NuNetSeg

Version 0.0.0 User Guide

Cameron Dennis Pain

July 17, 2020

Contents

Introduction

NuNetSeg is an implementation of a neural network based whole body CT segmentation algorithm designed for the purposes of nuclear medicine dosimetry. The NuNetSeg algorithm runs in a PACS style framework at a stand alone dicom location which is designed for easy workflow integration and generalises to various automated algorithmic processes. Version 0.0.0 of NuNetSeg is the first instance of this algorithm and so is yet to be thoroughly optimised and likely contains a few bugs relating to compatibility and the segmentation algorithm itself. If you come across any issues email cameron.pain@austin.org.au.

Set-Up

Dependencies

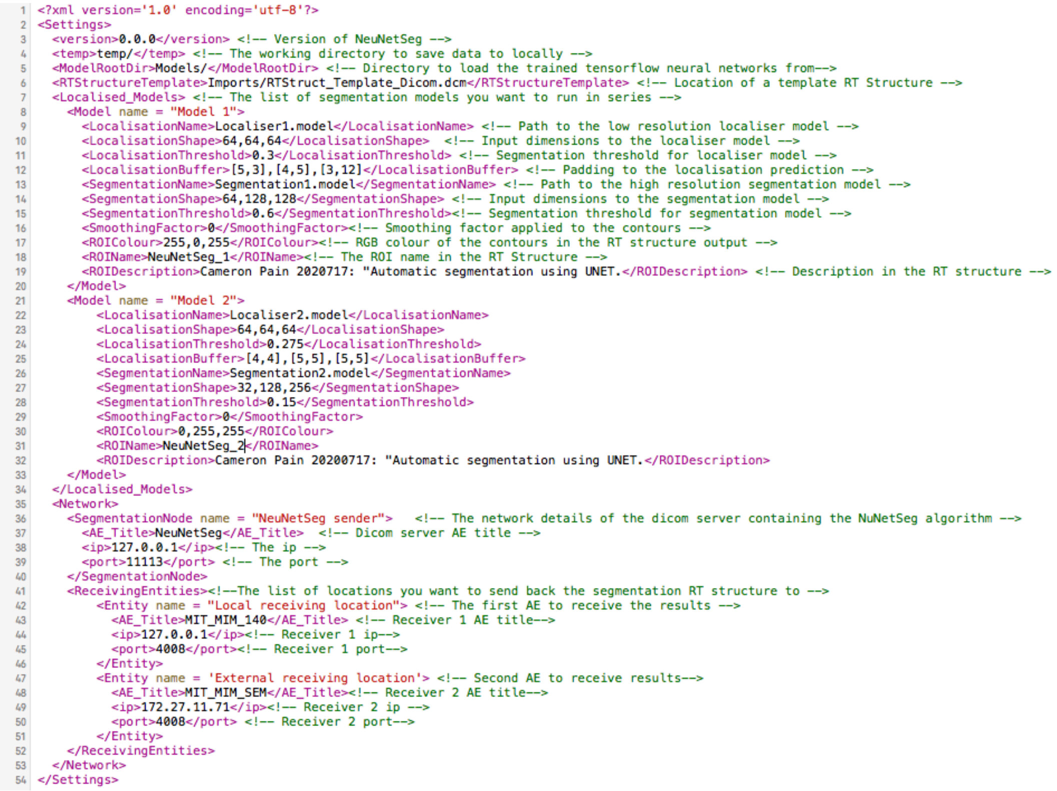
The following packages are required for running NuNetSeg0.0.0. The versions listed here are those for which the algorithm has been tested.

- python3.6.0
- numpy1.17.2
- matplotlib3.2.1
- scipy1.3.3
- pydicom1.4.2
- pynetdicom1.5.0.dev0
- tensorflow1.14.0

The algorithm has been ran on MacOS10.12 Sierra only. This shouldn't raise any issues with other versions of MacOS10, but it might raise issues with Linux installs. This has not been tested on Windows at all.

