Evaluation lab 3

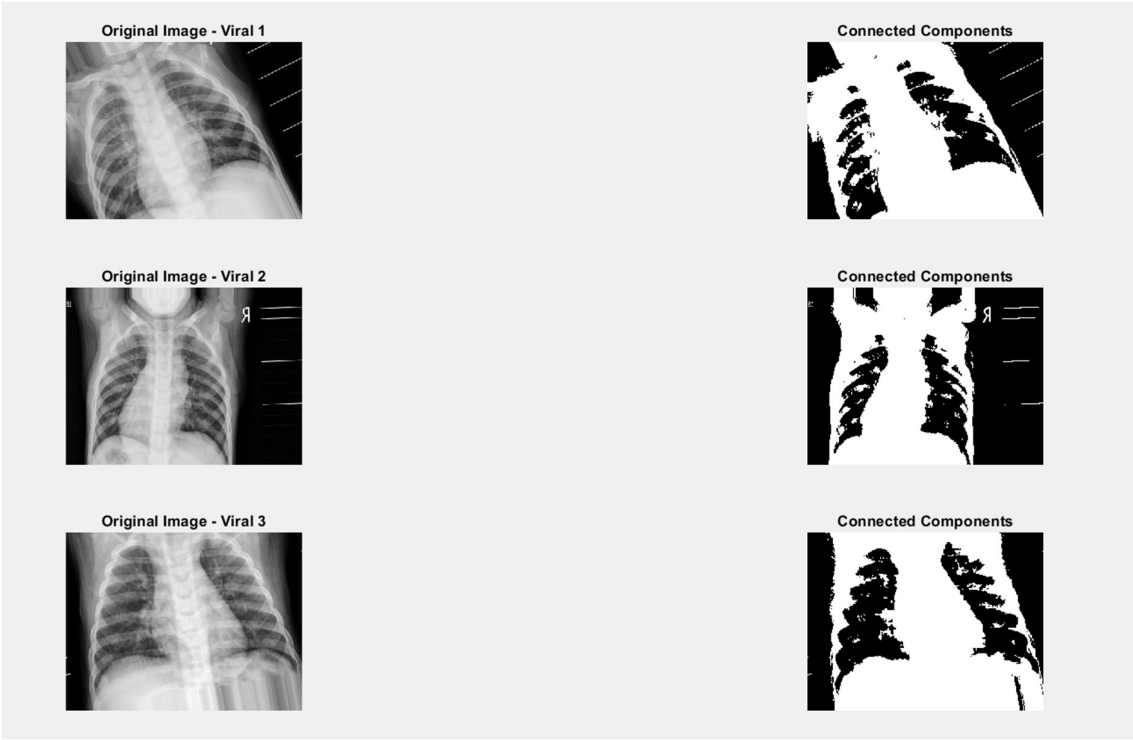
1. **Perform the following morphological operations. [10 M] [CO 05]**
   a. **Connected components.**

CODE:

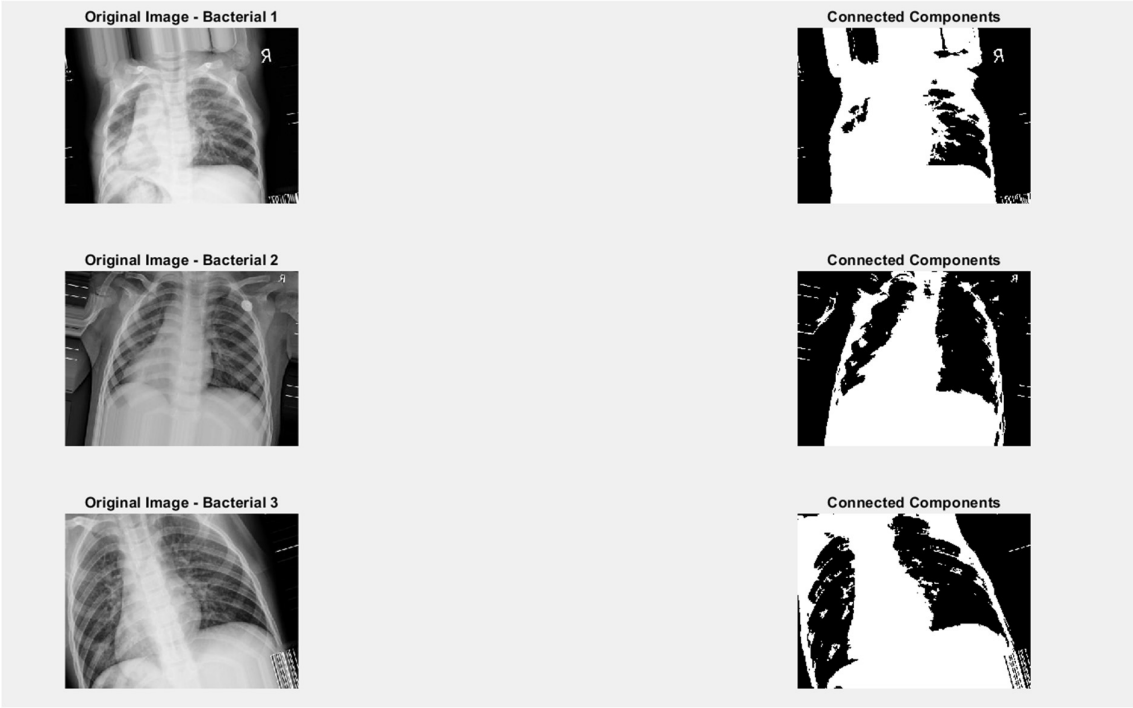
```
imagePaths = {
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral1.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral2.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral3.jpeg'
};

threshold = 0.5;

figure;

for i = 1:length(imagePaths)
    originalImage = imread(imagePaths{i});
    binaryImage = imbinarize(originalImage, threshold);

    % connected components
    labeledImage = bwlabel(binaryImage);

    subplot(length(imagePaths), 2, (i-1)*2 + 1);
    imshow(originalImage);
    title(['Original Image - Viral ', num2str(i)]);
    subplot(length(imagePaths), 2, (i-1)*2 + 2);
    imshow(labeledImage);
    title('Connected Components');
    colormap gray;
end
```

OUTPUT:

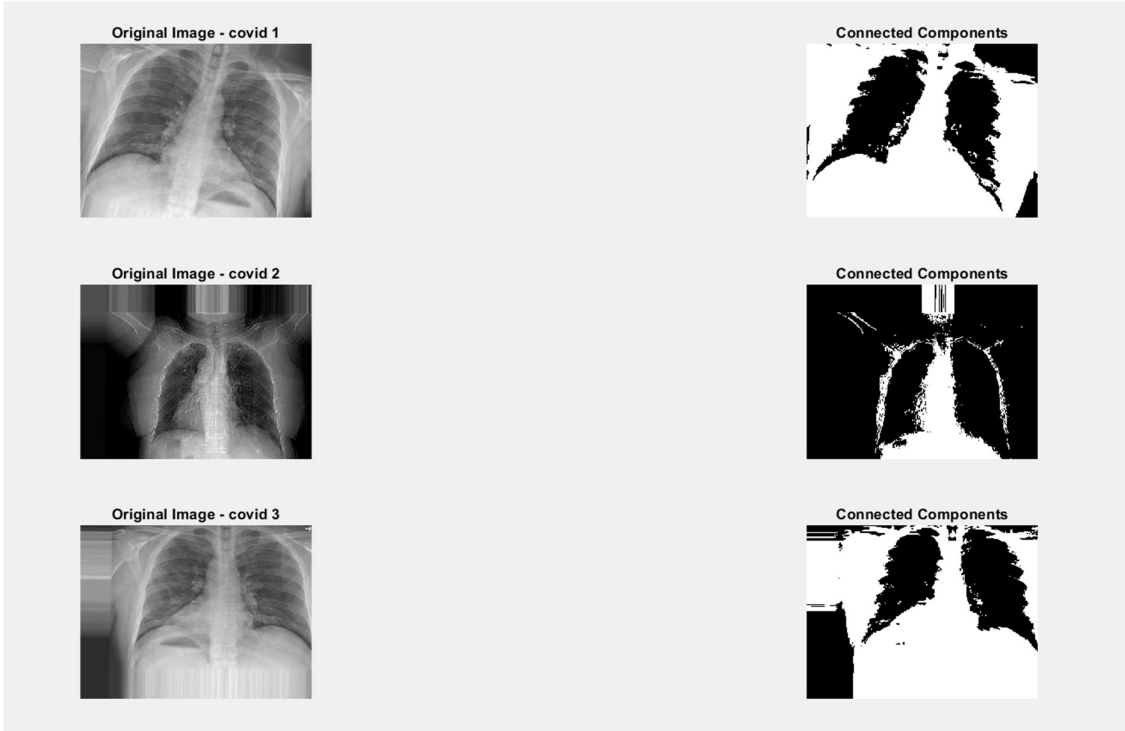
VIRAL:



BACTERIAL:



COVID:



b. convex hull

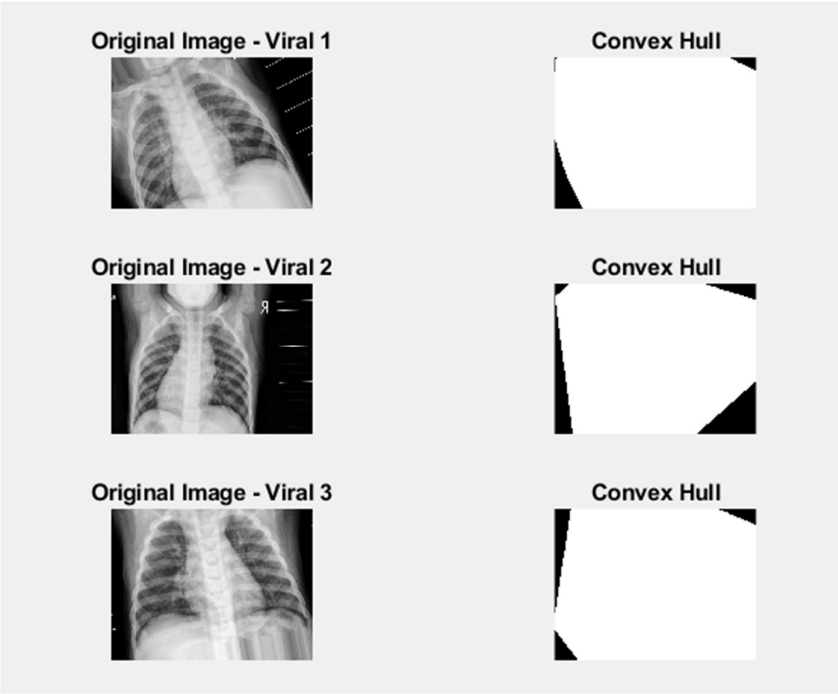
CODE:

```
imagePaths = {
       'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\bacterial\bact1.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\bacterial\bact2.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\bacterial\bact3.jpeg'
};

figure;

for i = 1:length(imagePaths)
    originalImage = imread(imagePaths{i});

    binaryImage = imbinarize(originalImage);

    %convex hull
    convexHullImage = bwconvhull(binaryImage);

    subplot(length(imagePaths), 2, (i-1)*2 + 1);
    imshow(originalImage);
    title(['Original Image - Bacterial ', num2str(i)]);
    subplot(length(imagePaths), 2, (i-1)*2 + 2);
    imshow(convexHullImage);
    title('Convex Hull');
```

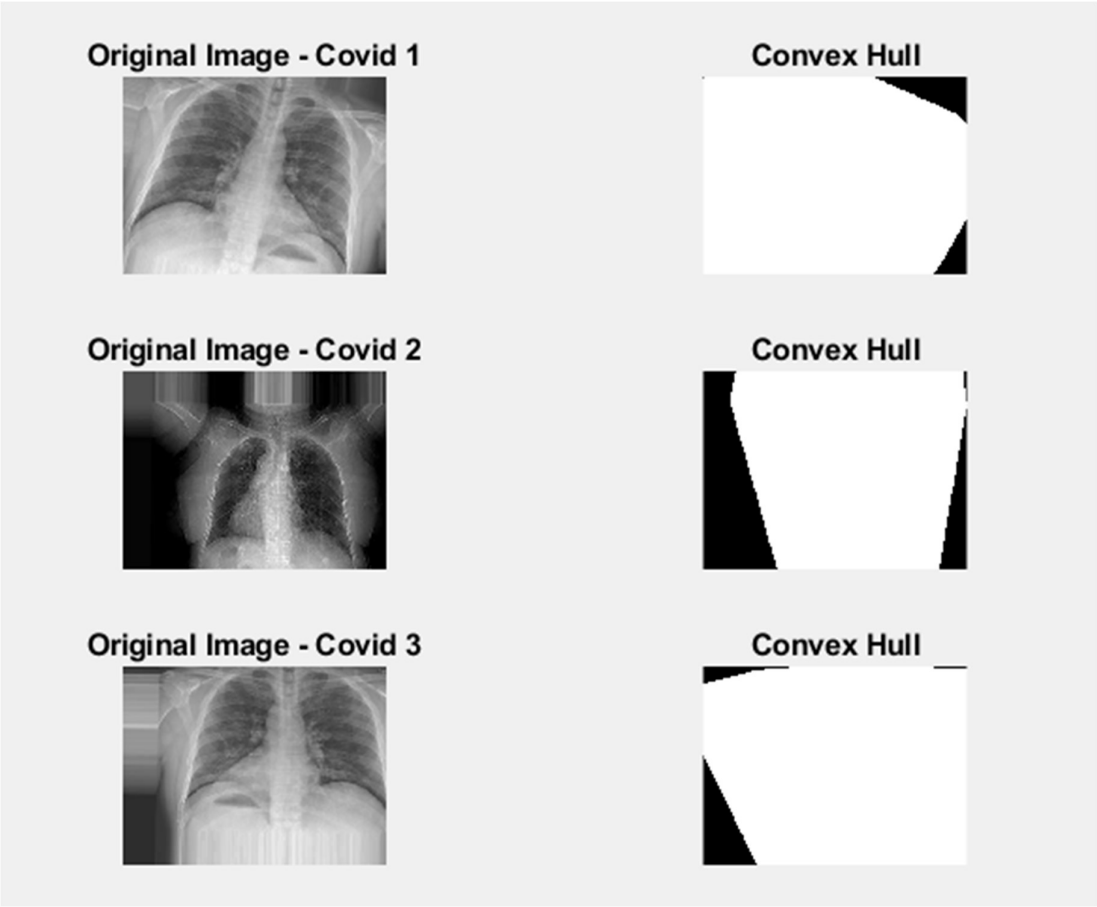
```
end

colormap gray;
```

OUTPUT:

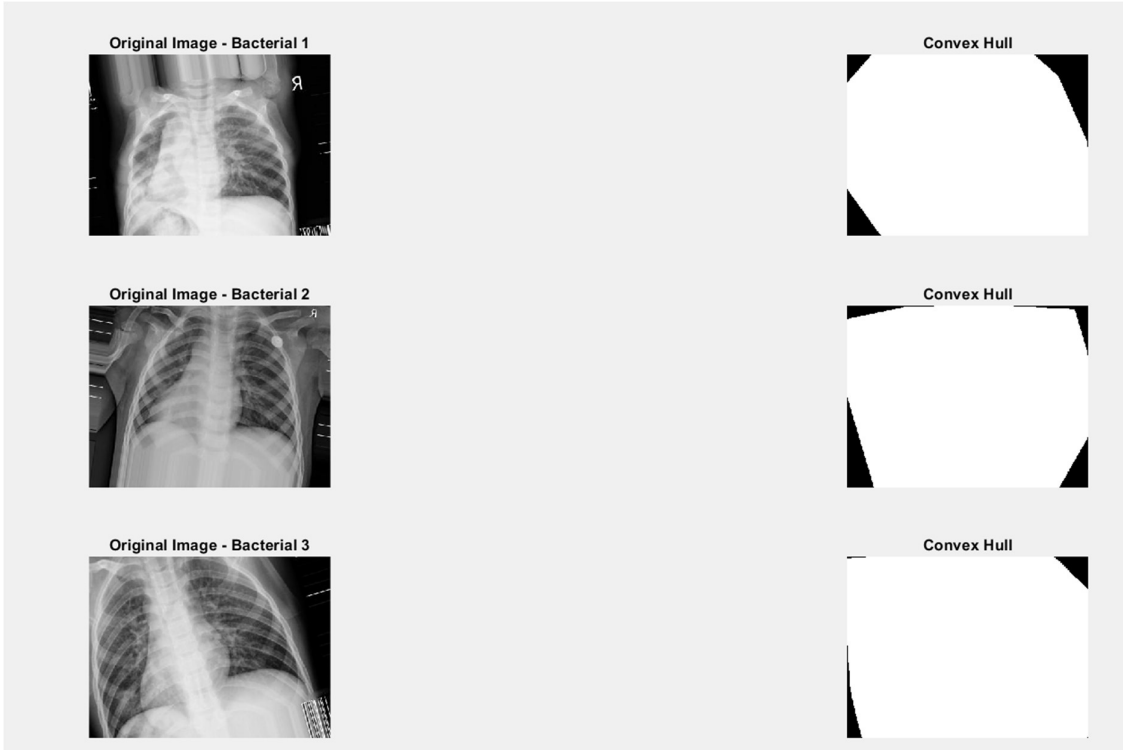
VIRAL:



COVID:


Original Image - Covid 1
Convex Hull
Original Image - Covid 2
Convex Hull
Original Image - Covid 3
Convex Hull

BACTERIAL:



2.Perform the following segmentation algorithms. [10 M] [CO 04]

Global thresholding using histogram analysis.

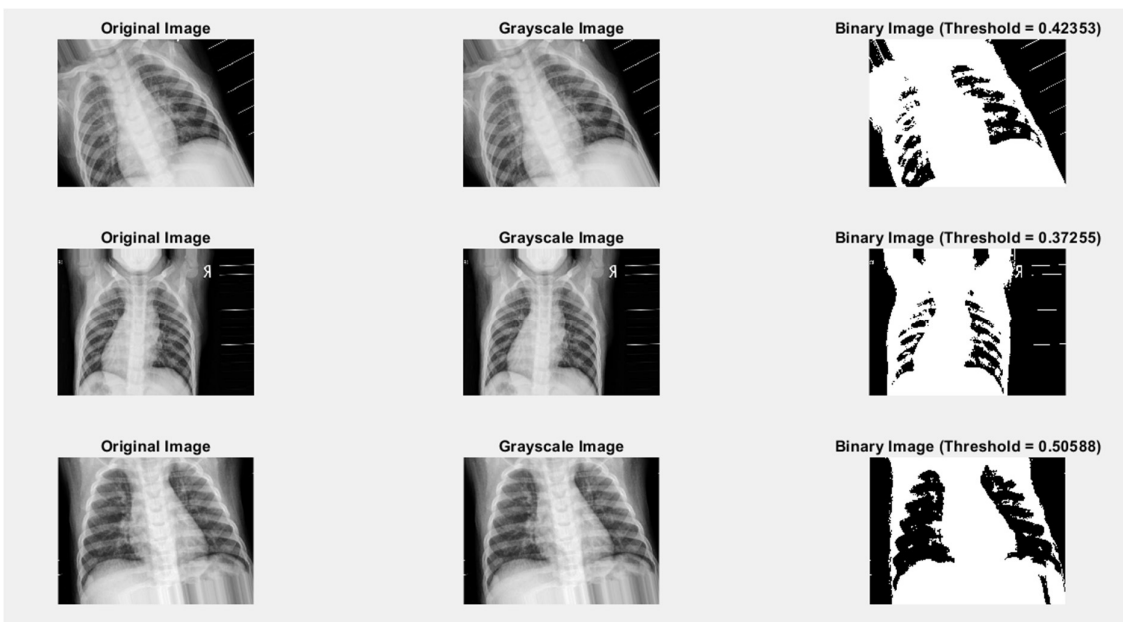
CODE:

```
imagePaths = {
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral1.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral2.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral3.jpeg'
};

outputPaths = {
    'output1.jpg',
    'output2.jpg',
    'output3.jpg'
};

figure('Position', [100, 100, 1200, 600]);

for i = 1:3
    image = imread(imagePaths{i});
    imageGray = image;
    threshold = graythresh(imageGray);
    binaryImage = imbinarize(imageGray, threshold);

    subplot(3, 3, (i-1)*3 + 1);
```

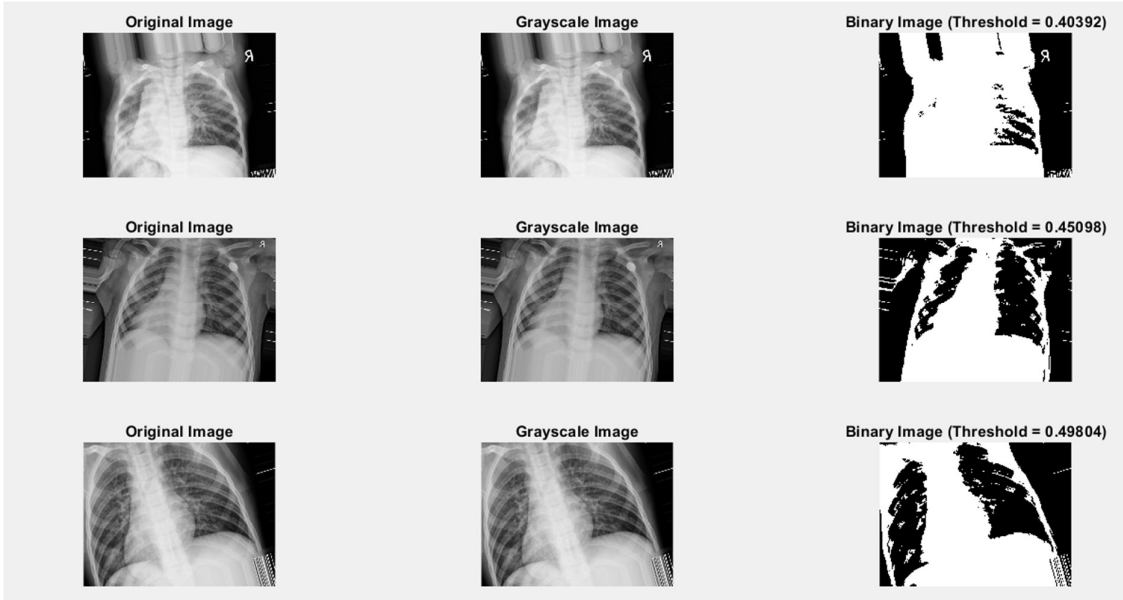
```
    imshow(image);
    title('Original Image');
    subplot(3, 3, (i-1)*3 + 2);
    imshow(imageGray);
    title('Grayscale Image');
    subplot(3, 3, (i-1)*3 + 3);
    imshow(binaryImage);
    title(['Binary Image (Threshold = ' num2str(threshold) ')']);
end

sgtitle('Image', 'FontSize', 16);
```

OUTPUT:

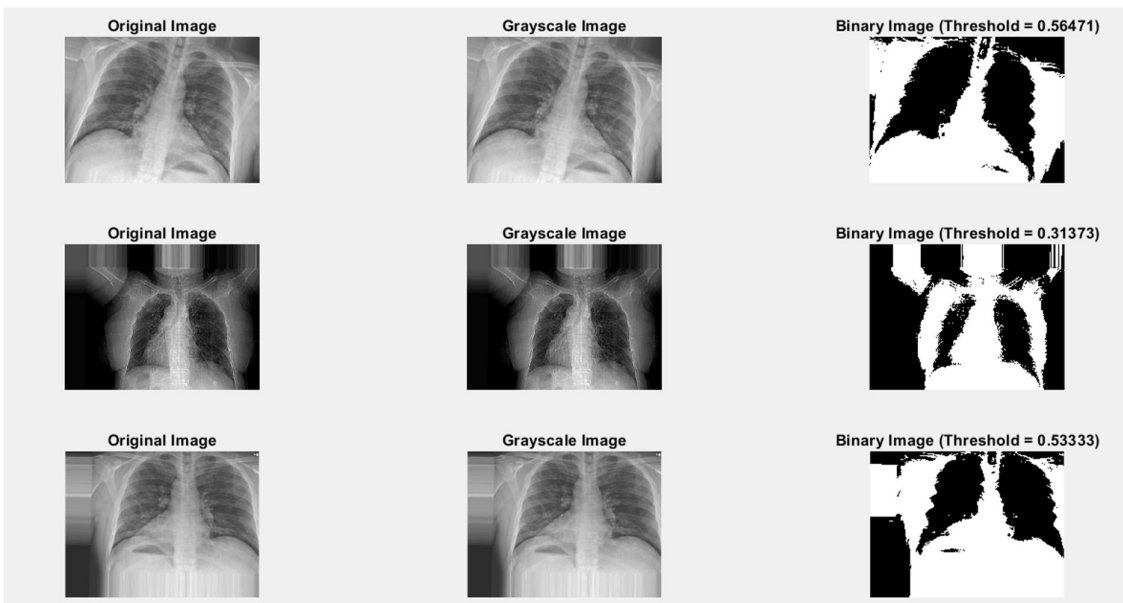
VIRAL:



BACTERIAL:



COVID:



Adaptive thresholding using histogram analysis.

