**CHAPTER:1**

**Abstract**

An algorithm is proposed for segmenting an image into multiple levels using Fuzzy-c-means clustering technique. Image segmentation is the technique of decomposing an image into meaningful parts or objects .Its results in a segmented image, where each object is labeled in a way that facilitates the description of the original image so that it can be interpreted by the system that handles the image .In this Technical report first determined the center of clustering andFuzzy partition matrix. Considering the spatial information of pixels and based on spatially weighted FCM, we propose a novel algorithm which can get faster and more accuracy colorimage segmentation results. The results of experiments show the effectiveness of the proposed algorithm. In this paper, an fuzzy *c*-means (FCM) clustering algorithm for image segmentation is presented. The originality of this algorithm is based on the fact that the conventional FCM-based algorithm considers no spatial context information, which makes it sensitive to noise. Thealgorithm is formulated by incorporating the spatial neighborhood information into the original FCM algorithm by *a* prioriprobability and initialized by a histogram based FCM algorithm. The probability in the algorithm that indicates the spatial influence of the neighboring pixels on the centre pixel plays a key role in this algorithm and can be automatically decided in the implementation of the algorithm by the fuzzy membership. To quantitatively evaluate and prove the performance of the proposed method, series of experiments and comparisons with many derivates of FCM algorithms are given in the paper. Experimental results show that the proposed method is effective and robust to noise.

**CHAPTER:2**

**INTRODUCTION**

With the development of computer technology, image segmentation has become an important tool in many computer vision applications and image processing applications. The goal of image segmentation is to find regions that represent objects or meaningful parts of objects. Based on the applications of an image, division of the image into regions, corresponding to objects of interest, is necessary before any processing can be done at a level higher than that of the pixel. Identifying real objects,pseudo-objects, and shadows or actually finding out anything of interest within the image require some form of segmentation. In this paper, we present reliable algorithms for fuzzy k-means and C-means that could improve MRI segmentation. Since the k-means or FCM method aims to minimize the sum of squared distances from all points to their cluster centers, this should result in compact clusters. Therefore the distance of the points from their cluster centre is used to determine whether the clusters are compact. For this purpose, we use the intra-cluster distance measure, which is simply the median distance between a point and its cluster centre. The intra-cluster is used to give us the ideal number of clusters automatically; i.e a centre of the first cluster is used to estimate the second cluster, while an intra-cluster of the second cluster is obtained. Similar, the third cluster is estimated based on the second cluster information (centre and intra cluster), so on, and only stop when the intra-cluster is smaller than a prescribe value. The proposed algorithms are evaluated and compared with established fuzzy kmeans and C-means methods by applying them on simulated volumetric MRI and real MRI data to prove their efficiency. These evaluations, which are not easy to specify in the absence of any prior knowledge about resulted clusters, for real MRI dataset are judged visually by specialists since a real MRI dataset cannot give us a quantitative measure about how much they are successful*.*Fuzzy c-means (FCM) clustering algorithm is a popular model widely used in image segmentation due to its good performance. Several algorithms have been proposed to improve FCM in incorporating the spatial information of the image, defining new distance or using histogram for segmentation. In the proposed method, input color

images are converted into HIS color images, which are more closed to human vision. Magnetic resonance image segmentation has been proposed for a number of clinical investigations of varying complexity. Automatic segmentation of MR scans is very useful for research and clinical study of much neurological pathology. The accurate segmentation of MR images into different tissue classes, especially Gray Matter (GM), White Matter (WM) and CerebroSpinal Fluid (CSF), is an important for the diagnosis and prognosis of certain illnesses. The automatic segmentation of brain MR images, however, remains a persistent problem. Magnetic Resonance Imaging (MRI) is an advanced medical imaging technique providing rich information about the human soft tissue anatomy. Segmentation of MR images into different tissue classes, especially grey matter (GM), white matter (WM) and cerebrospinal fluid (CSF) is an important task. Accurate and robust brain tissue segmentation from magnetic resonance (MR) images is key issue in many applications of medical image analysis for quantitative studies and particularly in the study of several brain disorders. A wide variety of approaches have been proposed for brain MR image segmentation. The major MR image segmentation problem when MR image is the corruption with an inhomogeneity bias field In region-based segmentation, the shape of an object can be described in terms of its boundary or the region it occupies. In its simplest form, region growing methods usually start by locating some seeds representing distinct regions in the image. The seeds are then grown until they eventually cover the entire image. The region growing process is therefore governed by a rule that describe the growth mechanism and a rule that check the homogeneity of the regions at each growth step. Region growing technique has been applied to MRI segmentation. This paper addresses these problems for overcoming the shortcomings of existing fuzzy methods. We present alternative k-means and FCM algorithms that could improve MRI segmentation. The algorithms incorporate spatial information into the membership function and the validity procedure for clustering. We use the intra-cluster distance measure, which is simply the median distance between a point and its cluster centre. The number of the cluster increases automatically according the value of intra-cluster, for

example when a cluster is obtained, it uses this cluster to evaluate intra-cluster of the next cluster as input to the FCM or k-means and so on, stop only when intracluster is smaller than a prescribe value. The most important aspect of the proposed algorithms is actually to work automatically. Alterative is to improve automatic image segmentation. The performance of the proposed method is illustrated using simulated volumetric MRI and real MRI. Due to the reference of real MRI dataset being unknown to measure how much our algorithms are successful, the opinion of specialists are considered. The rest of this paper is organized as follows.

**CHAPTER:3**

**Image Segmentation**

**3.1)Definition**: In [computer vision](http://en.wikipedia.org/wiki/Computer_vision), **segmentation** refers to the process of partitioning a [digital image](http://en.wikipedia.org/wiki/Digital_image) into multiple [segments](http://en.wikipedia.org/wiki/Image_segment) ([sets](http://en.wikipedia.org/wiki/Set_%28mathematics%29) of [pixels](http://en.wikipedia.org/wiki/Pixel), also known as superpixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.The result of image segmentation is a set of segments that collectively cover the entire image, or a set of [contours](http://en.wikipedia.org/wiki/Contour_line) extracted from the image . Each of the pixels in a region are similar with respect to some characteristic or computed property, such as [color](http://en.wikipedia.org/wiki/Color), [intensity](http://en.wikipedia.org/wiki/Luminous_intensity), or [texture](http://en.wikipedia.org/wiki/Image_texture). [Adjacent](http://en.wikipedia.org/wiki/Adjacent) regions are significantly different with respect to the same characteristic.

**3.2)Purpose of Segmentation**: The goal of image segmentation is to cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces,objects, or natural parts of objects.

A segmentation could be used for object recognition, occlusion boundary estimation within motion or stereo systems, image compression,image editing, or image database look-up.

**3.3)Level of segmentation:** It depends upon the application where it is being used.It is one of the most important part of machine vision application and one of the most difficult part of image analysis.The success of the process lies in the level of segmentation.

**3.4)Approach:** Though there are different techniques applied in image segmentation process,our project is mainly based on the thresholding operation which comes under the similarity based approach.

**3.5)Thresolding techniques:** Suppose that the gray-level histogram corresponds to an image, f(x,y), composed of dark objects in a light background, in such a way that object and background pixels have gray levels grouped into two dominant modes. One obvious way to extract the objects from the background is to select a threshold ‘T’ that separates these modes. Then any point (x,y) for which f(x,y) > T is called an object point, otherwise, the point is called a background point.

**3.6)Goals of our Project:**

* Firstly apply the techniques to static imagery and then proceed to real time error detections in practical life using fuzzy logic.
* Maybe then applications in industrial fields.

**CHAPTER:4**

**Different Types of Segmentation Technique**

**4.1)Overview**

Image segmentation methods identify objects that either have some measure of homogeneity or have some measure of contrast with neighboring objects. Most image segmentation algorithms are modifications, extensions or combinations of these two basic concepts. Homogeneity and contrast measures the quantities such as gray level,color and

texture. After performing preliminary segmentation, higher-level object properties, such as perimeter and shape, may be incorporated into the segmentation process.

