MACAU UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Computer Science and Engineering

Faculty of Innovation Engineering

Senior Thesis for the Degree of Bachelor of Science

Title: Tumor Detection in Brain MRI Images Based on the U-Net++

Student Name : Xiangyu Gao

Student No. : 19098535-I011-0055

Supervisor : Xiaolin Tian

April, 2023

澳門科技大學

創新工程學院

計算機科學與工程學院

理學學士學位畢業論文

論文題目: 基於 U-Net++的腦部核磁共振圖像中的腫瘤检测

姓　　名 : 高翔宇

學　　號 : 19098535-I011-0055

指導老師 : 田小林

2023 年 4 月

Abstract

Brain tumor is a large number of abnormal cells growing in the brain. It is not good for human healthy life, only bad. Serious malignant tumors have a great chance to cause disability or even death of human beings. In the medical field, nuclear magnetic resonance (NMR) is often used for diagnosis. The doctor can analyze and diagnose the patient's illness through the abnormal state in the MRI image.

In recent years, it is more and more common and effective to use computer programs to automatically segment and assist doctors to judge and analyze various medical images. In previous computer-aided projects, U-Net networks were commonly used to segment brain tumors. In this paper, the author conducted segmentation experiments on brain MRI images using U-Net++networks for the first time and compared the results with U-Net networks. The dataset used is sourced from the Kaggle website and contains approximately 3900 brain MRI images and annotations. After some data processing, the recognition rate exceeded 82%, which is a significant improvement compared to the 78% U-Net network, which also proves the potential of this network in this task.

Keywords: Brain Tumor; Magnetic Resonance Image; Image Segmentation; Deep Learning; U-Net++

摘要

脑肿瘤是大脑中生长的大量异常细胞，對人類的健康生活並無益處，只有壞處，嚴重的惡性腫瘤還有很大的概率會造成人類的殘疾甚至死亡。在醫學領域，常常使用核磁共振技術來診斷。醫生可以通過核磁共振圖像中的異常狀態分析幷診斷病人的得病情況。

而近些年來，使用計算機程序自動分割輔助醫生來判斷、分析各種醫學成像的情況也越來越常見，並且頗有成效。但是，其效率、準確度還有上升的空間，使用計算機進行輔助分割目前為止仍是一項挑戰。以往的计算机辅助项目通常使用 U-Net 网络分割脑部肿瘤，在這篇论文中，作者首次使用 U-Net++網絡，對腦部 MRI 圖像進行了分割实验，并和 U-Net 网络的结果进行了对比。使用的数据集来源于 kaggle 网站，包含大约三千九百张脑部核磁共振图片和标注。在进行一些数据处理之后，识别率超过了 82%，对比 U-Net 网络的 78%有明显提升，這也證明瞭此網絡在此任務上的潛力。

關鍵詞： 腦瘤；磁共振圖像；圖像分割；深度學習；U-Net++

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background and Motivation

Brain tumors are large numbers of abnormal cells that grow in the brain. In 2016 alone, there were 330 000 cases of brain cancer and 227 000 related deaths worldwide [1]. Thus, Brain tumors have become one of the important diseases endangering human health, and their mortality and disability rates are very high. Because doctors do not have the means to see directly with the naked eye what is going on inside the brain, the only way to determine and determine a tumor is to use MRI. However, on the MRI image, the specific location, size, benign and malignant of the tumor are difficult to judge.

In addition, at present, China's medical industry has the problem of relatively insufficient total quality medical resources and uneven distribution. The diagnostic level of brain tumors varies from region to region and at different levels. And, even for an experienced doctor, screening brain tumors through MRI images takes at least 20 to 30 minutes, which is very time-consuming and labor-consuming.

Therefore, Automatic detection and identification of tumors is very important. Because deep learning has great advantages and wide applications in automatic recognition and detection, it is often used to deal with medical image segmentation. Due to its good performance and versatility, the U-Net network has received widespread attention and use since it was proposed and has continued to this day[2]. Since then, many researchers have also improved and improved it. The more popular ones are 3D U-Net, U-Net++[3] and so on.

Based on this, the writer decided to use U-Net++ to achieve the Tumor Segmentation in Brain MRI Images.

1.2 Research Method

Various methods are available for brain tumor classification. Among various methods, artificial neural network (ANN), convolutional neural network (CNN) methods, and Fully Convolutional Networks (FCN) methods are widely used. Because

of many advantages of U-Net, more and more U-Net based networks have been designed for image segmentation in recent years. In this article, writer will use a kind of improved U-Net to detect the brain tumor.

1.3 Thesis Organization

Chapter 2 summarizes some previous research results

Chapter 3 describe writer's work to process the dataset

Chapter 4 describe what did the writer do in the experiment of U-Net++

Chapter 5 tells that the experiment of U-Net that the writer did to compare the result with U-Net++

Chapter 6 illustrates the result and comparison of two nets, and a simple analysis for the result.

Chapter 7 concludes the thesis, and makes a plan for the future work.

Chapter 2 Literature Review

2.1 Research Status

In this paper a Brain Cancer Detection and Classification System have been developed with the use of ANN[5]. The image processing techniques such as histogram equalization, image segmentation, image enhancement, and feature extraction have been used. The proposed approach using ANN as a classifier for classification of brain images provides a good classification efficiency as compared to other classifiers. The sensitivity, specificity and accuracy is also improved. The proposed approach is computationally effective and yields good result.

The paper MRI Brain Tumor Image Classification Using a Combined Feature and Image-Based Classifier, DNN model based on ANN is used[6]. DNN is a branch of ANN, on which the number of hidden layers is increased. In the paper, the researchers use least square fitting (LSF) to obtain the vectors of minimizing sum squared error (SSE) and weight. A classifier combining feature and image based classification (CFIC) is selected. And got a good results.

N. Varuna Shree propose a method using PNN to detect the brain tumor[7]. Probabilistic neural network (PNN) is composed of four nodes or layers: input layer, hidden layer, pattern layer and output layer. Researcher use preprocess to remove a noise and smoothen the image, which also results in the improvement of signal-to-noise ratio. And use GLCM for feature extraction. And successfully segmented the tumor.

Long et al. first introduced fully convolutional networks (FCN), while U-Net was introduced by Ronneberger et al. [8]. They both share a key idea: skip connections. In FCN, up-sampled feature maps are summed with feature maps skipped from the encoder, while U-Net concatenates them and add convolutions and non-linearities between each up-sampling step. The skip connections have shown to help recover the full spatial resolution at the network output, making fully convolutional methods suitable for semantic segmentation.

2.2 Methods Proposed

U-Net family is a kind of deep learning network based on CNN. Since its birth, it has been widely used in the field of medical image processing because it is excellent in using small data set training. What's more, it also well known as it's characteristic simple, efficient, easy to understand, easy to build. Later, many researchers improved U-Net and formed network structures such as U-Net++[3], U-Net3+[4]. which together formed the current U-Net family.

