Algorithm Bake-Off: Using Flywheel Data Exchange and Jupyter Notebook Integration to Compare AI Algorithms for CT Segmentation

Overview

"The Flywheel Core platform consists of a number of stellar product features. These include, a Data Exchange, Jupyter Notebook integration and the flexibility to package an AI algorithm as a Gear. In this whitepaper, we show how all these product features can becombined to compare two or more AI algorithms for CT Segmentation."

Background

Fully automated organ segmentation on Computed Tomography (CT) images is an important first step in many medical applications. Many different Deep Learning (DL) based approaches are being actively developed for fully automated organ segmentation. [1, 2] However, often it is hard to make a direct comparison between two segmentation methods since their performance is not reported on the same independent open-source dataset.

Specific Aims

In this paper, we compare the performance of two open-source deep learning based CT segmentation algorithms[1, 2] on an independent open-source CT dataset [3]. We highlight the importance of curating acquisition parameters for analyzing specific segmentation outliers.

Methods:

Two open-source DL based CT segmentation algorithms were selected for this work. These are:

> **1. Total-Segmentator [1]:** In this algorithm, the authors used a training set of 1204 semi-manually segmented CT as a training dataset. This will be referred to as algorithm-1 in this work.
>
> **2. Swin-UNETR [2]:** In this algorithm, the authors used Self-Supervised Learning (SSL) to pretrain a network on 5050 CT scans [4]. The pre-trained network was then fine-tuned on a set of 24 manually segmented CT scans [5]. This will be referred to as algorithm-2 in this work.

Both algorithms were packaged as docker images to ensure reproducibility of the analysis.

For this analysis, both algorithms were tested on an open-source CT dataset consisting of 140 CT volumes [3]. These datasets are available in NIfTI format and the original DICOM scans were not available. The dataset includes manual annotations of 6 organs on each of the 140 CT volumes. One image was removed from the analysis since it did not have liver segmentation. This dataset was not included in the training or validation sets of either of the segmentation algorithms. For this analysis, we compared the performance of the algorithms on the segmentation of liver (large organ) and kidney (small organ). These organs were selected as they are common between the algorithms and the test dataset. The Dice metric was selected to report the performance of the two algorithms. We ran both algorithms on all images and reported the statistics and number of cases with Dice <= 0.5.

Flywheel Implementation Details

A key aspect of this project is that all of the data processing and analysis is done in the Flywheel Core platform. We used a specific Flywheel instance for all steps. The project was labeled as "CT-ORG". Note that each project in Flywheel Core can have one or more workspaces. All segmentation results, Jupyter Notebooks and analysis plots are saved in a single workspace. We list a high-level overview of the steps on the right:

1. The data set [3] is available in Flywheel Exchange (figure 1). It took a few clicks to copy the data into the Flywheel instance. It consists of 140 NIfTI files and the related manual segmentations.

2. Both AI-based CT segmentation algorithms were packaged as Gears and released on Flywheel Exchange. Figure 2 shows the Gear card for the algorithm-1 on Flywheel Gear Exchange.

3. Both Gears were run on all images with a valid ground truth.

- **a.** This was done with a Jupyter Notebook. Note that the notebook is stored with the project.

4. Each Gear generated a segmentation output. Each output was uploaded to the appropriate acquisition container. This can be seen in Figure 3. Note that Gear generated files have a "gear" icon.

5. All segmentation results for both Gears were downloaded to the workspace.

6. Additional Jupyter Notebooks were created to compare the output from each Gear to the human ground truth. As mentioned above, the Dice metric was used for this comparison. These results were saved in csv files in the Flywheel "workspace" for the current project.

7. Statistical analysis was done in another Jupyter Notebook.

Results

The median Dice for liver segmentation for algorithm-1 is 0.954 (IQR = 0.027) and for algorithm-2 is 0.936 (IQR = 0.094).

The median Dice for kidney segmentation for algorithm-1 is 0.910 (IQR = 0.044) and for algorithm-2 is 0.838 (IQR = 0.316).

Figure 1 and 2 show the distribution of the Dice values for algorithm-1 and algorithm-2 respectively.