**4.2)Segmentation Techniques and Previous Work**

There are many challenging issues related to the development of a unified approach to image segmentation, which can (probably) be applied to all kinds of images. Even the selection of an appropriate technique for a specific type of is a difficult problem. Up to now, there is no universally accepted method of evaluating a segmented output.There are a wide variety of segmentation techniques in the literature; some are considered general purpose while others are applicable to a specific class of images.Using the basic properties of gray-level values, classical segmentation techniques have been developed. The classical segmentation techniques are based on histogram thresholding, edge detection, iterative pixel classifications, semantic and syntactic approaches .In addition to this, there are certain methods which do not fall clearly in any one of the above classes. In addition to the classical techniques, there are also methods based on fuzzy mathematics. The fuzzy mathematical approach has methods based on edge detection thresholding and relaxation. Some of these methods, particularly the histogram based methods, are not at all suitable for noisy images.Several methods have also been

developed using neural network methods. These algorithms work well even in highly noise environment and they are capable of producing output in real time.

**4.3)Grey level thresholding used in this thesis:**

Thresholding is one of the old, simple and popular techniques for image segmentation. Grey level thresholding is useful whenever the grey level features sufficiently characterize the object. The appropriate grey values are calibrated so that a given grey level interval represents a unique object characteristic. Thresholding can be done based on global information (e.g. the grey level histogram of the entire image) or it can be done using local information (e.g. the co-occurrence matrix) of the image. Under each of the schemes, if only one threshold is used for the entire image then it is called global thresholding. On the other hand, when the image is partitioned into several subregions and a threshold is determined for each of the subregions, the scheme is referred to as local thresholding. Some authors refer to these local thresholding methods as adaptive thresholding schemes. Thresholding can also be classified as bilevel thresholding and multithresholding. In bilevel thresholding the image is partitioned into two regions: object (black) and background (white). When the image is composed of several objects with different surface characteristics (for a light intensity image, objects with different coefficient of reflection, for a range image there can be objects with different depths and so on) one needs several thresholds for segmentation. This is known as multithresholding. Bilevel thresholding is equivalent to classifying the pixels into two classes: object and background. If the image is composed of regions with different grey level ranges, i.e. the regions are distinct, the histogram of the image usually shows different peaks, and each corresponding to a region and adjacent peaks are likely to be separated by a valley. For example, if the image has a distinct object on a background, the grey level histogram is likely to be bimodal with a deep valley. In this case the bottom of the valley is taken as the threshold for object background separation. Therefore, when the histogram has one (or a set of) deep valley(s), selection of threshold(s) becomes easy because it involves detecting valleys. Normally the situation is not like this and threshold selection is not a

trivial job. There are various methods available for this. For example, Otsu maximized a measure of class separability. He maximized the ratio between the class variance to the local variance to obtain thresholds. Kapur, assumed two probability distributions, one for the object area and the other for the background area. They then, maximized the total entropy of the partitioned image in order to arrive at the threshold level. Though these methods use only the histogram, they produce good results due to the incorporation of the image formation model.The philosophy behind grey level thresholding, "pixels with grey level <= T fall intoone region and the remaining pixels belong to another region", may not be true on many occasions, particularly, when the image is noisy or the background is uneven and illumination is poor. In such cases the objects will still be lighter or darker

than the background, but any fixed threshold for the entire image will usually fail to separate the objects from the background. This leads one to the methods of adaptive thresholding. In adaptive thresholding normally the image is partitioned into several non-overlapping blocks of equal area and a threshold for each block is computed independently (sometimes use overlapping regions and blend thresholds). The sub histogram of each block is used to determine local threshold values for the corresponding cell centers. These local thresholds are then interpolated over the entire image to yield a threshold surface.

**4.4)Boundary based techniques**

Boundary extraction techniques segment objects on the basis of their profiles. Thus, contour following, connectivity, edge linking and graph searching, curve fitting, Hough transform and other boundary extraction techniques are applicable to image segmentation. Difficulties with boundary-based methods occur when objects are touching or overlapping or if a break occurs in the boundary due to noise or artifacts in the image.

**4.5)Edge based techniques**

Segmentation can also be obtained through detection of the edges of regions, normally by locating points of abrupt changes in grey level intensity values. There are various types of edge detection operators in use today. Many are implemented with convolution masks, and most are based on discrete approximations to differential operators. Some edge detection operators return orientation information (information about the direction of the edge), whereas others only return information about the existence of an edge at each point.Different edge operators produce an edgeness value at every pixel location. However not all of them are valid candidate for edges. Normally, edges are required to be thresholded. The selection of the threshold is very crucial as for some parts of the image

low intensity variation may correspond to edges of interest while in other parts there is high intensity variation. Adaptive thresholding is often taken as a solution to this. Obviously it cannot eliminate the problem of threshold selection. A good strategy to produce meaningful segments would be to fuse region segmentation results and edge outputs. Incorporation of psycho visual phenomena may be good for light intensity images but not applicable for range images. Semantics and a prioriinformation about the type of image are critical to the solution of the segmentation problem.Since edges are local features, they are determined based on local information.Edge detection techniques are two categories: sequential and parallel. In the sequential technique the decision whether a pixel is an edge pixel or not is dependent on the result of the detector at some previously examined pixels.On the other hand, in the parallel method the decision whether a point is an edge or not is made based on the point under consideration and some of its neighboring points.As a result of this the operator an be applied to every point in the image simultaneously. The performance of a sequential edgedetection method is dependent on the choice of an appropriate starting point and how the results of previous points influence the selection and result of the next point. There are different types of parallel fferential operators such as Roberts gradient, Sobel gradient, Prewitt gradient and

the Laplacian operator. These difference operators respond to changes in grey level or average grey level . The gradient operators not only respond to edges but also to isolated

points. For Prewitt's operators the response to diagonal edges is weak while Sobel's operator gives greater weights to points lying close to the point(x, y) under consideration. However, both Prewitt's and Sobel's operators possess greater noise immunity than other difference operators. The preceding operators are called first difference operators. On the other hand, the Laplacian is a second difference operator. The digital Laplacian being a second difference operator has a zero response to linear ramps. It responds strongly to corners, lines and isolated points. Thus for a noisy picture, unless the picture has low contrast, the noise will produce higher Laplacian values than the edges. A good edge detector should be a differential operator and should have the following three properties: (1) Low probability of wrongly marking non-edge points and low probability of failing to mark real edge points (2) points marked as edges should be as close as possible to the center of true edges (i.e. good localization) (3) one and only one response to a single edge point (single response).

**4.6)Region based techniques**

The main objective in region-based segmentation techniques is to identify various regions in an image that have similar features. One class of region-based techniques involves region-growing . As implied by its name, region growing is a procedure that groups pixels or subregions into larger regions. The simplest of these approaches is pixel aggregation, where the growing process starts with "seed" points and from these grow regions by appending to each seed point those neighboring pixels that have similar properties (e.g. grey level, texture, color). If the absolute difference between the grey level of the neighboring pixel and the grey level of the seed is less than a threshold then that neighboring pixel is added to the seed Although this growing procedure is simple in nature, it suffers some important problems in region growing. Two immediate problems are the selection of initial seeds that properly represent regions of interest, and the

selection of suitable properties for including points in the various regions during the growing process. Another important problem in region growing is the formulation of a

stopping rule. Basically, a region growing process is stopped when no more pixels satisfy the criteria (e.g. intensity, texture) for inclusion in that region.Additional criteria that increase the power of a region-growing algorithm incorporate the concept of size, likeness between a candidate pixel and the pixel grown thus far (e.g. a comparison of the intensity of a candidate and the average intensity of the region), and the shape of a given region being grown.

**CHAPTER:5**

**CLUSTERING TECHNIQUES:**

**Clustering** is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

**5.1) Introduction**

Clustering can be considered the most important *unsupervised learning* problem; so, as every other problem of this kind, it deals with finding a *structure* in a collection of unlabeled data.

A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”.

A *cluster* is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

**5.2)The Goals of Clustering**

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such away that the result of the clustering will suit their needs.

