

Algorithmic Development of Volume Segmentation and Analysis of Tumors From Radiological Images

Vaibhav Mohan

Abstract

Computer vision is a branch of Artificial Intelligence which is a rapidly growing discipline in the field of Computer Science. Computer Vision deals with analyzing, processing, acquiring, and understanding images to produce numeric or symbolic information. The image segmentation techniques are used in the field of computer vision for partitioning an image into multiple segments. This paper describes the segmentation technique used in obtaining the tumor volume from radiological images.

We describe a new method for volume segmentation, analysis of the tumor volume, and development of GUI interface for our proposed method in this paper. The active contour model is used in segmentation. The algorithm developed by us finds the coarse boundaries of tumor using a segmentation algorithm, parameterized snake algorithm is used to refine those boundaries. We present the histogram analysis of the region, the histogram analysis of the result after finding the boundaries using our algorithm, the time taken by our algorithm in finding the ROI, and the analysis of spatial histogram of the region in this paper.

1 Introduction

Estimation of a tumor size is useful in analyzing the effects of therapy on patient with cancer[1]. The largest tumor diameter is widely used as an indication of the tumor size, however, it may not give accurate assessment of tumor due to its three-dimensional shape[2, 3]. Therefore, the segmentation of tumor volume is needed in order to estimate its correct size.

Image segmentation is the process of determining the objects and boundaries in an image by partitioning the image into multiple segments[4]. Image segmentation algorithms can extract a set of contours from images that comprise of similar characteristics, for example intensity, color, or texture[4]. Image segmentation can also be used in creating 3D reconstruction of images using various algorithms, for example *Marching cubes* [5]. In medical imaging, segmentation is used in segmenting tumors and other pathologies, measure tissue volumes, and surgery planning[6].

Image segmentation can broadly be demarcated into different methods[7]. For example, the combinatorial methods such as *path-based* graph methods and variational methods viz. *active contour*, *snakes* etc. use explicit boundary representations. The active contour methods have extensively been applied in segmenting tumors from radiological images. Level set,

is an active contour model, have been widely used in segmenting tumors due to its ability to handle complex geometries[8, 9]. However, the level set algorithm is user dependent as it requires adjustment of several parameters. Also, they are time consuming and need a large number of iterations for segmenting the image[1]. Another active contour model, snakes, are used in representing the boundaries in an image with elastic contours[10]. Snakes are widely used in image segmentation[11, 12] due to their ability to guarantee smoothness and continuity of boundaries with small segmentation errors[13]. However, snakes need user interaction for initializing the snake near a target tissue[14, 15]. We describe a new technique for segmenting the image that combines thresholding and mathematical morphology operations in automatically initializing the snakes for segmenting the boundaries that potentially overcomes this problem.

This paper deals with implementation of the proposed image segmentation algorithm based on the parametric snake algorithm. We describe how we segment the tumor volume from given series of images, the visualization of the region of interest on tumor and its analysis. We present an algorithm that extracts the useful information from the radiological images by manually thresholding the image, and then combining morphological operations to enhance object boundaries. This method is provided in our implemented tool for segmenting the tumor volumes by user interaction in case the user is not satisfied with automatic segmentation done by our algorithm.

The remainder of the paper is organized as follows. Section 2 describes the theoretical background for our algorithm. The morphological operators used, the Snake algorithm, the histogram analysis, the thresholding technique, the windowing of image and the spatial transformation of ROI drawn over image is described in this section. Section 3 comprises of details of the algorithms and software development methods used in developing this tool. We also describe the datasets used for implementing this volume segmentation tool in this section. Section 4 presents the results obtained after performing volume segmentation and analyzing the tumor region. Section 5 discusses the results. Finally, Section 6 concludes this work.

2 Theoretical Basis For The Algorithm

In this section, we describe the morphological operations, concepts and algorithms developed.

2.1 Spatial Transformation of ROI

Images comprise of a matrix that contains pixels in the rows and columns that is equal to the number of pixels. We can apply various spatial transforms on this matrix in order to rotate, translate, or scale an ROI. We use a combination of these spatial transformations in order to achieve various tasks while implementing our tool.

2.1.1 Translation of ROI in 2D

In order to translate an image by a given constant distance, we need to multiply each point in the matrix by a translation matrix[16, 17]. The formula for translating a point is given below:

$$\begin{bmatrix} p'_x \\ p'_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \quad (1)$$

where p is the position of point before translation, p' is the position of point after translation, and V is the vector that contains constants by which each point is translated.

2.1.2 Scaling of images in 2D

In order to scale an image, we need to multiply each point of the given image by the given scaling matrix [16, 17]. The formula for scaling the point is given below:

$$\begin{bmatrix} p'_x \\ p'_y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \quad (2)$$

where p is the position of point before scaling, p' is the position of point after scaling, and S is the vector that contains constants by which each point is scaled.

2.1.3 Rotation of images in 2D

In order to rotate an image by specified degree, we need to multiply each point in the given image by the 2D rotation matrix which results in the rotated image [16, 17]. The 2D rotation matrix $[R(\theta)]$ that rotates a point by specified degrees (θ) counter clockwise around origin is given below:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (3)$$

2.2 The Region of Interest (ROI)

The Region of Interest (ROI) is defined as a selected subset of pixels from an image for segmenting tumor boundaries. The ROI is selected on object of interest in the image to apply our algorithm. Once the ROI is selected, our implemented tool traverses through all the images in the given series, and removes the unwanted areas. Our tool can also be used in drawing the tumor boundaries manually.

2.3 Histogram Analysis

An image histogram is a type of histogram that is obtained when we plot the number of pixels for each grayscale pixel intensity present in an image. It can be used in discerning the variations of pixel intensity thus aiding user in selecting the maximum and minimum value for thresholding. These selected values are used in performing segmentation of a given image. In our tool, we have functionality for plotting horizontal, vertical and spatial histogram of a given image. We can also plot the spatial histogram for all image for a specified ROI.

2.4 Thresholding

Thresholding is the most basic method that is used in image segmentation[4, 18]. A binary image is obtained from a grayscale image by thresholding. We have implemented various algorithms in our tool for thresholding the given image that are described below. One approach is to sort all the pixel intensities in ascending order, and then use the median of pixel intensity as a threshold value. We also use manual method to input a minimum value for thresholding. All the pixel intensities that are less than the given threshold value are then excluded from the image. Finally, our last approach for thresholding is by using the histogram to discern the maximum and minimum value for thresholding. In this case, all the pixel intensities that are below minimum value, and above maximum value are removed from the image. Once the binary image is obtained, we perform some morphological operations discussed below before using it as an input to the snake algorithm.

2.5 Morphological operations

This section defines the morphological operations that we have applied on binary images obtained after thresholding the grayscale images.

2.5.1 Erosion and Dilation

Erosion and dilation are fundamental operators for the analysis of spatial structure [16, 17, 19]. An image is viewed as a subset of a 2-dimensional integer space \mathbb{Z}^2 . If A is a binary image in \mathbb{Z}^2 , and B is a structuring element which itself is a binary image in \mathbb{Z}^2 , the erosion of the binary image A is given by the formula:

$$A \ominus B = \{z | (B)_z \cap A^c = \phi\} \quad (4)$$

and the dilation of the binary image A is given by the formula

$$A \oplus B = \{z | (\hat{B})_z \cup A \neq \phi\} \quad (5)$$

where A^c is the complement of A , ϕ is the empty set, z is a point in \mathbb{Z}^2 , B_z is the translation of B by the point z and \hat{B}_z is obtained by reflecting B about its origin and translating this reflection by z . The erosion erodes the boundaries of regions of foreground pixels thereby shrinking the areas of foreground pixels, and expanding the areas of holes within those regions. The dilation operation gradually enlarges the boundaries of regions of foreground pixels thereby expanding the areas of foreground pixels, and shrinking the areas of holes within those regions.

2.5.2 Open and Close

Opening of an image is done by performing an erode operation followed by a dilate operation on a given image[17]. Opening smoothes the contour of an object, breaks narrow isthmuses, and removes thin protrusions[16]. Closing of an image is just opposite of open operation i.e.

a dilate operation is performed followed by an erode operation on a given binary image[17]. Closing operation smoothes, fills small gaps, fuses narrow breaks and eliminates small holes in the contour[16]. The opening of an image is given by the formula:

$$A \circ B = (A \ominus B) \oplus B \quad (6)$$

and the closing of an image is given by the formula

$$A \bullet B = (A \oplus B) \ominus B \quad (7)$$

where A and B are binary image in 2-dimensional integer space \mathbb{Z}^2 .

2.5.3 Fill and Edge

We perform fill and edge operations on our selected ROI. The fill operation is used in filling the area of selected ROI, while edge operation is used to extract the edge from filled ROI.

A hole is a background region surrounded by a connected border of foreground pixels[16]. The fill operations performed on the binary image is used for filling holes that cannot be reached by filling in the background from the edge of the image. The filling of holes is performed using the following iterative procedure:

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad (8)$$

where $k = 1, 2, 3, \dots$, A and B are binary image in 2-dimensional integer space \mathbb{Z}^2 , A^c is the complement of A and X_0 is the array of 0s having same size as that of A . The procedure terminates when $X_k = X_{k-1}$.

Edged ROI is obtained by creating a copy of ROI after applying erode operation on it, and then subtracting the eroded ROI from actual ROI. The formula for extracting the boundary of filled ROI ($\beta(A)$) is given below:

$$\beta(A) = A - (A \ominus B) \quad (9)$$

where A and B are binary image in 2-dimensional integer space \mathbb{Z}^2 .

2.6 Windowing of an Image

Windowing of image is done by changing the dynamic range of image pixels. This functionality is used to set the brightness and contrast of the image that is being displayed on the screen. There are two parameters viz. *window center* and *window width* that is used in windowing. *Window width* is used for specifying the range of the contrast of pixels that is being displayed and *window center* is the central point that is in the range of *window width*. All the pixels that fall outside the window defined by these parameters are either displayed

as black or white. The lowest visible pixel (LVP) and the highest visible pixel (HVP) are given by the formulas:

$$LVP = window_center - (window_width/2) \quad (10)$$

$$HVP = window_center + (window_width/2) \quad (11)$$

This feature is vital in medical imaging visualization as adjusting the window level can help in visualizing the image more clearly.

2.7 The Snake Algorithm

A snake is a spline that is guided by external constraint forces and internal image forces that pull it towards the feature on images such as line or edge of an object so as to minimize the energy[11, 20]. Snakes are valuable tools for segmenting images[12, 11]. Snakes are active contour model, i.e. they lock onto nearby edges in the image and try to find the boundary of the object in image. Snakes are widely used in performing shape recognition, edge detection, and segmentation. They guarantee smoothness and continuity of boundaries found by them with small segmentation error[13].

Snakes do not solve the problem of finding contours entirely. They require some user interactions in order to achieve this. The snake is placed near the region of interest after which it iteratively gets attracted towards the edge of the object by various parameters (discussed below) that control the shape and location of the snake.

A simple elastic snake is defined by a contour that is represented parametrically as

$$c(s) = (x(s), y(s)) \quad (12)$$

where $x(s)$ and $y(s)$ are the co-ordinates along the contour and $s \in [0, 1]$ [20].

The energy function used in this algorithm is a sum of several forces acting on the contour. The energy function of snake is given by the following formula:

$$E = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{Image})ds \quad (13)$$

The parameters α , β and γ can be varied by user and are used for controlling the influence of each energy term on the snake. The energy term E_{Image} , also known as *external* energy term, attracts the snake towards the closest image edge. The E_{Image} is given by the following function:

$$E_{Image} = -||\nabla I|| \quad (14)$$

where $\|\nabla I\|$ is the gradient of the intensity value calculated at each snake point. The energy terms E_{cont} and E_{curv} , also collectively known as *internal* energy terms, are responsible for continuity and smoothness of snake respectively. The E_{cont} term minimizes the first derivative, whereas the term E_{curv} minimizes the second derivative of the snake. The mathematical terms for these two energy functions are given below:

$$E_{cont} = \left\| \frac{dc}{ds} \right\|^2, \text{ and } E_{curv} = \left\| \frac{d^2c}{ds^2} \right\|^2, \quad (15)$$

In our algorithm, we use the binary image formed by thresholding as an input for finding coarse boundaries in the given image. Once we get the coarse boundaries, we use them as an input to snake algorithm for refining the boundaries.

3 Algorithms and Software Development Methods

In this section, we describe what type of datasets we used for testing, how our implemented tool can be used in segmenting the tumor from the given image, and, the algorithm that we use for volume segmentation. The implementation of the algorithm was tested on a system running Windows 7 on a six core Intel Xeon CPU E5-1650 clocked at 3.2 GHz with 64 Gigabytes of RAM. The tool was developed using Microsoft Visual Studio 2010, OpenCV v2.4.8, DCMTK v3.6.0, and .NET framework v4.0.

3.1 Data Sets

The datasets used in testing the implementation were obtained from Radiology Department of Johns Hopkins School of Medicine. The datasets are a series of MRI images conforming to DICOM standards set by National Electrical Manufacturers Association (NEMA) for handling, storing, and transmitting information in the field of medical imaging[21]. Each image in a DICOM series is called a slice. A DICOM data file comprises of various attributes such as patient information, and equipment information using which the DICOM file was created. The dataset that we used in tuning the parameters of the algorithm are a set of 36 256×256 DICOM images of breast. Once the parameters were tuned, we tested the algorithm on 5 different cases of breast datasets. The information pertaining to these datasets is given in Table 1.

