北京郵電大學
Beijing University of Posts and Telecommunications

Queen Mary
University of London

Undergraduate Project Report
2021/22

An algorithm for 3D MRI brain tumor segmentation

Name:	Wanqi Dong
School:	International School
Class:	2018215121
QMUL Student No.:	190019048
BUPT Student No.:	2018213196
Programme:	Internet of Things Engineering

Date: 14-04-2022

Table of Contents

Abstract ***2***

Chapter 1: Introduction ***4***

Chapter 2: Background ***7***

2.1 "Brain Tumor Segmentation Challenge 2020 (BraTS 2020)" **7**

2.2 Details about the Dataset **7**

2.3 Tumor regions and ground truth labels **8**

2.4 Related work **9**

2.5 Related technology **10**

2.5.1 Simple-ITK package 10

2.5.2 ITK-Snap Software 11

2.5.3 3D U-Net 12

2.5.4 Attention mechanisms and CBAM module 13

Chapter 3: Design and Implementation ***15***

3.1 Data pre-processing **16**

3.1.1 Normalization 16

3.1.2 Data cropping and padding 17

3.2 Data augmentation block **18**

3.3 Loss function and loss computation **19**

3.4 Optimizer and learning rate **20**

3.5 Model: Atten_Unet **21**

3.6 Evaluation metrics for validation **24**

3.6.1 Dice Similarity Coefficient 24

3.6.2 Hausdorff distance 25

3.6.3 Sensitivity and specificity 25

3.7 Training and Validation **26**

3.8 Testing **27**

Chapter 4: Results and Discussion ***29***

4.1 Segmentation results **29**

4.2 Qualitative results **31**

4.3 Comparison between methods **34**

4.4 Discussion **35**

Chapter 5: Conclusion and Further Work ***37***

References ***38***

Acknowledgement ***40***

Appendix ***41***

Risk and environmental impact assessment ***58***

Abstract

In recent years, the revolutionary development of artificial intelligence technology has stimulated huge advances in many fields like transportation, education and medical field. In the medical field, the automatic segmentation of 3D MRI brain tumor data plays an important role in tumor diagnosis and treatment. In this project, I proposed a novel algorithm for "3D MRI brain tumor segmentation". A new neural network called Atten_Unet was created based on the 3D-Unet baseline by modifying each layer of the architecture. In order to take the advantage of Transformer into consideration, attention mechanism is also applied to the new network by adding the CBAM module to each block of the Atten_Unet. In addition to this, I also came up with various data preprocessing and data augmentation strategies to refine the algorithm. Good 3D MRI brain tumor segmentation results have been achieved by training on just over half of the data in the original "BraTS 2020 training dataset". After applying the algorithm on the validation dataset, the DSC scores used to measure brain tumor segmentation performance reached 85.39%, 90.04%, and 88.65% for the three tumor subregions ET, TC, and WT, respectively, surpassing the results obtained by the winning team in the BraTS 2020 challenge. This suggests that the algorithm designed in this project is a meaningful practice in brain tumor segmentation and has great potential in the field of medical automation.

Keywords: 3D MRI brain tumor segmentation, Atten_Unet, CBAM, DSC score

摘要

近年来，人工智能技术的革命性发展带动了交通、教育、医疗等诸多领域的巨大进步。在医学领域，在医学领域，实现 3D MRI 脑肿瘤数据的自动分割对肿瘤的诊断和治疗有着重大的意义。为了充分利用 Transformer 的优势，我在 3D-Unet 网络的基础上，利用 CBAM 模块对神经网络添加注意力机制，并对网络每一层的结构进行了改动，提出了一个叫做 Atten_Unet 的新型网络。除此之外，我还采取了各种各样的数据预处理和数据增强策略来完善算法。在仅仅使用 BraTS 2020 原训练集中一半多数据的基础上，实验成功地对脑肿瘤的 3 个子区域进行了分割，并展现了非常优越的分割结果。在验证集上测试算法后，用来度量脑肿瘤分割表现的 DSC 分数在 3 个肿瘤子区域 ET,TC 和 WT 分别取得了 85.39%，90.04%，88.65%的结果，超越了 BraTS 2020 挑战赛中获得冠军的团队得出的结果。由此可见，本项目设计的算法对于脑肿瘤分割领域来说是一次非常有价值的实践，在医疗自动化领域展现出了巨大的潜力。

关键词：3D MRI 脑肿瘤分割，Atten_Unet，CBAM，DSC 评分

Chapter 1: Introduction

Recent advances in artificial intelligence have fostered disruptive innovation in many areas, including health, transportation, engineering, and various other aspects. Deep learning technology has also demonstrated high accuracy and advantages in decision-making in these areas, especially in medical area. In healthcare, the application of artificial intelligence and deep learning brings new prospects for development, such as the automatic detection of diseases like cancer.

In clinical practice, doctors could utilize MIR imaging technology (Magnetic Resonance Imaging)(Myronenko and Hatamizadeh, 2020) to observe and visualize the tissues inside the human body, so as to make diagnosis and prognosis of different diseases. Among these diseases, brain tumours are perhaps the most malignant one and have claimed millions of human lives. The deterioration and spread of brain tumor can be a fatal threat to human life and people's health conditions.

Tumors can be distinguished into 3 parts according to the degree of deterioration, which are: necrosis, enhancing tumor and edema(Myronenko and Hatamizadeh, 2020). The distribution of these 3 parts in the spatial area is roughly from inside to outside. Using MRI technology for fluoroscopy, physicians are able to observe brain tumors in different regions, identify regional differences in tumors, and make recommendations for surgery to remove certain tumors regions. However, visual observation of the various parts of the tumor may deviate greatly from the actual situation. Therefore, manual segmentation of the various parts of the tumor may be inaccurate, which may lead to errors during tumor resection surgery and have adverse post-operative effects on the patient. In addition, the skill level of different doctors varies greatly. doctors lacking experience may misjudge the segmentation of each region of the tumor and incorrectly leave part of the necrotic region intact or incorrectly remove part of the normally working brain cells during the resection surgery, which could lead to devastating result. In addition, manual segmentation of tumors is slow and costly, potentially delaying the best time for patients to be treated.

Automated brain tumor segmentation can provide useful reference suggestions for tumor diagnosis, segmentation, and treatment, and can be a valuable aid to physicians in the treatment and monitoring of brain tumors.

For all these reasons, in this research project, I applied various techniques related to artificial intelligence and deep learning to automatically segment three different parts of brain tumors:

TC, ET and WT. After extensive research in related medical and technical fields, I designed a new algorithm to automatically segment brain tumors from 3D MRI image and presented the segmentation results as three overlapping tumor regions with different colors: TC, ET and WT, which represents enhancing tumor, tumor core and whole tumor respectively(Henry et al., 2021).

In the project, I use the BraTS 2020 dataset for experiment, which is a well-established dataset containing various 3D MRI brain tumor data and their annotated segmentation files. First, various pre-processing method were performed and to transform the 3D MRI images into a format suitable for neural network training. Secondly, to increase the richness and completeness of the 3D MRI brain tumor dataset, I use a variety of innovative data augmentation approaches. Then, using the generalized CNN architecture, I designed a new network based on creative adaption. I combined the attention mechanism used in Transformer architecture with a common CNN image segmentation network U-net, added a new CBAM module, and fine-tuned the parameters and size of each network layer to form a modified version of the new network: Atten_Unet.

The processing and training procedures for 3D MRI data are very different from those of 2D images, due to the special dimensionality feature of 3D images. Data that is too large may result in a shortage of computing resources during training. Therefore, before putting the data into the neural network for training, I also perform several special selection, padding and interception strategy for the 3D MRI data patches. After that, by using the newly developed Atten_Unet network, I was able to train and validate the selected 3D MRI brain tumor dataset, resulting in excellent tumor segmentation results. Experiments to compare the results with those of other networks and other methods are also designed and operated, and the differences between those methods are carefully analyzed and recorded.

Various useful metrics have also been applied to my project to improve the results of tumor segmentation. A few examples include adjusting the use of optimizer, scheduler and loss functions, permutation and selection of strategies for data augmentation and dataset training, etc. These small strategies allow for maximizing the effectiveness of the segmentation algorithm, based on applying the smallest possible dataset and training fewer epochs.

The 3D MRI data in this research project is medical data, which has a unique format that is not the same as ordinary image formats, such as jpg or png. The data was read and transformed using the SimpleITK software package so that training and processing could be done efficiently.

For the presentation of the final segmentation results, I used ITK-Snap, a specialized medical image analysis tool.

During the training process, all the records regarding the training were monitored and recorded in Tensoroard. For each training session, I systematically recorded the loss, accuracy, and DSC scores, as well as the segmentation results and the training results for each part of the tumor. Finally, the average and maximum segmentation performance results are calculated for each data in the whole training procedure.

For validating the effectiveness of the brain tumor segmentation system, the DSC score and Hausdorff distance were used. Having trained only half of the original dataset with only 150 epochs, excellent segmentation results have been achieved on the validation set (ET:85.4%, TC:90.1%, WT:88.6%), which is almost close to the state-of-the-art in the field of automatic brain tumor segmentation and proves to be an exceptional result. Now, with the trained model weights in my project, it is possible to segment a new 3D MRI brain tumor data anytime and anywhere, and it only takes a few seconds to achieve a very good segmentation result, which is excellent for the task of automatic brain tumor segmentation. In the future, it can also be used in the medical community for automatic tumor segmentation to provide valuable preoperative advice to doctors.

Chapter 2: Background

2.1 "Brain Tumor Segmentation Challenge 2020 (BraTS 2020)"

For this project, the dataset came from "The Brain Tumor Segmentation Challenge 2020" (BraTS). BraTS examines different methods from various teams of segmenting brain tumors in multimodal MRI scans. The challenge makes use of pre-collected MRI scans from different hospitals and institutions. There are three primary tasks for this competition, Task 1, Task 2, and Task 3. Task 1 mainly focuses on the segmentation of heterogeneous brain tumors in different kinds of shape, appearance, and many other aspects. Task 2 is relevant to the prediction of the patients' survival probability. For Task 3, it mainly concentrates on how to evaluate the uncertainty of tumor segmentation. For this project, I only participated in Task 1's tumor segmentation task in a targeted manner. The remaining two tasks were not covered.

2.2 Details about the Dataset

BraTS 2020 consists of 369 training cases and 125 validation cases (Isensee et al., 2020a). Due to the limitations of funding and equipment, I could only train the data using Google Colab for this project. In the end, a total of 200 cases out of 369 training cases were selected for training and 5-fold cross-validation, while 30 cases out of 125 testing cases were selected for testing and presentation of the final tumor segmentation results.

Accordingly, we have the following overview of the dataset: the training set consists of 200 patient files. Three-dimensional MRI volumes have the shape of 240(height) * 240(width) * 155(number of slices) * 4(multimodality). If we look at patient 001 as an example, the 5 files in the folder are: PatientNo_ti.nii, PatientNo_t2.nii, PatientNo_t1ce.nii, PatientNo_flair.nii and PatientNo_seg.nii.

These include T1c (T1 with contrast agent), T2 and FLAIR (Fluid Attenuation Inversion Recover)(Myronenko and Hatamizadeh, 2020). The four different modality files can be interpreted as four different channels of the data during the training process, and the last file with a name of seg.nii is the ground truth label for three tumor regions for this particular patient. As for the 30 patient folders in the testing dataset, what is the same as the training dataset is that it also has four preceding nii files, namely t1, t1c, t2, and FLAIR, but it does not have the seg.nii file because the ground truth label is not necessary for testing at competition.

Figure 1 illustrates the structure of the entire dataset that was used for this project.

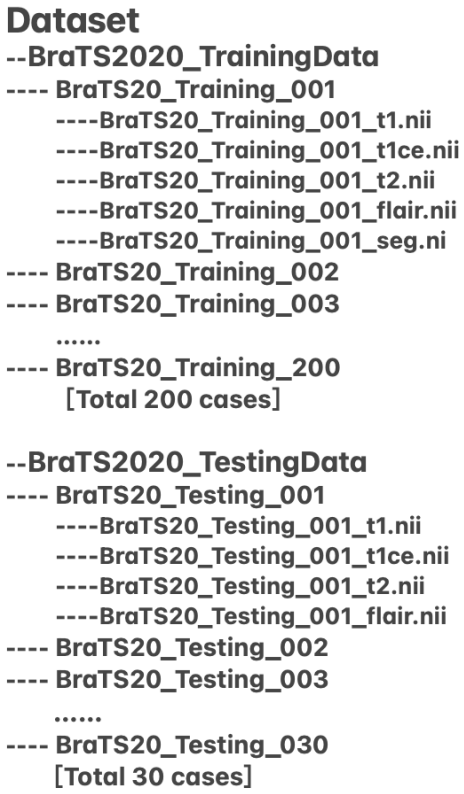
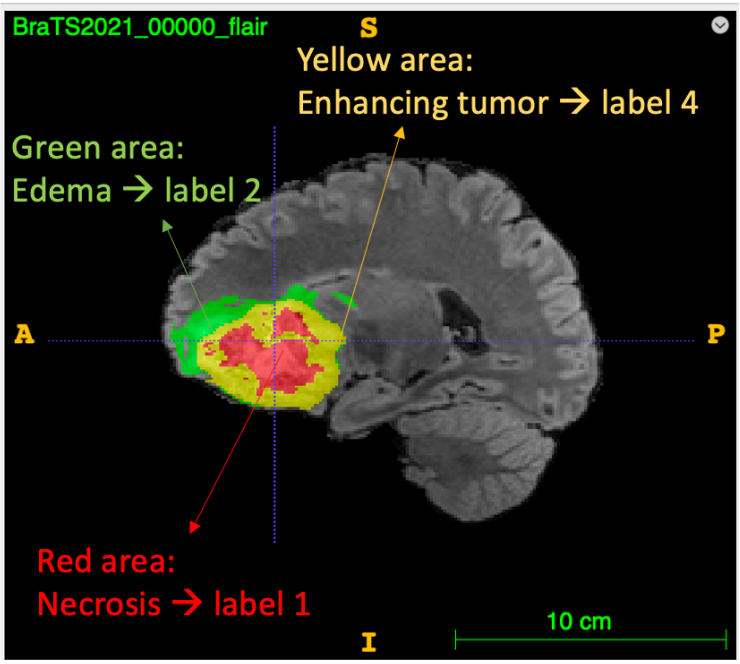


Figure 1 The structure of the entire dataset

2.3 Tumor regions and ground truth labels

From inside to outside, the three regions of the tumor and their corresponding labels are: (1) Necrosis --> label 1, (2) Enhancing tumor (ET) --> label 4, and (3) Edema (ED) --> label 2. Except for the tumor regions, the remaining parts of the background are filled with 0. Based on experience with several brain tumor segmentations, the semantic segmentation of the tumor region is divided into three categories: ET (enhancement tumor), TC (tumor core), and WT (whole tumor).(Myronenko and Hatamizadeh, 2020). Among them, ET represents the Enhancing tumor region with the label 4. TC represents the sum of necrosis and ET, which means, label 4 plus label 1, whereas WT represents the sum of the three tumour regions, that is, label 4 plus label 1 plus label 2. It should be noted that the official label of the data set does not include label 3. In the subsequent experiments, training and validation are heavily reliant on the division of these classes. The authors of many papers in this area: "BraTS evaluates segmentation using the partially overlapping ET, TC and WT regions, optimizing these regions instead of the three provided class labels (edema, necrosis and enhancing tumor) can be beneficial for performance" (Isensee et al., 2020).