For instance, we could be interested in finding representatives for homogeneous groups (*data reduction*), in finding “natural clusters” and describe their unknown properties (*“natural” data types*), in finding useful and suitable groupings (*“useful”* *data classes*) or in finding unusual data objects (*outlier detection*).

**5.3)Possible Applications**

Clustering in many fields, for instance algorithms can be applied:

* ***Marketing***: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
* ***Biology***: classification of plants and animals given their features;
* ***Libraries***: book ordering;
* ***Insurance***: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
* ***City-planning***: identifying groups of houses according to their house type, value and geographical location;
* ***Earthquake studies***: clustering observed earthquake epicenters to identify dangerous zones;
* ***WWW***: document classification; clustering weblog data to discover groups of similar access pattern.

**5.4)K-means clustering**

The *K*-means algorithm assigns each point to the cluster whose center (also called centroid) is nearest. The center is the average of all the points in the cluster — that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster.

***Example****:* The data set has three dimensions and the cluster has two points: *X* =

(*x*1, *x*2, *x*3) and *Y* = (*y*1, *y*2, *y*3). Then the centroid *Z* becomes *Z* = (*z*1, *z*2, *z*3),

where *z*1 = (*x*1 + *y*1)/2 and *z*2 = (*x*2 + *y*2)/2 and *z*3 = (*x*3 + *y*3)/2.

The algorithm steps are (J. MacQueen, 1967):

• Choose the number of clusters, *k*.

• Randomly generate *k* clusters and determine the cluster centers, or directly

generate *k* random points as cluster centers.

• Assign each point to the nearest cluster center.

• Recompute the new cluster centers.

• Repeat the two previous steps until some convergence criterion is met (usually

that the assignment hasn't changed).

**ADVANTAGE**:The main advantages of this algorithm are its simplicity and speed which allows it to run on large datasets.

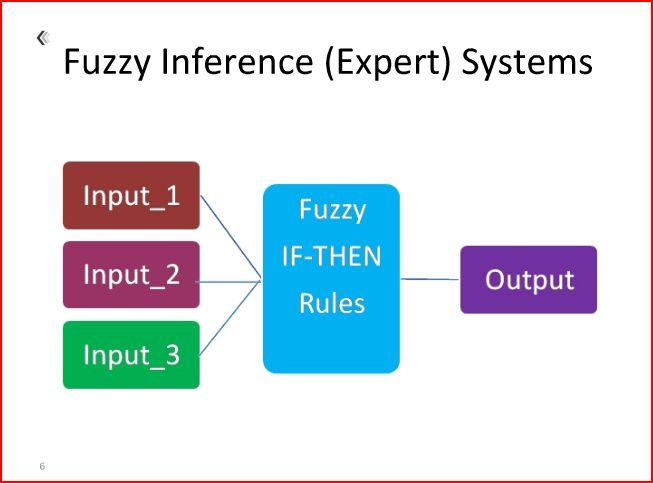
**DISADVANTAGE**:Its disadvantage is that it does not yield the same result with

each run, since the resulting clusters depend on the initial random assignments. It

minimizes intra-cluster variance, but does not ensure that the result has a global

minimum of variance.

**5.5)Fuzzy c-Means Algorithm**



The fuzzy C-Means algorithm (FCM) generalizes the hard c-mans algorithm to allow a point to partially belong to multiple clusters. Therefore, it produces a soft partition for a given dataset. In fact, it produces a constrained soft partition. To do this, the objective function *J1* of hard c-means has been extended in two ways:

(1) The fuzzy membership degrees in clusters were incorporated into the formula, and

(2) An additional parameter *m* was introduced as a weight exponent in the fuzzy membership.

The extended objective function [3], denoted *Jm*, is:

where *P* is fuzzy partition of the dataset *X* formed by *C1,C2,…,Ck*. The parameter *m* is a weight that determines the degree to which partial members of a clusters affect the clustering result.

Like hard c-means, fuzzy c-means also tries to find a good partition by searching for prototypes *vi* that minimize the objective function *Jm*. Unlike hard c-means, however, the fuzzy c-means algorithm also needs to search for membership functions that minimize *Jm*. To accomplish these two objectives, a necessary condition for local minimum of *Jm* was derived from *Jm*. This condition, which is formally stated below, serves as the foundation of the fuzzy c-means algorithm.

**Theorem.** Fuzzy c-means theorem. A constrained fuzzy partition {*C1,C2,…,Ck*} can be a local minimum of objective function *Jm* only if the following conditions are satisfied:

1≤ i≤ k,

1≤ i≤ k

Based on this theorem, FCM updates the prototypes and the membership function iteratively using equations 2 and 3 until a convergence criterion is reached. We describe the algorithm below.

FCM (*X, c, m,* ε*)*

*X*: an unlabeled data set.

*c*: the number the clusters.

*m*: the parameter in the objective function.

ε: a threshold for the convergence criteria.

Initialize prototype V={*v1,v2,…,vc*}

Repeat

*VPrevious* ← *V*

Compute membership functions using equations 3.

Update the prototype, *vi* in *V* using equation 2.

Until

Suppose we are given a dataset of six points, each of which has two features *F1* and *F2*. We list the dataset in table 1. Assuming that we want to use FCM to partition the dataset into two clusters (i.e., the parameter *c*=2), suppose we set the parameter *m* in FCM at 2, and the initial prototypes to *v1*=(5,5) and *v2*=(10,10).

|  |  |  |
| --- | --- | --- |
|  | F1 | F2 |
| X1 | 2 | 12 |
| X2 | 4 | 9 |
| X3 | 7 | 13 |
| X4 | 11 | 5 |
| X5 | 12 | 7 |
| X6 | 14 | 4 |

Table 1:Dataset Values



Figure 1:Dataset Graphical Representation

The initial membership functions of the two clusters are calculated using equation 2.

||x1-v1||2 = 32+72 = 58

||x1-v2||2 = 82+22 = 68

=0.5397

Similarly, we obtain the following

=0.4603

= 0.6852

= 0.3148

= 0.2093

= 0.7907

= 0.4194

= 0.5806

= 0.197

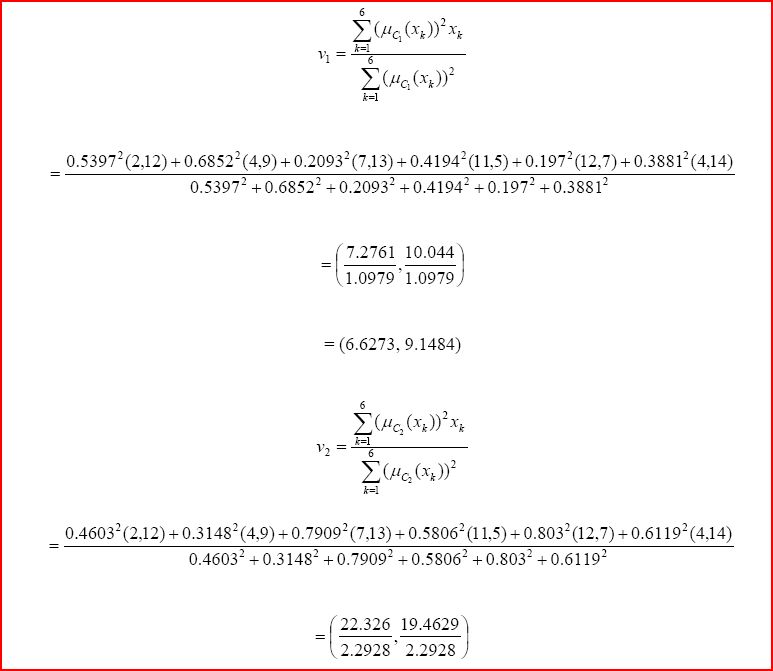
= 0.803

= 0.3881

= 0.6119

Therefore, using these initial prototypes of the two clusters, membership function indicated that *x1* and *x2* are more in the first cluster, while the remaining points in the dataset are more in the second cluster.

The FCM algorithm then updates the prototypes according to equation 3



The updated prototype *v1*, as is shown in fig 5, is moved closer to the center of the cluster formed by *x1*, *x2* and *x3*; while the updated prototype *v2* is moved closer to the cluster formed by *x4*, *x5* and *x6*.