Initialising Algorithm Parameters

A .xml settings file is used to set the parameters of the segmentation algorithm upon initialisation. Figure 1 shows an example settings file with the appropriate .xml tree structure for parsing. The segmentation and localisation parameters set in the default settings file have been optimised for the default models. Adjustments to default parameters will likely degrade the quality of the segmentation.



```xml
<?xml version='1.0' encoding='utf-8'?>
<Settings>
  <version>0.0.0</version> <!-- Version of NeuNetSeg -->
  <temp>temp/</temp> <!-- The working directory to save data to locally -->
  <ModelRootDir>Models/</ModelRootDir> <!-- Directory to load the trained tensorflow neural networks from-->
  <RTStructureTemplate>Imports/RTStruct_Template_Dicom.dcm</RTStructureTemplate> <!-- Location of a template RT Structure -->
  <Localised_Models> <!-- The list of segmentation models you want to run in series -->
    <Model name = "Model 1">
      <LocalisationName>Localiser1.model</LocalisationName> <!-- Path to the low resolution localiser model -->
      <LocalisationShape>64,64,64</LocalisationShape>  <!-- Input dimensions to the localiser model -->
      <LocalisationThreshold>0.3</LocalisationThreshold> <!-- Segmentation threshold for localiser model -->
      <LocalisationBuffer>[5,3],[4,5],[3,12]</LocalisationBuffer> <!-- Padding to the localisation prediction -->
      <SegmentationName>Segmentation1.model</SegmentationName> <!-- Path to the high resolution segmentation model -->
      <SegmentationShape>64,128,128</SegmentationShape> <!-- Input dimensions to the segmentation model -->
      <SegmentationThreshold>0.6</SegmentationThreshold><!-- Segmentation threshold for segmentation model -->
      <SmoothingFactor>0</SmoothingFactor><!-- Smoothing factor applied to the contours -->
      <ROIColour>255,0,255</ROIColour><!-- RGB colour of the contours in the RT structure output -->
      <ROIName>NeuNetSeg_1</ROIName><!-- The ROI name in the RT Structure -->
      <ROIDescription>Cameron Pain 2020717: "Automatic segmentation using UNET.</ROIDescription> <!-- Description in the RT structure -->
    </Model>
    <Model name = "Model 2">
        <LocalisationName>Localiser2.model</LocalisationName>
        <LocalisationShape>64,64,64</LocalisationShape>
        <LocalisationThreshold>0.275</LocalisationThreshold>
        <LocalisationBuffer>[4,4],[5,5],[5,5]</LocalisationBuffer>
        <SegmentationName>Segmentation2.model</SegmentationName>
        <SegmentationShape>32,128,256</SegmentationShape>
        <SegmentationThreshold>0.15</SegmentationThreshold>
        <SmoothingFactor>0</SmoothingFactor>
        <ROIColour>0,255,255</ROIColour>
        <ROIName>NeuNetSeg_2</ROIName>
        <ROIDescription>Cameron Pain 20200717: "Automatic segmentation using UNET.</ROIDescription>
    </Model>
  </Localised_Models>
  <Network>
    <SegmentationNode name = "NeuNetSeg sender">   <!-- The network details of the dicom server containing the NuNetSeg algorithm -->
      <AE_Title>NeuNetSeg</AE_Title>  <!-- Dicom server AE title -->
      <ip>127.0.0.1</ip><!-- The ip -->
      <port>11113</port> <!-- The port -->
    </SegmentationNode>
    <ReceivingEntities><!--The list of locations you want to send back the segmentation RT structure to -->
        <Entity name = "Local receiving location"> <!-- The first AE to receive the results -->
          <AE_Title>MIT_MIM_140</AE_Title> <!-- Receiver 1 AE title-->
          <ip>127.0.0.1</ip><!-- Receiver 1 ip-->
          <port>4008</port><!-- Receiver 1 port-->
        </Entity>
        <Entity name = 'External receiving location'> <!-- Second AE to receive results-->
          <AE_Title>MIT_MIM_SEM</AE_Title><!-- Receiver 2 AE title-->
          <ip>172.27.11.71</ip><!-- Receiver 2 ip -->
          <port>4008</port> <!-- Receiver 2 port-->
        </Entity>
    </ReceivingEntities>
  </Network>
</Settings>
```

Figure 1: An example settings file used to initialise the parameters of the NuNetSeg algorithm.