Figure 2 visualizes the three tumor parts and their corresponding labels, as well as the three targeted tumor segmentation regions:



Segmentation target (Prediction):

- ✓ ET (Enhancing tumor) → label 4
- ✓ TC (Tumor core) → label 4 + label 1
- ✓ WT (Whole Tumor) → label 4 + label 1 + label 2

Figure 2 Tumor parts, labels and segmentation target

2.4 Related work

Based on CNN architecture, many successful or state-of-the-art results are obtained in semantic segmentation, especially in 2D and 3D medical image segmentation tasks. In the field of tumor segmentation, computer scientists have devised different variants of U-Net-based networks, either based on well-cut 2D slices or 3D volumetric data. Approaches that directly apply the 3D-based method and process the volumetric image tend to produce better segmentation results than 2D-based methods, since it exploits the spatial relationships among voxels and could take into account the global context. On this basis, a number of attempts have been made. Using a modified U-Net baseline in combination with region-based training and aggressive data augmentation, Fabian Isensee won the BraTS 2020 championship with the design of nnU-Net(Isensee et al., 2020a). Santa Clara added a variational auto-encoder to the basic Unet architecture and achieved excellent performance(Myronenko, 2019). Using a U-Net backbone

in conjunction with atrous convolutions, residual connections, and pyramid scene parsing pooling(Myronenko, 2019), Foivos I. Diakogiannis proposed ResUNet-a.

Despite the effectiveness of these approaches, pure CNN-based U-net structures can have shortcomings in capturing long-range information and global context, limiting their effective use for brain tumour segmentation. Transformers have been a hot topic for the past few years. Originally, they were proposed for NLP tasks. The network has performed well in many basic computer vision tasks as well as their downstream segmentation and object detection tasks. Transformer's attention mechanism reduces information loss by pinpointing the most important points from a vast amount of data, focusing on the most important information, and minimizing distracting data. Many Vision Transformer-based methods have also been proposed for brain tumor segmentation tasks. Ali Hatamizadeh, for example, proposed UNet Transformers (UNETR), which use Transformer to encode the representations of volumetric data sequences(Hatamizadeh et al., 2021).

When it comes to Loss function, Dice loss(Milletari, Navab and Ahmadi, 2016) or its variants are commonly used in brain tumor segmentation algorithms. There has been some experimentation with combining Dice loss with other losses such as cross-entropy loss, but none of them are especially different. In terms of optimizers, the most commonly used ones include Adam, Adamw, RMSCrop, and AdaGrad.

In this project, the U-Net-based and encoder-decoder-based architecture is used to achieve good results. The attention mechanism in Transformer is also implemented in this project by adding a CBAM (Convolutional Block Attention module) (Woo et al., 2018) to every encoder and decoder's block to combine U-Net and attention mechanism.

Different optimization techniques were tried in this experiment, and finally, the Adam optimizer was selected. As compared to other methods, my method produced excellent segmentation results with a significant improvement.

2.5 Related technology

2.5.1 Simple-ITK package

Simple-ITK is an Open-Source toolkit developed for multidimensional image analysis, particularly for images that are not in the form of arrays of pixels. Support for a variety of image file formats, such as DICOM, NII, TIFF, etc, makes it useful for medical data processing and analysis. It also provides convenient format conversion tools. As part of this project, I use

Simple-ITK to convert medical MRI data in the NII format into an array that can be processed by Python.

2.5.2 ITK-Snap Software

ITK-Snap is a professional medical image visualization software. 3D MRI data can be displayed dynamically in 3 different axes, X, Y, and Z in ITK-Snap. By dragging the view on one of the axes with the mouse, the images of the other two views will also be changed accordingly. The segmentation results can be imported and viewed by simply dragging the seg.nii file into the existing 3D MRI view.

Additionally, manual annotation can be performed with the software. The annotation can be done by using the annotation tool in the Main Toolbar and proceeding along each of the three axes. After layer-by-layer annotation is complete, we can click 'update' in the lower-left window to update the views and observe the annotated 3D image.

With this software, I am able to visualize and compare the segmentation results for this project. Axial, sagittal, and coronal views of the brain tumor are shown in Figure 3.

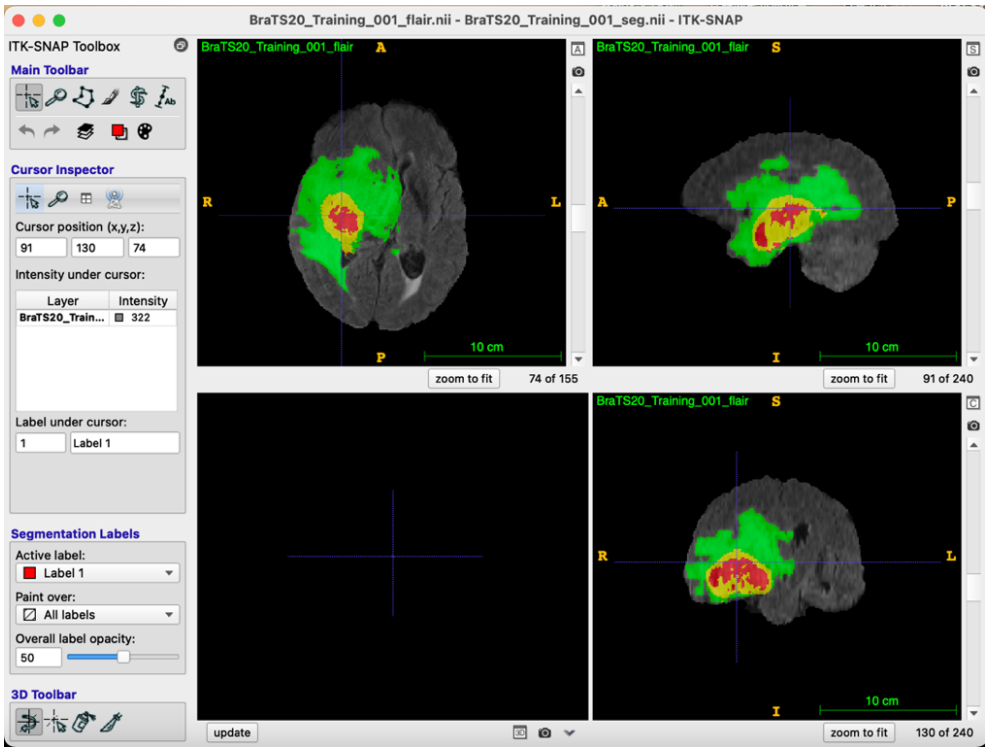


Figure 3 Axial, sagittal, and coronal views of the brain tumor in ITK-Snap

2.5.3 3D U-Net

3D-Unet based on dimensional transformations and is adapted from U-Net, which applied specifically to 3D data. This method has achieved impressive results in many computer vision tasks and is widely used in semantic segmentation of 3D data. It consists of both an encoder path and a decoder path. There are four resolution levels in both the encoder path and the decoder path(Çiçek et al., 2016). There are two convolutions in each block of the left descending encoder path. In the immediate following layer, there will be a ReLU layer for activation followed by a maximum pooling layer with a step of 2.

In terms of the decoding path, each block contains 2 deconvolution layers with a step size of 2. Following this, there is a 3*3*3 convolutional layer, followed by a RuLU layer. Using the shortcut, the left encoding block is directly connected to the right size decoding block, and a 1*1*1 convolutional layer is added at the end to reduce the number of output channels. Upon completion of the network, the final number of output channels corresponds to the number of class divisions of the target. A deformed version of 3D-Unet was used for this project.

Figure 4 illustrates the schematic representation of 3D-Unet in the original paper:

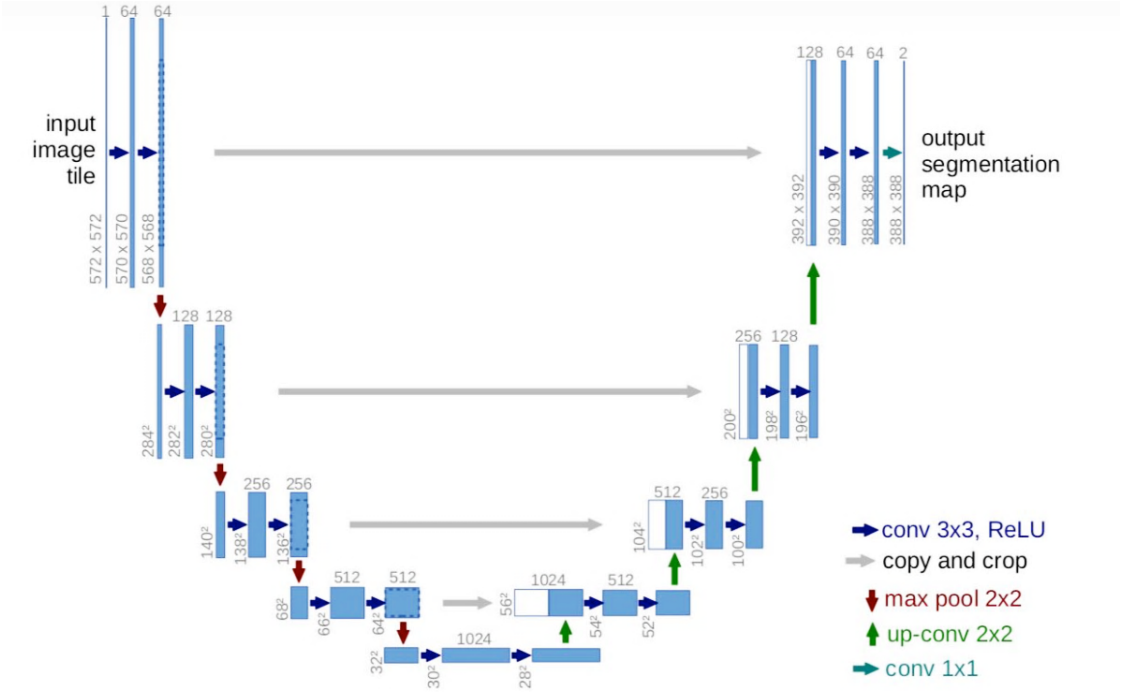


Figure 4 Pure 3D-Unet from the original paper

2.5.4 Attention mechanisms and CBAM module

The attention mechanism has been extensively used in various machine learning tasks, including computer vision, NLP, and speech recognition, in recent years due to the popularity of Transformer(Vaswani et al., 2017).

Using the analogy of how humans observe images, we can better understand the importance of the attention mechanism. A person looking at a picture will concentrate more on the central information rather than on every pixel equally. Our attention mechanism works in a similar way. By learning from input data, it can focus its attention automatically on the panels that require specialized attention. By employing the attention mechanism, the neural network can create a mask, then score the values on the mask. In this way, voxels that require special attention are given higher scores, thereby boosting the importance of useful features and suppressing the importance of less-important features.

There are three broad types of attention mechanisms: channel attention mechanisms, spatial attention mechanisms, and hybrid attention mechanisms(Woo et al., 2018). The channel attention mechanism generates and scores masks for channels, while the spatial attention mechanism generates and scores masks for spaces.

In Figure 5 below, we see how the channel attention mechanism works:

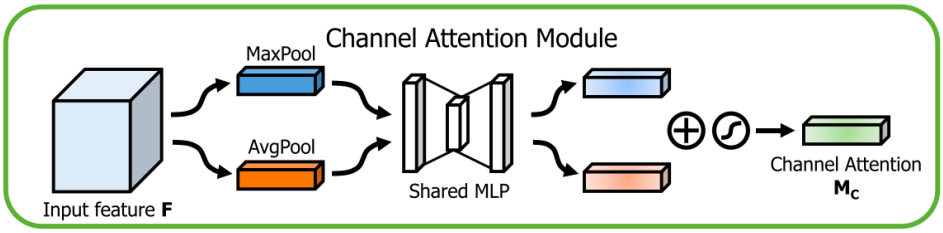


Figure 5 Channel Attention Module

In terms of channel attention module, the middle MLP layer is shared using a 1*1 convolution for information extraction.

The spatial attention mechanism is implemented according to the following Figure 6.

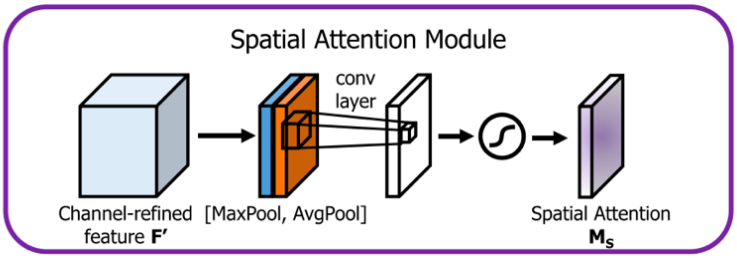


Figure 6 Spatial Attention Module

As shown in Figure 6, for each channel of the data, AvgPool and MaxPool are applied separately. Then results obtained from AvgPool and MaxPool are then merged to form a concatenated convolutional layer with a channel number of 2. Following that, another additional convolution computation is performed to obtain the final spatial attention with a channel number of 1.

CBAM stands for "Convolutional Block Attention Module". In this project, we use the CBAM module as a hybrid attention mechanism, i.e., a fusion of the channel attention mechanism and the spatial attention mechanism.

Finally, CBAM combines the above two modules: Channel Attention Module and Spatial Attention Module. Through a broadcasting mechanism, the integrated attention block is merged with the previous network output to produce the final feature map which has been processed by the attention mechanism. Figure 7 below summarizes the overall structure of the CBAM module:

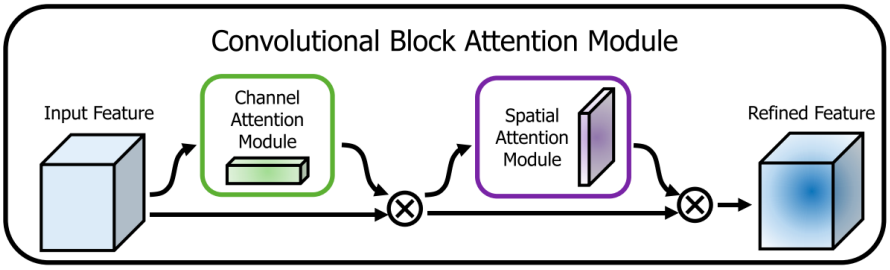


Figure 7 CBAM

CBAM can easily be integrated with other networks, such as 3D-Unet. This project combines the CBAM module with each of the encoder and decoder blocks in the 3D U-Net to create new algorithms.

Chapter 3: Design and Implementation

This project is written in Python and is implemented using the PyTorch framework. It was first developed in PyCharm, and then uploaded to GitHub. I also use Google Colab to complete the task since GPUs are needed to accelerate training. The dataset was uploaded to Google Drive prior to training. When training is needed, the code and the dependency packages needed for the project are downloaded from GitHub to a Jupyter Notebook editor on Google Colab. The training, validation, and testing is done using the dataset in Google Drive after the dependent environment has been configured on the remote server.

