Project 1 Report

刘喆骐 探微书院 0 字班
徐硕 化工系 9 字班
杨乔尹，电机系，2022 级工博班

1. Project description

Project 1: Brain MRI Image Segmentation report

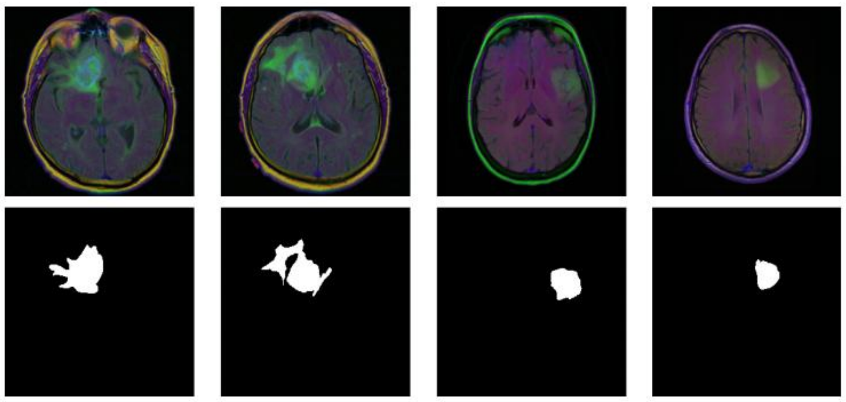


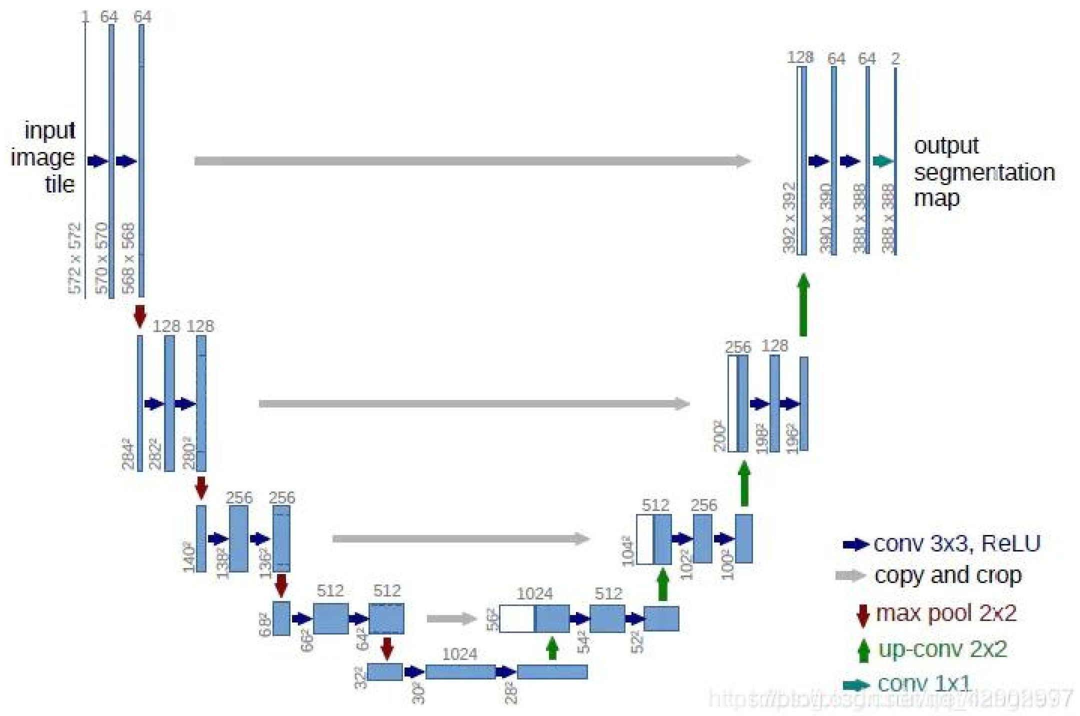
Figure 1: Brain MRI Images (first row) and their corresponding ground-truth masks (second row).

A series of common issues with medical images are

- the image sampling dataset is small;
- the image resolution is usually very high;
- The image slicing resolution is also required to be high.

