Image Processing for CT Scan of Lung Tumors Using Histogram for Thresholding and Grayscale Intensity Thresholding



Caleb Kennett

Justin Diep

ECE 6650 – Medical Image Processing

December 11, 2019

**TABLE OF CONTENTS**

**Abstract3**

**1. Introduction4**

**2. Background5**

**3. Method5**

**3.1 Method One5**

**3.2 Method Two7**

**4. Results10**

**4.1 Results and Discussion of Method One10**

**4.2 Results and Discussion of Method Two10**

**5. Results12**

**6. Conclusion12**

**References13**

**Appendix 14**

**ABSTRACT**

The application of image processing techniques with medical imaging has the potential to provide medical personals with improved images for better diagnosing. In the area of lung cancer, CT scans of the lungs will be enhanced by being entered into a MATLAB program and processed to segment the tumors away from the background. This is to make tumor easier to locate on the scan for the reader. There were two methods created to segment tumors. The first method utilized a histogram for creating a threshold to segment the tumor. The second method, after masking the lung region, used a grayscale intensity threshold, which was determined by trial, to extract the tumors and some of their dimensional properties.

**1. Introduction**

Lung cancer is a disease that causes the most deaths for both men and women [12]. Non-small cell lung cancer accounts for about 80 to 85 percent of those diagnosed with the disease, thus making it the more common while the other lung cancer type is small cell. Non-small cell lung cancer has 5 stages ranging from 0 to 4 including substages. The 5-year survival rate declines significantly when the cancer progresses into stage 2 with stage 2A and stage 2B having survival rates of 60 and 30 percent, respectively [12]. Given the low rates, it is imperative to detect cancer at earlier stages to receive treatment and increase survivability.

**2. Background**

There are a multitude of methods to diagnose for lung cancer non-invasively. Non-invasive procedures such as chest x-ray, computed tomography (CT), and magnetic resonance image (MRI) are scans of the body focused on the lungs. The quality of the scans varies with the chest x-ray providing the least pertinent scan of the body, and the MRI provides a detailed scan. CT scans will be the main source of data used for experimentation as it is typically the first scan a patient would receive for detecting lung cancer as it is more affordable than receiving an MRI scan and provides a higher quality than a chest x-ray. CT scans used in the project were located from an online database in DICOM format.

CT scans provide an image of the lungs and medical personnel would read the scan to search for abnormalities that may signify lung cancer. The scan of the lung includes blood vessels and possible tumors. The tumors may be confused for blood vessels or other anomalies, thus can result in misdiagnosing a patient. To prevent that possibility, a MATLAB program was created to process an inputted CT scan and implement imaging techniques to enhance the image. The enhanced image would allow for tumors to be easier discerned.

**3. Methods**

There were two methods formulated to enhance CT images of lungs with both utilizing MATLAB. Method one was the first attempt, and it took a rudimentary approach by utilizing a few image processing techniques to segment the tumor within the lung. Method two utilized Haralick features and spacial features from the extracted lung and tumor, which could later be used for classification. As stated earlier, the CT scans were obtained from an online database, and the image selected used for testing was arbitrary. In other words, the stage of the lung cancer was not taken into consideration prior to testing. The main objective was to test the functionality of the methods and its image processing techniques.

**3.1 Method One**

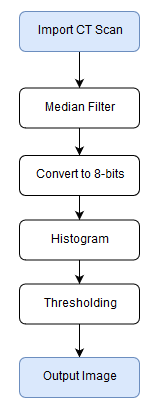
This method utilized a few image processing techniques and followed a similar methodology as a brain tumor extraction. The flow chart of the process can be seen below in figure 1, and the MATLAB code can be seen in appendix A. There was a presumption about the tumor within lungs having distinct characteristics and contrasts that will allow for segmentation of the tumor away from everything around it.

Following the flowchart of method one, the first step was to input a CT scan of a lung with a tumor presented. Images from CT may have noise, which is unwanted change in the pixels that decreases image quality and may affect some of the image processing techniques [2], [4]. To account for noise that may appear, a median filter will be applied. A median filter is a nonlinear operation that removes unwanted noise and preserving the edges. The filter does so by going to each pixel and converting it to the median between that pixel and its neighboring pixels.

After the median filter, the image will be converted into 8-bits. The conversion into 8-bits will change the range of pixels from 0 to 255. CT scans are in DICOM format; thus, the pixels have a range of in the thousands and makes it difficult to use image processing techniques. This is done by first applying the “mat2gray” MATLAB function to convert the image into grayscale of the pixels becoming 0s and 1s. The grayscale image is then multiplied by 255 and the “uint8” function is applied to complete the 8-bit conversion.

