Homework #1 Report
Brain MRI Segmentation
Medical Image Processing

資工碩一　張凱庭　R10922178

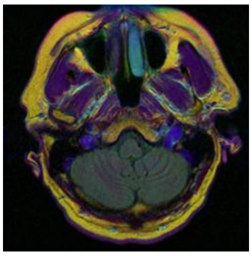
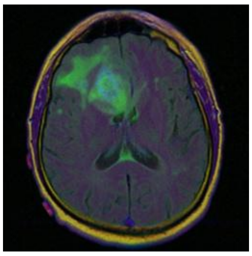
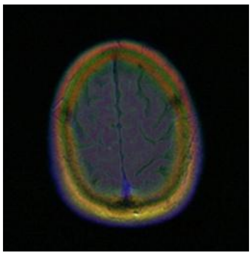
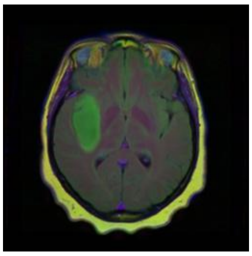
---

1. Dataset: LGG Segmentation Dataset

Dataset Summary:

Number of patients	110
Number of images	3929
Number of images with Tumor	1273
Number of images without Tumor	2556

MRI image visualization:

TCGA_CS_4941_19960909_5	TCGA_CS_4941_19960909_14	TCGA_CS_4941_19960909_20	TCGA_CS_4942_19970222_10
			

Split data into train/validation/test set:

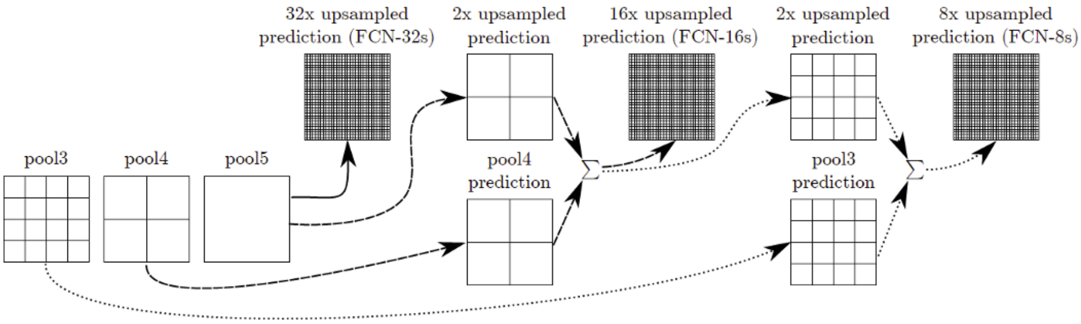
Train	Validation	Test
3005 (75%)	393 (10%)	531 (15%)

2. Model : Pretrained VGG16 with FCN8s

Implementation: Using torchvision's pretrained VGG16 as backbone(or so called "transfer learning"), then replaced fully connected layer(which is for image classification) with the FCN8s model for our semantic segmentation task.

Model architecture and visualization

Model Visualization



Model Architecture

```
Vgg16FCN8(
  (pool1_to_3): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (pool4): Sequential(
    (0): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): ReLU(inplace=True)
    (6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (pool5): Sequential(
    (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): ReLU(inplace=True)
    (6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fcn6): Sequential(
    (0): Conv2d(512, 7, kernel_size=(1, 1), stride=(1, 1))
    (1): ReLU()
  )
  (upsampling2): ConvTranspose2d(7, 7, kernel_size=(4, 4), stride=(2, 2), bias=False)
  (upsampling8): ConvTranspose2d(7, 7, kernel_size=(16, 16), stride=(8, 8), bias=False
  (score_pool3): Conv2d(256, 7, kernel_size=(1, 1), stride=(1, 1))
  (score_pool4): Conv2d(512, 7, kernel_size=(1, 1), stride=(1, 1))
)
```

3. Training

Training details:

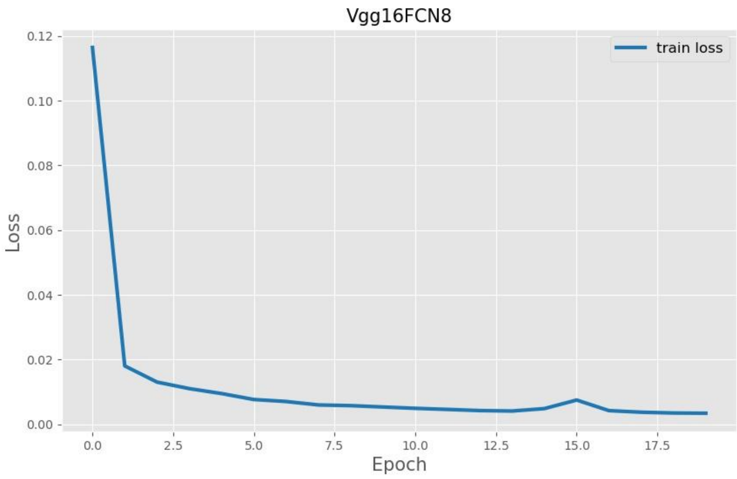
Loss function	Cross entropy loss(pixel wise)
Optimizer	Adam algorithm

Training parameters:

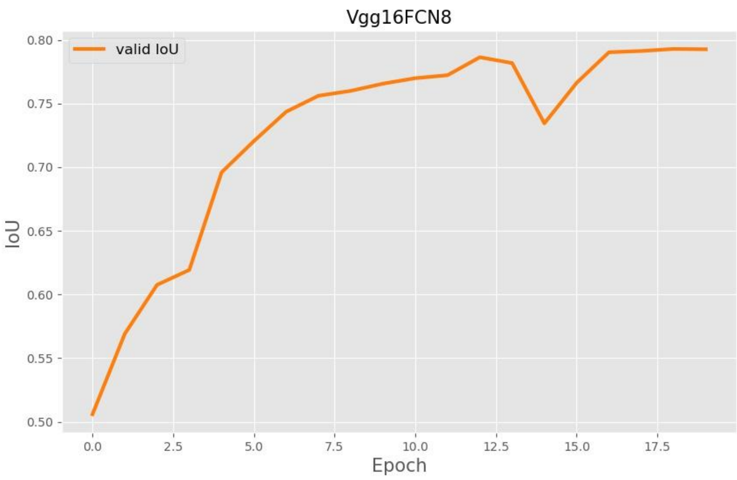
Learning rate	0.0005
Total Epochs	20
Batch size	32

Evaluation Metric: Intersection over Union $\frac{area\ of\ prediction\ intersect\ with\ ground\ truth}{area\ of\ prediction\ union\ with\ ground\ truth}$

Training Result:



IoU on Validation Set during training:



Best Model:
Since Epochs performed best on the validation set, I picked Epochs 19 as my final model.

