CHALMERS

Deep Learning for Brain Tumor Segmentation

training with foreground and background bounding box areas

Master's thesis in System, Control and Mechatronics

Xiaohan Bai

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Master's thesis 2021

Deep Learning for Brain Tumor Segmentation

training with foreground and background bounding box areas

Xiaohan Bai

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2021

Deep Learning for Brain Tumor Segmentation
training with foreground and background bounding box areas
Xiaohan Bai

Supervisor and examiner: Professor Irene Yu-Hua Gu, Department of Electrical Engineering, Chalmers

Master's Thesis 2021
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2021

Deep Learning for Brain Tumor Segmentation
training with foreground and background bounding box areas
Xiaohan Bai
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Deep machine learning on medical image analysis has become a popular topic since the demand for computer-aided diagnosis has increase in past decades. For MRI scans, U-net is a frequently used DL method with good segmentation results reported. However, it is common that annotated ground truth data is used in supervised training. For segmentation of brain tumors from MRIs, DL methods usually require annotated tumors from medical experts, which is time consuming process. To overcome this problem, this thesis investigates a novel approach by using foreground tumor area and background tissue area specified by 2 ellipse bounding boxes as the input of a multi-stream U-net. This is then followed by a small number of annotated tumor data (less than 20 patients) for refined training. To further improve the performance, we also study an approach where weights are added on unbalanced classes during the training.
Experiments have been conducted on 2 datasets (i.e., MICCAI with 4 modalities and US with 2 modalities). Since the US dataset is very small, we used FG-BG trained DL network by 2 modality MICAII dataset followed by a further refined training using a small number of patients in US dataset.
Results and evaluation are included. The proposed methods have achieved an average of 0.9724 test accuracy on 5 runs (where training and test subsets were patient-wise re-partitioned) from MICCAI dataset (4 modalities) and an average of 0.9910 test accuracy on 5 runs from US dataset after adding weights on unbalanced classes. Comparing with that of using annotated tumor GT trained scheme, the degradation of test performance is about 6% and 1% for MICCAI and US dataset respectively. The proposed method provides an alternative approach that can save time consuming manual tumor annotation by medical experts with a trade-off of a slightly reduced segmentation accuracy on the test sets.

Keywords: MRI, tumor segmentation, ellipse bounding box, supervised training by foreground-background areas, multi-stream U-net, multi-modality MRIs

Acknowledgements

First of all I would like to extend my deepest gratitude to Irene Gu for her invaluable guidance, without whose help this project would not have been possible. She trained me to think clearly, logically, and orderly. The sincerity and selfless on her have deeply inspired me.
Then, I would like to thank Linus Lagergren and Carl Rosengren for providing their programs on U-net segmentation, which helped me to quick enter the project.
I would also like to thank Prof. Asgeir Jakola from Sahlgrenska university hospital in Gothenburg for providing the US dataset for my experiments.
I would like to express my last gratitude to my parents and my friends for their love and understanding. Without their accompany and support, this experience would not be so special and meaningful.

Xiaohan Bai, Gothenburg, October 2021

Contents

List of Figures

List of Tables

1

Introduction

1.1 Medical image analysis

The origin of medical images is the x-ray invented in around the start of 20th century, which brings possibilities for doctors to locate trauma that could not be seen previously. The initial x-ray technique would take more than 11 minutes, and the risk of radiation exposure has not gotten any attention since late 1940s. The dangers associated with medical imaging dose not prevent people from exploring the potential of imaging technologies. Today's x-ray imaging has been strictly restricted by safety procedures and only takes milliseconds.