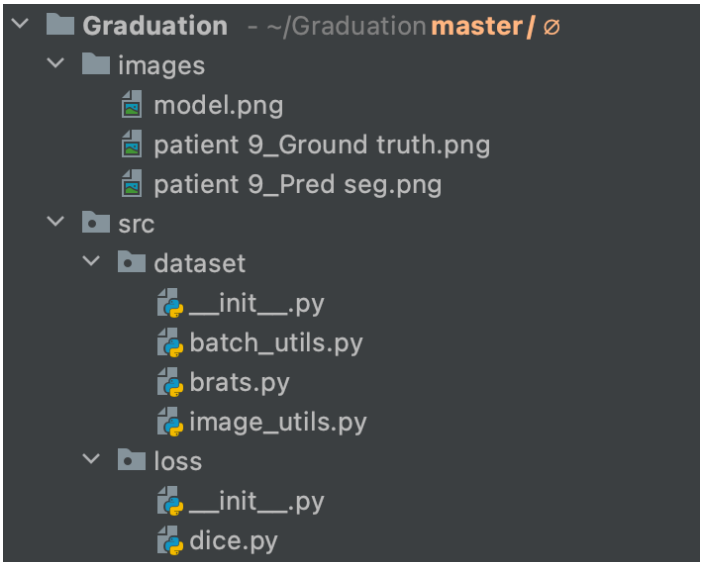
The packages required for the project are shown below in Table 1**.**

Table 1 required packages

Torch	Numpy	Pandas	SimpleITK	PyYAML	Matplotlib	Scikit-learn	Scipy
1.6.0	1.18.4	1.0.3	1.2.4	5.3.1	3.2.1	0.23.1	1.4.1

There are nine main modules of code for this project. The main modules in the code are parameter configuration, dataset pre-processing, 3D data chunking and padding, data augmentation block, loss functions, evaluation metrics, model, training, testing, and other utils functions needed for the project. In addition, the code files have been divided into sections according to their contents, and the corresponding python files are: config.py, brats.py, image_utils.py, augmentation_blocks.py, dice.py, unet.py, train.py, test.py, utils.py.

Here is a screenshot of the main code modules in PyCharm, as shown in Figure 8**.**



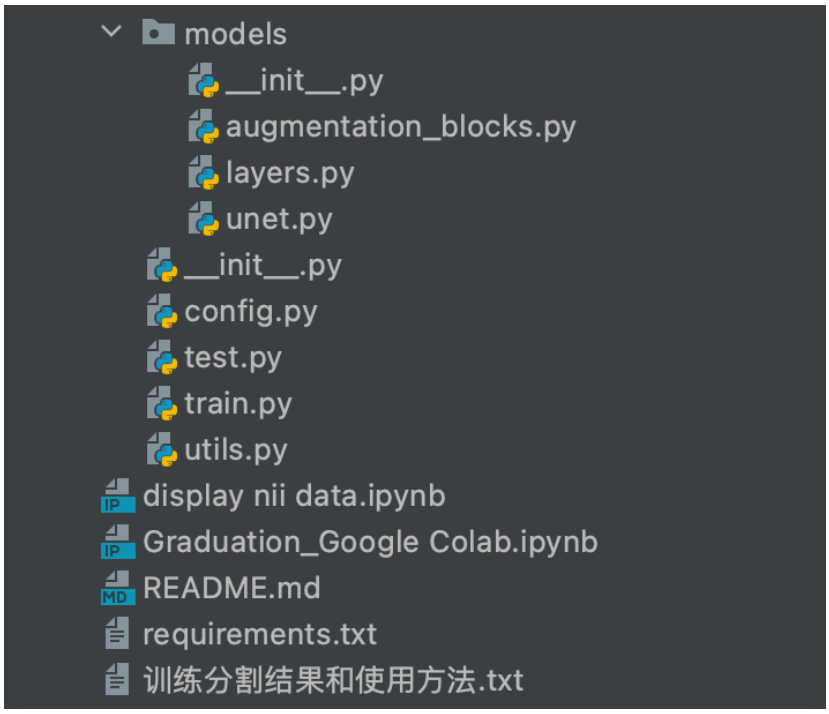


Figure 8 Structure of the code

There are also two ipynb files that can be run on jupyter notebook or the notebook editor on Google Colab. The first one is Graduation_Google Colab.ipynb, which can be used to run the train.py and test.py easily using a GPU environment. There are also commands that can be used to visualize the graphs about the training process and the change of scores on Tensorborad. The second one is display nii data.ipynb, which can be used to visualize 3D medical data and segmentation result.

3.1 Data pre-processing

3.1.1 Normalization

As MRI tumor data are gathered from patients in different institutions, hospitals, and the standards of the acquisition instruments may vary from institution to institution, the distribution of MRI tumor data in the dataset may vary widely. Models have to identify patterns and trends from data voxels during training, which can be difficult and error-prone when the sizes and scales of 3D MRI volumes vary widely. For this reason, we need to normalize the data volumes in the dataset. The purpose of normalization is to make the proportions of voxels in each volumetric data similar, so that each input feature has equal importance.

The most widely used normalization methods are min-max and z-score.

The "min-max normalization" process converts all data voxels into a decimal number between 0 and 1. By subtracting the minimum value from the previous data distribution and dividing it by the difference between the maximum and minimum value, the new values are obtained.

The formula for min-max normalization can be expressed as follows:

$$x_{new} = \frac{(x_{old} - X_{min})}{(X_{max} - X_{min})} \quad (1)$$

For this project, all data were normalized using the min-max method. Its disadvantage, however, is that it cannot handle outliers very well. It will have a significant negative impact on normalization results if the distribution of one of the values differs significantly from the rest of the data.

Thus, prior to doing min-max scaling, I also eliminated the outliers. This is a very important strategy. From each 3D data volume, I extracted the data voxels whose distributions ranged between 0% and 1% and 99% and 100%, and then I ruled out the voxels distributed between 0% and 1% and 99% and 100%.

Another way to avoid this outlier problem is to normalize using the Z-score method.

The formula of Z-score normalization is as follows:

$$x_{new} = \frac{(x - \mu)}{\sigma} \quad (2)$$

μ indicates the mean value, while standard deviation is denoted by σ.

According to comparison tests, minmax normalization produced better results. Therefore, minmax normalization is applied to the ultimate data pre-processing. After data pre-processing, all data voxels are transformed into a new value between 0 and 1 and the outliers were also removed.

3.1.2 Data cropping and padding

As MRI data is three-dimensional data, it has many voxels. Therefore, putting the whole volumetric data into the neural network for training will require a lot of computational power and memory, which is hard to realize due to the deficiency of GPU and advanced devices. Therefore, I implemented the padding strategy to ensure smooth execution of the project.

After normalization, I remove as many unnecessary background parts as possible from each volume of data, that is, regions that contain no tumor areas and are labeled as 0. To achieve this,

I use a *"Minimal Bounding Box Strategy".* As a first step, I calculate the value of the most edge of each data volume in the three dimensions x, y, and z. In the next step, I expand each edge value by one in each x, y, and z dimension, then crop the size of the volumes according to the expanded edge value. In this way, we can well separate the minimum patch containing all of the tumor regions, so that we could maximize the use of all the effective data, while at the same time ensuring that all the data containing labels 1,2, and 4 is not wasted.

I then used the random number method to cut this patch into a size of 128*128*128 based on the minimum tumor patch. However, there are some data volumes with tumor regions smaller than 128*128*128 due to uneven data sources. When we use the strategy "Minimal bounding box" to cut off the minimal area that contains the tumor, it is very likely that the patch is not big enough for size 128*128*128 and not enough for putting into the network for training.

To prevent bugs during training, I also used the padding strategy to fill the missing regions with zeros to ensure every training data has a size of 128*128*128. Data pre-processing is finished after the above series of operations, and the 128*128*128 patches are then input into the network for training and validation.

3.2 Data augmentation block

Considering the limitations of computing devices, I only selected a portion of the original dataset for this project. The data was augmented using a variety of strategies to increase its diversity and size. This fully improves the diversity and complexity of the 3D MRI tumor data. By using data augmentation strategies, the network is able to see more examples during training, which facilitates more accurate parameters and weights. Overfitting is another catastrophic phenomenon that can occur without the data augmentation strategy. As a result of being overfitted to the characteristics of the data in the training set, the neural network performs poorly when validated or tested. This is reflected by the large loss in validation and testing and the large accuracy gap between the training set and the validation set.

Therefore, in this project, I adopted various data augmentation strategies: which included rescaling, adding noise, channel dropping, transposing, and flipping.

The probability of rescaling is 0.8. For each voxel, it is multiplied by a number between 0.9 and 1.1. Adding noise has a probability of 0.8. Gaussian noise with a standard deviation of 0.1 is added separately to each channel. Channel dropping has a probability of 0.2.

If inverse channel dropping occurs, all voxels in a particular channel will be set to 0, which simply means that one of the four modes of the volume data will be erased. Although some TP values are sacrificed, this operation is designed to suppress overfitting. The final segmentation result also proves to be better with channel dropping.

The following Table 2 shows the adopted data augmentation strategy and the corresponding probabilities.

Table 2 Data augmentation in this project

Strategy	Rescaling	Noise	Transposing	Flipping	Channel dropping
Probability	0.8	0.8	0.8	0.8	0.2

3.3 Loss function and loss computation

The loss function used by most of the teams that have won BraTS competitions in the past is Dice loss, which comes from the V-Net paper(Milletari, Navab and Ahmadi, 2016) and proves to be a good approach in semantic segmentation. Inspired by them, Dice Loss is used for training in this project. Dice loss can be used to measure the difference between the predicted values and ground truth values in a segmentation task. The dice loss varies from 0 to 1, with a larger value representing a greater difference. Losses are calculated separately for each of the four different data modalities by channel, and finally, an average is taken from the four channels of different four modalities. There is no weight added to the four channels, it is simply pure averaged.

Each voxel's dice loss is represented by the equation in the following figure:

$$L_{dice} = 1 - \frac{2 * \sum p_{true} * p_{pred} + \varepsilon}{\sum p_{true}^2 + \sum p_{pred}^2 + \varepsilon} \quad (3)$$

p_{pred} is the predicted probability value of the neural network activated by the sigmoid function. p_{true} is the true label value of the voxel. The smoothing factor ε is used to prevent zero being the denominator, which results in catastrophic training errors.

In our experiments, the smoothing factor e is set to 1. By calculating the average loss of all voxels in each channel as Lc, and then averaging them together, we obtain the total loss for each data volume as Lt.

In the training, we optimize Lt. Note that in this case, the final neural network output is a volume with three channels, and each channel represents the probability map of the ET, TC, and WT tumor regions. The content of our optimization consists of three separate probability maps of three separate tumor regions ET, TC and WT.

These three probability maps then make up the entire output volume of the network. In order to optimize the loss, the feature maps of the three tumor regions must only contain zeros or ones rather than the ground truth labels 1, 2, and 4 provided in the seg.nii file. That's why we use sigmoid as activation function.

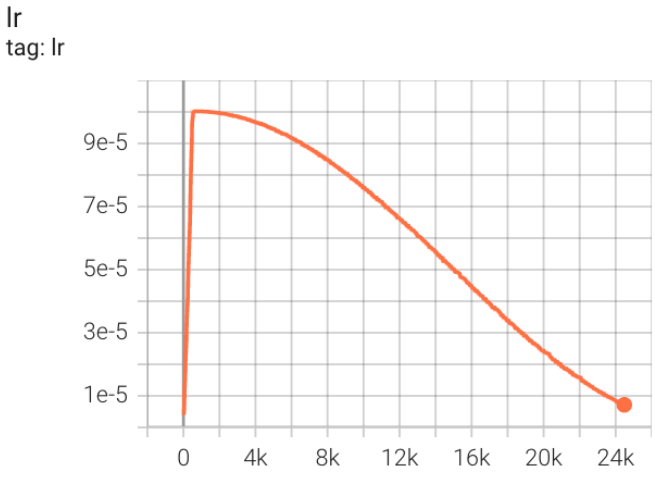
3.4 Optimizer and learning rate

In the training process, I used the Optimizer Adam, with no weight decay. Adam stands for Adaptive momentum estimation.

The initial learning rate was set to 1e-4.

The optimizer is a core component of deep learning. If we use an SGD optimizer, we have to manually select the learning rate and momentum parameters to reduce the learning rate over time. The Adam algorithm is actually an adaptive optimization algorithm that is basically an RMSprop optimizer with momentum so that it could dynamically adjust learning rates based on first and second-order moments of gradients.

Finally, Adam Optimizer was used in this project. According to the comparison study, the use of Adam leads to improved performance of the optimization in this project. According to the Tensorboard graph, the learning rate in the whole training procedure raises sharply at first, and then gradually decreases in the following procedure as shown in the figure below.



3.5 Model: Atten_Unet

Atten_Unet, which is creatively proposed by me in this project, is based on the 3D-UNet and CBAM modules (already described in detail in the background in part 2.5.4). It excellently combines the U-Net infrastructure with the attention mechanism. Its basic structure contains an encoder and a decoder structure.

The basic structure of Atten_Unet used in the project is shown in Figure 9.

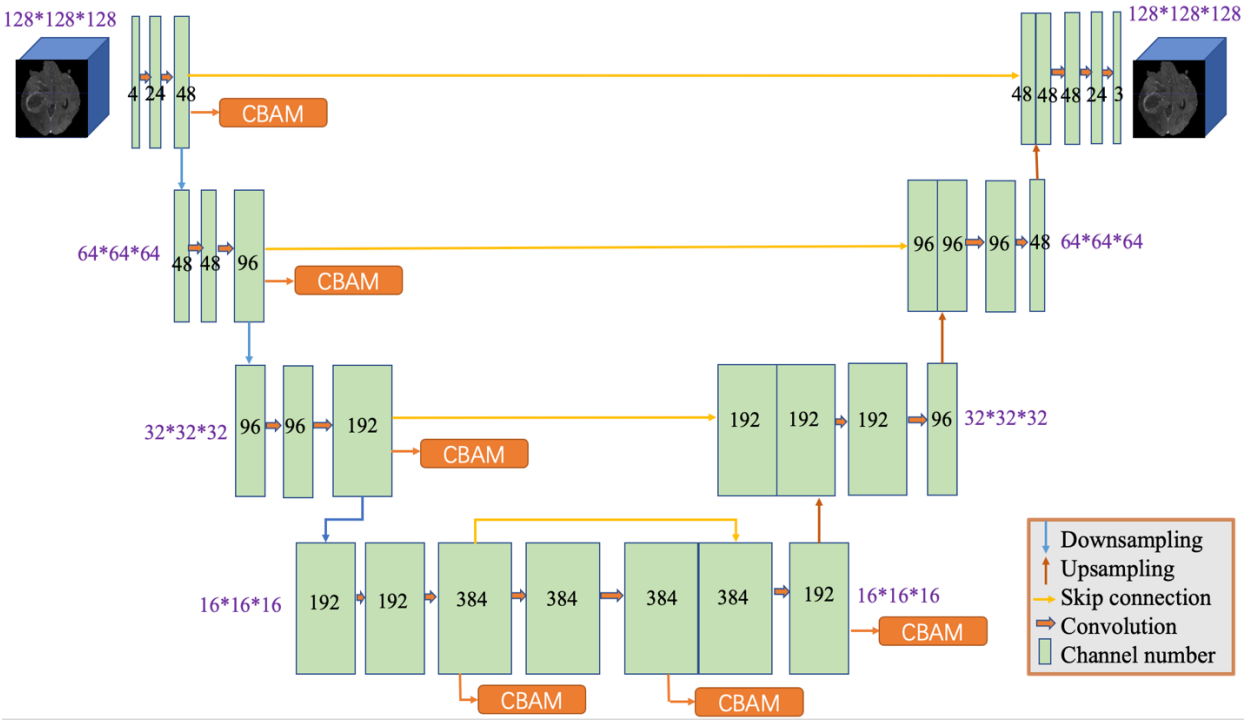


Figure 9 The structure of Atten_Unet

As shown in the Figure 9, the encoder structure on the left consists of four blocks, and the decoder on the right also consists of four blocks.

Each block consists of two 3x3x3 convolutions. Because the batch size used during training is 1, which is very small, I use Group normalization (Wu and He, 2018) instead of the usual Batch normalization. Group normalization method avoids the influence of batch size on the model, and group normalization of features can also solve the problem of "Internal Covariate Shift" and achieve better results.

In each block in encoder, it has two convolutions. Immediately following each convolution is a group normalization layer, a ReLU layer. At the end of each block, a CBAM block is added to realize attention mechanism. In this way, features with more importance can be better extracted.

For each CBAM block, each CBAM module contains a SpatialGate and a ChannelGate, which are used for spatial attention and channel attention implementations, respectively.