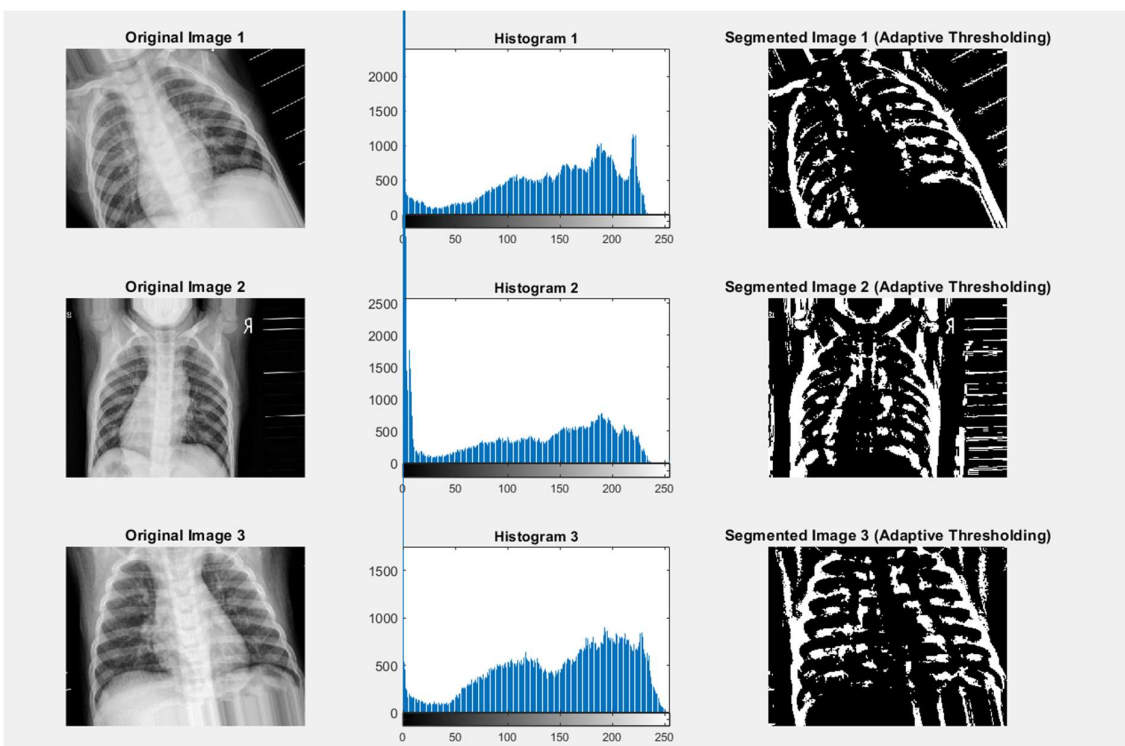
CODE:

```
image_paths = {
      'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\bacterial\bact1.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\bacterial\bact2.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\bacterial\bact3.jpeg'
};
```

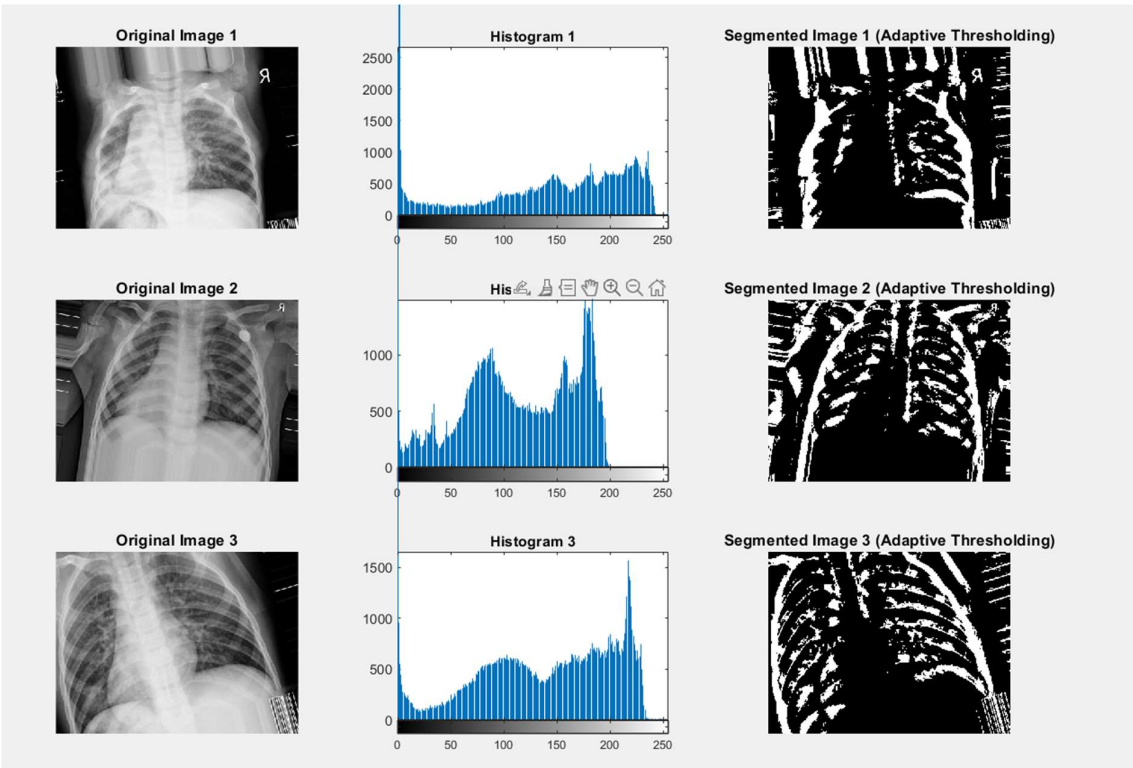
```
for i = 1:length(image_paths)
    image = imread(image_paths{i});

    if size(image, 3) == 3
        gray_image = rgb2gray(image);
    else
        gray_image = image;
    end

    subplot(3, 4, (i-1)*4 + 1);
    imshow(gray_image);
    title(['Original Image ', num2str(i)]);
    subplot(3, 4, (i-1)*4 + 2);
    imhist(gray_image);
    title(['Histogram ', num2str(i)]);

    binary_image_adaptive = imbinarize(gray_image, 'adaptive',
'ForegroundPolarity', 'bright', 'Sensitivity', 0.5);

    subplot(3, 4, (i-1)*4 + 3);
    imshow(binary_image_adaptive);
    title(['Segmented Image ', num2str(i), ' (Adaptive Thresholding)']);
end
```

OUTPUT:

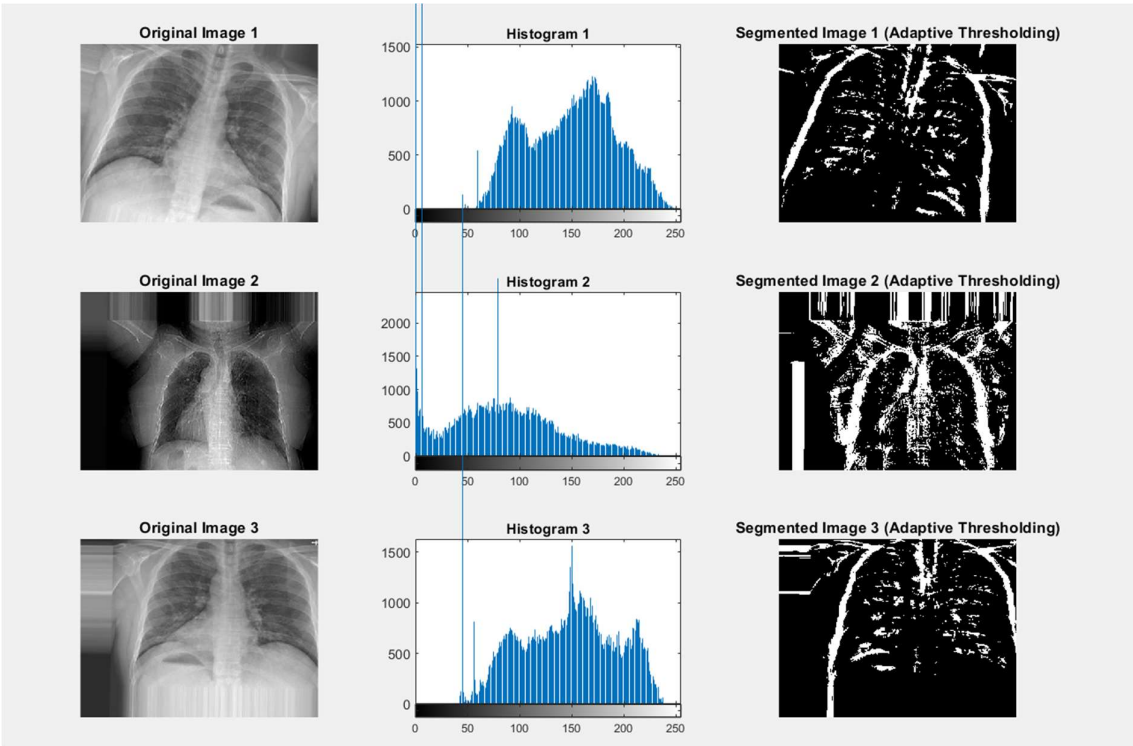
VIRAL:



BACTERIAL:


Original Image 1
Histogram 1
Segmented Image 1 (Adaptive Thresholding)
Original Image 2
Segmented Image 2 (Adaptive Thresholding)
Original Image 3
Histogram 3
Segmented Image 3 (Adaptive Thresholding)

COVID:


Original Image 1
Histogram 1
Segmented Image 1 (Adaptive Thresholding)
Original Image 2
Histogram 2
Segmented Image 2 (Adaptive Thresholding)
Original Image 3
Histogram 3
Segmented Image 3 (Adaptive Thresholding)

Region based segmentation region splitting and merging algorithm. (various region size)

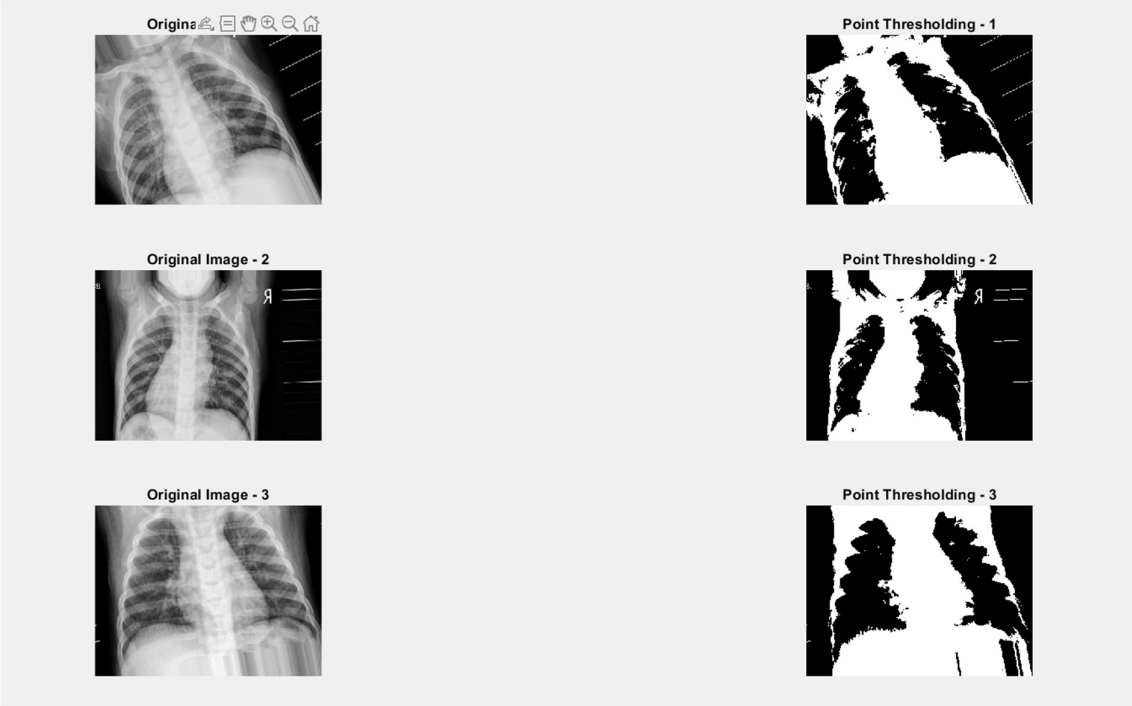
1.Point – thresholding

CODE:

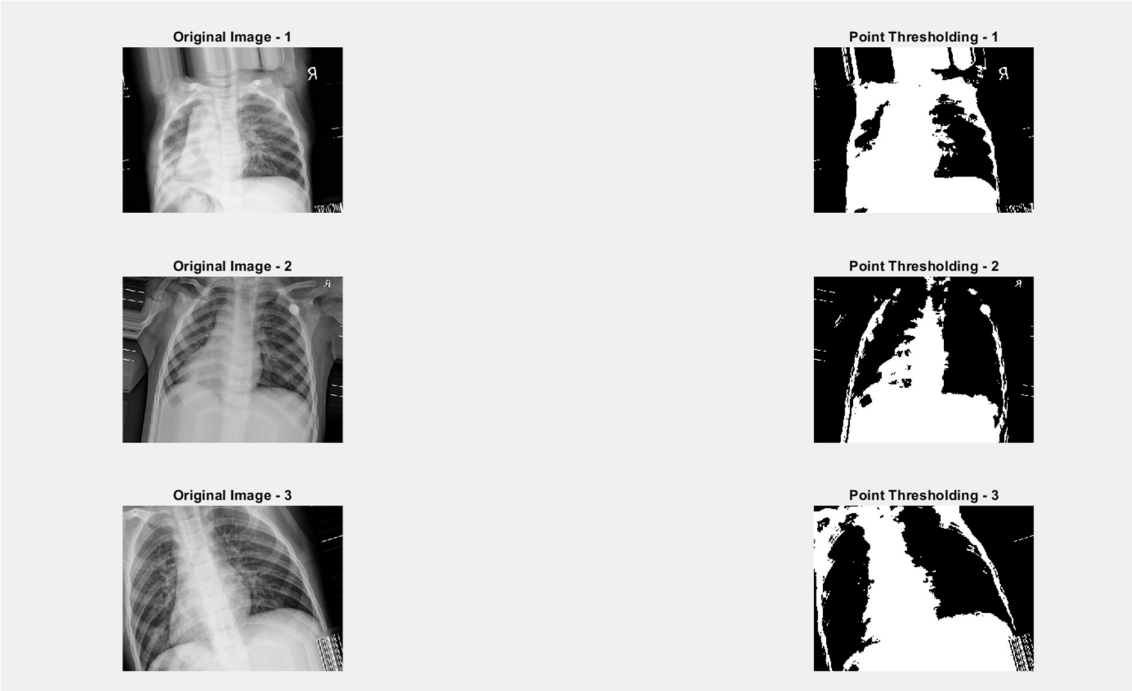
```
imagePaths = {
       'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\covid\covid_1.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\covid\covid_2.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\covid\covid_3.jpeg'
    };

threshold_point = 150;

figure;

for i = 1:length(imagePaths)
    originalImage = imread(imagePaths{i});

    if size(originalImage, 3) == 3
        grayImage = rgb2gray(originalImage);
    else
        grayImage = originalImage;
    end

    binaryImage_point = grayImage > threshold_point;

    subplot(length(imagePaths), 2, 2*i - 1);
    imshow(grayImage);
    title(['Original Image - ', num2str(i)]);

    subplot(length(imagePaths), 2, 2*i);
    imshow(binaryImage_point);
    title(['Point Thresholding - ', num2str(i)]);
end
```