The histogram of the image is attained next with the “imhist” function. The histogram will show a bar graph of the amount of each pixel value of the range from 0 to 255, the colors of black to white. First and foremost, an image of the tumor zoomed in upon is saved, and then, a histogram of its pixels can be attained. From the histogram the contrast level of the tumor’s pixels can observed and used to create a threshold for the next step.

Thresholding is a method that partitions a designated part of an image away from the background. In other words, it can segment a desired object such as a tumor. After attaining the contrast level of the tumor from the histogram, a threshold can be created by selecting a range of pixel values relative to the tumor’s contrast. This can be used to create a bandpass filter to apply to the image. The bandpass filter will keep all the pixels within the selected range and exclude everything else outside of it, thus outputting an image of the segmented tumor.



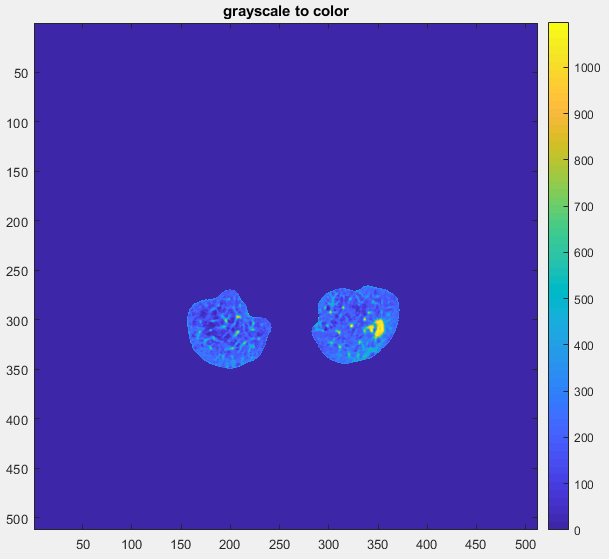
**Figure 1: Method One Flow Chart**

**3.2 Method Two**

Axial reconstructed slices of the lung are the designed input images for this algorithm. In this case DICOM image files were used, as there was a large database for these. Four steps can be used for processing image: preprocessing, masking, segmentation, and classification. Preprocessing includes median filtering. Masking of the lung region includes creating binary image and morphological operations. Segmentation includes thresholding and masking the tumor from the lung, and Classification would include collecting features and interpreting them [5], [6].

A mask was created which highlights the lung region of the computed tomography (CT) image. Since lung tumor within the lung region generally has higher cell density, the gray-level intensity will be higher; using this knowledge, a threshold was used which shows only pixels above a 537 grayscale intensity: this was determined by trial with the data set of the input CT images. The image was determined to have a tumor if the threshold condition was met and the pixel density was greater than 50 pixels [7].

To create the mask of the tumor, first the image was converted to binary black and white image by Otsu's thresholding technique using the MATLAB's built in function “imbinarize”. Next the image’s colors were inverted, and the white borders were removed using “imcomplement” and “imclearborder” functions respectively. Then the dilation and opening processes were used to make the lung area more prominent. Now the mask can be used for the grayscale image. These steps for creating mask can all be implemented and easily optimized using the “imageSegmenter” tool in MATLAB.

Using the masked grayscale image, the different intensity values can be viewed; as it can be seen from figure 2, the tumorous area has higher gray-level intensity, and this can be used to segment the tumor from the lungs. Using a threshold value, the image pixel values below this were set to zero, and everything above set to one; this new mask was then used on the gray-scale image to segment out only the tumor. This threshold value was determined by increasing the threshold value on known images without tumor until no tumor result was shown. 

**Figure 2: Grayscale to Color Graph**

Another step in the final masking process was to remove pixel groups with density less then 50 pixels, as this relates to 2.5cm, as seen in (reference Patel). In general, malignant lung tumors have diameter greater than 2.5 cm [7]. After final thresholding of grayscale intensity and removing smaller pixel groups, the lung image was classified as either having tumor(s) or not having a tumor using an ‘if’ statement in the MATLAB code.