The initial input size of the input network is (1,4,128,128,128), where 4 represents the feature map of 4 modalities, namely t1, t1ce, t2 and flair, which can be regarded as the channel in the convolution. In the encoder, every block contains two successive convolutions, each expanding the channel size to twice the previous size.

In the encoder on the left, spatial downsampling is performed between each block. A MaxPool layer is used to perform "spatial downsampling" with a kernel size of 2x2x2 and a step size of 2. After each downsampling, the x, y, and z of the 3D volume data in the input network are expanded by a factor of 2, respectively.

The structure of the encoder and decoder is almost symmetric. However, what is worth mentioning is that: for every decoder block on the right, it doesn't contain a CBAM module at the end of the block. In the decoder, a spatial upsampling operation is also performed between each block. Afterward, the x, y, and z dimension of the 3D volume data in the network are each compressed to half of their previous size.

There are two bottom blocks between the encoder and the decoder. The first one performs two convolutions without changing the channel size. The second bottom block first concatenates two same-sized output from the last encoder and the first bottom block to increase the total channel number to twice the previous size, and then reduces the channel size to the half. As a result, the two blocks at the bottom connecting the encoder and the decoder together do not affect the output channel size.

The horizontally symmetric encoder block and the decoder block have the same space size, as indicated in the figure. The output of every left encoder is directly connected through a skipped connection to the right decoder of the same size. In addition, for every decoder block, the decoder on the right of the figure is convolved twice, so that the number of channels for the output is reduced by half after every decoder block.

In the bottleneck part of the decoder, the last convolutional layer uses a 1x1x1 kernel. The final output size is (1, 3, 128, 128, 128), where the channel size is equal to 3, corresponding to the prediction maps of ET, TC, and WT accordingly, namely, the three tumor regions.

The prediction maps are generated after the network has been trained. Each voxel in the prediction map contains a probability value predicted by the neural network. It is followed by

a sigmoid layer that activates the probability values and then scales the prediction values to between 0 and 1. By using the evaluation metric, we can measure the algorithm's effectiveness by comparing predicted values with ground truth labels.

It is important to note that we use sigmoid instead of softmax for the activation function because we are trying to solve a dichotomous problem, i.e., to classify the labels as 0 or 1, rather than a multiclassification problem. Table 3 illustrates the structure of each block of the model and the size of the output for each layer.

Table 3 The structure and output of Atten_Unet

Name	Structure	Output Size
Input		4×128×128×128
Encoder 1	Conv×2, GroupNorm×2, Relu×2, CBAM	48×128×128×128
EncoderDown 1	DownSample	48×64×64×64
Encoder 2	Conv×2, GroupNorm×2, Relu×2, CBAM	96×64×64×64
EncoderDown 2	DownSample	96×32×32×32
Encoder 3	Conv×2, GroupNorm×2, Relu×2, CBAM	192×32×32×32
EncoderDown 3	DownSample	192×16×16×16
Encoder 4	Conv×2, GroupNorm×2, Relu×2, CBAM	384×16×16×16
Bottom 1	Conv×2, GroupNorm×2, Relu×2, CBAM	384×16×16×16
Bottom 2	Conv, GroupNorm, Relu, CBAM	384×16×16×16
Bottom2 UP	UpSample	384×32×32×32
Decoder 3	Conv×2, GroupNorm×2, Relu×2, CBAM	192×32×32×32
Decoder3 Up	UpSample	192×64×64×64
Decoder 2	Conv×2, GroupNorm×2, Relu×2, CBAM	96×64×64×64
Decoder 2 Up	UpSample	96×128×128×128
Decoder 1	Conv×2, GroupNorm×2, Relu×2, CBAM	48×128×128×128

Output		3×128×128×128

3.6 Evaluation metrics for validation

To evaluate the performance of the tumor automatic segmentation system I designed, some evaluation metrics must be used. Unlike ordinary 2D images, we cannot simply measure segmentation quality by comparing loss and accuracy.

For this project, the 'Dice Similarity Coefficient' is used as a measure of how closely the predicted segmentation map matches the real segmentation map.

To measure the severity of the outlier in the predicted image map, we also need to use the 'Hausdorff Distance' as another evaluation metric.

3.6.1 Dice Similarity Coefficient

"The commonly used Dice Similarity Coefficient (DSC) measures the overlap between two sets"(Cao *et al.*, 2021). Therefore, I use DSC score as one of the evaluation metrics for performance in the project. It can show the degree of overlap between the predicted tumor part and the real tumor part.

The formula of DSC can be expressed in the following form.

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{4}$$

TP stands for the true positives, that is, the number of voxels that are correctly classified. FP, namely false positives, refer to the value which is predicted to be 1 but what is actually assigned is 0. False negatives, or FNs, are results where the predicted value is 0 but the real value is 1. DSC has the range of 0-1. The larger the DSC, the better the segmentation effect.

In addition, there is a special setting in the BraTS competition, where the reference segmentation map is set all to 0, which is to say that the tumor does not exist in this part of the map.

It is considered a huge error to have even one voxel with a value of 1 existing in these kinds of empty segmentation maps. So, in this case, the DSC score must be set to 0. If all the voxel values are predicted as 0 in these situations, i.e., correspond perfectly to the empty map, the DSC score will be awarded with a value of 1. This also an importance setting in my system.

3.6.2 Hausdorff distance

The Hausdorff distance is a complement to the DSC and can be used to measure the maximum distance between two contour edges, thereby presenting the severity of outliers in the segmentation results. To intuitively explain why it is used, we can imagine how severe the consequences would be if doctors removed normal brain cells that is predicted by the computer as tumors. In other words, even if the predicted qualitative results match the true values perfectly, but there is a predicted voxel far from the true segmentation map, the Hausdorff distance will still be high.

In order to prevent excessive outliers from having catastrophic consequences for the patient, it is necessary to assess the Hausdorff distance between segmented brain tumors and the real tumor areas. Therefore, for this project, the use of Hausdorff Distance to measure the outlier is also a highlight.

Here is the detailed explanation of 'Hausdorff Distance'. When there are two sets of points A and B, the Hausdorff distance is calculated as follows: the shortest distance from every point ai in set A to every point bi in set B is calculated first, for each ai, the shortest distance is recorded as di. Among all these dis, the maximum di value is the Hausdorff Distance between set A and set B.

The Hausdorff Distance is expressed in the formula as shown in the following figure：

$$h(A,B) = \max_{a\in A}\left\{\min_{b\in B} d(a,b)\right\} \tag{5}$$

3.6.3 Sensitivity and specificity

To measure the difference between predicted and true values, another two metrics can be used: Specificity and Sensitivity.

Sensitivity is the percentage of true 1's in the part of the data where the predicted value is 1. Specificity is the percentage of true 0's in the part of the data where the predicted value is 0.

In this project, We can understand it in this way: Sensitivity represents the proportion of regions predicted to be tumors that are really tumors. Specificity represents the proportion of regions predicted to be background (non-tumors) that are really background.

Their formular are shown in the following figure:

$$Sensitivity = \frac{TP}{TP + FN} \tag{6}$$

$$Specificity = \frac{TN}{TN + FP} \tag{7}$$

3.7 Training and Validation

Training and validation were conducted on Google Colab Pro using a Nvidia Telsa V100 graphics card. The dataset used for training and validation was a partial MICCAI BraTS 2020 Training dataset containing 200 patients' 3D MRI data.

Before starting the formal training, 3 epochs were used for the warm-up procedure, where the scheduler used was LambdaLR. As the weights of the model are randomly initialized at the beginning of the training, if a large learning rate is chosen, the model may become very unstable, The warm-up procedure can help solve the problem.

There are 150 epochs for the entire training process. The dataset is read and preprocessed before formal training begins, as described in part 3.1 and 3.2. The training batch size is 1, i.e., each batch contains only one 3D volume of data. Instead of data augmentation of all data in the whole dataset in advance, data augmentation is performed in each batch individually, with different augmentation metrics and with different pre-defined probabilities. It is a rather bizarre data augmentation strategy but works well for data augmentation.

Then 5-fold validation is used for training and validation. That is to say, in each fold, the training dataset is randomly divided into 5 parts, 4 of which are used for training, and the remainder is used for validation. Then, the above steps are repeated 5 times and the results are finally averaged. Note that each fold contains 160 data volumes for training and 40 data volumes for validation.

The initial learning rate is 1e-4, and the optimizer used in the training process is Adam.

The training process and validation results are recorded in detail in the Tensorboard, and the dynamic results can be viewed at any time by accessing the events file in Google Colab.

I also add an argument parser module in the beginning of the training for it can help with changing different parameters for training. Some parameters that can be chosen include model type, epoch number, initial learning rate, optimizer, drop out and so on. This is a very convenient way for conducting ablation study.

The loss in the training process constantly drops for the whole time.

The changes in summary loss during the training and validation processes is displayed in Figure 10.

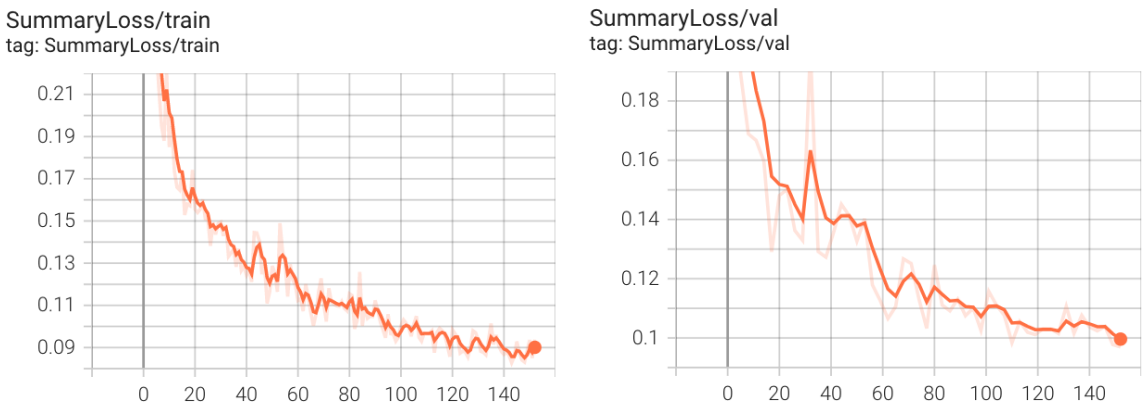


Figure 10 Summary loss for training and validation

From the graph, we can see that the loss is decreasing continuously.

Training loss at 150 epochs is 0.09, and validation loss is 0.1, which is not much of a difference, just a little overfitting existing. This suggest that it is a very effective algorithm that the model's performance is continually improving.

Based on these observation, conjecture could be made that I would have achieved greater training and validation accuracy and performance if I had better computing equipment and had trained more epochs.

3.8 Testing

Additional testing is performed in the Testing dataset, which contains 30 data volumes without ground truth labels. Using the weight file "model_best.pth.tar" generated by the best validation results from the previous 150 epochs of model training, the program for testing directly loads the parameters and weights into the neural network to generate the predictions for the 3D MRI data in the testing dataset. The parameter configured before in the best training process is also reloaded in the testing procedure.

As soon as the prediction is completed, a folder named seg is created which contains the segmentation results for the 30 data volumes predicted by the model, presented as a file named seg.nii. When I dragged it into ITK-Snap, I could see the automatically segmented brain tumor regions presented in different colors. With this method, we can predict all unlabeled 3D MRI brain data gathered from any source in the future. This is how the automated segmentation system works on future brain MRI data without annotations.

Below in Figure 11 is an example of the result of the automated segmentation in testing.

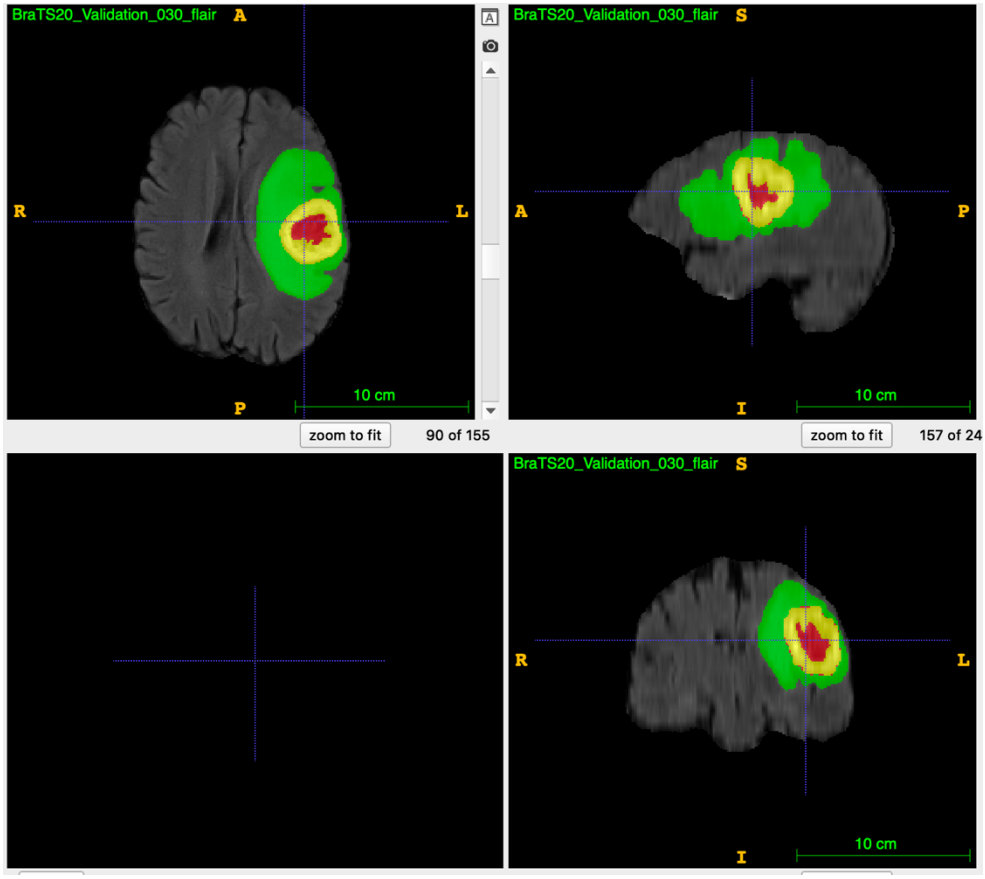


Figure 11 Segmentation result in testing dataset

Chapter 4: Results and Discussion

4.1 Segmentation results

As we have discussed before, the project's training and validation procedure is done in a 5-fold validation manner, which means 160 out of the 200 volumes of brain MRI data are randomly selected for training the model and 40 out of 200 volumes of MRI data are used for validation. For each data used for validation, a " BraTS20_Training_PatientId.nii.gz " was created in the "segs" folder under the "runs " folder, which is used for the final visualization of the segmentation result. For each

For each "BraTS20_Training_PatientId.nii.gz " file, it works like the seg.nii file in the original Training dataset. It is a newly created segmentation result with three different labels for each of the three tumor regions: Enhancing tumor, Tumor Core, and Whole tumor, respectively. As the three labels have different numbers as 3D arrays, it can be intuitively displayed in different colors in ITK-Snap Software or in the visualization block of the code: "display nii data.ipynb" file. Figure 12 shows the segmentation results for three tumor regions ET, TC and WT displayed in software ITK-Snap.

Note that the "BraTS20_Training_PatientId.nii.gz " file for segmentation only contains the areas for the tumor and their corresponding labels. Therefore, if we merely use this file, only tumor regions can be seen in the visualization. To see the tumor regions on the whole brain, we need to drag a nii file for the brain modality, too. Randomly choose one modality from the four modalities is fine, as there are clear textures on each modality. Combine the results of the segmentation and the empty modality file, we can see the corresponding segmentation areas in the brain parts.

