

IMPROVED CELLULAR AUTOMATA SEGMENTATION OF BRAIN TUMORS ON MR IMAGES AND SVM CLASSIFICATION

Sakthivel.S

Assistant Professor / Department of IT,
K.S.Rangasamy College of Technology,
Namakkal, Tamil Nadu, India
mcssakthivel@gmail.com

Dr.P.Rajendran

Professor / Department of CSE,
Knowledge Institute of Technology,
Kakapalayam, Tamil Nadu, India
peerajendran@gmail.com

Varun.M.R

PG Scholar

K.S.Rangasamy College of Technology,
Namakkal, Tamil Nadu, India
mrvarun88@gmail.com

Abstract— Image Processing is one of the developing research areas today. Medical image processing is the most demanding and hugely wanted field in image processing. Brain tumor recognition in Magnetic resonance imaging (MR) has grown to be an developing area in the field of medical image processing. In our paper Magnetic resonance imaging (MRI) is an imaging technique that has played an essential role in neuro science research for studying brain images. Classification is an main part in order to differentiate between normal patients and those who have the possibility of having abnormality or tumor. The proposed method consists of two stages: feature extraction and classification. In first stage features be extracted from images using ICA. In the next stage, extracted features are feed as input to SVM classifier. It classify the images between normal and abnormal along with type of disease depending upon features. . The aim of this paper is to design an automated tool for brain tumor detection using MRI scanned image records. Detection and extraction of tumor from MRI scan images of the brain was done using MATLAB software.

Keywords- Brain tumor segmentation, cellular automata, contrast enhanced magnetic resonance imaging (MRI), necrotic tissue segmentation, seeded segmentation.

I. INTRODUCTION

In imaging science, **image processing** is any form of indicate processing for which the giving information is an image, such as a photograph or video frame; the getting the information of image processing may be either an image or a set of distinctiveness or parameters related to the image. Mainly image-processing technique entail take care of the image as a two-dimensional pointer and apply standard signal-processing technique in the direction of it. Image processing usually refers to digital image processing, but visual and analog image processing also be feasible. An image processing is different in the “true world” is considered to be a function of two actual variables, for example, $a(x,y)$ with a as the amplitude (e.g. brightness) of the image at the real match up position (x,y) . Current digital technology has prepared it feasible to control multi-dimensional signals with systems that sort from simple digital circuits to advanced similar computers. The goal of this manipulation can be divided into three categories:

- Image Processing (image in -> image exposed)
- Image Analysis (image in -> measurements exposed)
- Image Understanding (image in -> high-level description exposed)

for the most part of image processing systems require that the images be available in digitized form, that is, array of finite length binary expressions. For digitization, the known Image [1] is sample on a discrete framework and each sample or pixel is quantized using a finite digit of bits. The digitized image is processed by

a supercomputer. To display a digital image, it be first transformed into analog signal, which is scanned against a display.

Before going to dealing out an image, it is changed into a digital form. Digitization includes variety of image and quantization of sampled values. Later than converting the image into fragment information, processing is performed. This processing technique may be Image development, Image re-establishment, and Image density.

Image development:

It refers to prominence, or sharpening, of image features such as borders, or contrast to make a graphic display more useful for display & analysis. This process does not increase the inbuilt information substance in data. It includes gray level & contrast management, noise lessening, edge christening and sharpening, filtering, interpolation and exaggeration, pseudo coloring, and so on [7].

Image re-establishment:

It is concerned with filtering the observed image to reduce the effect of degradations. Effectiveness of image re-establishment depends on the extent and correctness of the knowledge of degradation process as well as on clean design. Image re-establishment differs from image development in that the latter is concerned with more extraction or prominence of image features.

Image density:

It is disturbed with minimizing the number of bits required to represent an image. Application of solidity be in relay TV, remote sense via satellite, military message via aircraft, radar, teleconferencing, facsimile announcement, for educational & business documents, medical images [4] that occur in supercomputer tomography, magnetic resonance imaging and digital radiology, motion, pictures, satellite imagery, weather map, geological surveys and so on.

II. IMAGE SEGMENTATION

Image segmentation is the process of partitioning a digital image into multiple segments (set of pixels, also known as excellent 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 examine [2]. Image segmentation is in generally used to locate objects and boundaries (lines, curves, etc.) in images. More specifically, 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 [5] is a set of segments that jointly cover the total image, or a set of contours extracted from the image. Each of the pixels in a region is parallel with respect to some characteristic or computed property, such as color, greatness, or texture. Adjacent regions are drastically different with value to the same characteristics. When applied to a stack of images, usual in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions [6] with the help of interpolation algorithms like marching cubes.

