Covid 19-Xray Image Segmentation Using Thresholding And K-Means Clustering

Indrajith Wasala Mudiyanselage, Nisansala Wickramasinghe and Mohommad Afser Uddin

MATH 6346: Medical Image Analysis

The University of Texas at Dallas
Dec 07 , 2021

1 Introduction

The novel corona virus disease 2019 (COVID-19) is a global health emergency and causes severe inflammation in your lungs which leads to breathing difficulties. Even though most people recover from pneumonia without any lasting lung damages, others have breathing problems that can become severe enough to require treatment at the hospital with oxygen or even a ventilator.After experts identify a COVID-19, mainly they are concern with three factors. First and the most important one is severity of the infection. Next, whether patient have any pre-existing health problems that can increase the risk. Finally, how long will it take for a recovery and disease effects of patients long term lung health.

Chest X-ray is among the most commonly used imaging modalities used for above three concerns due to its low radiation, free of side effects and economic feasibility. Extraction of lung areas from the X-rays by accurately detecting lung boundaries is an important step, especially in determining whether the lung is normal or abnormal. Image segmentation is one of the methods widely used for this purpose.

This project proposes two unsupervised learning method for lung segmentation of COVID-19 patients based on threshold and K-means clustering. This computational procedures for image preparation can be used for further analysis by medical specialists. We proposes four-stage approach including pre processing, noise elimination, segmentation and border detection. X-ray images are typically low resolution, and the pixel intensity distribution can have wide verities due to patients age and configuration of X-ray capturing devise. Therefore, we carried out preprocessing step to improve the quality of the image and enhance significant areas. Then we used technique to eliminate noise already in X-ray images and noise occurred in the preprocessing steps by preserving the details of image. Finally, we used two unsupervised techniques to correctly separate lung area with background.

2 Proposed Methodology

Figure 1 depicts a principal scheme of proposed approach.

