Medical Image Processing – Homework 2
Rotem Amsalem 207596297

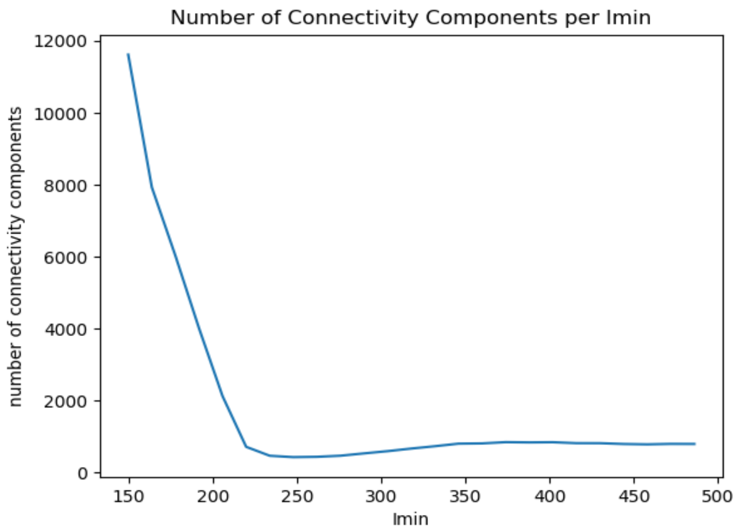
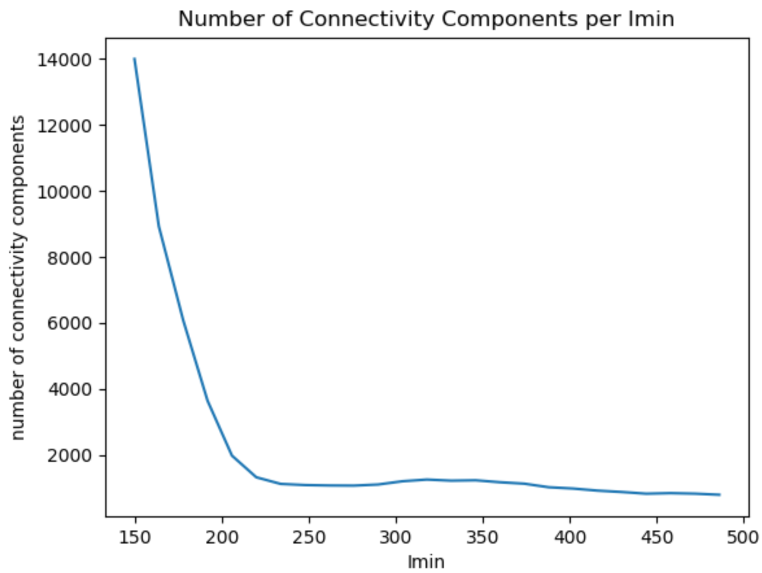
In this exercise I used the following libraries:

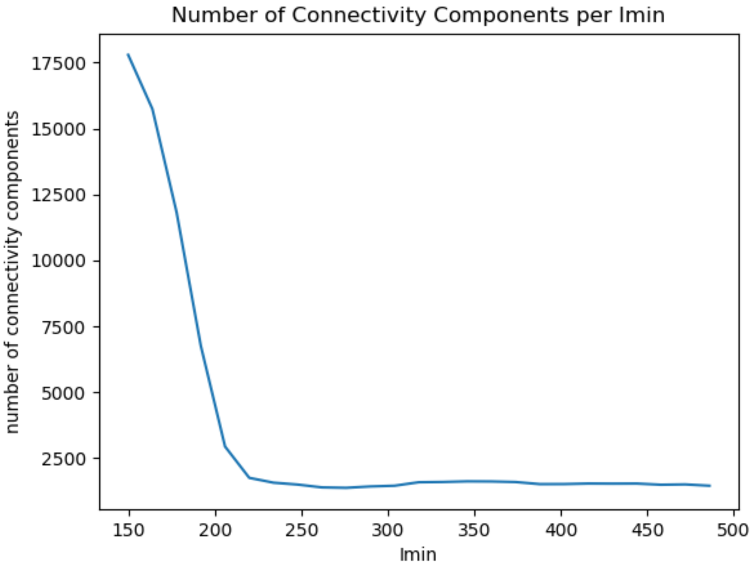
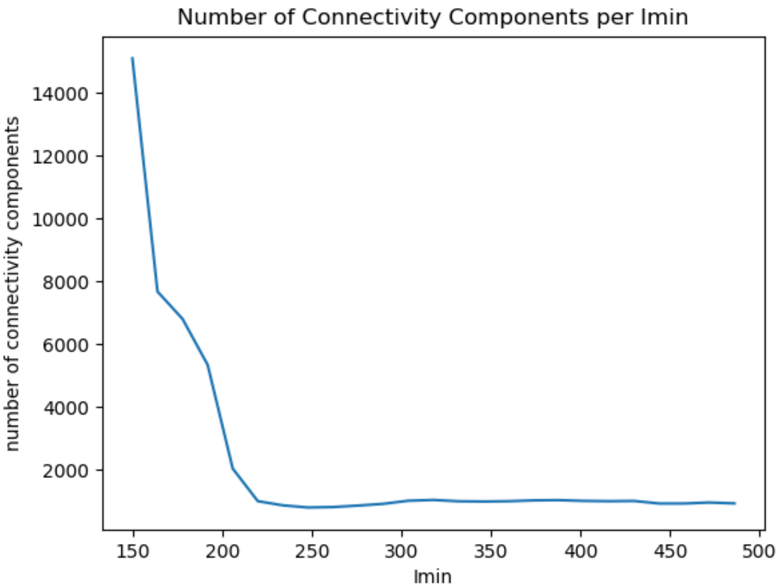
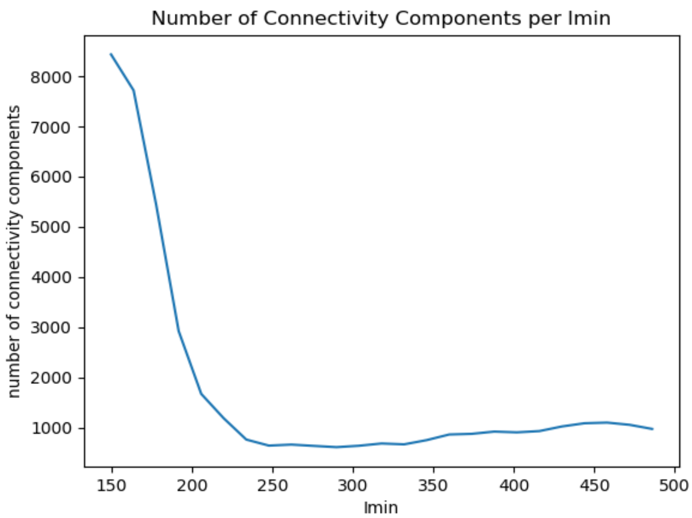
- nibabel – to save and load NIFTI images
- numpy – for the computing parts
- matplotlib.pyplot – to plot graphs and show them
- skimage – for algorithms for image processing
- scipy.signal – for finding local minima and maxima

Part 1:

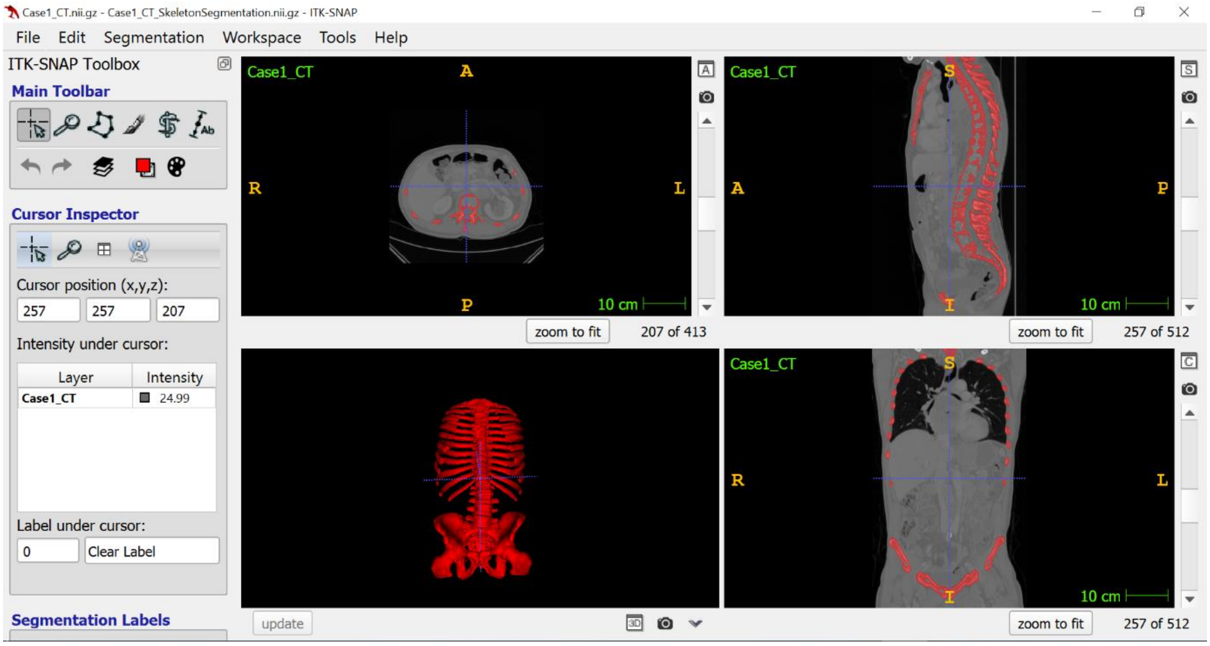
Function: SegmentationByTH – This function performs segmentation using Imin, Imax thresholds
Input: Nifty_file – a grayscale NIFTI file Imin – the minimal intensity Imax – the maximal intensity
Return: 1 in case of success 0 otherwise.

