Part 1: Parse the *o-contours*
After building the pipeline, please discuss any changes that you made to the pipeline you built in Phase 1, and why you made those changes.

Major changes:
I modularized the code to process contours into its own function (process_contour) so it can handle both cases of i- and o- contours. During this process, I noticed several cases where there were inner contours with pixels outside of the outer contour. I therefore wrote a quality control function to raise an exception if this was found, encapsulated within a try: except: block to write these files for later logging. While most of these initially appeared in DICOM id SCD0000501, there were several others bad contours.

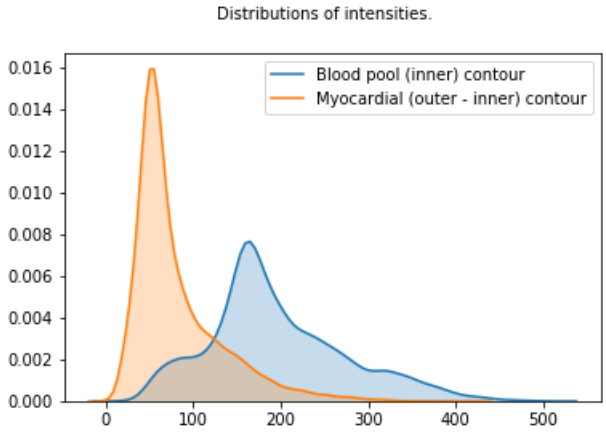
Minor changes:

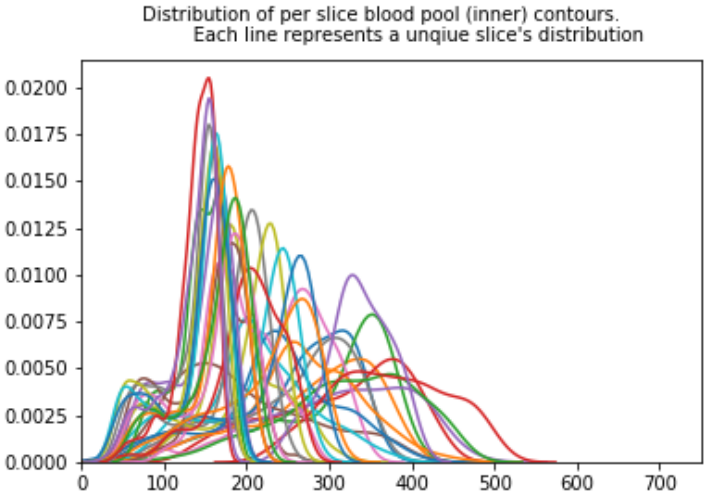
- Replaced i for _ in iterrows to reduce opportunity of conflicting i indexing for future
- Modified logging to display name of 'original_id' since we don't have scoping of the mapping to 'patient_id'
- Updated gitignore to include .DS_Store files on mac OS
- Updated write images to include subplots of both i- and o-contours
- Updated the way we determine the set of complete cases to include both i- and o-contours

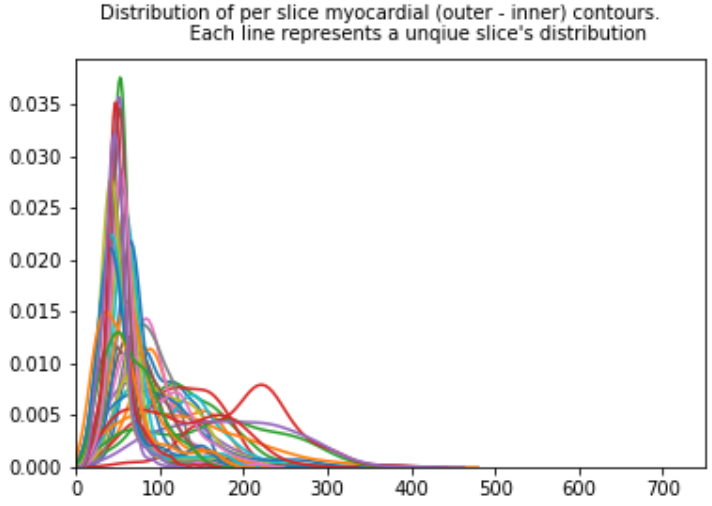
Part 2: Heuristic LV Segmentation approaches

2.1: Let's assume that you want to create a system to outline the boundary of the blood pool (*i-contours*), and you already know the outer border of the heart muscle (*o-contours*). Compare the differences in pixel intensities inside the blood pool (inside the *i-contour*) to those inside the heart muscle (between the *i-contours* and *o-contours*); could you use a simple thresholding scheme to automatically create the *i-contours*, given the *o-contours*? Why or why not? Show figures that help justify your answer.