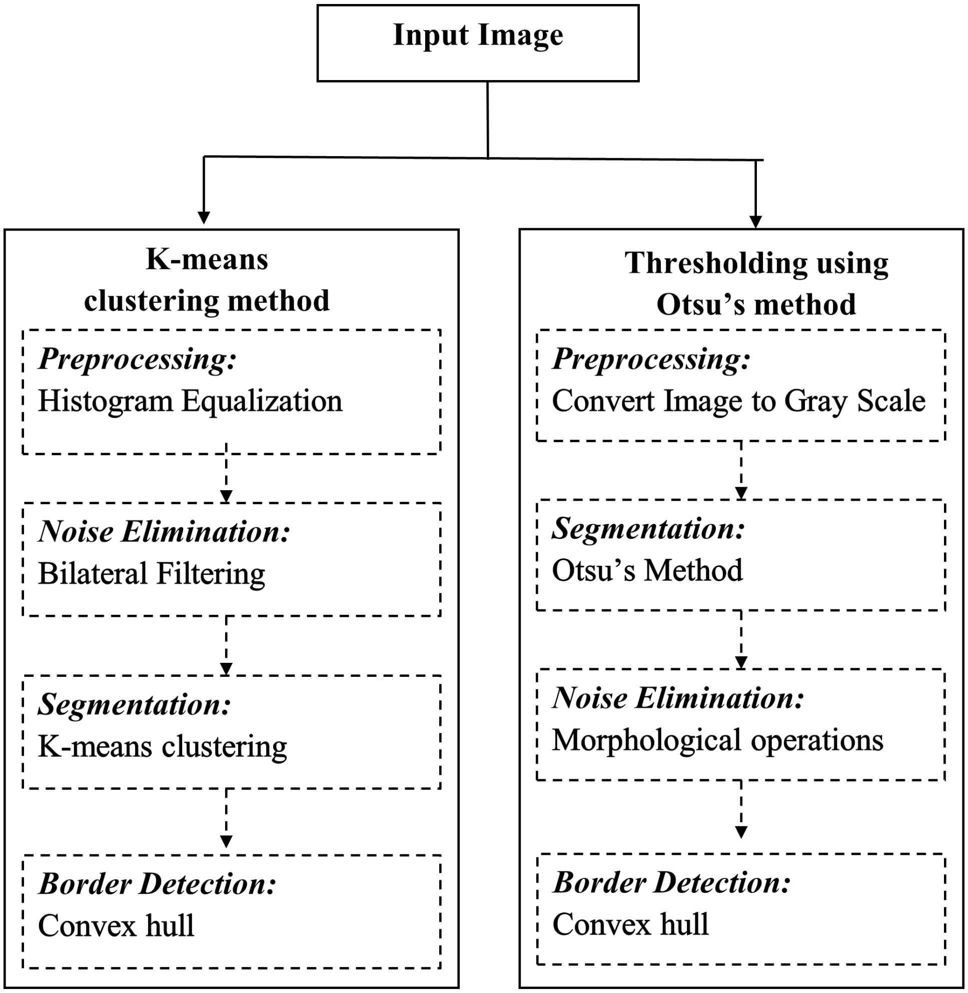

Figure 1: *The proposed algorithmic scheme for segmenting X-ray image*

2.1 K means clustering method

2.1.1 Preprocessing : Histogram Equalization

As the first step of our method we carried out histogram equalization for adjusting image intensities of the X-ray image to enhance contrast. The objective of this technique is to give a linear trend to the cumulative probability function associated to the image.

Let L be the number of possible intensity values for a given image ranging from 0 to $L-1$. Let

p denote the normalized histogram for given image with a bin for each possible gray level. Then

$$p(r_k) = \frac{n_k}{n}$$

where r_k is the k the gray level n_k is the number of pixels in the image having gray level r_k. Let r represent the gray levels of the image to be enhanced, $r \in [0, 1]$. Consider the transformation

$$s = T(r) = \int_0^r p_r(w)dw,$$

the cumulative distribution (CDF) of random variable r. Since $p_r \geq 0, T(r)$ is single valued monotonically increasing and $0 \leq T(r) \leq 1$,

$$\frac{ds}{dr} = \frac{d}{dr}T(r) = \frac{d}{dr}\int_0^r p_r(w) = p_r(r).$$

Hence

$$\frac{dr}{ds} = \frac{1}{p_r(r)}.$$

Then

$$p_s(s) = pr(r)\frac{dr}{ds} = pr(r)\frac{1}{pr(r)} = 1, 0 \leq s \leq 1.$$

So $p_s(s)$ is the uniform probability density function.

For discrete values, we deal with probabilities and summations instead of probability density functions and integrals

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}, k = 0, 1, \ldots, L-1.$$

The transformation given in above equation is called histogram equalization or histogram linearization.

The resulting function s_k in the range $0 \leq s_k \leq 1$ needs to be converted to the gray levels $0 \leq s_k \leq L-1$ by either of the two ways:

$$\lfloor (L-1)s_k + 0.05 \rfloor \quad \text{or} \quad \lfloor (L-1)\frac{s_k - s_{min}}{1 - s_{min}} + 0.05 \rfloor$$

both maps$s_{max} = L-1$ but second also maps $s_{min} = 0$

2.1.2 Noise elimination : Bilateral Filtering

This step helps to enhance the visual appearance of X-ray image by removing the irrelevant noise and undesired parts of background. We used bilateral filter as the main filter for noise suppression.

Gaussian Filter

The Gaussian filtering is a weighted average of the intensity of the adjacent positions with the average weighted more towards the value of central pixel.

Image filtered by Gaussian Convolution denoted by $GC[\cdot]$, is given by:

$$GC[I]_p = \sum_{q \in S} G_\sigma(\|p - q\|) I_q$$

where,

p is the center position.

$GC[I]$ is the output of a filter GC applied to the image I.

I_q is the image value at pixel position q.

S is set of all possible image locations,

$G_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$ which is the 2D Gaussian kernel.

The weight for pixel q is defined by the Gaussian $G_\sigma(\|p - q\|)$, where σ is a parameter defining the neighborhood size.

