Imperial College London

Imperial data science online summer school, 2022

Group 8 - Polaris

CNN-based Brain Tumour Segmentation Mode

Authors:

Yuyi Yang, Luming Pan, Kai Ma

8^{th} **Aug 2022**

Abstract

This project aims to establish a high-dice CNN-Based brain tumour segmentation model to improve the efficiency and accuracy of diagnosing early tumours. Different data augmentation methods and U-Net models were selected in the project, their effectiveness was compared, and a suitable model and augmentation method were finally selected.

Contents

1 Introduction

1.1 Background Information

Brain tumours are incurable cancers that develop from glial support cells or their progenitor cells. They diffusely invade the normal brain, escape surgery or focal radiation, and are protected from chemotherapeutic drugs by the blood-brain barrier. What is worse is that the presentation of a brain tumour patient can be quite variable[1]. Supervising the presence of tumours could help clinicians diagnose.

However, currently, clinical diagnosis mainly relies on the subjective judgments of medical experts, a process requiring their professional assessment and rich experience. Occasionally, it is difficult for medical experts to reach a consensus on brain tumour image segmentation of the same patient. The traditional segmentation process is based on manual labelling and is a complex process with poor repeatability. To address the above problems, according to David G T Thomas[2], it is critical to developing an effective algorithm to segment tumour sub-regions, which can provide the basis for quantitative image analysis, assistant diagnoses and surgical planning, and even patient survival prediction.

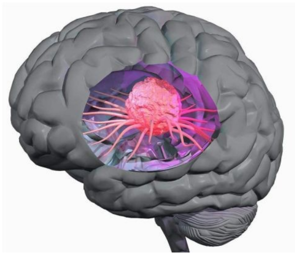


Figure 1: Picture of brain tumour

1.2 Main Experimental Method and Aim

Deep learning, as one of the most currently remarkable machine learning techniques, has achieved great success in many applications such as image analysis, speech recognition and text understanding. Deep learning in image analysis provides an efficient technology for the diagnosis of brain tumours.

The U-Net is a deep learning model widely used for medical images, which achieves very positive performance on very different medical segmentation tasks [3]. Additionally, thanks to the data augmentation, this model just needs very few train images, although the training dataset provided is not large, and takes a reasonable training time.

