IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

An Advanced Machine Learning Approach Using Supervised Learning Algorithm-Decision Tree for Data Mining and Segmentation of Images

[1] A. Gowthami, [2] M. Sri Venkat Rami Reddy, [3] Y. D. Solmon Raju
[1]Assistant Professor, [2] Associate Professor of ECE Dept.,
Holy Mary Institute of Technology and Science
(An UGC-Autonomous Institution),

[3]Associate Professor & HoD of ECE Dept., Holy Mary Institute of Technology and Science,
(An UGC-Autonomous Institution),
Bogaram(V), Keesara(M), Malkajgiri-Medchal (Dt) Telangana, India.

ABSTRACT

In this paper, a general mining approach based on decision trees for segmenting image data is proposed. Pixel-wise image features are extracted and transformed into a database-like table that allows existing data mining algorithms to dig out useful information. Each tuple in the table has a feature descriptor consisting of a set of feature values for a given pixel along with its label. With the feature label, we can employ the decision tree to (1) discover relationship between the attributes of pixels and their target labels, (2) build a model for image processing by using the training data set. Both experiments and theoretical analysis are performed in our research. The results show that the proposed model is very efficient and effective for image mining and image segmentation. It can also be used to develop new image processing algorithms, refine existing algorithms, or act as an effective filter. *The MIST algorithm generates randomly different addition chains for performing a particular exponentiation. This means that power attacks which require averaging over a number of exponentiation power traces becomes impossible. Moreover, attacks which are based on recognizing repeated use of the same pre-computed multipliers during an individual exponentiation are also infeasible. The algorithm is particularly well suited to cryptographic functions which depend on exponentiation and which are implemented in embedded systems such as smart cards. It is more efficient than the normal square-and-multiply algorithm and uses less memory than 4-ary exponentiation.*

Keywords: Data mining, decision tree, association rule, Image Indexing, classification, clustering, image segmentation, pixel. MIST

1. INTRODUCTION

The term-knowledge discovery in image databases as image mining. The main goal of data mining is to discover previously unknown knowledge from a huge amount of historical data that can help us initiate proper actions. "Knowledge mining from data" is another name for the term "data mining", which is more appropriate but somewhat too long. Many people treat data mining as a synonym for another popular term, Knowledge Discovery in Databases (KDD). Although plenty of knowledge can be hidden in image data, very few literatures discuss KDD in this type of data. Issues of image mining have classified as four classes. They were associations, classification, sequential patterns, and time series patterns. However, only the prototype of finding associations has been proposed.

Image segmentation is an important procedure to crop useful information from images. Knowledge can be more easily recognized when presented in the form of images. For example, geophysical and environmental data from satellite photos, Web pages containing images, medical imaging including Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Ultrasound Imaging (UI), are sources of useful information used in our daily life. They are conformed to various standard image protocols. Although many image segmentation algorithms have been proposed, only few of them can be applied to image mining.

Mining non-standardized data and multimedia data is the trend in the future. However, most existing data mining techniques have been designed for mining numerical data and are thus not well suited for image mining. In this paper, we solve this problem by presenting a new approach based on decision trees for both of image data mining and segmentation. Decision tree induction is a well-known methodology used widely on various kinds of domain, such as artificial intelligence, machine learning, data mining, and pattern recognition. A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. An advantage of decision trees over other methodologies, such as neural network, is that it could provide understandable English-like rules or logic statements, for example, "If a pixel's gray level ranges from 180 to 240 and its local variation is greater than 80 and its slope variation is greater than 0.5, then it is the pixel we wanted." This basic idea of simple and easily understandable is also the main principle of our approach.

In an image mining method that works at a higher generality level for mining image associations is proposed. In contrast to that, our proposed model works on a relative low generality level for image pixel classification. Pixel-wise image classification is an essential part of many image segmentation methods, for example, determining pixels of an edge (corner) in edge (corner) detection methods pixels of a particular object in objects segmentation based methods], pixels of abnormal tissue of medical image processing, and pixel classes in threshold, etc.

1.1 An Approximation Algorithm for Maximum Internal Spanning Tree:

Given a graph G, the Maximum Internal Spanning Tree (MIST for short) for computing a spanning tree T of G such that the number of internal vertices in T is maximized. MIST has possible applications in the design of cost-efficient communication networks and water supply networks and hence has been extensively studied in the literature. MIST is NP-hard and hence a number of polynomial-time approximation algorithms have been designed for MIST in the literature. The previously best polynomial-time approximation algorithm for MIST achieves a ratio of 3/4.

The proposed model can be used to mine hidden relationships between an image's pixel and its class label, and determine the interrelated features. Besides, the created model can be applied to perform pixel-wise segmentation on input images. The rest of the paper is organized as follows. Section 2 gives a brief overview of our approach.

The detailed process and the experiments are presented in Section 3. The applications of the mining result are in Section 4. In Section 5, we give theoretical analysis and discussions of the proposed model. Lastly in Section 6, we conclude our paper and discuss the future work.

2. OVERVIEW

The general processing flow of the proposed model is depicted in Fig.1.The data we used for input is formatted as a set of *raw* and *label* image pair. Each pixel's value of the *label* image is a class label with respect to the pixel in the raw image at the same position. The label of a pixel could indicate the type of a pixel, its frequency, etc. Section 3.1 has more detailed description of the input data. In addition, we propose three kinds of input data source: 1. defined by users, 2. from other algorithms, 3. hybrid, which will be explored further in Section 5.1.