Bilateral Filter

Bilateral filter replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels in a manner very similar to Gaussian convolution. The difference is that the bilateral filter takes into account the difference in value with the neighbors to preserve edges while smoothing.

Image filtered by Bilateral Filter denoted by $BF[\cdot]$, is given by:

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q$$

where normalization factor $W_p = \sum_{q \in S} G_{\sigma_s}(\|p-q\|)G_{\sigma_r}(|I_p - I_q|)$.
Parameters σ_s and σ_r will specify the amount of filtering for the image I.

2.1.3 Segmentation : K-means clustering

The clustering technique is the most common and popularly used image segmentation method. The K-means algorithm clusters data by trying to separate pixels in K groups of equal variance, minimizing within-cluster sum-of-squares.

K-means algorithm:

Step 1: Initialize the value of K and the centroid randomly.

Step 2: Calculate Euclidian distance for each pixel in an image from pixel to the centroid using $D_K(i,j) = \|I_K(i,j) - C_K\|$ where, I denote image and C_K is the centroid.

Step 3: Arrange the pixels with in the cluster based on their minimum distances $(D_1, D_2, \ldots D_K)$

Step 4: Recalculate the new position of the centroid using $C_K = \frac{1}{s_k}\sum_{(i,j)\in S_K} I_K(i,j)$ where $I_K(i,j)$ is the value of the pixel labeled with kth segment and S_K is the set of indices of the pixel belonging to class $K \in Y$ and Y is the set of labels of segments.

Step 5: Repeat the step 2 to 4 until any reassignment occurs.

Step 6: Reshape the cluster pixels into the image.

2.2 Thresholding usung Otsu's method

2.2.1 Pre-processing

At first, input RGB images are converted to grey scale to apply thresholding method for image segmentation.

2.2.2 Thresholding using OTSU's method

Thresholding is a method of image segmentation where we change the pixels of an image to make the image easier to analyze. Thresholding creates binary images from grey-level ones by turning all pixels below some threshold T to zero and all pixels above T to one, where each pixel is either categorized as background point or foreground point.

If $J(x, y)$ be the thresholded version of the grey-level image $I(x, y)$ at threshold T,

$$J(x,y) = \begin{cases} 1 & if \;\; I(x,y) > T \\ 0 & if \;\; I(x,y) \leq T \end{cases}$$

Otsu's method is a nonparametric thersholding method of automatic image segmentation. Otsu's method implicates iterating through all the possible threshold values and choose the optimal threshold value based on the criterion that it will minimize intra-class intensity variance or equivalently maximize inter-class variance.

In OTSU's method, we classify the pixel into two classes, C_1 and C_2 by a threshold at level T. The optimal T minimizes the within class variance σ_W^2, or equivalently, maximize the between class variance σ_B^2, where σ_W^2 and σ_B^2 defined as

$$\sigma_W^2 = w_1\sigma_1^2 + w_2\sigma_2^2$$

$$\sigma_B^2 = w_1(\mu_1 - \mu_L)^2 + w_2(\mu_2 - \mu_L)^2 = w_1w_2(\mu_2 - \mu_1)^2$$

where σ_1^2 and σ_2^2 are class variances, μ_1 and μ_2 are class mean levels, and w_1 and w_2 are the probabilities of class occurrence.

2.2.3 Noise elimination by Morphological operations

Morphological operations are used to eliminate noise or other artifacts from the image. Morphological operations typically probe an image with a small shape or template known as structuring element. Most morphological techniques operates on binary images and creates new binary images, where structuring element determines the output of the morphological operations.

There are different types of morphological operations, such as Dilation, Erosion, Opening, Closing etc. We used morphological operation opening for our analysis. Morphological opening is useful for removing small objects from an image while preserving the shape and size of larger objects in the image i.e. it has the ability to eliminate regions narrower than the structuring element.

The opening of an image by a structuring element is an erosion followed by a dilation. Let A and S be sets in $\mathbb{Z}^2$. The opening of A by S, denoted by $A \circ S$, is defined as

$$A \circ S = (A \ominus S) \oplus S$$

where $\ominus$ and $\oplus$ denote erosion and dilation respectively.

