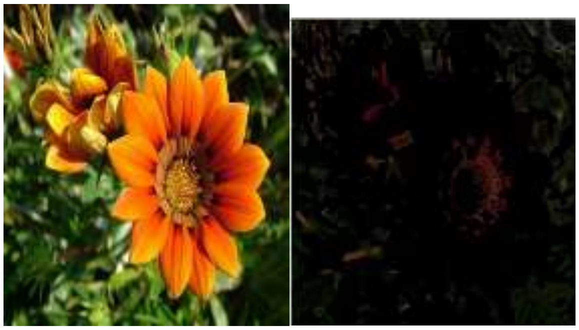

Fig 5(a).original image fig 5(b).segmented image

TABLE I
PERFORMANCE COMPARISON OF THE ALGORITHMS WITH TIME

Data image	**Size of image**	**Algorithm**	**Accuracy (%)**	**Time Taken (sec)**
Brain				5.26756
cancer	435X580	K-Means		
		Proposed Algorithm	22.7813	4.06754
Brain				3.290496
tumour	250X292	K-Means	8.6010	
		Proposed Algorithm		3.007480
Insect				1.879805
	244X179	K-Means	42.6028	
		Proposed Algorithm		1.078955
Butterfly	321 X			2.76588
	320	K-Means	19.2181	
		Proposed Algorithm		2.23433
Flower	1024 X	K-Means		6.67577
	768		24.42175	
		Proposed Algorithm		5.04543

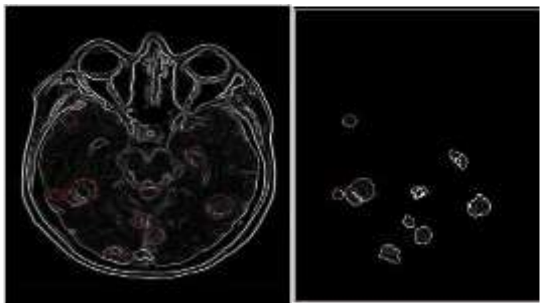

Fig6. (a) Edge detector image
Fig7 (b) Edge detector image Of original image of
Brain of object in cluster 1 using Cancer
K-means

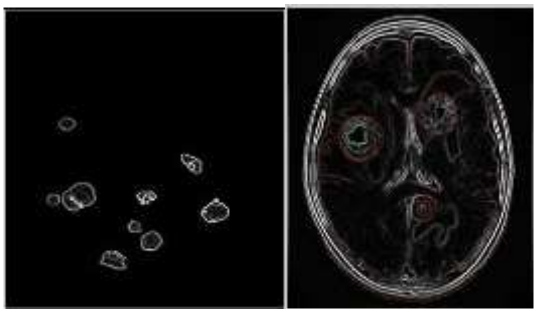

Fig7(c) Edge detector image Fig8 (a) Edge detector
image Of object in cluster 1 enhanced of original
image of Brain
K-means

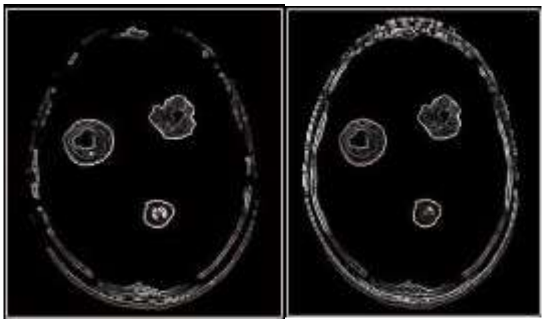

Fig8 (b) Edge detector image
fig8(c) Edge detector image Of object in cluster 1 using
of object in cluster 1 using
K-means enhanced k-means

We apply the edge detector on the original image of brain cancer fig6 (a) and then apply the edge detector on the segmented image of cluster 1 using k-means then calculate the psnr.

We can calculate the performance using psnr of k-means technique and enhanced k-means technique. Table II define the performance comparison of algorithms with psnr values of k-means and enhanced k-means.

TABLE II
PERFORMANCE COMPARISON OF THE ALGORITHMS WITH PSNR

Data image	**Size of image**	**Algorithm**	**Accuracy (%)**	**PSNR**
Brain				22.4324
cancer	435X580	K-Means		
		Proposed Algorithm	12.129	25.1534
Brain				20.0234
tumour	250X292	K-Means	7.579	
		Proposed Algorithm		21.5411
Insect				28.0654
	244X179	K-Means	8.037	

		Proposed Algorithm		30.3211
Butterfly	321 X 320	K-Means	18.083	24.76688
		Proposed Algorithm		30.23433
Flower	1024 X 768	K-Means	18.991	35.67977
		Proposed Algorithm		44.04443

Conclusion

Using colour based image segmentation; it is possible to reduce the computational cost avoiding feature calculation for every pixel in the image. The main purpose of this paper is to reduce the computational time with divide and conquer approach and find the suitable initial centroid. As we seen above, by implementing the algorithm at first figure we point out the infected cancer region of the data base colour image, and when we use the same algorithm in MRI data base brain-tumor image, if we seen above fig8 (b) edge detector image on image segmented using k-means, some edges are remove but if apply enhanced k-means fig8(c) then we can recover edge. We compare the result using time and psnr of segmented image using k-means and enhanced k-means. Though this experiment uses the MATLAB software to solve its problem so effectively it has work done in a very few of cost.

References

1. *Densheng Zhang and Guojun Lu ,Review of Shape Representation and Description Techniques ,Pattern Recognition,pp.1-19,2004*
2. *V Grau et al .Improved Watershed Tranform for Medical Image Segmentation using prior Information ,IEEE Transactions on Medical Imaging,Vol.23,No.4,April 2004*
3. *J Canny ,A Computational Approach to Edge Detection,IEEE Trans.Pattern Analysis and Machine Intelligence, Vol.29,pp.100-132,1985*
4. *R M Haralick and L G Shapiro, Image Segmentation Techniques, Computer Vision, Graphics, and Image Processing, Vol.29, pp.100-132, 1985.*
5. *S Y Chen,W C Lin and C T Chen, Split-and-Merge Segmentation based on Localised Feature Analysis and Statistical Tests ,CVGIP: Graphical Models and Image Processing Vol.5,pp.457-475,1991*
6. *Mehmet Nadir Kurnaz,et al; Segmentation of remote-sensing images by incremental .neural network, Pattern Recognition Letters 26 (2005) 1096–1104, pp 1096-1103*
7. *N Bartneck et al; Colour segmentation with polynomial classification, 0-8186-2915-0/92, 1992, pp 635-638.*
8. *Nae-Joung Kwak et al; color image segmentation using edge and adaptive threshold value based on the image characteristics,IEEE proceeding0-7803-8639-6,2004, pp 555-2010*
9. *Lingkui Meng, et al, Study on Image Segment Based Land Use Classification and Mapping, 2009 IEEE, pp 656-2011*
10. *Pal N.R.et al; A review on image segmentation techniques, Pattern Recognition 26(9), 1993, pp1277-1294.*
11. *Ricardo Dutra da Silva et al; Satellite image segmentation using wavelet transforms based on color and texture features, ISVC 2008, part II, LNCS 5359, 2008, pp 113-122*
12. *Robert A Schowengerdt, Remote sensing-models and Methods for Image Processing, IIIrd edition, Elsevier Inc.*