Figure 2:Prototype Updating

We wish to make a few important points regarding the FCM algorithm:

- FCM is guaranteed to converge for *m*>1. This important convergence theorem was established in 1980.

- FCM finds a local minimum (or saddle point) of the objective function *Jm*. This is because the FCM theorem (theorem 1) is derived from the condition that the gradient of the objective function *Jm* should be 0 at an FCM solution, which is satisfied by all local minima and saddle points.

- The result of applying FCM to a given dataset depends not only on the choice of parameters *m* and *c*, but also on the choice of initial prototypes.

The fuzzy *c*-means algorithm is very similar to the *k*-means algorithm:

* Choose a number of clusters.
* Assign randomly to each point coefficients for being in the clusters.
* Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than ε, the given sensitivity threshold) :
  + Compute the centroid for each cluster, using the formula above.
  + For each point, compute its coefficients of being in the clusters, using the formula above.

The algorithm minimizes intra-cluster variance as well, but has the same problems as *k*-means, the minimum is a local minimum, and the results depend on the initial choice of weights. The Expectation-maximization algorithm is a more statistically formalized method which includes some of these ideas: partial membership in classes. It has better convergence properties and is in general preferred to fuzzy-c-mean.

**CHAPTER:6**

**SEGMENTATION OF MRI IMAGE OF BRAIN:**

The field known as biomedical analysis has evolved considerably over the last couple of decades. The widespread availability of suitable detectors has aided the rapid development of new technologies for the monitoring and diagnosis, as well as treatment, of patients. Over the last century technology has advanced from the discovery of xrays

to a variety of imaging tools such as MRI, Computed Tomography (CT), Positron Emission Tomography (PET) and ultrasonography. Three-dimensional (3-D) processing and visualization of medical images is a rapidly growing area of research and MRI has provided a means for imaging tissue at very high resolutions providing the desired information for use in fields like reparative surgery, radiotherapy treatment planning, stereotactic neurosurgery, and others.Furthermore, new techniques are helping to advance fundamental biomedical research. Image segmentation plays a major role in the field of biomedical applications. The segmentation technique is widely used by the radiologists to segment the input medical image into meaningful regions. The specific application of this technique is to detect the tumor region by segmenting the abnormal MR input image. The size of the tumor region can be tracked using these techniques which aid the radiologists in treatment planning. The primitive techniques are based on manual segmentation which is a time consuming process besides being susceptible to human errors. Several automated techniques have been developed which removes the drawbacks of manual segmentation. Clustering is one of the widely used image segmentation techniques which classify patterns in such a way that samples of the same group are more similar to one another than samples belonging to different groups [3,9,10]. There has been considerable interest recently in the use of fuzzy clustering methods, which retain more information from the original image than hard clustering methods. Fuzzy C-means algorithm is widely preferred because of its additional flexibility which allows pixels to belong to multiple classes with varying degrees of membership. But the major operational

complaint is that the FCM technique is time consuming.The drawback of the FCM is improved by the improved FCM algorithm.Magnetic Resonance Imaging (MRI) is an advanced medical imaging technique providing rich information about the human soft tissue anatomy. Segmentation of MR images into different tissue classes, especially grey matter (GM), white matter (WM) and cerebrospinal fluid (CSF) is an important task. Accurate and robust brain tissue segmentation from magnetic resonance (MR) images is key issue in many applications of medical image analysis for quantitative studies and particularly in the study of several brain disorders. A wide variety of approaches have been proposed for brain MR image segmentation.