Low Dice values for algorithm-1 and algorithm-2:

1. For algorithm-1, the Dice was less than 0.5 for 21 liver segmentations.
2. For algorithm-1, the Dice was less than 0.5 for 9 kidney segmentations.
3. For algorithm-2, the Dice was less than 0.5 for 11 liver segmentations.
4. For algorithm-2, the Dice was less than 0.5 for 31 kidney segmentations.

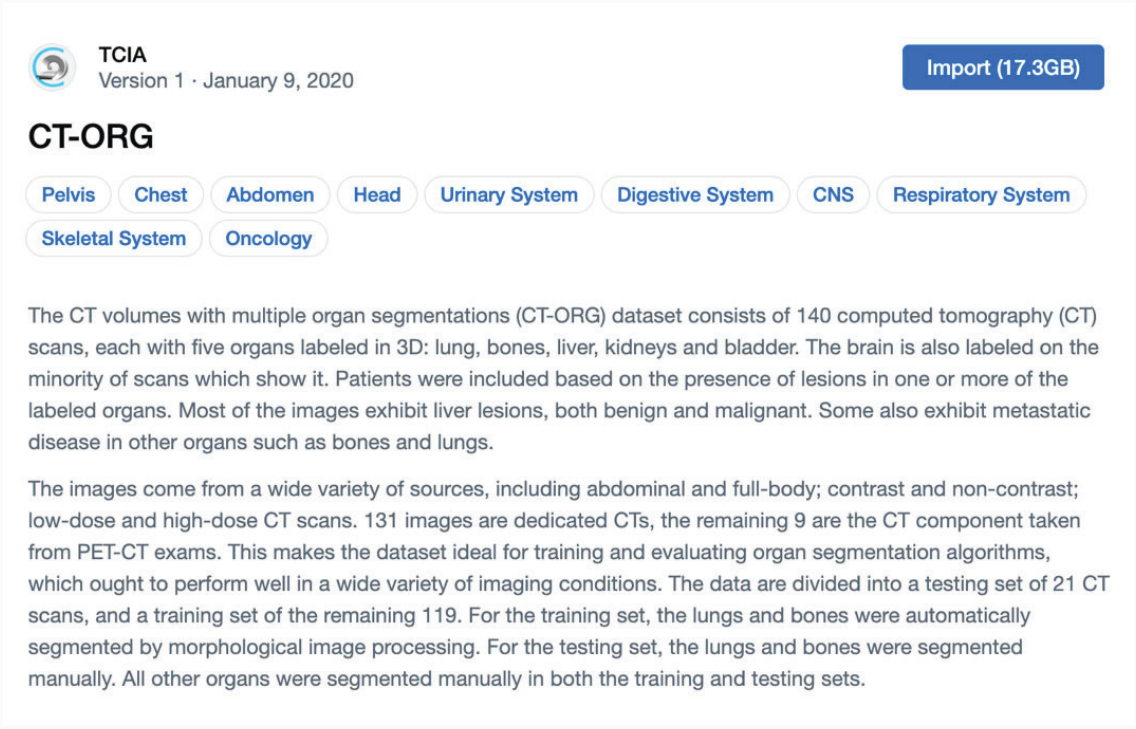


Figure 1: This figure shows the data card for the CT-ORG[3] dataset in Flywheel Data Exchange.