Dataset	Width	Height	Number of slices
Breast - Dataset1	256	256	128
Breast - Dataset2	256	256	102
Breast - Dataset3	512	512	56
Breast - Dataset4	256	256	84
Breast - Dataset5	256	256	364

Table 1: The information of breast datasets on which the algorithm was tested

In our tool, the user gets a preview of all the DICOM image. The user then selects the images that are relevant, draws ROI on it, sets threshold values, and runs the algorithm to obtain the segmented volume of tumor.

3.2 Tumor segmentation and volume analysis algorithm

Description of the tumor segmentation algorithm is given in this section. The pseudocode for our algorithm is given in figure 1. First we select the images that we need for segmenting, then we select the region of interest, and finally we run our algorithm for segmenting the tumor volume. If no ROI is specified, then the algorithm is run on entire image. The complete algorithm comprises of four steps: 1) Extraction of ROI from the given image, 2) Image enhancement, 3) Morphological operations on the given image, and 4) Finding contours and refining the boundaries using the Snake algorithm. A description of each step of our algorithm is described in subsequent subsections.

```

function SEGMENTIMAGE(Image, ROI)
    image = removeAreaOutsideROI(Image, ROI)
    for i = 1 → num_rows(Image) do
        for j = 1 → num_cols(Image) do
            if (pixelIntensityROI[i][j] == 0) then
                pixelIntensityimage[i][j] = 0
            end if
        end for
    end for
    for i = 1 → num_rows(image) do
        for j = 1 → num_cols(image) do
            if (pixelIntensityImage[i][j] < threshold) then
                pixelIntensityimage_binary[i][j] = 0
            else
                pixelIntensityimage_binary[i][j] = 1
            end if
        end for
    end for
    image_binary = imopen(image_binary)
    image_binary = imclose(image_binary)
    image_binary = imfill(image_binary)
    contours = findcontours(image_binary)
    for each contour ∈ contours do
        points = SnakeRefineBoundary(image_binary, contour,  $\alpha$ ,  $\beta$ ,  $\gamma$ )
    end for
    Save the boundary refined by Snake
end function

```

Figure 1: The algorithm for segmenting tumor from given image

3.2.1 Step 1: Extraction of ROI from the given image

The user selects the ROI on the image. The tool then discards the pixels outside the ROI. The selected filled ROI, the selected edged ROI, and the resultant image after extraction is shown in figure 2, 3 and 4 respectively.

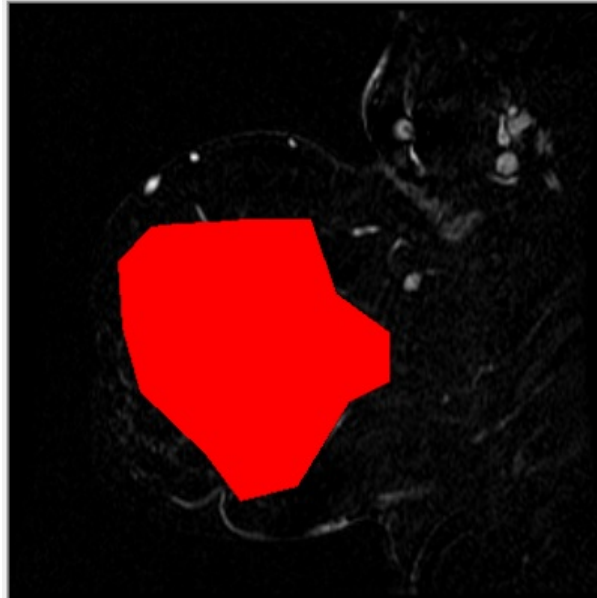


Figure 2: The selected filled ROI on the original image

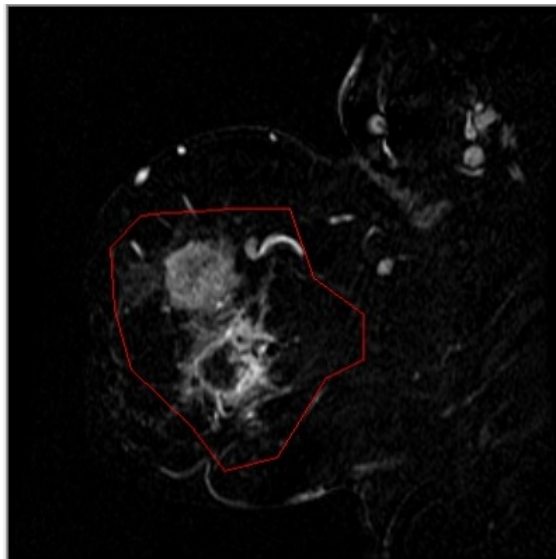


Figure 3: The edged ROI on the original image

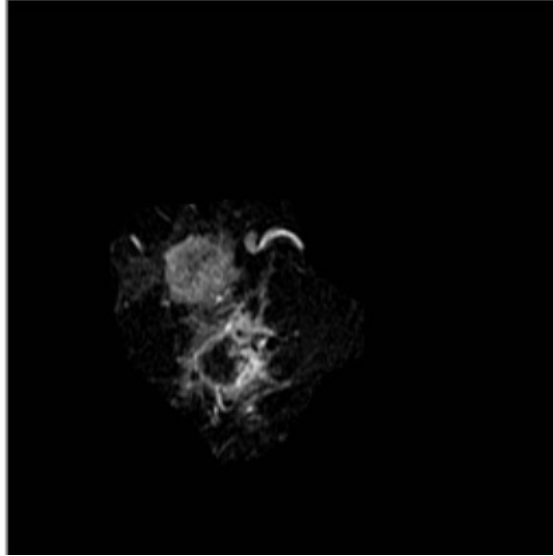


Figure 4: The image after extraction using ROI

3.2.2 Step 2: Image enhancement

Once we obtain the extracted image, we use aforementioned thresholding technique in obtaining the binary image from the extracted image and shown in figure 5.



Figure 5: The binary ROI image after thresholding

3.2.3 Step 3: Morphological operations on the given image

In this step, we perform open and close morphological operations on the image that we obtained after thresholding. The open operation removes all the non connected components, while the close operation fills the holes in remaining connected components. After this, a fill operation is used for filling holes present in the ROI and shown in figure 6 and 7 respectively.

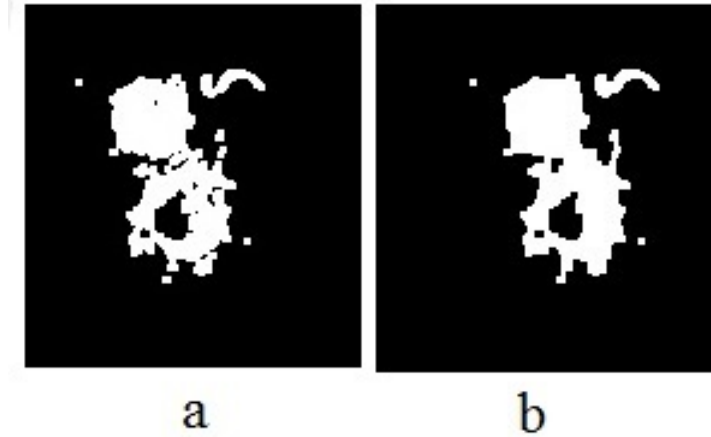


Figure 6: a: After applying open operation, b: after applying close operation to the ROI image



Figure 7: The resultant image after applying fill operation to the ROI image

3.2.4 Step 4: Finding contours and refining the boundaries using Snake algorithm

In this step, the binary image obtained from the previous step is used as an input to the Snake algorithm. We have used parametric Snake which uses the boundaries of the ROIs found in previous step as initial boundary and refines the boundary by minimizing the energy function mentioned in equation 11. The resultant image after running the snake algorithm (edged and filled ROI) is given in figure 8 and 9 respectively.

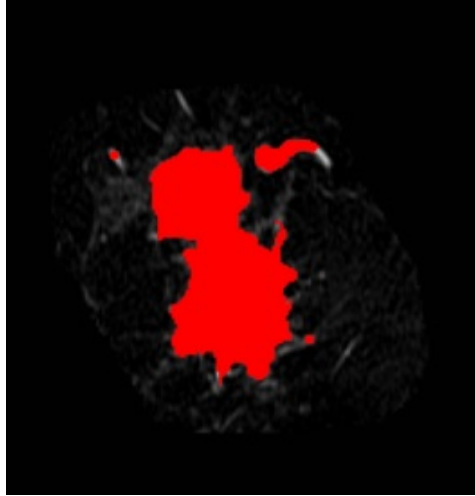


Figure 8: The tumor boundary found by Snake algorithm (filled ROI)

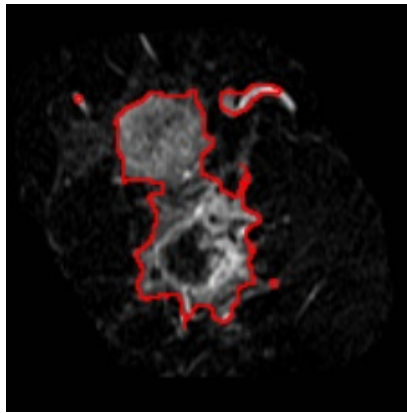


Figure 9: The tumor boundary found by Snake algorithm (edged ROI)

3.2.5 The calculation of slice area, slice volume, and total volume

Once we obtain the boundaries from snake algorithm, we calculate the number of pixels that are inside the region found by the Snake algorithm. This is the area of slice in pixels. The DICOM image format contains the attributes *Pixel Spacing* [Dicom Tag (0018, 1164)] and *Slice Thickness* [Dicom Tag (0018, 0050)] which is used in calculating the actual slice area and slice volume. The formulas for calculating slice area (in square cm) and slice volume (in cubic cm) is given below:

$$sliceArea = Pixel_Area \times Pixel_Spacing^2 \times 0.01 \quad (16)$$

$$sliceVolume = sliceArea \times Slice_Thickness \times 0.1 \quad (17)$$

The volume of the entire tumor can be calculated by calculating the volume of all the slices. If we have n number of DICOM images, then the formula for calculating total volume is given by:

$$totalVolume = \sum_{i=1}^n sliceVolume_i \quad (18)$$

3.3 Software Development

Our implemented algorithm is programmed using C++ in the user friendly GUI. The GUI is used for volume analysis and visualizing the tumor regions. We describe the functionality of our tool in this subsection.

3.3.1 The Volume Tool Software

The first screen of the GUI is Volume Tool and this is shown in figure 10. This screen allows us to view the images in the DICOM series. We have two browse buttons one each for selecting the directory from where DICOM series is to be loaded, and the other for selecting the directory where tool will store the results of algorithm. When the DICOM directory is selected, the tool processes the entire directory and shows the sequences present in the directory in a dialog box. The user can then select the DICOM series that they want to view. If no series is selected, the software by default loads all the series. The dialog box displayed after processing the folder is shown in figure 11. The tool also displays certain information like patient name, slice location, total number of images in the series, and the selected DICOM series. Once we know what images we want to use for running the algorithm, we can either specify the range of images to be selected, or a series of image number value separated by comma. The tool also supports jumping to the specified image number. Once all the parameters are selected, the *Run algorithm with selected configuration* button is pressed in order to go to the next screen.