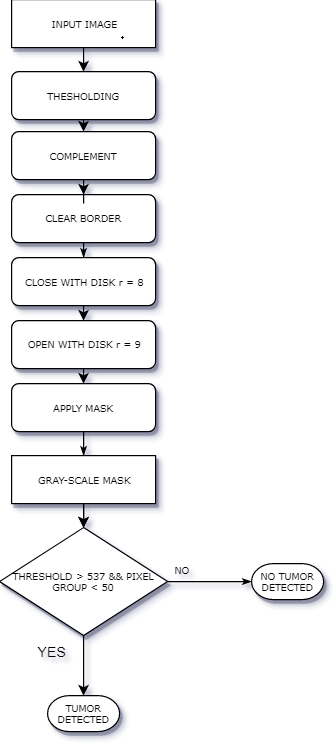
Although Classification was not stated in the original goal of the project, an attempt was made, and later future work could be done to improve this step. In order to classify the images as tumorous or non-tumorous, features of the image need to be acquired; the more features, the more likely the classifier will work. Two main different types of features were extracted from the images in this project: Spatial features and Harralick features.

Spatial features can be useful in determining the stage of the tumor: as the diameter determines the stage. These features can also be useful in determining the size of the tumor relative to the lung region [8]. Different spatial features were calculated from the tumor including area, perimeter, eccentricity, lung area, equivalent diameter, solidity, irregularity index. These features are described as follows: The area is calculated using the number of white pixels masking the tumor; Perimeter is approximated using MATLAB's edges function. Eccentricity is (4\*Area\*pi)/(Perimeter.^2). Lung area or Convex Area gives the number of pixels in convex image of the ROI. Perimeter is the distance between each adjoining pair of pixels around the border of the ROI. Solidity is the proportion of the pixels in the convex hull that are also in the ROI. Irregularity Index is the measure of how closely the shape of an object approaches that of a circle similar to eccentricity.

The Equivalent diameter can be useful to determine what stage tumor is in. The Equivalent diameter is the diameter of a circle with the same length as the tumor. This is currently in dimensions of pixels, in order to convert to cm, the Equivalent diameter is multiplied by 2.5/50 [7]. The area, perimeter and eccentricity were displayed as the output of this code because these features were used also in other papers [8], [9], [10].

After masking the lung areas, the gray-level co-occurrence-matrix (GLCM) was calculated for extracting Haralick features. The GLCM is a square matrix with dimensions equal to the number of gray levels in the image; each element is generated by counting the number of times a pixel with certain value “a” is adjacent to pixel with value “b” and dividing the entire matrix b the total number of those comparisons. The matrix is therefore probabilities of these pixels being found adjacent.

The Haralick features are 14 statistics that can be calculated from the co-occurrence matrix with the intent of describing the texture of the image [5]. In that reference 7 statistics were used: energy, entropy, contrast, homogeneity, maximum probability, cluster prominence and inverse difference moment. The “haralickTextureFeatures” function written by Rune Monzel was used to calculate all 14 Haralick features from the GLCM [11]. According to reference [6] these features can be used to train Machine learning algorithm to classify if the lung image has a tumor.



**Figure 3: Flow Chart of Method Two**

**4. Results**

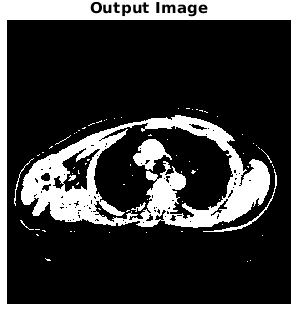
The results of method one and method two differentiated from each other. The process of method one seemed like a practical approach in theory, however; there were complications encountered during the construction of the MATLAB program and experimentation. Method two was the more viable method. Method two was able to achieve segmentation of the tumor and, in addition to that, it also classified the inputted CT images by determining if there is a tumor present.

**4.1 Results and Discussion of Method One**

The code for method one operated successfully but did not give the desired result. The output image of the result can be seen below in figure 4. The program did not segment the tumor as the surrounding area was included. There were multiple images tested, and the outcome resulted in a failure to segment the tumor.

It was initially assumed the tumor would have its own distinct pixel contrast that would allow for a threshold to be created to segment it from the surroundings. This assumption was proven to be ineffective after trial and error from creating the program and testing the images. This was due to the tumor having a similar contrast to other objects and background, thus making it difficult to narrow down the threshold range of the tumor.

With no distinct pixel contrast, method one was an impractical approach to segmenting a tumor. It should be noted that CT scans will vary from one person to another. In other words, each CT scan inputted must have the tumor studied heavily with its histogram to gauge the threshold for only that specific image. Thus, another flaw for method one is its dependence upon human interaction to adjust the threshold for segmentation.