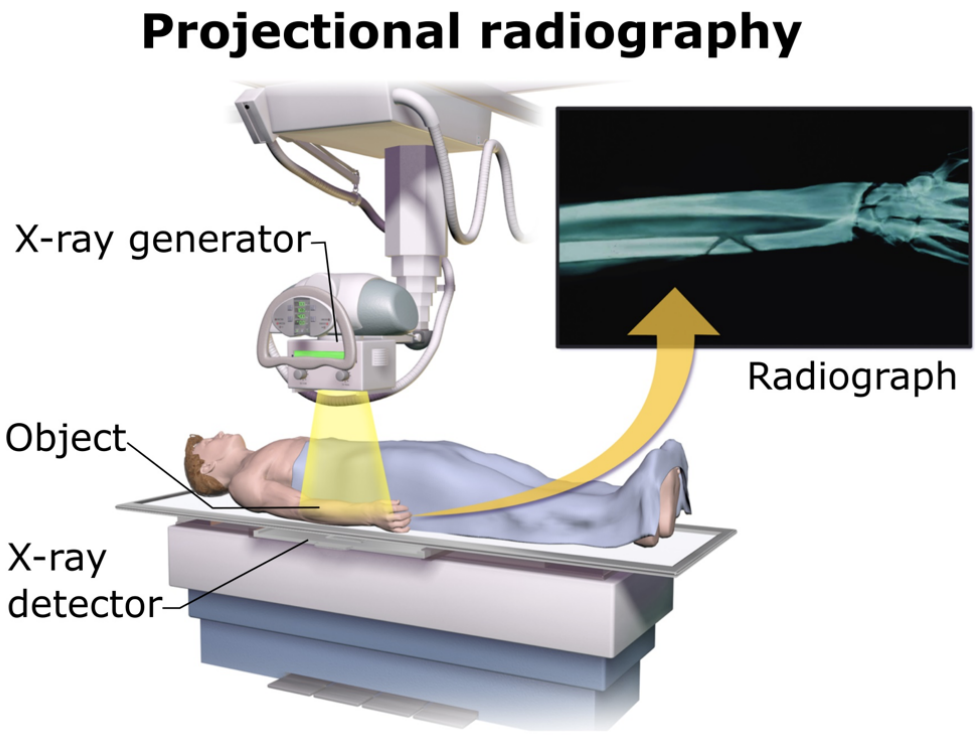


Figure 1.1: Diagram of how a x-ray scanner works. The figure is extracted from [1]

With the development of computed tomography (CT scanning) and magnetic imaging (MRI), computer stepped in the world of medical imaging in the 1970s. CT played an important role for first allowing to take a series of image slices of the

body and put them together with a computer to analysis the structure. It also benefits in smaller x-ray does compared with x-ray.

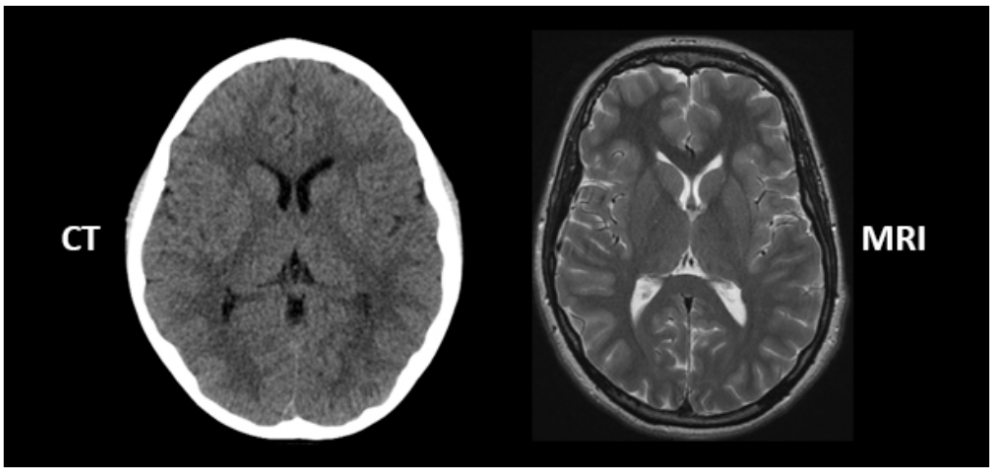


Figure 1.2: A CT scan vs a MRI scan. The figure is extracted from [1]

The clinical MRI today is based on the hydrogen nucleus with a resonance frequency. With a known transmitted frequency, only the protons at the right value would resonate, providing the location of the hydrogen nuclei. The RF pulse containing multiple frequencies is applied on the body and different timing of the RF pulses relates to different modalities, such as T1 and T2. Differentiated with CT, MRI performs with no ionizing radiation and has more richness of contrast mechanism. It is a common method to assess the brain and spinal cord abnormalities.
Data captured by x-rays, MRI, CT contains rich information, requiring professional radiologists. Automatic medical image analysis can ease the burden for radiologists and focus them on the crucial information picked up by computers. It could also be possible to save clinicians from radiology training, enabling all doctors to have the ability of interpreting medical images.