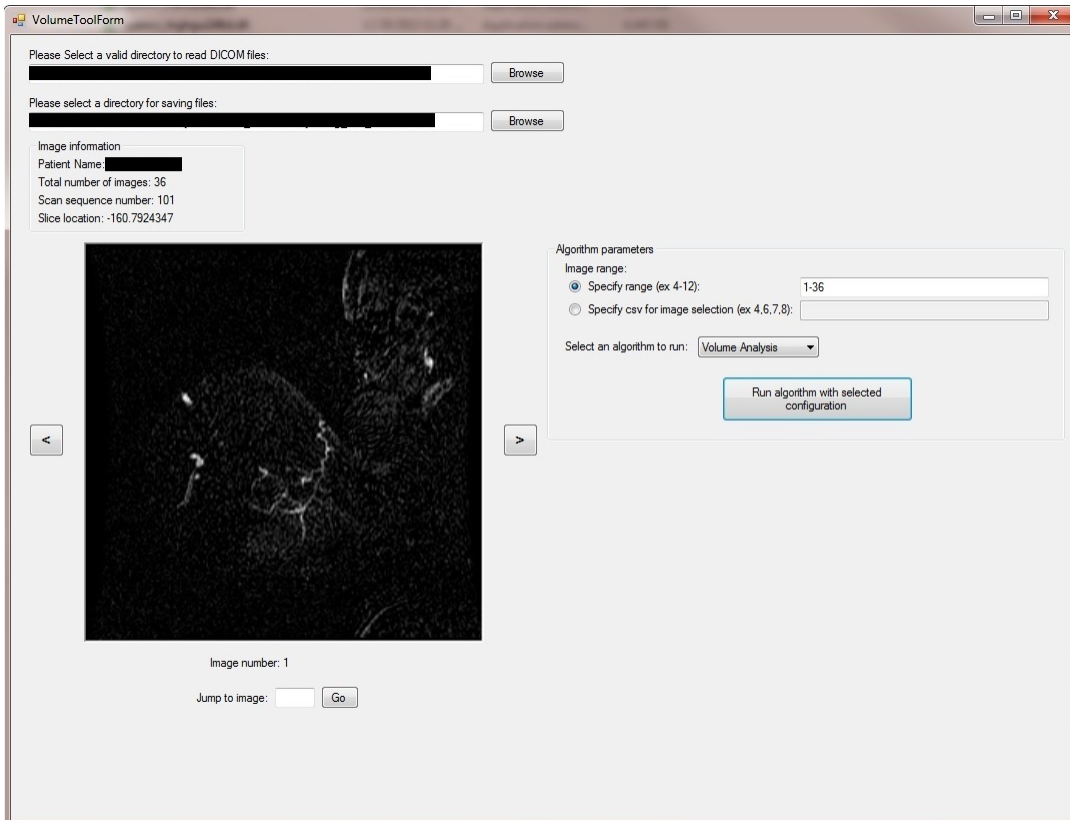


Figure 10: The first screen of Volume Tool Software GUI

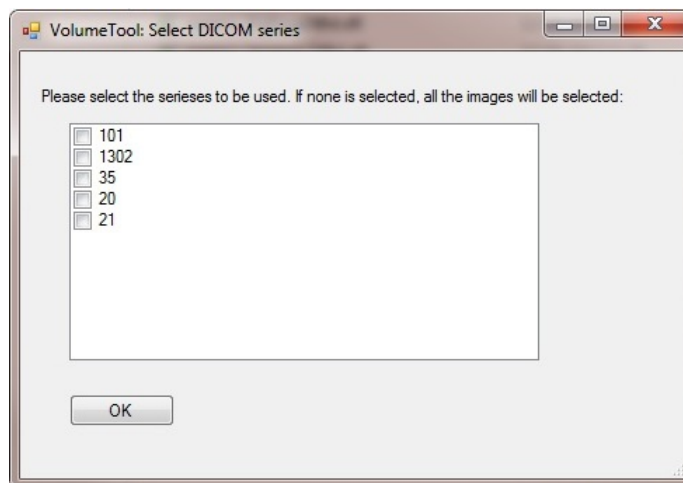


Figure 11: Dialog box for selecting DICOM series

3.3.2 The Volume Analysis Window

Once the user selects the images on which algorithm is to be run, the *Volume Analysis* window is displayed from GUI with a number of group boxes. The *Volume Analysis* window is shown in figure 12. This window displays the selected images from the last window. The GUI interface also have functionality for viewing previous and next image. The *Window level* control bar is used to vary the window center of the image being displayed, while the *Window width* control bar is used for varying the window width of the pixel intensities being displayed on the GUI screen. The *Histogram* window by default displays the histogram of current image.

The Snake parameters group box

This group box contains the control for configuring Snake algorithm's parameters. We can specify the value of $\alpha, \beta, \gamma, \epsilon$, and the number of iterations for snake using this group box. The parameter α is used for setting the weight of the energy term E_{cont} , which is responsible for the continuity of snake, the parameter β is used for setting the weight of the energy term E_{curv} , which is responsible for the smoothness of snake, the parameter γ is used for setting the weight of the energy term E_{Image} , which is responsible for attracting the snake towards the closest image edge, and the parameter ϵ is used for setting the convergence criteria for Snake algorithm.

The ROI control group box

This group box contains the multiple controls for manipulating the behavior of ROI. It displays the area and volume of the currently selected ROI. The ROI type is used for selecting the type of ROI that we want to draw on the image. There are three types of ROI selection type viz. free hand, rectangle, and elliptical. We can also change the color of drawn ROI by setting the ROI color parameter. There is also a control for whether we want to show the ROI or not, and whether the ROI should be filled or edged. The group box also contains controls for rotating, loading, saving and clearing the ROI. There is also a control for specifying whether we want to use automatic thresholding of the image or specify a minimum threshold value. Figure 13 shows ROIs selected in various color and shapes. Figure 14 shows the same ROIs after edging. The *Done* button is pressed once we are satisfied with our ROI selection. We can also cancel our ROI selection at any time by pressing the *Cancel* button.