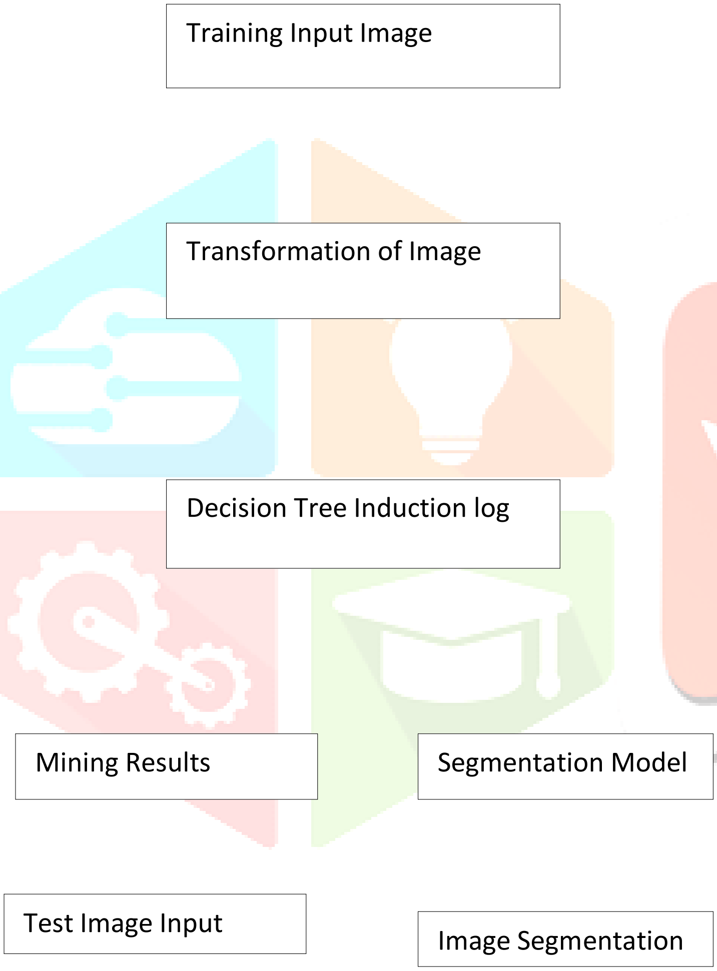


Fig-1: The Proposed Image Segmentation Model

Once a set of interested *raw* and *label* image pair has been obtained, they are transformed and stored in a database-like table. Each row of the transformed table represents a given pixel, and each column of such table represents an encoded feature associated with that pixel. After obtaining such a database-like table from the images we are interested in, we can then begin to dig on it. In this paper, we have chosen the decision tree methodology for this purpose. Based on the decision tree technology, our proposed model is able to generalize rules between the label of pixels and their features. This mining process and experiments will be described in Section 3.2. The results of such process could not only help us understand more about image properties as to the real world instance, but also to segment new cases of the same domain. The mining results and their applications are discussed in Section 4, and the segmentation model will be presented in Section 5.

2.1 Image Transformation and Feature Extraction:

As mentioned, the input data of the proposed model is formatted as a set of equal sized *raw* and *label* image pairs. The transformation of the input image dataset into a database like table and subsuming of the related features is described in this subsection. For the sake of clarity, various terms used for this process are defined below. In addition, we propose three kinds of input data sources, which will be discussed further in section 4.1.

Definition #1

The **raw image** is a d-dimensional light-intensity function, denoted by $R(c1, c2, \ldots, cd)$, where the amplitude (or value) of R at spatial coordinates $(c1, c2, \ldots, cd)$ gives the intensity of the raw image at that point (or pixel).

Definition #2

The **label image** is a d-dimensional light-intensity function, denoted by $L(c1, c2, , cd)$, where the value of L at spatial coordinates $(c1, c2, \ldots, cd)$ gives the class identifier of the pixel at same spatial coordinates of its corresponding raw image.

Definition #3

The **database-like table** $X = \{x1, x2, \ldots, xt\}$ is a set of records, where each record $xr \in \Re k$ is a vector with elements $<a1, a2, \ldots, ak>$ being the value of attributes / features of X

7	9	9	9	7
5	7	9	7	5
0	7	9	7	0
5	7	9	7	5
7	9	9	9	7

(a). Raw Image

1	0	0	0	1
1	1	0	1	1
0	1	0	1	0
1	1	0	1	1
1	0	0	0	1

(b). Label Image

Fig-2: An example of the input image data set.

In this work, only $d = 2$ is considered, *i. e.*, images with conditionality of 2. An example of the input image data set is shown in Fig. 2. Each pixel value of the *raw* image represents the gray level of a pixel. Each pixel value of the *label* image represents the class label of the pixel. Both pixel values are in the same position. In this example, the *raw* image contains the capital English letter “I” with certain degree of blur. Thus, the inside pixels of the letter are darker and the outside pixels are brighter. If a pixel in the *label* image has the value “1”, the pixel in the same position of the *raw* image is a pixel of *outside contour*. It is assumed to be a *pixel of interest* (POI) . In practice, the pixel value of the *label* image is not limited to the binary form but could take any kind of form. In addition, we can have as many *raw* and *label* image pairs at the same time as required for the input. In order to mine useful information from a set of *raw* and *label* images, we propose a methodology to transform them into a database-like table and allow any data mining algorithms to work on top of the table. This process is simple and straightforward as shown in Fig.3. Fig.4 shows a part of the results of this transformation process according to the data in Fig.2. Each row of such result table stands for a pixel. Hence its carnality(number of rows) equals the number of total pixels in the *raw* image. In addition, each column of such table represents a feature associated with the given pixels. In Fig. 4. F*eature*1 represents the gray level and *feature*2 the local variation [16]. In order to simplify this demonstration, the local variation in this case is replaced with the average difference of a pixel to its 4-neighbors. Other pixel-wised features [17, 18] such as entropy, contrast, mean, *etc*. can also be encoded into the table as long as they might have affection on the collected data set. Various encoding strategies such as normalization (*e.g.*, adjusting the value ranging from 0 to 1) or generalization (*e. g.*, transforming the value to high, medium, or low) can be applied when generating the desired features. Moreover, the *label* image was included as a column in that table. With the presence of the label feature, hidden relationships between these two kinds of images can be mined.

```
procedure img2tab(image: raw, label);
begin
    set feature_generated_functions[1..n];
    set label_generated_function;
    initiate table, pixel;
    while pixel exists do
        {pixel scanning process}
        insert into table value:=
            feature_generated1(raw, pixel),
            …,
            feature_generatedn(raw, pixel),
            label_generated(label, pixel);

        continue to scan on the next pixel;
    end while
    return table;
end
```

