Image Retrieval Using Image Segmentation Techniques

Ganga Sumanth A, Akash M B, Mrs. Vanitha
Bapuji Institute of Engineering and Technology
gangasumanth@gmail.com

Abstract— Image Segmentation is a technique wherein the subject digital image is partitioned to better understand, manipulate and utilize its constituents. This segmentation results in a set of pixels or superpixels which are related to each other by some computed property, such as intensity, colour or texture. Image Segmentation can be carried out in several methods including: Clustering, Thresholding, Edge Detection, Region based segmentation, etc. The segments thus formed correspond to some characteristics of image like colour, texture etc. which can then be used to retrieve similar images using confidence tests.

Keywords— *Segmentation, Threshold, Clustering, Edge Detection*

I. INTRODUCTION

Under Digital Image Processing, Image Segmentation is used to partition the image into essential regions with respect to pixel characteristics. Any given image is represented as a matrix where the location of each pixel is specified along with its intensity on a scale ranging from 0 to 255. In case of colour images, RGB intensities are also specified. [1] Image segmentation plays an important role in several fields of application, Medical Imaging being one of them. Medical images play vital role in assisting health care professionals with diagnosis. [2]. Several techniques have been developed for Image Segmentation by notable institutions such as Bell Labs, University of Maryland to name a few in the 1960s. Concept of image segmentation is applicable to medical imaging, video phone, photo enhancement, satellite imagery etc. Real world Image Processing involves contextual understanding of image components and segmentation is one of the widely used techniques to arrive at the desired result. A wide variety of algorithms are used to segment a given image.

The concept of fuzzy logic, pattern recognition and machine learning has been combined with artificial intelligence in digital image processing. Image techniques can be clubbed together in a general framework called Image Engineering. Image Engineering can be defined as that which contains three layers as:

a) Image understanding

b) Image Analysis

c) Image Processing

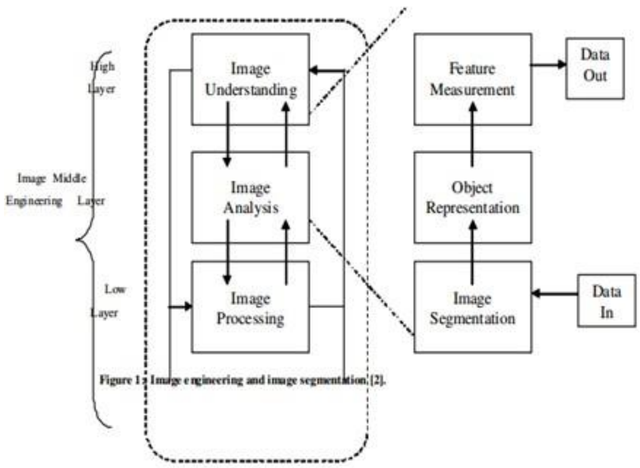

Figure 1.1: Layers of Image Engineering [2]

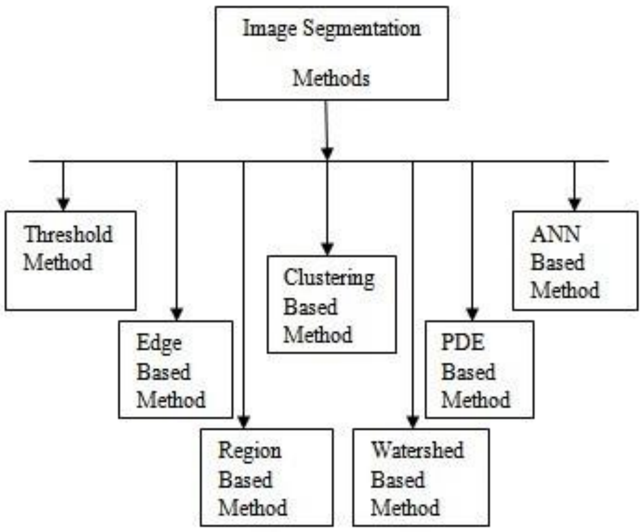

Figure 1.2: Image Segmentation Techniques [2]

