In [1]:

```
import numpy as np
import pandas as pd #for processing csv files
import skimage, os #stores images as 3D numpy array
from skimage.morphology import ball, disk, dilation, binary_erosion, remove_small_objects, erosion, closing, reconstruction, binary_closing
from skimage.measure import label, regionprops, perimeter
from skimage import measure, feature
from skimage import data
import matplotlib.pyplot as plt
from skimage import measure
from imutils import contours
import cv2
from scipy.ndimage.measurements import label
from scipy.ndimage import label
from matplotlib import cm
import scipy.ndimage
from PIL import Image, ImageFilter
from PIL import Image, ImageOps
import os
```

importing images

In [2]:

```
def load_images_from_folder(folder):
    paths = []
    images = []
    filenames=[]

    for filename in os.listdir(folder):
        filenames.append(filename)
        path = os.path.join(folder,filename)
        image = Image.open(path)

        if path is not None:
            paths.append(path)
            images.append(image)

    return images, filenames
```

Creating a function that creates a dictionary of manual values for tumor to lung area ratio for each file

In [3]:

```
#we calculated the area ratio by using software imageJ.
#for each picture, we cropped out the area of the lung and we cropped out the
 tumor area
#the software return the image area
manual_training=[2,8,5,6,7,4,4,7,5,4,2]
manual_cross_validation=[12,6,3,5,10,15,1,5,7,3,1]
manual_test=[1,6,4,3,4,5,5,7,1,4,3]
folder1='Data\Training'
folder2='Data\Cross_Validation'
folder3='Data\Test'
def manual_dict(manual, filenames):
    dictionary= {}
    for i in range(len(filenames)):
        dictionary[filenames[i]] = manual[i]

    return dictionary
folder1='Data\Training'
f1=load_images_from_folder(folder1)[1]
f2=load_images_from_folder(folder2)[1]
f3=load_images_from_folder(folder3)[1]
training_dictionary =manual_dict(manual_training, f1)
cross_validation_dictionary=manual_dict(manual_cross_validation, f2)
test_dictionary=manual_dict(manual_test, f3)
```

in the following cell we want to obtain an image the contains an outline only of the tumors in image without background or the rest of the organ

In [4]:
```
#function we learnt in class: subtracts two images
def minus(im1,im2):
    mat1=im1.load()
    mat2=im2.load()
    w, h = im1.size
    out = Image.new(mode='L', size=(w, h))
    out_mat = out.load()
    for x in range(w):
        for y in range(h):
            out_mat[x,y] = mat1[x,y]-mat2[x,y]
    return out
#in this function, the image is first split to seperatae colour bands then subtracted via minus
def created_subtracted_img(im):
    split_image= Image.Image.split(im)
    im_green=split_image[1] #this image contains only green bands

    im_blue=split_image[2] #this image contains only blue bands

    subtracted= minus(im_green,im_blue)
    #when blue is subtracted from green, only the bright pixel from original image remain showing the tumor
    return subtracted


#in this cell we obtained an image with a relatively black background where only the tumors appearing
```

in the following cell we will present the subtracted images after it has been filtered such that the image has been denoised by filtering and labeled with high contrast

In [5]:
```
def display_labeled(labeled):
    im=Image.fromarray(np.uint8(cm.gist_earth(labeled/np.max(labeled))*255))
    return im
def image_displayed(im):
    im1=im.filter(ImageFilter.BLUR)
    im2 = im1.filter(ImageFilter.MinFilter(11))
    im3 = im2.filter(ImageFilter.MinFilter)
    disp=display_labeled(im3.filter(ImageFilter.BLUR))
    return disp
```