2.3 Convex hull

As the final step, for both methods we carried out convex hull algorithm to search the border points and determine outline of the lung's boundaries. Convex hull (CH) is basically an important geometrical problem that could be solved computationally. The problem is all about constructing, developing, articulating, circumscribing or encompassing a given set of points in plane by a polygonal capsule called convex polygon. We used 'convhull' function in MATLAB to implement the algorithm.

3 Results

The algorithms are developed in Matlab 2015a and tested on real-time lung X-ray images. Results were based on X-ray image of mild COVID-19 case. We applied both K-means clustering method and Otsu's method for these images and compared results.

3.1 K means clustering method

In K-means clustering algorithm, prior to segmentation, preprocessing was performed by a histogram equalization. Figure 2 represent images and corresponding histograms for original image and image after histogram equalization. As can be seen in Figure 2, the area of the lungs in histogram equalization image is darker than the one in the original image and the contrast of the vessels, tissues and ribs is small. Moreover intensities spread uniformly over a large range after histogram equalization.

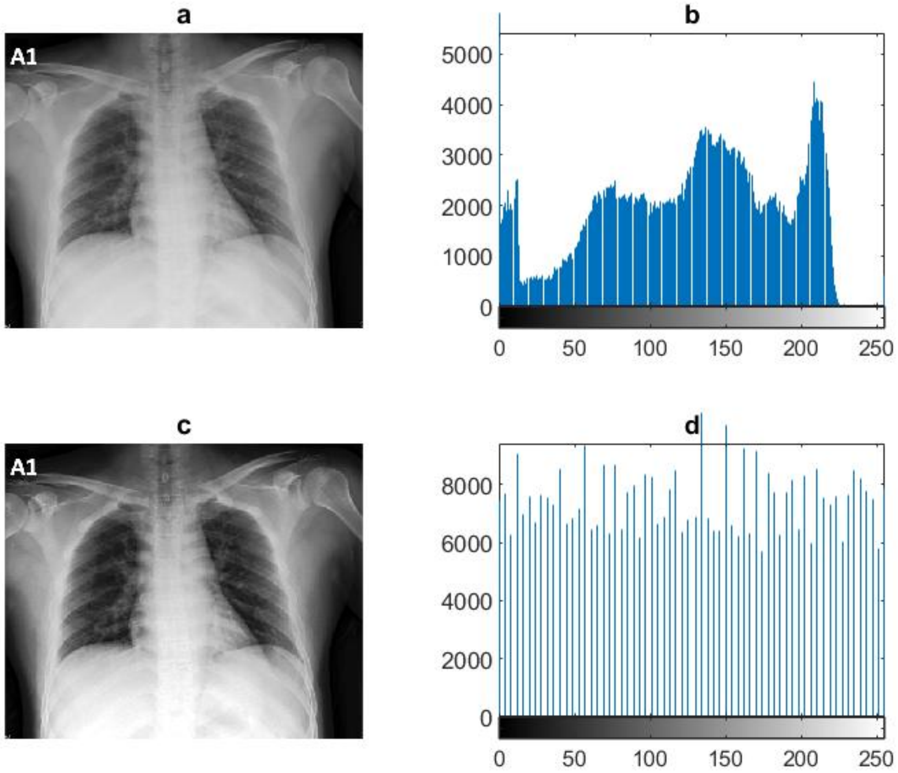

Figure 2: *Histogram equalization. a) Original image b) Histogram for original image c) Image after histogram equalization d) Corresponding histogram for image c*

Bilateral filtering method was used for noise elimination. Degree of smoothing, specified as a positive number and set it to be larger than the variance of the noise. Using this method the sharpness of strong edges have been preserved while canceling the noise. Output is presented in Figure 3a.