Epochs	Validation IoU
19	0.7929

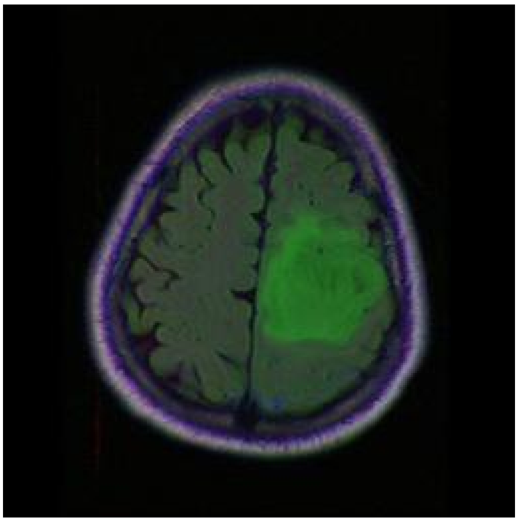
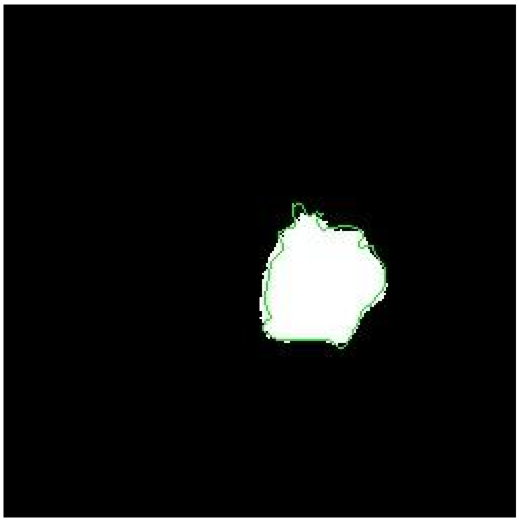
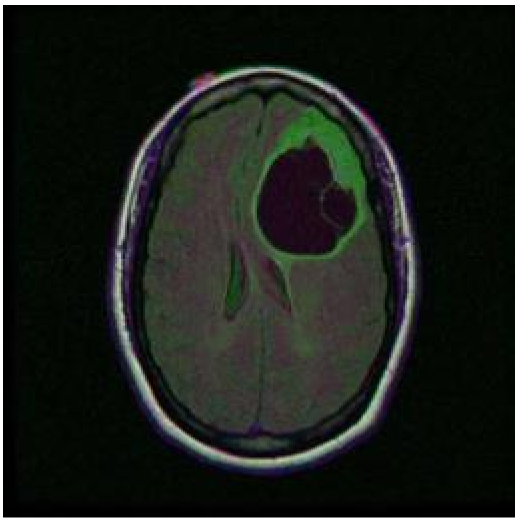
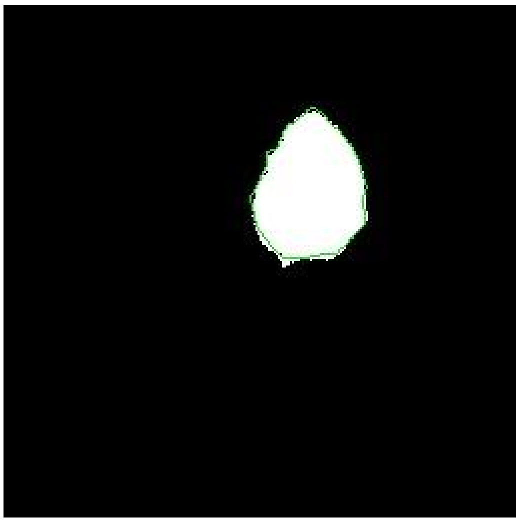
4. Testing and Prediction

Result:

Test IoU	0.8043

Prediction visualization:

The withe area is prediction and the green line is the ground truth

TCGA_CS_4943_20000902_16	Prediction and Ground Truth
	
TCGA_DU_5872_19950223_41	Prediction and Ground Truth
	

5. **Thoughts and Future work**

   Using the transfer learning technique can save a lot of time on the training process.With the pretrained model offered by torchvision, only need to focus on the Fully convolutional neural network. Also, the pretrained model was trained with huge amounts of images compared to our dataset, this made the model more stable and perform much better if we only random initialized the model's weight and trained with the dataset.With the pretrained model, the results on this Brain MRI segmentation task was very impressive. For the future improvement, there is some potential work that can be done.Such as implementing a different neural network like: U-Net, SegNet,fine tuning the hyperparameters, and applying data augmentation to the dataset for more images.

6. **Code Repositery**

   github: https://github.com/m1stborn/MIP2021

7. **Reference**

   a. dataset: https://www.kaggle.com/mateuszbuda/lgg-mri-segmentation
   b. github repository: wkentaro/pytorch-fcn
   c. github repository: IanTaehoonYoo/semantic-segmentation-pytorch
   d. FCN paper: https://arxiv.org/abs/1605.06211
   e. https://www.kaggle.com/lqdisme/brain-mri-segmentation-unet-pytorch/notebook