the following cell will find the area of the tumor

In [6]:

```
#we worked with otsu method to segment the images in the folllowing code block
def otsu_thrd(im):
    ''' return the optimal threshold for a 256 gray level image im '''
    width, height = im.size
    hist = im.histogram(im)
    var_max = 0
    for t in range(1,255): #t=0 and t=255 will yield 0 anyway
        back = sum(hist[0:t+1])
        fore = sum(hist[t+1:256])
        if back==0 or fore==0:
            continue
        mean_back = sum(hist[i]*i for i in range(t+1)) / back
        mean_fore = sum(hist[i]*i for i in range(t+1,len(hist))) / fore
        # Calculate Between Class Variance
        var_between = back * fore * (mean_back - mean_fore)**2
        # Check if new maximum found
        if (var_between > var_max):
            var_max = var_between
            threshold = t

    return threshold

def segment(im, thrd):
    ''' Binary segmentation of image im by threshold thrd '''
    width, height = im.size
    out = Image.new(mode='1', size=(width, height))
    mat = im.load()
    out_mat = out.load()
    for x in range(width):
        for y in range(height):
            if mat[x,y] <= thrd:
                out_mat[x,y] = 1   #white
            else:
                out_mat[x,y] = 0   #black
    return out
#the following function blurs an image
def filter_image(im):
    im1=im.filter(ImageFilter.BLUR)
    return im1

def segmenting_images(img_to_segment):

    thresh= otsu_thrd(img_to_segment.convert('L'))
    segmented_image= segment(img_to_segment.convert('L'),thresh)
    return segmented_image
#labeling the segmented image  will be done
#labels = measure.label(np.array(segmented_image), background=0)

#the following code is to graph the data obtained thus far


#plots = {'Original': image, 'Labeled': subtracted_labeled3, 'Segmented':segme
nted_image } #plotting the progress
#fig, ax = plt.subplots(1, len(plots))
#for n, (title, img) in enumerate(plots.items()):
    cmap = plt.cm.gnuplot if n == len(plots) - 1 else plt.cm.gray
    ax[n].imshow(img, cmap=cmap)
    ax[n].axis('off')
    ax[n].set_title(title)
#plt.show(fig)

#in order to obtain area of tumors, we used regionprops, a function from skimage
#props = measure.regionprops(labels)

#for prop in props:
    #print('Label: {} >> Object Area: {}'.format(prop.label, prop.area))
```

the following cell we have two goals:

1. determine if cancer is metastatic or benign
2. we will pick out the largest tumor (if there are more than one) and determine the cancer stage

In [7]:

```
def find_total_tumor_area(im,label):

    width, height = im.size
    total_pixels = width * height

    props = measure.regionprops(label)
    area = [prop.area for prop in props]

    pixels_background= max(area)

    total_tumor= total_pixels -pixels_background

    return total_tumor
```

In [8]:

```
def lung_area (im,tumor_area):
    seg=segmenting_images(im)

    lung_array = np.asarray(im, dtype=bool)

    lung_area = np.sum(lung_array)

    #the area of the lung in total will be the largest in area in segmented original image in props_lung + total tumor area
    lung_area+= tumor_area
    return lung_area
```

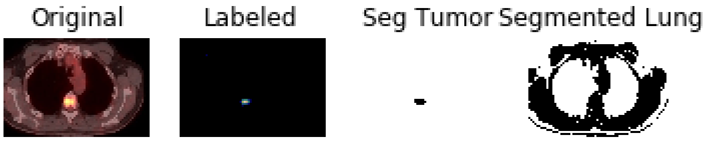
Our next goal is to find the total area of the lung