Based on the high performance, I choose to use U-Net++ to detect the brain tumor. Theoretically, it could detect the brain tumor similar to the traditional method. What's more, it could also existing some advantages. Comparing to the classical U-Net and find the advantages and disadvantages.

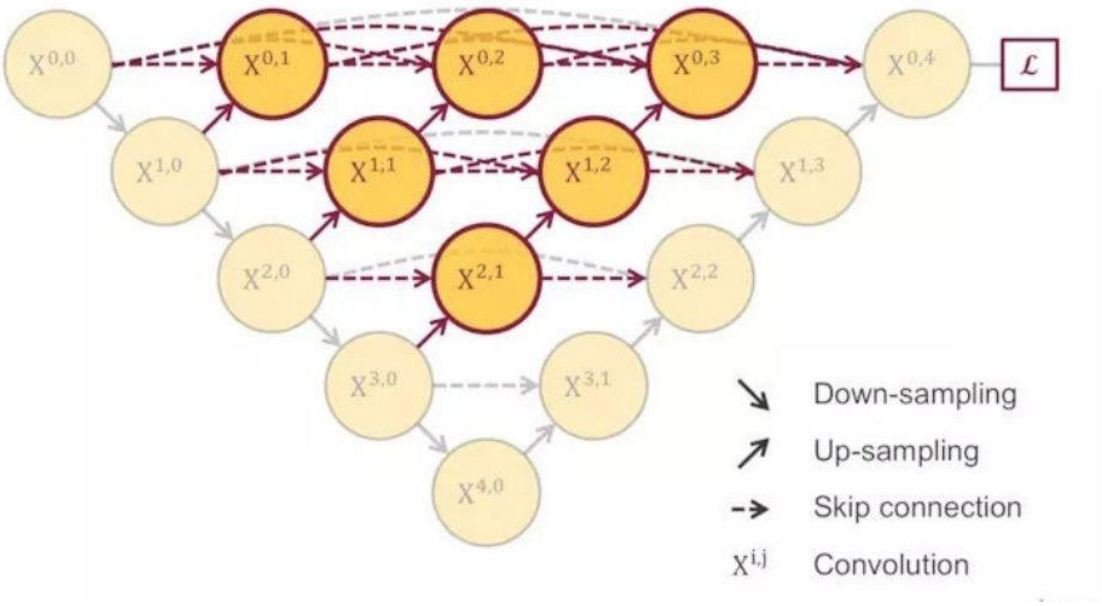


Figure 2-1 Network structure of U-Net++

As a deep learning network inherited from U-Net, U-Net++ is also an encoder-decoder network with skip connection layer. It reserve the structure of the skip connection layer of U-Net.

However, in the paper of U-Net++, they discuss about the times of down-sampling. In their opinion, Although down-sampling will extract the features of the

image for better segmentation, but will it perform good for all images to take four times of down-sampling?

There are two statement about this opinion. The first one is that if there is too much down-sampling, the extracted features are too abstract. This will cause the network to pay more attention to the most part and ignore the details which will cause poor perform on the small objects and the boundary of large objects. The second one is the sensory vision. More down-sampling means the smaller size of the feature image. Using the small size of the feature image to detect image is just like observe the world through a small hole. This will cause cognitive biases to the network.

Then, we can get that more down-sampling will bring the deep level features, and less down-sampling will reserve the shallow level features. We need both of them to create a better performing network. Then, U-Net++ came to the world.

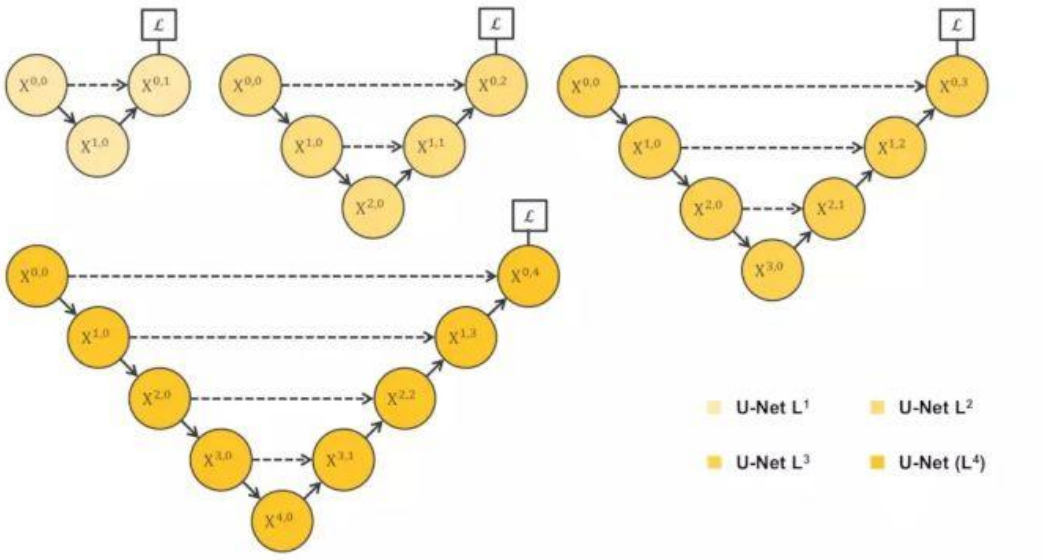


Figure 2-2 The Origin of U-Net++

Just like Figure 2-2, U-Net++ is composed of many different down-sampled U-Net combinations. In this way, all features will be used to segment the image, which will bring better performance.

Chapter 3 Process the Dataset

The project is to run deep learning models on brain MRI dataset. This section gives the introduction of works to process the dataset.

3.1 Brain MRI Dataset

My dataset is downloaded from kaggle (https://www.kaggle.com/)[9], which is a website stand for deep learning and have so many kinds of datasets. It contains 3929 volumes (including training volumes and test volumes). The dataset consists of brain MRI data from over 110 patients from 5 hospitals. All data include brain images of brain tumor patients and tumor location information manually marked. These data are grouped into many folders. But for the U-Net++ network, it is hard on it to training the model while searching for and entering varies of folders. So we need to help the network to separate all the folders. In our imagination, we want to traverse all subfolders, then move all images in all subfolders to the parent folder, and then delete all of the subfolders. We can use python We can achieve this process automatically by writing a Python program.

