

Engineering 2D Ground Truth and Labeled Images from 3D Models in the VMR

Noah Fields

Email: noahf@alumni.stanford.edu

Phone: 253-973-9404

1. Project Description

Dr. Gabriel Maher has created a convolutional neural network (CNN) which takes in ground truth medical images of vasculature and outputs a binary image, where the image is 240x240 pixels. The (CNN) performs well with the best performing model achieving DICE scores of 0.81. After looking at ways to fine-tune the model, we thought the best way to improve the models performance was to give he model more training data. To do this, we could no longer rely on the clinically segmented methods and had to institute a way to automatically generate tens of thousands of ground truth and labeled images. This code allows for tens of thousands of ground truth and labeled images to be created based on models found in the Vascular Model Repository (VMR).

2. Installation Instructions

Download the code from <https://github.com/noahfields1/synthetic-image-creation>. Use the file *img_env.txt* to install the conda environment on your computer. Make sure that you add a sub-directory called 'models', and place any VMR models you want to generate images for into this sub-directory. Additionally, make sure to put the corresponding centerline files into the 'centerlines' sub-directory. Centerlines files should have the file extension tag '.vtp'.

If you have made pre-existing images/files for a VMR model and you want to create new images/files for the VMR model, you need to first delete (or move them) the files and erase the directories leading to those files. In sum, the code will not create images for VMR patient models if it sees there are already images for the 3D model in the expected location.

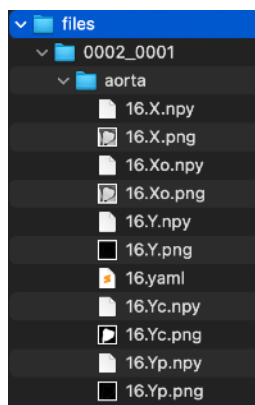
3. Usage

Once installed, all that needs to be done is to *cd* into the 'image-data-creation' directory and run the command line code 'python3 main.py'.

4. Outputs

4.1 Output Files

After the code has completed running, you will see outputs into the ‘image-data-creation’ directory. These include the *3D_points*, *files*, and *results* sub-directories as well as the *files.txt* and *files_tidy.zip* zipped file. The *3D_points* sub-directory is just an intermediate place to store 3D coordinates that are used for image creation, and can be deleted after the program runs. The *results* sub-directory is a vestigial output from when this code was adopted from Dr. David Parker and also can be deleted after the program runs (for the purposes of this project). The *files* subdirectory, however, is where all of the important files we need for our analysis reside. An example of the *files* outputted sub-directory can be seen to the left. This example is image 16 from the aorta of patient 0002_0001. Each individual image has 11 files that are related to it.



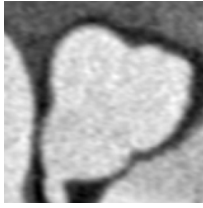
The *files_tidy.zip* is the zipped equivalent of the *files* sub-directory. The only caveat being that *files_tidy.zip* includes on *X.npy*, *Yc.npy*, and *.yaml* files as these are the only ones that are necessary to run Dr. Maher’s CNN; so only keeping 3 out of 11 files. The *files_tidy.zip* also only includes files that have undergone ‘tidying’, so vessels that are not circular or where the gradient is too low at the vessel boundary are not included in *files_tidy.zip*. You’ll want to use *files_tidy.zip* when you uploaded the images into the GPU to run the CNN.

When you unzip *files_tidy.zip*, you will also see a *files.txt*. You should also see a config file (of the form googlenet...{date}.yaml) recently created. This will be important to run the CNN later on.

4.2 Understanding the Image Files

Naming conventions are made to be compatible with Dr. Gabriel Maher’s original code for inputs and outputs into the neural network. Note that the example image used here is not a circular shape, so would not make it into the filtering of our dataset. The .npy versions of the .png files you see below are the same information, just neatly packed into a npy file which is what Dr. Maher’s neural network code uses.

4.2.1 X.png and X.npy (Ground Truth)



The image is `./files/0002_0001/aorta/16.X.png`. This is the Ground Truth image that is fed into the convolutional neural network. This image is more or less produced from Dr. Parker's work.

4.2.2 Yc.png and Yc.npy (Label)



The image is `./files/0002_0001/aorta/16.Yc.png`. This is the Labeled image that the convolutional neural network should hopefully spit out.

4.2.3 yamI

```
K: ./files/0002_0001/aorta/16.X.npy
Y: ./files/0002_0001/aorta/16.Y.npy
Yc: ./files/0002_0001/aorta/16.Yc.npy
bifurcation_id: -1
dimensions: 160
extent: 240
image: '0002_0001'
path_id: ''
path_name: aorta
point: 16
radius: 87.39
ratio (circum^2/area): 15.3
spacing: 0.029
```

The image is of `./files/0002_0001/aorta/16.yamI`. This yamI file is an essential part of Dr. Gabriel Maher's code and contains important information about the file in question. It contains the patient ('image'), the extent of the current image ('extent'): 240x240, and the dimensions which the image will be cropped to before feeding into the CNN ('dimensions'). The yamI most importantly, carries info to the pathway information for the 'X.npy' (Ground Truth) and 'Yc.npy' (Labeled) files.