2 Network Design

2.1 U-Net

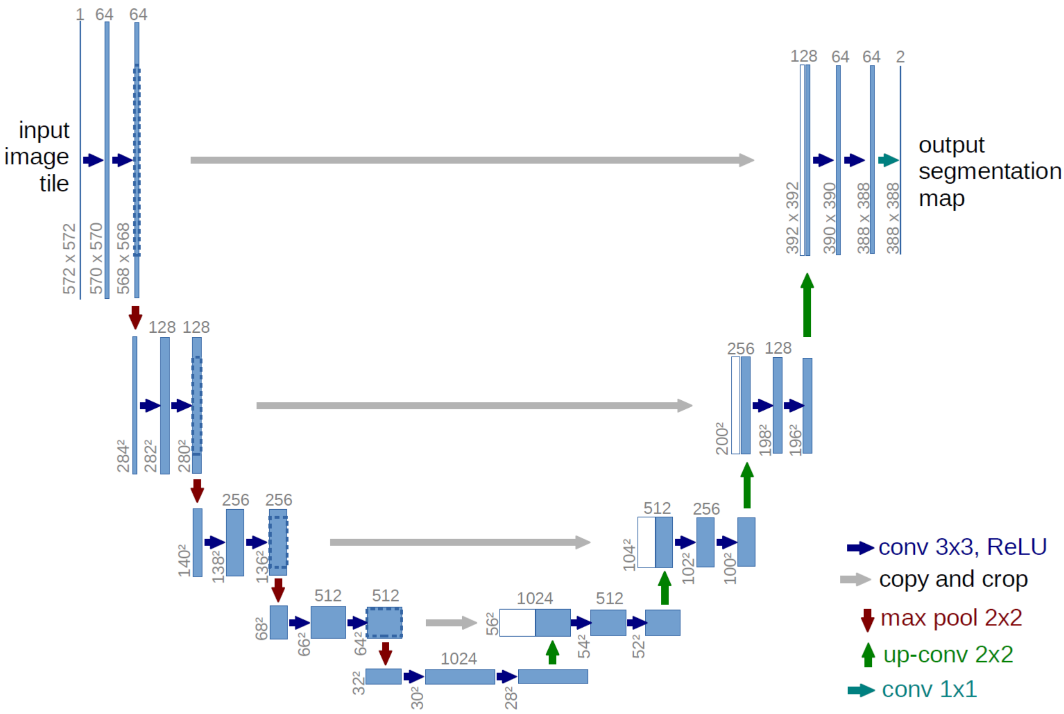


Figure 2: U-Net Architecture

Biomedical images tend to involve the detailed patterns of the object being imaged, such as a brain tumour, and the edges of the target image are variable. Therefore, to cope with the segmentation of the patterned images, Long et al.[4] proposed adopting skip-architecture(U-Net) which bonded the deep decoding layers with the shallow encoding layers to produce accurate segmentation. And this method has good performance on both natural images and biomedical images.

Our network architecture is based on U-Net and consists of an encoding path and a decoding path, as shown in Figure 2. In addition, our group choose the ReLU as the activation function, which is one of the more popular and suitable solutions. The encoded path has 5 convolution blocks. Each one has 2 convolutional layers, the filter size is 3×3, the bidirectional stride is 1, and the rectifier is activated to increase the number of feature maps from 1 to 1024. In the downsampling process, except for the last block, the maximum pooling process with a step of 2×2 is adopted at the end of each block, which reduces the size of the feature map from 240×240 to 15×15. In the up-sampling path, each block starts from a deconvolution layer with a filter size of 3×3 and a step size of 2×2, which doubles the size of the feature map in both directions, but the number of feature maps is reduced by two times, so the size of the feature map is increased from 15×15 to 240×240. In each up-sampling block, two convolutional layers reduce the number of feature maps concatenated by deconvolved feature maps and the number of feature maps from the encoded path. Unlike the original U-Net architecture, we use zero padding to preserve the output dimension of all convolution layers of the down sampled and up sampled paths. Finally, a 1×1 convolutional layer is used to reduce the number of feature maps to two, reflecting foreground segmentation and background segmentation respectively. No fully connected layer is invoked in the network.

2.2 Data Pre-processing and Augmentation

2.2.1 Data Pre-processing

Data Pre-processing refers to the processing of data prior to the main processing to minimize the impact of incompleteness, inconsistency, noise and redundancy of the original data on the experiment. Therefore, in general, pre-processing of experimental data is a very critical step in experiments.

However, since our experimental data are medical images and of high quality without a lot of interference, additionally, the experimental data provided by the project are already the result of data pre-processing, we do not need to perform additional processing on them for the time being.

2.2.2 Data Augmentation

Generalizability is crucial to a deep learning model and it refers to the performance difference of a model when evaluated on the training data versus the testing data.

Data Augmentation is an effective way of improving the generalization capabilities of deep neural networks and can be used as implicit regularization; moreover, it plays a critical role in scenarios in which the number of high-quality ground-truth data is limited, and obtaining new examples is very time-consuming and costly[5]. Therefore, It is the process of processing more representations from the original data without substantially increasing the data, improving the quantity and quality of the original data to approximate the value generated by the larger amount of data. Because the augmented data will represent a more comprehensive set of possible data samples and minimise the distance between the training and validation/testing sets, it is helpful for the model to discriminate statistical noise in the data and reduce model overfitting.