II. DIFFERENT TECHNIQUES OF IMAGE SEGMENTATION

A. Segmentation Based on Edge Detection

An Edge can be defined as the set of points at which image brightness changes sharply and are typically organized into a set of curved line segments. It plays very important role in image analysis and pattern recognition as it describes the physical extent of objects. Some Edge detection methods are as follows:

1) Roberts Edge Detection: Proposed by Lawrence Roberts in 1963, the Roberts edge operator is used in image processing for edge detection. It is a simple, quick to compute, 2-D spatial gradient measurement on an image. It thus highlights regions of high spatial gradient which often correspond to edges. In its most common usage, the input to the operator is a grayscale image, as is the output. Pixel values at each point in the output represent the estimate absolute magnitude of the spatial gradient at that point.

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Figure 2.2.1: Roberts cross convolution masks [4]

2) Sobel Edge Detection: Sobel edge detector named after Irwin Sobel and sometimes called the Sobel filter consists of a pair of masks, one vertical and other horizontal. These masks are generally 3*3 matrices. Standard Sobel operators, for a 3×3 neighbourhood, each simple central gradient estimate is vector sum of a pair of orthogonal vectors. [1] Each orthogonal vector is a directional derivative estimate multiplied by a unit vector specifying the derivative's direction. The vector sum of these simple gradient estimates amounts to a vector sum of the 8 directional derivative vectors. Thus for a point on Cartesian grid and its eight neighbours having density values as shown: [5]

a	b	c
d	e	f
g	h	i

Figure 2.2.2: Sobel's Density values [5]

3) Prewitt Edge Detection: Prewitt Edge Detector, also referred to as Discrete Differentiation operator, is used to calculate the gradient of the image intensity function. The Prewitt Edge filter is use to detect edges based applying a horizontal and vertical filter in sequence. Both filters are applied to the image and summed to form the final result. The two filters are basic convolution filters of the form: [6]

1	1	1
0	0	0
-1	-1	-1

-1	0	1
-1	0	1
-1	0	1

Figure 2.2.3: Horizontal Filter Vertical Filter [6]

B. Threshold Methods

Threshold method is used to discriminate foreground from background. In this method, a grey scale image is converted into binary image. The binary image contains data regarding location and shape of the objects. Conversion to binary image reduces the complexity of data. Threshold methods are following:

1) Global Thresholding: In the global thresholding, the intensity value of the input image should have two peak values which correspond to the signals from background and objects. Thus segmentation is carried out using the degree of intensity separation between two peaks in the image. Global thresholding, using an appropriate threshold T:

$$g(x,y) = \begin{cases} 1, & \text{if } f(x, y) > T \\ 0, & \text{if } f(x, y) \le T \end{cases} \quad [8]$$

2) Variable Thresholding: In variable thresholding, we separate out the foreground image objects from the background based on the difference in pixel intensities of each region.

a) Variable thresholding, for a changing value of T over the image.

b) Local or regional thresholding, when T depends on a neighborhood of (x, y).

c) Adaptive thresholding, for when T is a function of (x, y). [8]

3) Multiple Thresholding: The image is segmented into several distinct regions by defining more than one threshold for the given image and thus diving the image into certain brightness regions. Sets of these segments pertain to the background and several objects. Multiple thresholding:

$$g(x,y) = \begin{cases} a, & \text{if } f(x, y) > T2 \\ b, & \text{if } T1 < f(x, y) \le T2 \\ c, & \text{if } (x, y) \le T \end{cases}$$

C. Region Based Segmentation

In Region Based segmentation each pixel is classified into relevant classes. Partitioning is done based on grey values of the image pixels. Region based segmentation is done using the following methods:

1) Region Growing Methods: Region growing is a technique that groups pixels or sub regions into larger regions based on predefined criteria. The pixel aggregation starts with a set of seed points in a way that the corresponding regions grow by appending to each of the seed points of neighbouring pixels that have similar properties like grey scale, colour, texture, shape etc. [9]

2) Region Splitting and Merging: In case of region splitting, the whole image is taken as a single region and then this region is being break into a set of disjoint regions which are coherent with themselves. Region merging opposes Region Splitting. A merging technique is used after each split and compares adjacent regions and merges them. It starts with small regions and merges the regions which have similar characteristics like grayscale, variance etc.