Fig-3: Pseudo code of the image transformation algorithm.

	Feature 1	Feature2	…	Feature n	Label
Pixel 1	7	2	…	Value 1,n	1
Pixel2	9	1.25	…	…	0
Pixel 3	9	0	…	…	0
…	..	…	…	…	…
Pixel 4	7	2	…	Value 25,n	1

Fig-4: Result table of image transformation according to the input in Fig.2.

2.2 Data Reduction

Because of the image characteristics, pixels from a neighboring area will generate similar feature vectors in the transformation process. Under some circumstances, it will cause remarkable redundant information in the result table; for example, an image with a large portion of background. Here we present some basic types of redundancy and show how they can be eliminated while converting the input image set.

Definition #1

The **feature scope** of a pixel *M* with spatial coordinates (*c*1, *c*2) is an $n \times n$ pixel area with center at *M*, from which all the desired features of *M* can be generated. Usually *n* is an odd number, and the sub-image within the feature scope, *i. e.*, pixels within spatial coordinates2, (*c2*±n-1/2), is called the **root space** of the pixel *M*, denoted as {*RSM*}.

Definition #2

Two root spaces {*RSN*}, {*RSO*} are **rotation reachable** if {*RSN*} = {*RSO*}*R*,
where {.}*R* stands for a root space after rotating the angle once by 90°, 180°, or 270°.

Definition #3

Two root spaces $\{RSN\}$, $\{RSO\}$ are **mirror reachable** if $\{RSN\} = \{RSO\}F$,
where $\{.\}F$ stands for a root space after flipping horizontally or vertically.

Given two pixels P and Q at different *spatial coordinates* of an image I, they are said to be:

1. Equivalent Redundant, if $\{RSP\}$ is equal to $\{RSQ\}$,

2. Rotation Redundant, if $\{RSP\}$ and $\{RSQ\}$ are rotation reachable,

3. Mirror Redundant, if $\{RSP\}$ and $\{RSQ\}$ are mirror reachable,

4. Conflict Redundant, if $\{RSP\}$ and $\{RSQ\}$ satisfy any one of the first three conditions, but the label information of pixels P and Q is not equal to each other.

```
function RR (image: raw, label; pixel: C);
begin
apply quantization on {RSC} if necessary;
if {RSC} can be matched in Ξ do {redundant pixel} discard {RSC} for further record generation;
if the label information of the two matched entries are not equal do {conflict redundant pixel}
update the corresponding information in Ξ; retrieve or update previously generated record if necessary;
else
{non-redundant pixel} record all characterized redundancies of {RSC} and the corresponding label information in Ξ;
end
```

Fig-5: Pseudo-code of the Redundancy Reduction Algorithm

Users could characterize other types of redundancy according to the image problem they wish to solve. In order to pinch more redundancies, quantization techniques can be applied on the root space. The pseudo code regarding the function of redundancy reduction is shown in Fig. 5. This function can be added to the pixel scanning process of the image transformation algorithm in Fig. 3. Fig. 6 shows the results of this reduction process according to the images in Fig. 2. The number of pixels for transformation after reduction has reduced from 25 to 9.

2.3 Mining Results and their Applications

After having obtained such a database-like table in accordance to the desired input image dataset, mining algorithms can then be used on it. In this study, we have chosen the decision tree for this purpose. An advantage of the decision tree over other methodologies, such as neural networks, is that it can provide understandable English-like rules or logic statements. For instance, *if the gray level of a given pixel ranges between 180 and 240 and its entropy is greater than 0.5*, *then it is a pixel of interest*, *POI*. This basic idea of simplicity and easy understandability is also the main principle of our approach. The results of such a mining process may help us to better understand the image properties and relate to real world instances. The results can also be used to process new images of the same domain. Basically, the result of the proposed model is a decision-tree classifier. Fig.7 depicts a classifier derived from the data shown in Fig. 4 by using CART . A result classifier can be further straightforwardly translated into a set of human readable *if-then* rules. For instance, from the three leaf nodes in Fig. 7,
We can obtain the following three rules: –
-If the gray level of a given pixel is less than 8 and its local variation is less than 5, then it is a pixel of outside contour.
- If the gray level of a given pixel is less than 8 and its local variation is greater than or equal to 5, then it is not a pixel of outside contour.
-If the gray level of a given pixel is greater than or equal to 8, then it is not a pixel of outside contour.