**Figure 4: Output Image**

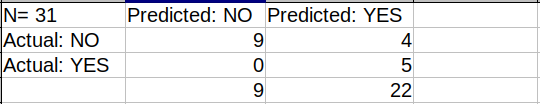
**4.2 Results and Discussion of Method Two**

Segmentation of tumors was the main purpose of the project classification would be something done more in future work, however, to determine if the segmentation was successful some classification was required. Multiple tumors were successfully segmented from the input CT scan, as well as, images which did not have tumors were successfully determined as such. When classification was done, over-classification of lung tumors happened: some regions within the lungs have gray-levels which are as high as threshold used to segment tumors, yet not tumors. In order for this to be resolved, a more detailed labeling of the images would be required, and possibly someone with experience looking at CT images to classify tumors.

31 images were tested; 9 images were determined to have no tumors, and 22 were determined to have tumors. There were 9 true negative results and 0 false negative. At least 4 false positive results and at least 5 true positive. The “at least” occurs with images that are not known to have tumor yet give result of having a tumor. This is one possible problem of using binary classifier opposed to using fuzzy logic classifier.

There were no False Positive results, which was good; given that this system was designed for use alongside of the diagnosing doctor, over-classification is preferable; this way cancer has less chance of going unnoticed.

*9images with no tumor and 22 with tumors:* 4 FP (or more); 9 TN; 5 TP (or more); 0 FN



**Table 1: Method Two Results**

The following are properties of the confusion matrix; other properties are not shown because they cannot be calculated properly. All values are the minimum because there may be more TP and FN if the database had each image labeled clearly.

Accuracy: (TP+TN)/total = (5+9)/31 = 0.4516

Error Rate: (FP+FN)/total = (4+0)/31 = 0.1290

Precision: TP/predicted yes = 5/22 = 0.2273

Prevalence: actual yes/total = 5/31 = 0.1613

**6. Conclusion**

Between the two methods presented, method two was more viable as it was able to segment the tumor and classified whether an image had a tumor present or not. Method one has the possibility to segment a tumor if it were to have distinct pixels but would be inefficient as it would require the threshold to be changed for each CT scan inputted. Method two can still be improved in a couple ways. The mask can be improved to be capable of more accurately masking the more unusual lung images; more reliable segmentation method, thresholding method has problem of over classifying. The stage of the cancer may be determined using some of the dimensions of the tumor, Masking of the tumors within the lung was done for the purpose of further processing features to determine if the tumor was malignant or benign. Overall, the project was a success in image processing procedures to extract a tumor from a CT lung scan.

**References**

[1] Devkota, B., Alsadoon, A., Prasad, P.W.C., Singh, A. K., Elchouemi, A. (2017). Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction. *Procedia Computer Science*, 115-124.

[2] Hambal, A. M., Pei, Z., and Ishabailu, F. L. (2015). Image Noise Reduction and Filtering Techniques. *International Journal of Science and Research*.

[3] Patil, R. C. and Bhalchandra, A. S. Brain Tumour Extraction from MRI Images using MATLAB. *Internation Journal of Electronics, Communication & Soft Computing Science and Engineering*, *2*(1).

[4] Kumar, N. and Nachami, M. (2017). Noise Removal and Filtering Techniques used in Medical Images. *Oriental Journal of Computer Science & Technology*, 10(1), 103-113.

[5] Moffy Crispin Vas, Amita Dessai (2017). Classification of Benign and Malignant Lung

Nodules Using Image Processing Techniques. International Research Journal of Engineering and Technology (IRJET). Volume: 04 Issue: 04 | Apr -2017

[6] Moffy Vas, [Amita Dessai](https://ieeexplore.ieee.org/author/37086452983) (2017). Lung Cancer Detection System Using Lung CT Image Processing. *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA).* IEEE Xplore: 13 September 2018

[7] Bindiya Patel, Dr. Pankaj Kumar Mishra, Amit Kolhe (2018). Lung Cancer Detection on CT

Images by Using Image Processing. *International Research Journal of Engineering and Technology (IRJET). Volume: 05 Issue: 04 | Apr*-2018

[8] Anjali Kulkarni, [Anagha Panditrao](https://ieeexplore.ieee.org/author/37397246300) (2014). Classification of Lung Cancer Stages on CT Scan Images Using Image Processing. *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies.* IEEE Xplore: 26 January 2015

[9] Aniket Gaikwad, Azharuddin Inamdar, Vikas Behera (2016). Lung Cancer Detection Using

Digital Image Processing On CT Scan Images. *International Research Journal of Engineering and Technology (IRJET). Volume: 0*3 Issue: 04 | Apr-2016

[10] Preeti Katiyar, Dr. Krishna Singh (2017). Lung Tumor Detection and Segmentation in CT

Images. *International Journal of Innovations & Advancement in Computer Science (IJIACS).* Volume 6, Issue 7. July 2017

[11] Rune Monzel (2019). haralickTextureFeatures (<https://www.mathworks.com/matlabcentral/fileexchange/58769-haralicktexturefeatures>), MATLAB Central File Exchange. Retrieved December 11, 2019.