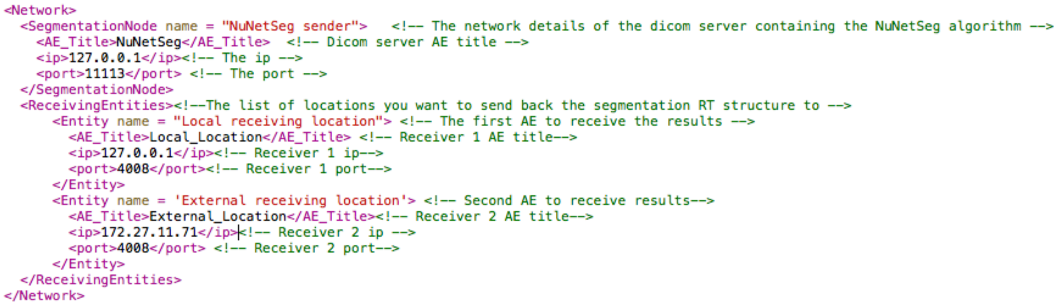
Network Settings

The network settings of the NuNetSeg dicom server and the receiving locations are specified. Multiple receiving locations can be specified by defining an <Entity> in the <ReceivingEntities> branch for each desired location. The NuNetSeg dicom node will receive promiscuously and send back to the specified location lists regardless of the details of the sending entity. The networking properties of the NuNetSeg framework will be made to allow greater user choice in sending and receiving data in future.

Running an Example

The following section will provide a step by step guide on running a segmentation example using the NuNetSeg algorithm.

1. **NuNetSeg network settings:** Open the settings.xml file and fill in the appropriate network details for your machine. If you are not connected to a LAN network, you can still run the segmentation algorithm on the 127.0.0.1 localhost IP address as shown in this example.



```
<Network>
  <SegmentationNode name = "NuNetSeg sender">   <!-- The network details of the dicom server containing the NuNetSeg algorithm -->
    <AE_Title>NuNetSeg</AE_Title>  <!-- Dicom server AE title -->
    <ip>127.0.0.1</ip><!-- The ip -->
    <port>11113</port> <!-- The port -->
  </SegmentationNode>
  <ReceivingEntities><!--The list of locations you want to send back the segmentation RT structure to -->
      <Entity name = "Local receiving location"> <!-- The first AE to receive the results -->
        <AE_Title>Local_Location</AE_Title> <!-- Receiver 1 AE title-->
        <ip>127.0.0.1</ip><!-- Receiver 1 ip-->
        <port>4008</port><!-- Receiver 1 port-->
      </Entity>
      <Entity name = 'External receiving location'> <!-- Second AE to receive results-->
        <AE_Title>External_Location</AE_Title><!-- Receiver 2 AE title-->
        <ip>172.27.11.71</ip><!-- Receiver 2 ip -->
        <port>4008</port> <!-- Receiver 2 port-->
      </Entity>
  </ReceivingEntities>
</Network>
```

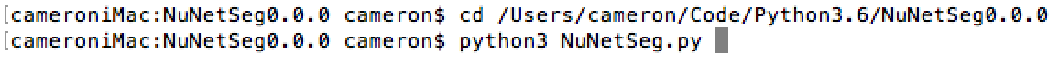
Figure 2: Specify the network details of the NuNetSeg algorithm, that is, the AE title, the IP address and the port to run on, then specify the network details of the receiving entities, for example the image analysis application on your work station.

2. **Receiving entities network settings:** Specify the NuNetSeg server network details on the receiving entities so that they will accept association from the NuNetSeg server and can receive the segmentation data. This will usually be just specifying the network location on which it runs, but you may also have to specify whether it is a query retrieve location, send location etc. The NuNetSeg server receives promiscuously, so any application entity on the network can send to it.

3. **Segmentation Models:** Simply delete or comment out the models in the settings.xml file you do not want to run in the algorithm. This can save time if you only want specific organs. It is not recommended to change

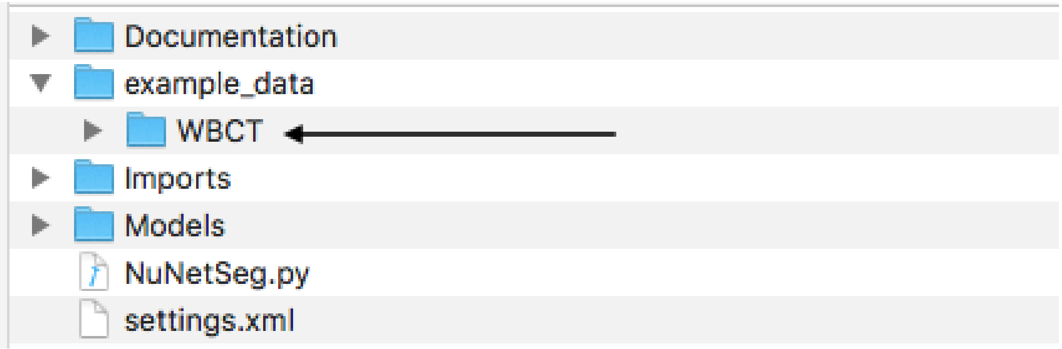
the segmentation or localisation parameters in the settings.xml file as these have been iteratively optimised and trained based on this specific preprocessing.