OUTPUT:

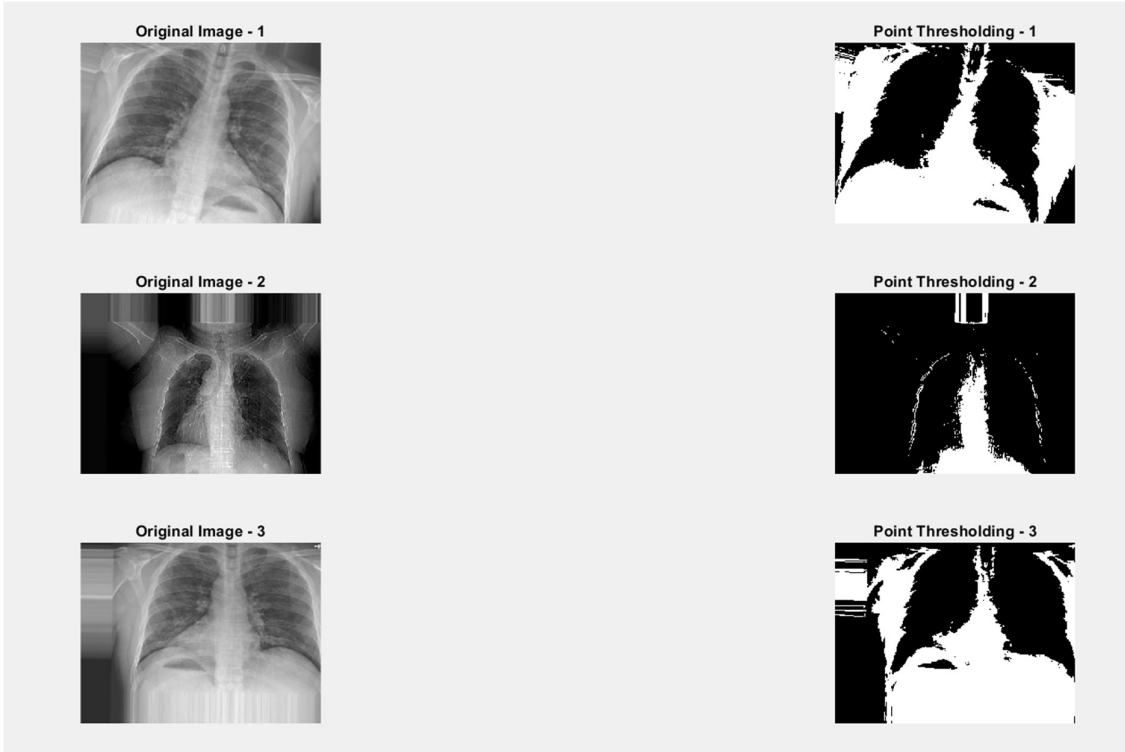
VIRAL:



BACTERIAL:



COVID:



2.Line – thresholding

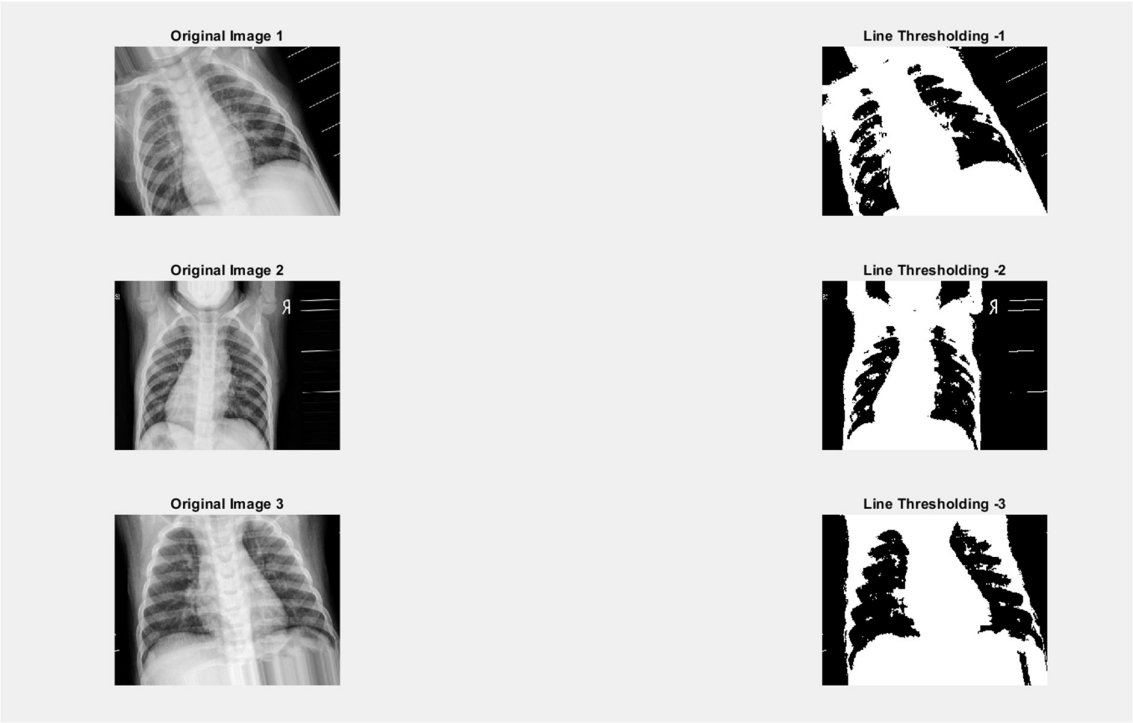
CODE:

```
imagePaths = {
       'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\covid\covid_1.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\covid\covid_2.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\covid\covid_3.jpeg'
};

threshold_line = 130;

figure;

for i = 1:length(imagePaths)
    if exist(imagePaths{i}, 'file') ~= 2
        fprintf('Error: File does not exist - %s\n', imagePaths{i});
        continue;
    end

    originalImage = imread(imagePaths{i});

    if size(originalImage, 3) == 3
        grayImage = rgb2gray(originalImage);
    else
        grayImage = originalImage;
    end
```