Function: downTo1ConnectivityComponent – This function performs post-processing (morphological operations – clean out single pixels, close holes, etc.) on the image until we are left with a single connectivity component. In this function I used some morphological operations such as remove_small_objects and removes all the objects smaller than min_size which is choosed by looping over the number of connectivity components till I got to 1 (if along the way I reached to 0 connectivity components then I did it all again but now without erosion) I used erosion too in order to erase thin limits. I also used dilation in order to close holes. This function saves the Skeleton Segmentation
Input: nifty_file_name – a grayscale NIFTI file segmentation – the nifty file after segmentation with the chosen Imin (by the SkeletonTHFinder function) aorta – 1 or 0 represents whether this is aorta segmentation or skeleton has_been_erosion – boolean. True means erosion has been made and we reached to 0 connectivity components and False mean otherwise.
Return: This function doesn't return a thing. It saves the Skeleton Segmentation. I present the final segmentation in the SkeletonTHFinder function.

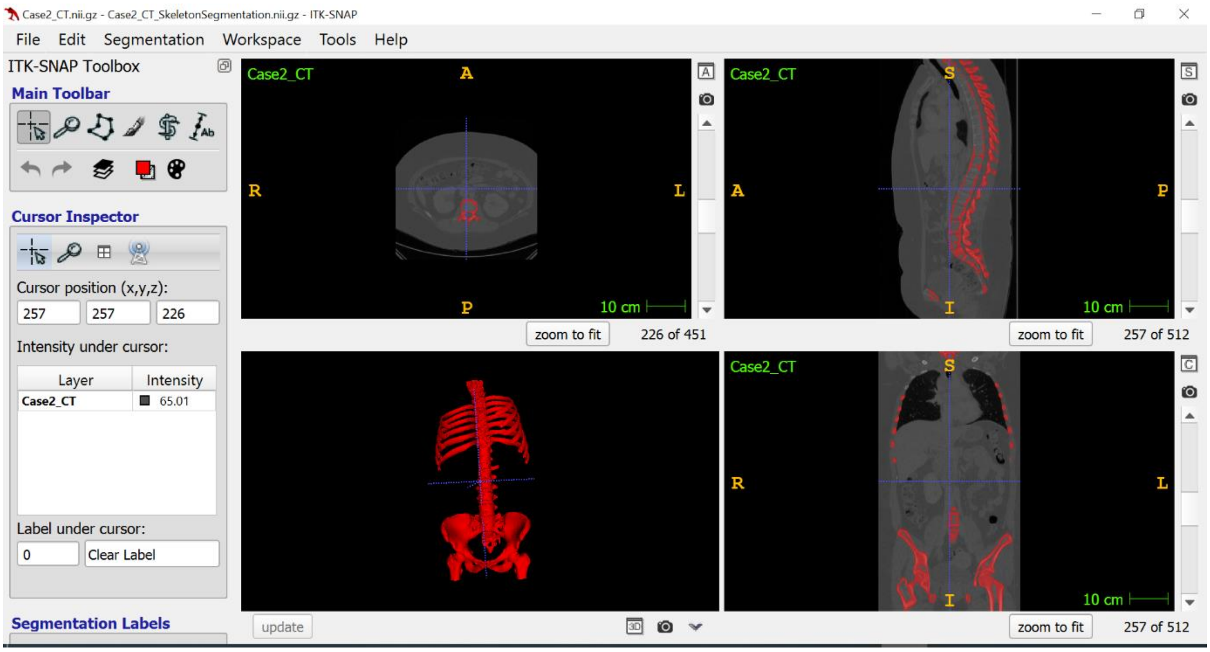
<table>
<tr><td colspan="3">Function:
SkeletonTHFinder – This function finds the best suited thresholds for each image after activating the segmentation.</td></tr>
<tr><td colspan="3">Input:
Nifty_file – a grayscale NIFTI file</td></tr>
<tr><td colspan="3">Return:
This function returns the Imin which is the threshold for the segmentation with the smallest number of connectivity components</td></tr>
<tr><td colspan="3">In this function I call the helper function downTo1ConnectivityComponent in order to be left with 1 connectivity component and this function saves the final segmentation which is now presented:
(The segmentations are presented below)</td></tr>
<tr><td>CT_case</td><td>Imin threshold</td><td>Plot</td></tr>
<tr><td>1</td><td>248</td><td>
</td></tr>
<tr><td>2</td><td>276</td><td>
</td></tr>
</table>