III. OVERVIEW OF THE PROJECT

Segmentation of brain tissues in gray matter white matter and tumor on medical images is not only of high interest in serial treatment monitoring of “disease burden” in oncologic imaging, but also in advance popularity with the advance of image guided surgical approaches. Exactness of the brain tumor contour is a major step in planning spatially contained radiotherapy (e.g., Cyber knife, iMRT) which is usually done manually on contrast enhanced T1-weighted magnetic resonance images (MRI) in current clinical practice. On T1 MR images acquired after management of a contrast agent, blood vessels and parts of the tumor, where the difference can pass the blood-brain complexity are observed as hyper deep areas. There are different attempts for brain tumor segmentation in the literature which use a single modality, merge multi modalities and use priors obtained from population atlases.

A. SVM Classification

Support vector machines (SVMs), also support vector networks are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. From the Segmented Image, feature extraction and feature selection is done and Classification of segmented image by SVM neural network process and identifies normal or affected image.

B. SVM (Support Vector Machine)

Viewing input data as two sets of vectors in an n-dimensional space, a SVM will construct a separating hyper plane in that space, which maximizes the margin between the two data sets In the case of support vector machine, an object is viewed as a n-dimensional vector and we want to separate such objects with a n-1 dimensional hyper plane. This is called a linear classifier. There are many hyper planes that might classify the data, we also interested in finding out whether we can get the maximum margin between the two data sets. The

goal of SVM is try to address the nearest distance between a point in one class and a point in the other class being maximized and draw a hyper plane to classify two categories as clearly as possible.

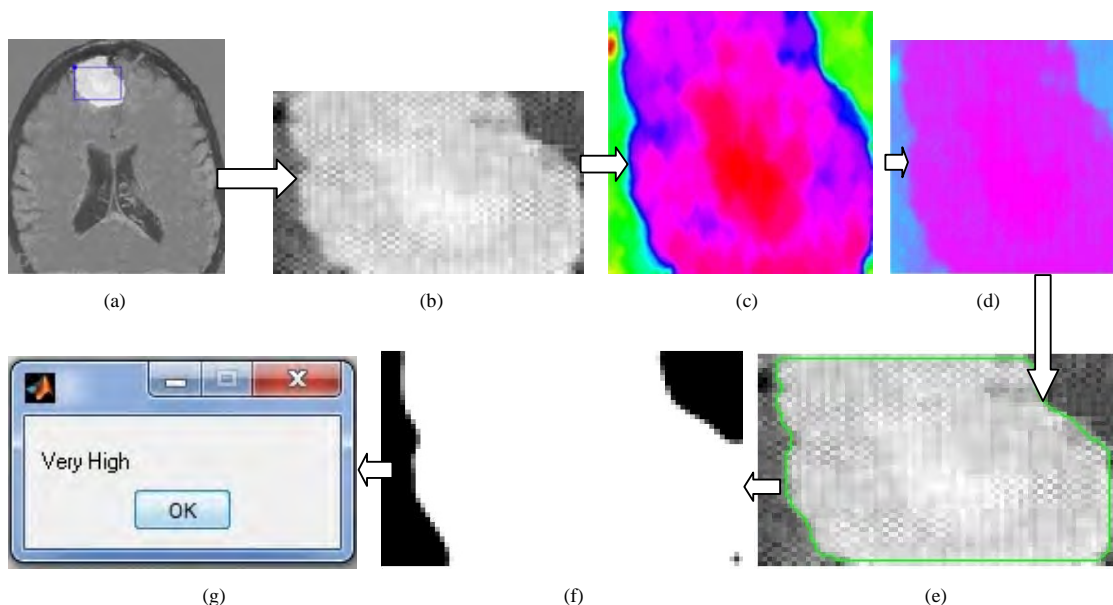


Fig. (a) Original image; (b) Segmented image; (c) Foreground image; (d) Background image ;(e) Green color edge; (f) SVM testing image; (g) Result is very high.