The principle could use a flow chat as the figure 3-1 shows:

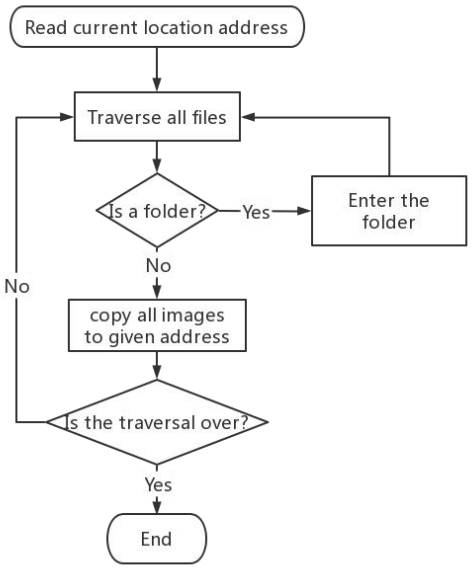
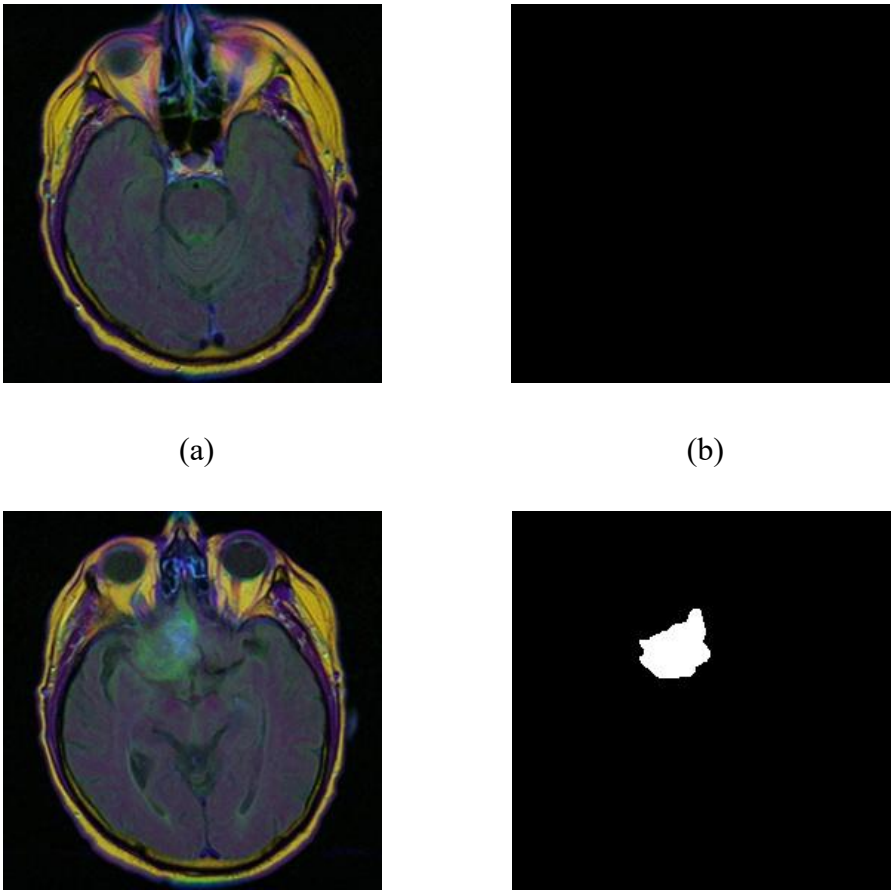


Figure 3-1 Principle of separate the folder

The code for this process of separate the fold is shown in the appendix. After this process, we can see the whole dataset in one folder.

Figure 3-2 shows an example of brain with tumor detected and an example of normal brain. And we can have a look on them.



(a) (b)

(c) (d)

Figure 3-2 Two examples of dataset. Case 1: (a) normal brain image and (b) its mask image. Case 2: (c) brain image with tumor and (d) its mask image

It is worth mentioning that the MRI image itself is a grayscale image, but the dataset has given false colors to the image for our observation. The annotation image is only black and white.

3.2 Convert the Data Format

After we downloaded the dataset from the website, we got the brain image and the mask in .tif image. For the convenience of the U-Net++ network, we need to convert the .tif files into .jpg files and .png files. As for the convenience and generality for the brain image, and the special needs of mask for transparency characteristic for the format, we decided to converse the .tif format into .jpg format (as for the brain image) and .png format (as for the mask). Since we have learned python, we could code a python program to convert the format automatically.

Thanks to the naming standard of the dataset, it is easily to distinguish the brain image and the grand truth image as the table 3-2 shows.

Table 3-1 Each image name in the dataset

serial number	brain image	grand truth image
1	TCGA_CS_4941_19960909_1.tif	TCGA_CS_4941_19960909_1_mask.tif
2	TCGA_CS_4941_19960909_2.tif	TCGA_CS_4941_19960909_2_mask.tif
...	...	...

From table 3-2, we can find that the file name of grand truth image have the sign "_mask" and the name of brain image doesn't have it. Using regular expressions, we can complete both tasks in one step. We could code a python program to separate the brain image and the grand truth image automatically.

The principle of the program could use a flow chat to show:

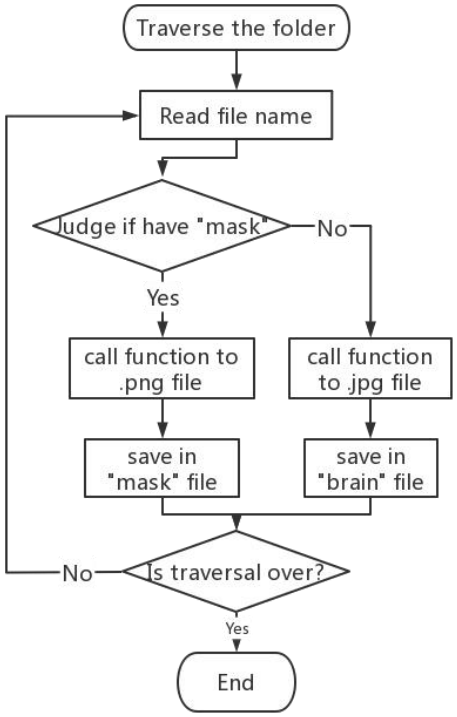


Figure 3-3 Principle of convert two formats in one program

The code for this automatic process is shown in the appendix. After this automatic process, we got the dataset composition by the .jpg format brain image and the .png format mask image. After convert the format, we can go into the data preprocessing part then.

3.3 Data Preprocessing

In the data preprocessing, we need to do two things. Data enhancement and partition of training set and testing set. Thanks to the convenient of a large number of packages of python, we can use functions encapsulated in the packages to do things easily. In package “albumentations”, there are functions such as “RandomRotate90” which means that this function will rotate the image randomly by 90 degrees. The function “Flip” to flip the image randomly. Later, perform the following operations according to the normalized probability selection:

1. Function “HueSaturationValue()” to enhance the image by changing image saturation.

2. Function “RandomBrightness()” to enhance the image by changing the brightness of the image randomly.

3. Function “RandomContrast()” to enhance the image by changing the contrast of the image randomly.

After enhance the data, we need to participate of training set and testing set. We use the “sklearn” package to do this participate. Use the function “train_test_split”, and we can set the parameters like that the allocation ratio. Here we want to set that the training set will count for 80% and the detect set will count for 20%. So, we were set the parameter “train_size” as 0.8. Then, we can achieve it. Because we don’t need other function like random state. So we don’t need to care about other parameters. Then, using this function, we have achieved to participate the dataset. Then, we could go to the experiment part.