The visualization of the segmented 3D MRI brain tumor regions in validation procedure is shown in Figure 12. Here in the picture, I have randomly selected one of the 40 samples for this presentation. And the modality chosen in the picture is T1ce. The segmentations can be displayed in a 3D way, in another word, the results for can be visualized in three axis dynamically. We can simply see the panoramic view of the whole brain and know every aspects of the brain and tumor parts. In the "display nii data.ipynb" code, we use OrthoSlicer3D from nibabel.viewers module to realize the goal. When running the code, a window with three views of the brain will pop up from Jupyter Notebook. Simply by clicking and dragging the window using the mouse can change the view of the brain and tumor.

The three pictures in the left column represent the predicted segmentation results generated by my algorithm, while the pictures in the right column is the ground truth segmentation in the seg.ii file in the original dataset.

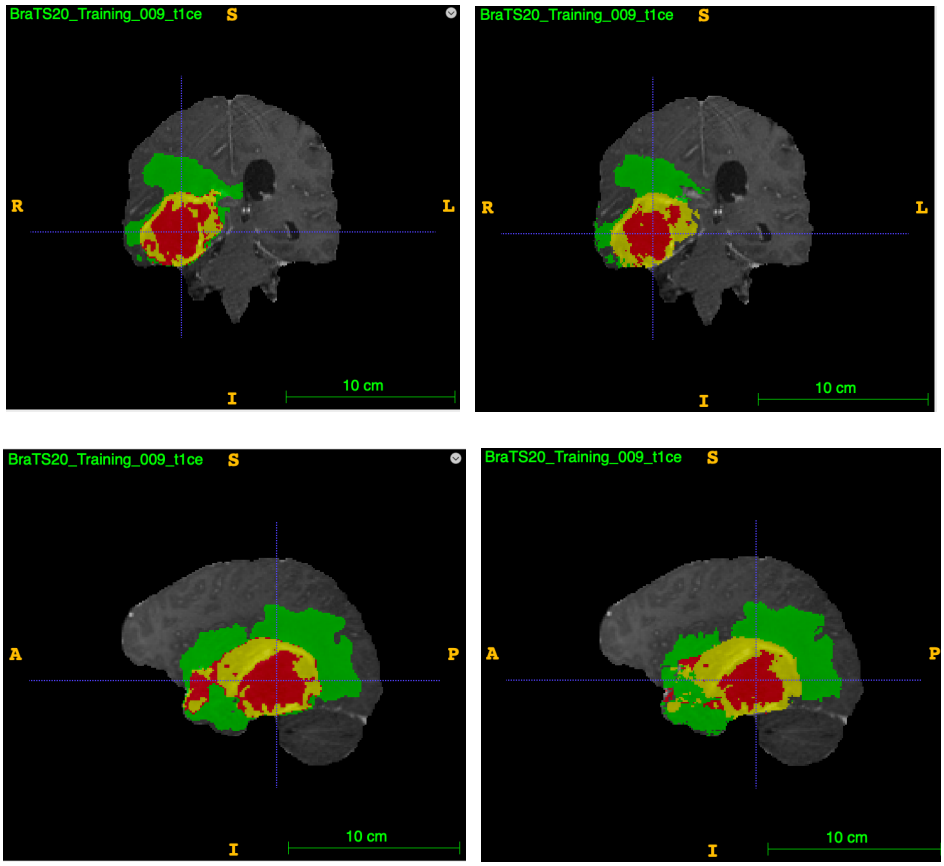
In this case, the results for patient 9 are used for illustration in the figure below. This is a comparison between the segmentation result predicted by my algorithm and model, and the segmentation annotated by the ground truth label in the original dataset.

As we mentioned before in the introduction of the dataset in background, the read part is the enhancing tumor, the yellow part is the necrosis area of the tumor, while the green part is the edema. The targets of segmentation are three overlapping area of these three tumor regions.

Based on the figure, it is easy to see that the predicted segmentation and the ground truth segmentation look very similar. Without boasting, we can say that they are nearly identical, which indicates that the automatic tumor segmentation system I designed performs very well.

For each three predicted segmentation tumor areas: ET, TC and WT, they all look very close when comparing the predicted result with the ground truth. This also demonstrates the effectiveness of my approach, algorithm and system design.

Predicted segmentation result: **Ground Truth label:**



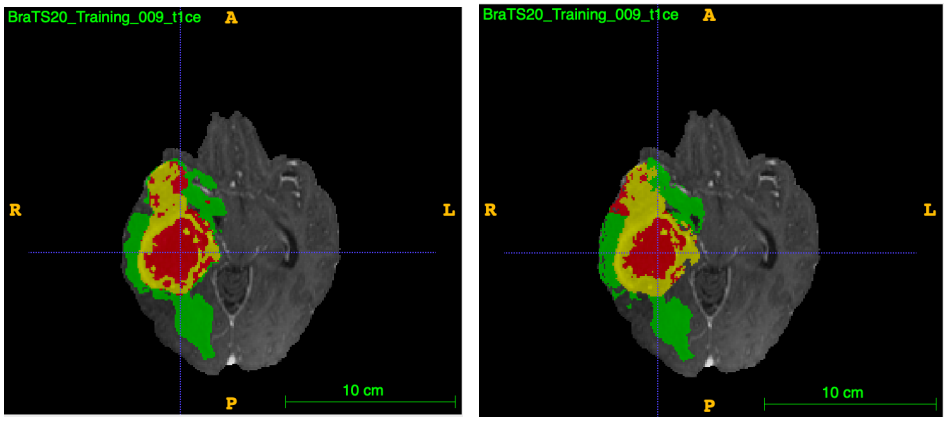


Figure 12 Comparison between predicted segmentation and the ground truth

4.2 Qualitative results

Table 4 shows the average validation results of 40 data volumes in the validation procedure. The metrics used for evaluation are DSC and Hausdorff Distance, Sensitivity, and Specificity(Henry *et al.*, 2021) , which can be used to check the performance of the automatic segmentation system. The principle, meaning and computation of these metrics are presented in the previous section 3.6: Evaluation metrics for validation. The three tumor regions used for qualitative evaluation are ET, TC, and WT, namely enhancing tumor, tumor core, and whole tumor(Wang *et al.*, 2021), respectively.

All these scores are recorded in a csv file for every patient. After the whole training and validation, I used the filter and Average function in EXCEL to calculate the average score for each evaluation metrics. The scores in Table 4 are the mean values for all the validation data predicted by the best training model weights.

Table 4 Mean validation results of DSC, Hausdorff Distance, Sensitivity and Specificity for ET, TC and WT

Metrics	ET	TC	WT
DSC	0.8539	0.9004	0.8865
Hausdorff	7.599	4.882	21.612
Sensitivity	0.894	0.898	0.936
Specificity	0.999	0.999	0.998

From the table, we can see that tumor regions ET, TC, and WT have DSC scores of 0.8539, 0.9004 and 0.8865, respectively. DSC represents the degree of overlap between the true segmentation and the predicted value. All three scores are above 0.85, which is very high, indicating that the segmentation predicted by my algorithm has a large overlap with the real segmentation map, which will be of great value in automatic medical tumor segmentation. From the result, we can see that ET scored lower in DSC, this may be due to the reason that part of the ET region in the training dataset may have been missing.

As shown in Table 4, the Hausdorff values of ET, TC, and WT are 7.591, 4.882, and 21.612 mm, respectively. Hausdorff Distance represents how bad is the outliers of. The higher the value for Hausdorff Distance, the worse the outlier is.

We can see that the Hausdorff value of WT is significantly larger than that of ET and TC. As WT includes the outer section for edema, this indicates that the delineation of the outermost edema part has a larger outlier. This result may also have been caused by the fact that the area occupied by the edema area is originally much larger than the other two tumor areas, which naturally results in a larger distance difference.

As for Sensitivity and Specificity, it is not difficult to conclude that my algorithm is very accurate in predicting the unlabeled background, for the specificity has a rate of nearly 100%. As for sensitivity, the performance is not as good as that of Specificity, but still reaches a result of 89.4%, 89.8% and 93.4%, which is very high. This indicates that the algorithm I developed could predict the positive values in the 3 tumor regions, ET, TC and WT accurately, as well as well predicting the background area labeled with 0.

The following Figure 13 shows the distribution of Hausdorff Distances of ET, TC and WT for all samples in validation. We can see the min, max, median, and quartile value among all the Haudorff Distance values for each tumor area.

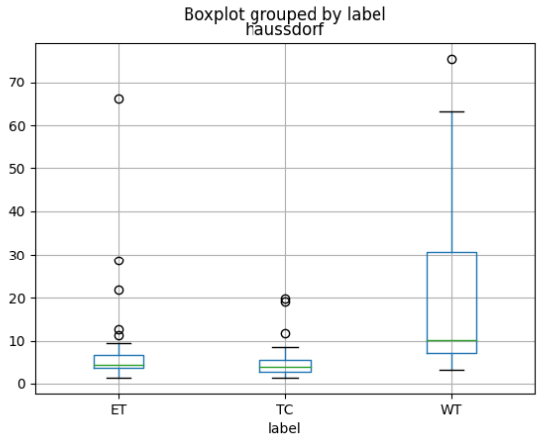


Figure 13 The distribution of Hausdorff Distances of ET, TC and WT for all samples in validation procedure.

As we can see in the box plot in Figure 13, the distribution of Hausdorff for ET is almost between (4, 6). For TC, the distribution is almost between (3, 5), and for WT, the distribution is almost between (8, 31). The median values for ET and TC are almost the same with a value around 5, while the median value for WT with a value around 10.

Figure 14 below shows the distribution of DSC score, sensitivity and specificity of ET, TC and WT for all samples in the validation procedure. They are also shown in box plot. We can see that DSC score and sensitivity for three tumor regions are all distributed around 0.82 to 0.9, and the value of specificity all almost all nearly 1. This means the minimal bounding box strategy taken before at data preprocessing module really worked. By cutting out the unnecessary background area without any tumor, the performance of the segmentation accuracy is enhanced.

We can also draw the conclusion that the evaluation for sensitivity contributes more to the evaluation of the segmentation performance, since sensitivity score is closer to DSC score.

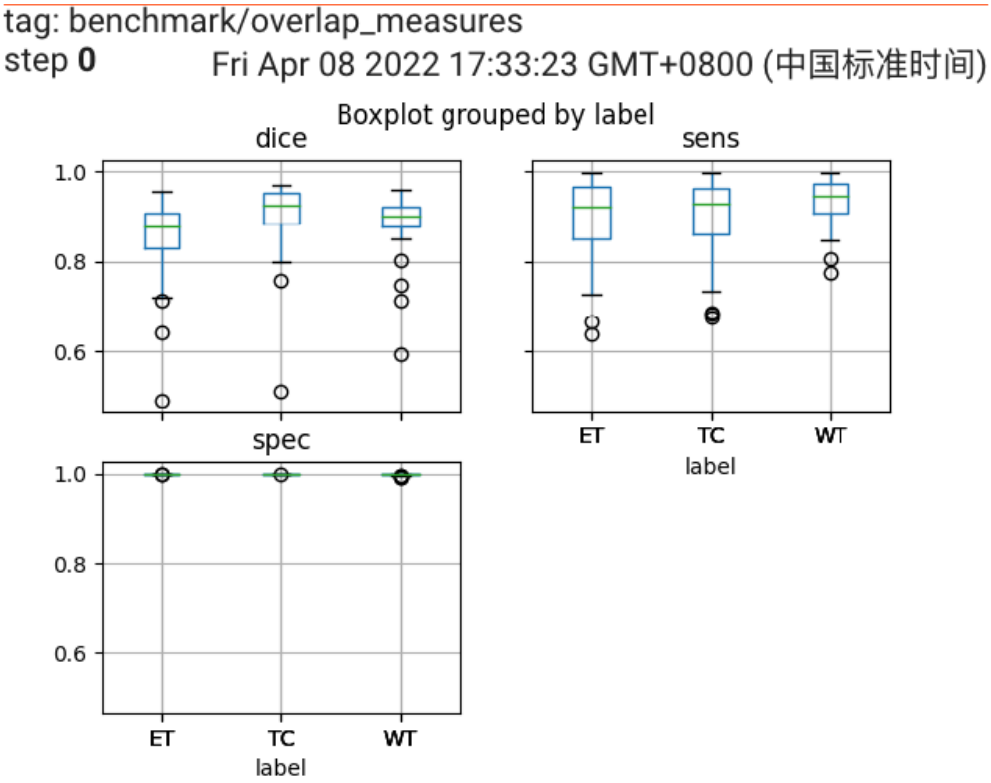


Figure 14 The distribution of DSC score, sensitivity and specificity of ET, TC and WT for all samples in the validation (Box plot).

4.3 Comparison between methods

In their nn-Unet paper, the first-place winner in the BraTS 2020 segmentation competition wrote that: "Our method took the first place in the BraTS 2020 competition with Dice scores of 82.03, 85.06 and 88.95, as well as HD95 values of 17.805, 17.337 and 8.498 for enhancing tumor, tumor core and whole tumor, respectively."

In comparison, my method gained Dice scores of 0.853, 0.900, 0.886 and Hausdorff distance of 7.599, 4.822, 21.612 for ET, TC and WT respectively.

Below is a table showing the comparison of results. Marked in bold are my results. The comparison and numbers are listed in Table 5.

Table 5 Comparison of evaluation metrics for ET, TC between my method and the winner solution nn-Unet in BraST 2020

nn-Unet / **Mine**	ET	TC	WT
DSC	82.03 / **85.39**	85.06 / **90.04**	88.95 / **88.65**
Hausdorff	17.805 / **7.599**	17.337 / **4.822**	8.498 / **21.612**

We can see that my method outperformed the team which won the first place in BraTS 2020 in the predictions for ET and TC. For the prediction of WT, it has a slightly worse performance prediction than that of nn-Unet, but not too far behind it. However, this result may also be attributed to the conjecture that the 40 randomly selected validation samples have a similar morphology and distribution to the whole dataset's distribution, which might allow for better segmentation result.

In addition, I also conducted comparisons between different networks proposed by others. I directly replaced the neural network Atten_Unet in my algorithm with 3D U-Net, and kept the other data preprocessing methods and hyperparameters unchanged, and compared the segmentation results with my method. Table 6 illustrates these results.

Table 6 Comparison between my method and pure 3D-Unet

Pure Unet / **Mine**	ET	TC	WT
DSC	84.83 / **85.39**	89.91 / **90.04**	87.14 / **88.65**
Hausdorff	9.47 / **7.599**	8.53/ **4.822**	33.57 / **21.612**

As these evaluation metrics can well reflect the performance for segmentation, we can interpret the performance from these results. As we can see, all of the DSC scores from my method for three tumor regions are higher than that of the 3D-Unet. And all of the Hausdorff scores from my method is way lower than that of the 3D-Unet. The results in the table above shows that my method with Atten_Unet outperforms the method 3D U-Net in every metric. This indicates that Atten_Unet invented by me is a more advanced network for medical tumor segmentation.

4.4 Discussion

In part 4.1 Segmentation result, we can see that the segmentation system functions very well. However, looking closely, it is not difficult to find some small differences in the voxels between the predicted result and the real tumor regions. In the real world for brain tumor diagnosis, even the slightest error may lead to very serious consequences.

For example, the read enhancing tumor area is smaller in the ground truth than that of the predicted result in the first pair of pictures. This means the system may draw a wrong conclusion of the severeness of different tumor parts, which might lead to mis-treating for patients. On the other hand, the predicted result can also have a larger area of the whole tumor, especially for the edema area. If the doctor really cut the area redundant to the ground truth area in the real world, this may lead to terrifying result as the normal functioning brain cells might be cut out by mistake.