2. UNET Model

2.1. Model Structure



The Unet model method adopted a U shape with downsampling(encoder), feature exactration, and up sampling(decoder) with features merged as the output. The method solves the issues with medical images indentificaiton.

The Left side shows feature extraction network (encoder) and right shows feature fusion network (decoder)

This network architecture provides the following functions:

High resolution-encoding-low resolution-decoding-high resolution.

The Unet model architecture is shown above.

If the input image size is of 572*572, step size of 1, and a padding size with 0, kernel 3*3. The image is convoluted into a 570*570 feature map.

Another layer of convolution with the same parameters will output a 568*568 feature map.

Output = (feature map size - kernel size+2*Padding size)/Step + 1

2.2. Model description

Code source：https://github.com/mateuszbuda/brain-segmentation-pytorch

2.2.1. 1st Unet model

Implement the unet as it is described earlier, with 4 encodings and 4 decodings.
This unet model is shown in unet.py.

2.2.2. 2nd Unet model

Add another layer just above the bottom layer, so that we encode 5 times and decode 5 times.

This unet model is shown in unet1.py.

2.3. Training results

Optimizer	Unet model	Learning rate	Accuracy	loss	Learning rate pipeline	Epoch
Adamw	Unet -4 layers	0.0001	0.916898	0.198469687075842	No	100
Adam	Unet-5 layers(one layer added)	0.0001	0.913079	0.25330203771591187	No	100
	Unet -4 layers	0.0001	0.917246	0.2015469925744193	No	100
		0.00001	0.903916	0.5202179352442423	No	100
		0.001	0.909318	0.26478990486689974	No	100
		0.00001	0.908328	0.39743986016228083	No	150
		0.0001	0.912947	0.2470121525582813	WarmupCosineSchedule	100

2.4. Disucssion

2.4.1. UNET 1st V.S UNET 2nd

UNET2 with a fifth layer added, which result in a loss=0.25330203771591187，dsc=0.913079. The result does not significantly improve as compared to UNET1 model.

2.4.2. Learning rate and pipeline

The testing added the CosineAnnealingSchedule, and tested different learning rates. The testing results are verified and no obvious results are shown with different learning rates.

With learning rates =0.001，the testing shows a loss=0.26478990486689974，dsc=0.917246.

With learning rates =0.00001, epoch = 100, the testing shows a loss=0.5202179352442423，dsc=0.903916。

With learning rates= 0.00001,epoch = 150, the testing shows a loss=0.39743986016228083，dsc=0.908328。

With learning rates =0.0001 and CosineAnnealingSchedule，the testing shows a loss=0.2470121525582813，dsc=0.912947.

The testing shows that changes in learning rate and the adoption of CosineAnnealingSchedule display neligiable effect on this project.

2.4.3. Generalized Dice Loss

For some reason, the generalized dice loss doesn't work well. It enters a local minimum of around 0.96 and doesn't reduce. And the dsc varies between 0.6 and 0.8, with the best mean dsc=0.815893.

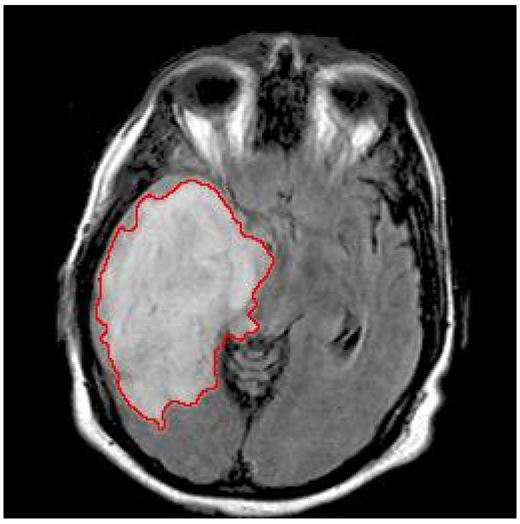
Generalized dice loss can be found in generalized_loss.py.

2.4.4. Predicted images

1. Using Adam

Loss=0.2015469925744193

dsc=0.917246



2. Using AdamW

loss=0.198469687075842，dsc=0.916898