CHAPTER 1
INTRODUCTION

1. Introduction

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analog image processing; it allows a much wider range of algorithms to be applied to input data, and can avoid problems such as the build-up of noise and signal distortion during processing. Image segmentation refers to the process of partitioning a digital image into multiple regions (set of pixels). 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 an image.

In this thesis the various popular segmentation algorithms for image segmentation are given. Various methods for better grouping and segmentation have been developed. The algorithms or methods developed are meant for MRI images, CT images, and dermatology images in real time applications.

1.1 Fundamentals of Digital Image Processing

Digital image processing is a subset of the electronic domain wherein the image is converted into an array of small integers, called pixels (derived from picture element), representing a physical quantity such as scene radiance, stored in a digital memory, and processed by computer or other digital hardware. Digital image processing, [35] either as enhancement for human observers or performing autonomous analysis, offers advantages in
cost, speed, and flexibility, and with the rapidly falling price and rising performance of personal computers it has become the dominant method in use. An image is denoted by two dimensional functions of the form $f(x,y)$. The value or amplitude of $f$ at spatial coordinates $(x,y)$ is a positive scalar quantity whose physical meaning is determined by the source of the image. In a digital image, $(x,y)$ and the magnitude of $f$ are all finite and discrete quantities.

It is a hard task to distinguish between the domains of image processing and any other related area such as computer vision. But the two areas are quite different in the kind of output we get from them. Computer vision is the science and technology of machines that see. As a scientific discipline, computer vision is concerned with the theory for building artificial systems that obtain information from images. The image data can take many forms, such as a video sequence, views from multiple cameras, or multidimensional data from a medical scanner. In computer vision, the input is a digital image and the output is some representation of its interesting features. Image processing is often used in computer vision as a pre-processing step. Image processing is defined as an area when both input and output are images.

As a technological discipline, computer vision seeks to apply the theories and models of computer vision to the construction of computer vision systems. The organization of a computer vision system is highly application dependent. Some systems are stand-alone applications which solve a specific measurement or detection problem, while other constitute a sub-system of a larger design which, for example, also contains sub-systems for control of mechanical actuators, planning, information databases, man-machine interfaces, etc. The specific implementation of a computer vision system also depends on if its functionality is
prespecified or if some part of it can be learned or modified during operation. There are, however, typical functions which are found in many computer vision systems.

1. **Image acquisition:** A digital image is produced by one or several image sensor which, besides various types of light-sensitive cameras, includes range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or color images), but can also be related to various physical measures, such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.

2. **Pre-processing:** Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method. Examples are
(a) Re-sampling in order to assure that the image coordinate system is correct.
(b) Noise reduction in order to assure that sensor noise does not introduce false information.
(c) Contrast enhancement to assure that relevant information can be detected.
(d) Scale space representation to enhance image structures at locally appropriate scales.

3. **Feature extraction:** Image features at various levels of complexity are extracted from the image data. Typical examples of such features are
(a) Lines, edges and ridges.
(b) Localized interest points such as corners, blobs or points.

More complex features may be related to texture, shape or motion.
4. Detection/Segmentation: At some point in the processing, a decision is made about which image points or regions of the image are relevant for further processing. Examples are
(a) Selection of a specific set of interest points
(b) Segmentation of one or multiple image regions which contain a specific object of interest.

5. High-level processing: At this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object.

The remaining processing deals with, for example:
(a) Verification that the data satisfy model-based and application specific assumptions.
(b) Estimation of application specific parameters, such as object pose or object size.
(c) Classifying a detected object into different categories

Hence it can be said that *image* segmentation forms an integral part of computer vision systems and *is more an area of computer vision than image processing.*

1.2 Image Segmentation

Segmentation of an image entails the division or separation of the image into regions of similar attribute. The basic attribute for segmentation is image amplitude- luminance for a monochrome image and color components for a color image. Image edges and textures [92] are also useful attributes for segmentation. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image.

Segmentation does not involve classifying each segment. The segmentation only subdivides the image; it does not attempt to recognize the individual segments or their relationships to one another.
Segmentation is the process of partitioning the image into non-intersecting regions such that each region is homogeneous and the union of no two adjacent regions is homogeneous. Formally, it can be defined as follows.