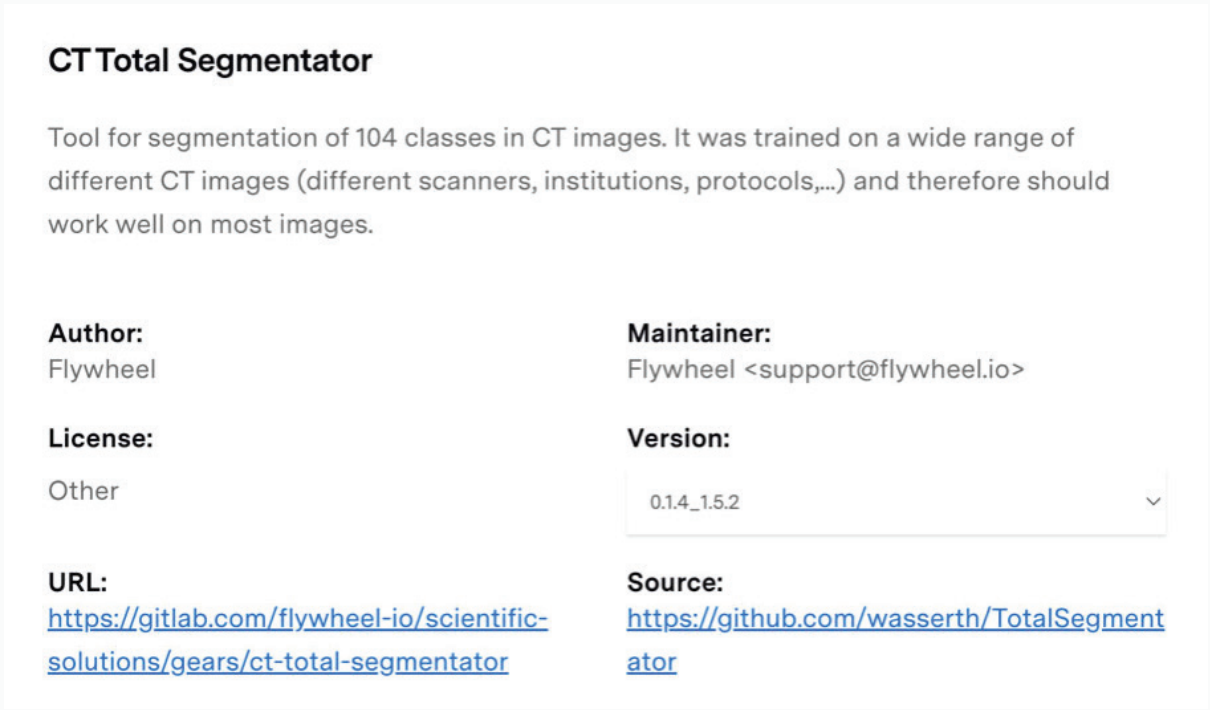


Figure 2: View of the CT-Total Segmentator gear from Flywheel Gear Exchange

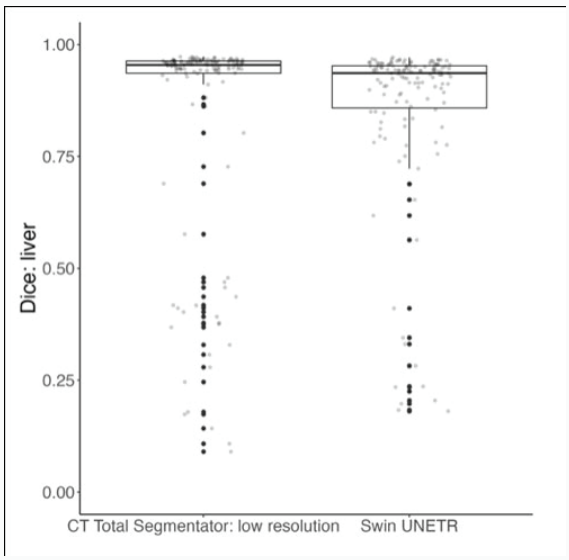


Figure 3: Dice values for Liver Segmentation

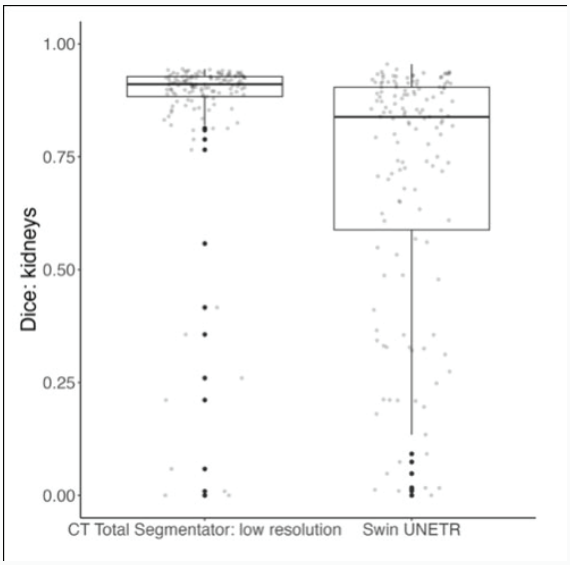
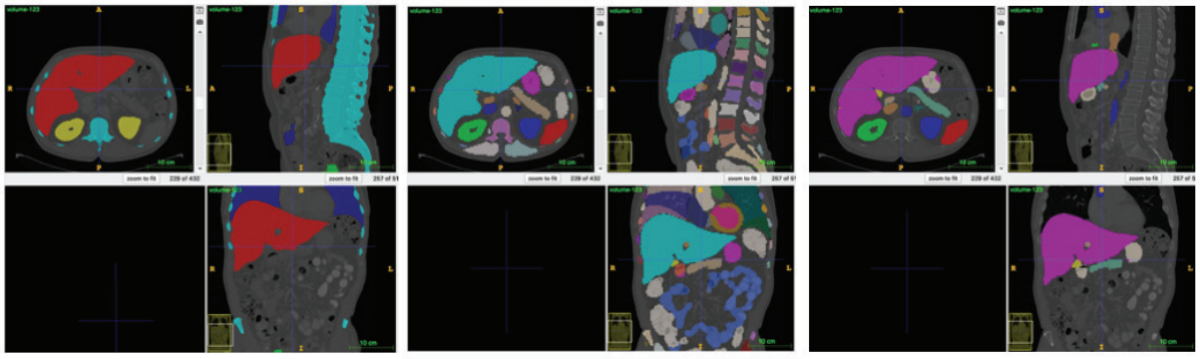


Figure 4: Dice values for Kidney Segmentation



Volume-123 with ground truth

Algorithm-1 Dice: Liver=0.97, Kidney=0.94

Algorithm-2 Dice: Liver=0.97, Kidney=0.93

Figure 5: Both algorithms had the best average Dice on the same volume. Liver in cyan (Algorithm-1) & magenta (Algorithm-2)

In Figure 5, we show good liver and kidney segmentations for both algorithms.

In Figure 6, we study the relationship between liver and kidney segmentation Dice values and x and z resolutions for both algorithms. The goal is to understand the impact of scan resolution. Note that no clear patterns are observed between the Dice values and the image acquisition parameters.

In Figures 5, 6 and 7, we show multiple volumes where both algorithms performed poorly. The goal was to find specific patterns among the outliers. While some trends were observed (both algorithms failed to segment kidneys on whole body CT scans), it was not possible to pinpoint the root cause.