3	276	
4	248	
5	248	

Segmentation for Case1_CT:

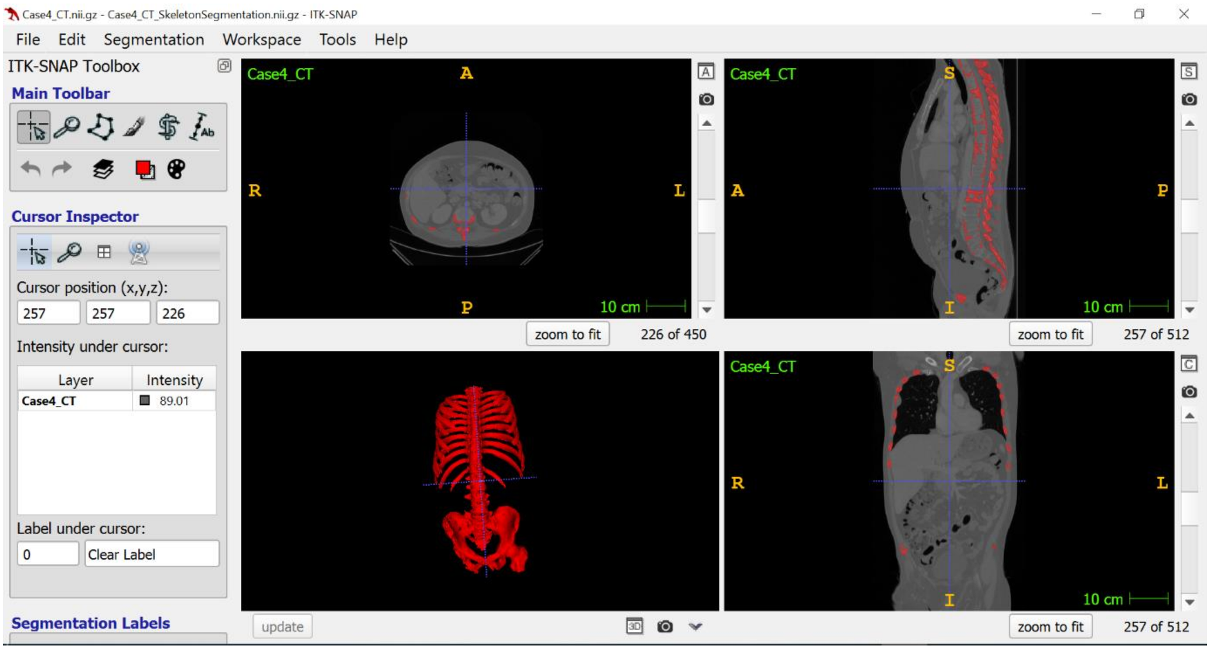


Segmentation for Case2_CT:

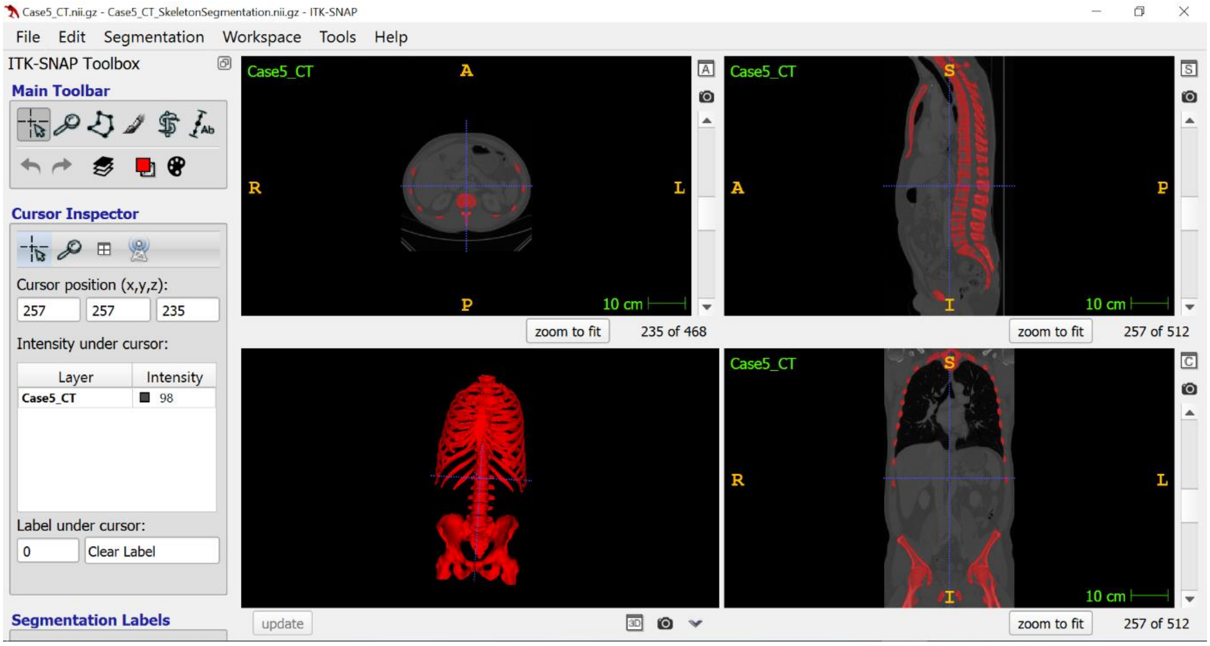


Segmentation for Case3_CT:

Segmentation for Case4_CT:



Segmentation for Case5_CT:

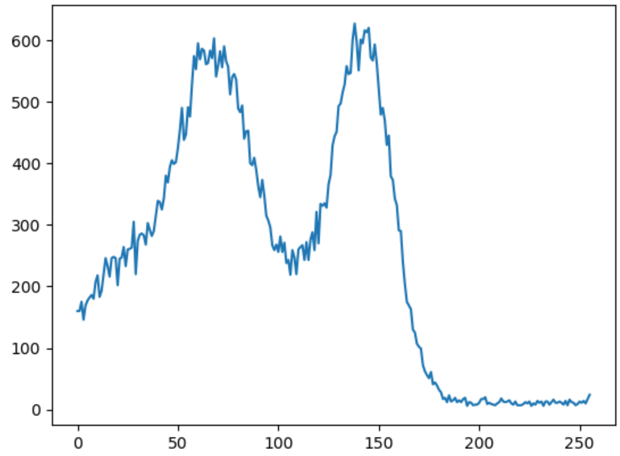


Part 2:

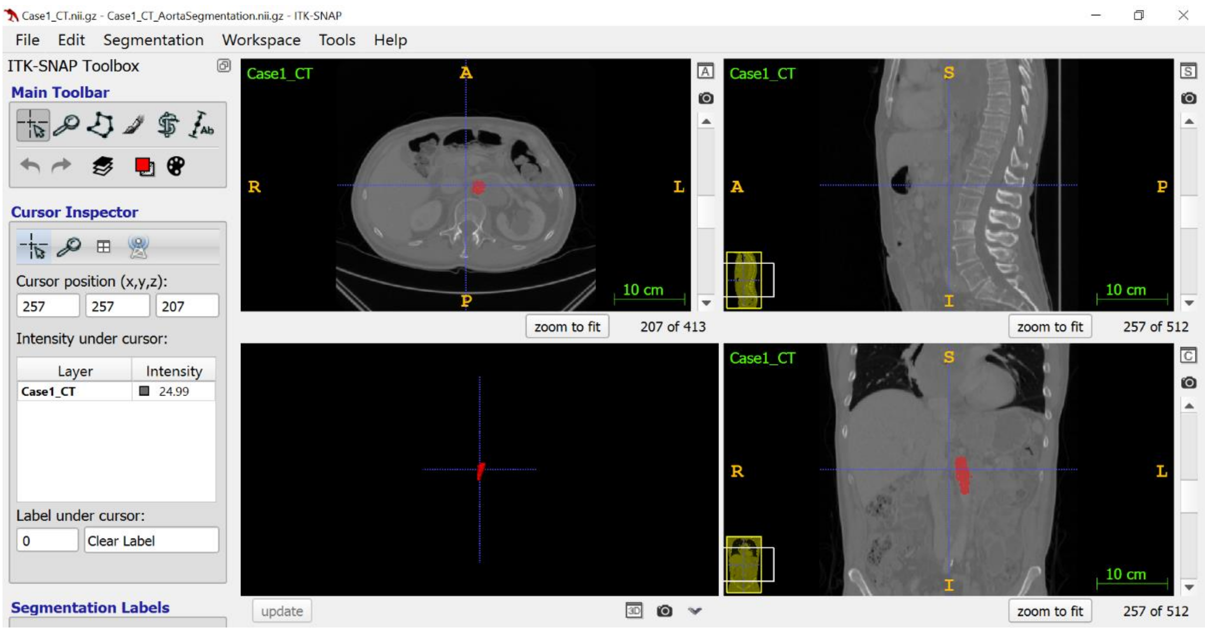
Function: AortaSegmentation – This function performs segmentation of the aorta by using L1 segmentation to tell on which axial slices we need to segment the aorta.
Input: nifty_file – a grayscale NIFTI file L1_seg_nifti_file - L1 segmentation NIFTI file
Return: Doesn't return a thing.
In this function I used the following algorithm in order to performs the aorta segmentation: 1. Finding ROI of L1 and calculating the coordinates of a box around L1 2. Building the ROI of the aorta using the above. 3. Finding the histogram of the above ROI 4. Finding the peak of the histogram (the one that matches the aorta because there is also a peak that matches to the background for example) 5. Thresholding to the ROI 6. Performing some morphological operations using the downTo1ConnectivityComponent function from part 1.