Chapter 4 Detect Brain Tumor in MRI by U-Net++

After we have processed the data to what we want to be, we can take these data to do experiment. This chapter tells the experiment we did on the U-Net++ network.

4.1 Experimental Set Up

Considering that training often requires a lot of computation, my own computer is not competent for this task, so I used the cloud computing platform and rented an NVIDIA RTX A5000 GPU with 24G memory which is based on a Linux operation system.

4.1.1 Environment Preparation

At the time of choosing operation system, I chose Ubuntu as the base OS considering that I have used it in the formal courses. So, I got an Ubuntu server. After that, I need to configure a deep learning environment for the server. As the U-Net++ network is based on the Pytorch environment, so I decided to install the whole Pytorch environment for the server. First of all, we must to install the drive for the GPU. CUDA is a computing platform developed specifically for NVIDIA GPU. So we had to install it first. I went to the Tsinghua Mirror Source and find the resource. After that, I use "wget" operation to download the resource from Tsinghua Mirror Source to my server. Then, I used command line to installed the CUDA in the server. After this, we can install miniconda. Conda is a package management system, including many packages which we need. Similar to the process which we install the CUDA. Then, we finally go to install Pytorch. Pytorch is a deep learning framework. Our U-Net++ network uses this framework to run. Using Conda, we can easily install Pytorch just like install a package. Used command "conda install pytorch torchvision torchaudio cudatoolkit =11.0 -c pytorch" and waited some minutes. Then, we have completed the entire environment installation process.

4.1.2 Upload Files

Cause the cloud computing server is a server without GUI and also hard to take data transmission between my laptop and the server. So, I created a git repository and used github for hosting. In this way, after I update the data on my laptop, I just need to use “git push” to update the data in github. Then, in the server, I could use “git clone” to get the data easily. Then, we completed to upload the file to the server.

4.2 Run U-Net++

After configuring the environment, we can formally entering the process of using U-Net++ network. This process can be divided into training part, detection part and verifying part.

4.2.1 Training Part

The training part is to take the images and labels of the training set into the designed U-Net++ network. Where these images will perform encoding, decoding, convolution, pooling, and other operations in the network. This process will be taken many times repeatedly in one training part. Every time is called an epoch. After running one epoch on this network, it means that all training set images had been trained once. And batch size means that in one epoch, how many images will be feed into the network at once. According to these, we can know that there are many epochs of training in one training part. And in each epoch, there are also many rounds of training. Each round of training feed the number of batch size images to the network. After the whole training set are fed into the network, the epoch ends. After each epoch finish, the network will produce a model file which is .pth format. This model file recorded the performance of this network got in this training epoch. In principle, this model could guide the network to determine whether the brain have tumor in a given input image and segment it. But the model has only one training experience usually will not having very ideal results. So, the epoch we set usually will not be one. Besides, the number of epoch and batch size usually effect the result of the model easily. After each epoch finish, the network will

calculate the total loss, validation loss and use detection algorithm to calculate the accuracy. After each epoch, the network will compare the model file produced in the current training and the model file produced in the formal training. If the evaluating indicator like total loss, validation loss and the accuracy. If the current model is better, then the network will just save the current model file, and delete the formal model file. Else the network will not save the current model file. Then, after the entire epoch runs, the last remaining model file will be the best model.

4.2.2 Detection Part

After training part, we have got the best model among we set the times of training which is named “model.pth”. Then, we could use this model to segment the brain tumor image. We copy this model file and saved it into a folder that can be safety kept. Then, we turn to change the detect program. In the program, we first set the position of the model file. And we also need to set the position of the detect set. The detect set have 28 images including both health brain and brain with tumor which is independent from the training set.

Later we could use the U-Net++ network and the best model “model.pth” to detect the brain tumor in the detect set.

4.2.3 Calculating Part

After finish the detection part, we could went to verify the accuracy of the detection. Because the detection part will detect the size and the position of tumor, and the result will be achieved by producing a mask image. So, there will be two masks of a brain image after the detection part. One is produced by the detection part, and the other is manually marked. By comparing the mask produced by the detection part with the mask manually marked, we could find the accuracy of the detection part. We usually use a formula to calculate the score to represent the accuracy. The formula is IoU. IoU will calculate the similarity between the mask produced by detection part and ground truth for images in the detect set. The formula of IoU is shown as following.

$$\text{IoU } Score = \frac{|A \cap B|}{|A \cup B|} \qquad (1)$$

In the formula 1, A and B stands for the detection mask and the grand truth. From this formula we can see that, the more similar the detection mask and the grand truth is, the more score the detection part have. According to this, we could use the score to reflect the accuracy of the detection part.

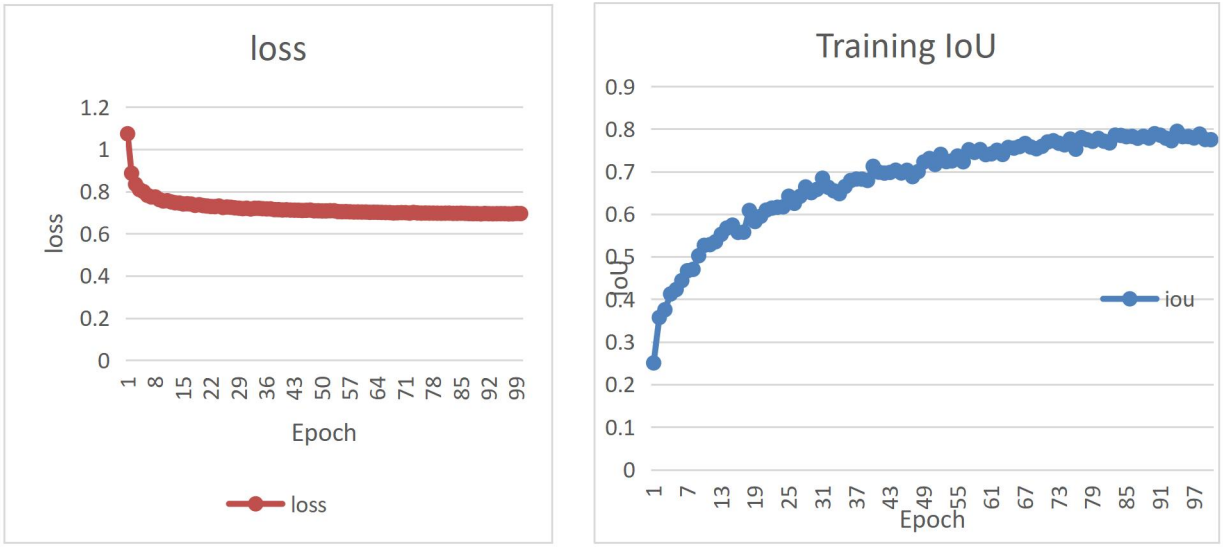
4.3 Experiment Result

After the introduction is completed, we went to run the program. At the beginning, we use the default parameters for training. The default parameter is that epoch is 100 and the batch size is 8. In the way what we were talking about in the 4.2, we can get the training result. For the convenience and visually review the training process, we have the program automatically recorded some important data in each epoch while training. After this training part, we got these data in a table called "log.csv". It recorded the value of loss function and the test IoU of the model after each epoch. A part of the table is shown below:

Table 4-1 First training result

Epoch	loss	IoU
...	...	...
32	0.719854361	0.654668653
33	0.719933243	0.648025542
34	0.718042903	0.664740573
...	...	...