K-means uses the squared Euclidean distance metric and the k-means++ algorithm introduced in MATLAB documentation for cluster center initialization. We started with 2 initial clusters and segments X-ray image into 2 clusters. Intermediate outputs and the segmented image after K-means clustering are plotted in Figure 3. In Figure 3c we can see the clear separation between lung area from the background but with rough edges.

Border detection is carried out using convexhull method to read the image properly and identify the content of the image carefully from the background. The objective was to produce a clean edge map by extracting the principal edge features of the image. Figure 3d represent image after border detection with clean edges.

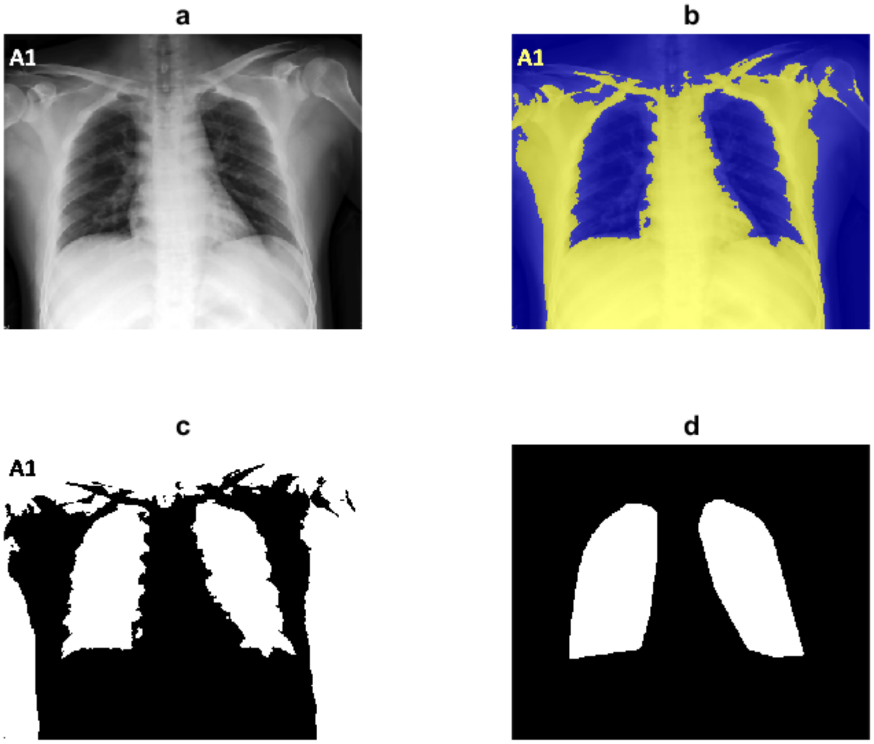

Figure 3: *a) Image after applying bilateral filer. b) Image after applying K-means c) Binary image d) Image after convexhull algorithm*

3.2 Thresholding using OTSU's method

In the pre-processing stage we convert the RGB image to grey-image. In figure 4, we notice that the histogram of the grey-image shows that the intensity is not distributed uniformly, whether it gives the hint that image can be separated based on the intensity of pixels.