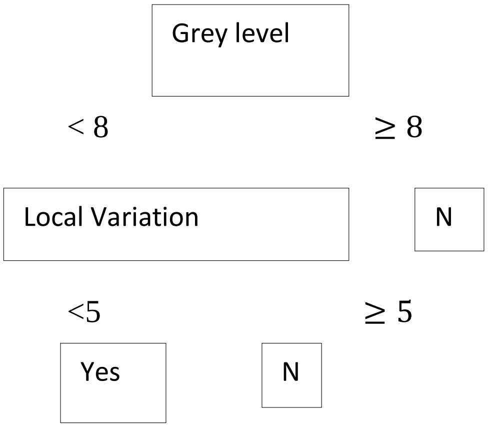
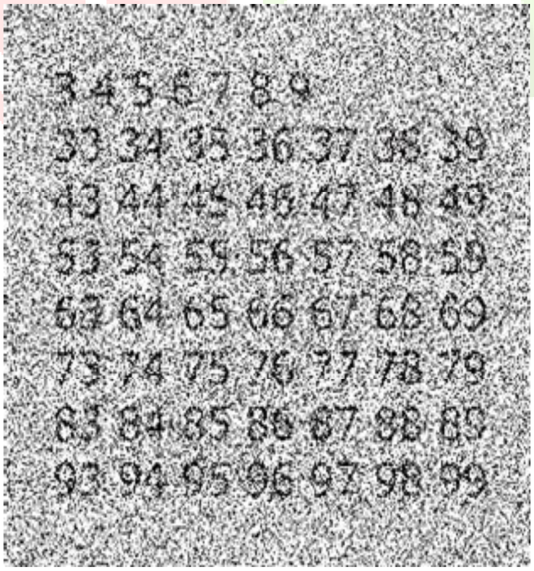
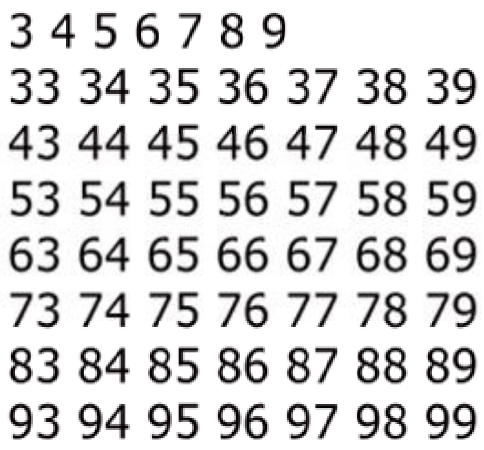


Fig-6: A decision tree for the concept *is outside contour*

Derived from Fig-6, indicating whether or not a pixel is a pixel of outside contour. These rules can provide useful information about the training image. Besides, in order to obtain a higher level of appearance and meet the different information granularity requirements, the rules can be post-processed by rule induction algorithms. More prominently, they can be used to process new images from the same domain. The practical image processing capabilities include image restoration, image enhancement, image segmentation, *etc*. Both experimental and theoretical analyses were performed in this study to examine the proposed model. The built classifier can also be used to select important features. Features used at higher tree levels for the splitting criteria show a higher significant influence on the pixel class. The selected features can reflect the characteristics of the *label* image and help design or refine other image processing algorithms.



(a) Raw Image



(b) Label Image

Similarly, in the English alphabet training data set, the distorted and the original images synthesized by the letters "*F*" to "*Z*" and their two combinations (*i.e*., *FF*, *FG*, …, *YZ*, *ZZ*) were used for the *raw* and *label* images, respectively. The other letters (*i.e*., *A*, *B*, …, *E*) were used to synthesize the testing image data set.

For image transformation, a feature scope of size 5 × 5 was used and the selected features included gray level, local variation, mean, local minimum, local maximum, and entropy. The label of a given pixel in the experiments of image restoration with enhancement was set to its gray level in the *label* image. We did not apply any encoding strategies mentioned in section 2.1 on the features to simplify the demonstrations. However, in practice, we can use any encoding strategy if required. In the image segmentation experiments, the label feature was transformed to 0 or 1 according to the threshold *label* image. In this way the segmentation nature was imitated to distinguish between "background" or "object". After we have settled the transformation details, a database-like table can be derived. By applying a classification algorithm on the database-like table, a classifier for label prediction can be obtained. Under the same way, testing images can be transformed into a database-like table to predict the label attributes. These predicted labels can moreover be visualized in a natural form of the input data, *i.e*., image. As we are proposing a general image mining and image processing framework and any existing decision tree algorithms can be used to do the job, we show only the testing result to simplify the demonstration. For the other results regarding the constructed classifier or the corresponding rules, if interested, examples can be found in our previous work

3. IMAGE MININGAND SEGMENTATION TECHNIQUE

Besides investigating suitable frameworks for image mining, early image miners have attempted to use existing techniques to mine for image information. The techniques frequently used include object recognition, image indexing and retrieval, image classification and clustering, association rules mining, and neural network.

3.1 Image segmentation Approach:

Image Segmentation is a key task in The image processing area at partitioning a digital image into multiple objects which share some common properties. Image segmentation is a critical issue as the quality aspect of its outcomes has a strong influence on the posterior image understanding task. Among its practical applications are medical imaging (where it is employed for tasks such as tumor location, computer guided surgery, and diagnosis); traffic control systems; object location in satellite images and machine vision. Segmentation is one of the most important techniques for image processing]. The purpose of segmentation is to partition an image into distinct, semantically meaningful entities by defining boundaries between features and objects in an image based on some constraint, or homogeneity predicate. Specifically, the segmentation problem is defined as sufficiently partitioning an image into non-overlapping regions.