In [9]:

```
def main_func(folder, manual_values):
    image_collection=load_images_from_folder(folder)[0]
    filename_collection=load_images_from_folder(folder)[1]
    list_of_errors=[]
    for i in range (len(image_collection)):

        subtracted=created_subtracted_img(image_collection[i])
        coloured_label_im=image_displayed(created_subtracted_img(image_collection[i]))
        segmented_img=segmenting_images(created_subtracted_img(image_collection[i]))
        matrix=np.array(segmented_img)
        labels = measure.label(matrix, background=255) #labeling the segmented image
        print()
        print()
        print()
        print()
        print('The Following Results Obtained For Patient Number:' + str(filename_collection[i]))
        plots = {'Original': image_collection[i], 'Labeled': coloured_label_im, 'Seg Tumor':segmented_img , 'Segmented Lung' : segmenting_images(image_collection[i]) } #plotting the progress
        fig, ax = plt.subplots(1, len(plots))

        for n, (title, img) in enumerate(plots.items()):

            cmap = plt.cm.gnuplot if n == len(plots) - 1 else plt.cm.gray
            ax[n].imshow(img, cmap=cmap)

            ax[n].axis('off')
            ax[n].set_title(title)

        plt.show(fig)
        #in order to obtain area of tumors, we used regionprops, a function from skimage
        props = measure.regionprops(labels)
        #now we will print the results for the values obtain by this algorithm

        total_tumor_area_in_lung= find_total_tumor_area(segmented_img, labels)
        area_of_lung=lung_area(image_collection[i],total_tumor_area_in_lung)
        ratio =round(total_tumor_area_in_lung/area_of_lung *100,0)
        print("The area of tumor in pixels is "+ str(total_tumor_area_in_lung) + " and the area of total lung is " + str(area_of_lung))
        print('The ratio of tumor area to lung area in % is ' + str(ratio))

        #in order to check if the algorithm works correctly, we manually calculates the area of tumor and lung
        #now we must compare the manual values with the algorithmic values
        results= [ratio, manual_values[filename_collection[i]]]

        error = round(abs((results[1] - results[0])), 0 )
        print("The Error between the algorithmic value and the value we measured with ImageJ is " + str(error) + "%")
        if error <=3:
            print ('The Algorithm Was Accurate')
        else:
            print( 'The Algorithm Detected The Tumor With Inaccurate Size Esti
mation')
        list_of_errors.append(error)
    success_list= 0
    unsuccess_list=0
    for i in range (len(list_of_errors)):
        if list_of_errors[i] >3: #if the error is greater than the standard de
viation of the entire sample pool
            unsuccess_list+=1
        else:
            success_list +=1

    Success= ['High Accuracy', 'Low Accuracy']
    Success_Rates= [ (success_list/11) *100 ,(unsuccess_list/11)*100 ]
    Accurate= round(Success_Rates[0],1)
    inaccurate= round(Success_Rates[1],1)
    print ("The percent of highly accurate results is " + str(Accurate) + '% a
nd the percent of inaccurate results is ' + str(inaccurate) + '%')
    plt.bar(Success,Success_Rates)
    plt.title('Accuracy')
    plt.xlabel('')
    plt.ylabel('Accuracy Rates in %')

    plt.show()
```

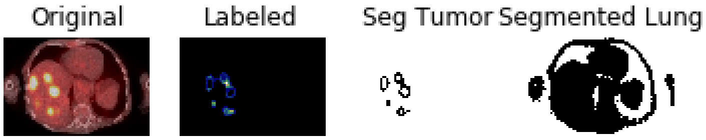
In [10]:
```
folder1='Data\Training'
main_func(folder1,training_dictionary )
```

```
The Following Results Obtained For Patient Number:001.jpg
```



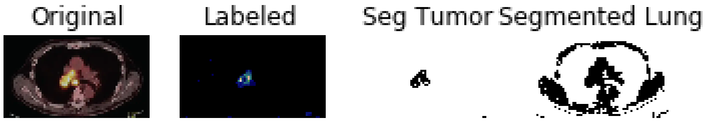
```
The area of tumor in pixels is 609 and the area of total lung is 366448
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 2.0%
The Algorithm Was Accurate
```

```
The Following Results Obtained For Patient Number:002.jpg
```



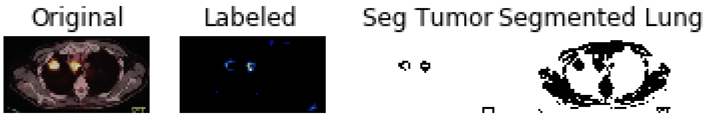
```
The area of tumor in pixels is 4682 and the area of total lung is 351664
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 7.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:004.png
```