For the consideration of more visually, we plot the entire data sheet into graphs. In the figure 4-1 we can see these graphs.

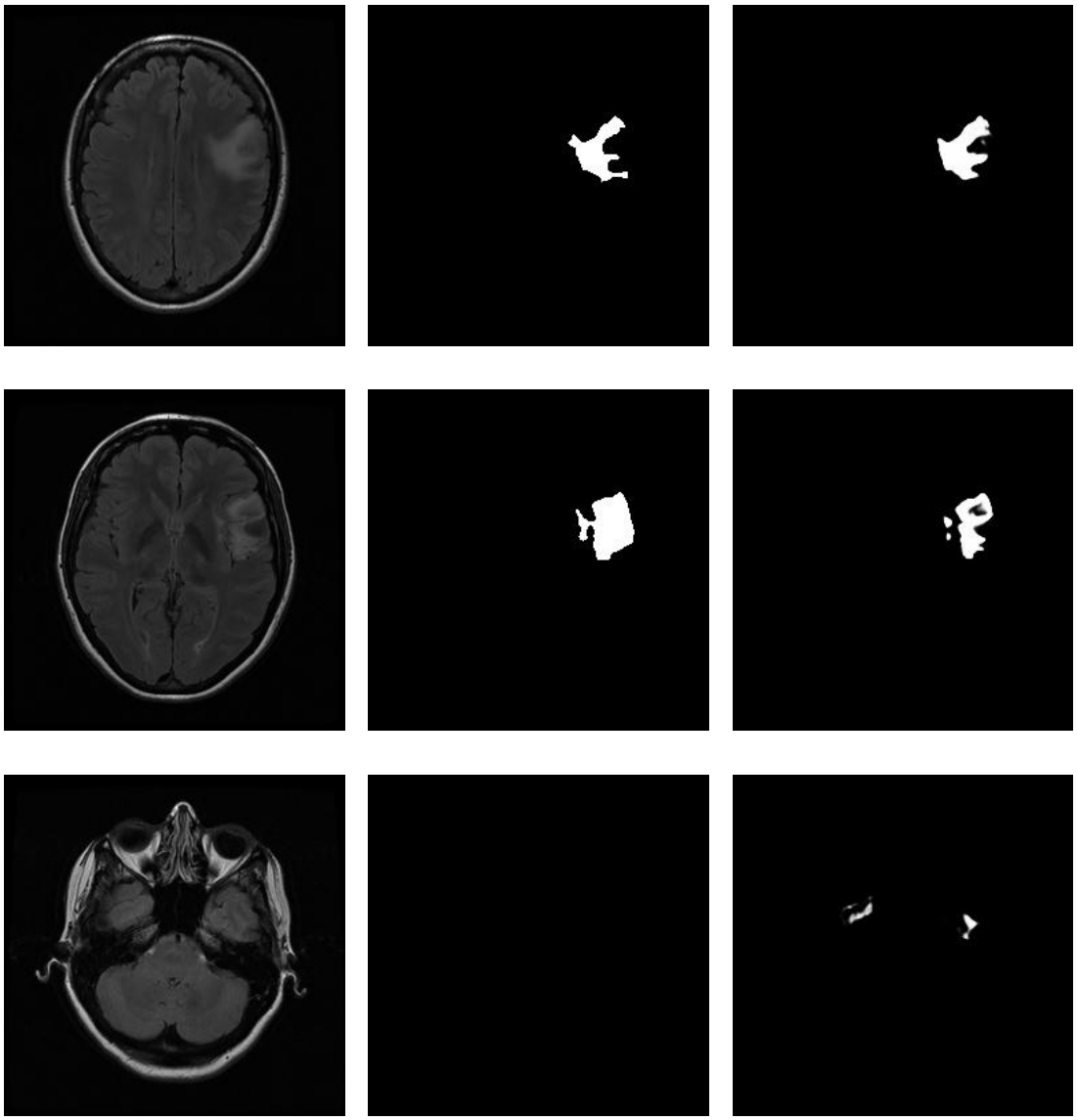


(a) (b)

Figure 4-1 First training result data

By observing the image of loss, we can see that after 100 epoch, the value of loss is tend to gentle and won't be slump. So, we can say that the learning process was basically complete after 100 epoch. So, the parameter of epoch we set don't need to modify.

Then, after the training part, we could went to the detect part. Using the detect set as we were talking before and the way said in 4.2. Run the program, and we could see the output after detection. The result is shown in figure 4-2.



(a) (b) (c)

Figure 4-2 First training result image. (a) detection input image (b) its grand truth (c) detection output

Form figure 4-2, we can see that the network has segment the tumor relatively accurately. But in some details, it is still not very perfect. So we decided to find a way to improve the result.

Considered that we have discussed that after 100 epoch of training, the learning is basically complete, so we can't improve the result by raising the epoch. The way we can do is to find the best batch size that this network for this project.

By fixing the epoch value and changing different batch sizes, we obtained different data. The data is as the Table 4-2 shows:

Table 4-2 IoU in different epochs

Epoch	8	12	16	20	24
IoU	0.76086864	0.813450388	0.803030041	0.798776163	0.798478393
Epoch	10	14	28	32	48
IoU	0.794410969	0.814897796	0.807308037	0.806128523	0.79081275

It is quite hard to find the law of the IoU between different epochs through this table.

So, for the convenience to see and find the law of the IoU between different epochs, I used Matlab to do curve fitting. The result is show as Figure 4-3.

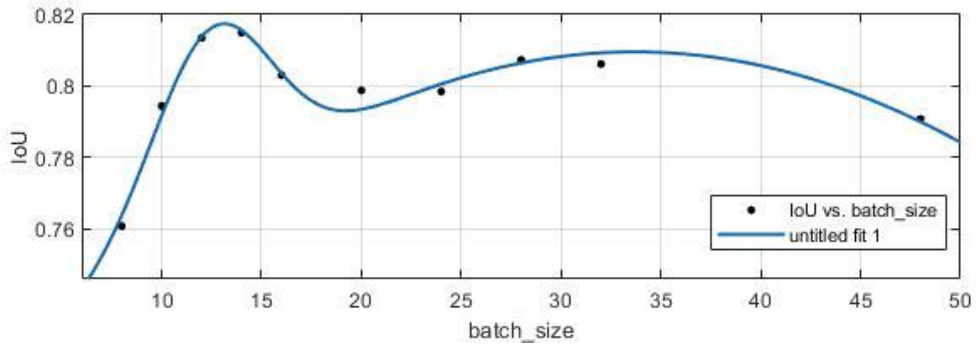


Figure 4-3 Curve fitting result

From Figure 4-3, we can see that the curve is such basically fit for the data points. And we could see that the network have the best IoU performance at the batch size of 14.