C. Mean Process

Our goal is to remove the noise using a 'filter' that removes the noise but leaves the break The most essential of these filters is the mean filter, which is **accurately** what Sebastian taught in class: take any pixel (x,y) of the image and the 8 pixels adjacent to it (3x3 grid), or take bigger neighborhoods (5x5 or 7x7 or other pixels, with our pixel, again, in the center of the neighborhood). This is equivalent to the example of the prices of the house. If you believe a neighborhood of one block about Sebastian's house, you get a agent value of Sebastian's block. If you consider a bigger neighborhood, say, 10x10 blocks, you will get a delegate of a larger area.

The filter works like this. Suppose that you have a pixel (the 205 pixel in the middle of the grid) and its 5x5 neighborhood as follows (this could easily be a portion of the sky, moderately smooth and bright):

```

200 201 205 204 198
207 201 12 206 203
205 197 205 200 199
195 199 204 203 203
202 196 202 197 201

```

The pixel that we want to replace, the 205, has a dead pixel (of value 12) just over it. To filter the pixel, we compute the mean:

$$\text{new Pixel} = 125(200+201+205+\dots+197+201) = 4845/25 = 193.8 = 194$$

In our newest filtered image, we will now replace the 205 with our mean, i.e., 194. We do again this for all the pixels of the image.

D. STANDARD DEVIATION

Definition

There are two common text definition for the standard deviation s of a data vector X.

$$(1) s = \left(\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}}$$

$$(2) s = \left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}},$$

where

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

and n is the number of elements in the sample. The two forms of the equation change only in $n - 1$ versus n in the divisor.

DESCRIPTION

$s = \text{std}(X)$, where X is a vector, returns the standard deviation using (1) above. The result s is the square root of an unbiased estimator of the variance of the population from which X is drawn, since X consists of autonomous, identically distributed samples.

If X is a matrix, $\text{std}(X)$ returns a row vector containing the standard deviation of the elements of every column of X . If X is a multidimensional array, $\text{std}(X)$ is the standard deviation of the elements along the first non singleton dimension of X .

$s = \text{std}(X, \text{flag})$ for $\text{flag} = 0$, is the same as $\text{std}(X)$. For $\text{flag} = 1$, $\text{std}(X, 1)$ returns the standard deviation using (2) above, producing the second moment of the set of values about their mean.

$s = \text{std}(X, \text{flag}, \text{dim})$ computes the standard deviations along the dimension of X specified by scalar dim . Set flag to 0 to normalize Y by $n-1$; set flag to 1 to normalize by n .

The input array, X , should be of type double or single for all syntaxes.

E. CORRELATION COEFFICIENTS

DESCRIPTION

$R = \text{corrcoef}(X)$ returns a matrix R of correlation coefficients calculated from an input matrix X whose rows are observations and whose columns are variables. The matrix $R = \text{corrcoef}(X)$ is related to the covariance matrix $C = \text{cov}(X)$ by

$$R(i, j) = \frac{C(i, j)}{\sqrt{C(i, i)C(j, j)}}.$$

$\text{Corrcoef}(X)$ is the zeroth delay of the normalized covariance function, that is, the zeroth lag of $\text{xcov}(x, \text{coeff}')$ packed into a square array.

$R = \text{corrcoef}(x, y)$ where x and y are column vectors is the same as $\text{corrcoef}([x \ y])$. If the x and y are not column vectors, corrcoef converts them to column vectors. For example, in this case $R = \text{corrcoef}(x, y)$ is equivalent to $R = \text{corrcoef}([x(:) \ y(:)])$.

$[R, P] = \text{corrcoef}(\dots)$ also returns P , a matrix of p-values for testing the hypothesis of no correlation. Each p-value is the probability of receiving a correlation as large as the observed value by random possibility, when the true correlation is zero. If $P(i, j)$ is small, say less than 0.05, then the correlation $R(i, j)$ is important.

$[R, P, \text{RLO}, \text{RUP}] = \text{corrcoef}(\dots)$ also returns matrices RLO and RUP , of the same size as R , containing lower and upper bounds for a 95% confidence interval for each coefficient.

$[\dots] = \text{corrcoef}(\dots, \text{'param1'}, \text{val1}, \text{'param2'}, \text{val2}, \dots)$ specifies extra parameters and their values. Valid parameters are the following.

'alpha'	A number between 0 and 1 to specify a confidence level of $100 \cdot (1 - \text{alpha})\%$. Default is 0.05 for 95% confidence intervals.
'rows'	Either 'all' (default) to use all rows, 'complete' to use rows with no NaN values, or 'pairwise' to compute $R(i, j)$ using rows with no NaN values in either column i or j .