We can see clearly in Figure 15 some slight differences between the predicted result and the ground truth in some local areas. In this way, we can understand the shortcomings and risks of automated tumor segmentation, so that we can better judge its usability and future development direction. Still, the pictures in the left column are predicted result and the pictures on the right are ground truth segmentations.

In the orange circle, we can see that automated segmentation systems sometimes predict normal values that are not part of the tumor as tumor region, and sometimes predict areas of severe tumor necrosis as normal brain cells, which can have dire consequences in real medical application. Therefore, how to segment the tumor correctly and exclude outliers is still a very interesting and valuable part to investigate.

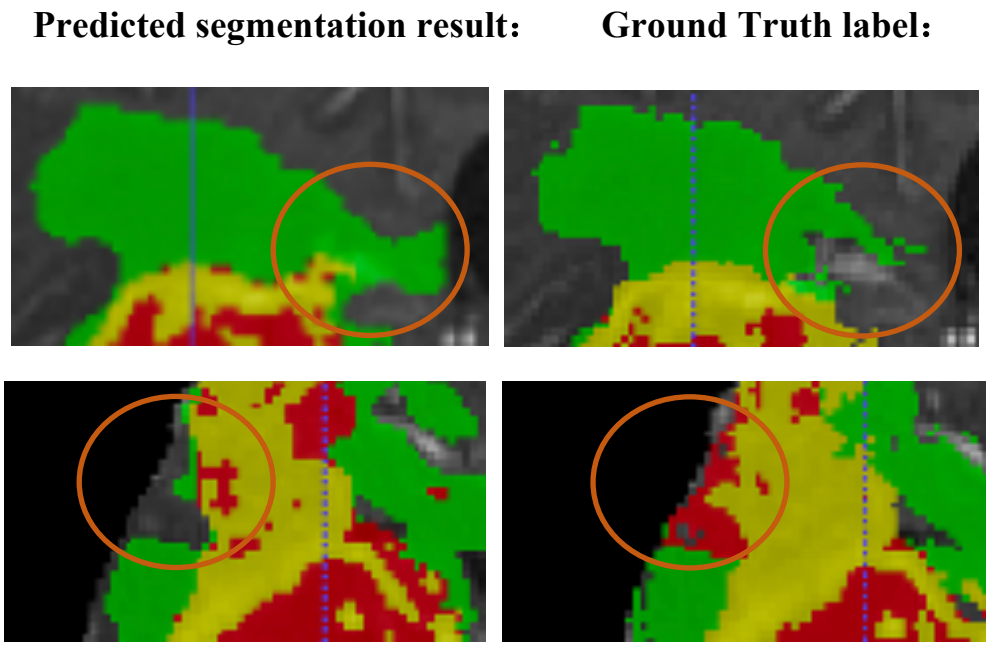


Figure 15 Differences in small regions between predicted and real segmentation

For the methods used for this project, in addition to using the above methods, some other methods and parameter changes have been tried: e.g., using Adamw optimizer instead of Adam optimizer, removing the 'dropping channel' data augmentation method, replacing the network with 3D-Unet, increasing the range of noise, etc. However, these changes in methods and parameters did not bring any improvements in the segmentation results. On the contrary, they somehow had a negative impact on the segmentation results. After a series of comparisons, the method I am using will give the best results with the existing dataset size.

In terms of segmentation results, the whole tumor (WT) region is not as good as the Enhancing tumor (ET) and Tumor core (TC). Since $WT = TC + edema$, $TC = ET + necrosis$, we can probably conclude that the main segmentation error comes from the edema region. If we can improve the processing and training of the edema region in the future, we might be able to achieve better segmentation results.

Moreover, the labelling of real labels in the dataset is done by experienced scientists currently. However, the results obtained by doctors with less experience may also be inaccurate. In addition, the physician cannot segment each region of the tumor precisely at the voxel level with visual observation. Therefore, it is possible that the physician's error was the root of the segmentation error. Therefore, automated tumor segmentation systems sometimes may only have the potential to provide advice to physicians and correct the possible wrong decisions made by physicians.

Chapter 5: Conclusion and Further Work

In this project, I developed a new algorithm for automatic tumor segmentation by proposing a new network called Atten_Unet based on 3D-Unet by adding CBAM modules. In addition, various new data preprocessing and data augmentation methods were adopted in the project. Using this segmentation algorithm, I was able to train, validate, and test on a portion of the BraTS 2020 dataset and achieve good segmentation performance, outperforming a lot of methods proposed by others, even the championship team in BraTS 2020 competition. Clearly, this fully demonstrates the effectiveness of my method.

Due to the limitations of computing devices, I only used a portion of the dataset for training and validation, which may lead to inaccurate results and may also show less convincing superiority in comparison experiments. I will need to test on the entire dataset in the future when better conditions are available.

As we can see from the loss graph from Tensorboard, the loss continues to decrease at 150 epochs, which indicates that it is very likely to achieve better segmentation results if I continue to train more epochs. In that case, I will also train and validate more epochs in the future to gain better results in the future and it may be a very valuable system for future medical segmentation field.

Also, due to time constraints, I have some other ideas that I have not been able to try, such as adapting the network to a Swin-Transformer variant in every block of the encoder and decoder, or permuting other data augmentation strategies to test the performance. These are all potentially effective approaches that I will also try in the future to improve my system.

In a nutshell, this system is likely to provide a more accurate segmentation effect and pave the way for its actual application in the medical field in the future. But there is still a lot of factors that worth working on and making improvements.

Perhaps, the future medical system will be improved to a great extent with the assistance of artificial intelligence to truly benefit human health. Let's look forward to it.

References

Cao, H. *et al.* (2021) ‘Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation’. Available at: https://arxiv.org/abs/2105.05537v1 (Accessed: 2 March 2022).

Çiçek, Ö. *et al.* (2016) ‘3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9901 LNCS, pp. 424–432. doi: 10.1007/978-3-319-46723-8_49.

Hatamizadeh, A. *et al.* (2021) ‘UNETR: Transformers for 3D Medical Image Segmentation’. Available at: https://arxiv.org/abs/2103.10504v3 (Accessed: 21 November 2021).

Henry, T. *et al.* (2021) ‘Brain Tumor Segmentation with Self-ensembled, Deeply-Supervised 3D U-Net Neural Networks: A BraTS 2020 Challenge Solution’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12658 LNCS, pp. 327–339. doi: 10.1007/978-3-030-72084-1_30.

Isensee, F. *et al.* (2020a) ‘nnU-Net for Brain Tumor Segmentation’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12659 LNCS, pp. 118–132. doi: 10.48550/arxiv.2011.00848.

Isensee, F. *et al.* (2020b) ‘nnU-Net for Brain Tumor Segmentation （2020 第一名）’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12659 LNCS, pp. 118–132. doi: 10.48550/arxiv.2011.00848.

Milletari, F., Navab, N. and Ahmadi, S. A. (2016) ‘V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation’, *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pp. 565–571. doi: 10.1109/3DV.2016.79.

Myronenko, A. (2019) ‘3D MRI brain tumor segmentation using autoencoder regularization (ResNet3D-VAE)’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11384 LNCS, pp. 311–320. doi: 10.1007/978-3-030-11726-9_28.

Myronenko, A. and Hatamizadeh, A. (2020) ‘Robust semantic segmentation of brain tumor regions from 3D MRIs’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11993 LNCS, pp. 82–89. doi: 10.1007/978-3-030-46643-5_8.

″An algorithm for 3D MRI brain tumor segmentation″

Vaswani, A. *et al.* (2017) 'Attention is all you need', *Advances in Neural Information Processing Systems*, 2017-Decem(Nips), pp. 5999–6009.

Wang, W. *et al.* (2021) 'TransBTS: Multimodal Brain Tumor Segmentation Using Transformer', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12901 LNCS, pp. 109–119. doi: 10.1007/978-3-030-87193-2_11.

Woo, S. *et al.* (2018) 'CBAM: Convolutional Block Attention Module', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11211 LNCS, pp. 3–19. doi: 10.48550/arxiv.1807.06521.

Wu, Y. and He, K. (2018) 'Group Normalization', *International Journal of Computer Vision*, 128(3), pp. 742–755. doi: 10.48550/arxiv.1803.08494.

Acknowledgement

My family has given me endless support during the process of completing my project. I really want to thank my mom, dad and my sister. They gave me endless patience, care, and spiritual support when I was under pressure. When I couldn't write my dissertation and started to complain about my poor written expression ability, they gave me the courage and confidence to keep going, so that I could re-dedicate myself to my in-depth research and love of research.

I would also like to thank my supervisor for honing my ability to complete a huge project on my own and inspiring my potential, showing me that human potential is infinite. The approval and support I received from my supervisor also gave me a lot of courage to start my research in such a completely uncharted field of medical artificial intelligence before. In the process of completing my project, my teacher also provided me with some clever inspirations for my project, which allowed me to design my project successfully. This is also the project I like because I want to combine medicine and artificial intelligence to contribute to the realization of medical automation and promote the development of some people's life and health.

Finally, I'd like to express gratitude to the field of computer science, which I love. I hope I can make some contribution to human beings in the field of computers, especially in the field of artificial intelligence, with the one I love in the future, which is my biggest dream.

Appendix

北京邮电大学 本科毕业设计（论文）任务书
Project Specification Form
Part 1 – Supervisor

论文题目 Project Title	An Algorithm for 3D MRI Brain Tumor Segmentation		
题目分类 Scope	Data Science and Artificial Intelligence	Research	Simulation
主要内容 **Project description**	"MRI modalities are acquired - such as T1, T1 with contrast agent (T1c), T2 and Fluid Attenuation Inversion Recover (FLAIR) - to emphasize different tissue properties and areas of tumor spread. Automated segmentation of brain tumors from 3D magnetic resonance images (MRIs) is necessary for the diagnosis, monitoring, and treatment planning of the disease. Manual delineation practices require anatomical knowledge, are expensive, time consuming and can be inaccurate due to human error." "Automated segmentation of 3D brain tumors can save physicians time and provide an accurate reproducible solution for further tumor analysis and monitoring. Recently, deep learning-based segmentation techniques surpassed traditional computer vision methods for dense semantic segmentation. Convolutional neural networks (CNN) are able to learn from examples and demonstrate state-of-the-art segmentation accuracy both in 2D natural images and in 3D medical image modalities." "We aim to propose a method for the segmentation of brain tumors by providing a 3D MRI dataset with ground truth tumor segmentation labels annotated by physicians."		
关键词 Keywords	MRI, image segmentation, brain tumor, CNNs		
主要任务 Main tasks	**1** Collect and preprocess brain tumor 3D MRI datasets		
	2 Compare the proposed segmentation algorithms		
	3 "Propose one or more method for the segmentation of brain tumors"		
	4 Optimize the algorithm to improve the segmentation accuracy		
主要成果 Measurable outcomes	**1** Create a more complete brain tumor 3D MRI dataset		
	2 "Increase 3D MRI brain tumor segmentation accuracy"		
	3 "A 3D MRI brain tumor segmentation system"		

"北京邮电大学 本科毕业设计（论文）任务书"
"Project Specification Form"
"Part 2 – Student"

"学院 School"	International School	**"专业 Programme"**	**Internet of Things Engineering**		
"姓 Family name"	Dong	**"名 First Name"**	Wanqi		
"BUPT 学号 BUPT number"	2018213196	**"QM 学号 QM number"**	190019048	**"班级 Class"**	2018215121
论文题目 Project Title	An Algorithm for 3D MRI Brain Tumor Segmentation				

"论文概述 Project outline Write about 500-800 words Please refer to Project Student Handbook section 3.2"	**Background and dataset** "Brain tumor is a mass or growth of abnormal cells in the brain, which has claimed millions of lives of human beings. High grade gliomas are an aggressive type of malignant brain tumor that grow rapidly, usually require surgery and radiotherapy. Therefore, it's significant to detect and know where exactly the tumor is and emphasize different tissue properties and areas of tumor spread. Magnetic Resonance Imaging (MRI) is a key diagnostic tool for brain tumor analysis, monitoring and surgery planning. Usually, several complimentary 3D MRI modalities are acquired - such as T1, T1 with contrast agent (T1c), T2 and Fluid Attenuation Inversion Recover (FLAIR). " "Compared to manual delineation practices, which are expensive, time consuming and can be inaccurate due to human error, automated segmentation of 3D brain tumors can save physicians time and provide an accurate reproducible solution for further tumor analysis and monitoring. Therefore, the main purpose of my experiment is to propose a better algorithm or network for brain tumor subregion segmentation. I think the result could provide suggestions for doctors and will be a crucial and meaningful practice for brain tumor diagnosis." In this project, the dataset comes from "Multimodal Brain Tumor Segmentation Challenge (BraTS)". "It is a 3D MRI dataset with ground truth tumor segmentation labels annotated by physicians. The data were collected from multiple institutions, using various MRI scanners. Annotations include 3 nested subregions: whole tumor (WT), tumor core (TC) and enhancing tumor (ET). I will use a portion of these data and perform tasks for volumetric 3D brain tumor subregion segmentation, constantly improving my methods to achieve better segmentation accuracy." **The algorithms methodologies and other techniques to be employed** First, I will read papers related to this field and learn knowledge about medical 3D MRI data properties and processing methods. "Second, I will use image pre-processing methods to do some data pre-processing on the dataset, such as enhancement on contrast, brightness, and then enlarge the dataset using several data augmentation methods such as rotation and flips." "Then I'll utilize several different 3D data segmentation methods such as Encoder-decoder based CNN architecture, 3D U-net and V-net, transformer-based methods to perform 3D MRI dataset segmentation tasks, then compare those methods based on their performance and properties." "Next, I'll study the recently hottest methods like ViT or Swin Transformer and try to use them for my experiment. I will make some adjustment and innovations based on them and propose my own method which might achieve a better segmentation result". Then, I will optimize my result and

	do some result fine-tuning work. "Moreover, for medical volumetric data (3D MRI scans) which is beyond 2D, local feature modeling among continuous slices is also critical for volumetric segmentation, therefore, in the process, I will try to utilize the local and global features in spatial and depth dimensions of volumetric data and get inspirations, then make improvements for my own method." "Finally, I'll use my method to perform segmentation task on 3D MRI brain tumor validation set and compare my result with the previously proposed methods. If time allowed, I might also develop a 3D MRI brain tumor segmentation system to visualize the segmentation result in a 3D way." **Programming language and Framework that'll be used:** Python, PyTorch, Numpy **A list of background materials consulted:** 1. https://zhuanlan.zhihu.com/p/355882011 2. https://zhuanlan.zhihu.com/p/331358945 3. https://zhuanlan.zhihu.com/p/48750068 4. https://www.synapse.org/#!Synapse:syn25829067/wiki/610863 5. https://zhuanlan.zhihu.com/p/336911305 6. https://zhuanlan.zhihu.com/p/360513527 " 7. Wang W, Chen C, Ding M, et al. Transbts: Multimodal brain tumor segmentation using transformer[C]//International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, 2021: 109-119. 8. Myronenko A, Hatamizadeh A. Robust semantic segmentation of brain tumor regions from 3D MRIs[C]//International MICCAI Brainlesion Workshop. Springer, Cham, 2019: 82-89. 9. Henry T, Carre A, Lerousseau M, et al. Brain tumor segmentation with self-ensembled, deeply-supervised 3D U-net neural networks: a BraTS 2020 challenge solution[J]. arXiv preprint arXiv:2011.01045, 2020. 10. Myronenko A. 3D MRI brain tumor segmentation using autoencoder regularization[C]//International MICCAI Brainlesion Workshop. Springer, Cham, 2018: 311-320."
道德规范 **Ethics**	"Please confirm that you have discussed ethical issues with your Supervisor using the ethics checklist (Project Handbook Appendix 1). [YES/NO] YES"

	"Summary of ethical issues: (put N/A if not applicable) N/A"
"中期目标 **Mid-term target.** **It must be tangible outcomes, E.g. software, hardware or simulation.** **It will be assessed at the mid-term oral."**	1. Learn how to use and process volumetric 3D MRI dataset and finish image pre-processing and data augmentation task. 2. Train the dataset utilizing several existing methods and try to reproduce the result. 3. Research on papers and think about how to make innovations based on those methods.