1.2 Motivation

The central nervous system (CNS) is made up with brain and spinal column, which controls all the vital functions of human, such as speech, thought and body movement [7]. A brain tumor is a collection of abnormal cells in brain or central spine, which can disrupt normal body functions. Unlike normal cells which are controlled by mechanisms, the brain tumors grow and multiply uncontrollably [8].
Depending on where the brain tumor starts, they can be classified as primary and metastases [9]. Primary brains tumors originate from the tissue of the brain or the brain surroundings. These tumors can be both malignant and benign. Metastases brain tumors are tumors that arises in another part of the body and then migrates to the brain, which are always cancerous and named by the beginning location.
The diagnostic tools of brain tumors include computed tomography (CT) and magnetic resonance imaging (MRI) [8]. Both methods can be complicated and time-

consuming, and it always involves a number of specialists. Sometimes, it is necessary to conduct a biopsy to identify whether the brain tumor is benign or malignant.

The concept of computer-aided diagnosis (CAD) was proposed in the 1980s [10], which aims to help the radiologists in image interpretation. Most of the earlier CAD systems were developed based on traditional machine learning methods, which could only provide limited help to improve the diagnostic accuracy. Such situation did not change until the deep learning methods have shown great success in machine learning area with excellent performance on image analysis.

The development of deep learning has made it possible to build a trainable model which can extract the features of specific tasks automatically. It gives a better performance than traditional machine learning in many fields, such as medical image segmentation. Using deep learning methods to segment tumors can save time for hospitals as well as patients. It also achieves to provide more accurate results in diagnosis.

In this thesis, we focus on automatic segmentation of brain tumor for MRI scans. It explores a new labeling method which save time for recognizing the detailed tumor boundaries.

1.3 Existing work

Image segmentation is an essential part in image processing. It gives a pixel-wise mask of provided image which extracts the area people are interested in.

The early image segmentation methods include thresholding [11], statistical approach [12], to more advanced algorithms such as active contours [13], conditional and Markov random fields [14]. In the deep learning area, convolutional neural network (CNN) is the most wildly used architecture. Early deep learning method on image segmentation uses a fully convolutional network (FCN), which is a modification of existing CNN architectures. Later algorithms for image segmentation tend to use encoder-decoder models, such as U-net [6], which includes a down-sampling path and an up-sampling path. There are also models based on regional convolutional network (R-CNN), which simultaneously perform object detection and semantic segmentation.

The existing segmentation methods require large amount of accurate labeled data, which is hard to meet in medical area. There are some research focus on dealing with imperfect data in medical images. Yan et al. [15] use self-supervised learning and randomly remove one modality during training to improve the model performance on missing modality. Mohammad et. al [16] propose a method which assign individual convolutional pipeline for each modality and merge available modalities in the end. Rihuan et. al [17] also explore practical situations without enough accurate labels, which combine coarse labels with pixel-wised annotations and learn in an end-to-end multi-task learning framework.

1.4 Aim of this thesis work

The main aim of this thesis is to exploit and test an improved labeling scheme for brain tumor segmentation problem. It uses ellipse bounding boxes to distinguish the tumor as foreground and the background area. Comparing with the traditional annotated tumor, this labeling method can save time-consuming manual data annotation by medical personnel. Experiments in this thesis are based on a multi-stream 2D U-net structure. The segmentation results with data labeled by the proposed labeling method are then compared with deep learning models trained by ground truth (GT). It also explores a weighting scheme to compensate for the unbalanced classes problem in brain tumor segmentation task.

2

Theories and Methods: Review

2.1 Deep learning in medical image analysis

Medical images such as computed tomography (CT), magnetic resonance (MR), ultrasound, and so on are widely used for the diagnosis in recent decades. Such images contain large amounts of information which require the medical professionals to perform an evaluation in a short time. The computer-aided diagnosis (CAD) which helps to analysis medical images has yielded an exciting development with the rising of deep learning.