After found the best batch size of this project, we could set the batch size as 14. And rise the value of epoch to find the ultimate performance. We set the epoch as 300, and repeat to take the experiment of 4.2. After the training part, we also plotted the data into figures as what we did in 4.3. The figure is shows as below:

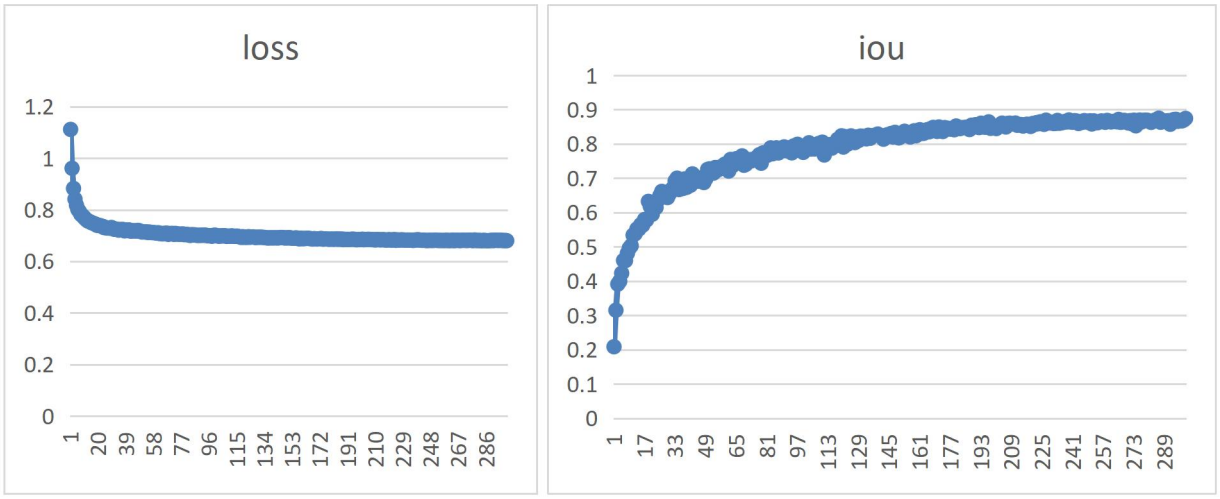
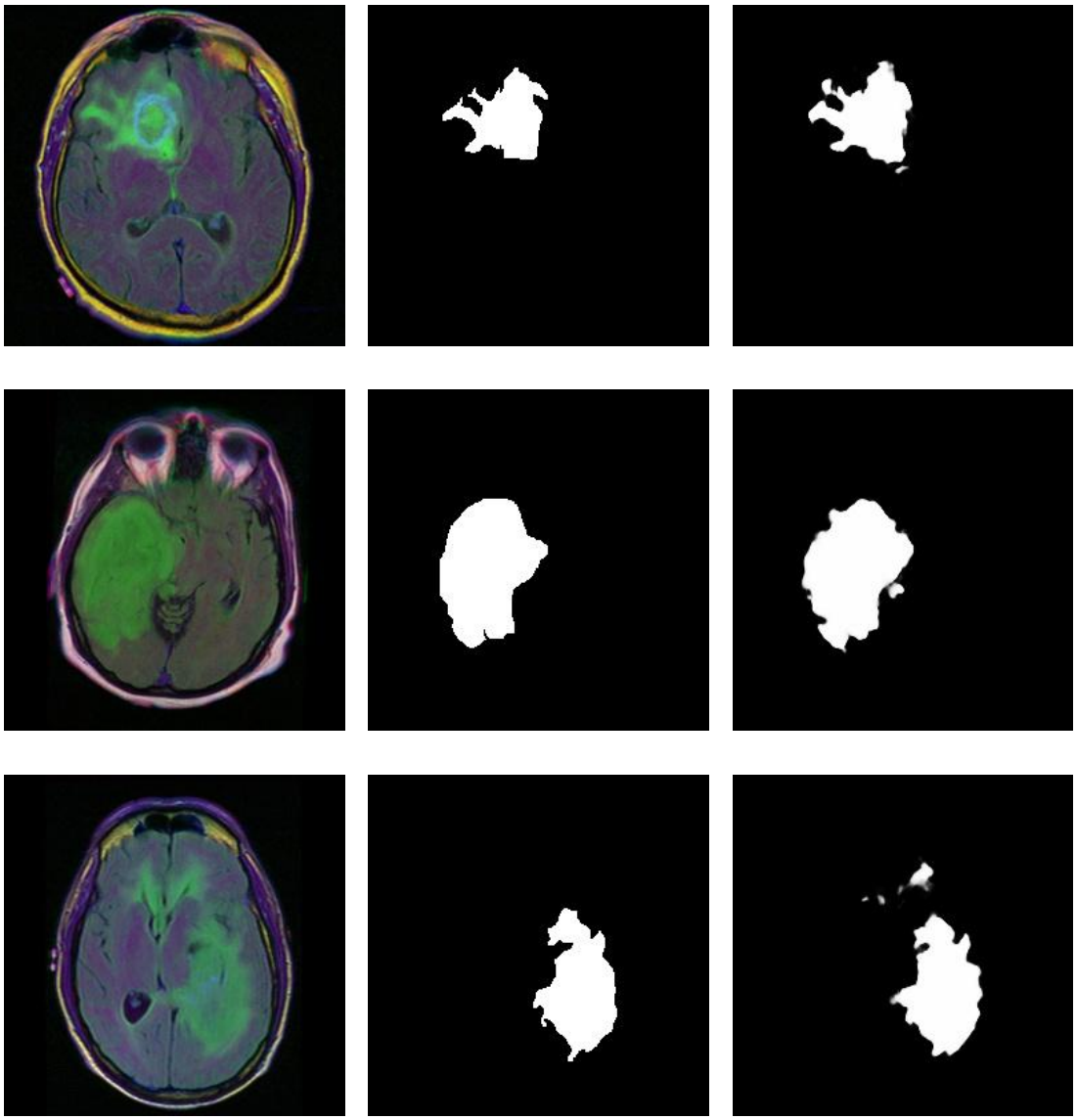


Figure 4-4 Best training result of U-Net++

Comparing Figure 4-4 with Figure 4-1, we can obviously salt that the training result have improved. The training IoU is very goes to 0.9, which was just around 0.8 before. Then, after the training part, we could went to the detect part. Using the detect set as we were talking before and the way said in 4.2. Run the program, and we could see the output after detection. The result is shown in figure 4-5.



(a) (b) (c)

Figure 4-5 Best training result image. (a) detection input image (b) its grand truth (c) detection output

From Figure 4-5, we can see that the recognition effect has indeed improved, the recognition effect is basically good. So the parameter adjustment was successful. Although there are really still some small differences compared to ground truth.

