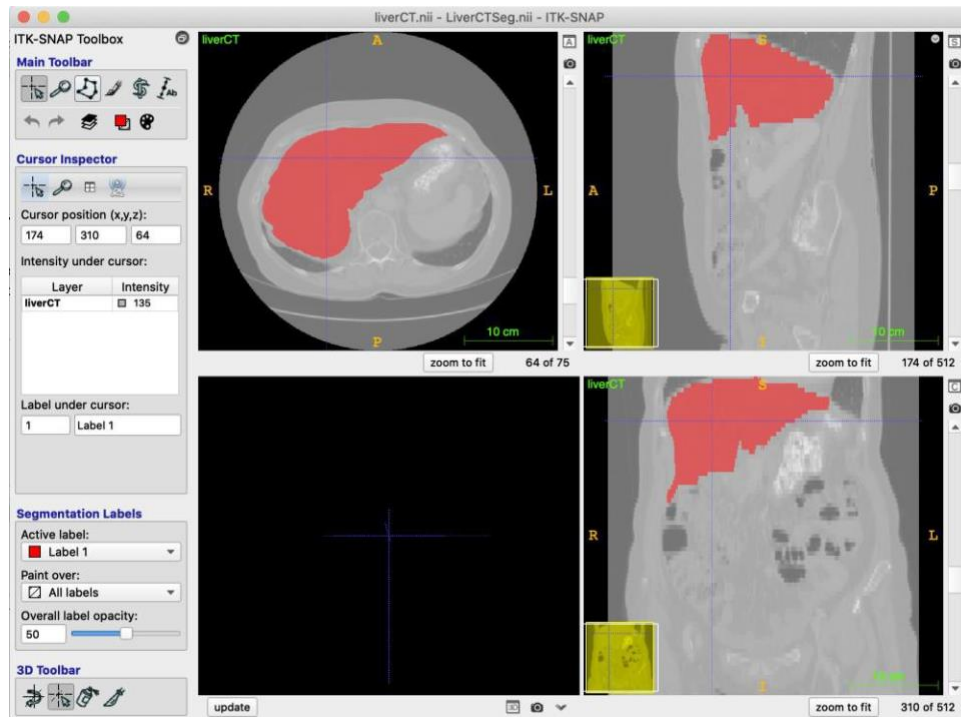


**REPORT**  
**CT Scan Liver Segmentation and MRI Inhomogeneity Correction**  
**Prepared by: Muhammad Ridzuan**

**Task 1: Windowing and detection from CT scans**

**Task 1.1. Read the provided liver CT scan and find the optimal intensity window for the liver structure by visualizing the result when choosing the best range. Output a screenshot from the axial, coronal and sagittal views and the best intensity window.**

I used ITK-SNAP to visualize the raw CT volume with its corresponding mask (Fig. 1). To find the optimal intensity window for the liver structure, I hovered the mouse over the masked area to estimate the Hounsfield range of the liver. I visualized different ranges using Python and found (experimentally) the optimal range to be between 105 and 140 (Fig. 2).



**Fig. 1.** Raw CT scan image with liver segmentation visualized using ITK-SNAP.

In order to improve upon the visualization, I experimented with two clipping methods. In the first method, I forced any value below 105 to be equal to 105, while any value above 140 to be equal to 140 (Fig. 2). In the second method, I binarized the image such that any value outside the range [105, 140] is equal to 0, while any value within the range is equal to 1 (Fig. 3). I found the second method to be more effective at making the liver's overall structure more pronounced while dampening the structures of other organs (note that the internal structures of the liver were also dissolved for the most part).