The essence of deep learning method is a multi-layer neural network, which automatically learns from large amount of data. It mimics the human brain and provides representation of data through a forward propagation and a back propagation. The forward propagation calculates and stores variables from input layer to output layer. It starts with a weight initialization and each neuron combines input from previous layer with weights $W^(1)$, then gives output after activation function ϕ.

$$z = W^{(1)}x \tag{2.1}$$

$$h = \phi(z) \tag{2.2}$$

Back propagation is a widely used algorithm for optimizing training result of neural network. In forward propagation, the output is not accurate due to the fixed weights. Back propagation feeds the error back to neural network and calculates the gradient backwards. The back propagation aims at updating the weights and bias in each iteration and reducing the loss.

$$\theta^{t+1} = \theta t - \alpha \frac{\partial E(X, \theta)}{\partial \theta} \tag{2.3}$$

Combining forward propagation and back propagation, it enables deep learning to extract features through training without requiring manually choosing features. Therefore, deep learning gives robust performance with data containing various features as long as large amount of data and enough training time are provided.

This section would focus on reviewing the deep learning methods used in medical image field, which contains three main tasks: image classification, object detection, and image segmentation.

2.1.1 Image classification

In medical image analysis area, one of the most common situation is to provide diagnostic conclusion with body scans, which is related to the image classification problem in deep learning. It takes an image as input and assigns a classification label for the image with Convolutional Neural Network (CNN).
One of the classical examples of medical image classification is gastrointestinal diseases classification in endoscopic images. Figure 2.1 is an classification result from reference [2], which classifies the endoscopic images as bleeding and ulcer.

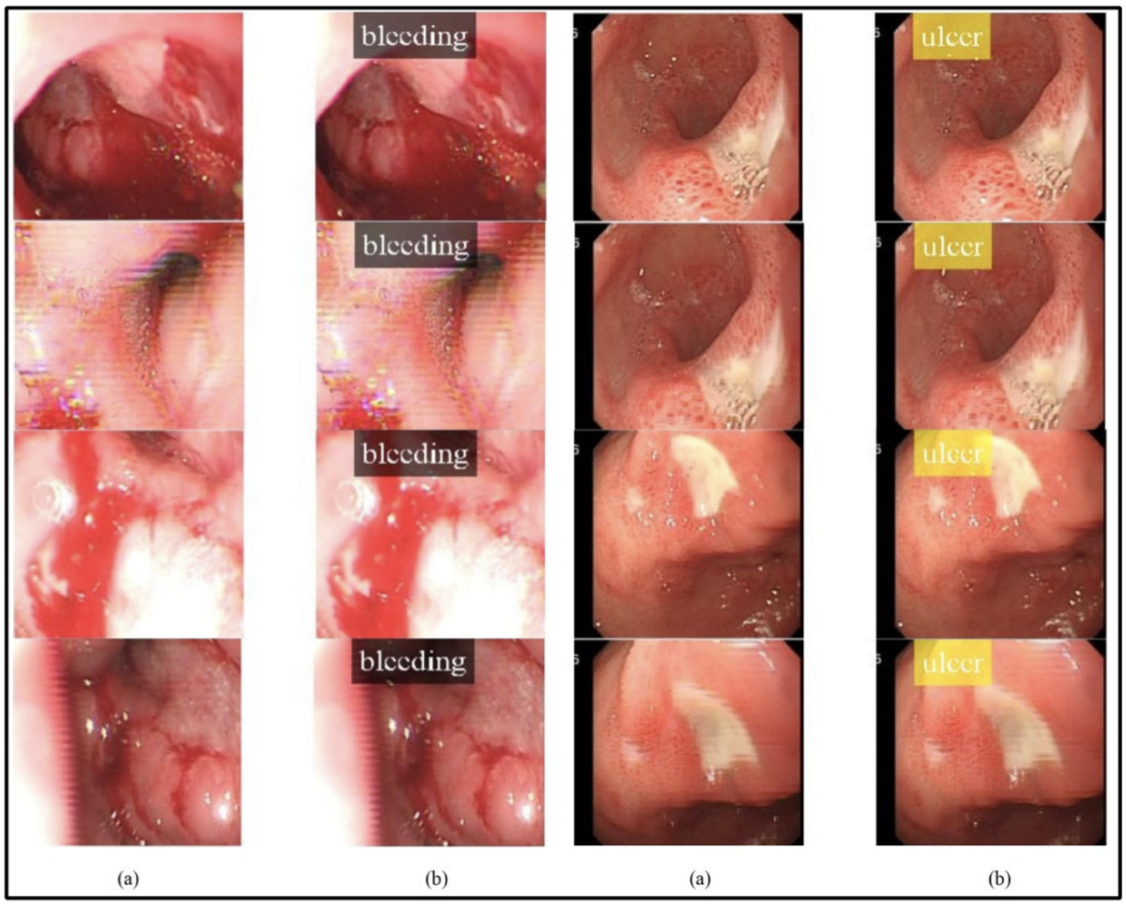


Figure 2.1: Example of image classification in gastrointestinal diseases from [2]

