Medical Image Processing – Homework 3
Rotem Amsalem 207596297

In this exercise I used the following libraries:
- nibabel – to save and load NIFTI images
- numpy – for the computing parts
- skimage – for algorithms for image processing (such as morphological operations)
- Copy - to copy image without changing the source one

Part 1: find an ROI

filterNoiseOnBiggestComponent: this function filtering the noise of the largest connectivity component
Input: labels- array of connectivity components, lungs- int. 1 means we segment the lungs and 0 means we are not
Output: the biggest component without any noise
It does so by applying morphological operations.

| IsolateBody: Isolate the patient's body from the air and scan gantry |
|---|
| Input: CT Scan |
| Output: Body segmentation |
| Algorithm:
 1. Perform thresholding to remove all pixels with gray level (HU) below -500 and above 2000 (keep only those between -500 and 2000)
 2. Filter out noise
 3. Compute the largest connected component |

The body segmentation of each case:

Case	Body Segmentation

1	
2	
3	

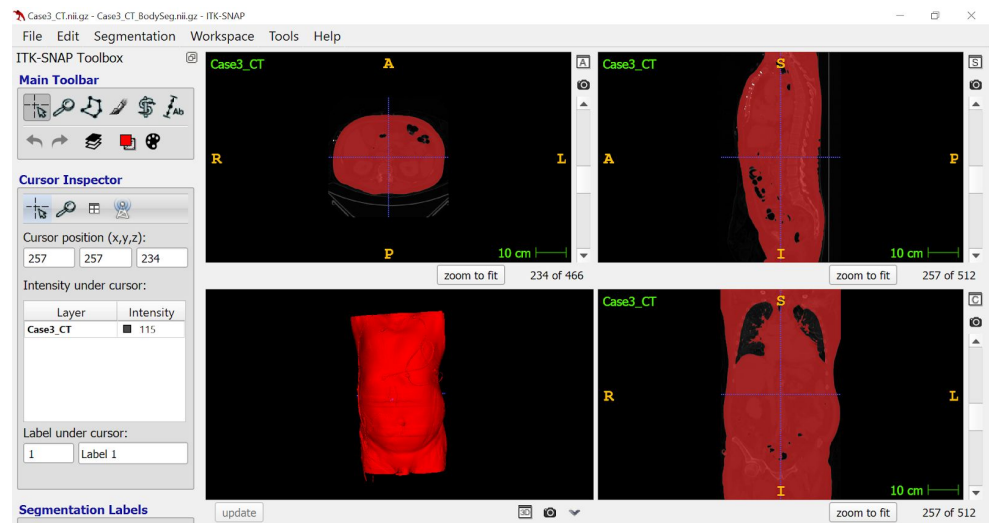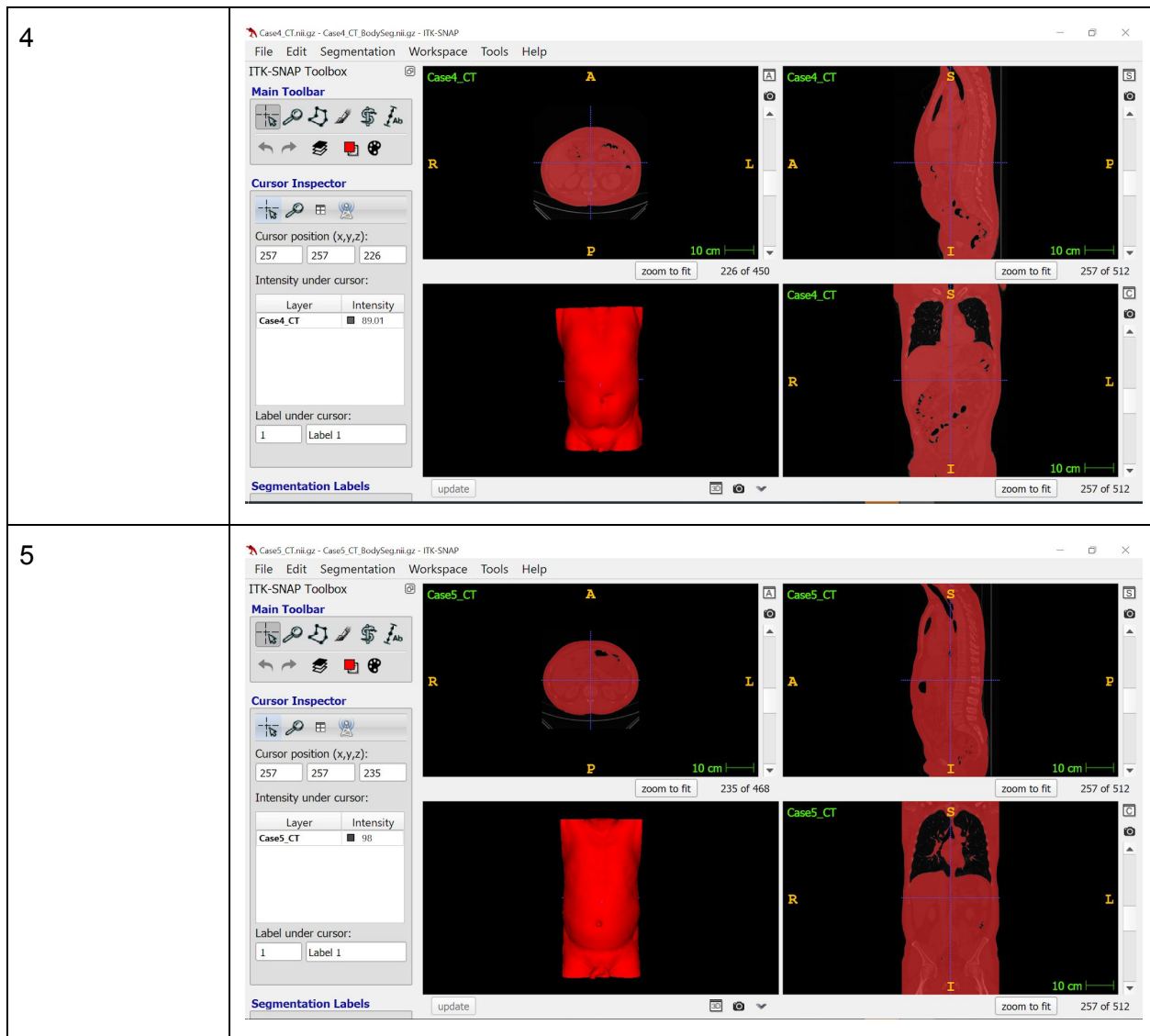|	
4	
5	
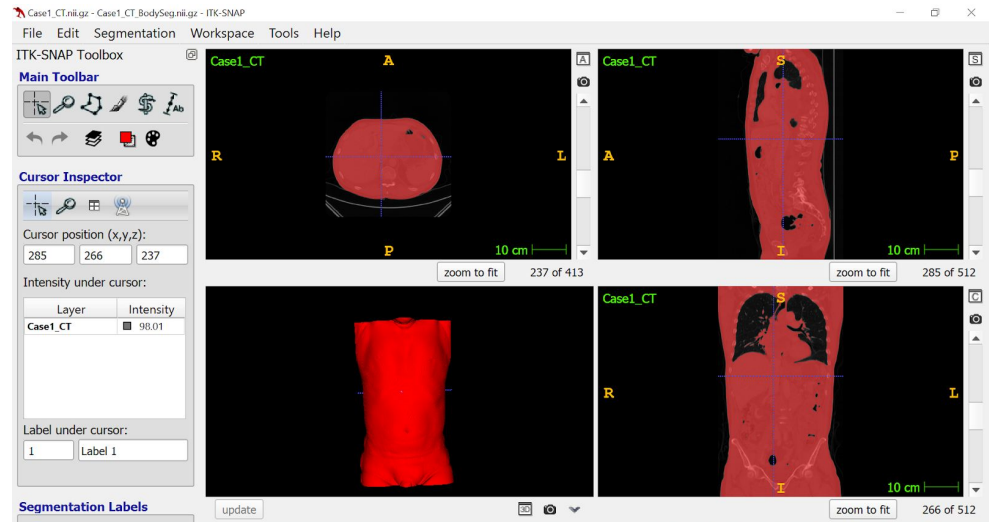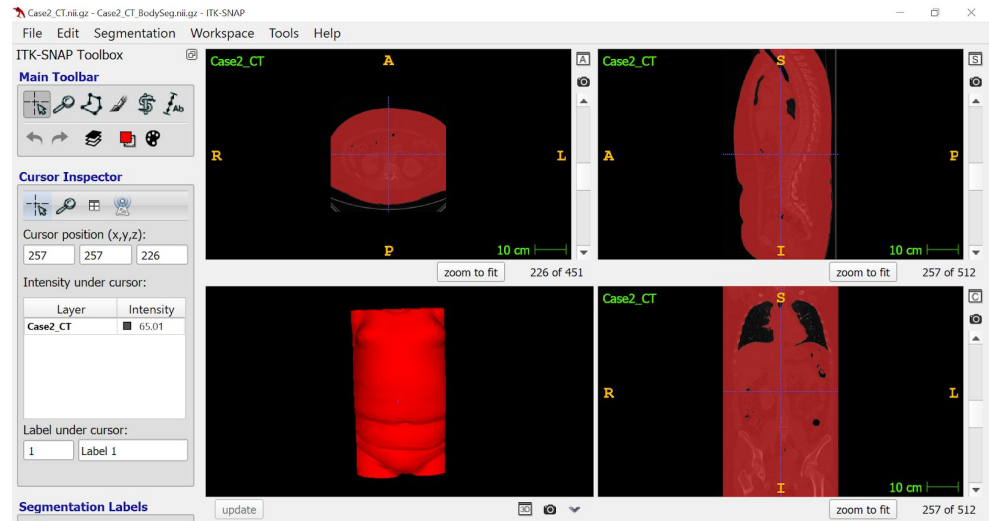
IsolateLungs: Isolate the lungs
Input: body segmentation from IsolateBody
Output: (a) The lungs segmentation
Algorithm: Find the two large cavities inside the body segmentation which correspond to the lungs