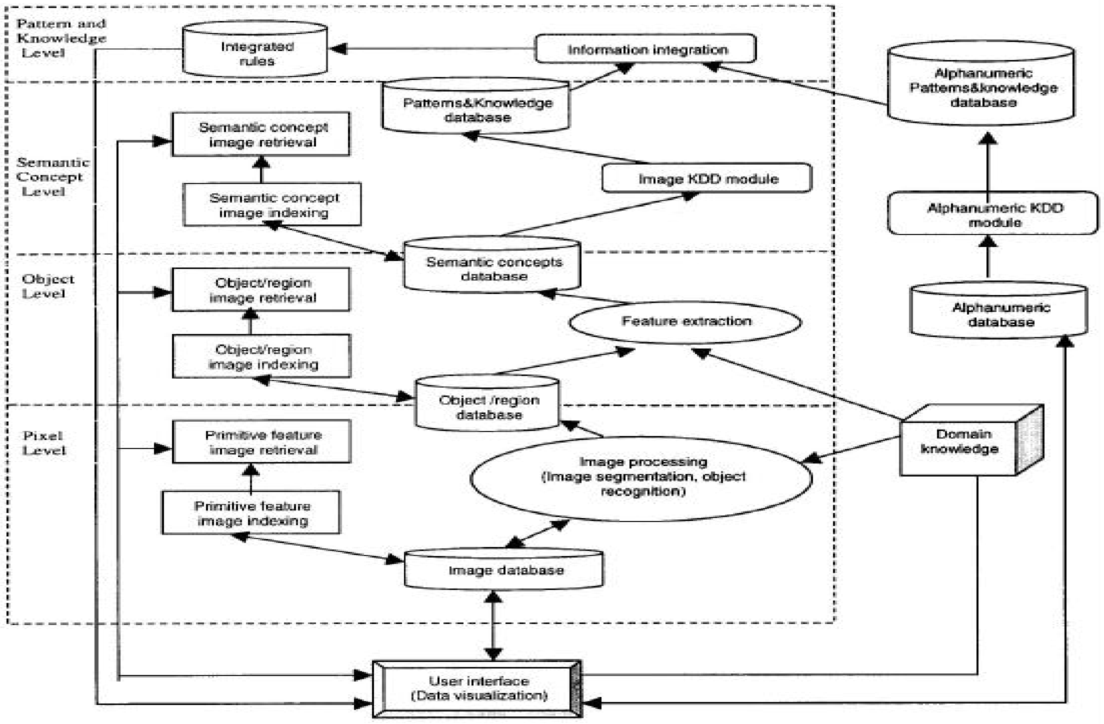

Pattern and Knowledge Level
Integrated rules
Information integration
Alphanumeric Patterns&knowledge database
Patterns&Knowledge database
Semantic Concept Level
Semantic concept image retrieval
Semantic concept image indexing
Image KDD module
Alphanumeric KDD module
Semantic concepts database
Object Level
Object/region image retrieval
Object/region image indexing
Feature extraction
Alphanumeric database
Object /region database
Pixel Level
Primitive feature image retrieval
Primitive feature image indexing
Image processing (Image segmentation, object recognition)
Domain knowledge
Image database
User interface (Data visualization)

Fig.-3.1 An information-driven image mining.

Segmentation can therefore be formally defined as follows:
If F is the set of all pixels and P( ) is a 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 (S1, S2, ..., Sn) such that:

U*ni S F* =1 = *with* I = Æ, *i j S S* i ¹ *j . The homogeneity predicate*
P(Si) = true for all regions (Si) and P(Si U *Sj)=false , when Si is adjacent to Sj.*
Homogeneity predicates are usually based on image intensity, *color*, texture, etc. According to Harlick and Shapiro, image segmentation can be classified into these categories:
spatial clustering, split and merge schemes, and region growing schemes.

3.2 Spatial Clustering

Haralick and Shapiro present that the difference between clustering and segmentation is that in image segmentation, grouping is done in the spatial domain of the image, while clustering is done in measurement space. It is also possible for clustering to result in overlapping regions, while that is not the case for segmentation results. Clustering and spatial segmentation can be combined to form spatial clustering, which combine histogram techniques with spatial linkage techniques for better results.

3.3 Split and Merge Segmentation:

Regions in an image are a group of connected pixels with similar properties]. The split method begins with the entire image, and repeatedly splits each segment into quarters if the homogeneity criterion is not satisfied. These splits can sometimes divide portions of one object. The merge method joins adjacent segments of the same object. In intensity based segmentation, the boundaries that separate regions may need to be redefined due to under- or over-segmentation of regions. Split and merge segmentation can also handle this task. Under-segmented regions are corrected by adding boundaries to, or splitting, certain regions that contain parts of different objects. Over segmented regions are corrected by eliminating false boundaries and merging adjacent regions if they belong to the same object or feature.

3.4 Region Growing

The focus of the remainder of this thesis will be with this class of segmentation. Region growing has shown to be a very useful and efficient segmentation technique in image processing. Region growing in its simplest sense is the process of joining neighboring points into larger regions based on some condition or selection of a threshold value. Seeded region growing starts with one or more seed points and then grows the region to form a larger region satisfying some homogeneity constraint. The homogeneity of a region can be dependent upon any characteristic of the region in the image: texture, color or average intensity.