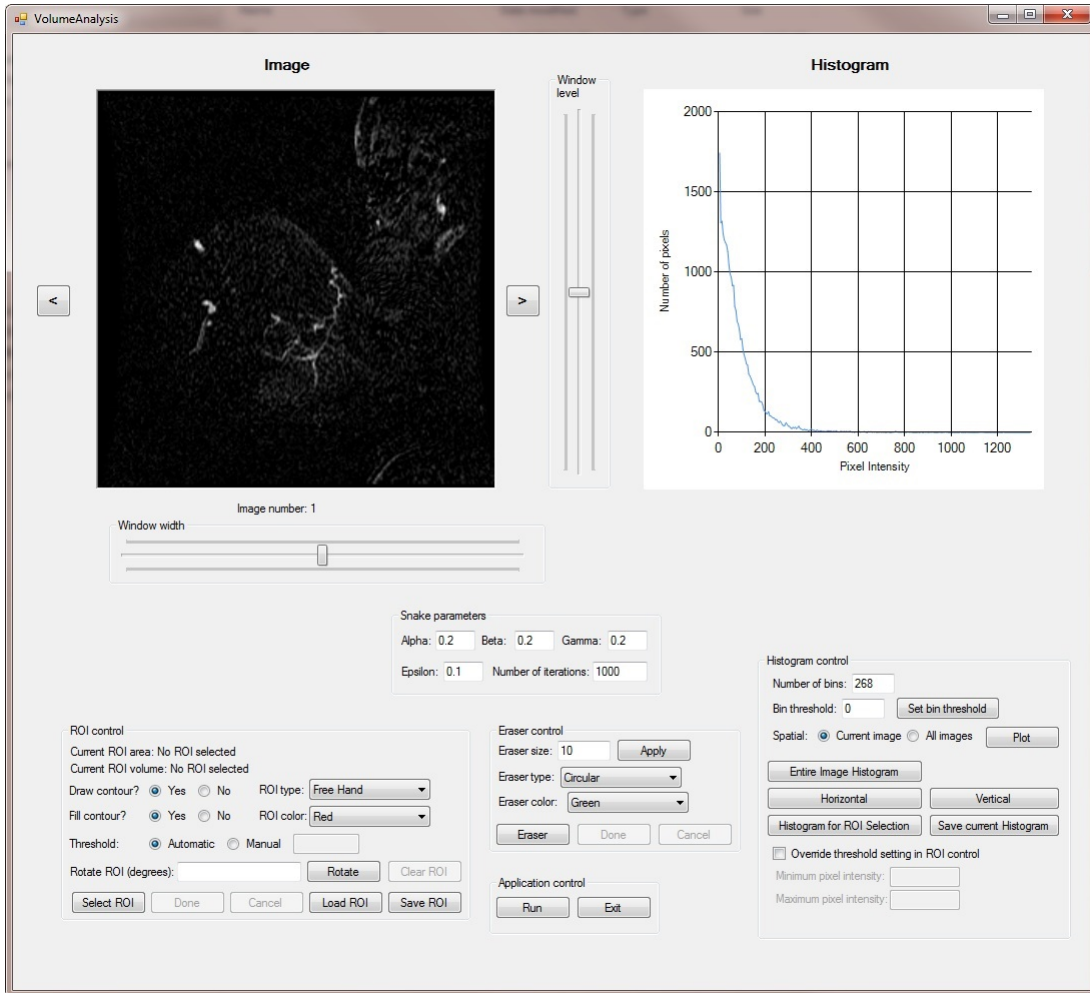


Figure 12: The Volume Analysis GUI window with different group boxes

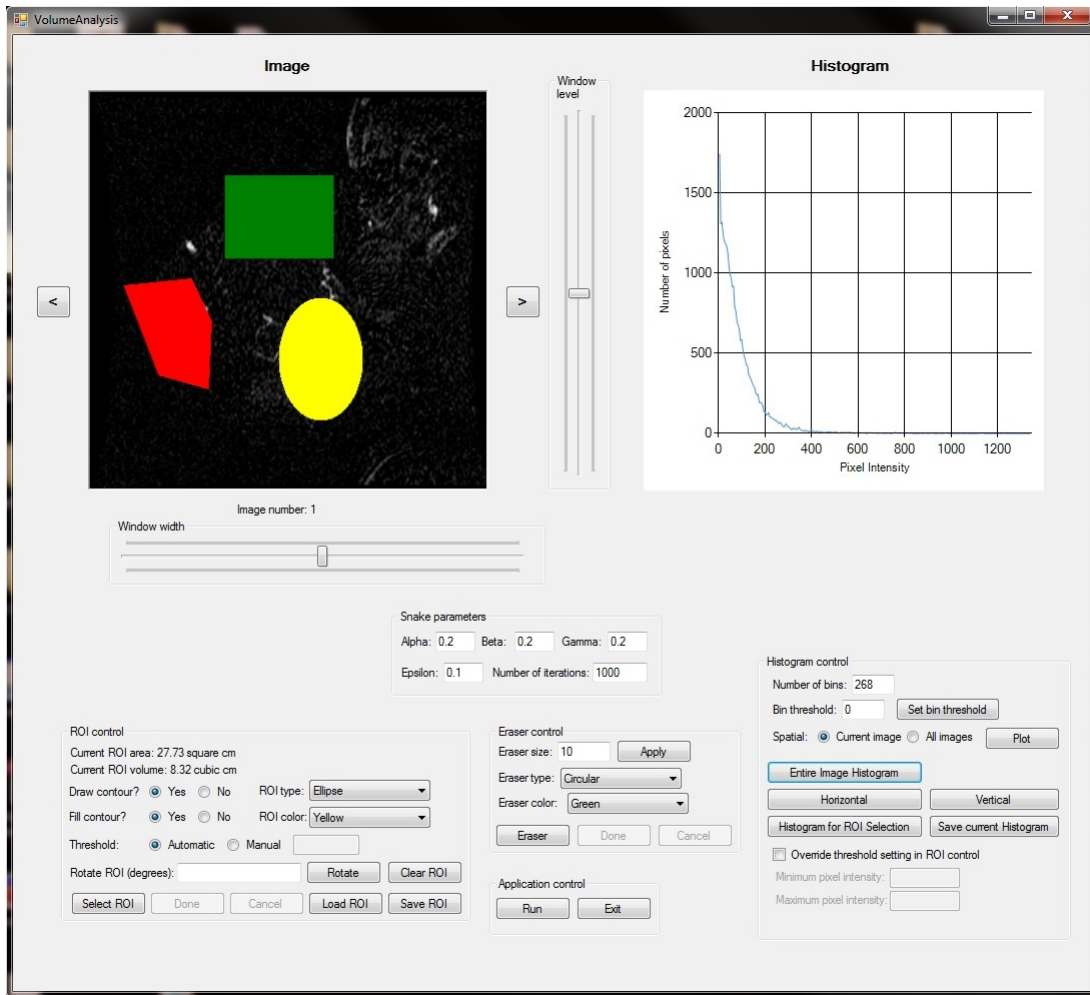


Figure 13: Various controls for group box ROI using the fill operation

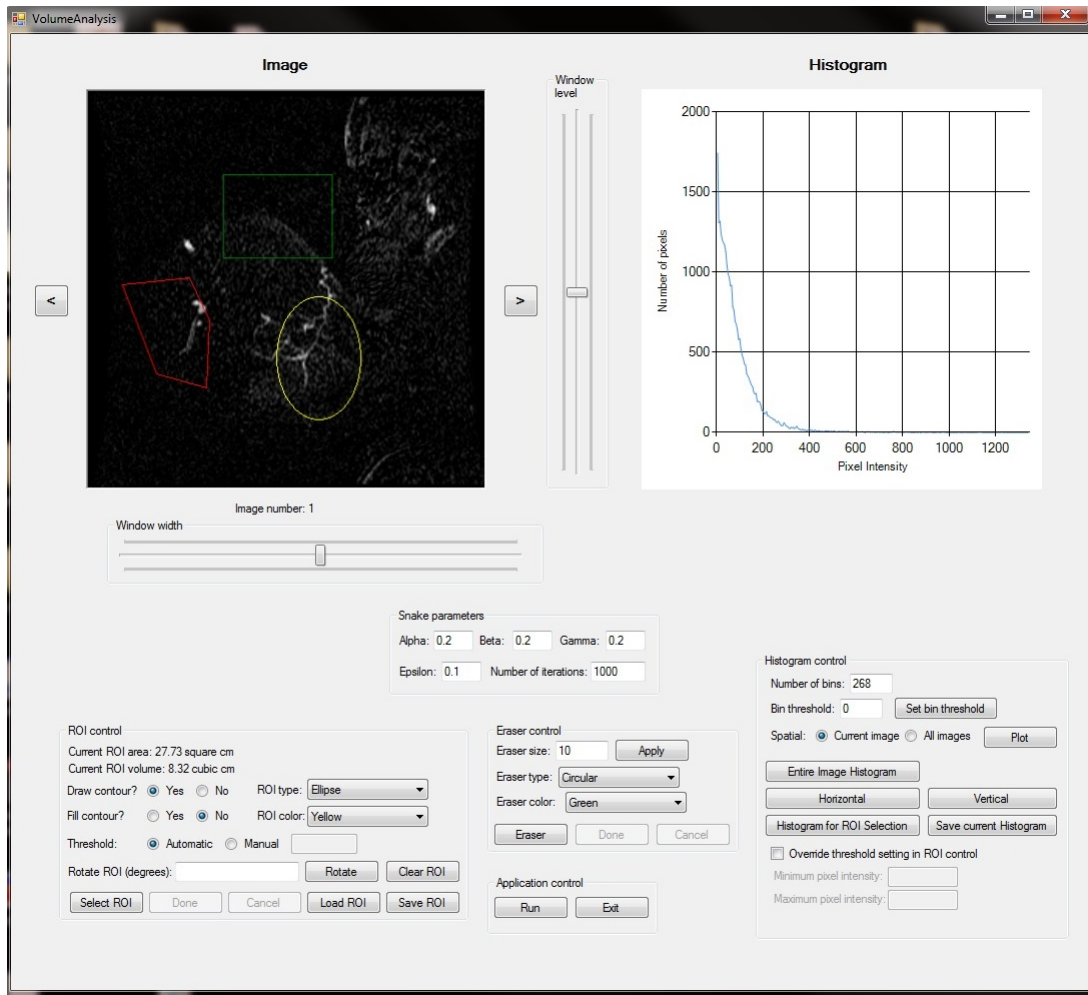


Figure 14: Various controls for group box ROI using the edge operation

The Eraser control group box

This group box contains the controls for eraser mode. We can change the size of eraser by specifying it in *Eraser size* textbox and clicking the *Apply* button. We can also choose the eraser shape to be rectangular or circular. The color of eraser can also be changed. If we want to enter the eraser mode, we need to press the *Eraser* button. Once we are in the eraser mode, we can erase the ROI using eraser. We can cancel this operation by clicking on *Cancel* button, or we can keep our modified ROI after erasing a part of it using *Done* button inside the eraser control group box.

The Application control group box

This group box contains the controls related to application. Once we are satisfied with our configured parameters, we can use the *Run* button to run our algorithm on each image. In order to exit this window and go to the *Volume Tool* window, the user can press the *Exit* button.

The Histogram control group box

This group box contains all the controls related to histograms. The *Bin threshold* value is used while plotting the histograms. The histogram, by default, is a plot of pixel intensity versus number of pixels. Only those pixel intensities are plotted on the histogram that have more than *Bin threshold* number of pixels. The *Number of bins* value is used for specifying the number of bins to be used while plotting the histogram. The **spatial histogram** for a single image or all the images can be plotted depending on the selection of user. For plotting the spatial histogram of single image, we plot the pixel intensity value inside the selected ROI. The spatial histogram for all the images is plotted by finding the average pixel intensity value inside the ROI for each image and plotting it against the image number. A plot for single image spatial histogram is shown in figure 15, and a plot for spatial histogram for all the images is shown in figure 16. The **horizontal histogram** is plotted by varying column number on x-axis and average pixel intensity inside each column of the ROI on y-axis. The **vertical histogram** is plotted by varying row number on x-axis and average pixel intensity inside each row of the ROI on y-axis. Figure 17 and 18 shows horizontal and vertical histograms respectively. We can plot the histogram only for the image part that is inside ROI. The histogram corresponding to the image part inside ROI is shown in figure 19. We can plot the histogram for entire image by using *Entire Image Histogram* button. We can save the currently displayed histogram by using the *Save current Histogram* button. The *Override threshold setting in ROI control* checkbox is used for overriding the values specified for thresholding in ROI control box. If this checkbox is checked, we can specify the minimum pixel intensity, and maximum pixel intensity that we want to allow after thresholding of the given image. Any pixel intensities outside this range is eliminated.