The lungs segmentation of each case:

Case	Lungs Segmentation
1	
2	

3	
4	
5	

In the above function, case3 was the only one who didn't fit the ROI, and its ROI didn't fit the other cases so I found a separate ROI for this case.

Part 2: Region Growing Segmentation of the Liver in CT Scans

findingLiverROI - finding an ROI in the liver from which we can sample seeds for the MSRG.
Input: CT Scan, Aorta segmentation
Output: Liver ROI
Algorithm: according to the given aorta segmentation and the lungs segmentation from the previous part - I computed the ROI of the liver

The liver ROI of each case:

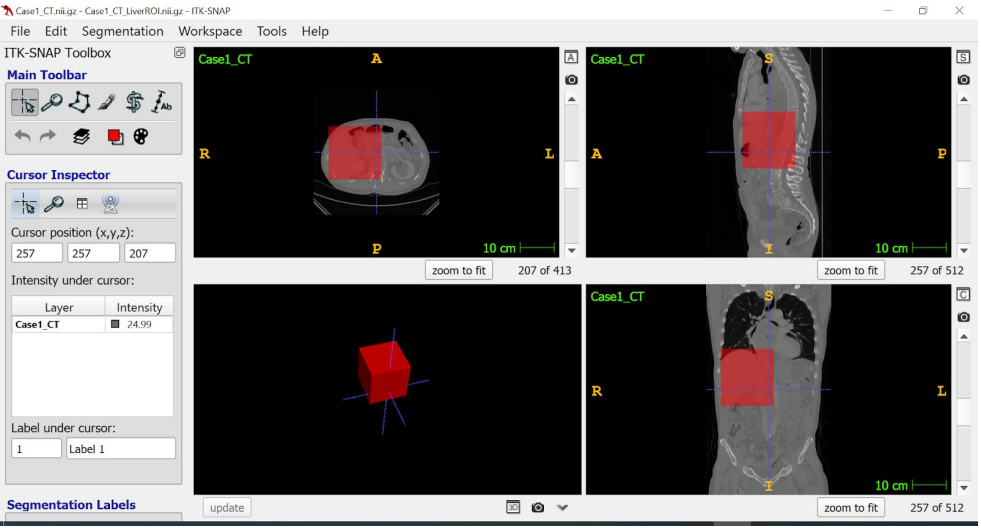| Case	Liver ROI