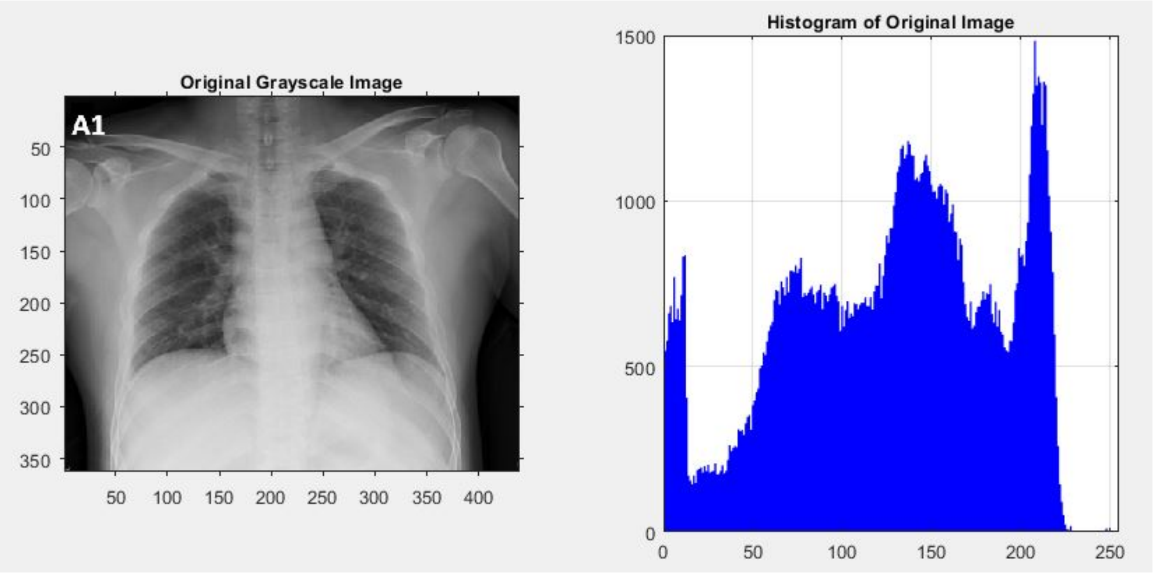

Figure 4: *Original grey-scale image with it's histogram*

MATALB's dedicated function "greythresh" for OTSU's method is used to get the optimal threshold level. A binary image is obtained based on this optimal threshold value. Segmented image using OTSU's method is placed on figure 5. Here, we see this image basically has two parts, one is background and the other is the foreground of the image.