2.1.2 Object detection

Objection detection combines image classification and object localization, which gives bounding boxes with classification labels as output. It is a supervised machine learning problem and each image is associated with a classification label and a boundary. Most of the commonly used neural network structures are based on Region-based Convolutional Neural Network (R-CNN), which starts with a region selector to extract region that might contain objects, then warps the selected regions into a pre-defined size feeding in CNN for classification.
Figure 2.2 presents a typical object detection example in medical image analysis,

which is a liver lesion detection result in 4 phases from reference [3].

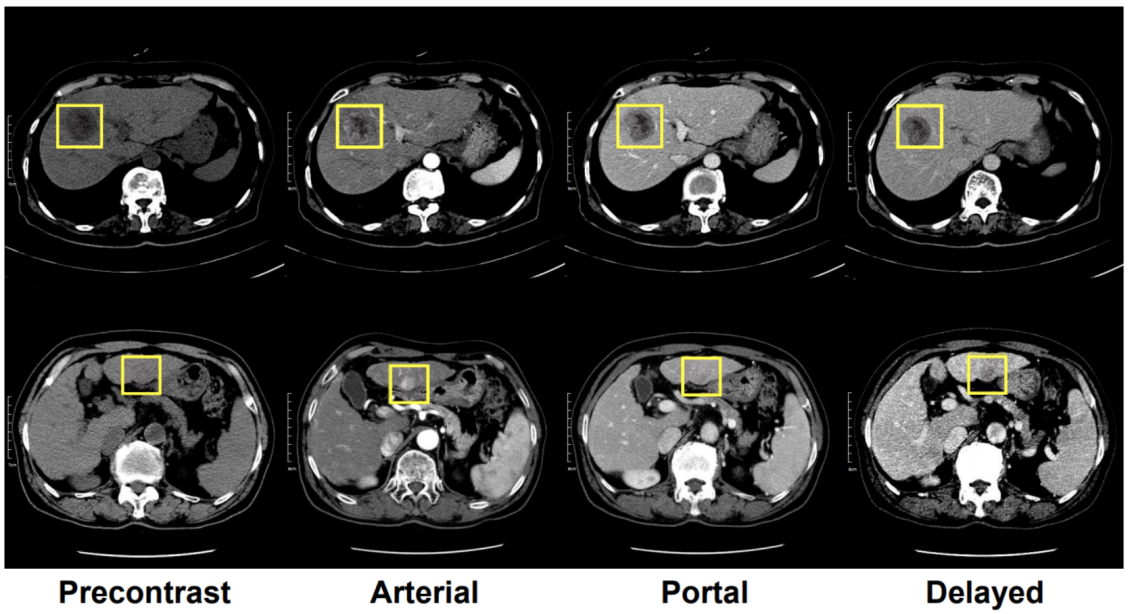


Figure 2.2: Example of object detection for liver lesion from [3]

2.1.3 Image Segmentation

The goal of image segmentation is to associate each pixel of input image with a corresponding class. It could determine the shape of objects and help to estimate measurements such as the size of objects. There are two kinds of segmentation tasks, one is semantic segmentation which classifies each pixel with label, the other is instance segmentation which segments individual objects and assigns different labels for each object.

Figure 2.3: The comparison between semantic segmentation and instance segmentation from [4]

FCN, U-net and Mask-RCNN are typical model architectures of solving semantic

segmentation problems. Figure 2.4 shows the intervertebral dic segmentation result from reference [5].