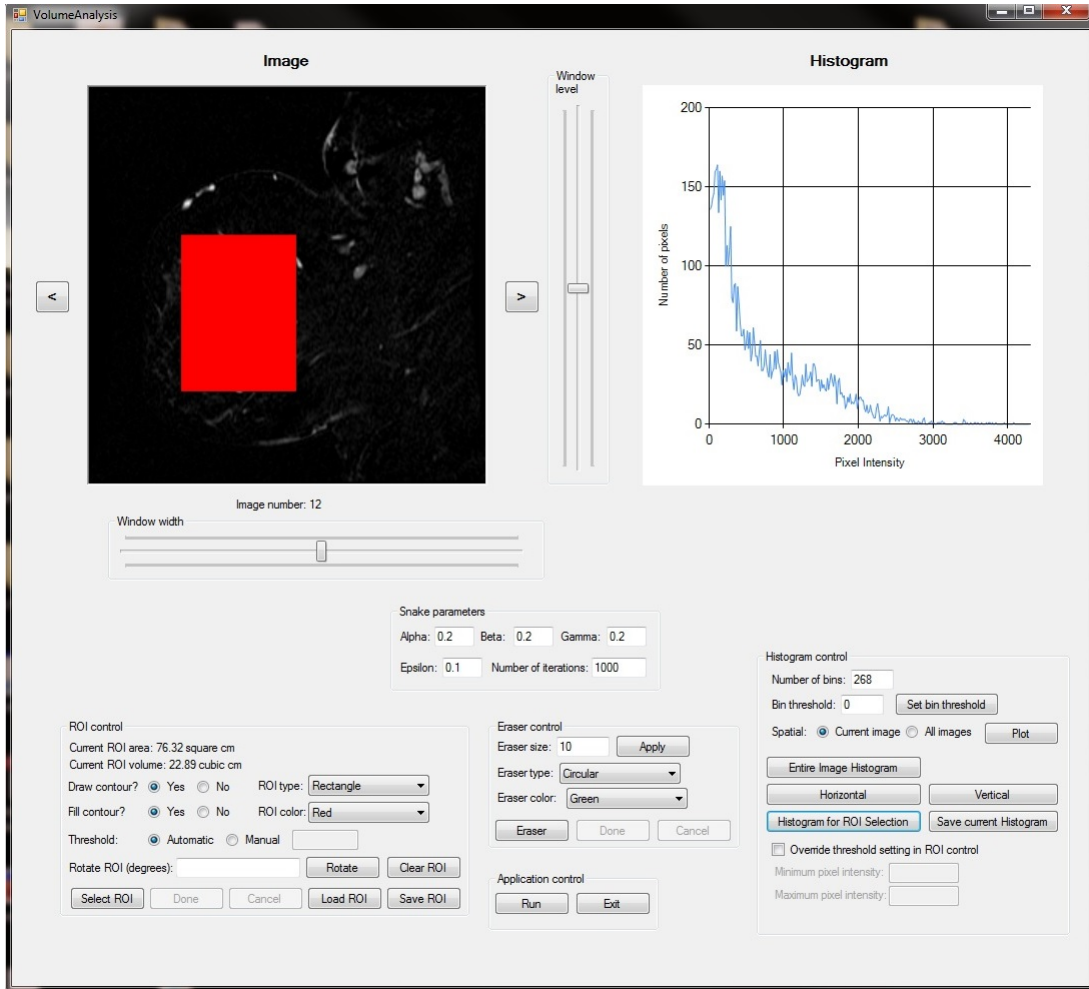


Figure 15: Spatial histogram for single image

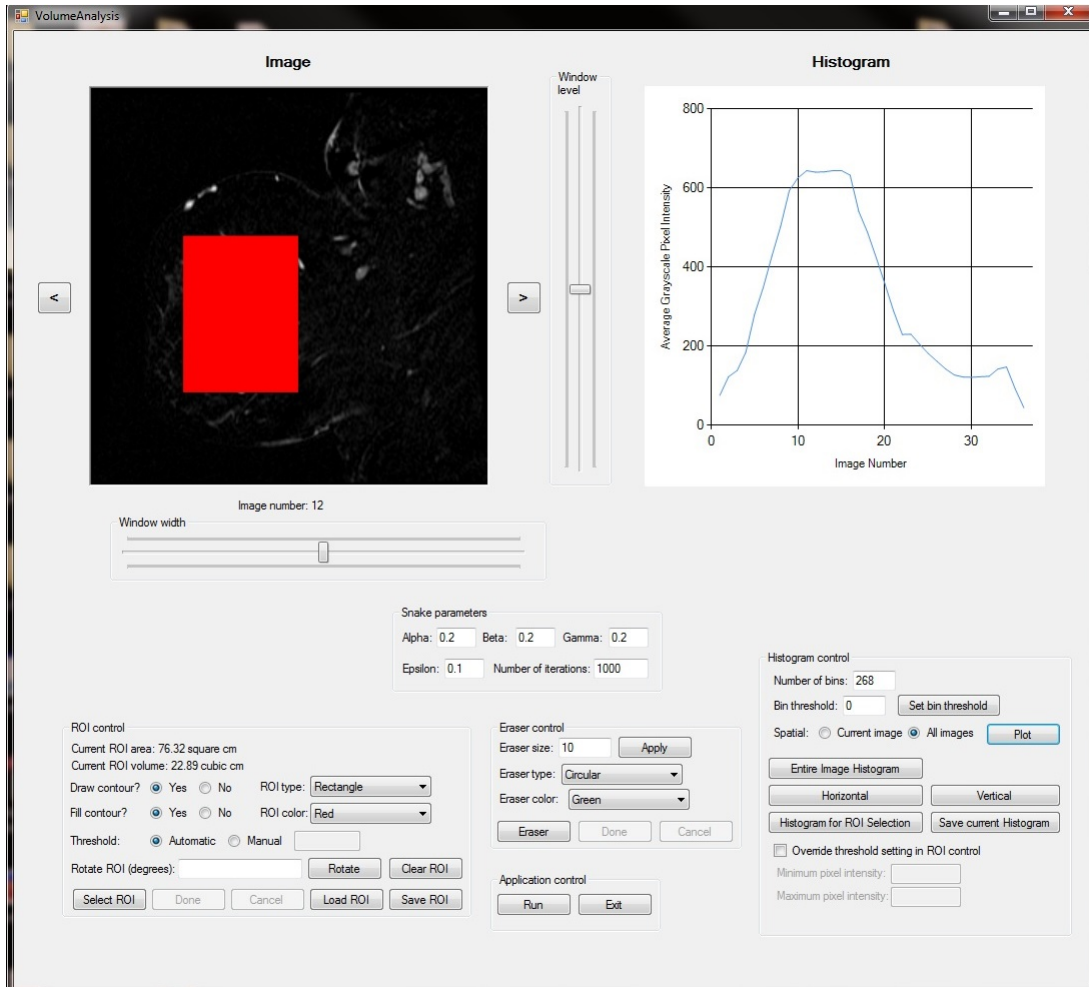


Figure 16: Spatial histogram for all the images

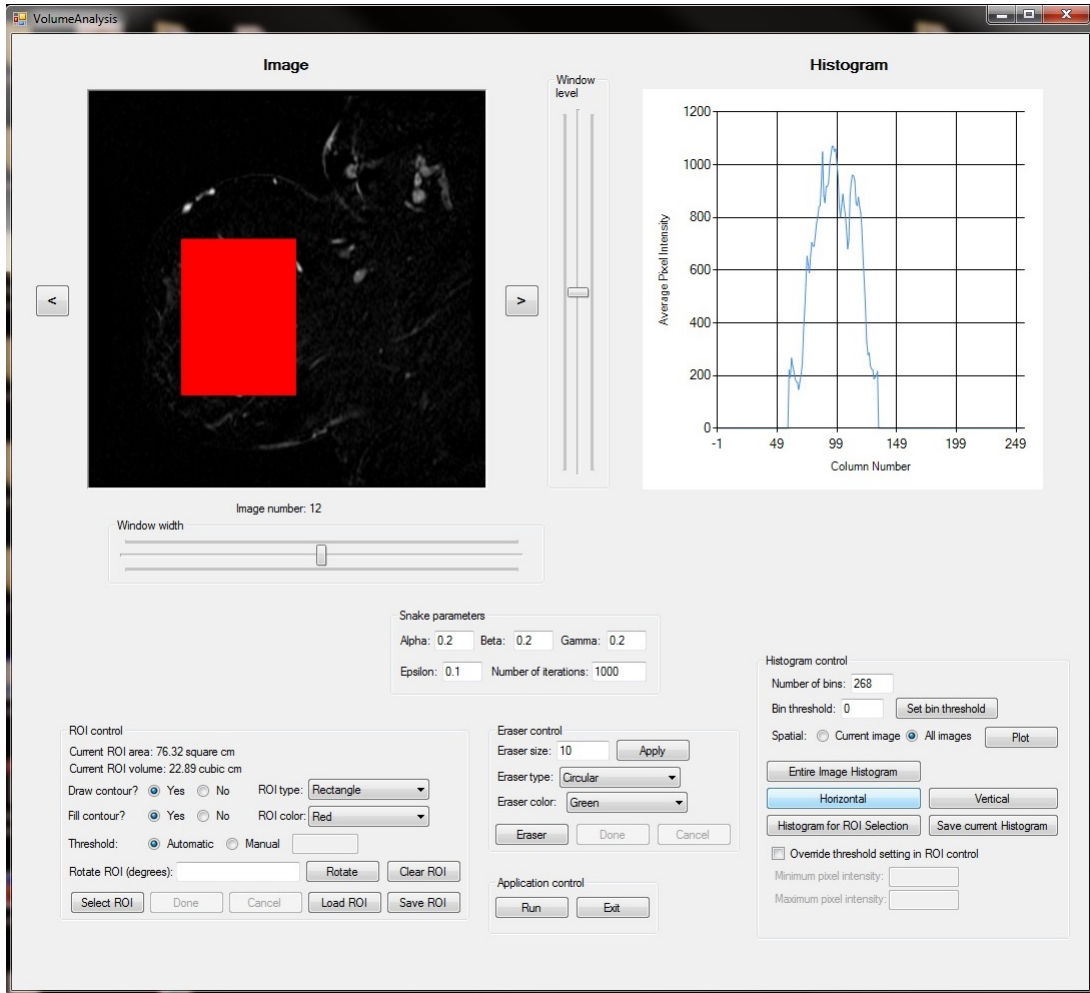


Figure 17: Horizontal Histogram

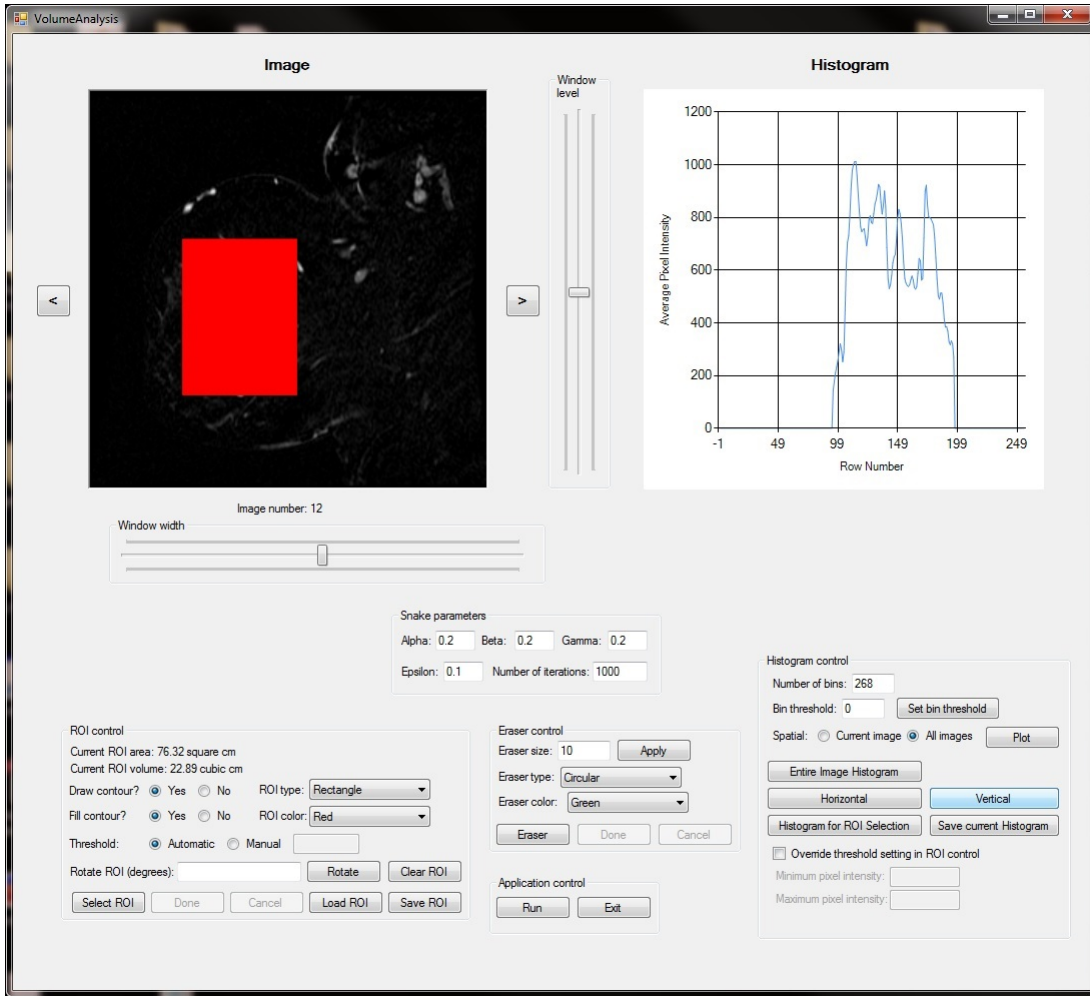


Figure 18: Vertical Histogram

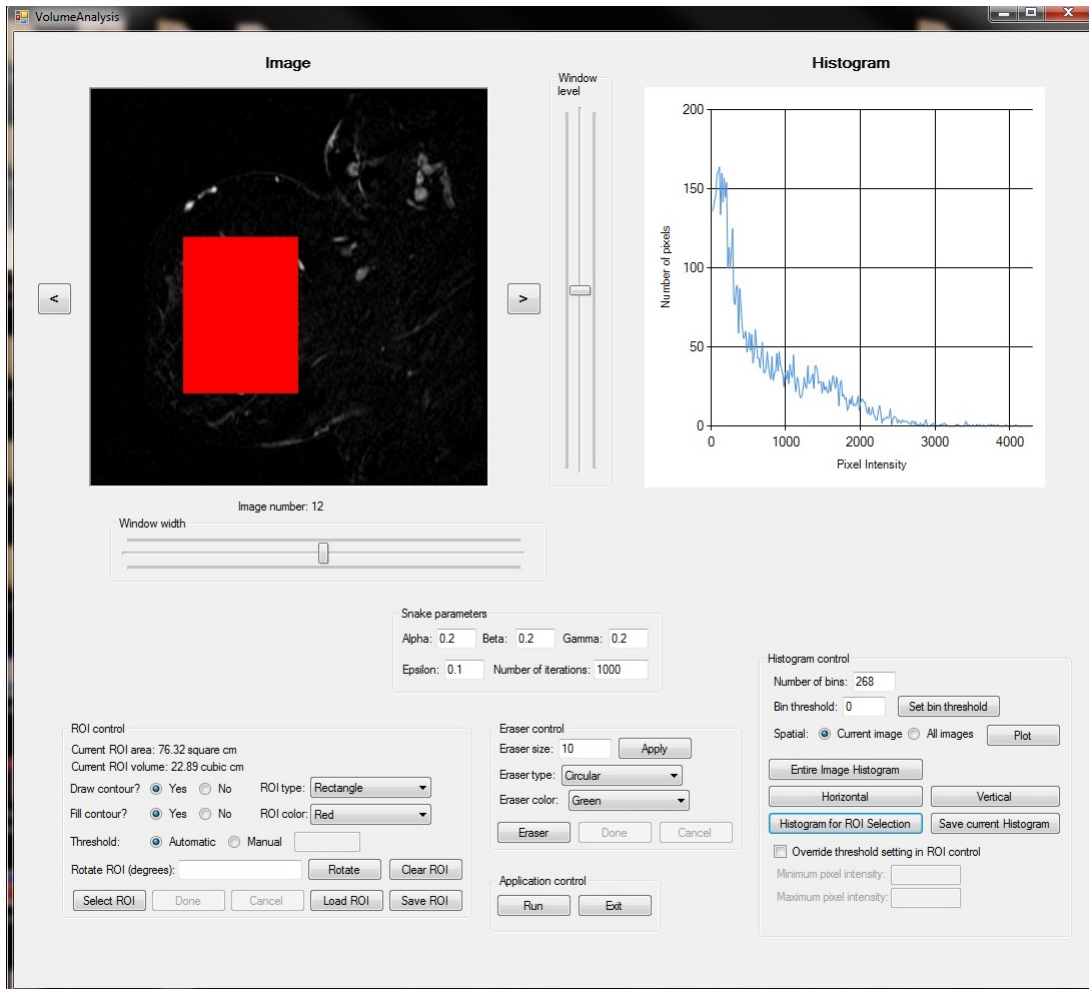


Figure 19: Histogram for image part inside ROI

3.3.3 The Volume Analysis Result Window

Once the algorithm is finished computing the tumor region and volume for each slice, this window is displayed for visualizing the results. This window is shown in figure 20. The current slice area, current slice volume, and total slice volume is displayed in this window. The *Previous* and *Next* buttons are used to navigate through the results. *Edit* button is used for editing the ROI that we obtain after running our algorithm. In edit mode, if *Grow ROI* option is selected, the ROI which we draw is added to the current ROI. If *Shrink ROI* mode is selected, the area that is inside the ROI we draw is removed from the ROI that is obtained by our algorithm. Both the operations are shown in figure 21 and 22 respectively. Once we are satisfied with the editing, we can save the ROI.

The *Load ROI* button is similar to that in Volume Analysis window. It is used for loading the ROI. Similarly *Draw custom ROI* can be used to draw custom ROI. The *Save custom ROI* and *Cancel ROI selection* buttons are used to save and clear the custom ROI selection. If we make any changes to the ROI, the program gives option whether we want to use the custom ROI in computing the total volume. The *Exit* button is used for exiting the current

window and returning to the *Volume Analysis* window.