1	

2	
3	
4	

| 5 | |
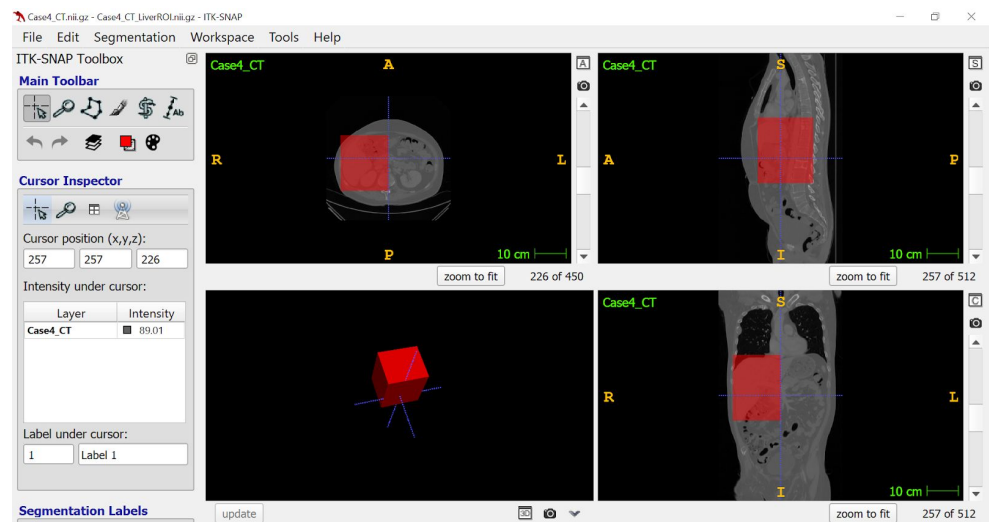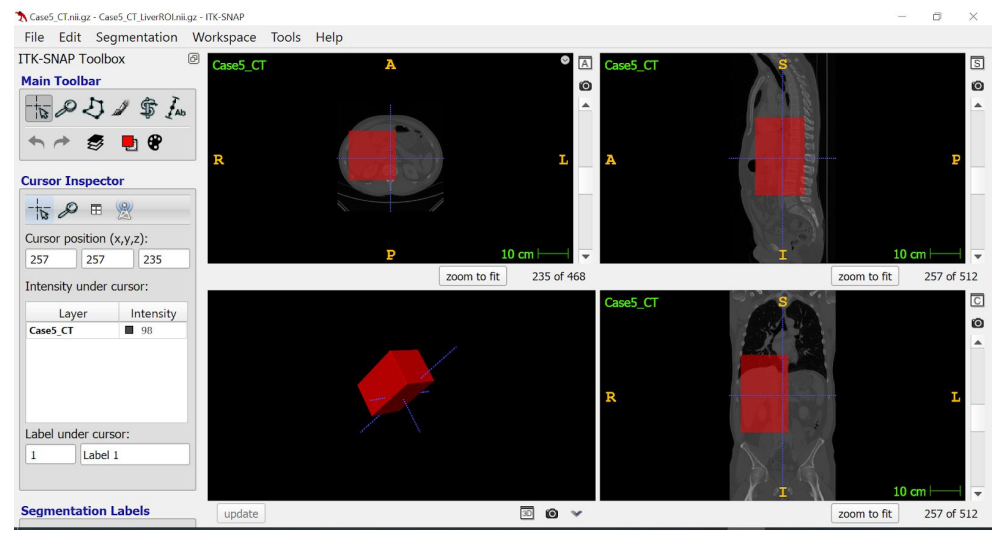
In the above function, case3 and case1 didn't fit the ROI, and them ROI didn't fit the other cases so I found a separate ROI for those cases.
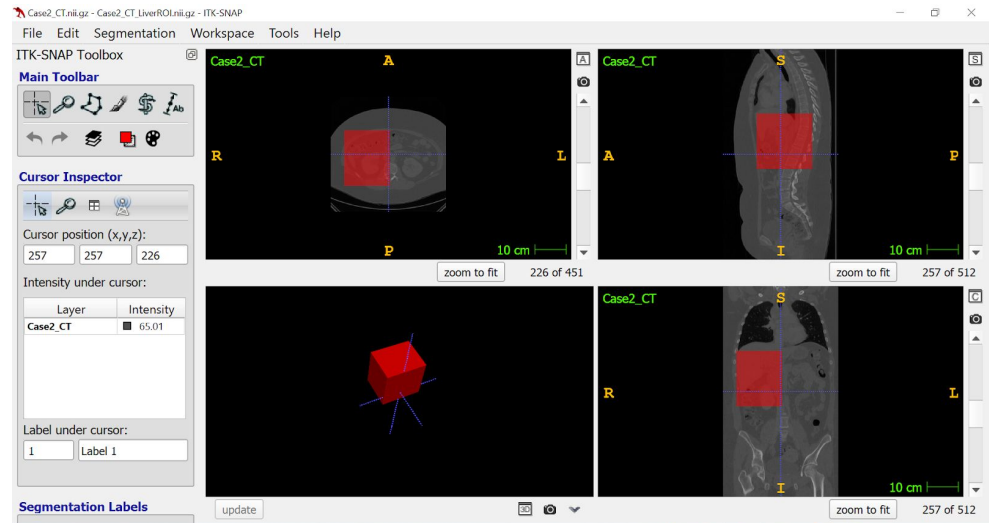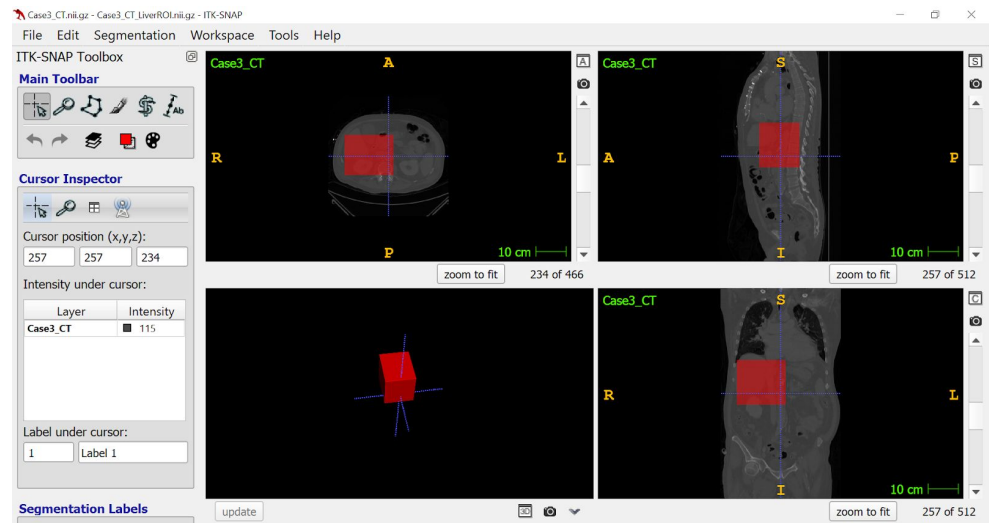
findSeeds
Input: CT Scan and ROI
Output: seeds list
Algorithm: sample seeds from ROI
In this function, at each iteration I chose randomly (using np.random.choice) coordinates (x, y, z) and checked if this value of this voxel is matching the range [-100, 200]. If so, I added this voxel to the seed list.