"Work Plan (Gantt Chart)"

"Fill in the sub-tasks and insert a letter X in the cells to show the extent of each task"

	Nov 1-15	**'Nov 16-30**	**Dec 1-15**	**Dec 16-31**	**Jan 1-15**	**Jan 16-31**	**Feb 1-15**	**Feb 16-28**	**Mar 1-15**	**Mar 16-31**	**Apr 1-15**	**Apr 16-30**
Task 1 Collect and preprocess brain tumor 3D MRI datasets												
Do background reading about the dataset	X	X										
Get familiar with 3D MRI dataset and learn how to process it		X	X									
Finish dataset pre-processing			X	X								
Finish data augmentation work on the dataset.			X	X	X							
Task 2 Compare the proposed segmentation algorithms												
Do background reading and study papers about existing segmentation methods			X	X	X	X						

Choose some 3D based segmentation methods, read and understand the paper (or code)				X	X	X						
Train the dataset utilizing those methods and try to reproduce the result, finish the code and get the result.				X	X	X	X	X				
Compare and test different methods. Finish a survey report about the comparison.						X	X	X				
"Task 3 Propose one or more method for the segmentation of brain tumors"												
“Study papers and derive innovations from them”						X	X	X				
Make improvements on existing methods or model. Propose at least one new method.								X	X			
Train the dataset on the newly proposed method and test the result. Finish the code.								X	X	X		
Compare my method with previous methods, and show them in chart or report.								X	X	X		
Task 4 Optimize the algorithm to improve the segmentation accuracy												
Utilize machine learning methods to make improvements to the algorithm									X	X		
Utilize spatial features to make improvements.										X	X	
Optimize the result using fine-tuning methods										X	X	X
Modify the code and improve the result. Finish the paper.										X	X	X

"北京邮电大学 本科毕业设计（论文）初期进度报告"
"Project Early-term Progress Report"

"学院 School"	International School	**"专业 Programme"**	**Internet of Things Engineering**		
"姓 Family name"	Dong	**"名 First Name"**	Wanqi		
"BUPT 学号 BUPT number"	2018213196	**"QM 学号 QM number"**	190019048	**"班级 Class"**	2018215121
"论文题目 Project Title"	An Algorithm for 3D MRI Brain Tumor Segmentation				

"已完成工作 Finished work: "

1. "Summary of materials read and researched:"

In the past few months, I read and researched a lot of materials about MRI brain tumor datasets and possible processing methods.

First, I learned about the 3D MRI brain Tumor dataset from various sources, such as the official website of "Multimodal Brain Tumor Segmentation Challenge (BraTS)". Through these ways, I learned about the format of 3D MRI datasets (nii.gz), The number of pictures, dimensions, labels, classes, and features of each MRI brain tumor volume.

Then I read a lot of related papers, trying to find a neural network or algorithm suitable for this problem. For example, several papers I have carefully studied are:

(1) 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation.

(2) UNETR: Transformers for 3D Medical Image Segmentation

(3) A Volumetric Transformer for Accurate 3D Tumor Segmentation.

(4) Extending 2D Deep Learning Architectures to 3D Image Segmentation Problems.

From these papers, I got some inspiration about how to process Brain MRI data and the consideration of extending 2D data to 3D data as well as the importance of taking into account spatial information of 3D data.

Finally, I decided to use 3D-Unet to try it first. More details will be explained in the following parts.

2. Summary of work done

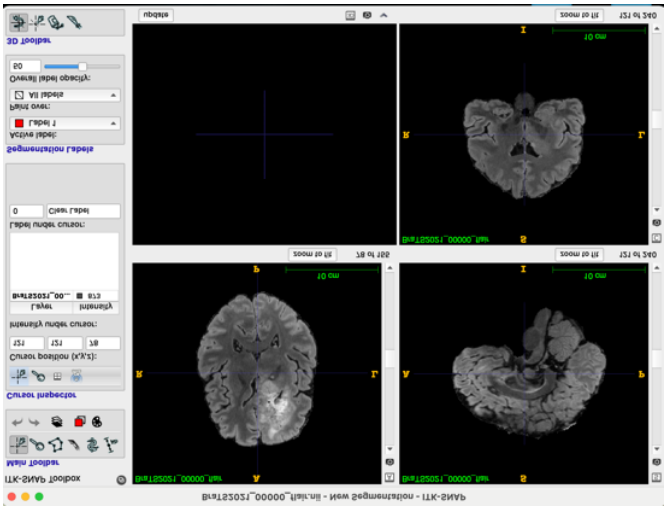
(1) Open and visualize the data in nii.gz format.

Because the brain tumor dataset is in 3D format, each brain MRI scan is stored in a nii.gz format file, which requires special opening, loading, and processing methods.

I learned to open and visualize the dataset in two ways, one is using the software called ITK-snap, while the other is by programming. As for the method of using the software, I downloaded a software specially used to view medical 3D data: ITK-snap, to view the data. you can view components from different axes and dynamically observe the brain and tumor in every aspect.

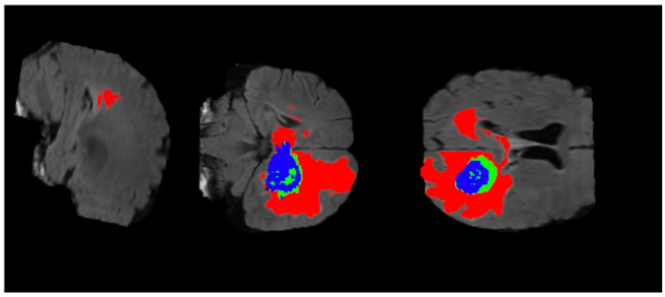
Regarding the way of programming, I use package nibabel, matplotlib, and some functions written by myself to visualize the brain MRI scan and the tumor parts of different channels. By using the code ***image.shape***, we can know that the shape of the data in each nii.gz file is

(240, 240, 155, 4) respectively (height, length, depth, channel), where channel represents the different parts of the brain and tumor areas.
The following figure is a function written in matplotlib, which can dynamically visualize different parts of a brain MRI scan.



Method 1: Open the nii.gz dataset using ITK-snap software

```
image, label = load_case(DATA_DIR + "imagesTr/BRATS_003.nii.gz", DATA_DIR + "labelsTr/BRATS_003.nii.gz")
util.visualize_data_gif(util.get_labeled_image(image, label))
```



Method 2: Write code to open a single nii.gz dataset, display it intuitively and interactively

(1) Data pre-processing

1. "Because the size of the data in the brain MRI dataset varies greatly, we need to pre-process the data for better training of the neural network. I learned how to normalize the data in the scan, and process one of the nii.gz datasets to have a mean of 0 and a standard deviation of 1. In this way, it provides convenience for the training of the neural network in later process."

2. In addition, since the dataset contains many nii.gz files and each nii.gz file contains many slices. it will cause an out of memory error in GPU if the slices of the whole brain MRI scan are put into the neural network to train together. Therefore, I learned how to slice subregions in the entire brain scan data, cyclically take multiple subregions and put them into the neural network. In order to make the obtained subregion most valuable for training, we need to ensure that the obtained subregion contains at least some of the part which contains the tumor. So, I set a threshold value and only the subregion in which the tumor occupies at least 5% of the whole part is considered as a valid subregion to slice.

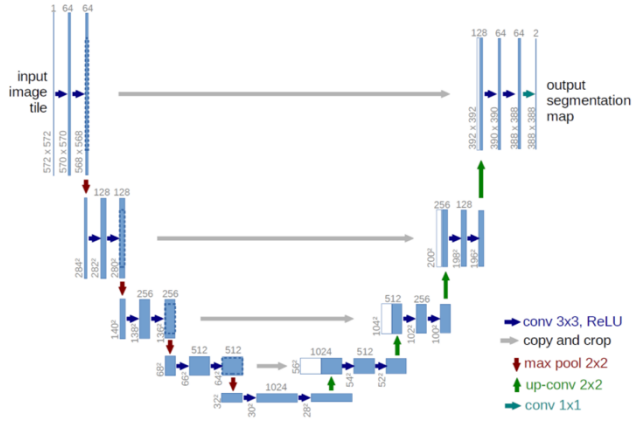
Here is part of the code (The header of the function)

```
def get_sub_volume(image, label,
                   orig_x = 240, orig_y = 240, orig_z = 155,
                   output_x = 160, output_y = 160, output_z = 16,
                   num_classes = 4, max_tries = 1000,
                   background_threshold=0.95):
    """
```

3. In addition, I also tried to learn the data augmentation method using pytorch to process the dataset, such as random crop, flip, rotate. etc.

3. Tried to use 3D-Unet network to train the data of some subregions of one nii.gz file

The following graph is the overall architecture of the 3D-Unet



The loss function used is Multi-Class Soft Dice loss:

$$\mathcal{L}_{Dice}(p,q) = 1 - \frac{1}{N}\sum_{c=1}^{C}\frac{2\times\sum_{i,j} p_{cij}q_{cij} + \epsilon}{\left(\sum_{i,j} p_{cij}^2\right) + \left(\sum_{i,j} q_{cij}^2\right) + \epsilon}$$

"The training uses the weights of 3D-Unet that have been trained by others. Training is on-going in Google Colab. It will finally combine the results obtained by each subregion and visualize together. I uses sensitivity and specificity as the evaluation metrics of the performance of the model."

$$\text{sensitivity} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$\text{specificity} = \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}}$$

The training results have not come out yet, I will continue to work hard.

3. Problems faced

(1) Since 3D MRI dataset is 3D data and contains a large quantity of data, it has high requirements on computing equipment. Training on the entire dataset will need many advanced GPUs and takes many days.

(2) Unfamiliar with 3D data, unfamiliar with 3D neural network, and need to make great modifications according to 2D data and 2D networks.

4. Solutions found

(1) I can take part of the nii.gz file for training.

(2) A better way can be found to take the most effective and valuable 3D brain subregion and fed them for model training.
(3) Using Google Colab Pro++ for training, in this way, I can get a V100 GPU. (A little expensive)

"是否符合进度？On schedule as per GANTT chart?"
YES

"下一步 Next steps: "
"For the next step of research, I plan to do it in the following steps:"
(1) Try to solve the problem of too much data to train, take parts of the valuable data for training
(2) Find and try more neural networks and algorithms for 3D tumor segmentation, and compare the results.
(3) Come up with my own algorithms, get the best results with as little training data as possible.

"北京邮电大学 本科毕业设计（论文）中期进度报告"
"Project Mid-term Progress Report"

"学院 School"	International School	**"专业 Programme"**	**Internet of Things Engineering**		
"姓 Family name"	Dong	**"名 First Name"**	Wanqi		
"BUPT 学号 BUPT number"	2018213196	**"QM 学号 QM number"**	190019048	**"班级 Class"**	2018215121
"论文题目 Project Title"	An algorithm for 3D MRI brain tumor segmentation				

"是否完成任务书中所定的中期目标？Targets met (as set in the Specification)?
YES"

"已完成工作 Finished work:"

5. The new usage of the 3D MRI brain tumor dataset.

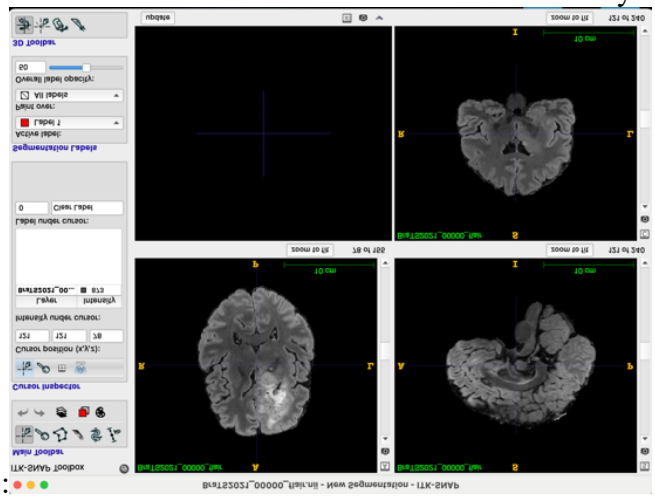
During the past few months, I've developed a more comprehensive understanding of the 3D MRI brain tumor dataset. In my experiment, I decide to use the dataset provided by MICCAI BraTS2021 Challenge. "Due to the limitation of the computational device, I randomly chose 200 volumes of 3D MRI data for training and validation from the total 1600 volumes of data from the BraTS training dataset. "The experiments below are all based on this premise.

"BraTS dataset contains four distinct tumor sub-regions: (1) The Enhancing Tumor (ET), (2) Non Enhancing Tumor (NET), (3) Necrotic Tumor (NCR) which alongwith NET, (4) Peritumoral Edema (ED). As for the segmentation task, I decide to classify these subregions into three semantically meaningful tumor classes: (1) Enhancing Tumor (ET), (2) the Tumor Core (TC) region (addition of ET, NET and NCR), and (3) the Whole Tumor (WT) region (addition of ED to TC). The Label ET, TC, and WT will be meaningful in the following segmentation experiment and evaluation procedure."

6. Visualize the dataset in an intuitive and interactive way

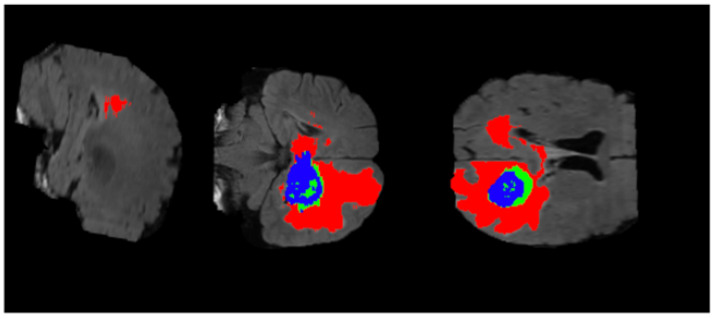
As the brain tumor dataset is in 3D volumetric format, each brain MRI volume is stored in a nii.gz format file, which requires special opening and processing methods.
I learned to open and visualize the dataset in two ways, one is using the software called ITK-snap, while the other is by programming. With the software ITK-snap, I can view the 3D MRI volume in 3 dimensions and from different axes. I can also use the mouse to rotate

the 3D MRI volume back and forth to view it from any angle. As shown in the following picture:



The second way to visualize the nii.gz data is by programming. I use the package nibabel, matplotlib, and some functions written by myself to visualize the 3D MRI data in different channels. In this way, I know that the Image Shape of each 3D MRI volume is: " 240(Slide Width) × 240(Slide Height) × 155(Number of Slide) × 4(Multi-mode)".
The four modes are FLAIR, T1, T1c, T2, respectively.
The following figure is a function written in matplotlib, which can dynamically visualize the 3D MRI data. We can change the view simply by dragging the picture.

```
image, label = load_case(DATA_DIR + "imagesTr/BRATS_003.nii.gz", DATA_DIR + "labelsTr/BRATS_003.nii.gz")
util.visualize_data_gif(util.get_labeled_image(image, label))
```



7. Read and researched more papers on volumetric segmentation methods.

In order to find a better find a neural network or algorithm suitable for volumetric medical data segmentation, I investigate a lot papers trying to find feasible methods.
Here are several papers I have carefully studied:
"(1) 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation.
(2) V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation
(3) 3D MRI brain tumor segmentation using autoencoder regularization (ResNet3D-VAE)
(4) A Volumetric Transformer for Accurate 3D Tumor Segmentation. (VT-UNet)
(5) TransBTS: Multimodal Brain Tumor Segmentation Using Transformer"

The paper (1)(2)(3) is based on relatively traditional volumetric segmentation methods, while the paper (4)(5) is based on Transformer model, which is more popular in recent years.
From these papers, I got some inspiration about how to process 3D MRI data and how to use the data for training and testing.