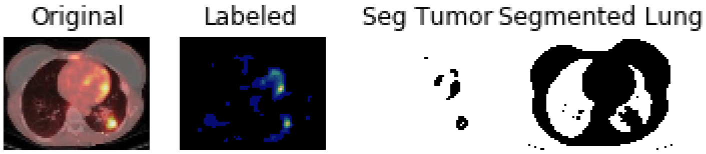
```
The area of tumor in pixels is 3383 and the area of total lung is 578818
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 4.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:005.png
```



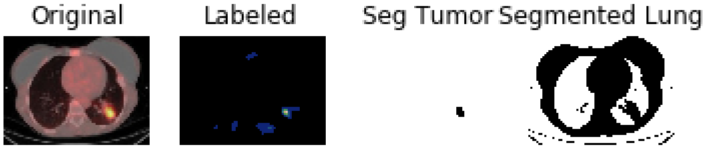
```
The area of tumor in pixels is 2202 and the area of total lung is 528731
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 6.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:012.jpg
```



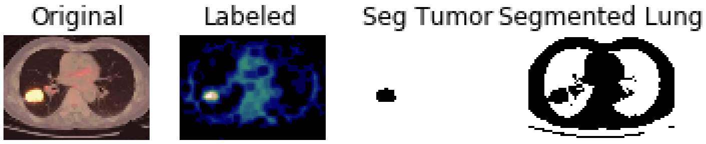
```
The area of tumor in pixels is 901 and the area of total lung is 109904
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 6.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:013.jpg
```



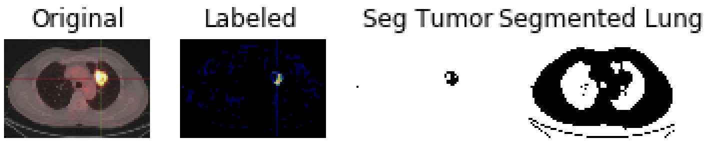
```
The area of tumor in pixels is 203 and the area of total lung is 123292
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 4.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:016.jpg
```



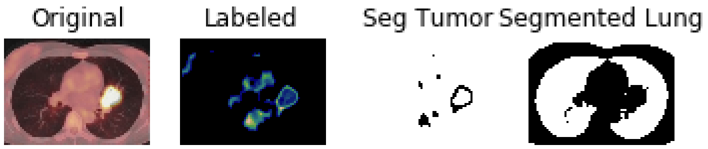
```
The area of tumor in pixels is 1284 and the area of total lung is 271172
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 4.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:024.jpg
```



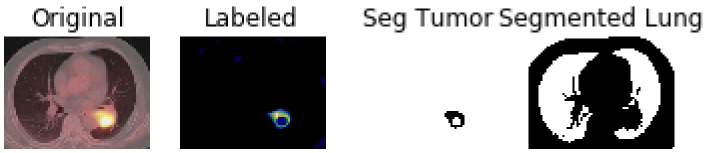
```
The area of tumor in pixels is 6622 and the area of total lung is 1325269
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 7.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:025.png
```

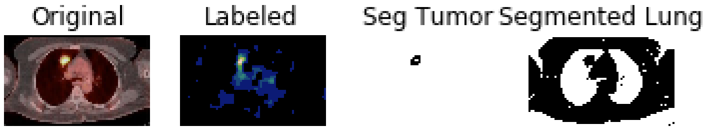


```
The area of tumor in pixels is 11747 and the area of total lung is 1223861
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 4.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

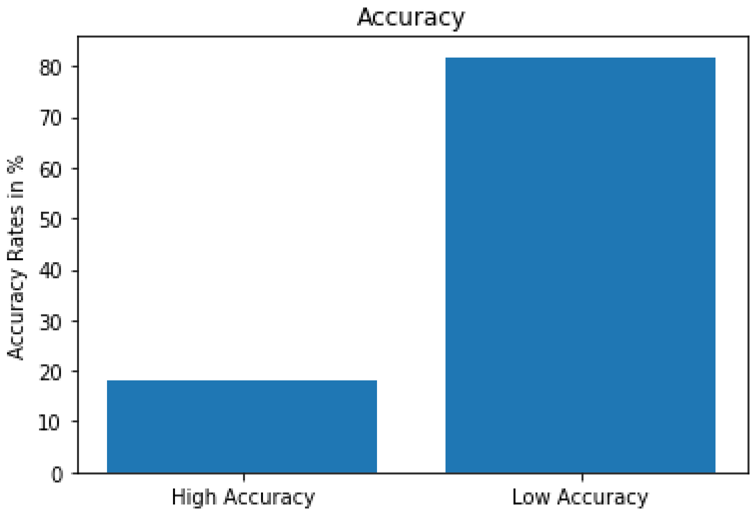
```
The Following Results Obtained For Patient Number:030.png
```



