Part 1: Parse the *o-contours*

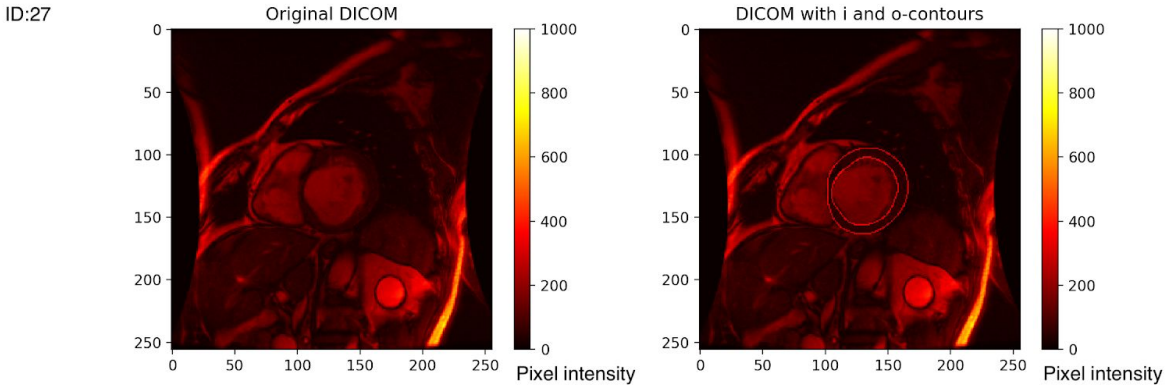
After building the pipeline, please discuss any changes that you made to the pipeline you built in Phase 1, and why you made those changes.

The code from phase 1 used dedicated class for loading data and methods for mapping DICOM files to their corresponding i-contour files and as such, no significant changes to the original pipeline were required to achieve part 1. The location of o-contour files is provided to the mapping function and the *ImageDataLoader* class this time returns a list of o-contour masks along with the i-contour masks of the corresponding DICOM images. The test cases are also updated accordingly.

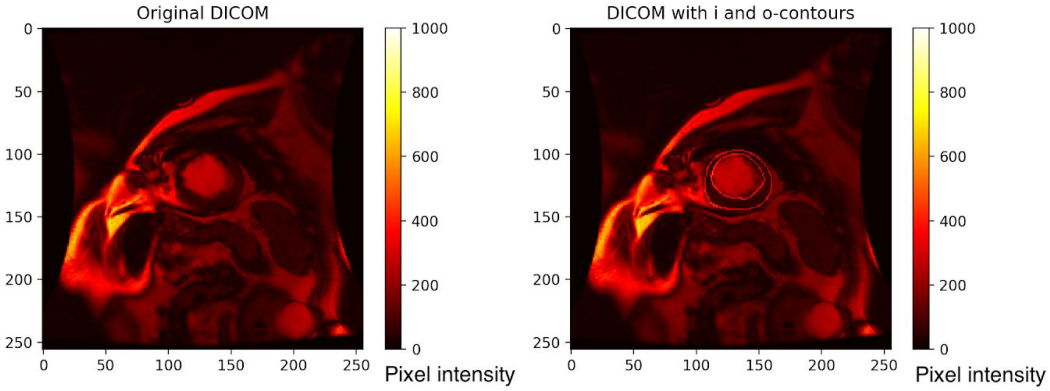
Part 2: Heuristic LV Segmentation approaches

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.

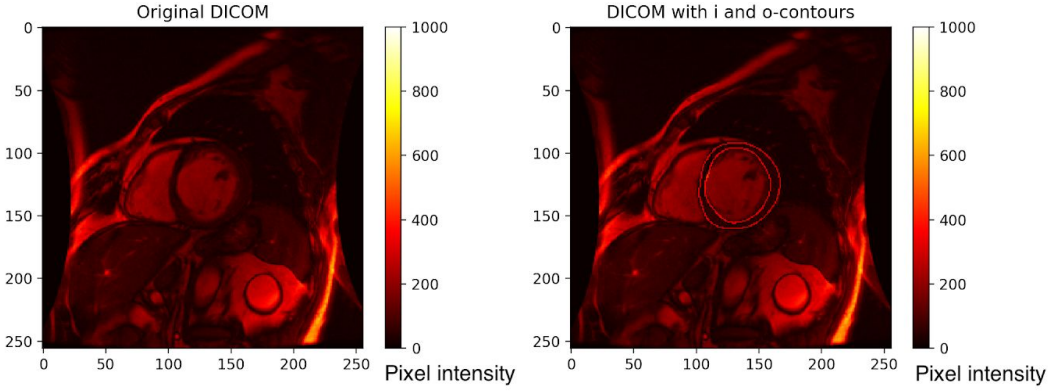
   Yes and No. By looking at a number of images, I am more inclined to say yes. Let's take a look at the heatmap of various DICOM images along with the contours.