Definition 1: Let $F$ be the set of all pixels and $P()$ be a uniformity (homogeneity) predicate defined on groups of connected pixels, then segmentation is a partitioning of the set $F$ into a set of connected subsets or regions ($S_1, S_2, \cdots, S_n$) such that $\bigcup_{i=1}^{n} S_i = F$ with $S_i \cap S_j = \emptyset$ when $i \neq j$. The uniformity predicate $P(S_i)$ is true for all regions $S_i$, and $P(S_i \cup S_j)$ is false when $S_i$ is adjacent to $S_j$.

This definition can be applied to all types of images.

The goal of segmentation is typically to locate certain objects of interest which may be depicted in the image. Segmentation could therefore be seen as a computer vision problem. A simple example of segmentation is thresholding [4] a grayscale image with a fixed threshold $t$: each pixel $p$ is assigned to one of two classes, $P_0$ or $P_1$, depending on whether $I(p) < t$ or $I(p) \geq t$. For intensity images (i.e., those represented by point-wise intensity levels), four popular segmentation approaches are: Fuzzy clustering, threshold techniques, edge based methods, region-based techniques.

Fuzzy technique [1] has been applied for various methods used for image segmentation. Fuzzy image segmentation is increasing in popularity because of rapid extension of fuzzy set theory, the development of various fuzzy set based mathematical modeling, synergistic combination of fuzzy, genetic algorithm and neural network and its successful and practical application in image processing, pattern recognition and computer vision system. In this work fuzzy edge detector and fuzzy clustering based image
segmentation [90] are studied. Fuzzy based region growing methods are extensively used for image segmentation. Efficient fuzzy technique based region growing [93] method which would yield good segmentation results on application of some region tracking techniques.

Threshold techniques [76] make decisions based on local pixel information and are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Because spatial information is ignored, however, blurred region boundaries can create havoc. Edge-based methods center around contour detection: their weakness in connecting together broken contour lines make them, too, prone to failure in the presence of blurring.

A region-based method usually proceeds as follows: the image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria creates fragmentation; lenient one overlooks blurred boundaries and over merge.

A connectivity-preserving relaxation-based segmentation method, usually refers to as the active contour model, starts with some initial boundary shape represented in the form of spline curves, and iteratively modifies it by applying various shrink/expansion operations according to some energy function. Although the energy-minimizing model is not new, coupling it with the maintenance of an “elastic” contour model gives it an interesting new twist. As usual with such methods, getting trapped into a local minimum is a risk against which one must guard; this is not an easy task.
In this work, the fuzzy clustering method, thresholding method, region growing method and edge detection method are discussed and implemented.

### 1.2.1 Applications of segmentation

Some of the practical applications of image segmentation are:

1. **Medical Imaging**
   - Locate tumors and other pathologies
   - Measure tissue volumes
   - Computer guided surgery
   - Diagnosis
   - Treatment planning
   - Study of anatomical structures

2. Locate objects in satellite images (roads, forests, etc.)

3. Face recognition

4. Fingerprint recognition

5. Automatic traffic controlling systems

6. Machine vision
1.3 Segmentation Methods

Several general-purpose algorithmic techniques have been developed for image segmentation. Since there is no general solution to the image segmentation problem, these techniques often have to be combined with domain knowledge in order to effectively solve an image segmentation problem for a problem domain.

1.3.1 Fuzzy Clustering

What is Clustering?

Clustering can be considered as the most important unsupervised learning problem; so, as every other problem of this kind, it deals with findings as 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 [91] is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

We can show this with a simple graphical example:
In this case we easily identify four clusters into which the data can be divided; the similarity criterion is distance [22]. Two or more objects belong to the same cluster if they are “close” according to a given distance (in this case geometrical distance). This is called distance-based clustering [79]. Another kind of clustering is conceptual clustering: two or more objects belong to the same cluster if this one defines a concept common to all that objects. In other words, objects are grouped according to their fitness to descriptive concepts, not according to simple similarity measures.

1.3.2 The Goals of Clustering

So, 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 the “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 a way 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).