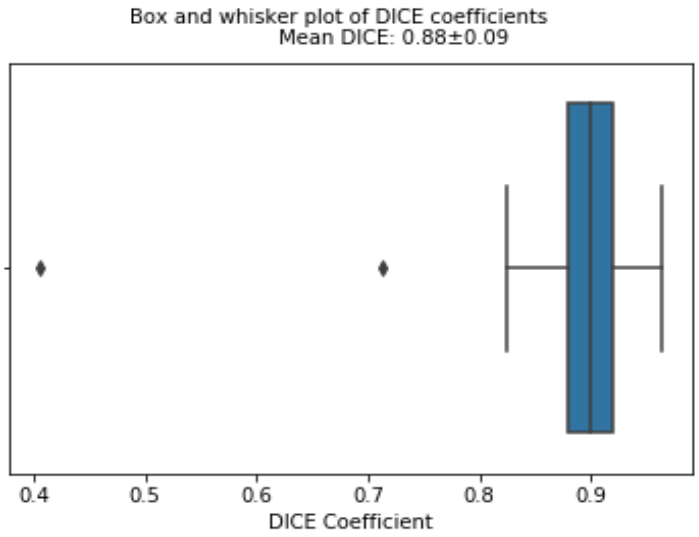
While there is a distinct difference between the intensities between the blood pool (inner) and myocardium (outer - inner) contours across all slices, this distinction alone does not appear to be sufficient to classify pixels into either myocardium or blood pool. Moreover, there appears to be heterogeneity across all slices with regard to distribution, making it hard to find a single threshold that will classify across all cases.