cleanLiver- this function cleans the liver segmentation and save it
Input: CT scan
Output: Liver segmentation without noise
It does so by applying morphological operations.

| multipleSeedsRG |
| --- |
| Input: CT scan, ROI |
| Output: Liver segmentation |
| Algorithm:
1. Extract N seeds points inside the ROI.
2. Perform Seeded Region Growing with N initial points |
| In this function I used the following algorithm:
1. I found seeds by activating the findSeeds function.
2. I created an array of all the voxels I already visit at
3. I found the seeds's neighbors by creating a cube around each voxel and subtracting the seeds themselves and remains with the neighbors only. Then I subtract the seeds which are in the visit array and updated the visit array to contain all the neighbors I'm checking now so that I remain with the neighbors which I haven't visited yet.
4. I checked which of those neighbors is matching the condition of the difference between the value of the mean and the neighbors. After several cases, the best difference was 40 (in absolute value).
5. Each neighbor who fits to the above condition is filling with the value 1 in the seeds_data.
6. While the neighbors array is not empty and the number of iterations is less than 40 - I do it all over again (from level 3 and on)

- the number of iteration restricted to be 40 because it's taking too long above it and after 40 iterations the liver is segmented pretty well (but not perfectly) |

The liver segmentation of each case:

Case	Liver Segmentation

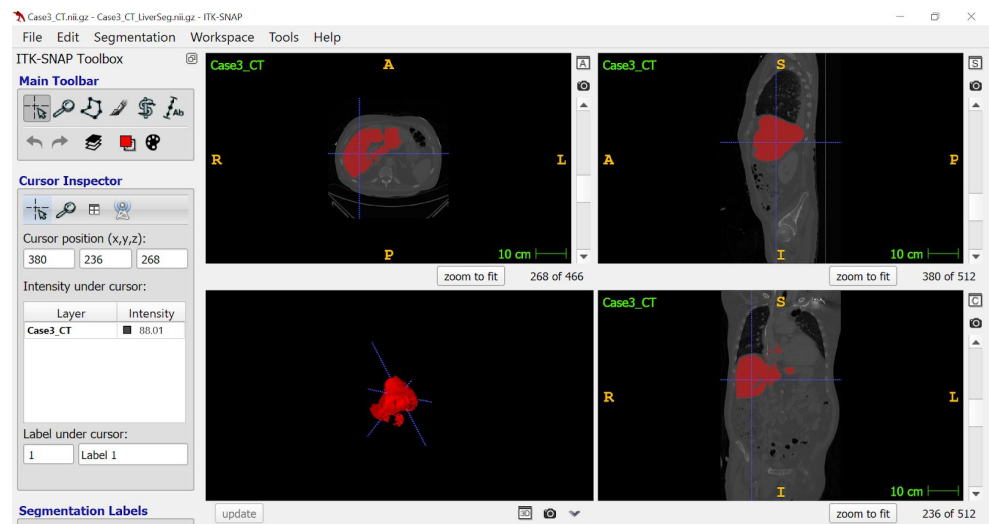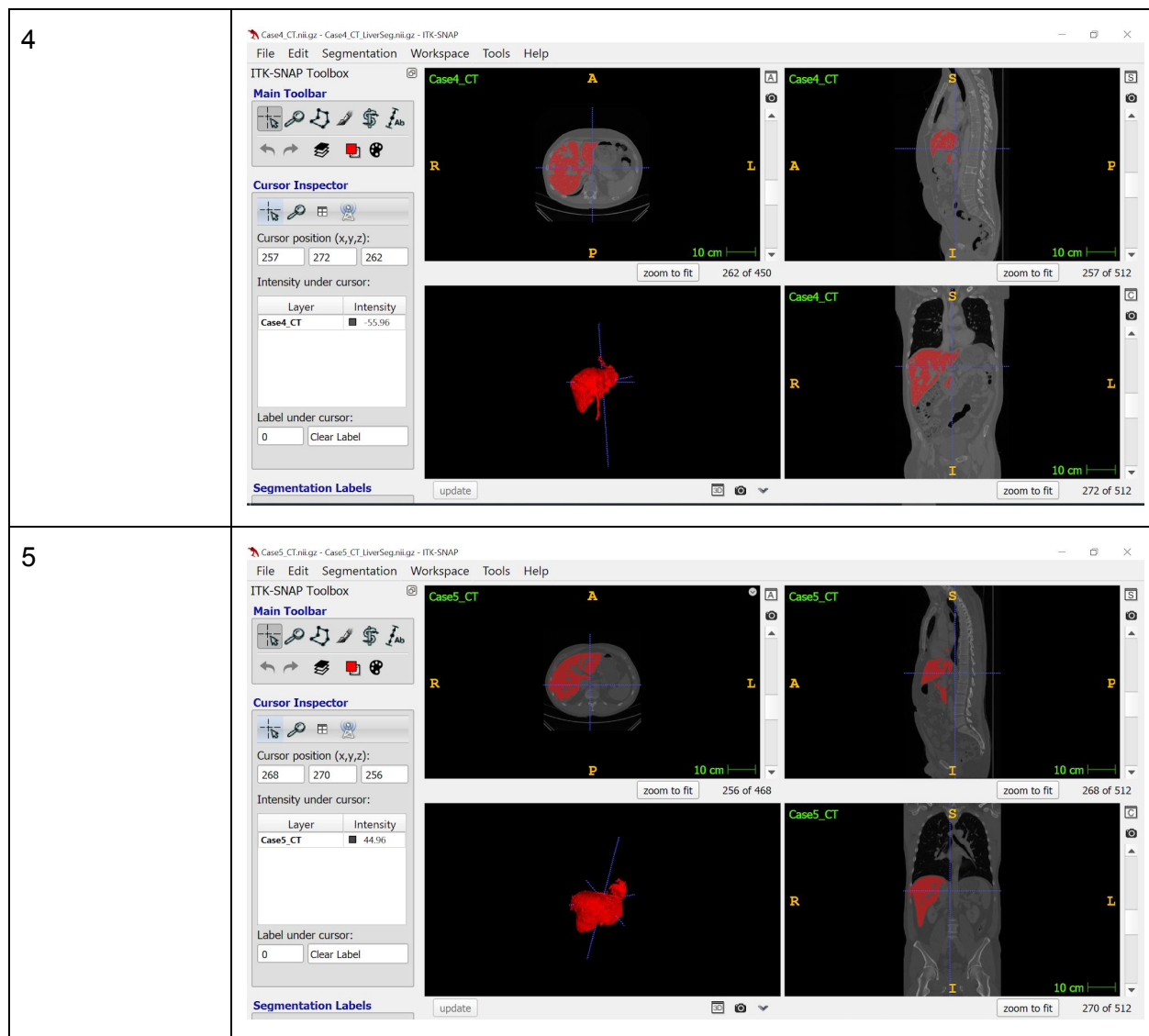| 1	
2	
3	

