

# Deep Learning pipeline for the automatic segmentation and 3D model generation of the four heart chambers; applied uses in pre-surgical planning

## Introduction

Medical Imaging datasets, such as DICOM image datasets produced from CT or MRI scans, contain a wealth of information about a patient's unique anatomy. For example, this information can be invaluable for physicians when preparing for a surgical procedure. However, this information, especially when dealing with complex anatomical features such as the heart, is not easily utilizable. DICOM datasets are two-dimensional images that are stacked to create a three-dimensional representation of the cardiac anatomy. Analyzing a DICOM dataset is traditionally done by sweeping through slices of the two-dimensional axis to get an idea of the actual three-dimensional anatomy (Figure 1). This can be highly complex as well as very time consuming. However, computational anatomical segmentations allow for the creation of 3D anatomical models, which have wide use in pre-surgical planning; as they offer the physician a clear representation of the anatomical features while retaining all the spatial information within the clinical scan. Unfortunately, segmentation is a time-consuming process, even for highly skilled individuals. This creates a prohibitive bottleneck for surgical planning since a physician must receive a segmented model of the patient anatomy between the pre-procedure scan and the start of the procedure. This is often logistically impossible which restricts the quality of care available to the patient. Here we propose a deep learning pipeline that could automatically segment the heart chambers of a patient's CT scan and rapidly produce a 3D anatomical model for use in surgical planning.

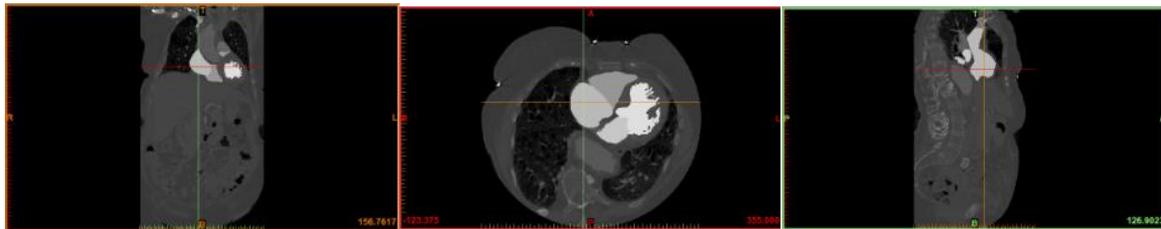


Figure 1: A patient CT scan with segmentations of the four heart chambers overlaid on the coronal plane (left), axial plane (middle), and sagittal plane (right).

## Methods

Our dataset includes 46 contrast-enhanced cardiac CT scans with each of the four chambers of the heart segmented. Pixel values in CT scans are recorded on the Hounsfield scale, represented by an integer between -1000 and 30000. Fortunately, the volumes of interest are only represented by a small range of brightness values, so we clip the Hounsfield units between -200 and 500 prior to normalization. Such a step was found to be critical as it counteracted under or over saturation in the images which would be problematic for a pretrained network. For ease of processing we also converted all DICOM slices and masks, from all three axes, to PNG files. During this step, we resized and interpolated the pixels to ensure the pixel spacing on each slice represented 1mm in the scan; i.e., since the distance a pixel represented differs on all three axes of these DICOM datasets. This crucial step ensures a consistent slice resolution across the three axes. Failure to do so results in distorted scan slices which decreases the performance of the network. The masks were produced simply by assigning the pixel values in each slice to 0, 1, 2, 3, and 4 corresponding to the right atrium, right ventricle, left atrium, left ventricle, and background respectively.

For our architecture we use a U-Net<sup>1</sup> with a pretrained resnet-34<sup>2</sup> network as the encoder, as implemented

in the Fast.ai library. We first trained the network with frozen weights for 10 epochs and then unfroze the weights and trained for 10 epochs with a lower learning rate. We used one-cycle policy<sup>3</sup> for both training cycles. For image augmentations we included rotations, zooms, lighting changes, and random crops at 256x256. We used cross entropy loss for our optimization objective.

For each study we evaluated each slice for all axes and the resulting activations were averaged between all planes to determine the label for each pixel in the 3D tensor. This output was resized back to its original shape before computing metrics. The marching cubes implementation in the scikit-image library was utilized to create a 3D surface mesh from the network segmentations for each heart chamber.

## Results

Creating an unambiguous metric for anatomical segmentation can be difficult, as a human segmenter needs to make many decisions as to where the true boundaries of a chamber are. Nevertheless, our network is robust to this variation and achieved results that matched well with the human generated masks (Figure 3 and 4). The performance of our network, defined by mean intersection over union, on all individual axis and when ensembling the axes is summarized below in Table 1. It was identified that our method of multi-axis ensembling is vital for yielding the best results.

	<b>Right Atrium</b>	<b>Right Ventricle</b>	<b>Left Atrium</b>	<b>Left Ventricle</b>	<b>Full Heart</b>
<b>Ensemble</b>	<b>0.8872</b>	<b>0.8257</b>	<b>0.9109</b>	<b>0.8803</b>	<b>0.8610</b>
<b>Axial</b>	0.8376	0.8122	0.8798	0.8736	0.8508
<b>Coronal</b>	0.8036	0.7786	0.8366	0.8598	0.8196
<b>Sagittal</b>	0.7228	0.7846	0.8351	0.8620	0.8011

Table 1: Comparison of mIOU for each chamber and ensemble

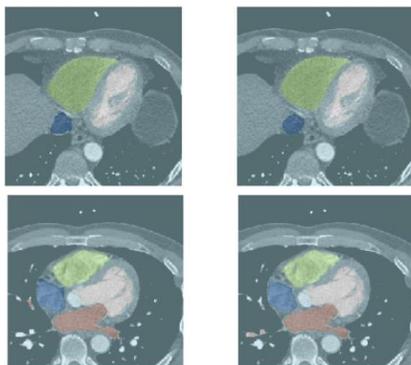


Figure 3: Ground Truth vs Predictions

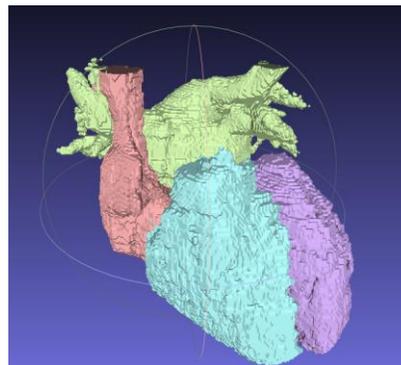


Figure 4: Network Generated 3D Model

## Conclusion

Our pipeline automatically creates a high-quality 3D model of the patient anatomy in mere minutes, removing the manual segmentation bottleneck from the pre-surgical pipeline. Importantly, this AI-assisted interpretation of medical imaging allows for the creation of a 3D model for every procedure with a pre-procedure scan, improving the physician's care team's ability to prepare for the procedure: e.g., determining upfront the accompanying therapeutic risks and/or improve the overall quality of care for a given patient.

## References

- [1] Ronneberger, O., Fischer, P., Brox, T., (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597.
- [2] He, K., Zhang, X., Ren, S., Sun, J., (2015). Deep Residual Learning for Image Recognition. arXiv:1512.03385.
- [3] Smith, L (2018). A Disciplined Approach to Neural Network Hyper Parameters: Part 1 – Learning Rate, Batch Size, Momentum, and Weight Decay. arXiv:1803.09820v2.