**Master of Biomedical Informatics Capstone Project Progress Report**

**Student Name: April Yan**

**Student Email: yyan@hms.harvard.edu**

**Title of Project: Automatic Segmentation for Biomedical Images: A Deep Learning-Based, Generalizable Semantic Segmenting Tool for MRI and CT**

**Progress Report #: 2**

**Date Submitted: 09/30/2022**

**Mentor Name: Jayashree Kalpathy-Cramer**

**Mentor Email: JKALPATHY-CRAMER@mgh.harvard.edu**

**Co-Mentor: Jay Patel**

**Co-Mentor Email: jpatel38@mgh.harvard.edu**

**Is IRB Approval Required:**  Yes  No

**Date of IRB Application: N/A**

**IRB Number: N/A**

**Work accomplished to-date (200-300 words):**

*Outline all progress completed to-date, detailing any issues that have been encountered with data collection, analysis, writing and IRB approval. Please differentiate clearly between work completed yourself versus work performed by your project mentor and/or your project mentor’s lab members.*

I have finished building the pipeline by modifying code provided by my co-mentor. This pipeline contains numerous preprocessing steps for MRI (e.g., resampling, bias correction, normalization) and CT (e.g., resampling, windowing, normalization), which are slightly different. Resampling function was written based on a package of image processing, called Simple ITK. Bias correction was done through implementing Advanced Normalization Tool (ANTs) in python environment. Other steps were done based on built-in python functions and the package, NiBabel, which is used to process Nifti images. Then I made the algorithm to generate smaller images patches (i.e., cropped regions of interest) based on its labels so that we can localize and eliminate huge amount of non-labeled regions. These are positive samples. I also included patches cropped from background regions as negative samples since we believe the model could be robust if trained in this way. All the patches were fed into a more complicated version of U-Net, with more encoding and decoding blocks. The optimization was accomplished by minimizing dice coefficient loss for MRI images and joint dice cross entropy loss for CT.

I first trained a brain MRI model using the brain tumor images provided among the 10 medical segmentation decathlon data sets. This training went well because MR images are usually small (e.g., a few Mb) compared to CT images (e.g., hundreds of Gb). However, since there were about 500 brain images, each of which contains four channels, technically there were more than 2000 images, and each training takes around a week on an 8GB GPU. After this, I tried to train a pancreas model, but it always failed to output a prediction on pancreas and did poorly on predicting tumor. I have attempted tons of combinations of hyperparameters, such as patch size, learning rate, augmentation methods, pooling layers, weight factor of loss, etc. After six weeks, I still haven’t solved the problem. It took so long mainly because one training takes about a week. Then I went back the images themselves and found that resampling images using NiBabel didn’t save labels as integers, float instead, which caused trouble in our data loader. That’s why the data loader can’t generate patches localized as regions of interest, meaning that the positive sample were likely to be negative as well. I used Simple ITK for resampling and solved the problem.

**Modifications to original project plan (Bulleted List):**

*If changes have been made/need to be made to your project plan since the time of your project proposal based on oversight feedback, consultation with your project mentor and/or any difficulties mentioned above, please detail these changes.*

* *Training takes very long time and a lot of computation so there is no way to do automatic model tuning, like grid search. I am just going to tune hyperparameters myself and see if my co-mentor and I both are satisfied with the prediction.*
* *I probably won’t have enough time to train all 10 datasets, so we decide to pick two CT (e.g., pancreas+tumor, liver+tumor) and two MRI (e.g., brain+tumor, hippocampus) data sets.*

**Does the work accomplished so far align with your project proposal timeline?**

Yes  No

**Results**

Either below or attached, include any and all results obtained so far in the course of your capstone project.

Pancreas model results before debugging: (couldn’t predict anything about pancreas but very limited areas of tumors)

Chart, box and whisker chart

Description automatically generated

Current results after debugging:

Chart, box and whisker chart

Description automatically generatedA picture containing graphical user interface

Description automatically generated

Chart, box and whisker chart

Description automatically generatedGraphical user interface

Description automatically generated

**Figure 1**. Prediction results of MRI (brain tumor) and CT (pancreas + tumor) models trained using auto-segmentation pipeline. Left panels show the distribution of volumetric dice coefficients for each class. Right panels show examples of predicted labels of different classes in different colors.

Brain: red 🡪 non-enhancing tumor, yellow 🡪 enhancing tumor, green 🡪 edema

Pancreas + tumor: green 🡪 pancreas, yellow 🡪 pancreatic tumor

As expected, non-enhancing tumor would be slightly more difficult to quantify than the other two, where there is more contrast around edema and enhancing tumor (Norden et al., 2012). Also, the pancreas model did much worse on tumor prediction in validation set. The pancreas + tumor is not so hard to predict but the contrast between pancreas and tumor may be not enough to robustly distinguish between pancreas and tumor and separate them. However, it is not convincing either to be so determined since I am still tuning this pancreas model by modifying hyperparameters, which perhaps will have a more robust model for tumor prediction.

**Next Steps (Bulleted List):**  *List out the steps you plan to complete before your next submission.*

* *Keep tuning the pancreas model for tumor prediction*
* *If have more time, use the pipeline to train another two tasks, one for liver + tumor CT segmentation and another for hippocampus MRI segmentation (****Actually no. My mentors want to do something more challenging, so they give a visceral data set with 20 organ notations to see if this pipeline works. We’ll see how that goes.****)*
* *Compile code, model weights, and results 🡪 upload them to a public GitHub repository*

**Journal Selected for Submission: nature method**

**Journal Instructions for Authors Link:**

* [**https://www.nature.com/nature/for-authors/formatting-guide**](https://www.nature.com/nature/for-authors/formatting-guide)
* [**https://www.nature.com/nmeth/submission-guidelines**](https://www.nature.com/nmeth/submission-guidelines)

**References:**

Norden, A. D., Pope, W. B., & Chang, S. M. (2012). Current concepts in Brain tumor imaging. *American Society of Clinical Oncology Educational Book*, (32), 119–124. <https://doi.org/10.14694/edbook_am.2012.32.119>

Package documentation & code references:

1. Simple ITK: <https://simpleitk.org>
2. NiBabel: <https://nipy.org/nibabel/>
3. ANTs: <http://stnava.github.io/ANTs/>