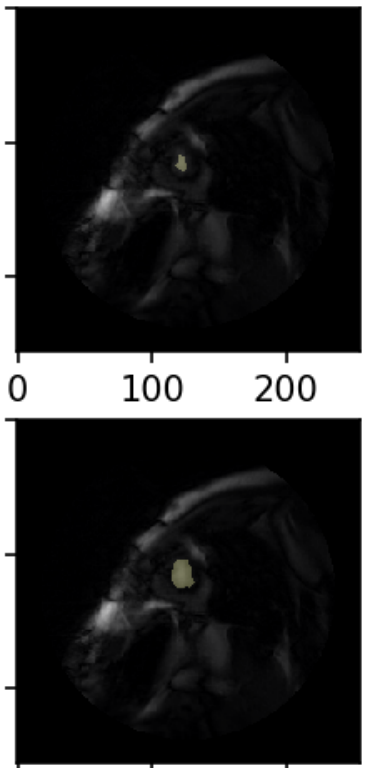




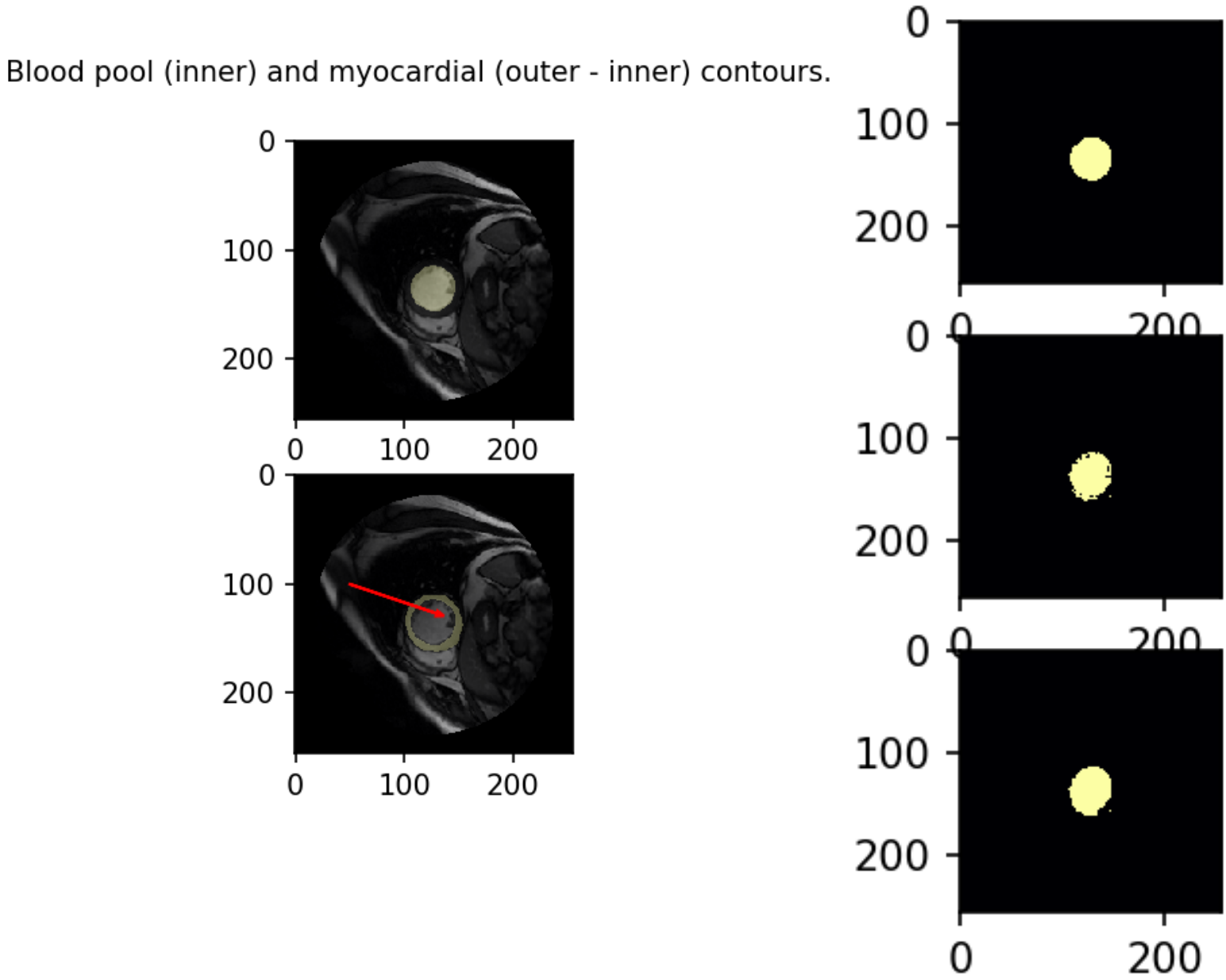
Since we are given the outer contour, I masked the input image with zeros outside the o-contour, and used Otsu's thresholding algorithm on a per slice basis to determine the threshold to use. I then threshold, and compare the DICE coefficient between threshold predicted and ground truth inner contours for blood pool. I show that the average DICE coefficient was 0.88±0.09.



The lowest DICE prediction was based on a slice near the apex (ground truth on top, threshold predicted on bottom).



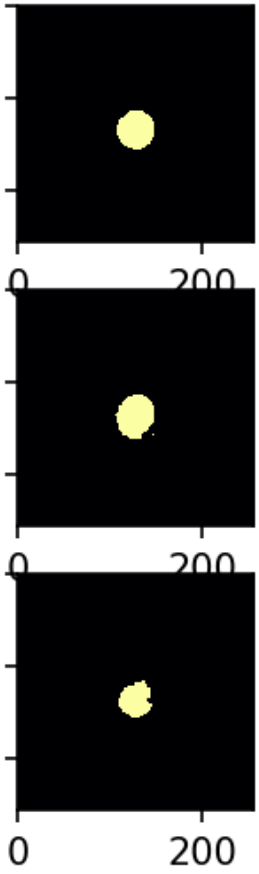
While these are promising initial results, there predicted segmentations tend to miss parts of the papillary muscle, which is often present in the blood pool contour. Example papillary muscle is shown with the red arrow on the bottom left image. One thing we can do is to clean up these results using closing. The right image has ground truth segmentation on top, threshold only approach in the middle, and closing (cleaning the boundaries) on bottom.

