Brain Tumor Segmentation with U-Net

10801128 陳俊鴻

Implementation Details

PL	Python 3.8
API	Tensorflow 2.4.0
Packages	os, numpy, matplotlib, imgaug, tf_Segmentation
GPU	M1
Github	None

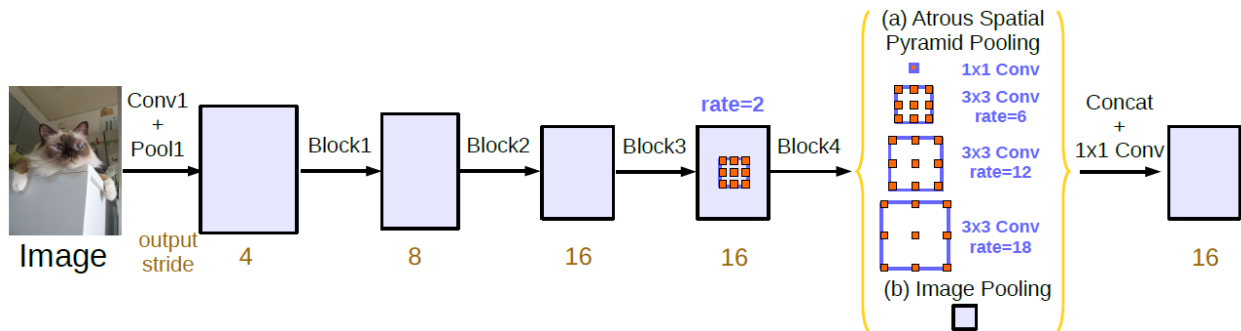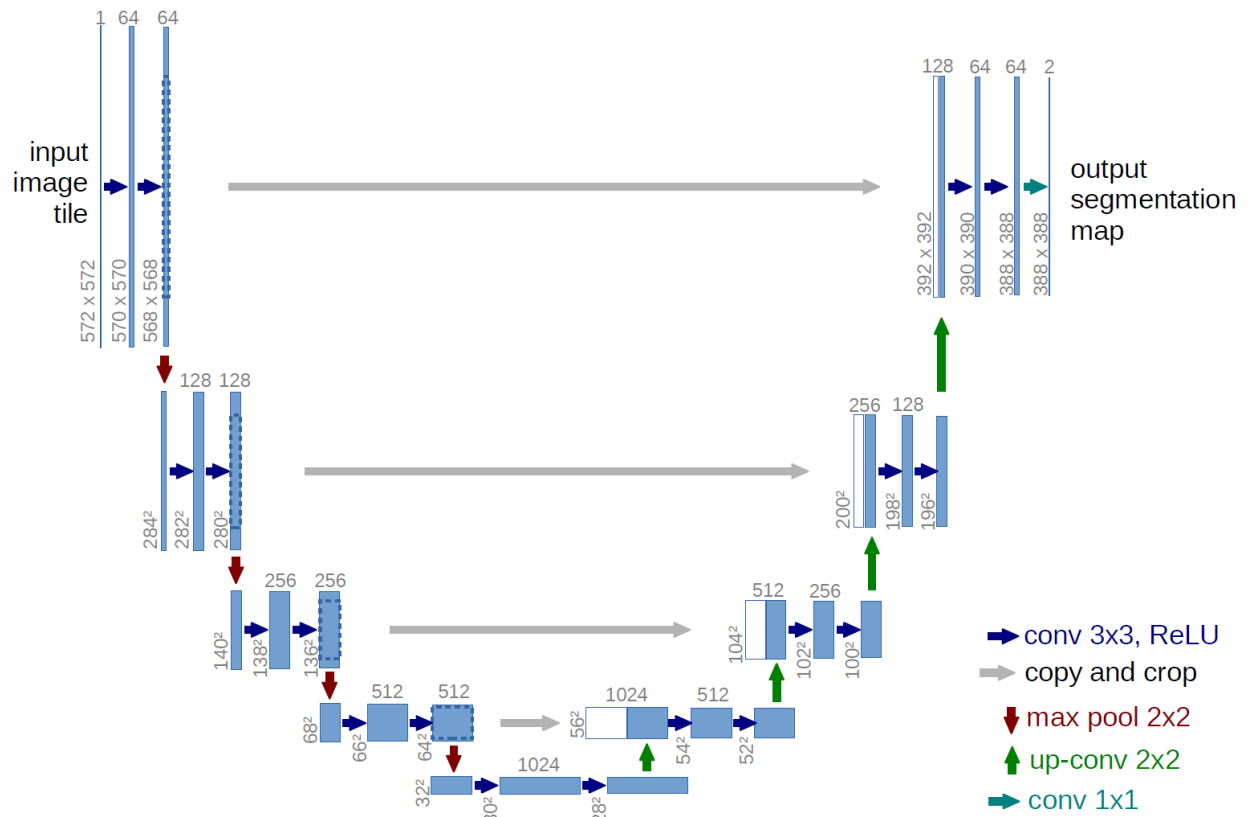## Dataset