**Fig. 2.** Axial, coronal, and sagittal views using the clipping method (Hounsfield range [105, 140]).



**Fig. 3.** Axial, coronal, and sagittal views using the binarized method (Hounsfield range [105, 140]).

**Task 1.2. Following Task 1.1, regardless of the intensity window used, intensities on other structures will still be visible.**

- **Research and identify effective non-ML methods to segment the liver within the 3D volume.**
- **Write a Python code to remove all non-liver regions keeping the liver only (in 3D).**
- **Ensure that there are no holes within the liver structure.**
- **Based on one of the non-ML methods you have identified, generate a 3D segmentation mask of the liver.**
- **Given a segmentation mask called “LiverCTSeg.nii”, compute the Dice coefficient between the generated mask and the provided mask on the liver region *only* (exclude background voxels).**

There are many ways of segmenting a liver [1]. [2] uses a region-growing based approach, where starting with a seed point, the segmented region grows outwardly based on the similarity of intensity of neighboring pixels until an accurate segmentation is obtained. The disadvantage of this method is that it is highly dependent upon the initialization location of the seed point.

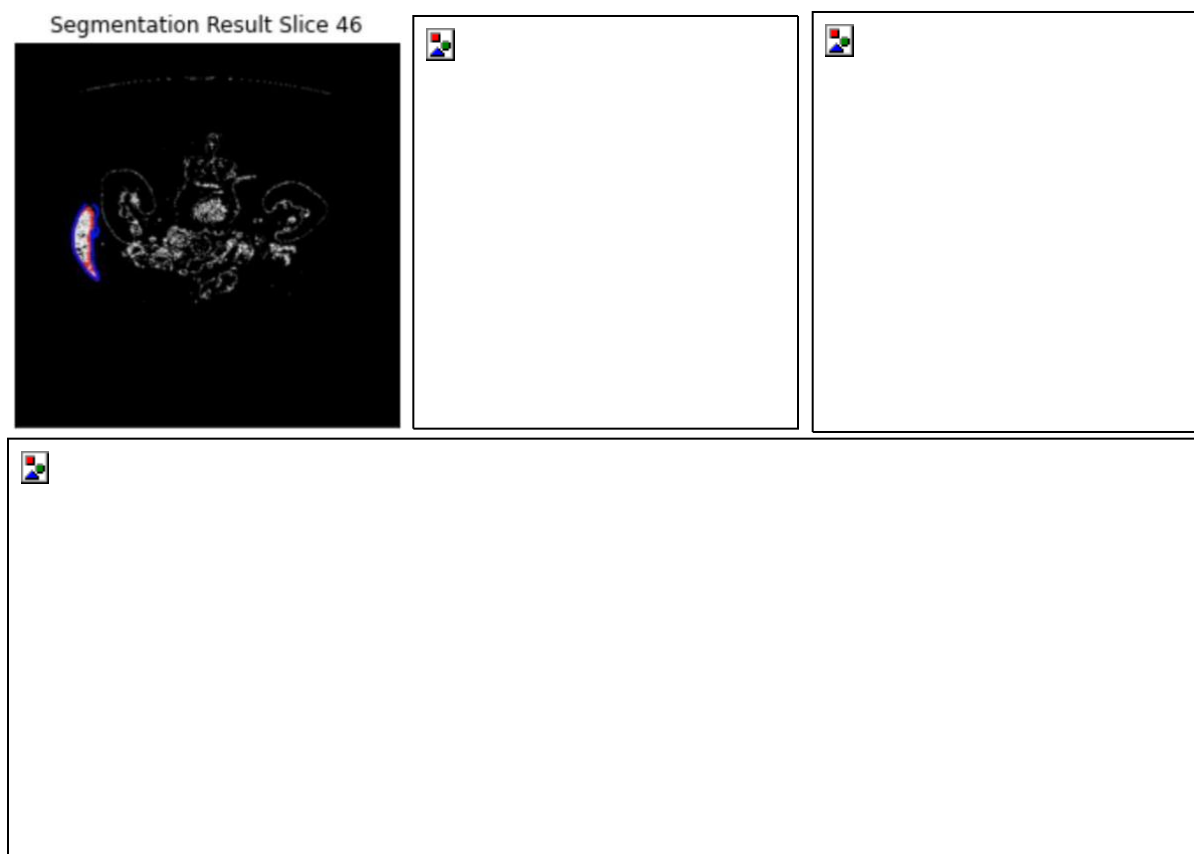
[3] and [4] uses a threshold-based approach, where a global threshold intensity is used to initialize the liver segmentation, then subsequent filters are applied to further isolate the region of interest. [5] uses a histogram-based threshold to eliminate other organ structures, followed by a binary morphological filter. The disadvantage of a threshold-based method is that a fixed manual threshold is unlikely to generalize well to other images due to differences in CT scanners and patient slices [1].