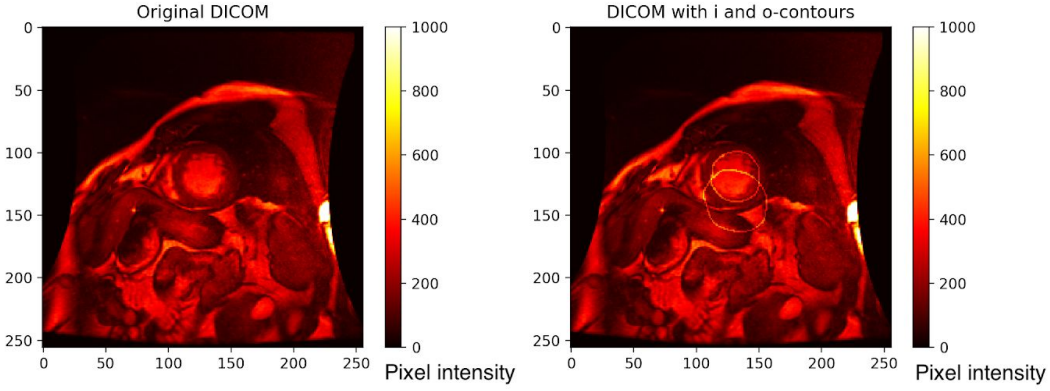
ID=31


Original DICOM
DICOM with i and o-contours
0
50
100
150
200
250
1000
800
600
400
200
0
Pixel intensity

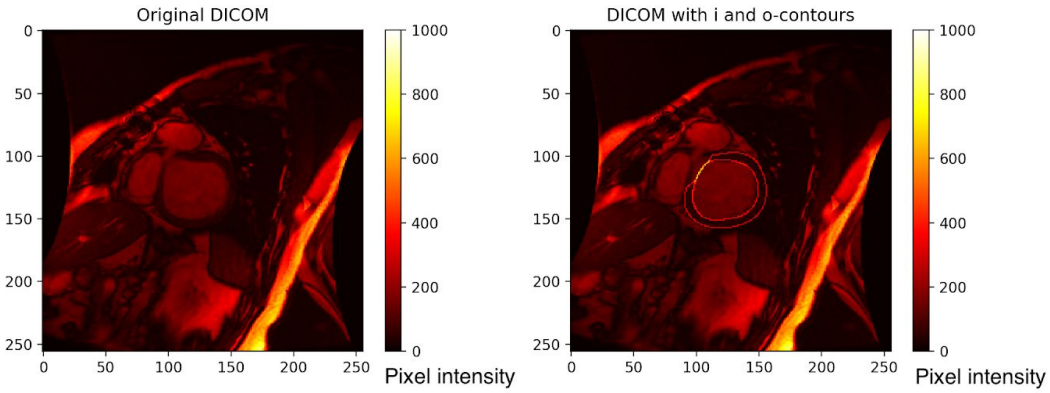
ID=2


Original DICOM
DICOM with i and o-contours
0
50
100
150
200
250
1000
800
600
400
200
0
Pixel intensity

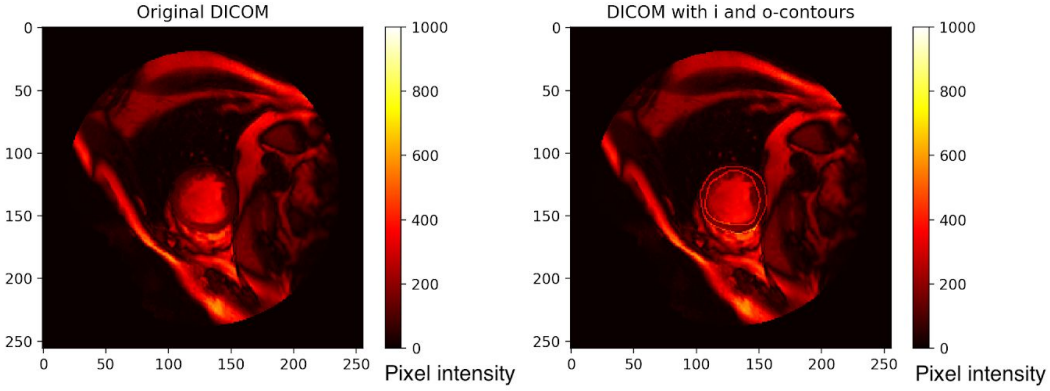
ID=9


Original DICOM
DICOM with i and o-contours
0
50
100
150
200
250
1000
800
600
400
200
0
Pixel intensity