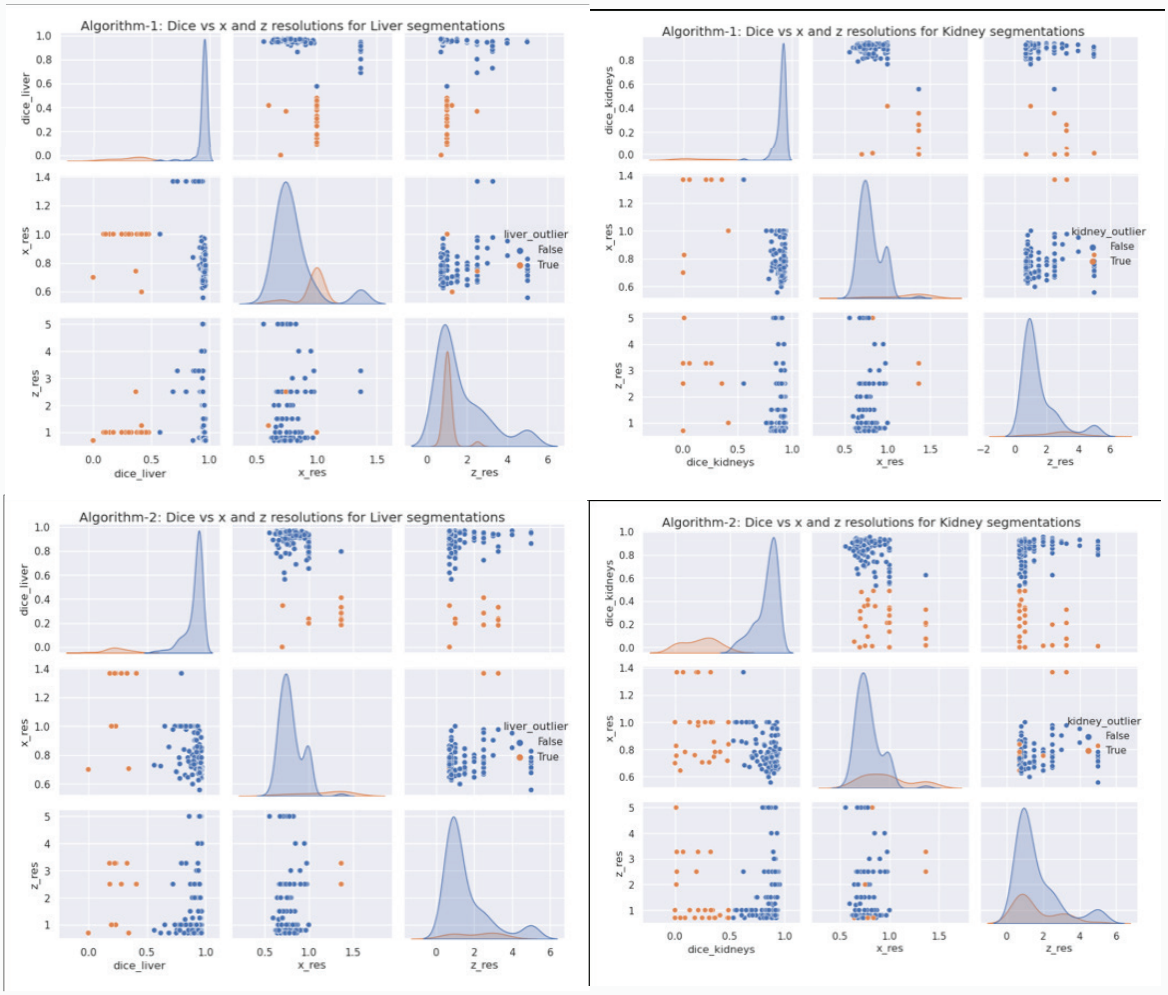


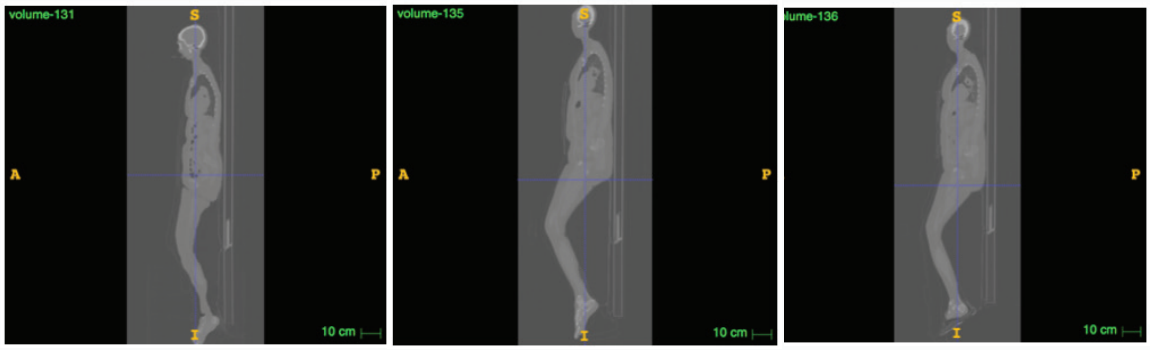
Figure 6: In the upper row, we plot scatter plots between the Liver and Kidney Segmentation Dice values and x and z-resolutions for Algorithm-1. In the lower row, the same analysis is performed for Algorithm-2. In all four plots, Blue indicates volumes for which Dice values are >0.5 and Orange indicates volumes for which the dice values <=0.5. Note that no clear patterns are observed between the Dice values and the image acquisition parameters x and z-resolutions.