We present seven data enhancement methods here, which are listed below:

- Vertical Flip
- Random Cropping
- Grid Distortion
- Random Rotate 90 degrees

- Elastic Transformation
- Optical Distortion
- Random Brightness Contrast

The original and data-augmented images and corresponding segmentations are shown in Figure 3:

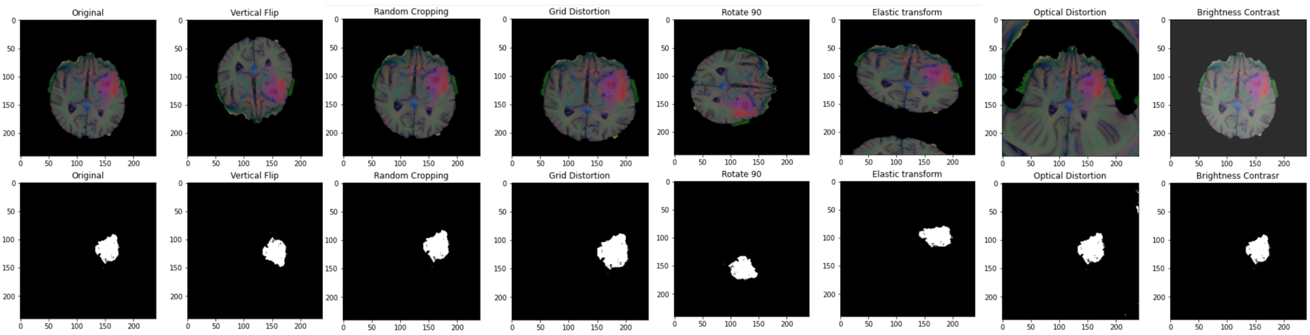


Figure 3: Seven Ways for Augmentation: Vertical Flip, Random Cropping, Grid Distortion, Rotate 90 degrees, Elastic Transform, Optical Distortion, Brightness Contrast

2.3 Metric for the performance of the model

2.3.1 Dice Coefficient

The dice coefficient is a statistic used to assess the similarity of two samples, essentially measuring the overlap of the two samples. It is named after Lee Raymond Dice, and is an ensemble similarity measure function that is usually used to calculate the similarity between two samples (values in the range [0, 1]).

Dice loss is a coefficient related to dice coefficient, and their relationship can be expressed as:

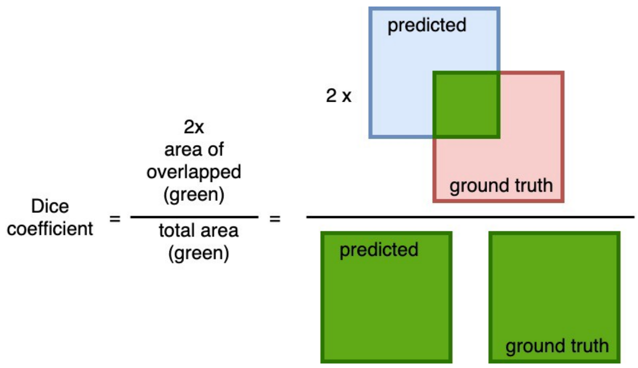


Figure 4: Dice (from google.com)

2.3.2 IoU

IoU coefficient is a statistic showing the overlap ratio of the target window generated by the model and the original marker window. That is, the intersection of Detection Result and Ground Truth divided by their union is the detection accuracy(IoU coefficient).

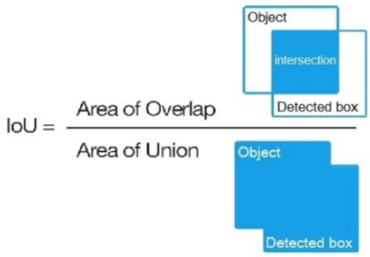


Figure 5: IoU (from google.com)

Judging the coincidence of two images by IoU has the following advantages:

1. It has scale invariance;

2. Satisfy non-negativity;

3. Satisfy symmetry;

But at the same time, IoU also has several obvious shortcomings:

1. If the two images do not intersect, the distance between the two images cannot be compared;

2. It doesn't show how exactly the two images intersect. training times

We use the IoU coefficients and the dice loss of the training and test sets, as well as accuracy to evaluate the efficiency of the current training situation.

2.4 Optimizer

The optimizer is to guide each parameter of the loss function to update the appropriate size in the right direction during the backpropagation process of deep learning, so that the updated parameters keep the loss function value approaching the global minimum.