4.4 Experiment of U-Net

To compare performance of U-Net++ and its basis U-Net, I also decided to run U-Net network using the same machine and the same brain MRI dataset. Then, we can obtain them and compare them.

4.4.1 Preparation

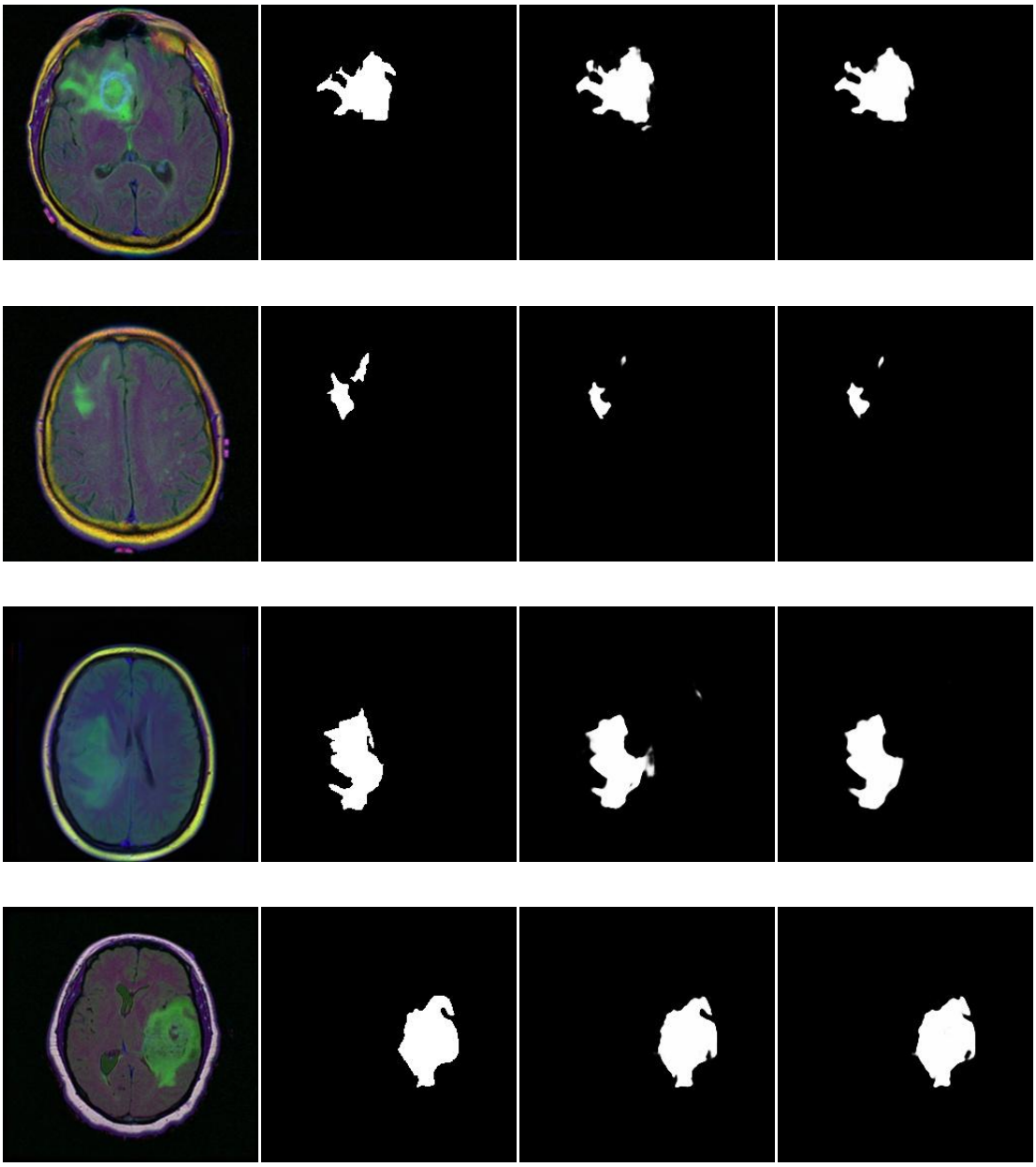
U-Net, as one of the most popular network in the image processing area, has been quoted thousands of times after it was proposed in 2015. It is very easy to find the network model in the website. So, we download the network file without any effort. Thanks to the modular design of the program of the U-Net++ project which I used, I could load the U-Net network module into the U-Net++ project and replace the U-Net++ network module at the time when the project loading network module. Then, we can ensure that other conditions remain unchanged, especially the data loader and the data enhance part. Other wise, there also some adaptability issues in the actual experiment process. Considering that these are all minor issues of changing parameters and variables, I won't go into too much detail here. Then, we have complete the construction of U-Net project, using the U-Net++ project.

4.4.2 Run U-Net

I also use the same server which I have mentioned in the 4.1. Because the changes in the project are minimal, so it doesn’t need to change the environment. Then environment set in the 4.1 was still applicable. Subject to the lack of time, the U-Net network trained only once, and using the default parameter. But we can still basically see the difference between the two networks from the results. Using the way of Chapter 4, we could begin the experiment.

4.5 Result Comparison Between Two Networks

The result images are show as Figure 6-1, and the scores of IoU are shown in the Table 6-1.



(a) (b) (c) (d)

Figure 6-1 Comparison between two nets(a) Brain MRI image (b) its ground truth (c) segmentation of U-Net++ (d) segmentation of U-Net

Table 6-1 IoU score of two networks

Network	U-Net	U-Net++
IoU score	0.781698364	0.824034756

From Table 6-1, it is obviously see that the overall detection result of U-Net++ is better than U-Net. So, it has better performance on segmentation.

The image result in Figure 6-1 also prove this theory. From Figure 6-2, We can see that in some details, such as some gap position, the result of U-Net++ is better which in U-Net. And overall, the segmentation of U-Net++ is definitely a bit better than the segmentation of U-Net.

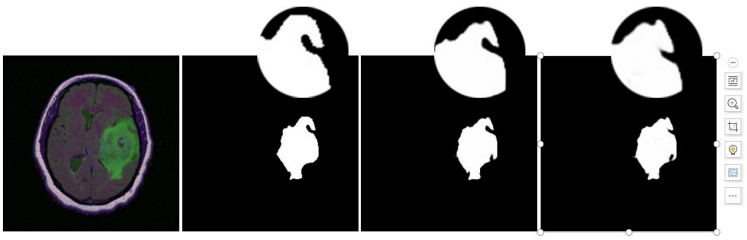


Figure 6-2 Detail for part of figure 6-1 (a) Brain MRI image (b) its ground truth (c) segmentation of U-Net++ (d) segmentation of U-Net

This proves that due to the addition of a skip connection layer into the U-Net network, the U-Net++ network has indeed increased learning power, resulting in better segmentation performance.

Chapter 5 Conclusion

5.1 Conclusion

In this work, the author created a project to help doctors to segment the brain tumor on the brain MRI images.