4. THE EXPERIMENT ON SEGMENTATION TECHNIQUES

4.1 Description of Test Images:

The 10 test images used in the following experiments are taken , full color anatomical images are from the thorax and abdomen regions of the Male dataset. The images are stored as 24-bit 2046x1214 pixel RGB images in RAW format. Color images can be separated into color components based on a specific model. Some of the common color models include red, green, blue (RGB), luminance, chrominance (YUV) and hue, saturation, intensity (HSI). The images are decomposed into three parts representing each of the three components (i.e. red, green, blue for the RGB color model). Our application resizes the image proportionally to an 8-bit 512x302 pixel resolution image. The resolution is reduced so that more image slices can be kept in memory. The reduction from a 24-bit image to 8-bit image results in utilizing only the red component of the original RGB image to retain the color information, since information is lost when converting color images to grey-scale images. We expect to retrieve more detailed edge information than that retrieved from performing operations on grey-scale images. The original 10 test images are shown in Figure 5.1a-j. For each of the experiments in this section we attempt to segment the liver from the images. Figure 5.2 shows one test image outlining the ideal region to be segmented from all of the images.

```
Volume_Grow();
Let W be the set of dataset voxels and t be a threshold on
magnitude difference
S = {};
Choose seed voxel w0 with intensity |w0|.
Determine median intensity I in the window of voxels about w0.
Recursive_Region_Grow(W, S, w0, I, t).
Remove Isolated Interior Voxels; Close.
Expand Region Boundary; Dilate One Voxel.
Recursive_Volume_Grow( W, S, w0, I, t):
S = S+W.
If wi Î W adjacent to w0, wi Ï S, and
if |I| - |wi| < t then
Recursive_Region_Grow(W, S, wi, I, t).
```

Figure 4.1: Volume-Growing Algorithm

4.2 Reconstruction Tool:

The overall goal of this proposed paper is to use an appropriate segmentation technique to segment 2D regions to form one 3D object. The 3D objects are rendered using the free source toolkit, ImageJ, using the stack of 2D segmentations as input into the application. In addition to 3D projections, ImageJ can display, edit, analyze, process, save and print 8-bit, 16-bit and 32-bit images. It can read many image formats including TIFF, GIF, JPEG, BMP, DICOM, FITS and "raw". It supports "stacks", a series of images that share a single window. ImageJ was designed with an open architecture that provides extensibility via Java plug-in that can be written with its built in editor and Java compiler. User-written plug-in make it possible to solve almost any image processing or analysis problem.

4.3 The Experiments using MIST:

This section is dedicated to showing how the MIST algorithm evolved into its final state. The experiments conducted in this section use a seeded region growing algorithm. The region is grown using the threshold, equal to the standard deviation of each input image in the sequence. The idea is to segment the same anatomical feature from each of the sequential 2D images. To accomplish this task, the center of mass of a segmented region is used as the seed point for the next image in the sequence. These 2D segmentations are joined together using ImageJ, to create a 3D visualization of the object of interest. do not alter the size of our segmentations by a significant amount. This ensures us that the segmentations from the modified MIST algorithm produced segmentations with consistent sizes as the segmentations resulting from the MIST algorithm in Experiment Three.

Image Number	Seed Point for Experiment One	Area of Region after Experiment One, in pixels	Seed Point for Experiment Two	Area of Region after Experiment Two, in pixels
1	(198, 160)	11,647	(198, 160)	12,984
2	(214, 143)	11,176	(215, 143)	12,315
3	(213,144)	11,381	(213, 145)	15,675
4	(210, 130)	1	(209, 129)	12,597
5	(212,145)	11,623	(209, 129)	12,597
6	(212, 145)	11,098	(212, 145)	12,961
7	(211, 143)	11,415	(212, 143)	12,383
8	(211, 143)	3	(213, 143)	13,038
9	(211, 142)	1	(213, 143)	12,966
10	(211, 142)	10,942	(213, 141)	12,623

TABLE#5.1: Comparing Area of segmented regions from Experiments

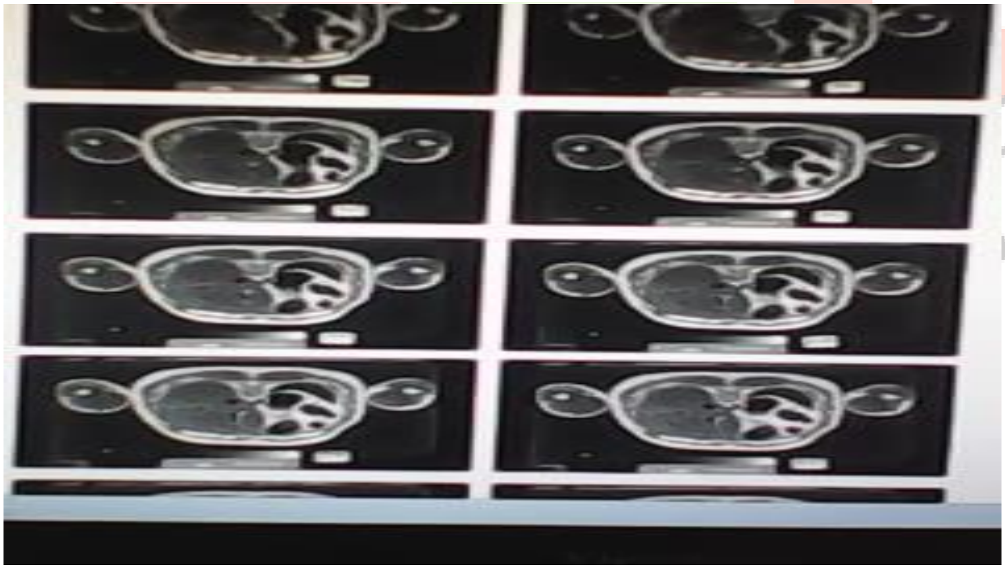


Figure-5.3 a-j: The set of 10 sequential 2D Test images used for Experiments.