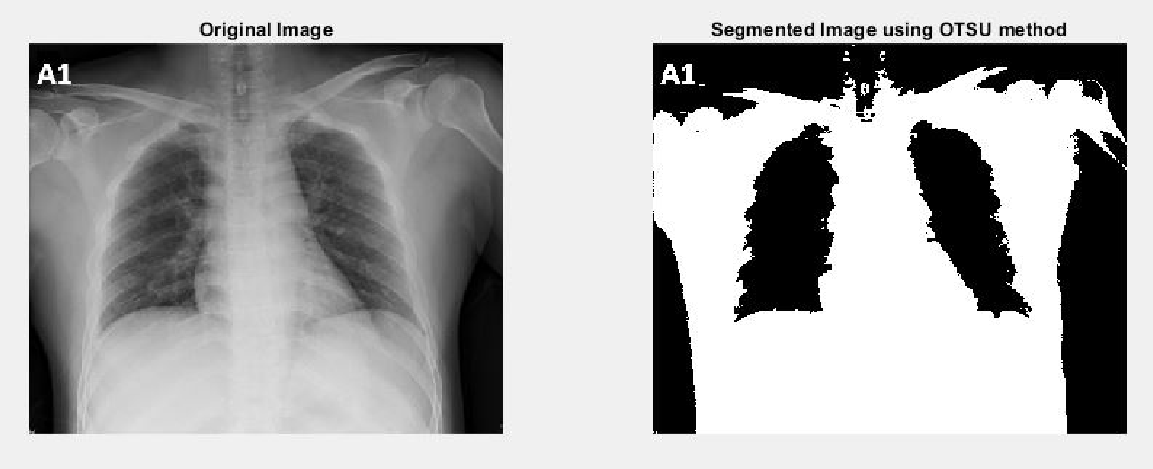

Figure 5: *Original image and Segmented iamge using OTSU's method*

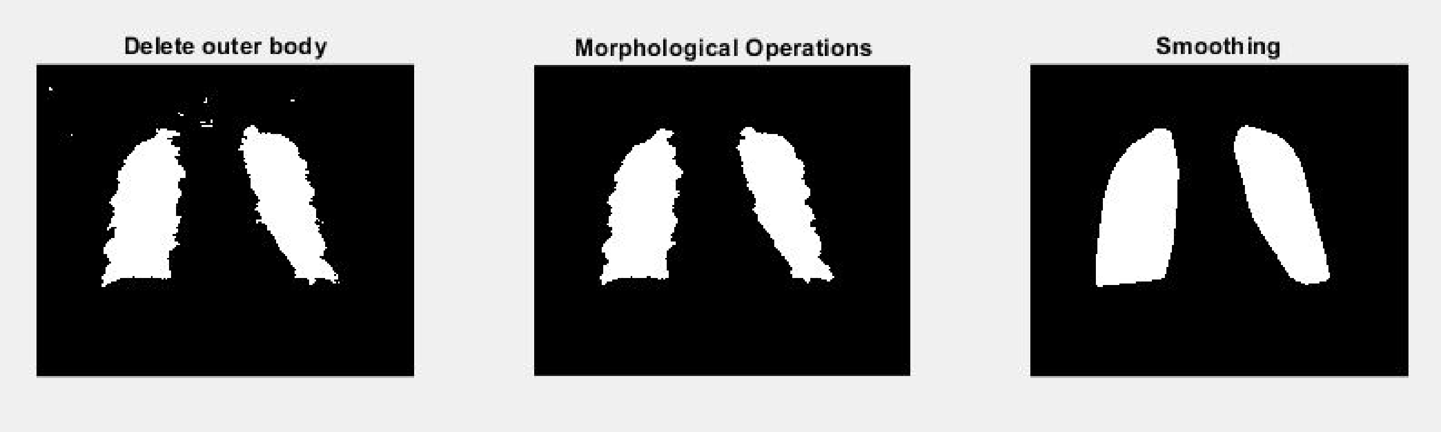

Figure 6: .

The first image of figure 6 is obtained by deleting the outer body of the image, but we see there are few small white regions, which are the noise of the image, and we know thresholding is sensitive to noise. Morphological operation opening is applied to eliminate those white region with disk shaped structuring element. This output is displayed on the second image of figure 6. The last output of figure 6 is achieved by smoothing the outline of the lung's boundary by Convex Hull operation.

4 Discussion and Conclusion

In this project, we have developed methodology which allows to extract region of interest (ROI) from X-ray image using image segmentation. The information of lung segments can be used to evaluate size and lung features which can be useful for medical assessment by medical specialists. However this method is incapable of extracting lung region associated with heart.

References

[1] A. Zotin, Y. Hamad, K. Simonov and M. Kurako (2019). ”Lung boundary detection for chest X-ray images classification based on GLCM and probabilistic neural networks”, *Procedia Computer Science*, Vol. 159, pp 1439-1448.

[2] R. Kumari, N. Gupta and N. Kumar (2021). ”Segmentation of Covid-19 Affected X-Ray Image using K-means and DPSO Algorithm”. *International Journal of Mathematical, Engineering and Management Sciences*, Vol. 6, pp 1255-1275.

[3] P. Pattrapisetwong and W. Chiracharit (2016). ”Automatic lung segmentation in chest radiographs using shadow filter and local thresholding” , *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)* , pp. 1-6.

[4] S. Paris, P. Kornprobst, J. Tumblin and F. Durand (2009), ”Bilateral Filtering: Theory and Applications”, *Foundations and Trends® in Computer Graphics and Vision*, Vol. 4: No. 1, pp 1-73.

Author Contributions

Indrajith Wasala Mudiyanselage:

Finding literature and figuring out K-means method for image Segmentation.

Writing Project Proposal

Presenting and making slides related to Introduction, Methodology, Discussion Referencess

Contribution for final report by writing sections related to Introduction, Methodology, Discussion Referencess.

Nisansala Wickramasinghe:

Performing K-means Method using MATLAB

Writing Progress report

Presenting and making slides related to K-means Method

Contribution for final report by writing sections related to K-means Method (Methodology and Results).

Mohammad Afser Uddin:

Finding literature and figuring out Otsu's method for image Segmentation.

Performing Otsu's Method using MATLAB

Presenting and making slides related to Otsu's Method

Contribution for final report by writing sections related to Otsu's Method.