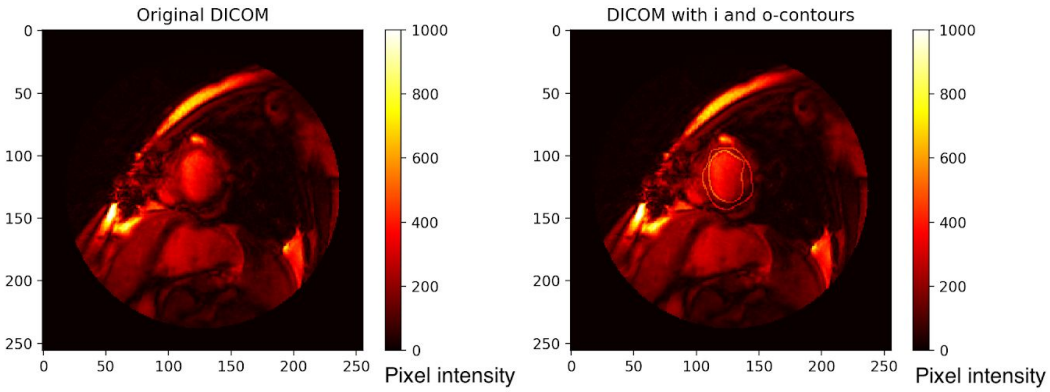
ID=19



ID=37



ID=22



The intensity values here are normalized (0-1000). We can notice that the intensity values of the region inside the blood pool are almost double the intensity values of the region inside the heart muscle (region between i-contour and o-contour). Based on this knowledge, we can find an appropriate threshold to identify the blood pool region.

However, this simple thresholding approach comes with limitations. Firstly, there isn't always a very distinct shift in the intensities from the heart muscle region to the blood pool region. This makes it hard to identify the exact location of the contour that separates the blood pool region from the heart muscle region. Secondly, in certain cases, there are some patches inside the blood pool region that have intensity values similar to the intensities in the heart muscle region, e.g., in the above image with ID=2. To deal with that, we will need some sort of smoothing.

One simple algorithm I can think of to find the i-contour, given the o-contour is to take the center of the o-contour as a reference point and from there, for each angle (degree or any not so small discrete angular step), find the corresponding distance from the reference point at which the chosen threshold is met. Subsequently, convert those polar coordinates into cartesian, fit a curve (angle against distance) making sure to avoid overfitting and finally, convert the fitted curve back to polar coordinates.

2. Do you think that any other heuristic (non-machine learning)-based approaches, besides simple thresholding, would work in this case? Explain.

   We can apply discontinuity based approaches to locate the i-contour region as intensity values of the neighboring pixels differ significantly around the contour area. We can use derivatives to find the discontinuity/edges (Canny, LoG, Robert's, etc.). We'll need to apply smoothing filters beforehand to overcome the same problems we faced with simple thresholding scheme.

   We can also use clustering based methods, e.g. K-means, DBScan, etc. that allow grouping regions of similar intensity levels. We can start from the center of the o-contour and keep adding pixels to the blood pool cluster based on a distance metric that depends on the distance from the center and intensity values of the neighboring pixels. In fact, once we have cropped the relevant region of the image, it would be possible to apply a histogram-based method to find the cluster of pixels that form the blood pool.

   Besides, there are various energy based, fuzzy logic based and hybrid models in the literature that are worthy of exploration in our case.

3. What is an appropriate deep learning-based approach to solve this problem?

   Our goal is to find a particular segment in the image, the blood pool region. Various Convolutional Neural Network architectures have empirically proven to be very successful in Semantic Segmentation tasks, where we wish to classify individual pixels in an image. In our problem, we'd like to find out which pixels belong to the blood pool region. One popular deep neural network for such biomedical image

segmentation tasks is the U-Net. U-Net is based on encoder-decoder style architecture. Its contraction or encoding path extracts more advanced features and reduces the size of the feature map. The expansion or decoding path recovers the size of the segmentation map and with the help of skip connections, gradients and localization information are propagated through the network. Among other popular architectures used for image segmentation task are SegNet and VNet.

4. What are some advantages and disadvantages of the deep learning approach compared your chosen heuristic method?

   Heuristic based methods discussed above require the application of some sort of smoothing or non-linearity to the inputs to accurately find i-contours in an imperfect image. These non-linearities are not always easy to model based on a pre-determined heuristic. Activation functions of neurons in neural networks help exactly with that, they provide non-linearities that could be trained. Unlike in heuristic based methods, part of the responsibility of feature engineering is automatically taken care of by the deep neural networks through their hidden layers. Moreover, deep learning methods have shown superior performance than heuristic based methods in a number of image processing tasks in recent years. Results do matter.

   The downside of the deep learning approach is the lack of transparency - it is not easy to understand why a network suggested a particular result. Secondly, a large number of hyper parameters need to be manually chosen including the basic model architecture (number and type of layers, operations on inputs, neurons per layer, etc.). Deep learning models can also be computationally expensive to train and may require a significant amount of data to get properly trained. Research is actively being done to overcome these shortcomings though.