[12] Everything You Need To Know About Lung Cancer. Retrieved from <https://www.healthline.com/health/lung-cancer>

**Appendix**

**A) Method One Code**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lung Tumor Segmentation - Method One

clc

clear

close all

path = '000000.dcm';

% read the dcm image file

img1 = dicomread(path);

img = img1;

figure

imshow(img1,[]);

title('CT Scan of Lungs with Tumor')

% Image conversion

img1 = uint8(255\*mat2gray(img1));

% Medium filter applied for noise

img1 = medfilt2(img1);

figure

imshow(img1,[])

title('Medium Filtered Image')

% Zoomed-up image of tumor

I = imread('tumor.PNG');

figure

imshow(I)

title('Zoomed-up Image of Tumor')

% Histogram of Tumor

figure

imhist(I)

% Min and Max pixel values of tumor for threshold

thresholdmax = 85;

thresholdmin = 80;

n = length(img1);

% Threshold/Bandpass filter

for a=1:n

for b=1:n

if img1(a,b) < thresholdmax;

img1(a,b) = 1;

else img1(a,b) > thresholdmax;

img1(a,b) = img1(a,b);

end

if img1(a,b) < thresholdmin;

img1(a,b) = 0;

else img1(a,b) > thresholdmin;

img1(a,b) = 1;

end

end

end

% Output Image

figure

imshow(img1,[])

title('Output Image')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**B) Method Two Code**

% read the dcm image file

img = dicomread(path);

% this is how to show the dcm image

imshow(img, [])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Threshold image - global threshold

X =img;

BW1 = imbinarize(X);

% Invert mask

BW = imcomplement(BW1);

% Clear borders

BW = imclearborder(BW);

% Fill holes

BW = imfill(BW, 'holes');

% Close mask with disk

radius = 8;

decomposition = 0;

se = strel('disk', radius, decomposition);

BW = imclose(BW, se);

% Open mask with disk

radius = 9;

decomposition = 0;

se = strel('disk', radius, decomposition);

BW = imopen(BW, se);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% mask grayscale image

% Mask the image using bsxfun() function

maskedGrayImage = bsxfun(@times, img, cast(BW, 'like', img));

figure; imshow(maskedGrayImage, []); title('masked with grayscale')

figure; imagesc(maskedGrayImage); title('grayscale to color');colorbar

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%

% Threshold image - manual threshold

T = maskedGrayImage > 537;

%Remove objects containing fewer than 50 pixels using bwareaopen function.

T2 = bwareaopen(T, 50);

figure; imshow(T2); title('Gray brigther than 537');

if T2 == zeros(size(img))

disp('lung has no tumor')

else

disp('lung has tumor(s)')

end

%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% glcms & haralick

I = uint8(maskedGrayImage); % very important makes glcms be correct

%[a,h,v,d] = haart2(I);

%figure; imagesc(a)

glcms = graycomatrix(I, 'offset', [0 1], 'Symmetric', true);

stats = graycoprops(glcms);

[x] = haralickTextureFeatures(glcms);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% theshold masked image

tumor = T2;

%figure; imshow(tumor, []); title('Tumormask')

TumorGrayImage = bsxfun(@times, maskedGrayImage, cast(tumor, 'like', maskedGrayImage));

figure; imshow(TumorGrayImage, []); title('TumorGrayImage')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% area of tumor more features which could be used as input to ann

numWhite = sum(T2(:));

Area = numWhite;

%Convex Area: It gives the number of pixels in

%convex image of the ROI.

LungArea = sum(BW(:));

%Perimeter: It is the distance between each

%adjoining pairs of pixel around the border of the

%ROI.

Edges = edge(TumorGrayImage); %approx

Perimeter = sum(Edges(:));

%Equivalent Diameter: It is the diameter of a circle

%with the same area as the ROI.

E\_dia = (Area\*4/pi)^0.5;

%Solidity: It is the proportion of the pixels in the

%convex hull that are also in the ROI.

Solidity = Area/LungArea;

%Irregularity Index: is the measure of how closely

%the shape of an object approaches that of a circle.