The method used in this assignment is based on [4] and is outlined below with additional steps shown by (\*):

For each 2D slice:

1. Apply intensity thresholding and binarize (from Task 1.1)
2. Apply median filter to remove noise\*
3. Find contours of the same values using the marching squares algorithm [6]
4. Isolate liver contours by applying thresholds on the contour area and perimeter\* to eliminate non-liver contours
5. If a list of contours is found, only the largest is returned as the liver contour. This helps ensure there are no holes within the liver structure.
6. Create a mask polygon from the contour
7. Apply dilation, erosion, and median filter to smoothen the mask edges\*
8. Append each 2D mask to create a 3D mask

Fig. 4. shows the resulting mask from a few 2D slices. The overall Dice score with the ground truth mask is 0.9337.



**Fig. 4.** Result of liver segmentation (red) vs. ground truth mask (blue).

## **Task 2: Reduction of MRI image inhomogeneity and noise**

**Given 2 noisy and inhomogeneous MRI images (Inh1 and Inh2) and the corresponding clean MRI images (T1 and T2), apply inhomogeneity correction and filtering to clean the inhomogeneous inh1 and inh2 MRI images to get as close as possible to the clean MRI. Measure the correlation coefficient (CC) on each slice and report the average and standard deviation of CC between the noisy and inhomogeneous images, the clean images, and the outputs of the different correction and filtering methods you selected.**

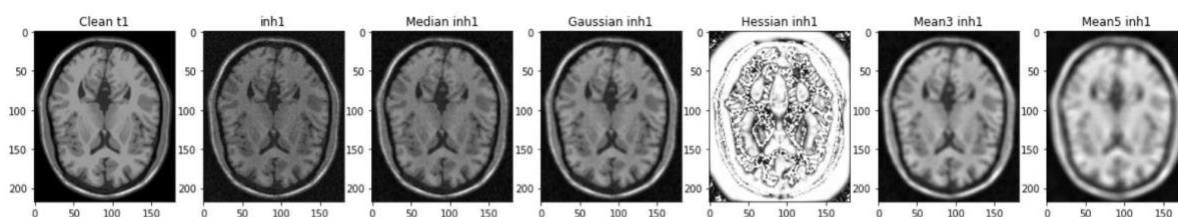
I applied inhomogeneity correction using 5 classical filters: median, mean (3x3), mean (5x5), Gaussian, and Hessian. Interestingly, the inhomogeneity correction resulted in a lower average correlation coefficient (r-score), however the score improved when the filters were applied on top of it. The best average correlation coefficient for T1 was found using the Gaussian filter, followed by the mean (5x5) filter (r-score of 0.983 and 0.980 respectively, compared to 0.965 for the clean and inhomogeneous image). The best average correlation coefficient for T2 was found using the mean (5x5) filter, followed by the Gaussian filter (0.980 and 0.953 respectively, compared to 0.940 for the clean and inhomogeneous image) (Table 1; Fig. 5).

These two filters (Gaussian and mean) are effective at removing noise because they calculate the average of each pixel's neighborhood, with the Gaussian filter taking a weighted average towards the central pixel value, while the mean filter taking a uniform average of the neighborhood pixels. Interestingly, the median filter which is generally good for edge preservation, does not perform as well in this task. This suggests that there are not as many edge structures in these 3D MRI images as there are whole areas/volumes.

