ReadMe:

process_dicom_files.py

    Preprocesses data into a hdf5 storage container ready for training
    Inputs:
        contours: data/contourfiles
        dicom: data/dicoms
        mapping_df: data/link.csv
    Output:
        Writes files to output/contour_annotations.hdf5
        Writes images to output/patient_id_instance_number.png
        Writes log files to logs/dicom_processing_dicom.log

unit_tests.py

    unit tests for testing data generator

train_segmentations.py

    Trains a CNN for segmentation
    Inputs:
        output\contour_annotations.hdf5

Part 1.

How did you verify that you are parsing the contours correctly?

I wrote out images overlaid with the contour to an output directory. I additionally looked over the resulting images to manually verify that the contours were correctly overlaid.

What changes did you make to the code, if any, in order to integrate it into our production code base?

First, I moved the existing provided code into the src directory. I then added additional exceptions to catch potential areas where the code may fail due to trying to read in bad files. I finally added logging warnings to catch cases where there was a failure to read in data.

Part 2:

Did you change anything from the pipelines built in Parts 1 to better streamline the pipeline built in Part 2? If so, what? If not, is there anything that you can imagine changing in the future?

Knowing that for biomedical deep learning it’s important to make your test/train/validation splits across biological/patient replicates, making a subdirectory structure for the hdf5 helps facilitate this. This helps keeps a separation layer between each patient, which would be necessary down the road when making these data separation splits.

How do you/did you verify that the pipeline was working correctly?

I created a unit test for the DataGenerator to validate that A) we see all cases at least once per epoch (defined by steps_per_epoch) and B) each successive loop through the fit_generator will see a different order of examples.

Given the pipeline you have built, can you see any deficiencies that you would change if you had more time? If not, can you think of any improvements/enhancements to the pipeline that you could build in?

1. Given that we are missing some of the contours, interpolation of the contours in both slices direction and over cine time points of the cardiac cycle could help create a greater training set for the CNN to utilize.
2. Refactoring the data processing/training scripts to handle both inner and outer contours (perhaps given with command line arguments) would enable a more extensive use cases.
3. Preserving additional DICOM tags in the parse_dicom_files may be useful for additional preprocessing; such as standardizing FOV or rotating DICOMs into standard orientation.