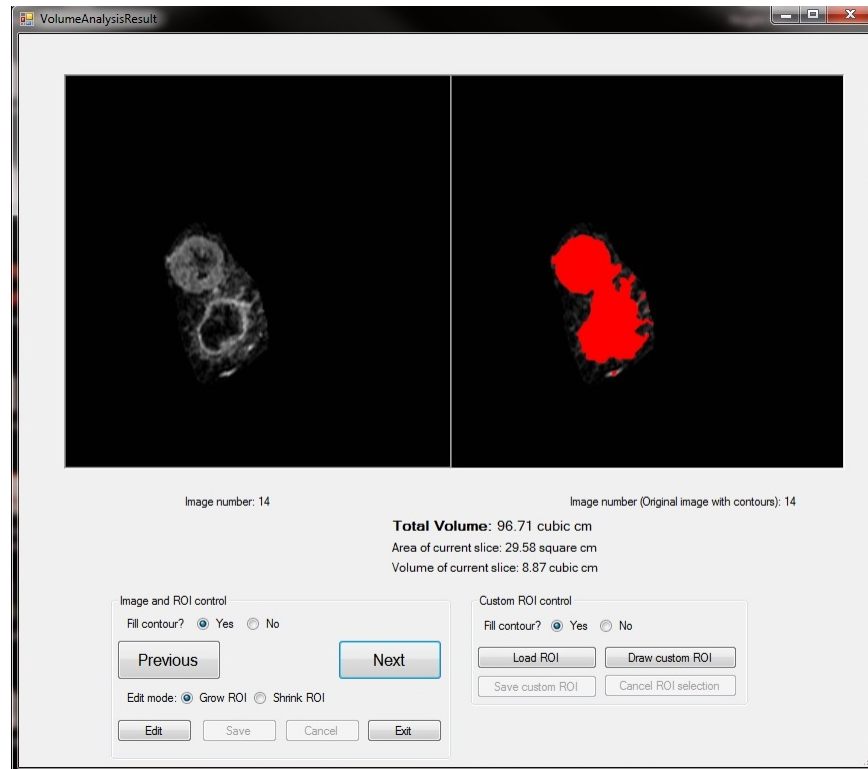


Figure 20: The Volume Analysis Result window

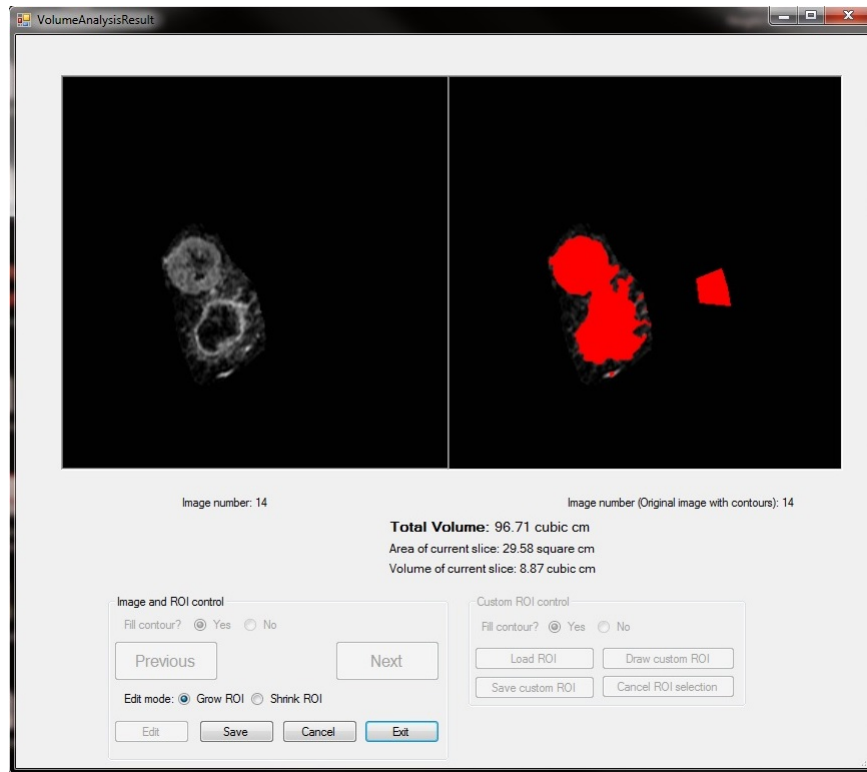


Figure 21: Editing ROI in grow mode

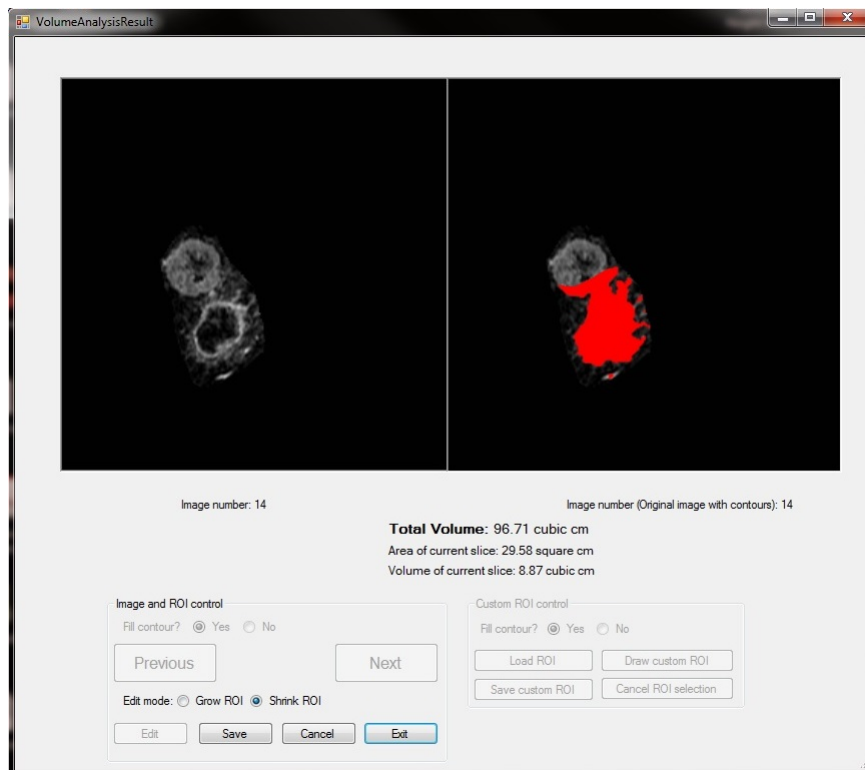


Figure 22: Editing ROI in shrink mode

4 Results

4.1 Optimization and training

The results for breast dataset, used for optimizing the tool, were obtained by setting minimum threshold value to 540. This value was chosen by seeing the normal histogram and spatial histogram of the images. The Snake parameters α , β , and γ were set to 0.2, 1.0 and 0.7 respectively. We experimented with the value of α , β and γ by choosing 0.2, 0.5, 0.7 and 1.0 as value for each of the parameters. For example, if α and β was set to 0.2 and 0.2 respectively, the value of γ was set to 0.2, 0.5, 0.7 and 1.0 and the results were recorded. The plot for (α, β, γ) vs percentage error is shown in figure 23. The x-axis in figure 23 varies from 1 to 64 where 1 represents $(\alpha = 0.2, \beta = 0.2, \gamma = 0.2)$, 2 represents $(\alpha = 0.2, \beta = 0.2, \gamma = 0.5)$ and so on. The percentage error is calculated by using the difference in calculated volume, and actual volume found by breast certified radiologist using planometric method. The actual breast volume found by breast certified radiologist used in optimizing the parameters was found to be 83.6 cubic centimeters. It can be seen from figure 23 that there are a few values of α , β , and γ , at which percentage error is less than 8.24%. Those values of the parameters were discarded because snakes were segmenting less volume. The ϵ value was set to 0.1. This value is used by Snake to determine whether it has converged or not. If the change in energy values between two consecutive iterations is less than ϵ , the algorithm terminates. We have set the number of iterations for Snake to 1000. This was chosen because even if the Snake is

not converged, we would get a good boundary around tumor, and we will get results in real time. The user can also change all these values using our tool.

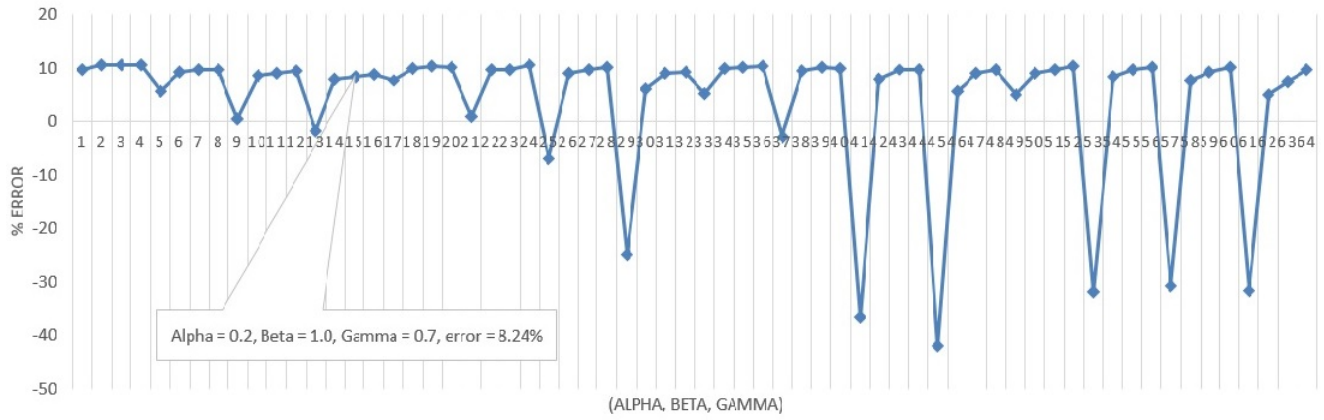
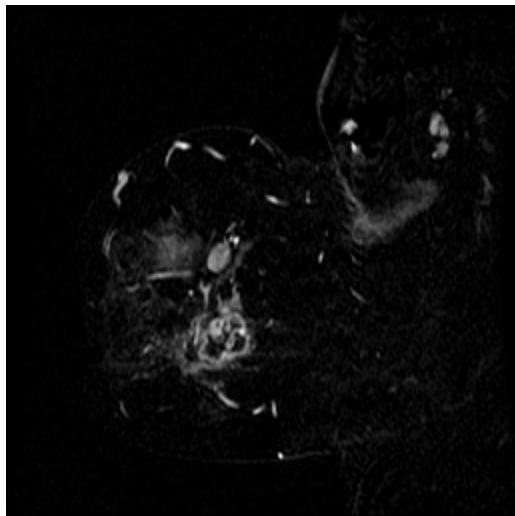
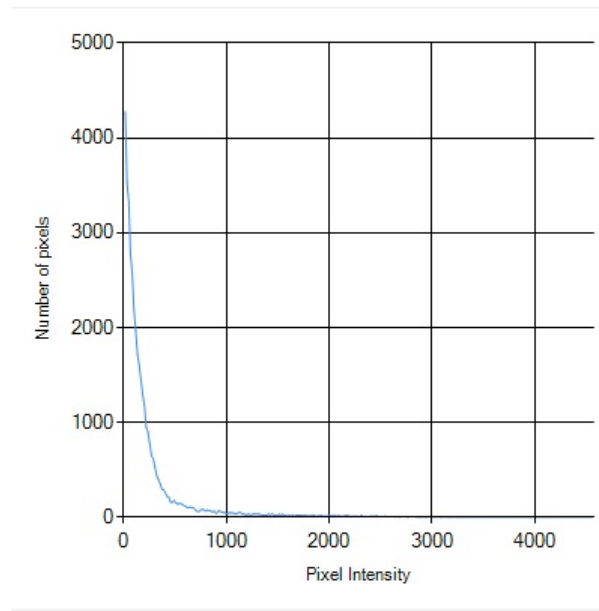


Figure 23: Tuning the parameters - (α, β, γ) vs *percentage error*

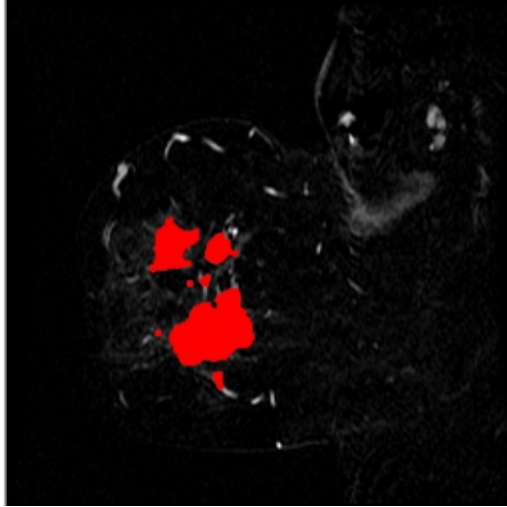


(a) Slice 8

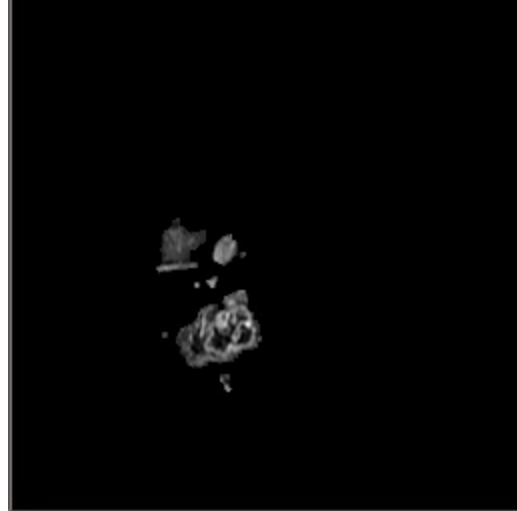


(b) Histogram for slice 8

Figure 24: Image and histogram for slice 8



(a) Full image with boundary



(b) Image part inside the boundary

Figure 25: Processed image for slice 8

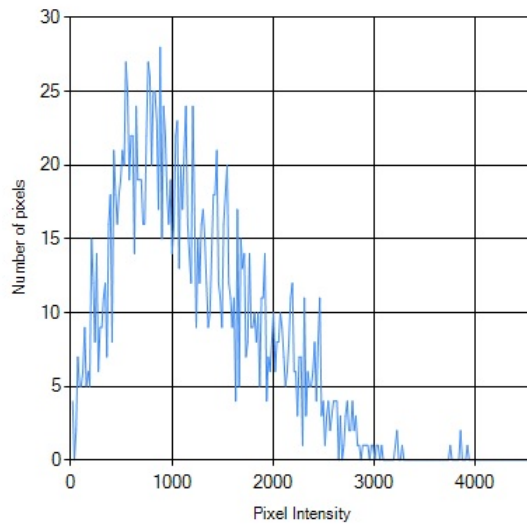
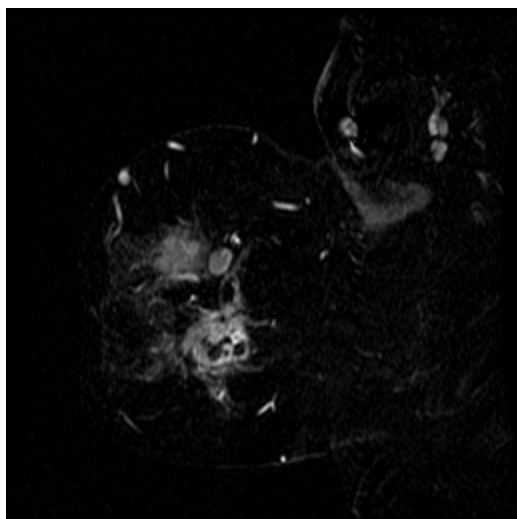
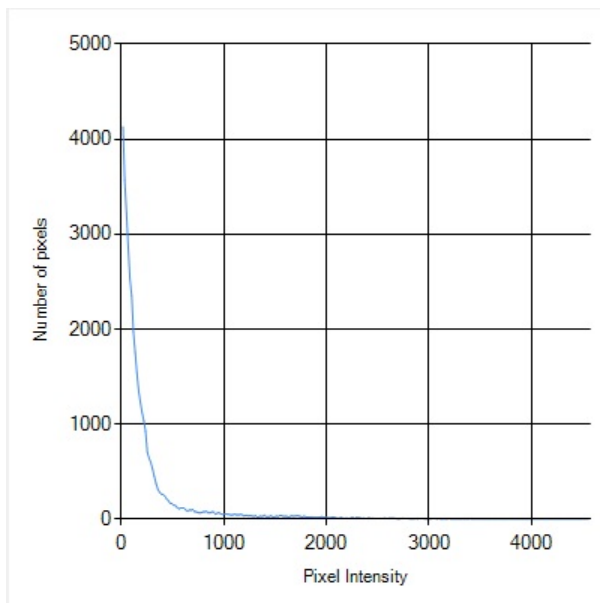