Image Number	Area of Region after Experiment Three, in pixels	Area of Region after Experiment Four, in pixels
1	12,984	13,603
2	12,315	13,866
3	12,466	13,033
4	12,597	13,269
5	12,961	13,662
6	12,383	13,001
7	12,038	13,631
8	12,966	13,589
9	12,733	13,287
10	12,623	13,153

Table#5.2 Area of Segmented regions in Experiments

6. CONCLUSION

The MIST algorithm corrects the issues faced with Newman's algorithm. Region growing guarantees the segmentation of a connected closed contour, while the use of the contour filling operation ensures that the segmented region of interest is free of gaps and hole artifacts unlike the segmentation produced by Newman etal. In this chapter we have presented results from the segmentation results produced by Newman's algorithm as well as in each step of our proposed MIST algorithm. Experimental results show that our MIST method performs better for whole organ and tissue segmentation. Image Segmentation of the Visible Human Data set offers many additions to the original goal of a three-dimensional representation of a computer generated anatomical model of the human body and to the general study of human anatomy. In this paper, we have presented a new automatic region growing algorithm called the Medical Image Segmentation Technique (MIST) that improves image segmentation of 2D contours for the purpose of reconstructing 3D anatomical structures. It is our first attempt to address the issue of segmenting organs, tissue and other structures from color anatomical images. Seeded region growing offers several advantages over conventional segmentation techniques. Unlike gradient and Laplacian based edge detection methods, a region found by region growing is guaranteed to be connected and consist of a one pixel thick boundary, since we only add pixels to the exterior of our region. MIST addresses the adjacency problem, therefore the segmented region will never contain too much of adjacent tissues, as long as the parameters are defined correctly. In addition, our technique guarantees that he seed is contained in the region by addressing what we call the centroid problem, unlike the method presented in [10]. We have compared the results from MIST with papers attempting to achieve the same goals. In our experiments, our method proved to perform better and produce better 3D visualizations.

7. ABOUT AUTHORS

Ms. A.Gowthami, Working as Assistant Professor of E. C. E. Dept. in Holy Mary institute of Technology & Science, Hyderabad. She Graduated B. Tech. in Electronics & Communications Engineering from J. N. T. Univesity, Kakinada, in the year 2014. She Post-Graduated M. Tech. In DECS from J. N. T. Univesity, Kakinada, in the year 2018. Her Research interests are Antennas, V.L. S. I. System Design, Image Processing, Cyber Security.

Mr. M. Sri Venkat Rami Reddy, Working as Assistant Professor of E. C. E. Dept. in Holy Mary institute of Technology & Science, Hyderabad. He is Graduated B. Tech. in Electronics & Communications Engineering from J. N. T. University, Hyderabad, in the year 2004. He Post-Graduated M. Tech. In V. L. S. I. System Design from J. N. T. University, Hyderabad, in the year 2012. He is CISCO (Routing, Switching & Security) Certified Networking Engineer.

His Current Research interests are V. L. S. I. System Design, SoC, Image Processing, Cyber Security and **A**rtificial **I**ntelligence & **M**achine **L**earning (**AI& ML**). He is a **L**ife **M**ember of **I**ndian **S**ociety for **T**echnical Education (**LM-ISTE**), New Delhi, India.

Mr. David Solomon Raju is pursuing Ph.D from Rayalaseema University, Kurnool, Andhra Pradesh in Image Processing. Completed M. Tech in ECE with specialization Systems and Signal Processing from JNTU Hyderabad, in 2007 and B. Tech in Electronics and Communication Engineering from V. R. Siddhartha Engg. College, Nagarjuna University, Guntur, Andhra Pradesh in 2000. Currently working as an Associate Professor and Head of the Dept. in the ECE Department at Holy Mary Institute of Technology and Science (College of Engineering), Hyderabad.

Areas of research interest include Image Processing, Segmentation techniques, Object-Based Compression, Image Sequence Restoration and Enhancement, Cryptography, and Signal Processing. He is a Life Time member of ISTE, ISOI, Fellow Member of IETE, Member of IE and Member of IAENG.

8. REFERENCES

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, (2):157–166, 1994.

[2] C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.

[3] W. L. Briggs, S. F. McCormick, et al. A Multigrid Tutorial. Siam, 2000.

[4] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC, 2011.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. IJCV, pages 303–338, 2010.

[6] R. Girshick. Fast R-CNN. In ICCV, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010.

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. arXiv:1302.4389, 2013.

[10] Dean, Jeff; Monga, Rajat; et al. (November 9, 2015). "TensorFlow: Large-scale machine learning on heterogeneous systems" (PDF). TensorFlow.org. Google Research. Retrieved November 10, 2015.

[11] Perez, Sarah (November 9, 2015). "Google Open-Sources The Machine Learning Tech Behind Google Photos Search, Smart Reply And More". TechCrunch. Retrieved November 11, 2015.

[12] Oremus, Will (November 9, 2015). "What Is TensorFlow, and Why Is Google So Excited About It?". Slate. Retrieved November 11, 2015.

[13] Ward-Bailey, Jeff (November 25, 2015). "Google chairman: We're making 'real progress' on artificial intelligence". CSMonitor. Retrieved November 25, 2015.