Finally, I decided to use 3D-Unet and VT-UNet to implement the result, for the reason that one method is based on traditional segmentation method and the other is based on Transformer. I can compare the result by training with these two networks. More details will be explained in the following parts.

3. Data pre-processing and augmentation

"As the BraTs dataset is a mature dataset collected by several institutions, there's not so much pre-processing work to do. But as the data from different background has some difference, I learned how to normalize the data in the 3D volume, and process the image to have a mean of 0 and a standard deviation of 1, which provides convenience for the training of the neural network in later process."

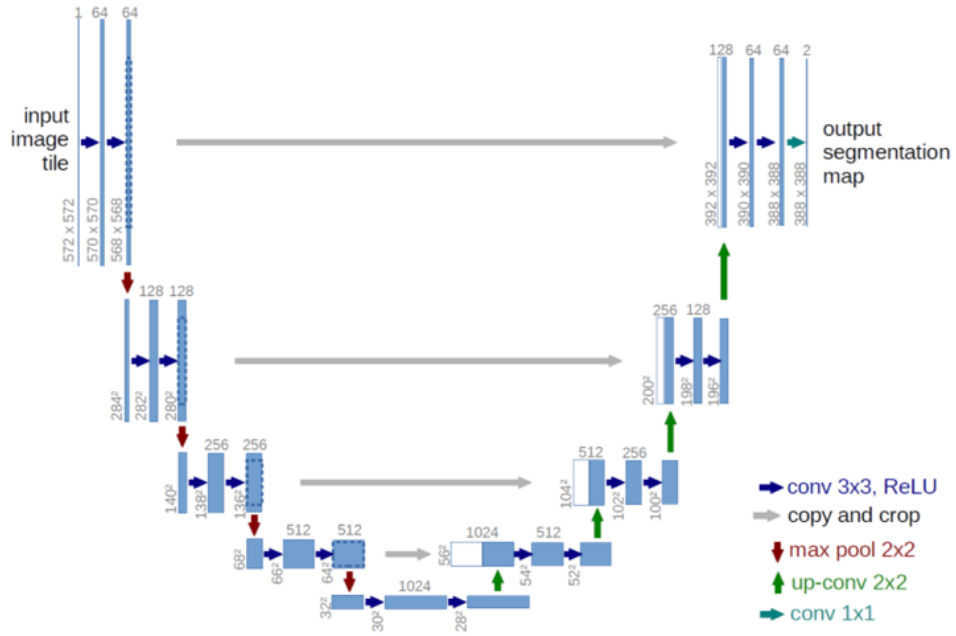
In addition, I also tried to use data augmentation method to augment the data, the method I used are listed below: (Flip, Rotation, Shift, Shear)

Methods	Range
Flip horizontally	50% probability
Flip vertically	50% probability
Rotation	±20°
Shift	10% on both horizontal and vertical direction
Shear	20% on horizontal direction

These augmentation methods can be implemented using PyTorch framework before the training process. Maybe I'll also try other augmentation methods like zoom, changing brightness and elastic distortion on the dataset in the future.

4. Try to train 3D-UNet network on 2/3 of the 200 volumes of 3D MRI brain tumor data and test the result on the rest of the data (validation set).

The following graph is the overall architecture of the 3D-Unet



The loss function used is Multi-Class Soft Dice loss:

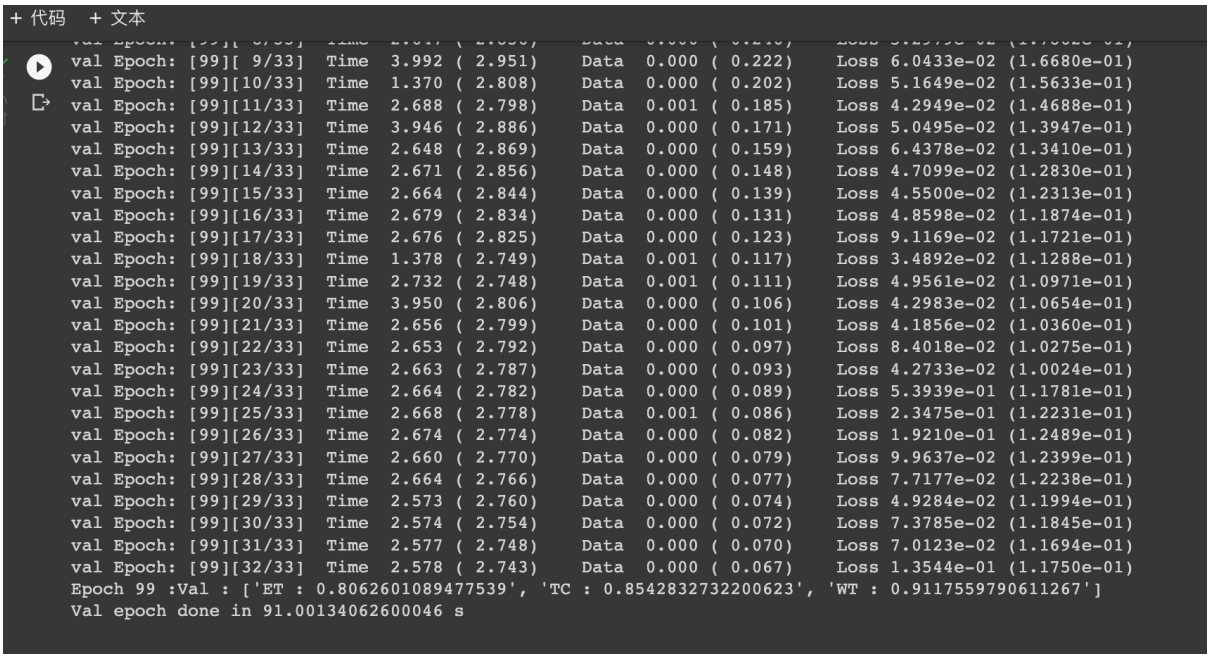
$$\mathcal{L}_{Dice}(p,q) = 1 - \frac{1}{N}\sum_{c=1}^{C}\frac{2\times\sum_{i,j} p_{cij}q_{cij} + \epsilon}{\left(\sum_{i,j} p_{cij}^2\right) + \left(\sum_{i,j} q_{cij}^2\right) + \epsilon}$$

"The training procedure is done in Google Colab, where I could use one Telsa V-100 GPU for training. For the training and validation process, the five-fold cross validation has been employed."
"Dice coefficient is used as the validation metrics.
After validation, the Dice coefficient score for ET, TC and WT is 0.72, 0.79 and 0.88 respectively."

5. Try to train VT-UNet network on 2/3 of the 200 volumes of 3D MRI brain tumor data and test the result on the rest of the data (validation set).

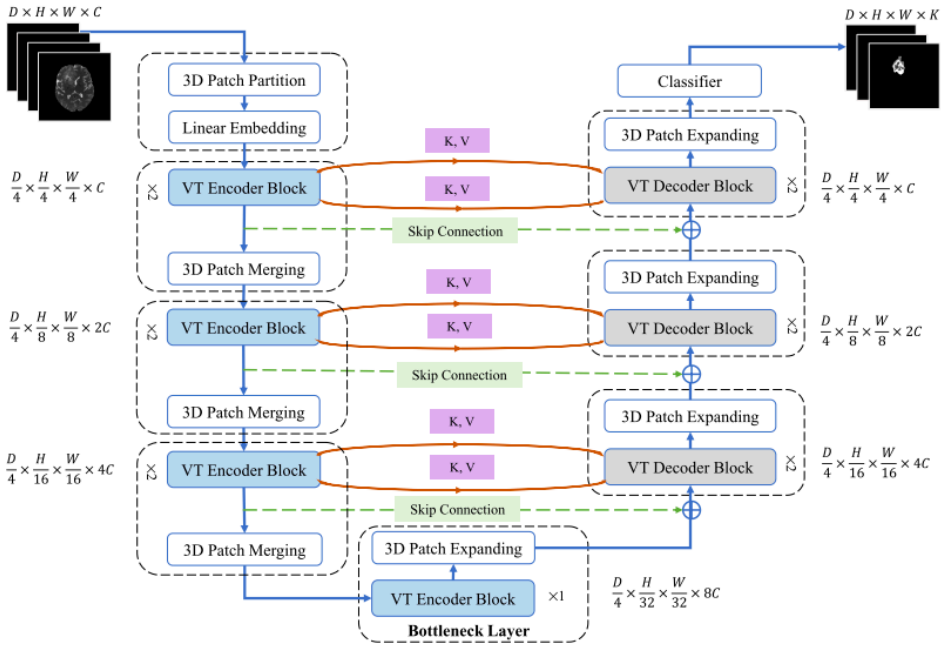
"As Transformer based approaches for 3D medical image segmentation have shown their advantages compared with their CNN counterparts, I decided to use one Transformer based model for testing. In this experiment, I chose VT-Unet for training.
The training process is implemented with a single Nvidia Telsa V-100 GPU on Google Colab. The weights of Swin-T pre-trained on ImageNet-1K are used to initialize the model. For training, Adam optimizer with a learning rate of 1e−4 is used.
Due to the limitation of the computational resource, I only train the model for 100 epochs using a cosine decay learning rate scheduler and a batch size of 1. The total training time is 7 and a half hours on Google Colab.
The model with best results are selected on validation set for evaluation. Dice coefficient (Dice Score) and Hausdorff Distance (HD) are used as evaluation metrics. I separately compute them for the three classes (ET, TC and WT)"

The following picture is the screen shot of the evaluation result of the traninng and validation result of the last epoch:



"As we can see, the Dice coefficient score for ET, TC and WT is 0.806, 0.854 and 0.911, respectively."

"The VT-Unet architecture is as below:"



"To train VT-UNet, I jointly minimize the Dice Loss (DL) and Cross Entropy (CE) loss. The two losses are modified and computed in a voxel-wise manner. The Dice loss is defined as:"

$$\mathcal{L}_{\mathrm{dl}}(\theta;\mathcal{X}) = -\mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{X}}\left[\frac{2\langle \mathbf{Y}\ ,\ \mathcal{H}(\mathbf{X})\rangle}{\|\mathbf{Y}\|_1 + \|\mathcal{H}(\mathbf{X})\|_1}\right],$$

"where H(·) denotes the transformer model and θ denotes the model parameters.
The Cross Entropy loss is defined as:"

$$\mathcal{L}_{\mathrm{ce}}(\theta;\mathcal{X}) = \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{X}}\Big[-\mathbf{Y}\log\mathcal{H}(\mathbf{X})\Big],$$

Therefore, the total segmentation loss used is the combination of the above two loss:

$$\mathcal{L}_{\mathrm{seg}}(\theta;\mathcal{X}) = \mathcal{L}_{\mathrm{dl}}(\theta;\mathcal{X}) + \mathcal{L}_{\mathrm{ce}}(\theta;\mathcal{X}).$$

"After validation, the Dice coefficient score for ET, TC and WT is 0.806, 0.854 and 0.911 respectively, which is a lot higher than the result gotten from the 3D-Unet above."

6. Compare the two methods

"For 3D-Unet, the Dice coefficient scores for ET, TC and WT (3 classes) are 0.72, 0.79 and 0.88 respectively ".
"For VT-Unet, the Dice coefficient score for ET, TC and WT (3 classes) is 0.806, 0.854 and 0.911 respectively".
"According to the validation result of the two models, VT-Unet outperforms 3D-Unet. This could probably because the VT-Unet model could processes the volumetric data in its entirety, and fully encoding the interactions between slices. VT-Unet model is built on Transformers and is superior in its ability of dealing with global features. These design elements might contribute to achieving a better segmentation performance".

"尚需完成的任务 Work to do:"

(1) Compare more neural networks and algorithms for 3D tumor segmentation and find a rather better one.
(2) Make improvements on the above method and come up with my algorithms or network based on them.
(3) Try to use only a small part of the dataset and train fewer epochs, but gain a relatively good segmentation result.

存在问题 Problems:

(1) Since 3D MRI dataset is 3D volumetric data, it has high requirements on computing equipment. Training on the dataset will takes a long time when having only one GPU.
(2) Traditional CNN segmentation method achieves worse result than Transformer model.
(3) Need to find a maybe better evaluation method for model performance.

"拟采取的办法 Solutions:"

(1) Try to use pre-trained model or light-weighted transformer model for training due to limited computational resources.
(2) Try to extract the most valuable subregion of the data volumes and feed them for model training.
(3) Maybe I can set a threshold to select the valuable subregion of the data volume for training.
(4) Try other evaluation method other than Dice score.

"论文结构 Structure of the final report:"

- Abstract
- Introduction
- Related work
- Methodology
- Experiments
- Results
- Conclusion
- Reference

"北京邮电大学 本科毕业设计（论文）教师指导记录表"
"Project Supervision Log"

学院 School	International School	**专业 Programme**	**Internet of Things Engineering**		
姓 Family name	Dong	**名 First Name**	Wanqi		
BUPT 学号 BUPT number	2018213196	**QM 学号 QM number**	190019048	**班级 Class**	2018215121
论文题目 Project Title	An algorithm for 3D MRI brain tumor segmentation				

"Please record supervision log using the format below:

Date: dd-mm-yyyy
Supervision type: face-to-face meeting/online meeting/email/other (please specify)
Summary:"

[These are examples, please DELETE them and add yours. DELETE this line.]

"Date: 10-11-2021
Supervision type: online meeting
Summary: discussed the project specification"

"Date: 21-11-2021
Supervision type: online meeting
Summary: discussed the specific work to do about this project."

Date: 05-01-2022
Supervision type: online meeting
Summary: made a presentation about the work done in the previous months and got some suggestions from the supervisor.

Date: 13-04-2022
Supervision type: online meeting
Summary: made a presentation about the mock-viva and the got advice from the supervisor about the final thesis.

Risk and environmental impact assessment

"Events that prevent the successful completion of the project:"

Events	L(Likelyhood)	C(How bad)	R(Risk)	Solution
Hospitals might not apply automatic diagnosis systems in actual brain tumour diagnosis.	3	2	6	Use the automatic brain tumor segmentation system only as an auxiliary tool to assist doctors in diagnosis, rather than the main diagnostic tool.

"Events that cause potential harm to people and /or animals:"

Events	L(Likelyhood)	C(How bad)	R(Risk)	Solution
Automated brain tumor segmentation system may produce some errors that have bad impact on patients	2	4	8	Continuously improve the accuracy of automatic segmentation systems; Use the system as an auxiliary tool

"Events that cause potential harm to the environment:"

Events	L(Likelyhood)	C(How bad)	R(Risk)	Solution
Model training requires a lot of GPU resources and consumes energy	2	1	2	Apply the pre-trained model to an automatic tumor segmentation system

″An algorithm for 3D MRI brain tumor segmentation″

"Events that cause potential financial loss to the project or other individuals or organization"

Events	L(Likelyhood)	C(How bad)	R(Risk)	Solution
Training the model requires a lot of GPU resources and cost me a lot of money	2	1	2	Get into a good lab and take advantage of public GPUs
Errors generated by the automatic segmentation system may cause economic losses and compensation to the institutions or personnel who use the system	2	2	4	Continuously improve the accuracy and metrics of tumor automatic segmentation system