1.3.3 Possible Applications

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

- *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;
- *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;

**Requirements**

The main requirements that a clustering algorithm should satisfy are:

- scalability
- dealing with different types of attributes
- discovering clusters with arbitrary shape
- minimal requirements for domain knowledge to determine input parameters
- ability to deal with noise and outliers
- insensitivity to order of input records
- high dimensionality
- interpretability and usability

**Problems**

There are a number of problems with clustering. Among them:

- current clustering techniques do not address all the requirements adequately (and concurrently)
- dealing with large number of dimensions and large number of data items can be problematic because of time complexity
the effectiveness of the method depends on the definition of “distance” (for distance-based clustering)
if an *obvious* distance measure doesn’t exist we must “define” it, which is not always easy, especially in multi-dimensional spaces
the result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

### 1.3.4 The Algorithm

Fuzzy c-means (FCM) [101] is a method of clustering which allows one piece of data belongs to two or more clusters. This method [14] (developed by Dunn in 1973 and improved by Bezdek in 1981) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \| x_i - c_j \|^2, \quad 1 \leq m < \infty$$  \hspace{1cm} (1-1)

where $m$ is any real number greater than 1, $u_{ij}$ is the degree of membership of $x_i$ in the cluster $j$, $x_i$ is the $i$th of d-dimensional measured data, $c_j$ is the d-dimension center of the cluster, and $\|*\|$ is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership $u_{ij}$ and the cluster centers $c_j$ by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left( \frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m} \hspace{1cm} (1-2)$$

This iteration will stop when $\max_{\psi} \left\{ u^{(k+1)}_{\psi} - u^{(k)}_{\psi} \right\} < \varepsilon$, where $\varepsilon$ is a termination criterion between 0 and 1, whereas $k$ are the iteration steps. This procedure converges to a local minimum or a saddle point of $J_m$. 


The algorithm is composed of the following steps:

1. Initialize $U=[u_{ij}]$ matrix, $U^{(0)}$

2. At k-step: calculate the centers vectors $C^{(k)}=[c_j]$ with $U^{(k)}$

   $$C_j = \frac{\sum_{i=1}^{N} U_{ij}^m \cdot X_i}{\sum_{i=1}^{N} U_{ij}^m}$$

3. Update $U^{(k)}$, $U^{(k+1)}$

   $$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{\|x_i - c_k\|}{\|x_i - c_j\|} \right)^{\frac{2}{m-1}}}$$

4. If $\|U^{(k+1)} - U^{(k)}\| < \varepsilon$ then STOP; otherwise return to step 2.

As already told, data are bound to each cluster by means of a Membership Function, which represents the fuzzy behavior of this algorithm. To compute the Membership Function, it should be built an appropriate matrix named $U$ whose factors are numbers between zero and one, and represent the degree of membership between data and centers of clusters.

For a better understanding, it is considered that a set of data suppose to represented as distributed on an axis, so that it may be identified as the clusters and their proximity of the data concentrations. Figure-1 shows it has two data groups A and B and these data groups are associated with each other to a specific centroid, therefore this membership function looks like this:
In the FCM approach, instead, the same given datum does not belong exclusively to a well defined cluster, but it can be placed in a middle way. In this case, the membership function follows a smoother line to indicate that every datum may belong to several clusters with different values of the membership coefficient.
In the above figure, the datum shown as a red marked spot belongs more to the B cluster rather than the A cluster. The value 0.2 of ‘m’ indicates the degree of membership to A for such datum. Now, instead of using a graphical representation, we introduce a matrix U whose factors are the ones taken from the membership functions:

\[
U_{MC} = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
1 & 0 \\
\vdots & \vdots \\
0 & 1
\end{bmatrix}
\]

\[
U_{hC} = \begin{bmatrix}
0.8 & 0.2 \\
0.3 & 0.7 \\
0.6 & 0.4 \\
\vdots & \vdots \\
0.9 & 0.1
\end{bmatrix}
\]

(a) (b)

The number of rows and columns depend on how many data and clusters we are considering. More exactly we have \(C = 2\) columns (\(C = 2\) clusters) and \(N\) rows, where \(C\) is the total number of clusters and \(N\) is the total number of data. The generic element is so indicated: \(u_{ij}\).

