Project Title: Tomographic Medical Image Reconstruction using Deep Learning

Group Members: Asher Burrell, Christopher Hinton, Ty Mercer

Faculty Advisor/Client: Dr. Debasis Mitra

1: Introduction

1.1: Purpose

The purpose of this document is to specify the functional, interface, and performance requirements of the software associated with our project “Topographic Medical Image Reconstruction using Deep Learning”.

1.2: Scope

This software will generate a 3D reconstruction of the heart from cardiographic SPECT data. This software will be made possible through a machine learning model, which requires a lot of data to make accurate 3D reconstructions. So, the project also focuses on generating synthetic 3D heart reconstructions.

The 3D heart reconstructions should be generated faster than from the current iterative reconstruction and optimization algorithms defined in [1], which run in 5-30 minutes, while ours will make a determination in milliseconds. In SPECT data, there are meaningless features of the data called noise, which can throw off reconstruction techniques. So, these 3D heart reconstructions should be unaffected by this noise.

Some requirements are labeled as stretch goals. Due to the experimental nature of this project, we cannot guarantee that these requirements will be met. However, we will make our best effort to meet these requirements.

1.3: References

[1] Bruyant, Philippe P. "Analytic and iterative reconstruction algorithms in SPECT." Journal of Nuclear Medicine 43, no. 10 (2002): 1343-1358.

[2] Chang, Haoran, Valerie Kobzarenko, and Debasis Mitra. "Inverse radon transform with deep learning: an application in cardiac motion correction." Physics in Medicine & Biology 69, no. 3 (2024): 035010.

2: Overall Description

2.1: Product Perspective

2.1.4: Software Interfaces

XCAT: XCAT, or eXtended CArdiac Torso, is a phantom image that shows the placement of different organs in a simulated human torso. These XCAT phantoms can be populated with statistically generated data representing different concentrations of radioactive tracer, which can be used in the OpenGATE simulation to produce sinograms.

OpenGATE: OpenGATE is a software that provides medical imaging simulations. When given a torso phantom from XCAT, OpenGATE can produce a sinogram by simulating the medical imaging process. This gives us a more realistic sinogram than a mathematical model would.

ImageJ: ImageJ is a tool that allows 3D binary images to be viewed. It builds images using voxels (3D pixels). Users can easily move between different slices of the image, resize the image viewer, and view the image from different planes.

Requirements

1. The system shall allow for the creation of XCAT phantoms using statistical data

1.1 The system shall include a variety of XCAT phantoms that can be populated with statistical data in particular organs

1.1.1 These phantoms should include values for the heart, left ventricle, right ventricle, and liver, which are distinct from each other and from all other organs in the phantom

1.1.1.1 These values should be at least 100, to avoid any potential conflicts with values representing tracer concentration.

1.1.2 All phantoms should use the same values for the heart, left ventricle, right ventricle, and blood pool

1.2 The system shall include a program which, given an XCAT phantom and one or more statistical distributions, can populate the specified organs of the XCAT phantom based on those statistical distributions

1.2.1 This program shall be written in Python and should be callable from the command line interface

1.2.2. This program should take its input from a file in its directory

1.2.2.1 This file should include a list of organs to populate with statistical values and, for each organ, the name and parameters for a SciPy distribution whose values should be used to populate the phantom

1.2.3. This program should output the populated XCAT phantoms to a subfolder in its directory.

2. The system shall allow for the creation of artificial sinogram data

2.1 The system shall include an OpenGATE simulation which, when provided with an XCAT phantom, can simulate the SPECT medical imaging process and produce a sinogram from the phantom

2.1.1 This OpenGATE simulation should use Monte-Carlo simulation techniques to mimic the random radioactive decay of particles

2.1.1.1 This simulation should interpret any values in the inputted XCAT phantom that are not one of the default organ values as representative of tracer concentration in a given part of an organ

2.1.1.2 This simulation should interpret each voxel in the inputted XCAT phantom as being a part of an organ, or a part of the background (the part of the torso that is not an organ).