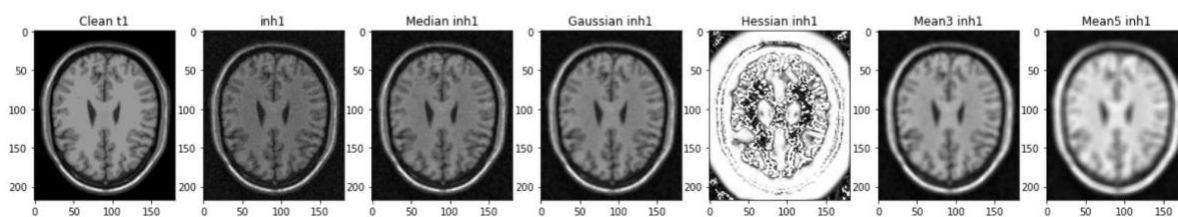
**Table 1.** Correlation coefficient scores, means, and standard deviations for different filters. The full list of coefficient scores can be found on the attached ipynb.

		r_score	mean_r_score	std_r_score
image1	inh	[0.962, 0.961, 0.96, 0.96, 0.959, 0.959, 0.958, 0.958, 0.957, 0.956, 0.957, 0.958, 0.957, 0.958, 0.958, 0.958, 0.959, 0.958, 0.959, 0.96, 0.961, 0.961, 0.962, 0.963, ...	0.965	0.023
	corrected_inh	[0.951, 0.95, 0.949, 0.949, 0.947, 0.946, 0.945, 0.945, 0.944, 0.944, 0.944, 0.945, 0.945, 0.947, 0.947, 0.948, 0.949, 0.949, 0.949, 0.951, 0.953, 0.954, 0.955, 0.956...	0.947	0.032
	med	[0.979, 0.979, 0.978, 0.979, 0.978, 0.978, 0.978, 0.978, 0.977, 0.978, 0.978, 0.979, 0.979, 0.98, 0.979, 0.98, 0.98, 0.981, 0.981, 0.982, 0.982, 0.983, 0.983, 0.984, ...	0.979	0.010
	gauss	[0.983, 0.982, 0.982, 0.982, 0.981, 0.982, 0.982, 0.982, 0.981, 0.982, 0.982, 0.982, 0.983, 0.983, 0.983, 0.984, 0.984, 0.984, 0.985, 0.985, 0.986, 0.986, 0.987, 0.98...	0.983	0.007
	hessian	[0.028, 0.034, 0.031, 0.044, 0.034, 0.034, 0.033, 0.034, 0.027, 0.029, 0.039, 0.021, 0.024, 0.013, 0.012, 0.016, 0.014, 0.015, 0.017, 0.014, 0.007, -0.0, -0.014, 0.00...	0.021	0.187
	mean3	[0.975, 0.975, 0.974, 0.974, 0.973, 0.972, 0.973, 0.972, 0.971, 0.971, 0.971, 0.972, 0.973, 0.973, 0.973, 0.974, 0.974, 0.974, 0.974, 0.975, 0.976, 0.977, 0.977, 0.97...	0.978	0.006
	mean5	[0.98, 0.9...	0.980	0.000
image2	inh	[0.894, 0.894, 0.898, 0.901, 0.906, 0.911, 0.914, 0.917, 0.922, 0.924, 0.925, 0.927, 0.928, 0.927, 0.927, 0.926, 0.927, 0.928, 0.931, 0.932, 0.932, 0.935, 0.935, 0.93...	0.940	0.018
	corrected_inh	[0.865, 0.865, 0.868, 0.871, 0.876, 0.883, 0.885, 0.888, 0.895, 0.897, 0.896, 0.898, 0.9, 0.899, 0.899, 0.898, 0.899, 0.899, 0.904, 0.905, 0.905, 0.908, 0.909, 0.911, ...	0.894	0.020
	med	[0.938, 0.938, 0.938, 0.939, 0.943, 0.945, 0.945, 0.948, 0.951, 0.952, 0.951, 0.953, 0.954, 0.953, 0.952, 0.951, 0.951, 0.951, 0.954, 0.954, 0.955, 0.956, 0.957, 0.95...	0.948	0.011
	gauss	[0.947, 0.948, 0.948, 0.95, 0.952, 0.954, 0.954, 0.956, 0.959, 0.959, 0.959, 0.96, 0.961, 0.96, 0.96, 0.959, 0.958, 0.958, 0.96, 0.961, 0.96, 0.962, 0.962, 0.963, 0.9...	0.953	0.011
	hessian	[-0.078, -0.089, -0.091, -0.084, -0.115, -0.112, -0.102, -0.124, -0.106, -0.115, -0.103, -0.136, -0.128, -0.126, -0.104, -0.117, -0.12, -0.098, -0.11, -0.106, -0.095, ...	-0.033	0.103
	mean3	[0.899, 0.9, 0.899, 0.9, 0.904, 0.907, 0.906, 0.909, 0.912, 0.914, 0.914, 0.915, 0.916, 0.915, 0.913, 0.912, 0.91, 0.909, 0.912, 0.913, 0.912, 0.916, 0.918, 0.919, 0...	0.914	0.028
	mean5	[0.98, 0.9...	0.980	0.000