Appendix

Matlab code of OTSU's method

```
clc;
close all;
imtool close all;
clear;
workspace;
format long g;
format compact;
fontSize = 10;

userpath('C:/Users/mxu190001/Desktop/Math6346CovidProjectData/Math634
fullFileName = fullfile('00006473.png');
grayImage = imread(fullFileName);
[rows, columns, numberOfColorBands] = size(grayImage);
if numberOfColorBands > 1
% It's not really gray scale like we expected - it's color.
% Convert it to gray scale by taking only the green channel.
grayImage = grayImage(:, :, 2); % Take green channel.
end

% Display the original gray scale image.
subplot(1, 2, 1);
imshow(grayImage, []);
axis on;
title('Original␣Grayscale␣Image', 'FontSize', fontSize);

% Enlarge figure to full screen.
```

```
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
% Give a name to the title bar.
set(gcf, 'Name', 'Demo␣by␣ImageAnalyst', 'NumberTitle', 'Off')
drawnow;

%================================================================================
% Let's compute and display the histogram.
[pixelCount, grayLevels] = imhist(grayImage);
% The first and last bin of pixelCount are so huge that it suppresses
% of the rest of the histogram when plotted.  Zero out these bins so
pixelCount(1) = 0;
pixelCount(end) = 0;
subplot(1, 2, 2);
bar(grayLevels, pixelCount, 'BarWidth', 1, 'FaceColor', 'b');
grid on;
title('Histogram␣of␣Original␣Image', 'FontSize', fontSize);
xlim([0 grayLevels(end)]); % Scale x axis manually.

%================================================================================
% Threshold (binarize) the image using OTSU's method

level = graythresh(grayImage);
grayImage = imbinarize(grayImage,level);

img = imread('00006473.png');
imshowpair(img,grayImage,'montage');
subplot(1,2,1);
imshow(img);
title('Original␣Image', 'FontSize', fontSize);
subplot(1,2,2);
```

```
imshow(grayImage);
title('Segmented␣Image␣using␣OTSU␣method', 'FontSize', fontSize);

binaryImage = imclearborder(-grayImage);
binaryImage = imfill(binaryImage, 'holes');

SE = strel('disk',2);
a = imopen(binaryImage,SE);

b=bwconvhull(a,'objects');

subplot(1,3,1);
imshow(binaryImage);
title('Delete␣outer␣body', 'FontSize', fontSize);
subplot(1,3,2);
imshow(a);
title('Morphological␣Operations', 'FontSize', fontSize);
subplot(1,3,3);
imshow(b);
title('Smoothing', 'FontSize', fontSize);
```

Matlab code of K-means method

```
clc;
clear all;

subplot(2,2,1)
im = imread('img.png');
imshow(im)
title('a')
```

```
subplot(2,2,2)
imhist(im)
title('b')

% performing histogram equalization
subplot(2,2,3)
J = histeq(im);
imshow(J)
title('c')

subplot(2,2,4)
imhist(J)
title('d')

%Remove gaussian noise (IE random noise in image)
subplot(2,1,1)
imshow(J)
title('Image␣after␣histogram␣equalizaton')

% Degree of smoothin set it to be larger than the variance of
subplot(2,1,2)
patch = imcrop(J,[170, 35, 50 50]);
patchVar = std2(patch)^2;
DoS = 2*patchVar;
% apply bilateral filter
I = imbilatfilt(J,DoS);
imshow(I)
title(['Degree␣of␣Smoothing:␣',num2str(DoS)])

subplot_tight(2,2,1, [0.1,0.1]);
```

```
imshow(I)
title('a')

subplot(2,2,2)
%perform k means clustering.
[L,Centers] = imsegkmeans(I,2);
B = labeloverlay(I,L);
imshow(B)
title('b')

% binary image
subplot(2,2,3)
BW = imbinarize(L);
imshow(BW)
title('c')

%apply convex hull method
l=imclearborder(BW);
l = bwareafilt(l,2);
l = imfill(l,'holes');
%imshow(l,[]);


subplot(2,2,4)
k=bwconvhull(l,'objects');
imshow(k);
title('d')
```