```
The area of tumor in pixels is 9191 and the area of total lung is 2267631
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 4.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:033.jpg
```



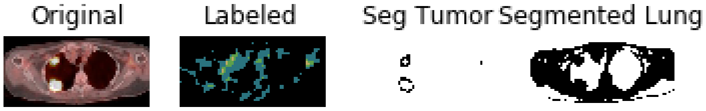
```
The area of tumor in pixels is 398 and the area of total lung is 165999
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 2.0%
The Algorithm Was Accurate
The percent of highly accurate results is 18.2% and the percent of inaccurate
results is 81.8%
```



In [11]:

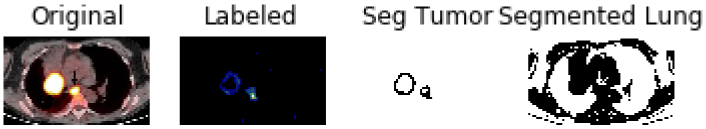
```
folder2='Data\Cross_Validation'
main_func(folder2,cross_validation_dictionary)
```

```
The Following Results Obtained For Patient Number:003.jpg
```



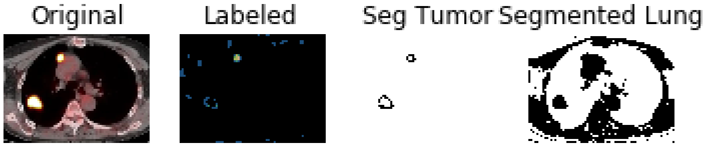
```
The area of tumor in pixels is 1365 and the area of total lung is 126617
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 11.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:006.png
```



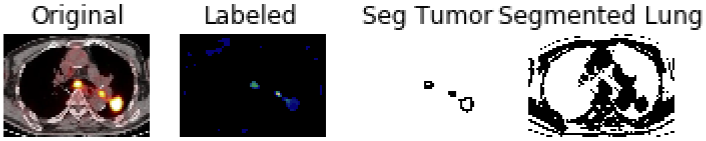
```
The area of tumor in pixels is 3421 and the area of total lung is 321696
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 5.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:008.png
```

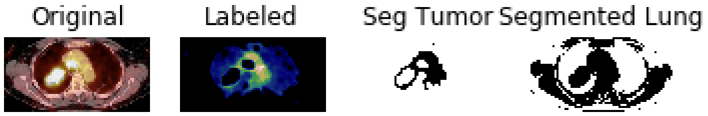