D. Clustering Based Image Segmentation

Clustering based image segmentation is used to segment the images of grey level. Grey level methods can be directly apply and easily extendable to higher dimensional data. Clustering is also applicable in colour and multispectral images. There are two main methods in clustering:

1) K-Means: The k-means methods of clustering are obtained based on the principle of minimization of the sum of squared distances from all points in each cluster domain to the cluster centre. This sum is also known as the within cluster as opposed to the between cluster distance which is the sum of distance between different cluster centre and the global mean of the entire data set. [10]

2) Fuzzy K-Means: The Fuzzy K-means method is a two stage process involving a "coarse" segmentation followed by a "fine" segmentation. The "coarse" segmentation involves smoothing the histogram of each of the colour components and using the first and second derivatives of the smoothed histograms to find the valleys which will then be used as thresholds. A safe tolerance area surrounding the thresholds is then determined, and every pixel not falling into any safe area is assigned to a cluster based on its red, green and blue values and cluster centres are calculated. The "fine" segmentation involves assigning each pixel which belongs to a safe area to its closest cluster by calculating fuzzy membership functions. [10]

III. IMAGE RETRIEVAL

Image Retrieval is the retrieval of images based on visual features such as colour, texture and shape. Image texture [7] is a widely used and primitive feature of an image. Texture feature plays important role to separate regions. The most commonly used texture feature is Tamura representation. This is widely used because it is based on human texture representation. The ability to retrieve images on the basis of texture similarity may not seem very useful, but the ability to match on texture similarity can often be used for the difference between areas of images with similar colour (such as sky, and sea, or leaves and grass). A variety of techniques has been used for measuring texture similarity; the best established rely on comparing values of what are known as second order statistics calculated from query and stored images.
In CBIR, each image that is stored in the database has its features extracted and compared to the features of the query image. It involves two steps:

Feature Extraction: The first step in the process is extracting image features to a distinguishable extent.

Matching: The second step involves matching these features to yield a result that is visually similar.

A. CBIR SYSTEM

Content-based image retrieval, also known as query by image content and content-based visual information retrieval is the application of computer vision techniques to the image retrieval problem, that is, the problem of searching for digital images in large databases. [15] This process involves retrieval of desired images from a large collection on the basis of features (such as colour and texture) that can be automatically extracted from the images themselves. The features used for retrieval can be either primitive or semantic, but the extraction process must be predominantly automatic.

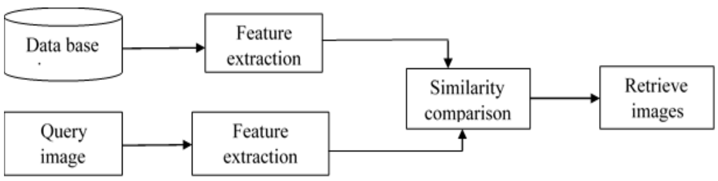

Fig3.1: Block Diagram of CBIR System

The block diagram as shown in Figure identifies the CBIR use cases:

Query by Image: The query image of JPEG format instead of query by text etc.

Database Images: It contains a large list of images with which the query image is compared to extract related images.

Feature Extraction of query image: The features of the query image are extracted and stored in the temporal storage.

Feature Extraction of database images: The features of the database images are extracted and stored in the temporal storage

Similarity Measurements: The task of comparing the query image features with the database images individually and the matched result is obtained.

Retrieve Images: The images related to query image in serial form.

1) Creating database

IR database contains 450 images collected. The database images of size 85×128 or 128×85 in dimension, which means each has a total pixel size of 10880 if we place them in a single array row.

The local database has some categories - butterfly images, flower images, waterfalls images, food images, satellite images. The images are in jpg format.