I = (4\*pi\*Area/(Perimeter)^2)^0.5;

%%% Eccentricity

Ecc = (4\*Area\*pi)/(Perimeter.^2);

%%%%

data = [Area,LungArea,Perimeter,E\_dia,Solidity,I]';

Simple\_data = [Area, Perimeter, Ecc]'

figure; imshow(Edges)

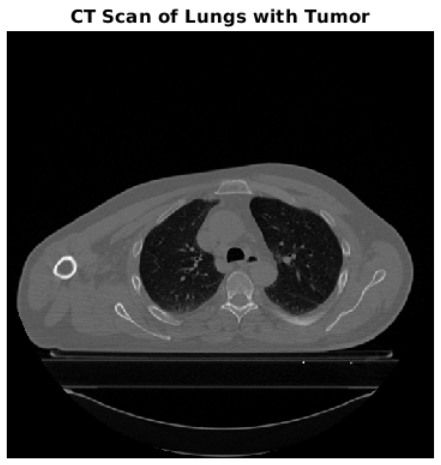
im00 = imfuse(img,Edges);

im0 = imfuse(img,im00, 'montage');

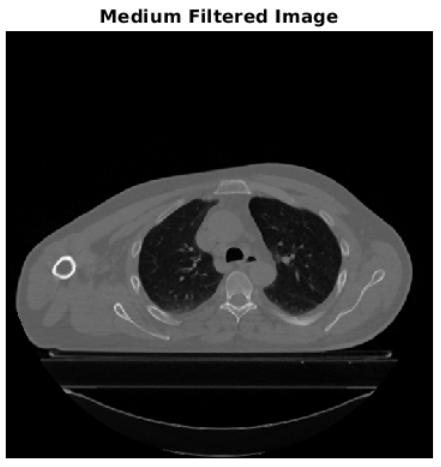
figure; imshow(im0); title('orig and tumor edges overlay')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

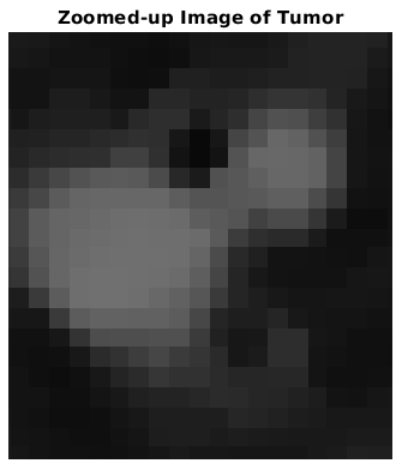
**C) Method One Pictures**



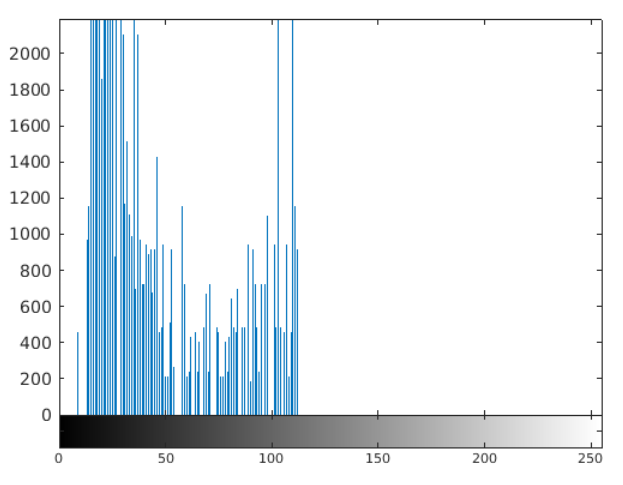
**Figure 5: CT Scans of Lungs with Tumor**