**CHAPTER:7**

**MATLAB CODES**

**7.1)MATLAB CODE FOR FUZZY-C-MEANS CLUSTERING**

**I=imread(‘mri.tif’);**

**I1=unique(I);**

**[maxrow maxcol]=size(I1);**

**n=(maxrow)\*(maxcol);**

**class=3;**

**m=2;**

**L=[1 0 0;0 1 0;0 0 1];**

**L1=[1 0 0;0 1 0;0 0 1];**

**for J=1:(n-3)/3**

**L=vertcat(L,L1);**

**end**

**A=[1 0 0];**

**U=vertcat(L,A);**

**Uprev=U;**

**for l=1:class**

**for p=1:n**

**B(p,l)=(U(p,l).^m).\*I1(p,1);**

**end**

**end**

**s=sum(B);**

**for l=1:1:class**

**Zc(1,l)=s(1,l)/sum(U(:,l).^m);**

**end**

**for j=1:class**

**for i=1:n**

**Dist(i,j)=abs(I1(i,1)-Zc(1,j));**

**end**

**end**

**for j=1:class**

**for i=1:n**

**if (Dist(i,j)==0)**

**U(i,j)=1;**

**else**

**ss=0;**

**for k=1:class**

**ss=ss+((Dist(i,j)/Dist(i,k))^2/(m-1));**

**U(i,j)=1/ss;**

**end**

**end**

**end**

**end**

**XX=abs(U-Uprev);**

**AAA=0.01\*ones(88,1);**

**iter=1;**

**while(iter<=10000)**

**Uprev=U;**

**for l=1:class**

**for p=1:n**

**B(p,l)=(U(p,l).^m).\*I1(p,1);**

**end**

**end**

**s=sum(B);**

**for l=1:1:class**

**Zc(1,l)=s(1,l)/sum(U(:,l).^m);**

**end**

**for j=1:class**

**for i=1:n**

**Dist(i,j)=abs(I1(i,1)-Zc(1,j));**

**end**

**end**

**for j=1:class**

**for i=1:n**

**if (Dist(i,j)==0)**

**U(i,j)=1;**

**else**

**ss=0;**

**for k=1:class**

**ss=ss+((Dist(i,j)/Dist(i,k))^2/(m-1));**

**U(i,j)=1/ss;**

**end**

**end**

**end**

**end**

**iter=iter+1**

**end**

**Newclassifiedmatrix=zeros(n,class);**

**for i=1:n**

**[best classno]=max(U(i,:));**

**Newclassifiedmatrix(i,classno)=1;**

**end**

**7.2)CODE FOR IMAGE SEGMENTATION**

**f=imread('mri.tif');**

**imshow(I)**

**[M N]=size(I);**

**for i=1:M**

**for j=1:N**

**if (I(i,j)<=T1)**

**g(i, j)=Zc1;**

**elseif (I(i,j)>T1&I(i,j)<=T2)**

**g(i,j)=Zc2;**

**else**

**g(i,j)=Zc3;**

**end**

**end**

**end**

**g**

**x=uint8(g);**

**figure, imshow(x);**

**7.3)ALTERNATIVE CODE FOR SEGMENTATION:**

**for k=1:769**

**for l=1:777**

**if(I(k,l)<d)**

**I(k,l)=0;**

**if(d<=I(k,l)&&I(k,l)<=q)**

**I(k,l)=0.5;**

**else**

**I(k,l)=1;**

**end**

**end**

**end**

**end**

**7.4)CODES FOR CONVERTING RGB IMAGE INTO DOUBLE GRAY SCALE IMAGE:**

**I=rgb2gray(i);**

**I=im@double(I);**

To check the type of image we have to type

**whos I**

**CHAPTER:8**

**OBSERVATION AND RESULTS:**

**8.1)Segmentation on Brain Image:**

Zc =

0.5000 0.1521 0.8479

U =

0.0823 0.8891 0.0286

0.0797 0.8928 0.0275

0.0771 0.8965 0.0264

0.0744 0.9002 0.0254

0.0717 0.9040 0.0243

0.0691 0.9077 0.0232

0.0664 0.9114 0.0222

0.0637 0.9152 0.0211

0.0610 0.9189 0.0201

0.0583 0.9226 0.0191

0.0556 0.9263 0.0181

0.0529 0.9300 0.0171

0.0502 0.9337 0.0161

0.0476 0.9373 0.0151

0.0449 0.9410 0.0141

0.0422 0.9445 0.0132

0.0396 0.9481 0.0123

0.0370 0.9516 0.0114

0.0344 0.9551 0.0105

0.0319 0.9585 0.0097

0.0294 0.9618 0.0088

0.0269 0.9650 0.0080

0.0245 0.9682 0.0072

0.0222 0.9713 0.0065

0.0199 0.9743 0.0058

0.0177 0.9772 0.0051

0.0156 0.9800 0.0044

0.0135 0.9827 0.0038

0.0116 0.9852 0.0032

0.0098 0.9875 0.0027

0.0080 0.9897 0.0022

0.0065 0.9918 0.0018

0.0050 0.9936 0.0014

0.0037 0.9953 0.0010

0.0026 0.9967 0.0007

0.0017 0.9979 0.0004

0.0009 0.9988 0.0002

0.0004 0.9995 0.0001

0.0001 0.9999 0.0000

0.0000 1.0000 0.0000

0.0002 0.9998 0.0000

0.0007 0.9992 0.0002

0.0014 0.9983 0.0003

0.0025 0.9969 0.0006

0.0039 0.9952 0.0009

0.0056 0.9931 0.0013

0.0077 0.9905 0.0018

0.0103 0.9874 0.0023

0.0132 0.9839 0.0029

0.0166 0.9798 0.0037

0.0204 0.9752 0.0044

0.0247 0.9700 0.0053

0.0295 0.9643 0.0062

0.0348 0.9579 0.0073

0.0407 0.9509 0.0084

0.0471 0.9433 0.0095

0.0542 0.9351 0.0108

0.0618 0.9261 0.0121

0.0700 0.9165 0.0135

0.0789 0.9062 0.0150

0.0884 0.8951 0.0165

0.0985 0.8834 0.0181

0.1093 0.8709 0.0197

0.1208 0.8578 0.0214

0.1330 0.8439 0.0231

0.1458 0.8293 0.0249

0.1593 0.8140 0.0267

0.1735 0.7980 0.0285

0.1883 0.7814 0.0303

0.2037 0.7641 0.0322

0.2198 0.7462 0.0340

0.2365 0.7277 0.0358

0.2537 0.7087 0.0376

0.2715 0.6892 0.0393

0.2898 0.6692 0.0410

0.3086 0.6488 0.0427

0.3278 0.6280 0.0442

0.3474 0.6069 0.0457

0.3674 0.5855 0.0471

0.3876 0.5640 0.0484

0.4081 0.5423 0.0496

0.4288 0.5205 0.0507

0.4497 0.4986 0.0517

0.4706 0.4769 0.0525

0.4916 0.4552 0.0532

0.5126 0.4336 0.0538

0.5335 0.4123 0.0542

0.5543 0.3912 0.0545

0.5750 0.3705 0.0546

0.5954 0.3501 0.0545

0.6155 0.3301 0.0543

0.6354 0.3106 0.0540

0.6549 0.2916 0.0535

0.6741 0.2730 0.0529

0.6928 0.2551 0.0521

0.7111 0.2377 0.0511

0.7289 0.2210 0.0501

0.7463 0.2049 0.0489

0.7631 0.1894 0.0475

0.7793 0.1746 0.0461

0.7950 0.1605 0.0445

0.8101 0.1470 0.0429

0.8247 0.1342 0.0411

0.8386 0.1221 0.0393

0.8520 0.1107 0.0374

0.8647 0.0999 0.0354

0.8769 0.0897 0.0334

0.8884 0.0803 0.0313

0.8994 0.0714 0.0292

0.9097 0.0632 0.0271

0.9195 0.0555 0.0250

0.9287 0.0485 0.0228

0.9372 0.0420 0.0207

0.9453 0.0361 0.0187

0.9527 0.0307 0.0166

0.9595 0.0258 0.0146

0.9659 0.0214 0.0127

0.9716 0.0175 0.0109

0.9768 0.0141 0.0091

0.9815 0.0110 0.0075

0.9856 0.0084 0.0060

0.9892 0.0062 0.0046

0.9923 0.0043 0.0034

0.9948 0.0028 0.0023

0.9969 0.0017 0.0014

0.9984 0.0008 0.0008

0.9994 0.0003 0.0003

0.9999 0.0000 0.0000