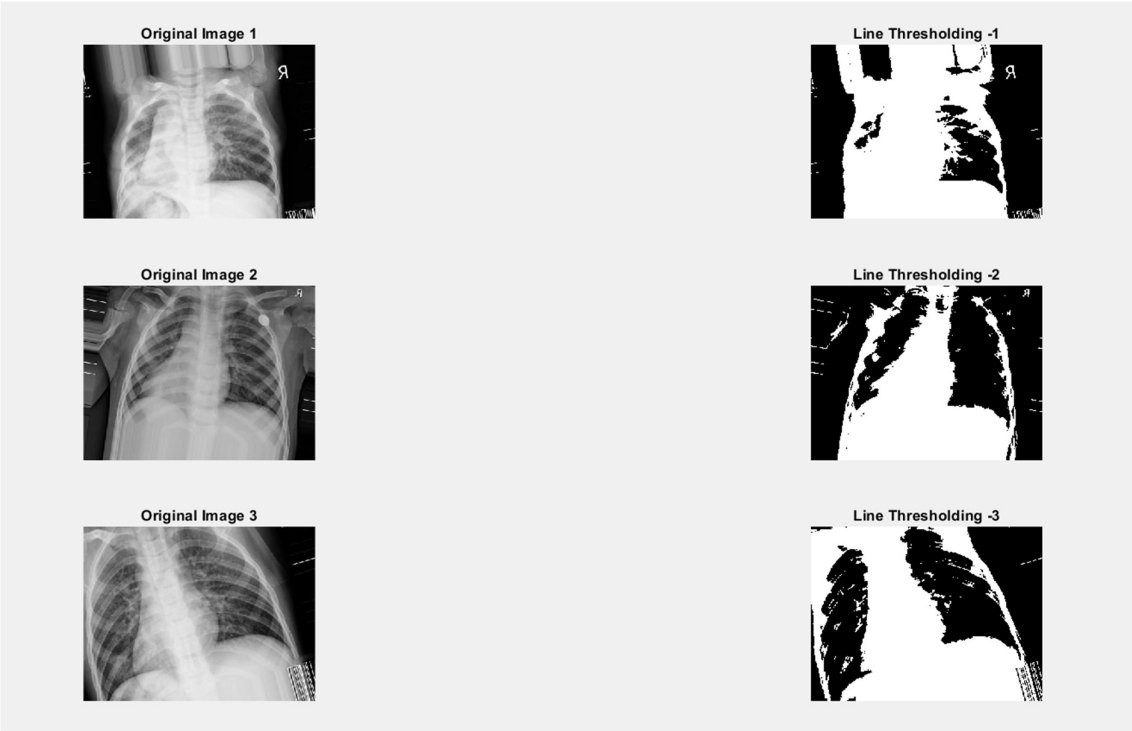
```
    binaryImage_line = grayImage > threshold_line;

    subplot(3, 2, i*2 - 1);
    imshow(originalImage);
    title(['Original Image ', num2str(i)]);

    subplot(3, 2, i*2);
    imshow(binaryImage_line);
    title(['Line Thresholding -', num2str(i)]);
end
```

OUTPUT:

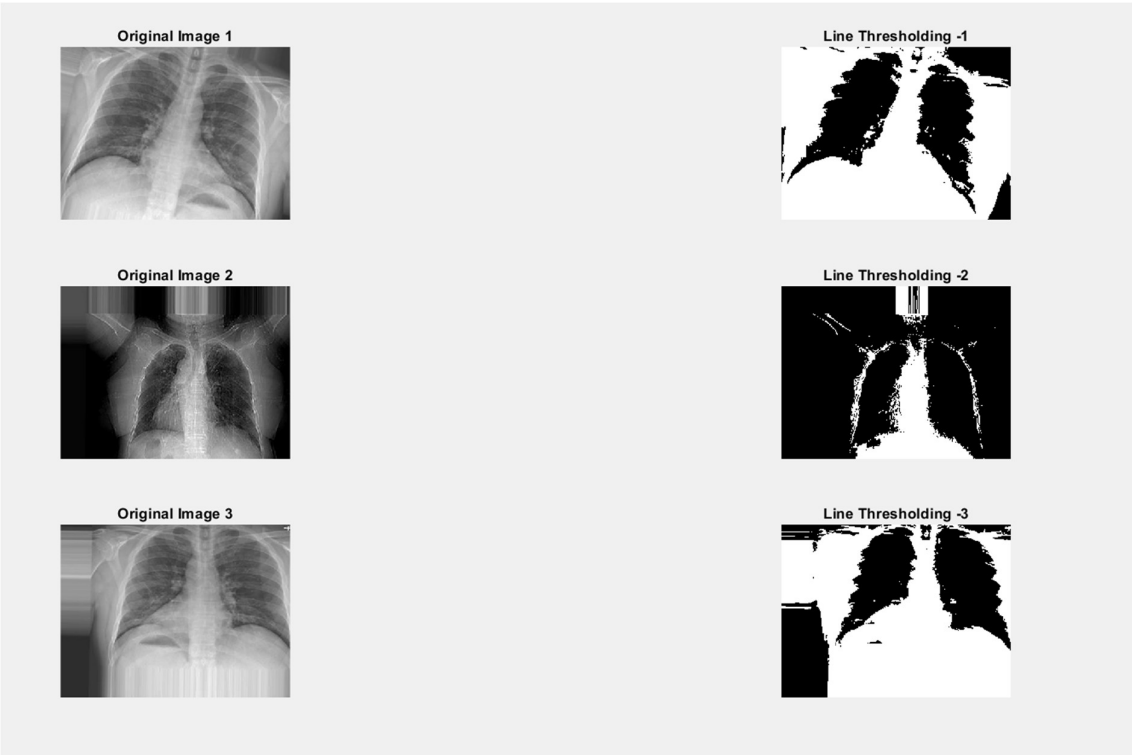
VIRAL:



BACTERIAL:



COVID:



3.Edge – thresholding

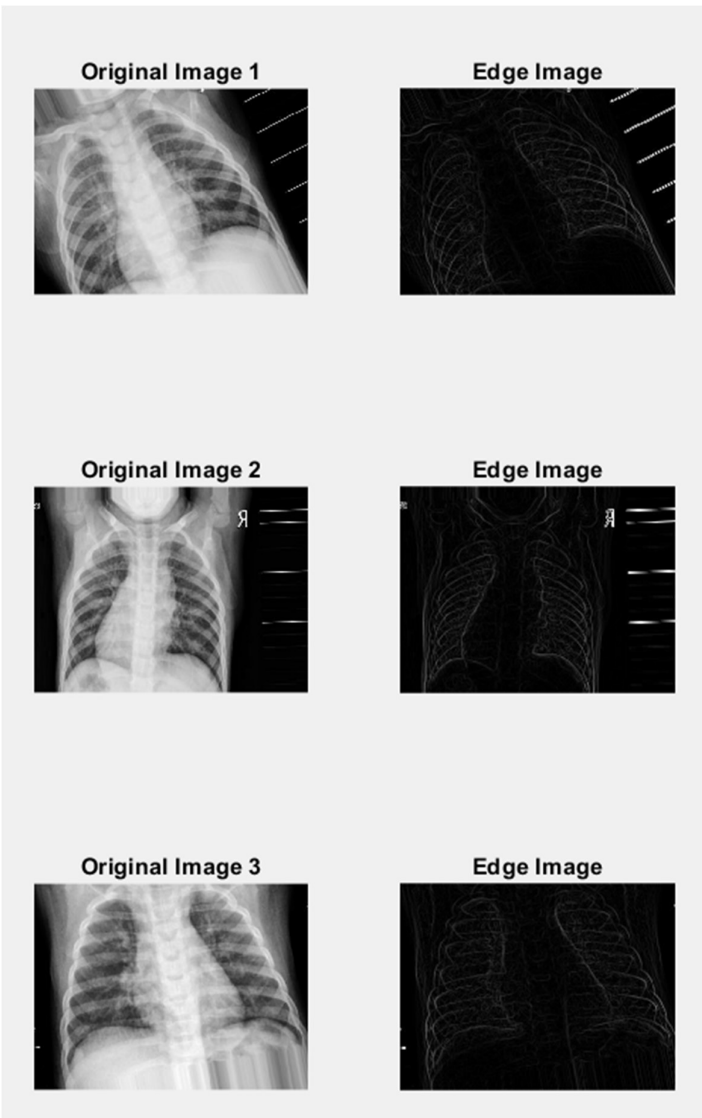
CODE:

```
image_paths = {
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral1.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral2.jpeg',
    'C:\Users\Praveen nallasivam\Desktop\codes\New\Lung Disease
Detection\Dataset\viral\viral3.jpeg'
};

threshold_value = 20;
edge_segmentation(image_paths, threshold_value);

function edge_segmentation(image_paths, threshold_value)
    figure;

    for i = 1:length(image_paths)
        % Step 1: Compute edge image (using Sobel)
        img = imread(image_paths{i});

        % Check if the image is color (RGB)
        if size(img, 3) == 3
            img = rgb2gray(img); % Convert to grayscale if the image is in color
        end

        % Compute the magnitude of the gradient using Sobel
        [gradient_x, gradient_y] = imgradientxy(img, 'sobel');
        edge_image = sqrt(gradient_x.^2 + gradient_y.^2);

        % Step 2: Specify a threshold value
        % Step 3: Threshold the image to produce a binary image
        binary_mask = edge_image > threshold_value;

        % Step 4: Compute histogram using pixels corresponding to 1-valued pixels
in binary_mask
        histogram_values = img(binary_mask);

        % Step 5: Global segmentation using Otsu's method
        global_threshold = graythresh(img);

        % Display the images
        subplot(length(image_paths), 6, (i-1)*6 + 1);
        imshow(img);
        title(['Original Image ', num2str(i)]);

        subplot(length(image_paths), 6, (i-1)*6 + 2);
        imshow(edge_image, []);
        title('Edge Image');

        subplot(length(image_paths), 6, (i-1)*6 + 3);
        imshow(binary_mask);
        title('Binary Mask');

        subplot(length(image_paths), 6, (i-1)*6 + 4);
        imshow(imbinarize(img, global_threshold));
        title('Global Segmentation');

        subplot(length(image_paths), 6, (i-1)*6 + 5);
```