1.3.5 Edge Detection

To use this approach, we examined a number of standard edge detectors. The goal was to find the “best” edge detector for our application. The best edge detector is the one that produces a closed contour of one pixel thick edges.

In gradient edge detection methods [82], the assumption is that edges are formed by pixels with a high gradient. A fast rate of change of intensity at some direction given by the angle of the gradient vector is observed at edge pixels. The magnitude of the gradient indicates the strength of the edge. All of the edge detectors in the gradient class approximate
the first derivative. A common disadvantage of these operators is their failure to provide reliable segmentation due to their gradient-based nature.

### 1.3.6 Region Growing

The main goal of segmentation is to partition an image into regions. Some segmentation methods such as “Thresholding” achieve this goal by looking for the boundaries between regions based on discontinuities in gray levels or color properties. Region based segmentation is a technique for determining the region directly. The basic formulation for Region based segmentation is:

\[
0
\]

(a) \( \bigcup_{i=1}^{n} R_i = R \)

(b) \(R_i\) is a connected region, \( i = 1, 2, ..., n \)

(c) \( R_i \cap R_j = \emptyset \) for all \( i = 1, 2, ..., n \).

(d) \( P(R_i) = TRUE \) for \( i = 1, 2, ..., n \).

(e) \( P(R_i \cup R_j) = FALSE \) for any adjacent region \( R_i \) and \( R_j \).

\( P(R_i) \) is a logical predicate defined over the points in set \( R_i \) and \( \emptyset \) is the null set.

(a) means that the segmentation must be complete; that is, every pixel must be in a region.

(b) requires that points in a region must be connected in some predefined sense.

(c) indicates that the regions must be disjoint.

(d) deals with the properties that must be satisfied by the pixels in a segmented region. For example, \( P(R_i) = True \) if all pixels in \( R_i \) have the same gray level.

(e) indicates that region \( R_i \) and \( R_j \) are different in the sense of predicate \( P \).
1.3.7 Basic concept of seed points

The first step in region growing is to select a set of seed points. Seed point selection is based on some user criterion (for example, pixels in a certain gray-level range, pixels evenly spaced on a grid, etc.). The initial region begins as the exact location of these seeds.

The regions are then grown from these seed points to adjacent points depending on a region membership criterion. The criterion could be, for example, pixel intensity, gray level texture, or color.

Since the regions are grown on the basis of the criterion, the image information itself is important. For example, if the criterion was a pixel intensity threshold value, knowledge of the histogram of the image would be of use, as one could use it to determine a suitable threshold value for the region membership criterion.

There is a very simple example here, that is, we use 4 connected neighborhood point to grow from the seed points. We can also choose 8 connected neighborhood point for our pixels have adjacent relationship. And the criteria we make here is the same pixel value. That is, we keep examining the adjacent pixels of seed points. If they have the same intensity value with the seed points, we classify them into the seed points. It is an iterated process until there is no change in two successive iterative stages. Of course, we can make other criteria, but the main goal is to classify the similarity of the image into regions.
Figure. 3 The histogram of Figure 4

Figure. 4 The Original Figure
Figure 5: Threshold below 100

Figure 6: Threshold: 225–255
Figure. 7 Threshold : 190~255

Figure. 8 Threshold : 155~255
Then there are several important issues about region growing:

1. **The suitable selection of seed points is important.**
   The selection of seed points is depending on the users. For example, in a gray level lightning image, we may want to segment the lightning part from the background. Then probably, we can examine the histogram and choose the seed points from the highest range of it.

2. **More information of the image is better.**
   Obviously, the connectivity or pixel adjacent information is helpful for us to determine the threshold and seed points.

3. **The value, “minimum area threshold”.**
   No region in region growing method result will be smaller than this threshold in the segmented image.