In this step, we use cosine annealing learning rate as our optimizer, which would linearly increase then exponential decrease. And Figure 5 is the picture of the learning rate: The Cosine annealing algorithm can reduce the learning rate by the cosine function. In the cosine function, as the epoch increases, the cosine value first increases slowly and then decreases rapidly. This mode of descent works well with the learning rate in a very computationally efficient way. Compared with the normal gradient descent algorithm, the Cosine annealing algorithm is more computationally efficient and can effectively avoid the initial oscillation of the model. Thus,

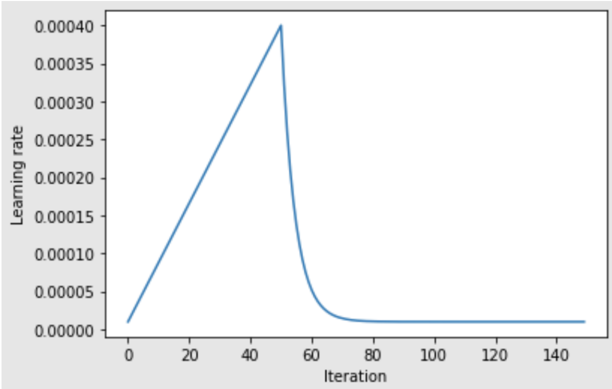


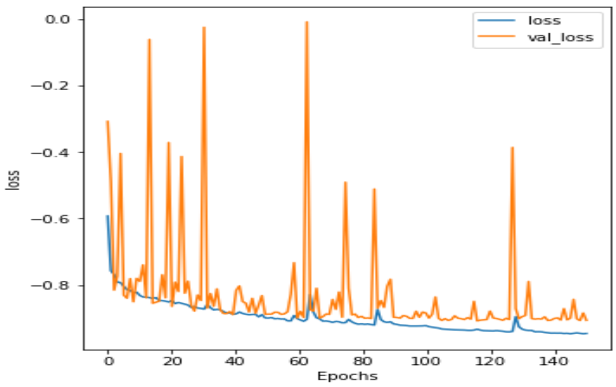
Figure 6: Our Cosine Annealing Learning Rate

using Cosine annealing can increase the computing scale, improve the learning efficiency and get a better learning effect. Due to the above characteristics, we choose the Cosine annealing algorithm to build our training model.

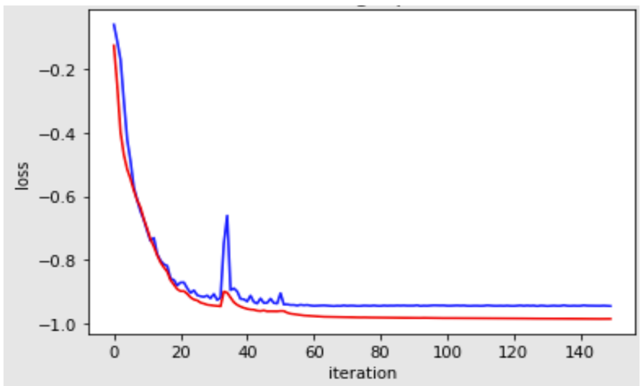
3 Training Results

3.1 Comparison between Different Data Augmentations

By using our initial data augmentation scheme, a huge fluctuation in loss function is found (shown in Figure 6(a)). From the graph, the curve has extremely large and violent fluctuations.



(a) Loss of original augmentation



(b) Loss of second augmentation (red line is the val set)

Figure 7: Picture of Loss Function

Thus, the experiment introduced a second set of data-enhanced models, which only saves Vertical Flip, Random Cropping and Random Rotate. And, Figure 6(b) shows its curve, which is a smoother curve.

3.2 Comparison between Different Epochs

In addition, the experiment compared the effects of different epochs. Initially, we choose **epoch=150** as the original setting, which comes from similar projects found on open source project websites. To highlight the special nature of our experiment, we also use **epoch=30** as a control group to explore the effect of different epochs on the experiment.