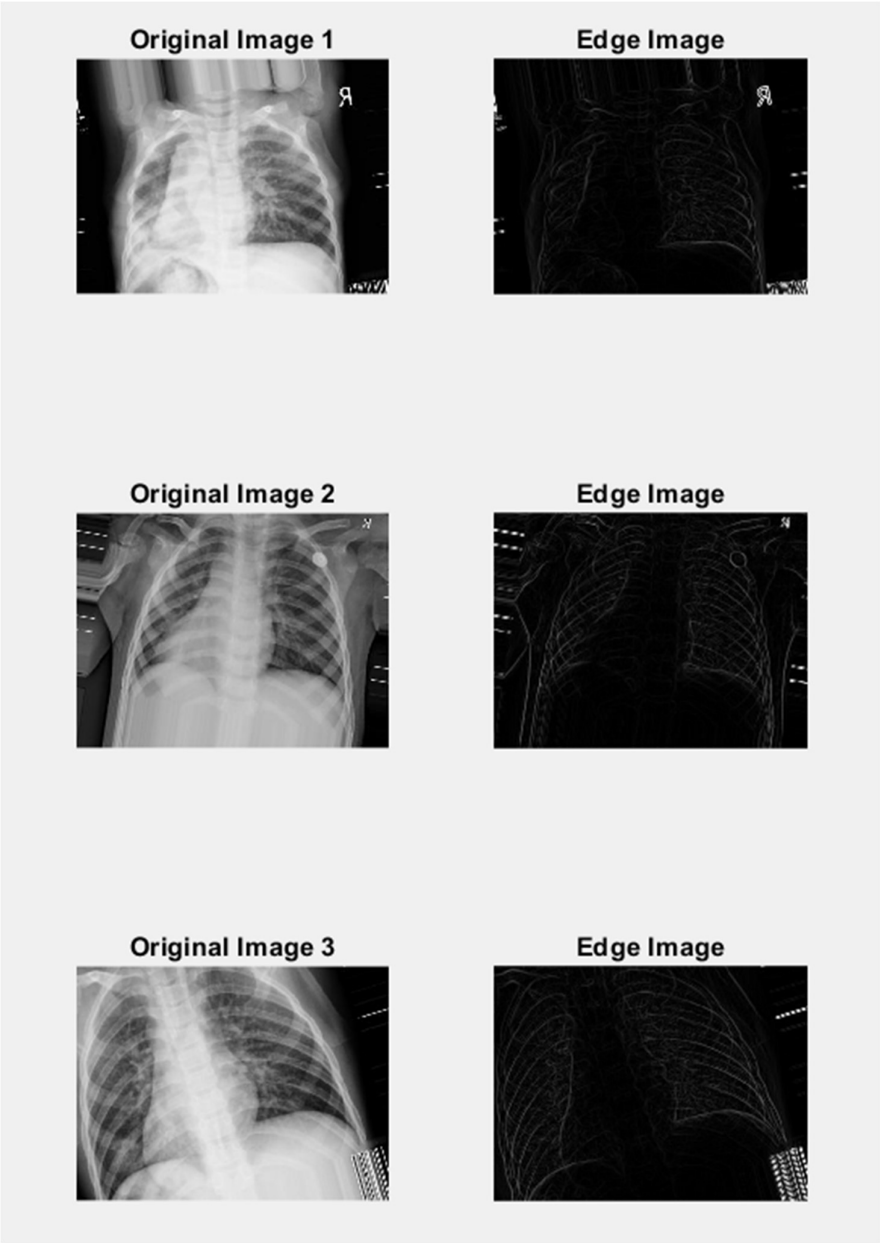
```
        imshow(gradient_x);
        title('Horizontal Gradient');

        subplot(length(image_paths), 6, (i-1)*6 + 6);
        imshow(gradient_y);
        title('Vertical Gradient');
    end

    sgtitle('Edge Segmentation in MATLAB');
end
```

OUTPUT:

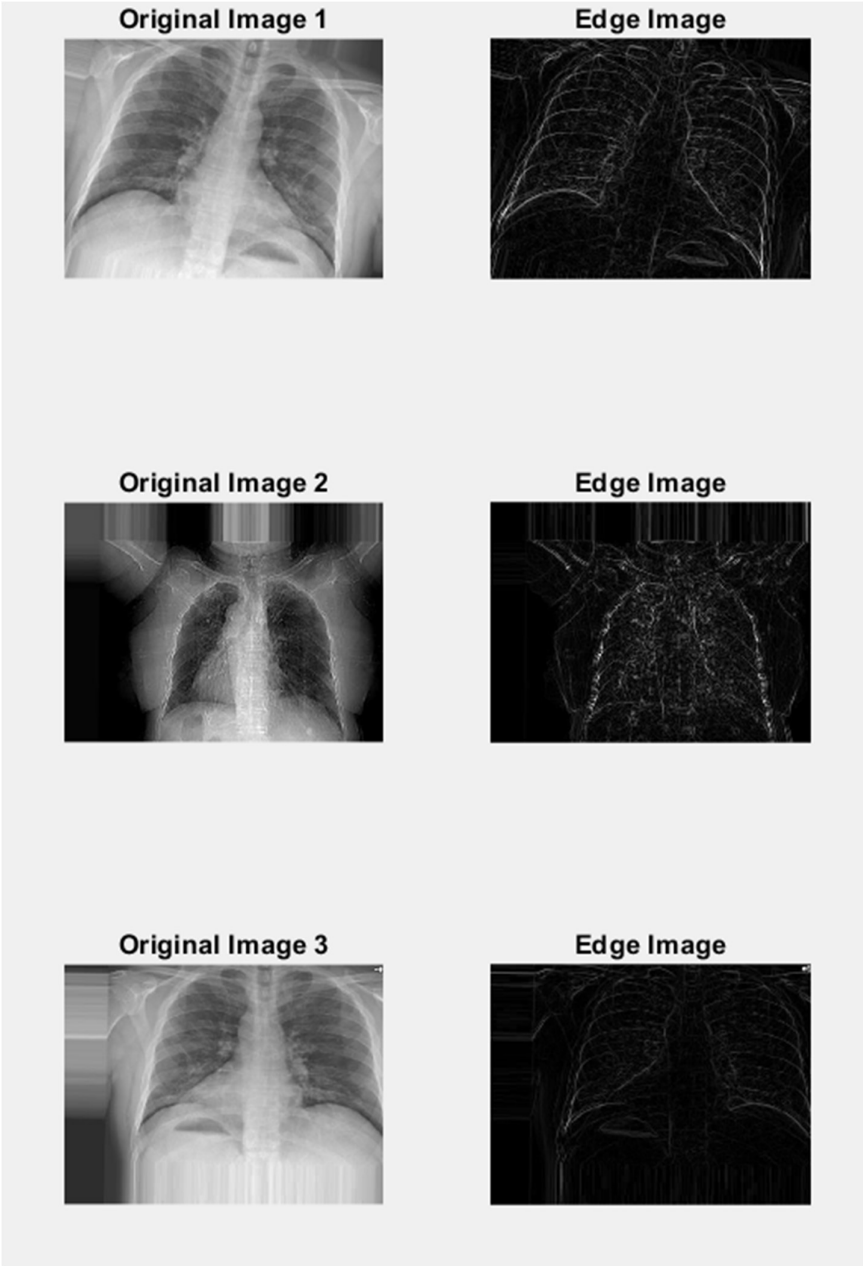
VIRAL:



BACTERIAL:



COVID:


Original Image 1
Edge Image
Original Image 2
Edge Image
Original Image 3
Edge Image