4. **The value, “Similarity threshold value“.**
   If the difference of pixel-value or the difference value of average gray level of a set of pixels less than “Similarity threshold value”, the regions will be considered as a same region. The criteria of similarities or so called homogeneity we choose are also important. It usually depends on the original image and the segmentation result we want. Here some criteria often used are gray level (average intensity or variance), color and texture or shape.
1.4 Medical Image Segmentation

Medical image analysis is one of the most critical studies in the field of medicine, since results gained by the analysis guide radiologists for diagnosis, treatment planning, and verification of administered treatment. Therefore, accuracy in analysis of medical images is at least as important as accuracy in data acquisition processes. In the field of biomedical image processing, Image segmentation lays the foundation stone for image processing and pattern recognition. Image segmentation can be termed as the process of grouping of the neighboring pixels. The general properties of the pixels are said to be coherent if the intensity value is same. The grouped regions may show the existence of objects or its parts. During the later stage, the regions may be verified or modified using a top down analysis of the image and recognition. The resultant of image segmentation obtained is a collection of segments that jointly cover the entire image, or a set of contours taken out from the image. Each and every pixel in a region is similar in terms of features or computed property like color, intensity, or texture. The nearby regions [56] are contrasting with respect to the same characteristics. In real time, image segmentation is applied in measuring tissue volumes, the study of anatomical structure, locating tumors and other pathologies.

Medical images require sequential application of several image post-processing techniques in order to use for quantification and analysis of intended features. Main objective of this thesis study is to build up an application framework, which enables analysis and quantification of several features in medical images with minimized input-dependency over results. Intended application targets to present a software environment, which enables sequential application of medical image processing routines and provides support for radiologists in diagnosis, treatment planning and treatment verification phases of
neurodegenerative diseases and brain tumors; thus, reducing the divergence in results of operations applied on medical images.

In scope of this thesis study, a comprehensive literature review is performed, and a study of medical image segmentation algorithms - including modules responsible for automation of separate processes and for several types of segments such as real tumor and real lesion area - is implemented. Performance of the fully-automated segmentation module is evaluated.

Medical image processing deals with the development of problem-specific approaches to the enhancement of raw medical image data for the purpose of selective visualization as well as further analysis. There are many topics in medical image processing, some emphasize general applicable theory and some focus on specific applications. A comprehensive overview on a broader range of topics in medical image processing appears [103] in . Here, the main focus is on image segmentation.

Image segmentation is defined as a partitioning of an image into regions that are meaningful for a specific task; it is a labeling problem. This may, for instance, involve the detection of a brain tumor from MR or CT images given below in Fig. 9.
Figure. 9 A 3D rendering of segmented skin surface (pink), brain tissue (brown), major blood vessels navy blue), and a tumor (green) from MRI volume. This allows surgeons to visualize the actual location and to plan and simulate specific procedures.

Segmentation is one of the first steps leading to image analysis and interpretation. The goal is easy to state, but difficult to achieve accurately. Image segmentation approaches can be classified according to both the features and the types of techniques used. Features include pixel intensities, edge information, and texture, etc. Techniques based on these features can be broadly classified into structural and statistical methods. Structural methods are based on the spatial properties of the image, such as edges and regions. Various edge detection algorithms have been applied to extract boundaries between different brain tissues [55][64][68]. However such algorithms are sensitive to artifacts and noise. Region growing [15,84] is another popular structural technique. In this approach, one begins by dividing an image into small regions, which can be considered as `seeds". Then, all boundaries between adjacent regions are examined. Strong boundaries (in terms of certain specific properties) are kept, while weak boundaries are rejected and the adjacent regions merged. The process is carried out iteratively until no boundaries are weak enough to be rejected. However, the
performance of the method depends on seed selection and whether the regions are well
defined, and therefore is also not considered robust. Starting from a totally different
viewpoint, statistical methods label pixels according to probability values, which are
determined based on the intensity distribution of the image.

Gray-level thresholding is the simplest, yet often effective, segmentation method. In
this approach, structures in the image are assigned a label by comparing their gray level value
to one or more intensity thresholds. A single threshold serves to segment the image into only
two regions, a background and a foreground. Sometimes the task of selecting a threshold is
quite easy, when there is a clear difference between the gray-levels of the objects we wish to
segment.

However, things are not normally so simple. In homogeneity in the imaging
equipment and the partial-volume effect (multiple tissue class occupation within a voxel) give
rise to a smoothly varying, non-linear gain field. While the human visual system easily
compensates for this field, the gain can perturb the histogram distributions, causing
significant overlaps between intensity peaks and thus leading to substantial misclassification
in traditional intensity based classification methods.