0.9999 0.0000 0.0000

0.9994 0.0003 0.0003

0.9984 0.0008 0.0008

0.9969 0.0014 0.0017

0.9948 0.0023 0.0028

0.9923 0.0034 0.0043

0.9892 0.0046 0.0062

0.9856 0.0060 0.0084

0.9815 0.0075 0.0110

0.9768 0.0091 0.0141

0.9716 0.0109 0.0175

0.9659 0.0127 0.0214

0.9595 0.0146 0.0258

0.9527 0.0166 0.0307

0.9453 0.0187 0.0361

0.9372 0.0207 0.0420

0.9287 0.0228 0.0485

0.9195 0.0250 0.0555

0.9097 0.0271 0.0632

0.8994 0.0292 0.0714

0.8884 0.0313 0.0803

0.8769 0.0334 0.0897

0.8647 0.0354 0.0999

0.8520 0.0374 0.1107

0.8386 0.0393 0.1221

0.8247 0.0411 0.1342

0.8101 0.0429 0.1470

0.7950 0.0445 0.1605

0.7793 0.0461 0.1746

0.7631 0.0475 0.1894

0.7463 0.0489 0.2049

0.7289 0.0501 0.2210

0.7111 0.0511 0.2377

0.6928 0.0521 0.2551

0.6741 0.0529 0.2730

0.6549 0.0535 0.2916

0.6354 0.0540 0.3106

0.6155 0.0543 0.3301

0.5954 0.0545 0.3501

0.5750 0.0546 0.3705

0.5543 0.0545 0.3912

0.5335 0.0542 0.4123

0.5126 0.0538 0.4336

0.4916 0.0532 0.4552

0.4706 0.0525 0.4769

0.4497 0.0517 0.4986

0.4288 0.0507 0.5205

0.4081 0.0496 0.5423

0.3876 0.0484 0.5640

0.3674 0.0471 0.5855

0.3474 0.0457 0.6069

0.3278 0.0442 0.6280

0.3086 0.0427 0.6488

0.2898 0.0410 0.6692

0.2715 0.0393 0.6892

0.2537 0.0376 0.7087

0.2365 0.0358 0.7277

0.2198 0.0340 0.7462

0.2037 0.0322 0.7641

0.1883 0.0303 0.7814

0.1735 0.0285 0.7980

0.1593 0.0267 0.8140

0.1458 0.0249 0.8293

0.1330 0.0231 0.8439

0.1208 0.0214 0.8578

0.1093 0.0197 0.8709

0.0985 0.0181 0.8834

0.0884 0.0165 0.8951

0.0789 0.0150 0.9062

0.0700 0.0135 0.9165

0.0618 0.0121 0.9261

0.0542 0.0108 0.9351

0.0471 0.0095 0.9433

0.0407 0.0084 0.9509

0.0348 0.0073 0.9579

0.0295 0.0062 0.9643

0.0247 0.0053 0.9700

0.0204 0.0044 0.9752

0.0166 0.0037 0.9798

0.0132 0.0029 0.9839

0.0103 0.0023 0.9874

0.0077 0.0018 0.9905

0.0056 0.0013 0.9931

0.0039 0.0009 0.9952

0.0025 0.0006 0.9969

0.0014 0.0003 0.9983

0.0007 0.0002 0.9992

0.0002 0.0000 0.9998

0.0000 0.0000 1.0000

0.0001 0.0000 0.9999

0.0004 0.0001 0.9995

0.0009 0.0002 0.9988

0.0017 0.0004 0.9979

0.0026 0.0007 0.9967

0.0037 0.0010 0.9953

0.0050 0.0014 0.9936

0.0065 0.0018 0.9918

0.0080 0.0022 0.9897

0.0098 0.0027 0.9875

0.0116 0.0032 0.9852

0.0135 0.0038 0.9827

0.0156 0.0044 0.9800

0.0177 0.0051 0.9772

0.0199 0.0058 0.9743

0.0222 0.0065 0.9713

0.0245 0.0072 0.9682

0.0269 0.0080 0.9650

0.0294 0.0088 0.9618

0.0319 0.0097 0.9585

0.0344 0.0105 0.9551

0.0370 0.0114 0.9516

0.0396 0.0123 0.9481

0.0422 0.0132 0.9445

0.0449 0.0141 0.9410

0.0476 0.0151 0.9373

0.0502 0.0161 0.9337

0.0529 0.0171 0.9300

0.0556 0.0181 0.9263

0.0583 0.0191 0.9226

0.0610 0.0201 0.9189

0.0637 0.0211 0.9152

0.0664 0.0222 0.9114

0.0691 0.0232 0.9077

0.0717 0.0243 0.9040

0.0744 0.0254 0.9002

0.0771 0.0264 0.8965

0.0797 0.0275 0.8928

0.0823 0.0286 0.8891

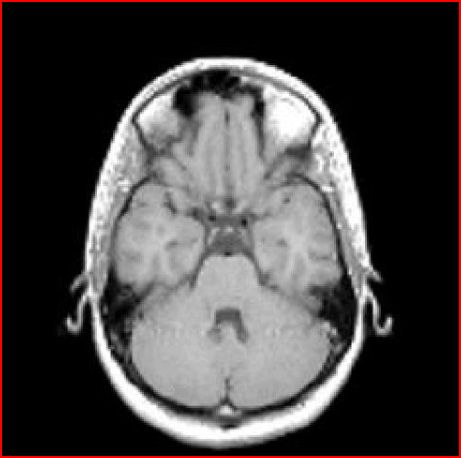


Figure 3:REAL IMAGE

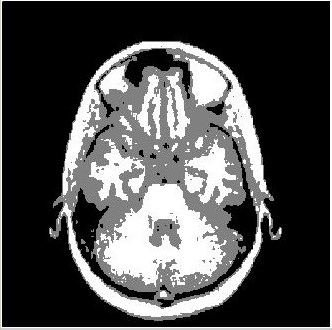
****

Figure 4:SEGMENTAED IMAGE

**8.2)Segmentation of a face picture:**

Zc =

0.4310 0.1359 0.7261

U =

0.0877 0.8814 0.0309

0.0846 0.8858 0.0296

0.0816 0.8901 0.0283

0.0785 0.8945 0.0270

0.0691 0.9077 0.0232

0.0659 0.9121 0.0220

0.0627 0.9165 0.0208

0.0596 0.9209 0.0195

0.0564 0.9253 0.0183

0.0532 0.9296 0.0172

0.0500 0.9340 0.0160

0.0469 0.9383 0.0149

0.0437 0.9425 0.0137

0.0406 0.9467 0.0126

0.0376 0.9509 0.0116

0.0345 0.9550 0.0105

0.0315 0.9590 0.0095

0.0286 0.9629 0.0086

0.0257 0.9667 0.0076

0.0229 0.9704 0.0067

0.0202 0.9739 0.0059

0.0176 0.9774 0.0051

0.0151 0.9806 0.0043

0.0127 0.9837 0.0036

0.0105 0.9866 0.0029

0.0084 0.9892 0.0023

0.0066 0.9917 0.0018

0.0049 0.9938 0.0013

0.0034 0.9957 0.0009

0.0021 0.9973 0.0006

0.0012 0.9985 0.0003

0.0005 0.9994 0.0001

0.0001 0.9999 0.0000

0.0000 1.0000 0.0000

0.0003 0.9996 0.0001

0.0010 0.9987 0.0002

0.0022 0.9973 0.0005

0.0037 0.9954 0.0009

0.0058 0.9929 0.0013

0.0084 0.9897 0.0019

0.0115 0.9859 0.0026

0.0153 0.9814 0.0034

0.0196 0.9761 0.0043

0.0246 0.9701 0.0053

0.0303 0.9633 0.0064

0.0368 0.9556 0.0076

0.0440 0.9470 0.0090

0.0520 0.9376 0.0104

0.0608 0.9272 0.0119

0.0705 0.9159 0.0136

0.0811 0.9036 0.0153

0.0925 0.8903 0.0171

0.1049 0.8761 0.0190

0.1182 0.8608 0.0210

0.1324 0.8445 0.0231

0.1476 0.8273 0.0251

0.1637 0.8091 0.0273

0.1807 0.7899 0.0294

0.1986 0.7698 0.0316

0.2174 0.7489 0.0337

0.2370 0.7272 0.0358

0.2574 0.7047 0.0379

0.2785 0.6815 0.0400

0.3004 0.6577 0.0419

0.3229 0.6333 0.0438

0.3459 0.6085 0.0456

0.3695 0.5833 0.0472

0.3934 0.5578 0.0488

0.4177 0.5322 0.0501

0.4422 0.5065 0.0513

0.4669 0.4808 0.0524

0.4916 0.4552 0.0532

0.5163 0.4298 0.0538

0.5410 0.4047 0.0543

0.5654 0.3801 0.0545

0.5896 0.3559 0.0545

0.6135 0.3322 0.0543

0.6369 0.3092 0.0539

0.6599 0.2868 0.0533

0.6823 0.2652 0.0525

0.7041 0.2444 0.0515

0.7253 0.2245 0.0503

0.7458 0.2054 0.0489

0.7655 0.1872 0.0473

0.7845 0.1699 0.0456

0.8028 0.1536 0.0437

0.8202 0.1382 0.0416

0.8368 0.1237 0.0395

0.8526 0.1102 0.0373

0.8675 0.0976 0.0349

0.8816 0.0859 0.0325

0.8949 0.0750 0.0301

0.9073 0.0651 0.0276

0.9189 0.0560 0.0251

0.9297 0.0477 0.0226

0.9397 0.0402 0.0201

0.9489 0.0335 0.0177

0.9573 0.0274 0.0153

0.9649 0.0221 0.0130