Figure 26: Histogram for image part inside boundary for slice 8

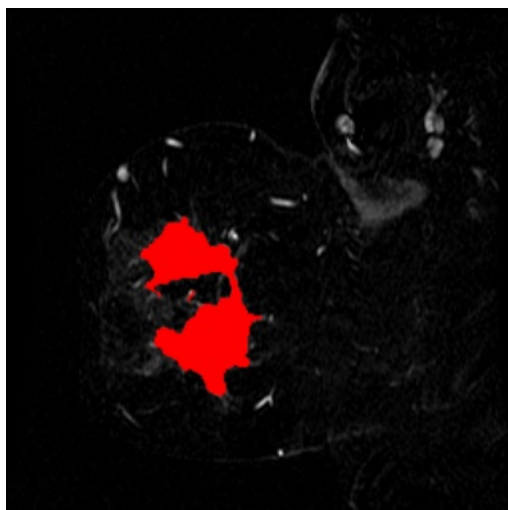


(a) Slice 9



(b) Histogram for slice 9

Figure 27: Image and histogram for slice 9



(a) Full image with boundary



(b) Image part inside the boundary

Figure 28: Processed image for slice 9

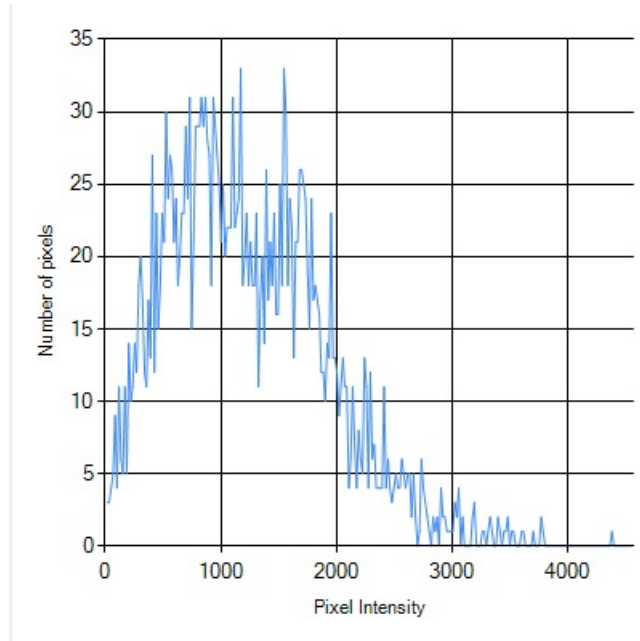


Figure 29: Histogram for image part inside boundary for slice 9

The given DICOM series of breast images for optimizing the parameters of the algorithm had 36 slices. We went through every image and saw that slice 6 through 20 shows tumor region, therefore these slices were chosen for running the algorithm and results were obtained for them. Figure 24 through 29 shows actual slice image, histogram corresponding to that slice, actual image with boundary found by our algorithm, image part that is inside the boundary found by our algorithm, and the histogram of image part found inside the boundary found by our algorithm for slice 8 and 9.

Figure 30 shows the ROI corresponding to which spatial histogram is plotted. Figure 31 shows the spatial histogram of all the 36 slices in the DICOM series of breast images. The image number is plotted on X-axis, while average gray-scale value inside the ROI corresponding to each image is plotted on Y-axis.

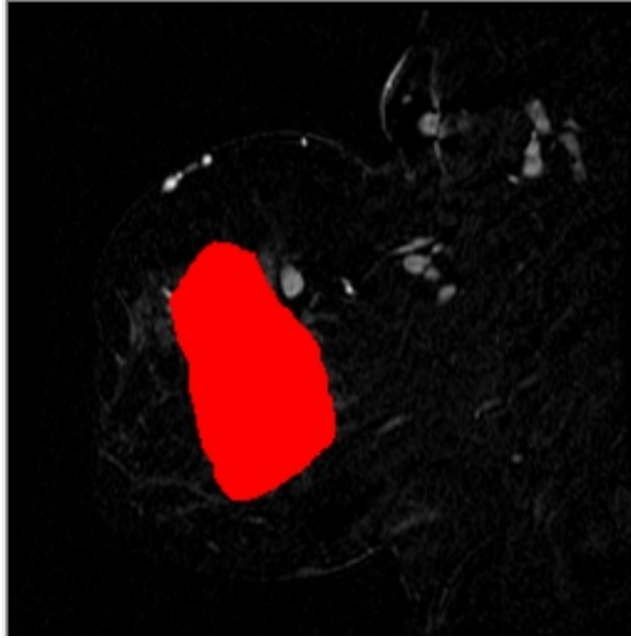


Figure 30: ROI corresponding to which spatial histogram is plotted on breast dataset

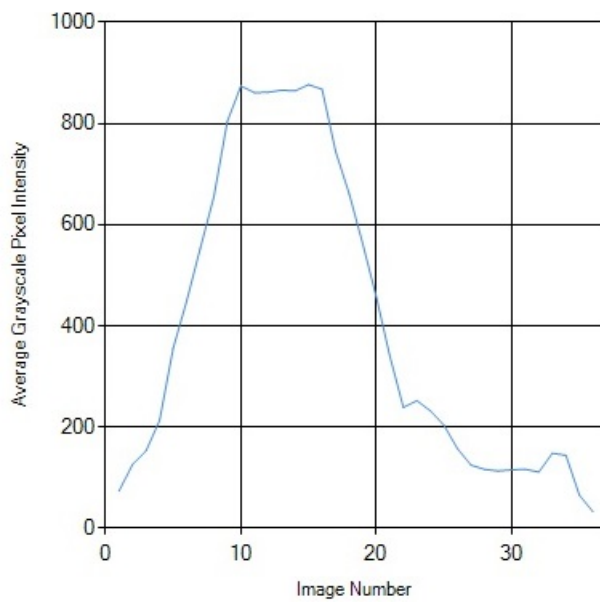


Figure 31: Spatial histogram for all the 36 slices in the DICOM series of breast images

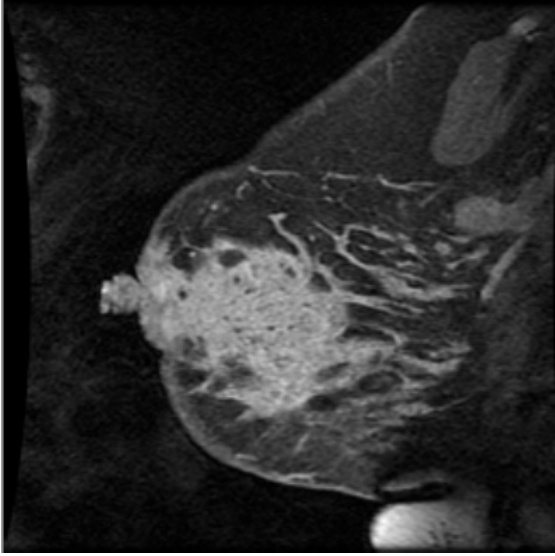
Table 2 shows the time taken for running our algorithm, area in square centimeter, and volume in cubic centimeter corresponding to each selected slice in breast images. The time taken for running the algorithm was computed by running the algorithm on each slice for 5 times, and then taking the average of those values.

Image No.	Time (milliseconds)	Slice Area (cm^2)	Slice Volume (cm^3)
6	23.94	7.45	2.23
7	3.41	12.19	3.65
8	3.57	15.95	4.78
9	4.36	22.97	6.89
10	6.98	26.49	7.94
11	5.07	27.67	8.3
12	4.33	30.34	9.1
13	3.75	30.28	9.08
14	4.66	28.44	8.53
15	5.08	26.89	8.06
16	5.37	25.73	7.72
17	9.94	15.08	4.52
18	3.28	16.71	5.01
19	3.74	11.01	3.3
20	2.75	4.6	1.38
Total	90.23	301.8	90.49

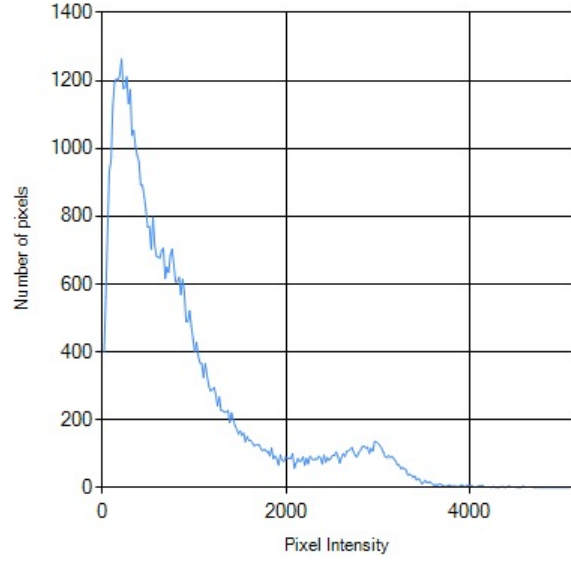
Table 2: Breast dataset - Algorithm timings, slice area, and slice volume for each slice

4.2 Post optimization results

Once the parameters were optimized, we applied our algorithm on five different datasets of breast images and calculated the volume of tumor corresponding to each dataset. Figure 32 through 35 and figure 37 through 38 shows actual slice image, histogram corresponding to that slice, actual image with boundary found by our algorithm, image part that is inside the boundary found by our algorithm, and the histogram of image part found inside the boundary found by our algorithm for slice X of Breast-Dataset1, and slice Y of Breast-Dataset5 respectively. Figure 36 shows slice 324 of Breast-Dataset5 after adjusting the window levels. We calculated the percentage error corresponding to each dataset by comparing the volume calculated by our algorithm with that calculated by breast certified radiologist using planometric method. The results corresponding to each dataset is shown in table 3. For each dataset, the algorithm was run for five times and average of all the five runs is used as final value for volume calculated by our algorithm.

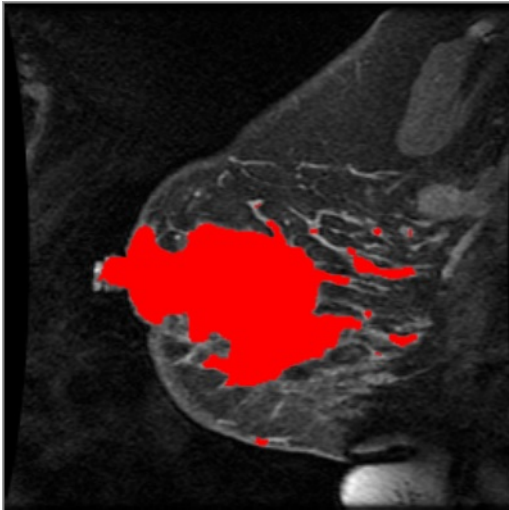


(a) Slice 103

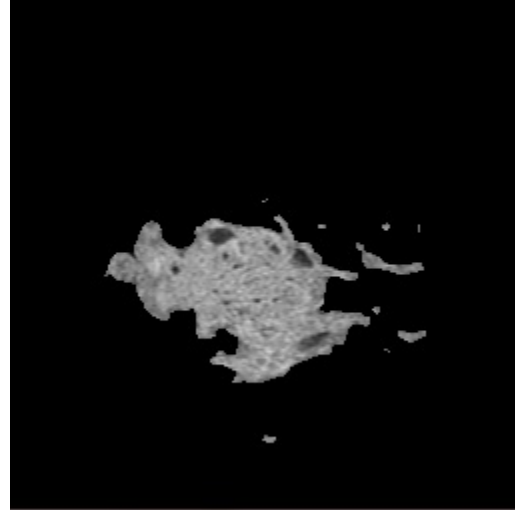


(b) Histogram for slice 103

Figure 32: Breast-Dataset1 - Image and histogram for slice 103



(a) Full image with boundary



(b) Image part inside the boundary

Figure 33: Breast-Dataset1 - Processed image for slice 103

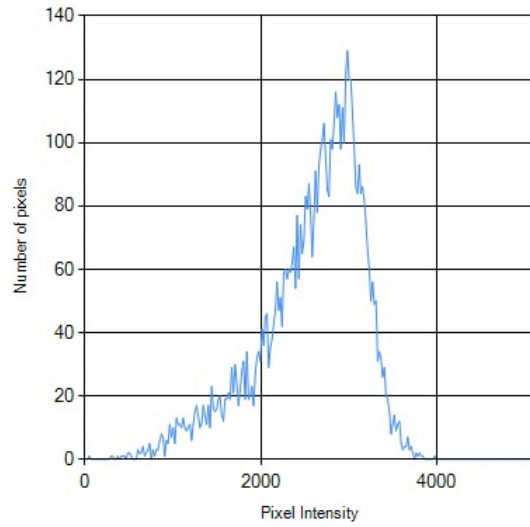
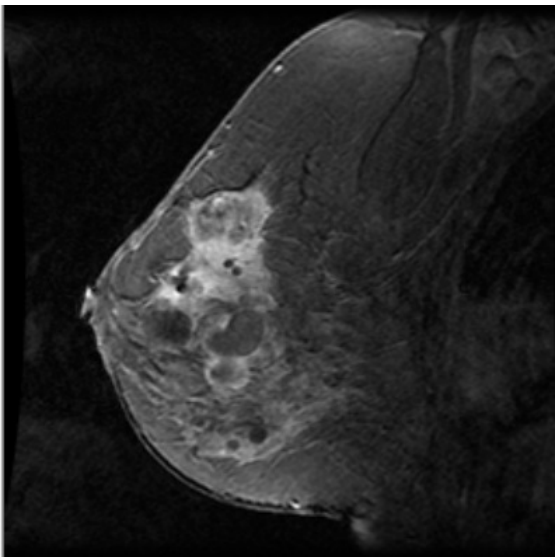
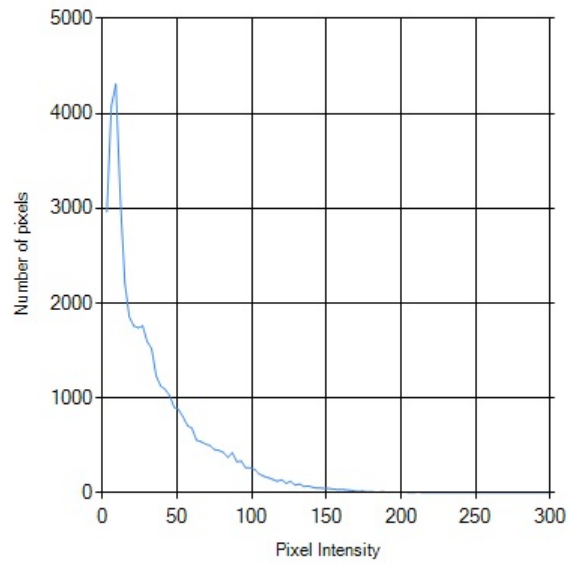