**Figure 6: Medium Filtered Image**

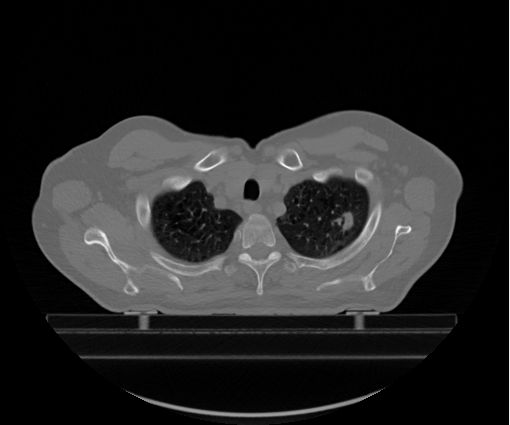


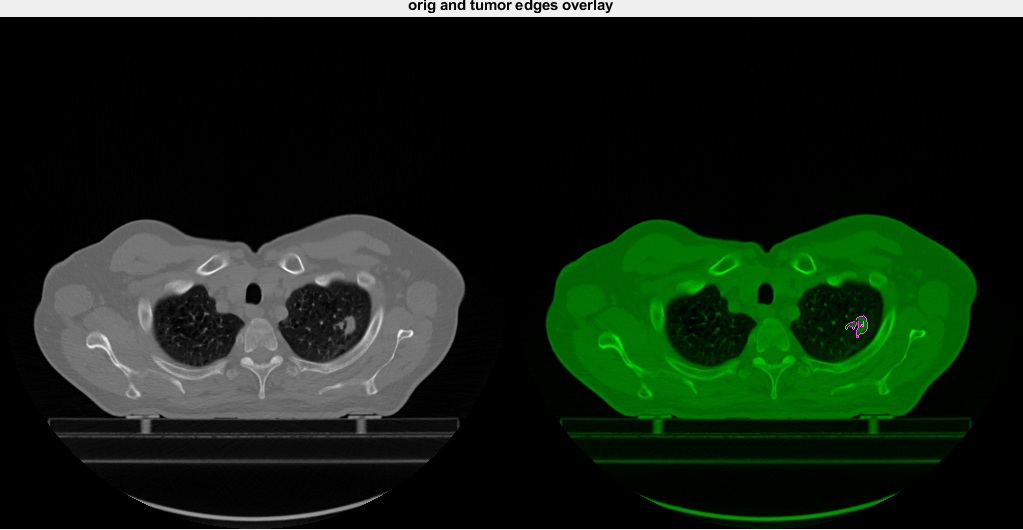
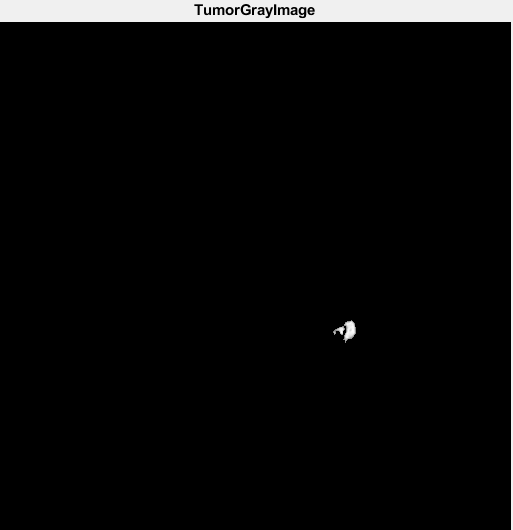
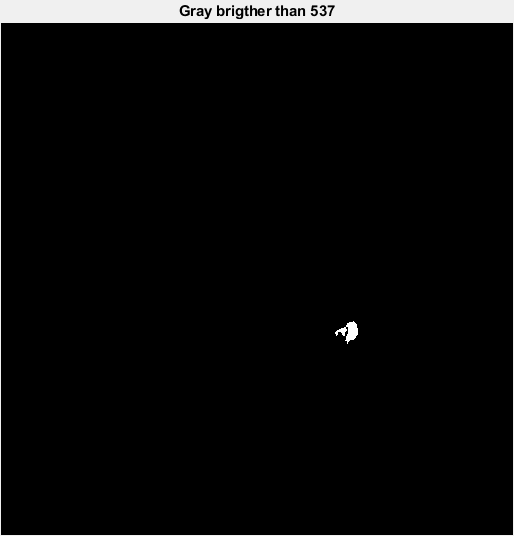
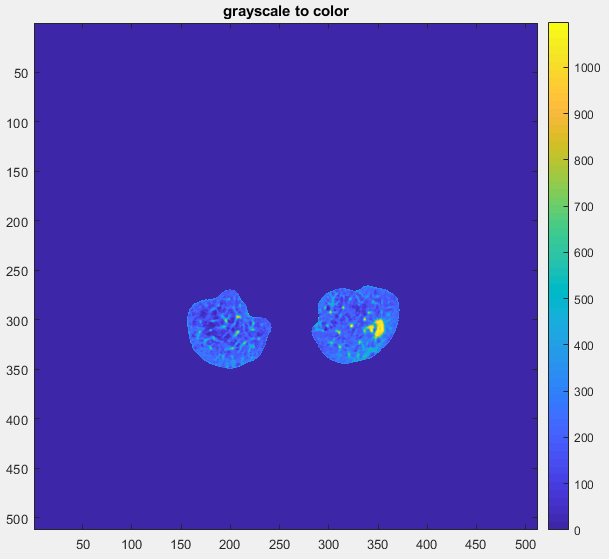
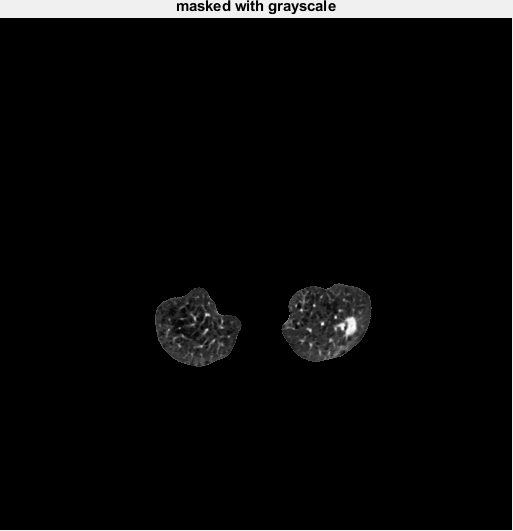
**Figure 7: Zoomed-Up Image of Tumor**