[14] TensorFlow (April 26, 2018). "Introducing Swift For TensorFlow". Medium. Retrieved August 14, 2019. not just a TensorFlow API wrapper written in Swift

[15] "tensorflow: TensorFlow for R". RStudio. February 17, 2018. Retrieved February 18, 2018.

[16] "Tensorflow Release 1.0.0" Docs.

[17] "Build and train machine learning models on our new Google Cloud TPUs". Google. May 17, 2017. Retrieved May 18, 2017.

[18] "Cloud TPU". Google Cloud. Retrieved May 24, 2019.

[19] "A MATLAB wrapper for TensorFlow Core". November 3, 2019.
Retrieved February 13, 2020.

[20] He, Horace (October 10, 2019). "The State of Machine Learning Frameworks in 2019". The Gradient. Retrieved May 22, 2020.

[21] "TensorFlow in other languages | TensorFlow Core". TensorFlow. Retrieved August 2019.

[22] "Release 2.4.0". 17 June 2020. Retrieved 18 June 2020.

[23] "Kerasbackends". keras.io. Retrieved 2018-02-23.

[24] Jump up to: "Why use Keras?". keras.io. Retrieved 2020-03-22.

[25] Vicente, Sara; Rother, Carsten; Kolmogorov, Vladimir (2011). Object cosegmentation. IEEE. doi:10.1109/cvpr.2011.5995530. ISBN 978-1-4577-0394-2.

[26] Linda G. Shapiro and George C. Stockman (2001): “Computer Vision”, pp 279–325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3

[27] H. Liu, H. Motoda.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publisher, 1998.

[28] Haralick, R.M., and Shapiro, L.G. “SURVEY: image segmentation techniques”, Computer Vision Graphics Image Processing, 1985, 29, pp. 100-132.

[29] Revol, C., and Jourlin, M., “A new minimum variance region growing algorithm for image segmentation”, Pattern Recognition Letters, 1997, 18, pp. 249-258.

[30] Pal, N., and Pal, S., “A Review on Image Segmentation Techniques”, Pattern Recognition, 1993, 26, pp. 1277-1294.

[31] Jiawei Han, Micheline Kamber.: Data Mining: Concepts and Techniques. Simon Fraser University, 2001.

[32] Rafael C. Gonzalez, Richard E. Woods.: Digital Image Processing, 1992.

[33] C. Clavier & M. Joye, Universal Exponentiation Algorithm, Cryptographic Hardware and Embedded Systems-CHES 2001, Ç. Koç, D. Naccache & C. Paar (editors), Lecture Notes in Computer Science, 2162, Springer-Verlag, 2001, 300–308.

[34] K. Gandol., C. Mourtel & F. Olivier, Electromagnetic Analysis: Concrete Results, Cryptographic Hardware and Embedded Systems-CHES 2001, Ç. Koç, D. Naccache & C. Paar (editors), Lecture Notes in Computer Science, 2162, Springer-Verlag, 2001, 251–261.

[35] D. E. Knuth, The Art of Computer Programming, vol. 2, “Seminumerical Algorithms”, 2nd Edition, Addison-Wesley, 1981, 441–466.

[36] P. Kocher, Timing Attack on Implementations of Diffie-Hellman, RSA, DSS, and other systems, Advances in Cryptology-Crypto’ 96, N. Koblitz (editor), Lecture Notes in Computer Science, 1109, Springer-Verlag, 1996, 104–113.

[37] P. Kocher, J. Jaffe & B. Jun, Differential Power Analysis, Advances in Cryptology-Crypto’ 99, M. Wiener (editor), Lecture Notes in Computer Science, 1666, Springer-Verlag, 1999, 388–397.

[38] D. May, H.L. Muller & N.P. Smart, Random Register Renaming to Foil DPA, Cryptographic Hardware and Embedded Systems-CHES 2001, Ç. Koç, D. Naccache & C. Paar (editors), Lecture Notes in Computer Science, 2162, Springer-Verlag, 2001, 28–38.

[39] T. S. Messerges, E. A. Dabbish & R. H. Sloan, Power Analysis Attacks of Modular Exponentiation in Smartcards, Cryptographic Hardware and Embedded Systems (Proc CHES 99), C. Paar & Ç. Koç (editors), Lecture Notes in Computer Science, 1717, Springer-Verlag, 1999, 144–157.

[40] E. Oswald & M. Aigner, Randomized Addition-Subtraction Chains as a Countermeasure against Power Attacks, Cryptographic Hardware and Embedded Systems-CHES 2001, Ç. Koç, D. Naccache & C. Paar (editors), Lecture Notes in Computer Science, 2162, Springer-Verlag, 2001, 39–50.

[41] C. D. Walter, Exponentiation using Division Chains, IEEE Transactions on Computers, 47, No. 7, July 1998, 757–765.

[42] C. D. Walter & S. Thompson, Distinguishing Exponent Digits by Observing Modular Subtractions, Topics in Cryptology-CT-RSA 2001, D. Naccache (editor), Lecture Notes in Computer Science, 2020, Springer-Verlag, 2001, 192–207.

[43] C. D. Walter, Sliding Windows succumbs to Big Mac Attack, Cryptographic Hardware and Embedded Systems-CHES 2001, Ç. Koç, D. Naccache & C. Paar (editors), Lecture Notes in Computer Science, 2162, Springer-Verlag, 2001, 286–299.

[44] C. D. Walter, Precise Bounds for Montgomery Modular Multiplication and Some Potentially Insecure RSA Moduli, Topics in Cryptology-CT-RSA 2002, B. Preneel (editor), Lecture Notes in Computer Science, 2271, Springer-Verlag, 2002, 30–39.