Name	2021 Brain Tumor Segmentation Asia Championship Challenge
Content	MRI images and labelled images of 1600 patients
Source	https://aidea-web.tw/topic/a0264109-3fc6-49ea-9e4b-4b0a9433f02f
Size	512x512 pixels, channel number=3
Classes	Brain tumor region, none
Distribution	Train: 1280, validation: 160, test: 160

Experimental Design

U-Net is an encoder-decoder-like model for semantic segmentation task; it contains several stacks of convolution and max pooling layers just as common CNN but also up-convolution layers and the action, copy and crop, like residual. The up-convolution layers and residual-like action help the model increasing the resolution of the output and localizing features in higher resolution layers.

DeepLabv3 is also a model for semantic segmentation task that improves upon DeepLabv2. To implement segmentation, modules were designed to employ atrous convolution in cascade or in parallel. Atrous convolution is quite different from common convolution; it enlarged size of kernel to capture feature instead of contracting. Other details are most identical to DeepLabv2.

input image tile

64 64

1

572 x 572
570 x 570
568 x 568

output segmentation map

128 64 64 2

392 x 392
390 x 390
388 x 388
388 x 388

128 128

284²
282²
280²

256 128

200²
198²
196²

256 256

140²
138²
136²

512 256

104²
102²
100²

512 512

68²
66²
64²

1024 512

56²
54²
52²

1024

32²
30²
28²

→ conv 3x3, ReLU
→ copy and crop
↓ max pool 2x2
↑ up-conv 2x2
→ conv 1x1

Image

Conv1 + Pool1

output stride

Block1 Block2 Block3 rate=2 Block4

4   8   16   16

(a) Atrous Spatial Pyramid Pooling

1x1 Conv
3x3 Conv rate=6
3x3 Conv rate=12
3x3 Conv rate=18

(b) Image Pooling

Concat + 1x1 Conv

16

Modified Parameters

To have an insight into the cross-effect of input size and loss function to U-Net, I implemented the following four trials altering input size and loss function. Otherwise, with the interest in the performance of DeepLabv3, it also be tried to compare with U-Net.

Trial #	1	2	3	4	5
Architecture	U-Net	U-Net	U-Net	U-Net	DeepLabv3
Input size	$416 \times 416 \times 3$	$416 \times 416 \times 3$	$512 \times 512 \times 3$	$512 \times 512 \times 3$	$512 \times 512 \times 3$
Loss function	Dice	Crossentropy	Dice	Crossentropy	Crossentropy
Metrics	Accuracy	Dice coefficient	Accuracy	Dice coefficient	Dice coefficient

All the models were trained with Adam optimizer, 30 epochs, batch size=10, and learning rate=1e-5.

Results

Trial 1:

Confusion metric	Tumor	None	Precision	Recall	F1-score
Tumor	107764	9647	0.0025	0.9178	0.0050
None	4148264	23423285	0.9996	0.8495	0.9185

Trial 2:

Confusion metric	Tumor	None	Precision	Recall	F1-score
Tumor	112538	17462	0.0315	0.8657	0.0608
None	3259183	24099777	0.9993	0.8809	0.9364

Trial 3:

Confusion metric	Tumor	None	Precision	Recall	F1-score
Tumor	25783	196340	0.0336	0.1161	0.0521
None	742701	40978216	0.9952	0.9822	0.9887

Trial 4:

Confusion metric	Tumor	None	Precision	Recall	F1-score
Tumor	207693	6007	0.0216	0.9719	0.0353
None	9394094	32335246	0.9998	0.7749	0.8731

Trial 5:

Confusion metric	Tumor	None	Precision	Recall	F1-score
Tumor	1573	100964	0.0030	0.0153	0.0050
None	528972	27057451	0.9962	0.9808	0.9884

Reference

1. U-Net: Convolutional Networks for Biomedical Image Segmentation. https://arxiv.org/abs/1505.04597

2. Rethinking Atrous Convolution for Semantic Image Segmentation. https://arxiv.org/abs/1706.05587