Figure 34: Breast-Dataset1 - Histogram for image part inside boundary for slice 103



(a) Slice 324



(b) Histogram for slice 324

Figure 35: Breast-Dataset5 - Image and histogram for slice 324

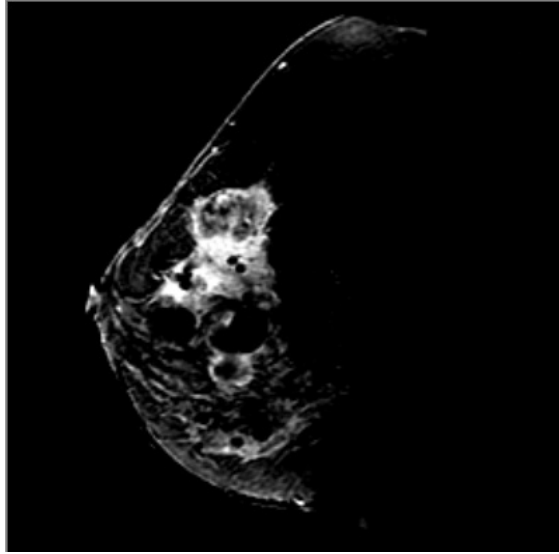
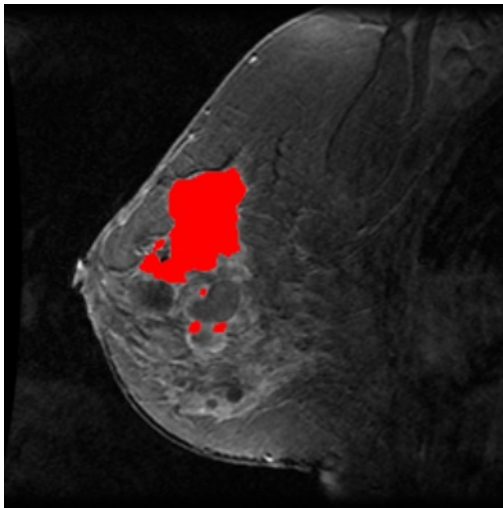
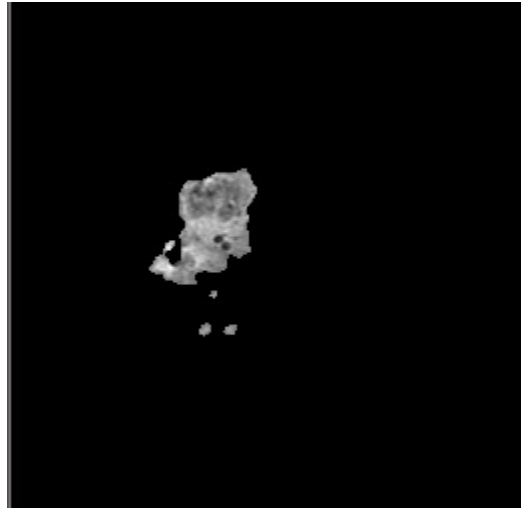


Figure 36: Breast-Dataset5 - Slice 324 after adjusting the window levels



(a) Full image with boundary



(b) Image part inside the boundary

Figure 37: Breast-Dataset5 - Processed image for slice 324

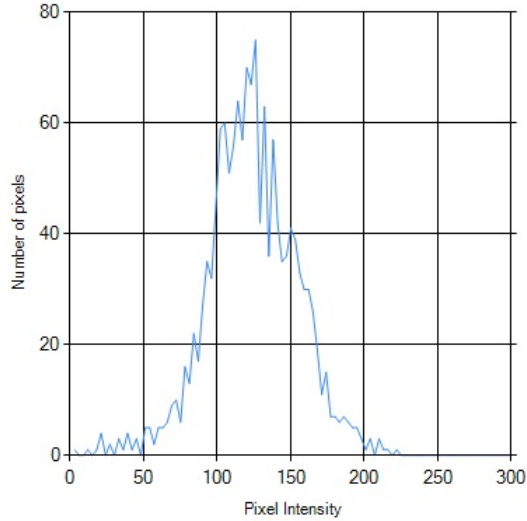


Figure 38: Breast-Dataset5 - Histogram for image part inside boundary for slice 324

Dataset	Calculated volume (in cc)	Ground truth (in cc)	Percentage error (%)
Breast - Dataset1	205.64	208.5	1.37
Breast - Dataset2	190.88	193.2	1.2
Breast - Dataset3	32.11	32.4	0.9
Breast - Dataset4	54.21	50	-8.42
Breast - Dataset5	11.37	12.5	9.04

Table 3: The results corresponding to various datasets

5 Discussion

We have used volume found by breast certified radiologist using planometric method as gold standard and used it in validating our results. It can be seen that our implemented algorithm is successfully able to segment the tumor region in all the slices. We were also able to calculate the area and volume of each slice. The total volume of tumor calculated by our algorithm on the breast dataset used for tuning the parameters of algorithm was found to be 90.49 cubic centimeter, and that by breast certified radiologist is 83.6 cubic centimeter. The percentage error in segmenting the volume of given breast dataset was found to be 8.24%.

Figure 23 shows how the percentage error is varied with respect to the parameters α , β and γ of snake. It can be seen that if the value of γ is less than that of α , the snake is not able to segment the volume correctly. The spikes in the graph represent the parameter settings in which γ is less than α . Using this data, we have set the parameter of α , β and γ to be 0.2, 1.0 and 0.7 respectively.

Table 2 shows time taken to segment the tumor, area of the tumor, and volume of the tumor corresponding to each slice. It can be seen from the table that our algorithm takes an average of 6 milliseconds in segmenting each slice. The total time for segmenting the volume was found to be 90.23 milliseconds.

In order to test our algorithm once the parameters were optimized, we tested it on various datasets. Breast-Dataset1 and Breast-Dataset2 were similar datasets, each containing large tumors in the given series of images. Our algorithm gave a very good estimate of tumor volume on these datasets. The percentage error in volume of tumor was 1.37% and 1.2% respectively. Our algorithm gave a very good estimate of tumor volume on Breast-Dataset3. The error in volumes was less than 1%. Breast-Dataset4 and Breast-Dataset5 were more difficult than the other datasets due to smaller volume of tumor and very less contrast difference between tumor and other tissues. From figure 35, it can be seen that tumor is not properly visible, and after adjusting the window levels using GUI, the tumor is visible as shown in figure 36. Our algorithm gave an error of 8.42% and 9.04% in calculating the volume of tumor on Breast-Dataset4 and Breast-Dataset5 respectively. However, the average error was low and was equal to 4.19% in finding the volume of the given images. From table 3, it can be seen that our tool was able to give a very good estimate of total volume of the tumor.

6 Conclusions

We have developed new method for segmenting volume from given radiological images. The results indicate good segmentation of tumor volume. In addition, our tool provides graphical interface for the user to interact with images. The GUI can be used in selecting the ROI, adjusting the parameters, and saving the results for future overlay.

The limitation of our algorithm is that it might fail to segment the volume of the tumor accurately if there is decreased contrast difference between tumor region and other tissues.

We are using thresholding techniques for finding the initial pixels. The problem with thresholding technique is that we only consider the intensity, and not any relationship between the pixels. Therefore it is not guaranteed that the pixels obtained by thresholding are contiguous. Another problem with thresholding is that if the noise in the image increases, there is possibility that we might include extraneous pixels, or we can miss the isolated pixels in the obtained region. Therefore, in future, our implementation of algorithm can be improved in these areas.

Acknowledgment

I owe many thanks to people who helped and supported me in completing this project.

I would like to deeply thank Dr. Michael Jacobs, my advisor and mentor. He have been really helpful in teaching me how to conduct a research, and mathematically formulate my ideas. He has taken initiatives to suggest ideas throughout my project work.

I express my thanks to Dr. Vladimir Braverman, my advisor, for guiding me and providing me with valuable encouragement for carrying out this project.

I'm grateful to Vishwa Parekh from Computer Science department for introducing me to Dr. Michael Jacobs and helping me in understanding the research expeditiously.

Finally, I would like to thank my parents Virendra Mohan Sinha and Munni Sinha for always being there for me and encouraging me in doing my research.

References

- [1] S. Taheri, S. H. Ong, and V. F. H. Chong. "Level-set Segmentation of Brain Tumors Using a Threshold-based Speed Function". *Image Vision Comput.* 28.1 (Jan. 2010), pp. 26–37. ISSN: 0262-8856.
- [2] P. Therasse, E.A. Eisenhauer, and J. Verweij. "{RECIST} revisited: A review of validation studies on tumour assessment". *European Journal of Cancer* 42.8 (2006), pp. 1031–1039. ISSN: 0959-8049.
- [3] Sanjay Saini. "Radiologic Measurement of Tumor Size in Clinical Trials: Past, Present and Future". *American Journal of Roentgenology* 176.2 (2001), pp. 333–334.
- [4] George Stockman and Linda G. Shapiro. *Computer Vision*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN: 0130307963.
- [5] William E. Lorensen and Harvey E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm". *Computer Graphics* 21.4 (1987), pp. 163–169.
- [6] Dzung L. Pham; Chenyang Xu; and Jerry L. Prince. "Current Methods in Medical Image Segmentation". *Annual Review of Biomedical Engineering* 2 (2000), pp. 315–337.
- [7] Y. Boykov and G. Funka-Lea. "Graph Cuts and Efficient N-D Image Segmentation". *International Journal of Computer Vision* 70.2 (2006), pp. 109–131.
- [8] Stanley Osher and James A. Sethian. "Fronts Propagating with Curvature-dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations". *J. Comput. Phys.* 79.1 (Nov. 1988), pp. 12–49. ISSN: 0021-9991.
- [9] R. Dubey et al. "Semi-automatic Segmentation of MRI Brain Tumor". *ICGST-GVIP Journal* 9 (2009), pp. 33–40.
- [10] L. He et al. "A comparative study of deformable contour methods on medical image segmentation". *Image and Vision Computing* 26 (2008), pp. 141–163.
- [11] Michael Kass; Andrew Witkin and Demetri Terzopoulos. "Snakes: Active Contour Models". *International Journal of Computer Vision* (1988), pp. 321–331.
- [12] G. D. Sullivan et al. "Active contours in medical image processing using a networked SIMD array processor". *Proceedings of the British Machine Vision Conference*. BMVA Press, 1990, pp. 70.1–70.6.
- [13] S. Rahnamayan, H.R. Tizhoosh, and M.M.A. Salama. "Automated Snake Initialization for the Segmentation of the Prostate in Ultrasound Images". *Image Analysis and Recognition*. Ed. by Mohamed Kamel and Aurlio Campilho. Vol. 3656. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 930–937. ISBN: 978-3-540-29069-8.

- [14] J. Ivins and J. Porrill. “Active region models for segmenting medical images”. *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*. Vol. 2. 1994, 227–231 vol.2.
- [15] Xianghua Xie and Majid Mirmehdi. “Rags: Region-aided geometric snake”. *IEEE Transactions on Image Processing* (2004), pp. 640–652.
- [16] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN: 013168728X.
- [17] Jean Serra. *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983. ISBN: 0126372403.
- [18] H. P. Ng et al. “Masseter Segmentation Using an Improved Watershed Algorithm with Unsupervised Classification”. *Comput. Biol. Med.* 38.2 (Feb. 2008), pp. 171–184. ISSN: 0010-4825.
- [19] Mariusz Jankowski. “Erosion, dilation and related operators”. *8th International Mathematics Symposium* (2006).
- [20] George Bebis. *Deformable/Active Contour*. URL: <http://www.cse.unr.edu/~bebis/CS791E/Notes/DeformableContours.pdf> (visited on 08/05/2014).
- [21] National Electrical Manufacturers Association. *DICOM*. URL: <http://medical.nema.org/dicom/geninfo/Brochure.pdf> (visited on 08/05/2014).