As of right now, the radius, is not calculated in the same fashion that Dr. Maher calculated, and the spacing is set to a constant of '0.029'. This is the same as Dr. Maher's yamI file, yet I don't know how he calculated his.

— — — — — End of Absolutely Necessary Files to Run Dr. Maher's Code — — — — —

4.2.4 Y.png and Y.npy



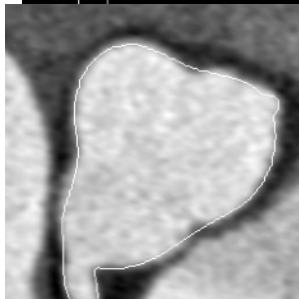
The image is `./files/0002_0001/aorta/16.Y.png`. The image represents the points that we are given from the VMR translated from 3D into a 2D plane. This image represents an intermediate step between the VMR 3D model and the 2D labeled image. For some images we may see multiple vessel outlines, however, this image shows just one.

4.2.5 Yp.png and Yp.npy



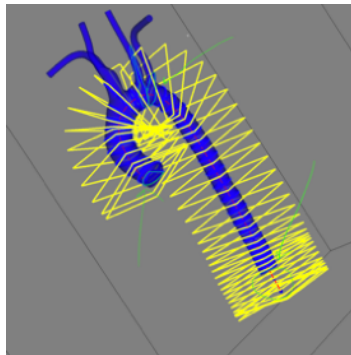
The image is `./files/0002_0001/aorta/16.Yp.png`. The image represents the outline for the vessel of interest. This must be produced after the corresponding Yc.png is produced.

4.2.6 Xo.png and Xo.npy



The image is `./files/0002_0001/aorta/16.Xo.png`. The image represents the outline for the vessel of interest overlaid on top of the ground truth image. This is meant to serve as a quick sanity check to see how the quality of the code's segmentations. To visualise many of these images at one time, simply run 'python3 GUIXo.py' in the command line making sure that the conda environment has been activated. This must be produced after Yp.png is produced.

5. Figures



This figure is an example of slices that are generated by Dr. Parker's code. The images take cross-sections of the model along the pathways. This code leverages this to make our images. There are 3 edge cases that we run into however (shown with green arrows).

The bottom left arrow shows a case when the aorta model is ending yet we still take a cross section here; this would yield a normal ground truth image yet the labeled image would be a semi-circle.

The bottom right arrow is a case where the pathway extends further than the model. This scenario also causes problems.

The top arrow is a situation where we see a bifurcation. There is nothing wrong with how the labeled images reflect the ground truth images. However, we are aiming for segmenting circular vasculature, so we are uninterested in these images.

6. Applications to Dr. Maher's Work

6.1 Download the CNN code from Github (https://github.com/gmaher/seg_regression)

Upload this directory into your GPU, and install the conda environment. The environment in the directory *requirements.txt* should be sufficient for training and testing the CNN. If not, the environment loaded earlier *img_env.txt* will work.

6.2 Upload Files into GPU

The files created are designed to be used to train Dr. Maher's segmentation. Upload the 'files.zip' folder onto the gpu where the CNN code is located. Unzip *files.zip*, and you will see the *files* sub-directory as well as the file *files.txt* and *googlenet_c30_train300k_aug10_clean_{mm:dd:yyyy}.yaml*.

6.3 Change config file in GPU

Edit the following pathways:

- DATA_DIR: /pathway/to/files_directory
- FILE_LIST: /pathway/to/files.txt
- NAME: unique_descriptive_name
- RESULTS_DIR: ./results
- LOG_FILE: ./results/unique_descriptive_name/log/train.txt
- ITER_FILE: ./results/unique_descriptive_name/log/iter.txt
- MODEL_DIR: ./results/unique_descriptive_name/model

The other parameters should be left alone, unless you are trying to change hyper-parameters of the CNN.

6.4 Run the CNN

Command Line: python3 train.py pathway/to/config_file.yaml TRAIN

7. Related Work in the Marsden Laboratory

This section provides links to relevant work:

Dr. Maher's Published work on CNN: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7785699/>

Dr. Maher's CNN: https://github.com/gmaher/seg_regression

Dr. Maher's Image-generation: https://github.com/gmaher/vascular_data

Dr. Parker's Volume Slicing:

8. Methods

Finding the Perimeter of a Vessel (produces Yp files)

File: perimeter.py

Function: extract_perimeter()

Making Xo images (perimeter overlayed on top of Xo)

File: Xo.py

Function: make_Xo_images()

Filtering Based on Eccentricity (How circular is our vessel)

File: tidy.py

Function: find_circular_vessels(), is_non_circular_vessel(image_data, threshold=0.75)

Filtering Based on Gradient (Is there a clear difference between vessel and outside?)

File: tidy.py

Function: filter_sobel(files, gradient_threshold=1)