2) Texture Feature (Edge Histogram Descriptor)

The edge histogram descriptor [2] captures the spatial distribution of edges. The distribution of edges is a good texture signature that is useful for image to image matching even when the underlying texture is not homogeneous. The computation of this descriptor is fairly straightforward. A given image is first sub-divided 4×4into sub-images, and local edge histograms for each of these sub-images is computed.

To compute the edge histograms, each of the 16 sub-images is further subdivided into image blocks. The size of these image blocks scale with the image size and is assumed to be a power of 2. The number of image blocks per sub-image is kept constant, independent of the original image dimensions, by scaling their size appropriately. A simple edge detector is then applied to each of the macro-block, treating the macro-block as a 2×2pixel image. The pixel intensities for the2×2 partitions of the image block are computed by averaging the intensity values of the corresponding pixels. The edge-detector operators include four directional selective detectors and one isotropic operator (Fig.). Those image blocks whose edge strengths exceed a certain minimum threshold are used in computing the histogram.

$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix} \quad \begin{bmatrix} 0 & \sqrt{2} \\ -\sqrt{2} & 0 \end{bmatrix} \quad \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

Fig3.2: filters for edge detection

Five edge strengths for the image block (i,j) as follows:

Ver_egde_stg (i , j)

$$= \sum_{i=0}^{3} |Ak(i,j) \times ver\_edge\_filter(k)| \quad (1)$$

hor_egde_stg (i , j)

$$= \sum_{i=0}^{3} |Ak(i,j) \times hor\_edge\_filter(k)| \quad (2)$$

dia45_egde_stg(i , j)

$$= \sum_{i=0}^{3} |Ak(i,j) \times dia45\_edge\_filter(k)| \quad (3)$$

dia135_egde_stg (i , j)

$$= \sum_{i=0}^{3} |Ak(i,j) \times dia135\_edge\_filter(k)| \quad (4)$$

nond_egde_stg (i , j)

$$= \sum_{i=0}^{3} |Ak(i,j) \times nond\_edge\_filter(k)| \quad (5)$$

If the maximum value among five edge strengths obtained from equations (1) to (5) is greater than a threshold (Th_{edge}) as in equation (6), then the image-block is considered to have the corresponding edge in it.

max{ver_edge_stg(i,j), hor_edge_stg(i,j), dia45_edge_stg(i,j), dia135_edge_stg(i,j), nond_edge_stg (i, j)} > Th_{edge}

(6)

3) Similarity Matching:

Note that there are a total of 80 bins, 3 bits/bin, in the edge histogram. One can use the 3-bit number as an integer value directly and compute the distance between two edge histograms. An interesting variation is to compute an extended histogram from these 80 bins. The extended histogram is obtained by grouping the image blocks (and the corresponding bins). The extended bins are referred to as the global and semi-global histograms. The global histogram is obtained by combining all the 16 image blocks. The semi-global histograms are computed by pooling the image blocks/bins by rows (four rows), columns (four columns) and in groups of (five groups). This results in five bins for the global histogram and for the semi global histograms from the 80 local histogram bins. The total number of bins is 150.The edge histogram descriptor is found to be quite effective for representing natural images with the primary application being image-to-image matching. The performance can be further enhanced by using this descriptor in conjunction with other image features, such as colour.

B. Performance analysis of CBIR using Color and Texture features (butterfly image)

 Query image			
Retrieved Similar Images			
 ✓ butterfly1	 ✓ butterfly64	 ✓ butterfly54	 ✓ butterfly82
 ✓ butterfly39	 ✓ butterfly15	 ✓ butterfly91	 ✓ butterfly88
 • flower(43)	 •	 ✓ butterfly77	 • flower1 (83)
 ✓ butterfly62	 ✓ butterfly9	 ✓ butterfly99	 • flower1
 ✓ butterfly75	 • flower1 (35)		

Table 3.5: Performance analysis of CBIR using Color Features and Texture features

- Indicates Retrieve Non Relevant Images=7
- ✓ Indicates Retrieve Relevant Images =13

