

Brain Tumor Segmentation using U-Net and Callback

Siyuan Chen siyuanch@kth.se

Introduction

Brain tumor segmentation consists of separating the different tumor tissues (active tumor, edema and necrosis) from normal brain tissues. The purpose of this project is to implement a deep learning model for better tumor segmentation on various kinds of brain MRI images. The chosen framework is U-Net, and the used dataset comes from the BraTS, which is an open-access dataset with 494 subjects' MRI images with brain tumors.

The main idea of this project is to build the code framework to achieve a multi-label segmentation task combined with all the knowledge I learned from the previous lab works. The data were divided into a training set and a validation set. The segmentation is based on the U-NET, the fitting process is combined with the callback tech, and various metrics are used in this project to evaluate the model performance.

Methods

Processing the data

Read the list of all the folders and convert the list to id. The final data are made up of train, test, and validation. The image size is set as 128×128 with two channels. For each subject, there are five different kinds of file (Flair, t1, t1ce, t2, mask). The flair file and the t1ce file are inputted into two channels, and there are 4 labels in the Mask, where 0 represents background, 1 means core, 2 means edema, 3 means enhancing.

Besides that, considering the first several and the last several slices don't have enough useful info for segmentation between tumor and normal brain tissue, the middle 100 slices from the 22th slice are picked as the training data, which means the final dimension of the data is $100 \times 128 \times 128 \times 2$.

U-Net Architecture

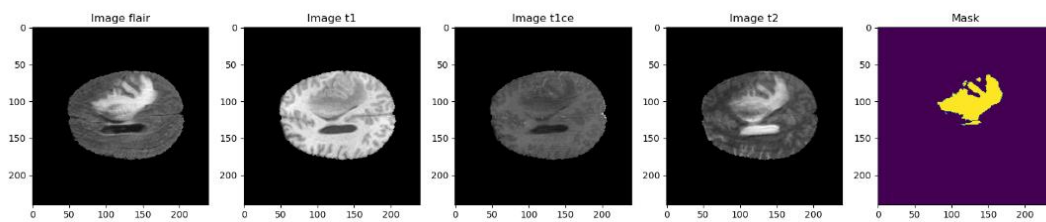
The framework used for this project is U-Net (In total 4 concatenate layer which is same with what we built in Labwork). However, combined with the original U-Net, I add the callback for training process to make the model's training process focus more on the test data.

The callback setting includes two parts. 1. I use CSVLogger to save my history, and use ModelCheckpoint to direct my training model develop towards the direction of making good validation result.

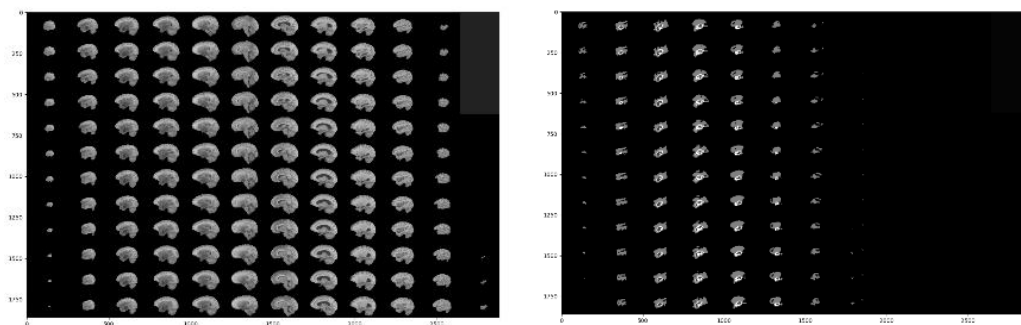
Results

Show the data

For each training data, there are five files. Below I show the first sample as the example.



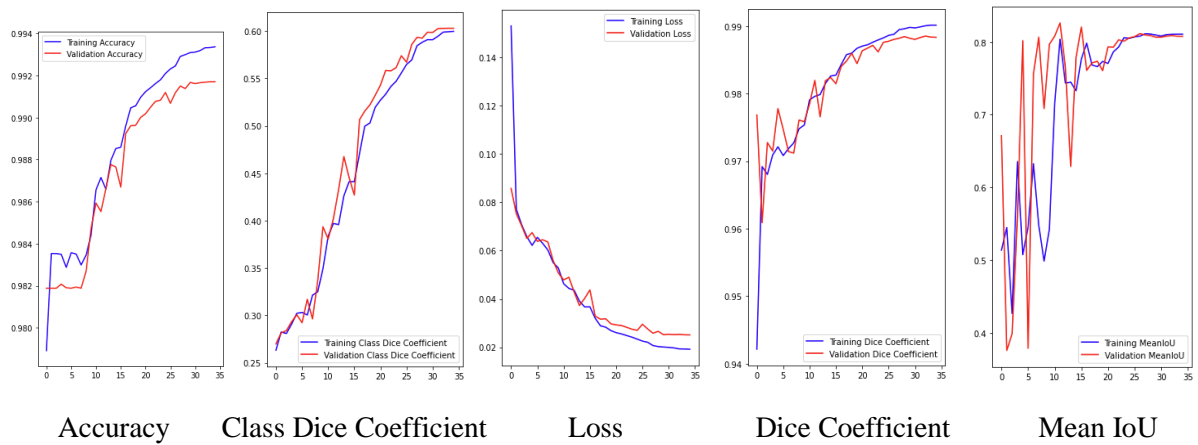
Different slices of this subject's T1 image and mask image are shown as below.



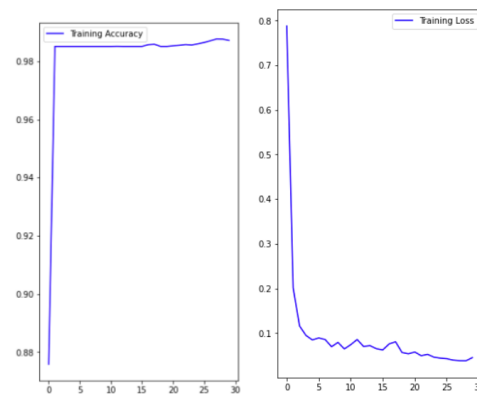
U-Net Model Result

Compile:

Use the categorical_crossentropy as loss function, and use Adam as optimizer, and use various metrics. Epochs is set as 30 combined with the callbacks setting. Below shows the accuracy result and the loss results with the callback. Compared with the accuracy result and loss results without a callback, we can see the result with callback has better performance, and the accuracy can increase with the increase of epoch.

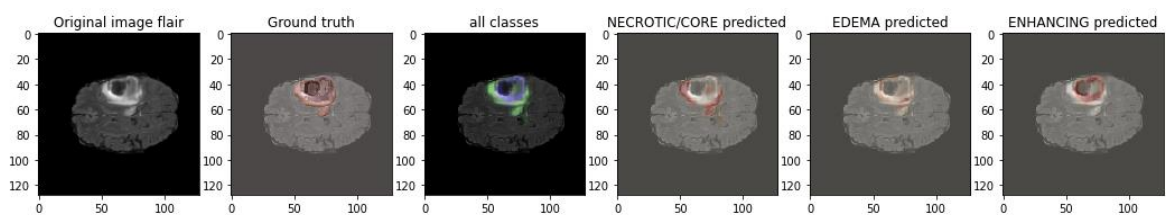


Below shows results without callback setting.



Predicted Result

Define visualization functions to visualize the predicted tumor and compare it with the real one. Below you can see the predicted tumors result and different types of tumor in the brain. (multi-label)



Below compare the difference between the real tumor and the predicted tumor of one training data.