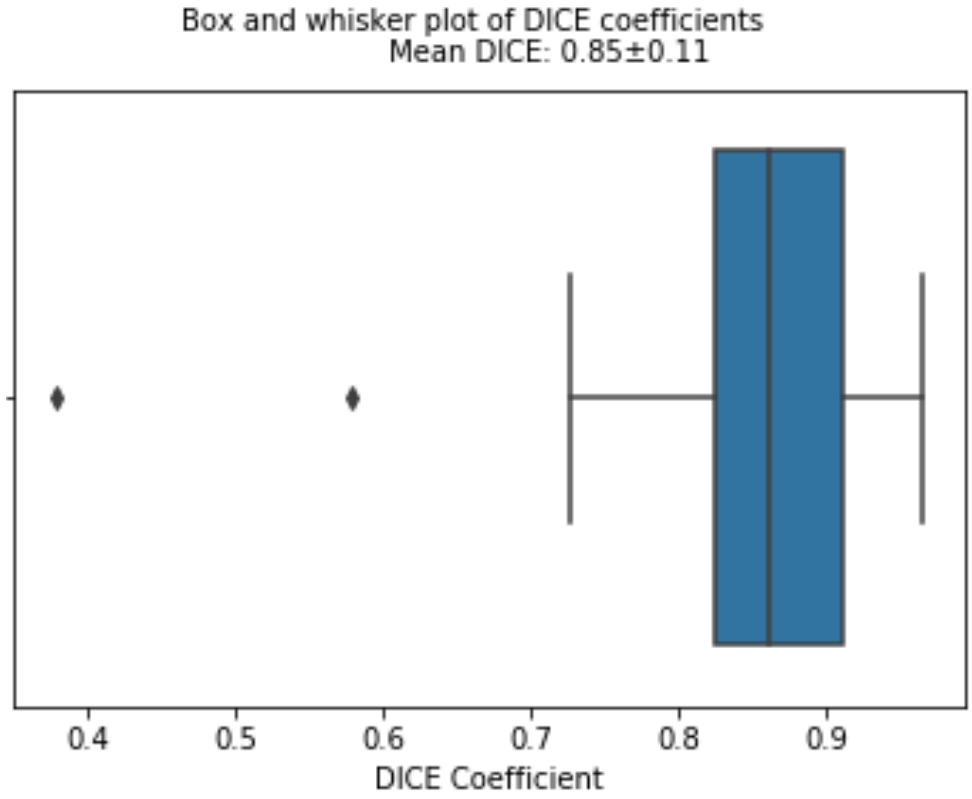


2.2: Do you think that any other heuristic (non-machine learning)-based approaches, besides simple thresholding, would work in this case? Explain.
Another heuristic that may work well is flood filling. Flood fill is a heuristic similar to that as a creeping wildfire, where it will expand to different areas depending on its accessibility (here determined by pixel intensities). This works on a similar basic intuition that was given for thresholding, where there is a pronounced intensity difference between the myocardium and the blood pool.

While it may initially appear that this approach would not be valid due to the necessity of starting point, this may not be the case. In almost all of the contours, the inner contour almost always in the middle of the outer contour. Therefore, a good seed point is the center of the outer contour. After flood filling, I then did post processing using closing and filling to remove any papillary muscle artifacts.

Picture on left is ground truth on top, threshold segmentation in middle, and flood fill on bottom. As you can see, both did alright, but each is missing a portion of the full blood pool. Perhaps if I had more time I would try several different heuristic methods and then ensample the methods. In any case, we got slightly worse DICE values.





2.3: What is an appropriate deep learning-based approach to solve this problem?
A simple deep learning (DL) strategy would be to use the input images to semantically segment the cardiac contours. Since we are given the outer contour already and we want to predict the inner contour, we could use the image and outer contour as two different inputs (i.e. image is a channel, while the outer contour is an additional channel. This alternatively could be done by using a multimodal model with two inputs.). The label would then be the inner contour. Adding the outer contour would provide the network more explicit information of where it should be "looking", and likely produce a more robust model.

For network structure, a fully connected network such as U-net or E-net would be an appropriate choice given their robustness in biomedical imaging datasets.

2.4: What are some advantages and disadvantages of the deep learning approach compared your chosen heuristic method?
While deep learning approaches have proven their ability to produce highly accurate segmentations for biomedical datasets, it has come at the cost of computational power and need for large datasets. A minimum "proof of concept" dataset would likely be ~250 slices. In order to get this many, interpolation of segmentations across slices may help provide this number of cases. However, curation of a significant number of slices near the apex may be necessary to get good segmentations of this anatomic area since it seems to be a consistent problem.