4	
5	
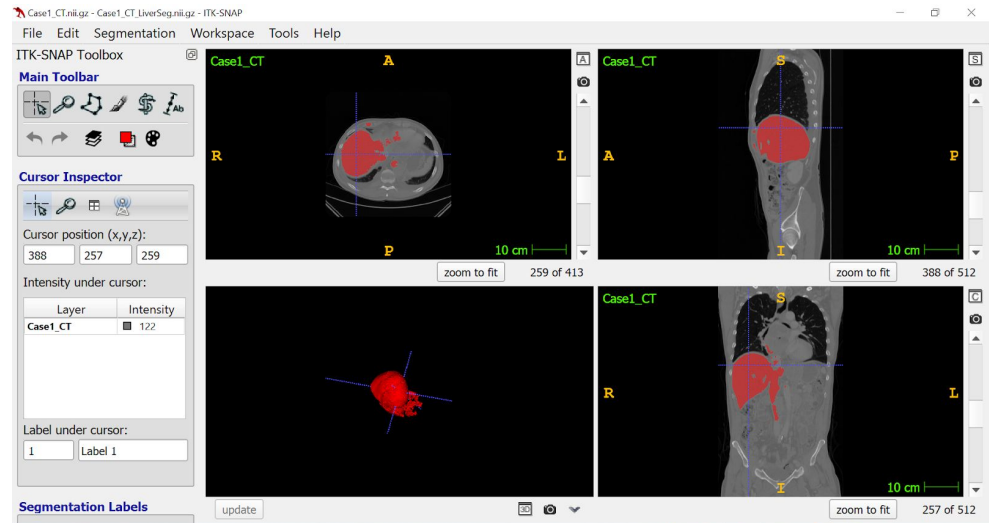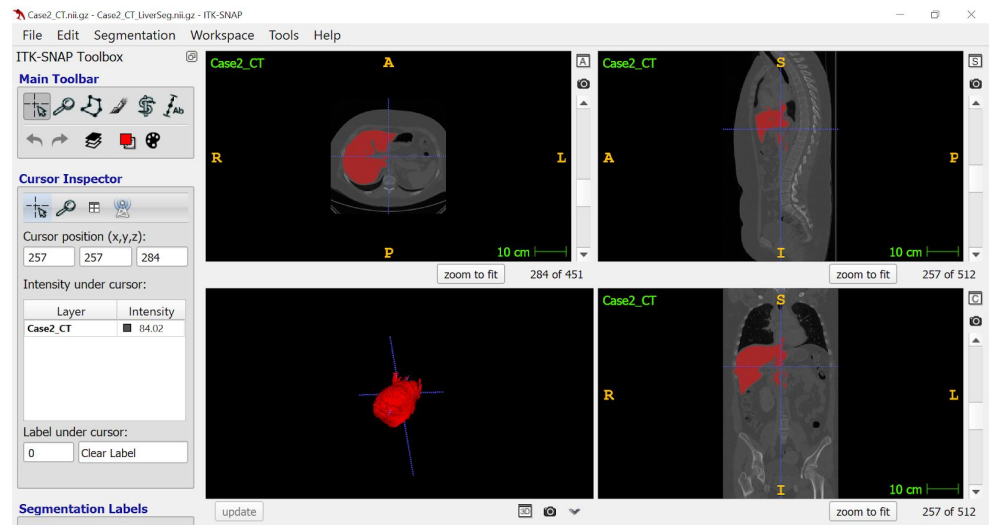
Part 3: Evaluation Metrics for the Segmentation

evaluateSegmentation
Input: Ground truth segmentation, Estimated segmentation
Output: Volume Overlap Difference and Dice Coefficient

The only two cases with GroundTruth were case1 and HardCase1 (the rest of the cases has no liver segmentation). Unfortunately, I had some technical problems and wasn't able to download HardCase1's files so I execute the function with case1 only.

The Volume Overlap Difference and Dice Coefficient for case1:

Volume Overlap Difference = 0.5568722684412866
Dice Coefficient = 0.8862554631174269

Part 4: MSRG Liver Segmentation

segmentLiver
Input: CT nifti file name, aorta segmentation nifti file name, output file name
Output: liver segmentation nifti file (to be saved under the given name)

The liver segmentation for each case is like in the multipleSeedsRG but here they are saved under the given name in the "output file name" parameter.