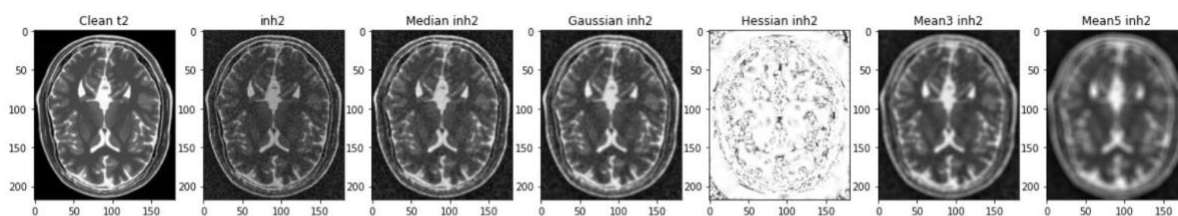
Sample Slice 75



Sample Slice 100



Sample Slice 75



Sample Slice 100

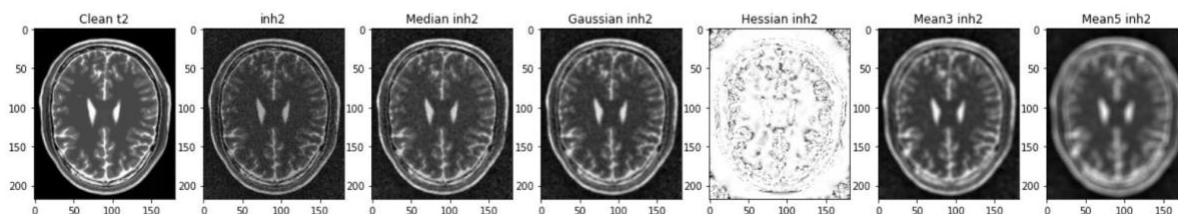


Fig. 5. Bias correction and filtering results from select slices.

## REFERENCES

- [1] S. Priyadarsini and D. Selvathi, "Survey on segmentation of liver from CT images," 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2012, pp. 234-238, doi: 10.1109/ICACCCT.2012.6320777.
- [2] S.S. Kumar ,R.S. Moni, J. Rajeeesh, " Automatic Segmentation of Liver and tumor for CAD of Liver", Journal of Advances in Information Technology, Vol. 2, No. 1, February 2011.
- [3] M. Usman Akram, A. Khanum and K. Iqbal, "An automated System for Liver CT Enhancement and Segmentation", ICGSTGVIP Journal, Volume 10, Issue IV, October 2010.
- [4] N. Adaloglou, "Introduction to medical image processing with Python: CT lung and vessel segmentation without labels", The AI Summer, February 2021. <https://theaisummer.com/medical-image-python/>
- [5] O. Zayane, B. Jouini, M. Ali Mahjoub, "Automatic liver segmentation method in CT images", Canadian Journal on Image Processing & Computer Vision Vol. 2, No. 8, December 2011.
- [6] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Guillard, T. Yu and the scikit-image contributors, "scikit-image: Image processing in Python", PeerJ 2:e453, 2014. [https://scikit-image.org/docs/0.8.0/api/skimimage.measure.find\\_contours.html](https://scikit-image.org/docs/0.8.0/api/skimimage.measure.find_contours.html)