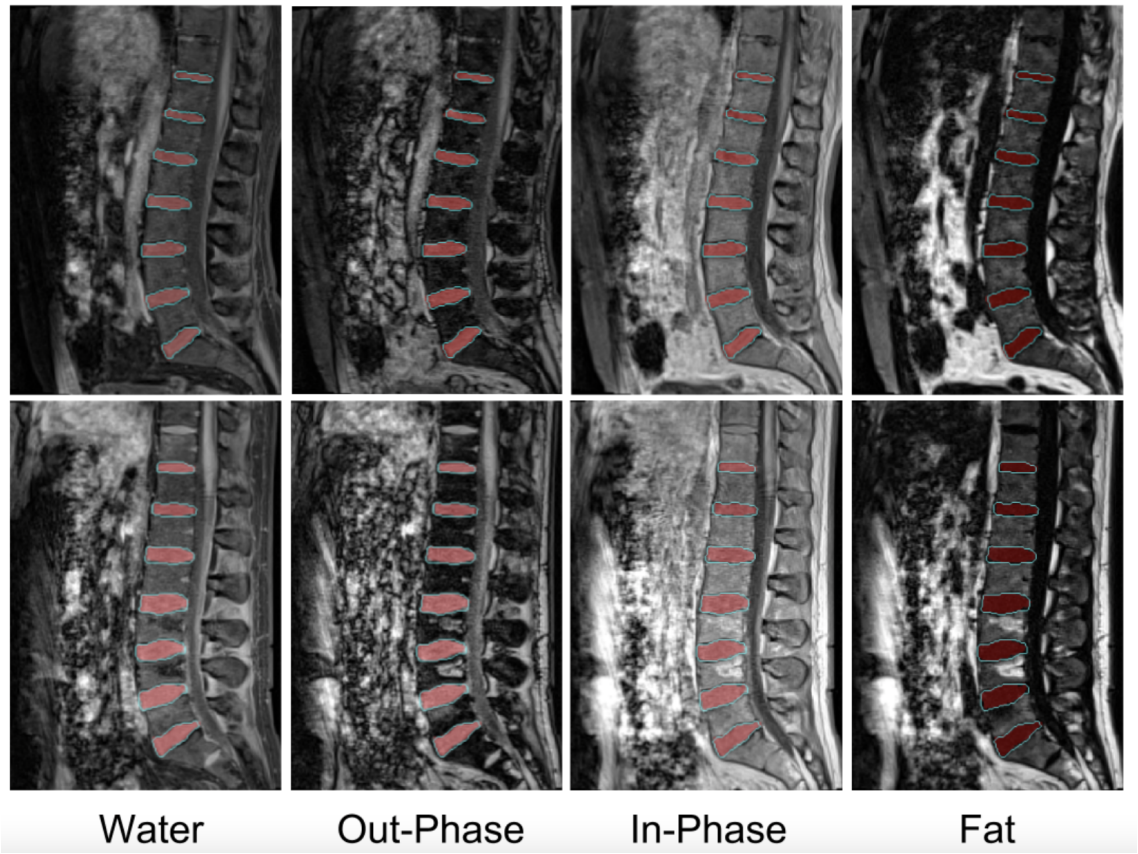


Figure 2.4: Example of image segmentation for intervertebral disc from [5]

2.2 Image segmentation architecture

This section presents commonly used neural network architecture for image segmentation task. It starts from the general Convolutional Neural Network (CNN), which is the foundation of other network architectures, and follows with two typical architectures, Fully Convolutional Network (FCN) and encoder-decoder architecture.

2.2.1 Convolutional Neural Network (CNN)

Convolutional neural network (CNN) is the most widely used multi-layer network structure for image analysis, which allows complex images and achieves various tasks. It always contains three basic structures, convolutional layers, pooling layers and fully connected layers [18]. A classic CNN structure look like Figure 2.5.