```
The area of tumor in pixels is 1739 and the area of total lung is 422908
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 3.0%
The Algorithm Was Accurate
```

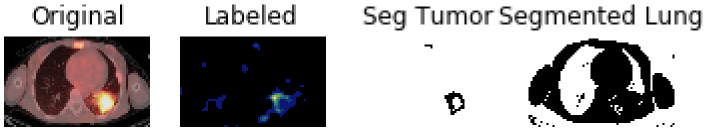
```
The Following Results Obtained For Patient Number:010.png
```



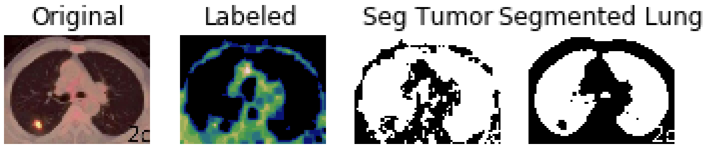
```
The area of tumor in pixels is 1859 and the area of total lung is 334744
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 4.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:011.jpg
```



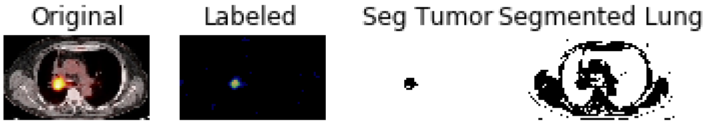
```
The area of tumor in pixels is 26308 and the area of total lung is 686653
The ratio of tumor area to lung area in % is 4.0
The Error between the algorithmic value and the value we measured with ImageJ
is 6.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:014.jpg
```



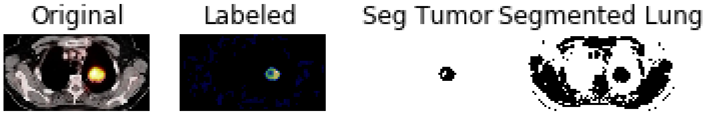
```
The area of tumor in pixels is 977 and the area of total lung is 105625
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 14.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:017.jpg
```



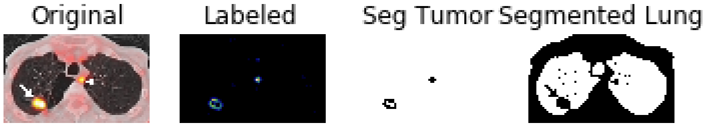
```
The area of tumor in pixels is 25523 and the area of total lung is 257112
The ratio of tumor area to lung area in % is 10.0
The Error between the algorithmic value and the value we measured with ImageJ
is 9.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:018.jpg
```



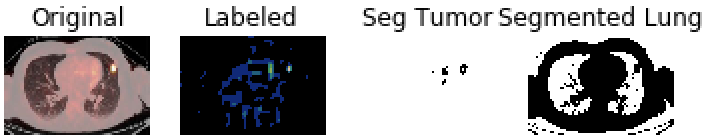
```
The area of tumor in pixels is 707 and the area of total lung is 151979
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 5.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:019.jpg
```



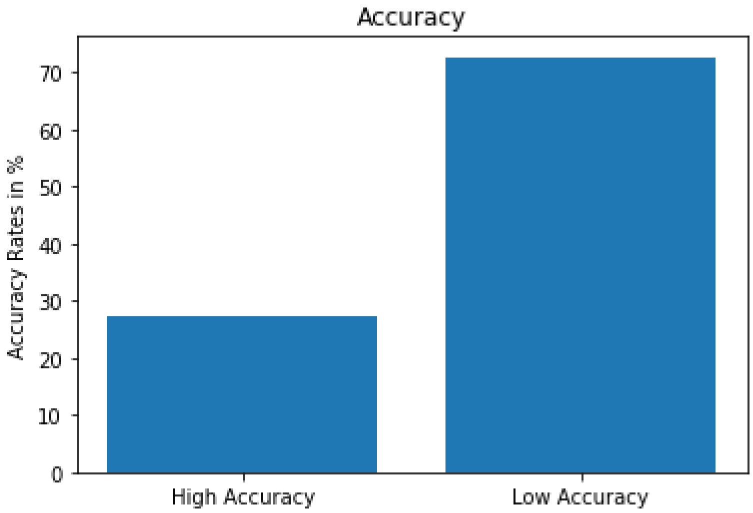
```
The area of tumor in pixels is 5043 and the area of total lung is 771875
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 6.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:023.jpeg
```



```
The area of tumor in pixels is 4904 and the area of total lung is 1571982
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 3.0%
The Algorithm Was Accurate