The author innovatively applies U-Net++ network for the detection of brain tumors, which is created to segment the nuclear. The author used the brain tumor MRI dataset downloaded in the Kaggle. After convert the data format and some data preprocess, and some location adaption, the project was constructed completely.

Then, the author went to determine the parameter. The author has attempted to set the batch size to 8,10,12,14,16, 20, 24, 28, 32, 48. And tried to do curve fitting for these data point. After found that the result is best at the batch size is 14, the author fasten the batch size to be 14 then. Afterwards, the author rose the epoch to do the formal training for the better performance.

To show if the result of U-Net++ is improving, the author also worked on U-Net, from which U-Net++ is based. In order to fix variables, the same data partitioning and data processing are also used on U-Net. After that, experiment stated and the result is obtained.

Comparing to the result of U-Net++ and U-Net, we could see that the result of U-Net++ is a little better and the score of IoU is larger. This approves that the improvement on U-Net++ is exist. Besides, we got a better performance project of brain tumor segmentation, comparing with the U-Net project.

5.2 Future work

In this project, the author only focus on a part of the parameter. There are also some parameters which probably influence the result of the segment. What's more, we can see that there are many cases of improvement on U-Net. Such as changing the encoder, adding some attention mechanism and so on. This causes the appearance of

many network like “TransUNet” , “AttentionUNet” and so on. So why can’t I add this mechanism or changing the encoder and other modular based on U-Net++. Perhaps it will bring us good performance on the medical image process. This is what I will do in the future.

References

[1] “Global, regional, and national burden of brain and other CNS cancer,1990–2016: a systematic analysis for the Global Burden of Disease Study 2016”, Neurology

[2] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” (2015): n. pag. Print.

[3] Zhou, Zongwei et al. “UNet++: A Nested U-Net Architecture for Medical Image Segmentation.” (2018): n. pag. Print.

[4] Huang, Huimin et al. “UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation.” ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020. 1055–1059.

[5] Rajeshwar Nalbalwar Umakant Majhi Raj Patil Prof.Sudhanshu Gonge 2014 Detection of Brain Tumor by using ANN International Journal of Research in Advent Technology 2.

[6] Veeramuthu, A et al. “MRI Brain Tumor Image Classification Using a Combined Feature and Image-Based Classifier.” Frontiers in psychology 13 (2022): 848784–848784. Web.

[7] Varuna Shree, N., and T. N. R. Kumar. “Identification and Classification of Brain Tumor MRI Images with Feature Extraction Using DWT and Probabilistic Neural Network.” Brain informatics 5.1 (2018): 23–30. Web.

[8] Long J, Shelhamer E, and Darrell T Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[9] The world's largest online platform for data analysis. https://www.kaggle.com . Web.

Appendix

1. Code to classify

```
import os
from os import path
import re
import shutil


def scaner_file(url):
    file = os.listdir(url)
    for f in file:
        real_url = path.join(url, f)
        if path.isfile(real_url):
            file_url = path.abspath(real_url)
            print(file_url)

            # 正则表达式判断是否为 mask 图片
            target = r'(mask)'
            temp = re.findall(target, file_url)
            if len(temp) == 0:
                shutil.copy(file_url, "E:\数据集\脑瘤 2\mri")
            else:
                shutil.copy(file_url, "E:\数据集\脑瘤 2\mask")

            # 如果是文件，则进正则判断
        elif path.isdir(real_url):
            # 如果是目录，则是地柜调研自定义函数 scaner_file (url)进行多次
            scaner_file(real_url)
        else:

            print("其他情况")
            pass

def main():
    scaner_file("E:\数据集\脑瘤 2")

if __name__ == '__main__' :
    main()
import os
from os import path
import re
import shutil


def scaner_file(url):
    file = os.listdir(url)
```

```
    for f in file:
        real_url = path.join(url, f)
        if path.isfile(real_url):
            file_url = path.abspath(real_url)
            print(file_url)

            # 正则表达式判断是否为 mask 图片
            target = r'(mask)'
            temp = re.findall(target, file_url)
            if len(temp) == 0:
                shutil.copy(file_url, "E:\数据集\脑瘤 2\mri")
            else:
                shutil.copy(file_url, "E:\数据集\脑瘤 2\mask")

            # 如果是文件，则进正则判断
        elif path.isdir(real_url):
            # 如果是目录，则是地柜调研自定义函数 scaner_file (url)进行多次
            scaner_file(real_url)
        else:

            print("其他情况")
            pass

def main():
    scaner_file("E:\数据集\脑瘤 2")

if __name__ == '__main__' :
    main()
```

2.Code to converse the format

```
import os
import cv2 as cv2

imagesDirectory = r"E:\dataset\brain_tumor\label"  # tiff 图片所在文件夹路径
distDirectory = r"E:\dataset\brain_tumor\label_png"

print(distDirectory)

for imageName in os.listdir(imagesDirectory):
    print("imageName", imageName)
    imagePath = os.path.join(imagesDirectory, imageName)
    print("imagePath", imagePath)
    img = cv2.imread(imagePath)
    try:
        img.shape
    except:
```

```
17.         print('读取图片失败')
18.         break
19.
20.     print("imageName.split('.')[0]", imageName.split('.')[0])
21.     distImagePath = os.path.join(distDirectory, imageName.split('.')[0] + '.pn
    g')  # 更改图像后缀为.jpg，并保证与原图像同名
22.     print("distImagePath", distImagePath)
23.     cv2.imwrite(distImagePath, img)
```

Resume

1. Resume

Name: GAO XIANGYU

Gender: Male

Email: dd201178@163.com

Education:

2016 ~ 2019, High School, ShiJiaZhuang JingYing Middle School.

2019 ~ 2023, Macao University of Science and Technology.

Awards:

2019, Excellent Scholarship

Work experience:

Industrial Bank Hebei Branch, ShiJiaZhuang, China, Assistant Software Engineer.

Acknowledgements

After implementing the new plan, my heart was both happy and uneasy. I am happy because in the new plan, I can have a whole year to do FYP, which is relatively relaxed. On the other hand, a full year of FYP time represents that we need to complete some more arduous and comprehensive work, and we cannot use the standards of the previous six months of FYP to demand ourselves. For me, this brings even greater challenges.

But at the critical moment, Prof. Tian was by our side. Thanks to Prof. Tian's experience and composure. She helps us determine the topic, sets goals in stages, and provides guidance and encouragement, just like a lighthouse on the sea, with enough attention but not too much intervention. She is both a teacher and a friend.

At the same time, I am also very grateful to every teacher I have met in my four years of university, who has been strict but not strict, which has greatly benefited me.

Finally, I also want to thank my classmates and good friends. They always give me encouragement and help me relieve stress when I am hesitant and confused.

The smooth completion of this project cannot be separated from your efforts. Although my college life is about to come to an end, I want to continue this project, continue to improve, and strive to achieve better results, achieving my goal in the background of the article.