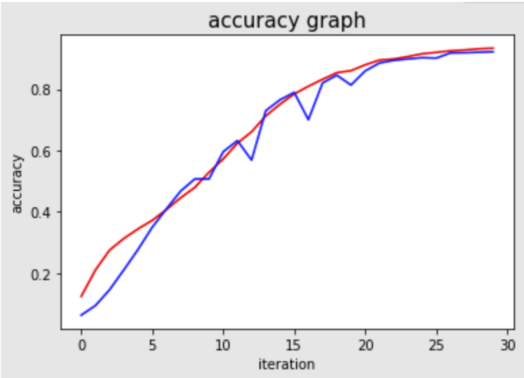


Figure 8: accuracy graph(epoch=30)

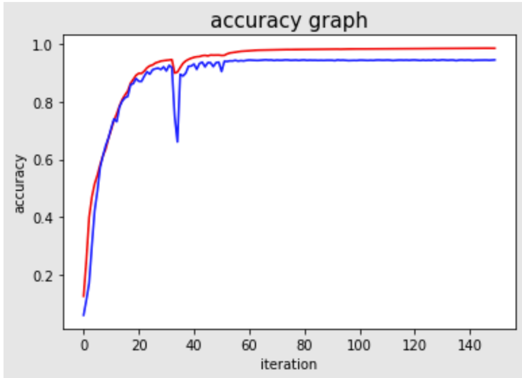


Figure 9: accuracy graph(epoch=150)

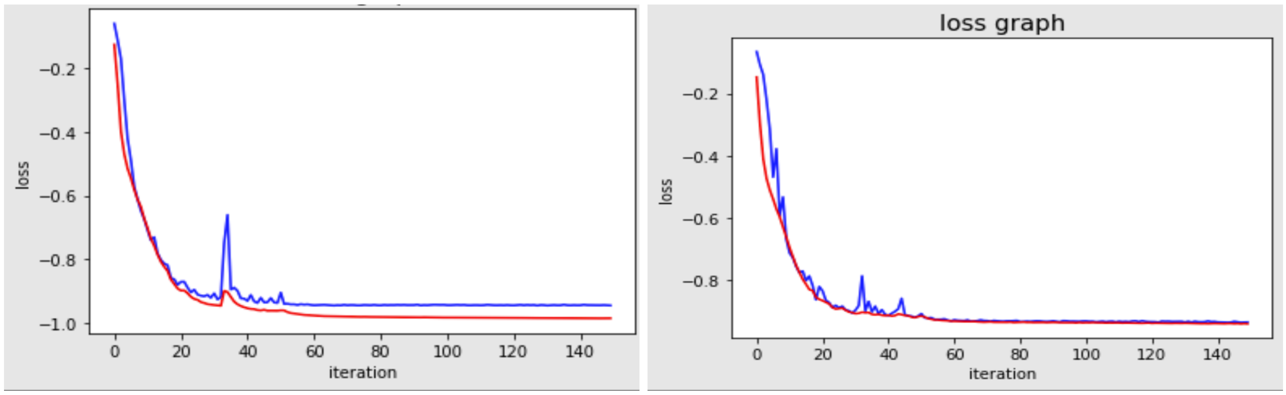
Figures 7 and 8 show the curves under different epochs very clearly, and we can easily find that the correct rate changes slowly after **epoch=30** and hardly changes after **epoch=60**.

4 Discussion

4.1 Different Data Augmentation Methods

In the experiment above shown by Figure7, we can find dramatic fluctuations in it when we use the initial data augmentation scheme, and the absolute value of the value after data augmentation is also less than the value without data augmentation. However, this condition is not normal; normally data augmentation should bring positive results, which can enhance the Dice and IoU. Thus, it is reasonable to speculate that the initial scheme is too aggressive or not applicable to our dataset.

In order to test our conjecture, we trained the model without data augmentation and without changing other conditions by removing the less common or aggressive methods from the original scheme and keeping only Vertical Flip, Random Cropping, Random Rotate 90 degrees, respectively. The results show a significant improvement over the original solution in both cases, which could be found in Figure 10.