Outputs:

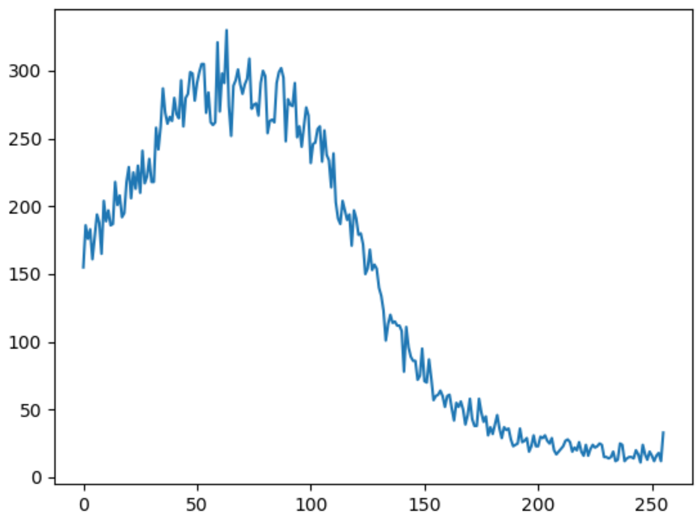
Histogram for Case1_CT:



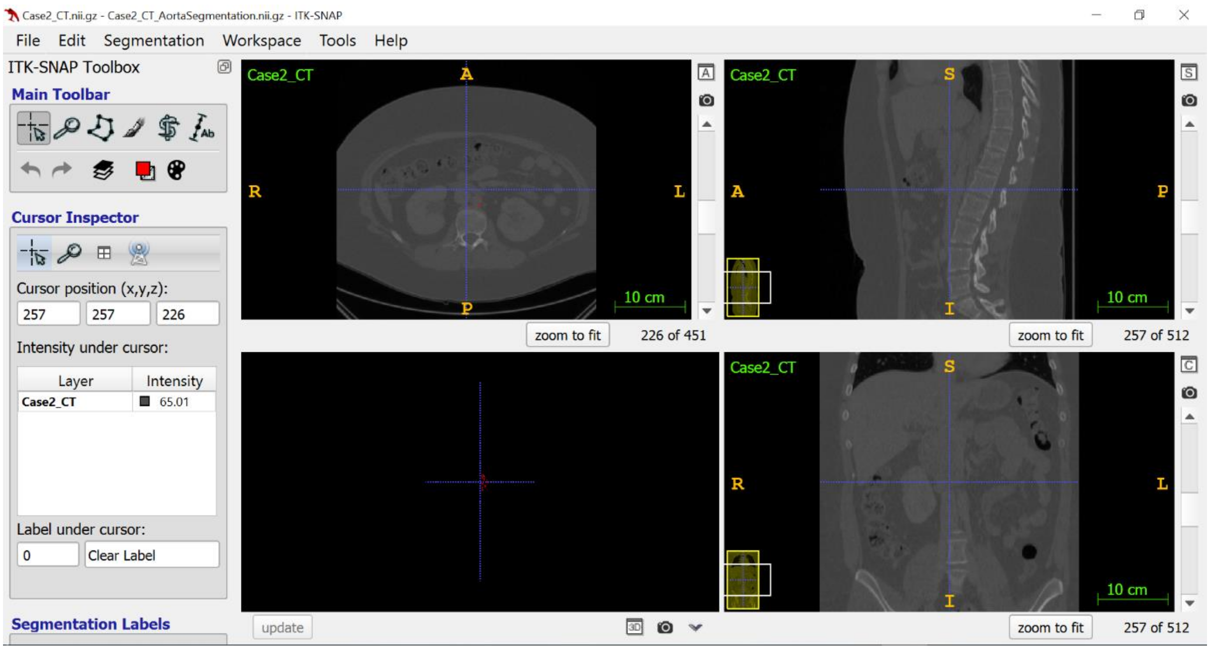
Segmentation for Case1_CT:



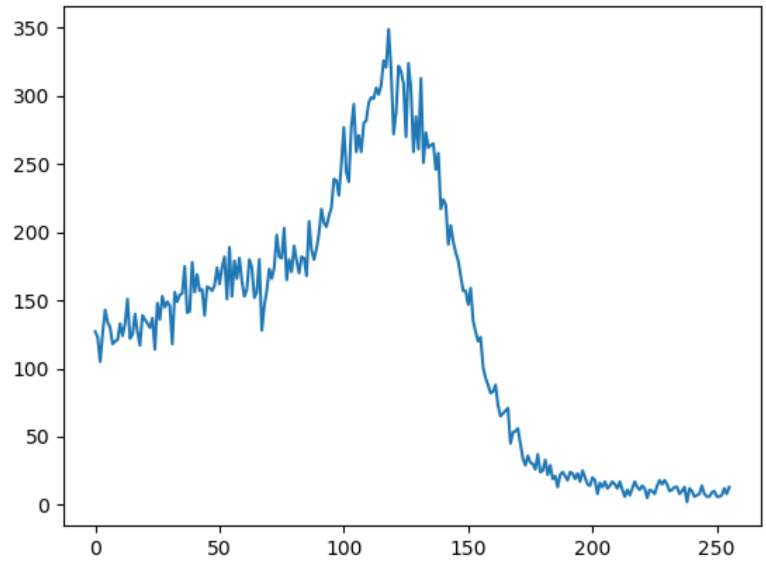
Histogram for Case2_CT:



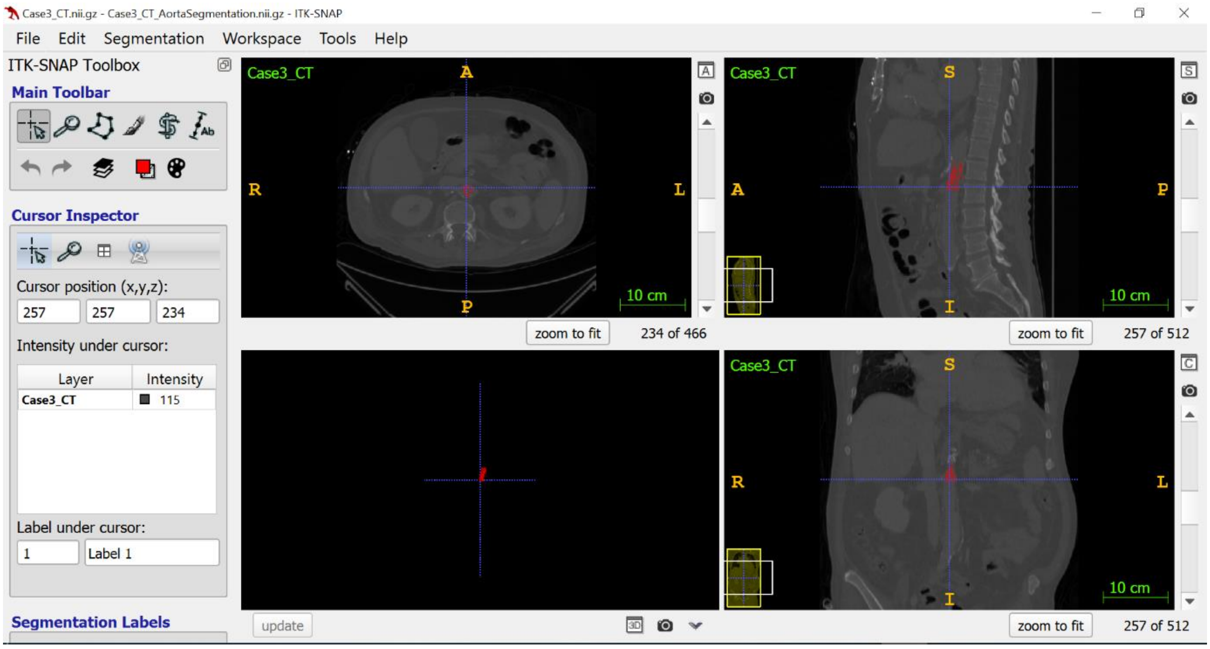
Segmentation for Case2_CT:



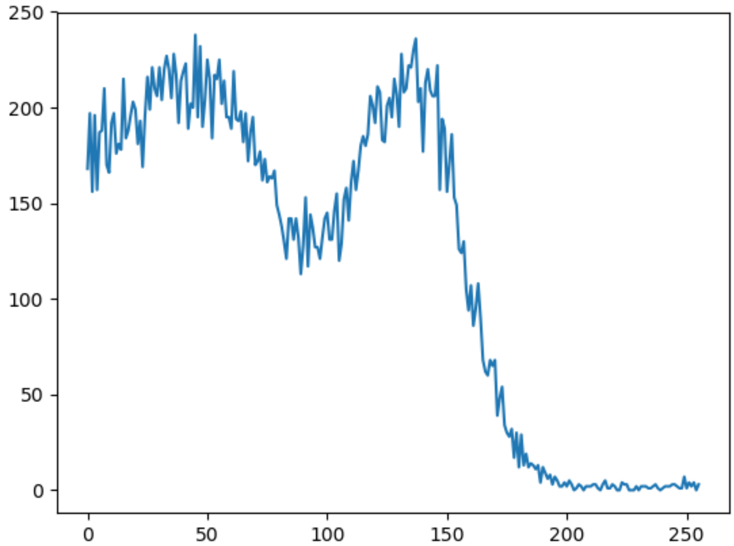
Histogram for Case3_CT:



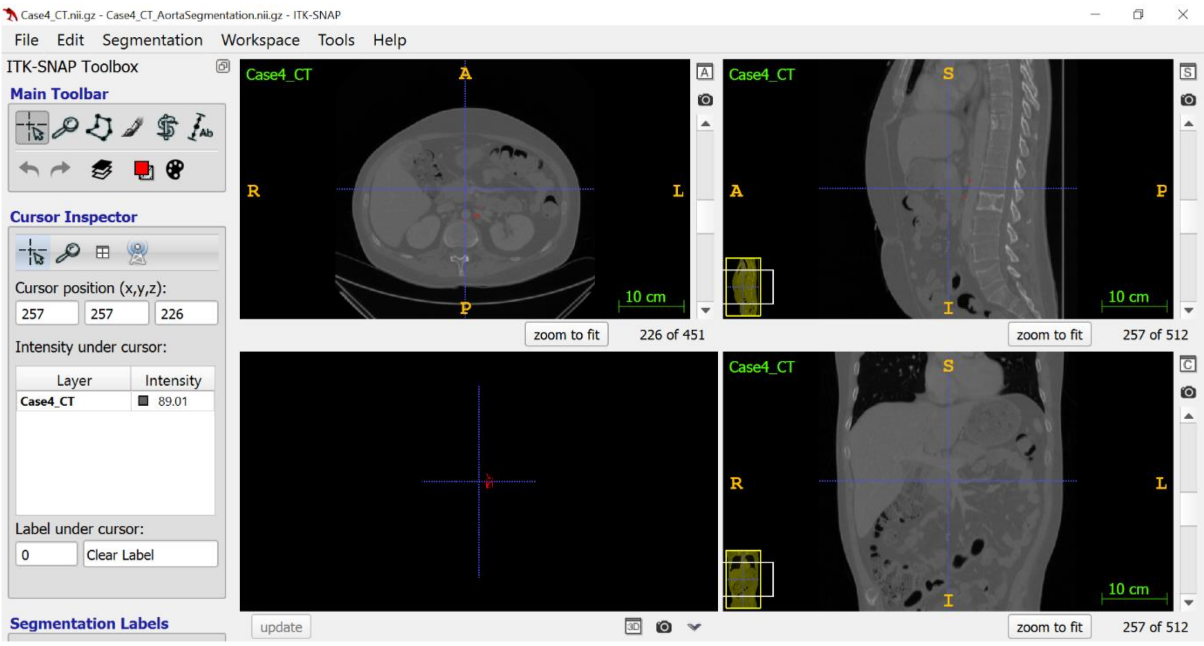
Segmentation for Case3_CT:



Histogram for Case4_CT:



Segmentation for Case4_CT:



Function: evaluateSegmentation - This function is given two segmentations, a GT one and an estimated one, and returns a tuple of (VOD_result, DICE_result).
Input: GT_seg – Ground Truth segmentation est_seg – estimated segmentation
Return:

tuple of (VOD_result, DICE_result).

Outputs:

CT_case	VOD_result	DICE_result
1	0.18648588575238945	1.627028228495221
2	0.9585424800148726	0.08291503997025469
3	0.5482048062104001	0.9035903875791997
4	0.999047783933518	0.001904432132963989