2.1.2 This OpenGATE simulation should save the sinograms in a file directory specified by the user.

3. The system shall include a neural network that can reconstruct SPECT images from sinograms.

3.1 This neural network should be trained exclusively on artificial SPECT data

3.1.1 This artificial SPECT data should have a sinogram as its input and a reconstructed image as its output.

3.1.1.1 The artificial sinograms should be produced by the OpenGATE simulation described in Requirement 2.

3.1.1.2 The XCAT phantoms used for the generation of the artificial sinograms should be populated using statistical data generated from real reconstructed medical images.

3.1.1.3 The reconstructed images should be reconstructed from artificial sinograms using the iterative techniques described in [1].

3.1.2 The neural network should be trained on a minimum of 1,000 data points

3.1.2.1 The training data for the neural network should include at least 500 data points representing patients under stress conditions

3.1.2.2. The training data for the neural network should include at least 500 data points representing patients under stress conditions

3.1.2.3 The training data for the neural network should only include sinograms generated based on XCAT phantoms that include simulated values for the myocardium (heart) and their corresponding reconstructed images.

3.1.2.4 The sinograms used for training should be in the form of dicom (.dcm) files.

3.1.2.5 The reconstructed images used for training data should be in the form of raw binary files

3.1.2.6 The reconstructed images used for training data should be representative of 3d images of the shape 128x128x128, with each voxel represented by a 32-bit floating point number.

3.2 This neural network should take one or more sinograms as input

3.2.1 The input sinograms should be in the form of dicom (.dcm) files

3.2.2 The input sinograms should be in a folder in the same directory as the neural network.

3.2.2.1 The input sinogram should all be in the same folder

3.2.3 The neural network should individually reconstruct every sinogram in the input folder.

3.3 The neural network shall output a reconstructed image based on each inputted sinogram

3.3.1 The output image should be a 128x128x128 raw binary image, where each voxel is a 32-bit floating point number.

3.3.2 The output image should be viewable in ImageJ using the Input Raw feature.

3.3.3 All output images should be placed in a subfolder in the same directory as the neural network

3.3.3.1 All output images should go in the same folder

3.3.3.2 This folder should not contain any files or subfolders except for the output images.

3.3.4 The output image's name should include the name of the sinogram file used to construct it, or a unique identifier from that name, such that each input sinogram can be easily associated with its output image.

3.4 The neural network shall be interfaced via a Python program

3.4.1 This program shall be in the same directory as the neural network and the input/output folders

3.4.2 This program shall run in no more than one second per sinogram in the input file if there are any sinograms in the input file

3.4.3 This program shall call the neural network on every dicom file in the input folder, as though that dicom file is a sinogram

3.4.4 This program shall continue to run if it finds a non-dicom file in the input folder, ignoring the non-dicom file.

3.4.5 If this program encounters an exception while processing a sinogram, it should handle the error without crashing.

3.4.5.1 If such an exception occurs, the program should print an error message

3.4.5.1.1 This error message should include a description of the error encountered.

3.4.5.1.2 This error message should include the file name of the sinogram that was being reconstructed when the error occurred.

3.4.5.2 If such an exception occurs, the sinogram currently being reconstructed should be skipped, and the program should resume execution on the next sinogram in the folder.

3.5 (Stretch Goal) The neural network should output accurate reconstructed images, as tested on a validation set of real data.

3.5.1 The neural network will meet requirements 3.5.2-3.5.3 for a minimum of 80% of the data in the validation set.

3.5.2 The output images will be similar to the ground-truth reconstructed image as determined by a statistical analysis comparing the two images in the region designated to include myocardial data.

3.5.3 The output images will not contain extra data in the myocardial area where the ground-truth images do not.

4. The system will be secure.

4.1 The system will have no connection with external devices or servers

4.1.1 The system will be able to run without any internet connection or external interface beyond the programs mentioned in this document

4.1.2 The system will not attempt to communicate with any external server or device if such a connection is possible.

4.1.3 No program in the system will create or write to files outside of the directory that the program's executable or runnable file is located in.