(a) Loss without Data Augmentation (red line is the val set) (b) Loss of second augmentation (red line is the val set)

Figure 10: Picture of Loss Function

Based on the above experiments and after collecting relevant information, we hypothesize that the poor performance of the original data augmentation is that the dataset used in the experiments is high quality medical data, distortion and aberration are almost non-existent in the dataset, therefore, if we use the original scheme, there will be a lot of abnormal data in the dataset and lead to data contamination.

Therefore, by comparing the three data processing methods, it can be found that the second data augment can effectively reduce the dice loss coefficient and improve the accuracy. However, the original data augmentation method will not only increase the dice loss coefficient and reduce the accuracy, but also cause the curve to fluctuate violently. According to the results returned by the program, the accuracy of the original data augmentation group also decreased.

4.2 Different Epochs

In the **training results** section, this report shows the effect of the model at different epochs. We find that the curve improves very slowly after a certain stage, so much so that there is no significant difference between the cases of epoch=30 and epoch=150.

For this reason, we decided to reduce the epoch, but considering the characteristics of the learning rate set by the model, and carefully observing the correctness curves in both cases, we decided to set the epoch to 70, so as to have a high correctness rate, IoU while saving a lot of model training time.

5 Conclusion and Results showcase

In conclusion, this report describes some of the design details of UNet and in this experiment the design details of the whole segmentation model and its corresponding design reasons, as well as some problems and findings during its training process, such as the effects of different data enhancements and epochs on the experiment. In this process, by comparing and controlling the variables, the possible causes of the problem and the feasible solution - setting up a suitable data augmentation scheme - are finally proposed. After this, by choosing the appropriate data enhancement scheme and epoch, we successfully built our own CNN-based Brain Tumour Segmentation Model.

The following is a demonstration of some results of the model(epoch=70, use the second data augmentation scheme) trained by our group:

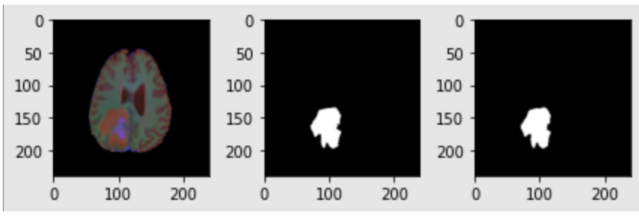


Figure 11: Results 1

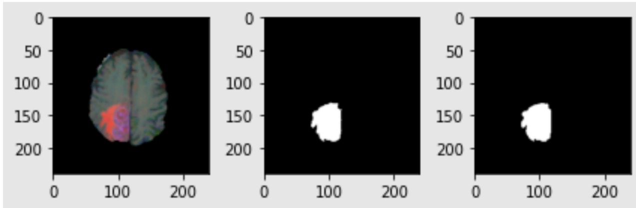


Figure 12: Result 2

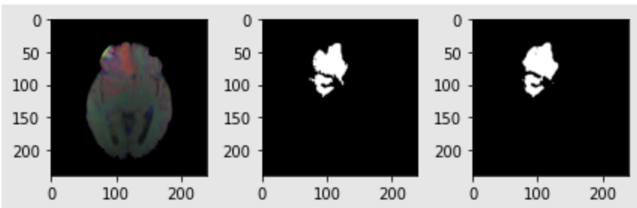


Figure 13: Results 3

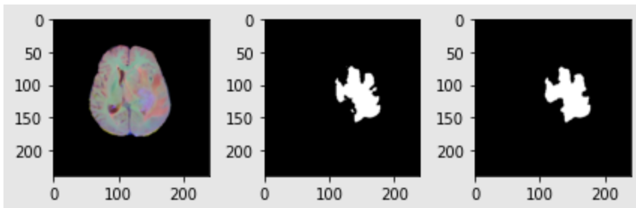


Figure 14: Result 4

References

[1] H. Sontheimer, *Diseases of the Nervous System (Second Edition)*. Academic Press, second edition ed., 2021.

[2] D. G. T. Thomas, *Contemporary Neurology*. Butterworth-Heinemann, 1984.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[5] J. Nalepa, M. Marcinkiewicz, and M. Kawulok, "Data augmentation for brain-tumor segmentation: a review," *Frontiers in computational neuroscience*, vol. 13, p. 83, 2019.