**Figure 8: Histogram of Tumor**

**D) Method Two Pictures**





lung has tumor(s)

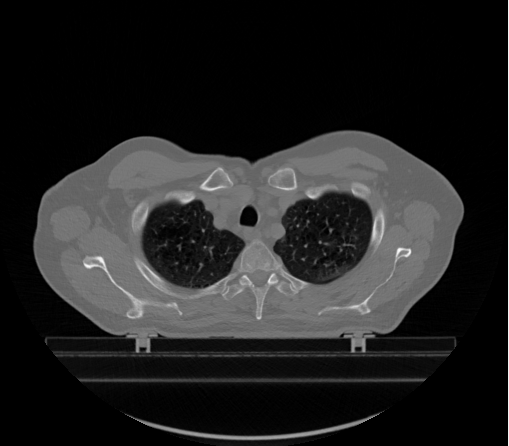
Simple\_data =

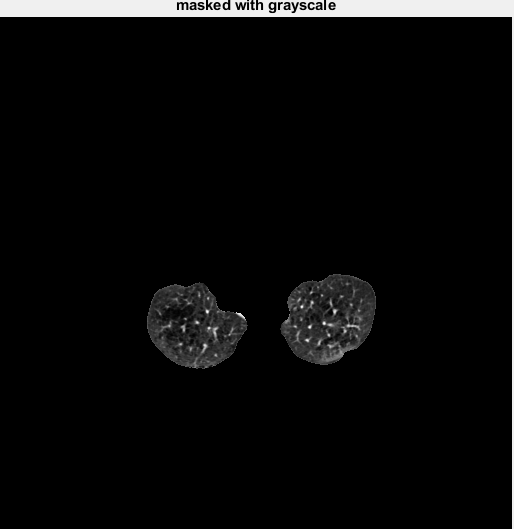
240.0000

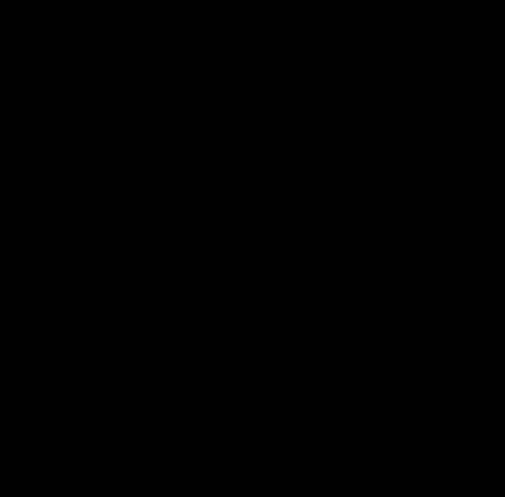
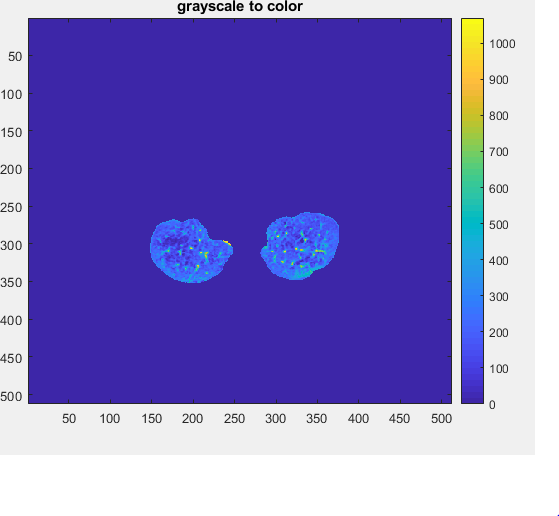
93.0000

0.3487

----------------------------------------------- image with no tumor-------------------------------------------







lung has no tumor

Simple\_data =

0

0

NaN