4. **Initialising the algorithm:** Once the settings.xml parameters and network details have been defined, we can initialise the model. Navigate to the NuNetSeg0.0.0 directory and run the NuNetSeg.py file. The model will take about 5 minutes to initialise after running this if you are loading all of the default models. During this time, the models are loaded and the dicom server is started.



```
[cameroniMac:NuNetSeg0.0.0 cameron$ cd /Users/cameron/Code/Python3.6/NuNetSeg0.0.0
[cameroniMac:NuNetSeg0.0.0 cameron$ python3 NuNetSeg.py
```

5. **Example WBCT:** Load the example whole body CT into the analysis environment you will send data to the NuNetSeg server from.



6. **Send WBCT to NuNetSeg:** Once the algroithm is initialised, send the WBCT from the analysis environment to the NuNetSeg server. On receiving the data, you will see the UID's of the data printed to terminal after which, the segmentation process will begin. All up, the segmentation will take about 80 seconds. In its current instantiation, the NuNetSeg0.0.0 algorithm will shut down the server while it processes the data which will prevent other users from accessing it. This will be the first upgrade made in moving to version 0.1. It should not cause problems if a single person is segmenting, however for future and more broad clin-

ical application, I will need to include functionality for queuing using a multithreading approach.

7. **Receive the data:** Upon completing the segmentation process, an RT structure with the segmented contours will be sent back to the specified receiving locations. The CT dataset which was segmented must also be present to view the segmentation.

Evaluation of the Segmentation Model

An evaluation of the segmentation algorithm with the seven default models was done on 10 wholebody CT datasets and compared to manual segmentations. The intersection over union and the relative difference in the ROI activity was used as a metric for testing the quality of the segmentation. Figure 3 demonstrates a box and whisker plot of the results.

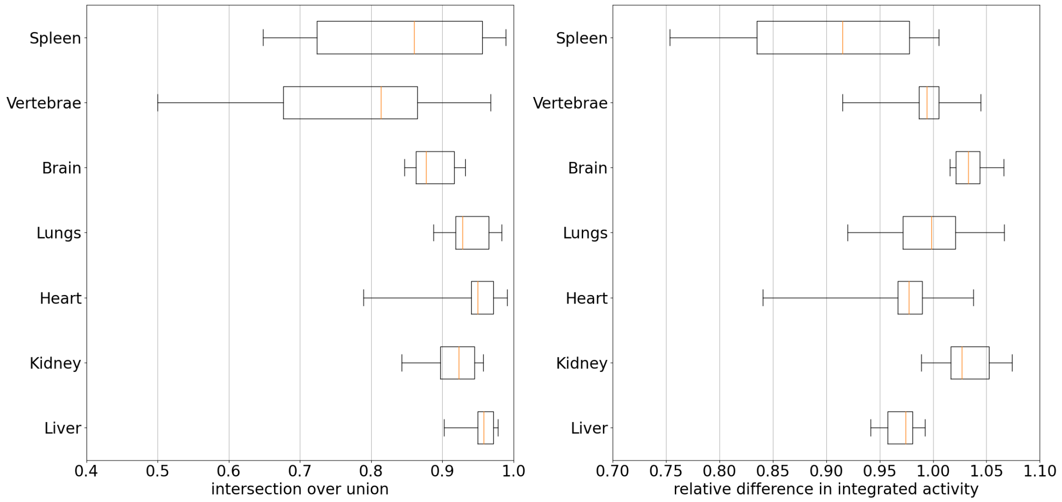


Figure 3: Results from comparing 10 manually segmented whole body CT scans to the NuNetSeg segmentations. The box represents the standard deviation of the results and the whiskers represent the maximum and minimum values from the datasets analysed.