0.9717 0.0175 0.0109

0.9778 0.0134 0.0088

0.9831 0.0100 0.0069

0.9877 0.0071 0.0052

0.9915 0.0048 0.0037

0.9946 0.0030 0.0024

0.9970 0.0016 0.0014

0.9987 0.0007 0.0006

0.9997 0.0001 0.0001

1.0000 0.0000 0.0000

0.9996 0.0002 0.0002

0.9984 0.0007 0.0008

0.9966 0.0016 0.0018

0.9940 0.0027 0.0033

0.9907 0.0040 0.0052

0.9867 0.0056 0.0077

0.9820 0.0073 0.0107

0.9765 0.0092 0.0142

0.9703 0.0113 0.0184

0.9633 0.0135 0.0232

0.9556 0.0158 0.0286

0.9470 0.0182 0.0348

0.9377 0.0206 0.0417

0.9275 0.0231 0.0493

0.9166 0.0256 0.0578

0.9048 0.0281 0.0671

0.8922 0.0306 0.0772

0.8788 0.0331 0.0882

0.8645 0.0355 0.1001

0.8494 0.0378 0.1129

0.8334 0.0400 0.1266

0.8167 0.0421 0.1412

0.7991 0.0441 0.1568

0.7807 0.0460 0.1733

0.7615 0.0477 0.1908

0.7416 0.0492 0.2092

0.7210 0.0506 0.2284

0.6997 0.0518 0.2485

0.6778 0.0527 0.2695

0.6552 0.0535 0.2912

0.6322 0.0541 0.3137

0.6087 0.0545 0.3369

0.5848 0.0546 0.3606

0.5606 0.0545 0.3849

0.5361 0.0543 0.4097

0.5114 0.0538 0.4348

0.4867 0.0531 0.4602

0.4620 0.0522 0.4858

0.4373 0.0512 0.5115

0.4129 0.0499 0.5372

0.3887 0.0485 0.5628

0.3648 0.0470 0.5882

0.3414 0.0453 0.6133

0.3184 0.0435 0.6381

0.2961 0.0416 0.6623

0.2744 0.0396 0.6860

0.2534 0.0376 0.7091

0.2331 0.0355 0.7314

0.2137 0.0333 0.7530

0.1951 0.0312 0.7737

0.1774 0.0290 0.7936

0.1605 0.0269 0.8126

0.1446 0.0248 0.8306

0.1296 0.0227 0.8477

0.1156 0.0207 0.8637

0.1025 0.0187 0.8788

0.0903 0.0168 0.8929

0.0790 0.0150 0.9060

0.0686 0.0133 0.9181

0.0591 0.0117 0.9292

0.0504 0.0101 0.9394

0.0426 0.0087 0.9487

0.0355 0.0074 0.9571

0.0292 0.0062 0.9646

0.0236 0.0051 0.9713

0.0188 0.0041 0.9771

0.0145 0.0032 0.9823

0.0109 0.0025 0.9866

0.0079 0.0018 0.9903

0.0054 0.0013 0.9934

0.0034 0.0008 0.9958

0.0019 0.0005 0.9976

0.0009 0.0002 0.9989

0.0002 0.0001 0.9997

0.0000 0.0000 1.0000

0.0001 0.0000 0.9998

0.0006 0.0001 0.9993

0.0013 0.0003 0.9983

0.0024 0.0006 0.9970

0.0036 0.0010 0.9954

0.0052 0.0014 0.9935

0.0069 0.0019 0.9912

0.0088 0.0024 0.9888

0.0109 0.0030 0.9861

0.0131 0.0037 0.9832

0.0155 0.0044 0.9800

0.0180 0.0052 0.9768

0.0207 0.0060 0.9733

0.0234 0.0069 0.9697

0.0262 0.0078 0.9660

0.0291 0.0087 0.9622

0.0320 0.0097 0.9583

0.0350 0.0107 0.9542

0.0381 0.0118 0.9502

0.0412 0.0128 0.9460

0.0443 0.0139 0.9418

0.0474 0.0151 0.9375

0.0506 0.0162 0.9332

0.0538 0.0174 0.9289

0.0569 0.0186 0.9245

0.0601 0.0198 0.9201

0.0633 0.0210 0.9157

0.0664 0.0222 0.9113

0.0696 0.0235 0.9069

0.0727 0.0247 0.9025

0.0759 0.0260 0.8981

0.0790 0.0273 0.8938

0.0821 0.0285 0.8894

****

Figure 5:Real Image

****

Figure 6:Segmented Image

**8.3)Segmentation of another picture:**

Zc=

0.5000 0.1521 0.8479

U =

0.0823 0.8891 0.0286

0.0797 0.8928 0.0275

0.0771 0.8965 0.0264

0.0744 0.9002 0.0254

0.0717 0.9040 0.0243

0.0691 0.9077 0.0232

0.0664 0.9114 0.0222

0.0637 0.9152 0.0211

0.0610 0.9189 0.0201

0.0583 0.9226 0.0191

0.0556 0.9263 0.0181

0.0529 0.9300 0.0171

0.0502 0.9337 0.0161

0.0476 0.9373 0.0151

0.0449 0.9410 0.0141

0.0422 0.9445 0.0132

0.0396 0.9481 0.0123

0.0370 0.9516 0.0114

0.0344 0.9551 0.0105

0.0319 0.9585 0.0097

0.0294 0.9618 0.0088

0.0269 0.9650 0.0080

0.0245 0.9682 0.0072

0.0222 0.9713 0.0065

0.0199 0.9743 0.0058

0.0177 0.9772 0.0051

0.0156 0.9800 0.0044

0.0135 0.9827 0.0038

0.0116 0.9852 0.0032

0.0098 0.9875 0.0027

0.0080 0.9897 0.0022

0.0065 0.9918 0.0018

0.0050 0.9936 0.0014

0.0037 0.9953 0.0010

0.0026 0.9967 0.0007

0.0017 0.9979 0.0004

0.0009 0.9988 0.0002

0.0004 0.9995 0.0001

0.0001 0.9999 0.0000

0.0000 1.0000 0.0000

0.0002 0.9998 0.0000

0.0007 0.9992 0.0002

0.0014 0.9983 0.0003

0.0025 0.9969 0.0006

0.0039 0.9952 0.0009

0.0056 0.9931 0.0013

0.0077 0.9905 0.0018

0.0103 0.9874 0.0023

0.0132 0.9839 0.0029

0.0166 0.9798 0.0037

0.0204 0.9752 0.0044

0.0247 0.9700 0.0053

0.0295 0.9643 0.0062

0.0348 0.9579 0.0073

0.0407 0.9509 0.0084

0.0471 0.9433 0.0095

0.0542 0.9351 0.0108

0.0618 0.9261 0.0121

0.0700 0.9165 0.0135

0.0789 0.9062 0.0150

0.0884 0.8951 0.0165

0.0985 0.8834 0.0181

0.1093 0.8709 0.0197

0.1208 0.8578 0.0214

0.1330 0.8439 0.0231

0.1458 0.8293 0.0249

0.1593 0.8140 0.0267

0.1735 0.7980 0.0285

0.1883 0.7814 0.0303

0.2037 0.7641 0.0322

0.2198 0.7462 0.0340

0.2365 0.7277 0.0358

0.2537 0.7087 0.0376

0.2715 0.6892 0.0393

0.2898 0.6692 0.0410

0.3086 0.6488 0.0427

0.3278 0.6280 0.0442

0.3474 0.6069 0.0457

0.3674 0.5855 0.0471

0.3876 0.5640 0.0484

0.4081 0.5423 0.0496

0.4288 0.5205 0.0507

0.4497 0.4986 0.0517

0.4706 0.4769 0.0525

0.4916 0.4552 0.0532

0.5126 0.4336 0.0538

0.5335 0.4123 0.0542

0.5543 0.3912 0.0545

0.5750 0.3705 0.0546

0.5954 0.3501 0.0545

0.6155 0.3301 0.0543

0.6354 0.3106 0.0540

0.6549 0.2916 0.0535

0.6741 0.2730 0.0529

0.6928 0.2551 0.0521

0.7111 0.2377 0.0511

0.7289 0.2210 0.0501

0.7463 0.2049 0.0489

0.7631 0.1894 0.0475

0.7793 0.1746 0.0461

0.7950 0.1605 0.0445

0.8101 0.1470 0.0429

0.8247 0.1342 0.0411

0.8386 0.1221 0.0393

0.8520 0.1107 0.0374

0.8647 0.0999 0.0354

0.8769 0.0897 0.0334

0.8884 0.0803 0.0313

0.8994 0.0714 0.0292

0.9097 0.0632 0.0271

0.9195 0.0555 0.0250

0.9287 0.0485 0.0228

0.9372 0.0420 0.0207