The Following Results Obtained For Patient Number:031.jpg
```



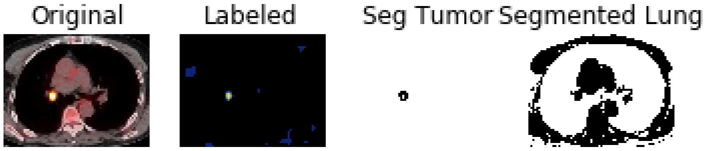
```
The area of tumor in pixels is 512 and the area of total lung is 196115
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 1.0%
The Algorithm Was Accurate
The percent of highly accurate results is 27.3% and the percent of inaccurate
results is 72.7%
```



In [12]:

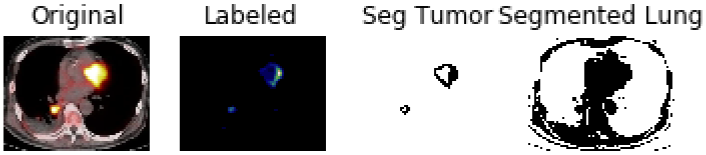
```
folder3='Data\Test'
main_func(folder3,test_dictionary)
```

```
The Following Results Obtained For Patient Number:007.png
```



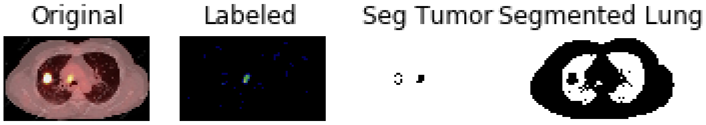
```
The area of tumor in pixels is 491 and the area of total lung is 392391
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 1.0%
The Algorithm Was Accurate
```

```
The Following Results Obtained For Patient Number:009.png
```



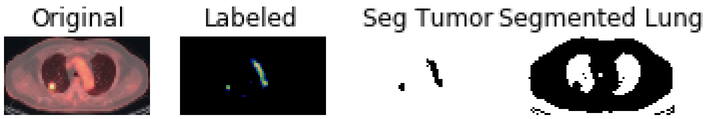
```
The area of tumor in pixels is 2572 and the area of total lung is 308673
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 5.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

```
The Following Results Obtained For Patient Number:015.jpg
```

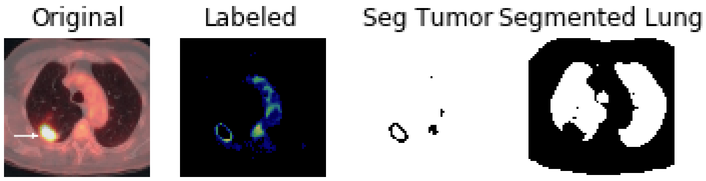


```
The area of tumor in pixels is 3340 and the area of total lung is 753574
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 4.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation
```

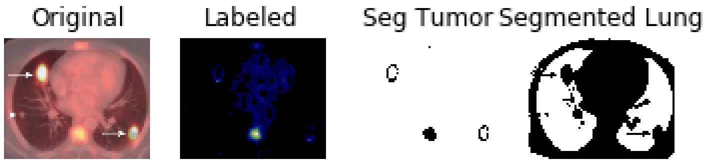
```
The Following Results Obtained For Patient Number:020.jpg
```



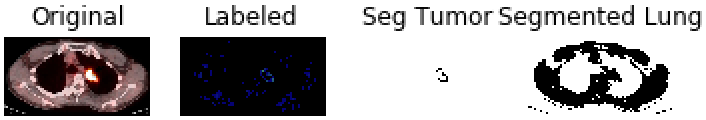
```
The area of tumor in pixels is 2492 and the area of total lung is 394578
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 2.0%
The Algorithm Was Accurate



The Following Results Obtained For Patient Number:021.jpg
```



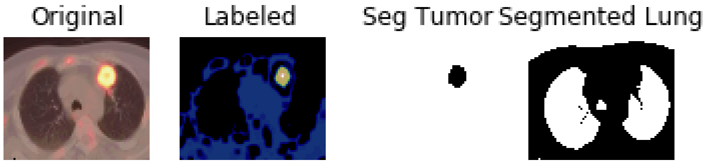
```
The area of tumor in pixels is 8629 and the area of total lung is 1549680
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 3.0%
The Algorithm Was Accurate



The Following Results Obtained For Patient Number:022.jpg
```