Performance Results= (Retrieve Relevant Images/Total images) = (13/20) = 0.65 =65

C. Performance analysis of CBIR using Color and Texture features (flower images).

 Query image			
Retrieved similar images			
 ✓ flower1 (51)	 ✓ flower1 (8)	 ✓ flower1 (51)	 ✓ flower1 (30)
 ✓ flower1 (48)	 ✓ flower1 (33)	 ✓ flower1 (73)	 ✓ flower1 (51)
 ✓ flower1 (11)	 ✓ flower1 (26)	 ✓ flower1 (14)	 ✓ flower1 (67)
 ✓ flower1 (79)	 ✓ flower1 (13)	 ✓ flower1 (2)	 ✓ flower1 (80)
 waterfalls	 ✓ flower1 (51)	 • food image	 ✓ flower1 (51)

Table 3.6: Performance analysis of CBIR using color and texture features

- ✓ Indicates Retrieve Relevant Images
- • Indicates Retrieve Non Relevant Images

Performance Results= (Retrieve Relevant Images/ Total images) or Performance Results = (18/20) = 0.9 =90

III. CONCLUSION

In this paper, we have discussed image segmentation and the various approaches available to go about it. These techniques are applicable in different fields like medical imaging, object recognition, pattern recognition etc. Image Segmentation is an often used method when it comes to computer vision. As an extension, object recognition by the use of suitable classifiers allows for libraries of objects classes to be built and raw inputs can be matched against pre-defined classes. A pertinent set of confidence testing techniques can be used to retrieve similar objects when an image input is provided to a system. CBIR is a widely used a system for building image recognition models and querying large databases. A sample system and its test results were discussed.

IV. REFERENCES

[1] W. X. Kang, Q. Q. Yang, R. R. Liang, "The Comparative Research on Image Segmentation Algorithms", IEEE Conference on ETCS, pp. 703-707, 2009.

[2] Zhang, Y. J, An Overview of Image and Video Segmentation in the last 40 years, Proceedings of the 6th International Symposium on Signal Processing and Its Applications, pp. 144-151, 2001

[3]] Engr. V. C. Chijindu, H. C. Inyiama PhD, Engr. G. Uzedhe, "Medical Image Segmentation Methodologies – A Classified Overview", IEEE African Journal of Computing & ICT September, 2012 - ISSN 2006-1781

[4] http://www.robotics.technion.ac.il/courses/advanced_laboratory/lab7/arl_7_read.pdf

[5] SOBEL, I., An Isotropic 3×3 Gradient Operator, Machine Vision for Three – Dimensional Scenes, Freeman, H., Academic Pres, NY, 376-379, 1990.

[6] http://www.roborealm.com/help/Prewitt.php

[7] http://www.ams.sunysb.edu/~lindquis/3dma/man_3dma/manual/node19.html

[8] Stefano Ferrari, universita degli studi di Milano, "Image Segmentation", a.a 2011/12

[9] http://www.di.ubi.pt/~agomes/cvm/teoricas/07-regionsegmentation.pdf

[10] Siddheswar Ray, rose H.Turi and Peter E. Tischer, Clustering-based colour Image Segmentation: An Evolution Study.

[11] T. Shraddha, K.Krishna , B.K Singh and R.P Singh, "Image Segmentation: A Review" International Journal of Computer Science and Management Research vol.1 issue. 4 November 2012.

[12] S. Saleh, N.V. Kalyankar and S.Khamitkar, "Image Segmentation by using edge detection", (IJCSE) International journal on Computer Science and Engineering vol. 01, No. 03, 2010

[13] Inderpal and K. Dinesh, "A review on Different ImageSegmentation Technique", IJAR, Vol.4, April, 2014.

[14] N.Senthilkumaran and R.Rajesh, "Edge Detection Techniques for Image Segmentation - A Survey", Proceedings of the International Conference on Managing Next Generation Software Applications (MNGSA-08), 2008, pp.749-760

[15] https://en.wikipedia.org/wiki/Content-based_image_retrieval