0.9453 0.0361 0.0187

0.9527 0.0307 0.0166

0.9595 0.0258 0.0146

0.9659 0.0214 0.0127

0.9716 0.0175 0.0109

0.9768 0.0141 0.0091

0.9815 0.0110 0.0075

0.9856 0.0084 0.0060

0.9892 0.0062 0.0046

0.9923 0.0043 0.0034

0.9948 0.0028 0.0023

0.9969 0.0017 0.0014

0.9984 0.0008 0.0008

0.9994 0.0003 0.0003

0.9999 0.0000 0.0000

0.9999 0.0000 0.0000

0.9994 0.0003 0.0003

0.9984 0.0008 0.0008

0.9969 0.0014 0.0017

0.9948 0.0023 0.0028

0.9923 0.0034 0.0043

0.9892 0.0046 0.0062

0.9856 0.0060 0.0084

0.9815 0.0075 0.0110

0.9768 0.0091 0.0141

0.9716 0.0109 0.0175

0.9659 0.0127 0.0214

0.9595 0.0146 0.0258

0.9527 0.0166 0.0307

0.9453 0.0187 0.0361

0.9372 0.0207 0.0420

0.9287 0.0228 0.0485

0.9195 0.0250 0.0555

0.9097 0.0271 0.0632

0.8994 0.0292 0.0714

0.8884 0.0313 0.0803

0.8769 0.0334 0.0897

0.8647 0.0354 0.0999

0.8520 0.0374 0.1107

0.8386 0.0393 0.1221

0.8247 0.0411 0.1342

0.8101 0.0429 0.1470

0.7950 0.0445 0.1605

0.7793 0.0461 0.1746

0.7631 0.0475 0.1894

0.7463 0.0489 0.2049

0.7289 0.0501 0.2210

0.7111 0.0511 0.2377

0.6928 0.0521 0.2551

0.6741 0.0529 0.2730

0.6549 0.0535 0.2916

0.6354 0.0540 0.3106

0.6155 0.0543 0.3301

0.5954 0.0545 0.3501

0.5750 0.0546 0.3705

0.5543 0.0545 0.3912

0.5335 0.0542 0.4123

0.5126 0.0538 0.4336

0.4916 0.0532 0.4552

0.4706 0.0525 0.4769

0.4497 0.0517 0.4986

0.4288 0.0507 0.5205

0.4081 0.0496 0.5423

0.3876 0.0484 0.5640

0.3674 0.0471 0.5855

0.3474 0.0457 0.6069

0.3278 0.0442 0.6280

0.3086 0.0427 0.6488

0.2898 0.0410 0.6692

0.2715 0.0393 0.6892

0.2537 0.0376 0.7087

0.2365 0.0358 0.7277

0.2198 0.0340 0.7462

0.2037 0.0322 0.7641

0.1883 0.0303 0.7814

0.1735 0.0285 0.7980

0.1593 0.0267 0.8140

0.1458 0.0249 0.8293

0.1330 0.0231 0.8439

0.1208 0.0214 0.8578

0.1093 0.0197 0.8709

0.0985 0.0181 0.8834

0.0884 0.0165 0.8951

0.0789 0.0150 0.9062

0.0700 0.0135 0.9165

0.0618 0.0121 0.9261

0.0542 0.0108 0.9351

0.0471 0.0095 0.9433

0.0407 0.0084 0.9509

0.0348 0.0073 0.9579

0.0295 0.0062 0.9643

0.0247 0.0053 0.9700

0.0204 0.0044 0.9752

0.0166 0.0037 0.9798

0.0132 0.0029 0.9839

0.0103 0.0023 0.9874

0.0077 0.0018 0.9905

0.0056 0.0013 0.9931

0.0039 0.0009 0.9952

0.0025 0.0006 0.9969

0.0014 0.0003 0.9983

0.0007 0.0002 0.9992

0.0002 0.0000 0.9998

0.0000 0.0000 1.0000

0.0001 0.0000 0.9999

0.0004 0.0001 0.9995

0.0009 0.0002 0.9988

0.0017 0.0004 0.9979

0.0026 0.0007 0.9967

0.0037 0.0010 0.9953

0.0050 0.0014 0.9936

0.0065 0.0018 0.9918

0.0080 0.0022 0.9897

0.0098 0.0027 0.9875

0.0116 0.0032 0.9852

0.0135 0.0038 0.9827

0.0156 0.0044 0.9800

0.0177 0.0051 0.9772

0.0199 0.0058 0.9743

0.0222 0.0065 0.9713

0.0245 0.0072 0.9682

0.0269 0.0080 0.9650

0.0294 0.0088 0.9618

0.0319 0.0097 0.9585

0.0344 0.0105 0.9551

0.0370 0.0114 0.9516

0.0396 0.0123 0.9481

0.0422 0.0132 0.9445

0.0449 0.0141 0.9410

0.0476 0.0151 0.9373

0.0502 0.0161 0.9337

0.0529 0.0171 0.9300

0.0556 0.0181 0.9263

0.0583 0.0191 0.9226

0.0610 0.0201 0.9189

0.0637 0.0211 0.9152

0.0664 0.0222 0.9114

0.0691 0.0232 0.9077

0.0717 0.0243 0.9040

0.0744 0.0254 0.9002

0.0771 0.0264 0.8965

0.0797 0.0275 0.8928

0.0797 0.0275 0.8928

0.0823 0.0286 0.8891

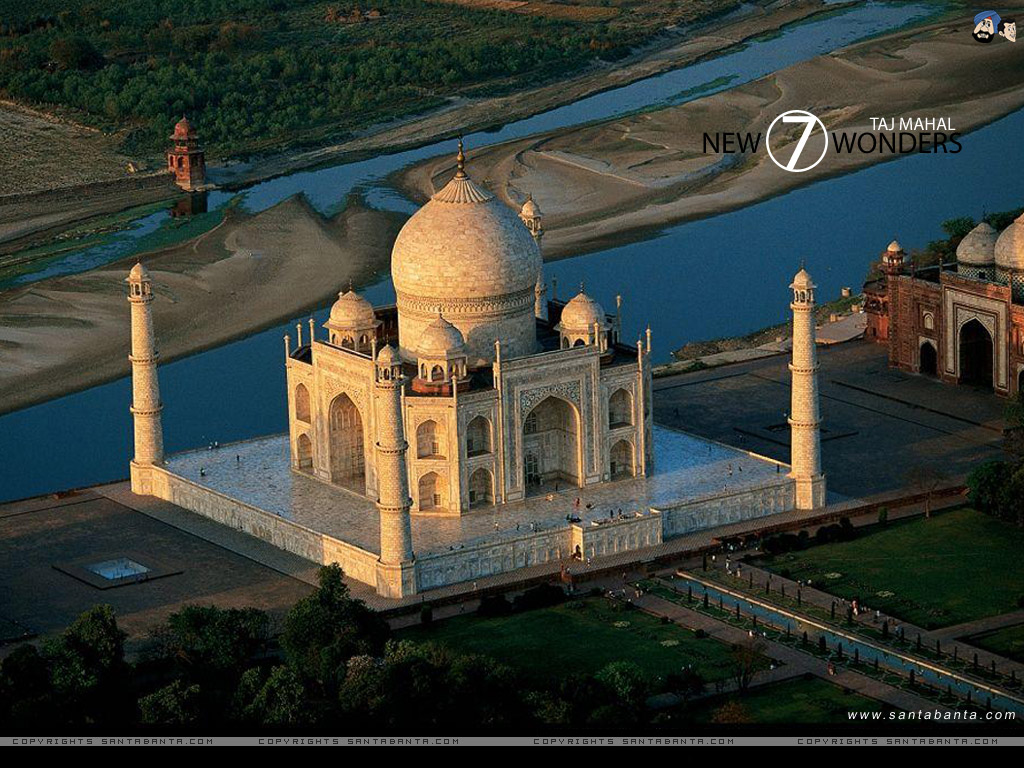


Figure 7:Real Image

****

Figure 8:Segmented Image

**CHAPTER:9**

**Summary**

This reviews and summarizes some existing methods of image segmentation and their drawbacks. It is well known that no method is equally good for all images and all methods are not good for a particular type of image.Selection of an appropriate segmentation technique largely depends on the type of images and application areas. The important problem is how to make a quantitative evaluation of segmentation results. Such a quantitative measure would be quite useful for vision applications where automatic decisions are required. It is very difficult to find a single quantitative index for this purpose because such an index should take into account many factors like homogeneity, contrast,compactness, continuity, psycho-visual perception etc. Possibly the human being is the best judge to evaluate the output of any segmentation algorithm.

**CHAPTER:10**

**Reference:**

* **Digital Image processing** by Gonzalez & Woods
* **Fundamentals of DIP** by Annadurai & Shanmugalakshmi
* Journals on **Histogram Based fuzzy C-mean Algorithm for Image Segmentation** by Ye Xiu Qing,Huang Zhen Hua,Xiao Qiang Zhejiang University Hangzhou,P.R.China
* **Fuzzy segmentation applied to face segmentation** Technical Report Erik Cuevas, Daniel Zaldivar, and Raul Rojas,Freie Universität Berlin, Institute of Informatik Takusstr,Berlin, Germany
* **Improving Fuzzy Algorithms for Automatic Magnetic Resonance Image Segmentation ,**Elnomery Zanaty and Sultan Aljahdali,College of Computer Science, Taif University, Saudi Arabia
* Canadian Journal on Image Processing & Computer Vision, **Improved FCM Algorithm applied to Color Image Segmentation** by Mohamed Ali Mahjoub
* **Segmentation of Lip Images by Modified Fuzzy C-means**

Clustering Algorithm

**G.Sudhavani,** Dept. of ECE, R.V.R& J.C College of Engg. Guntur, Andhra Pradesh,India

**Dr.K.Sathyaprasad**, Director,Evaluation,JNT University,Kakinada, Andhra Pradesh, India.