The p-value is computed by transforming the correlation to create a t statistic having $n-2$ degrees of freedom, where n is the number of rows of X . The confidence bounds are based on an asymptotic normal distribution of $0.5 \cdot \log((1+R)/(1-R))$, with an approximate variance equal to $1/(n-3)$. These bounds are accurate for large

samples when X has a multivariate normal distribution. The 'pairwise' option can produce an R matrix that is not positive definite.

BWArea

total = bwarea(BW)

Estimates the area of the objects in binary image BW. total is a scalar whose value corresponds roughly to the total number of on pixels in the image, but might not be exactly the same because different patterns of pixels are weighted differently.

BW can be numeric or logical. For numeric input, any nonzero pixels are considered to be on. The return value total is of class double.

IV. EXISTING SYSTEM

In existing system re-examine the cellular automata CA algorithm to establish the connection of the CA-based segmentation to the graph-theoretic methods to show that the iterative CA framework solves the shortest path problem with a proper alternative of the conversion rule. Next, as our application is in the clinical radio surgical treatment planning, where manual segmentation of tumors are carried out on contrast enhanced T1-MR images by a radio-oncology specialist, we adjust the CA segmentation towards the nature of the tumor properties undergoing radiation therapy by adapting related change the rules. Finally, a gentleness constraint using level set energetic surfaces is imposed over a probability map constructed from resulting CA states.

Seed selection algorithm [3] employs the similar idea to follow the well-known clinical routine to which the clinicians areas used to: the volume of interest (VOI), the tumor seeds and the background seeds are determined by using the line already drawn by the user to calculate the longest diameter of the solid tumor.

Disadvantages of Existing Methods:

EM Segmentation and Otsu are some of existing methods for this processes and Otsu thresholding methods, both of which were based on finding a single "optimal" threshold over the intensity distribution within the tumor volume. This single threshold Selection is not validated against for several 2-D slices. Computational time is not such efficient as our proposed method and it mainly works on homogeneous tumor types.

V. CONCLUSION

This system presented a segmentation algorithm for the problem of tumor definition which display varying tissue characteristics. As the change in necrotic part and enhancing part of the tumor after radiation therapy becomes important, this system also applied the Tumor-cut segmentation to partition the tumor tissue further into its necrotic and enhancing parts. The performance over particularly datasets of highly heterogeneous tissue content demonstrated an overlap in the range 80%–90%, however, with a preferred low surface distance error, standard median surface distances of 1.0–1.5 mm, in that order.

Furthermore, performance change over varying initial seeds were also reported as standard deviations, and shown to be important in assessing true robustness of the proposed algorithm in real application scenarios. The user interaction time is just a few seconds and typical computation times vary between 1 s to 16 min (on a 3.17 GHz dual processor workstation) depending on the volume of the tumor which ranges between 0.5 and 32 cc. Due to the proposed algorithm, computation time can be significantly reduced.

ACKNOWLEDGMENT

We would like to give special thanks to the authors who helped us to finish the project in an efficient manner.

REFERENCES

- [1] E. D. Angelini, O. Clatz, E. Mandonnet, E. Konukoglu, L. Capelle, and H. Duffau, "Glioma dynamics and computatione models: A review of segmentation, registration, and in silico growth algorithms and their clinical applications" *Curr. Med. Imag. Rev.*, vol. 3, no. 4, pp. 262–276, 2007.
- [2] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest," in *ICCV*, 2009, pp. 731–738.
- [3] A. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *ICCV*, 2007, pp. 1–8.
- [4] S. Warfield, K. Zou, and W. Wells, "Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation," *IEEE Trans. Med Imag.*, vol. 23, no. 7, pp. 903–921, Jul. 2004.
- [5] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proc. ICCV*, 2001, pp. 105–112.
- [6] S. Ho, E. Bullitt, and G. Gerig, "Level-set evolution with region competition: Automatic 3-D segmentation of brain tumors," in *Proc. ICPR*, 2002, vol. 1, p. 10532.
- [7] M. Prastawa, E. Bullitt, S. Ho, and G. Gerig, "A brain tumor segmentation framework based on outlier detection," *Med. Image Anal.*, vol. 8, no. 3, pp. 275–283, 2004.