Conclusions:

In this work, we tested the performance of two algorithms [1,2] on an independent dataset of 139 CT scans [3]. Algorithm-1 performed much better on the segmentation of the kidney (small organ). In contrast, the performance of the two algorithms was similar for the segmentation of the liver (large organ).

For both algorithms, a number of outliers (Dice <= 0.5) were observed. The data in the test set were available in NIfTI format and thus the only metadata available for the test dataset was the x, y and z voxel resolutions.. With these limited scan acquisition parameters, it was not possible to diagnose the root cause for the outliers.

For example, we could not ascertain if the algorithm fails on certain imaging conditions e.g. whole body CT scan, PET-CT scan or contrast CT). This work highlights the urgent need for complete DICOM header curation. The DICOM header information could help to pinpoint the scanning parameters that lead to segmentation errors by Deep Learning algorithms. We recommend that future public imaging datasets should store de-identified source DICOM files or at least a curated list of DICOM header values.



Algorithm-1 Dice: Liver=0.69, Kidney=0.0

Algorithm-2 Dice: Liver=0.28, Kidney=0.02t

Algorithm-1 Dice: Liver=0.87, Kidney=0.0

Algorithm-2 Dice: Liver=0.18, Kidney=0.02

Algorithm-1 Dice: Liver=0.88, Kidney=0.0

Algorithm-2 Dice: Liver=0.24, Kidney=0.07

Figure 5: By checking Dice scores and visually examining the images, we found that both algorithms performed poorly on kidney segmentations for the three volumes shown above (whole body CT). It could be that such images were not included in the training set.

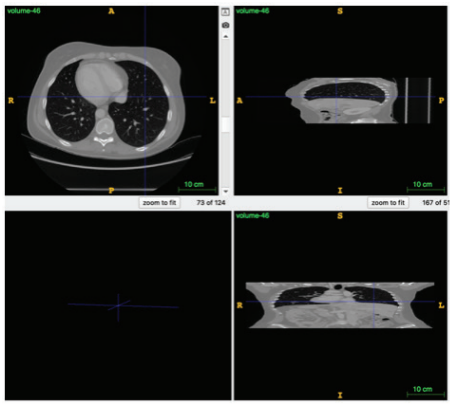


Figure 6: Poor performance due to incorrect data format. The axial resolution appears correct. However, in the coronal and sagittal planes we see the data is incorrectly compressed.

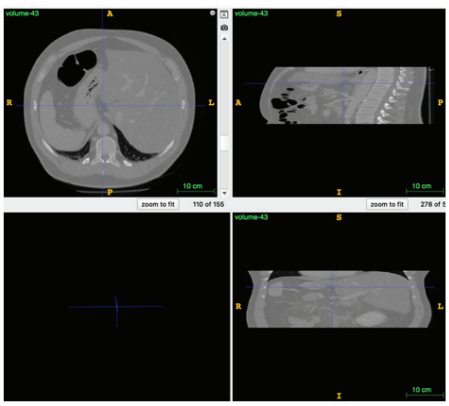


Figure 7: Both algorithms performed poorly due to a left/right shift in the data. The Liver should be on the right side but it appears on the left side of the volume.

References:

[1] J. Wasserthal et al., "TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images," Radiology: Artificial Intelligence, vol. 5, no. 5, p. e230024, Sep. 2023.

[2] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. Roth, and D. Xu, "Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images," arXiv [eess.IV], Jan. 04, 2022. [Online]. Available: http://arxiv.org/abs/2201.01266

[3] B. Rister, D. Yi, K. Shivakumar, T. Nobashi, and D. L. Rubin, "CT-ORG, a new dataset for multiple organ segmentation in computed tomography," Sci Data, vol. 7, no. 1, p. 381, Nov. 2020.

[4] Y. Tang et al., "Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2022, pp. 20698–20708.

[5] Jupyter_Notebook_For_Training