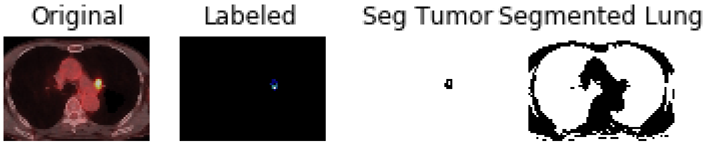
```
The area of tumor in pixels is 4917 and the area of total lung is 590759
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 4.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:026.jpg
```



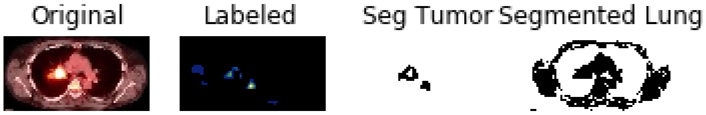
```
The area of tumor in pixels is 6427 and the area of total lung is 1469769
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 5.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:027.png
```



```
The area of tumor in pixels is 3784 and the area of total lung is 886472
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 7.0%
The Algorithm Detected The Tumor With Inaccurate Size Estimation



The Following Results Obtained For Patient Number:028.jpeg
```

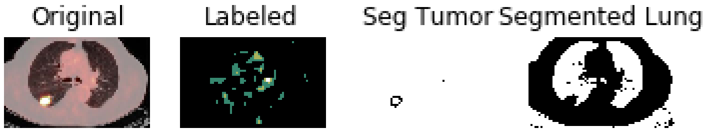


```
The area of tumor in pixels is 1350 and the area of total lung is 901828
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 1.0%
The Algorithm Was Accurate



The Following Results Obtained For Patient Number:029.png
```

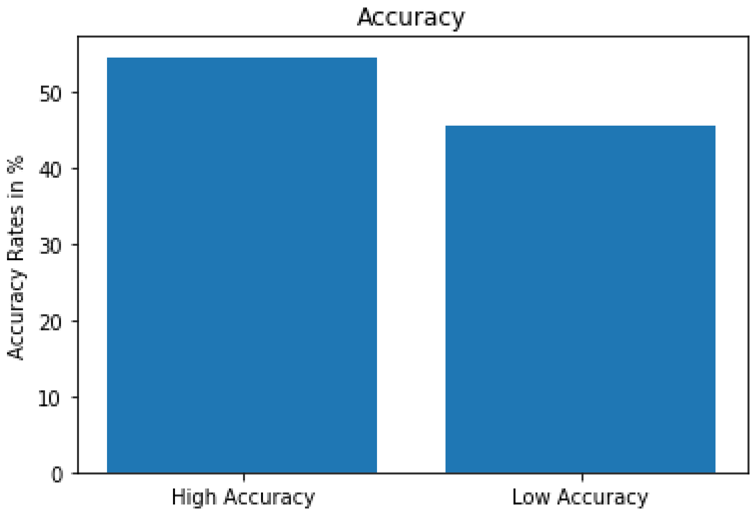


```
The area of tumor in pixels is 960 and the area of total lung is 129746
The ratio of tumor area to lung area in % is 1.0
The Error between the algorithmic value and the value we measured with ImageJ
is 3.0%
The Algorithm Was Accurate
```

```
The Following Results Obtained For Patient Number:032.jpg
```



```
The area of tumor in pixels is 526 and the area of total lung is 185702
The ratio of tumor area to lung area in % is 0.0
The Error between the algorithmic value and the value we measured with ImageJ
is 3.0%
The Algorithm Was Accurate
The percent of highly accurate results is 54.5% and the percent of inaccurate
results is 45.5%
```


Accuracy
Accuracy Rates in %
High Accuracy
Low Accuracy