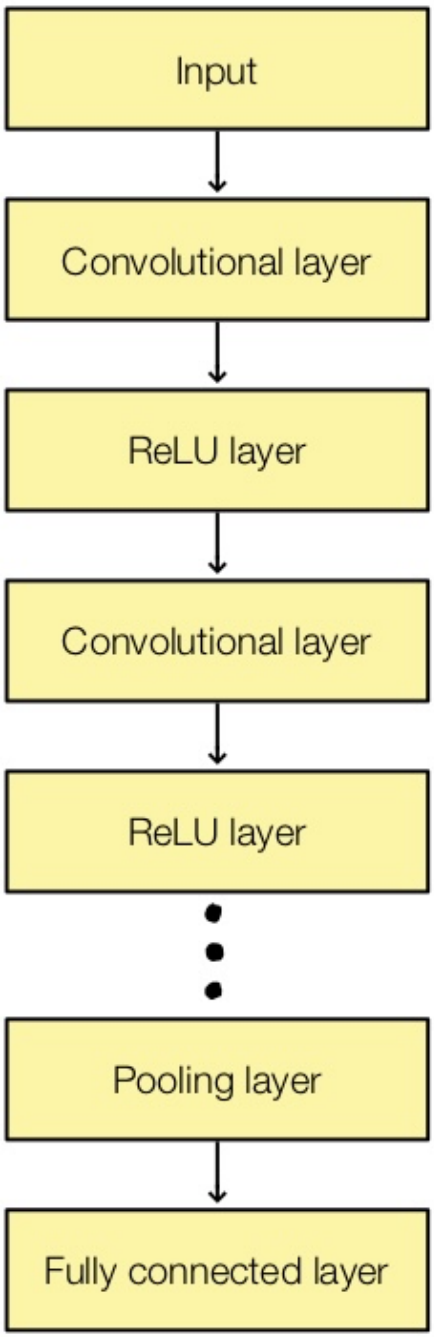
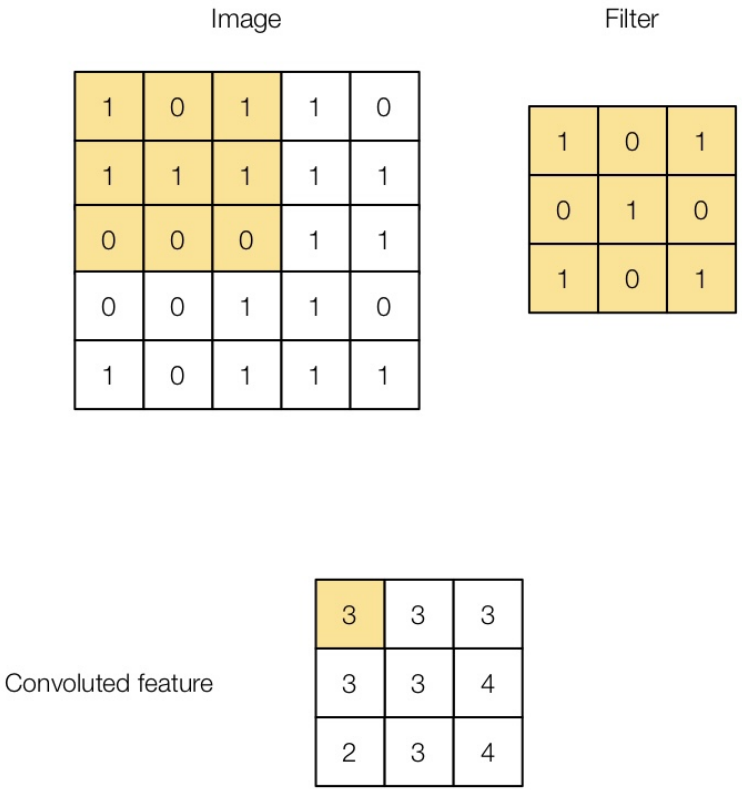


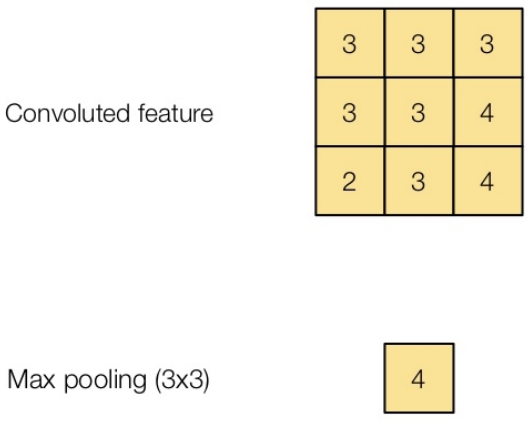
Figure 2.5: A classic CNN structure with convolutional layers, pooling layers and fully connected layers

Convolutional layers can extract features of input image by convolving a small region, which is called a filter or a kernel. The filter is applied on the input with a dot product multiplication. It is an element-wise multiplication between filter and a

patch of input image which is the same size of filter. The output is created by sliding the filter over input image and results in a 2-dimensional array which is called feature map.



(a) Convolute a 5x5 image with a 3x3 filter to get a 3x3 feature map



(b) 3x3 max pooling over a 3x3 feature map

Figure 2.6: Illustration of convolutinal layer and max pooling layer.

Pooling layers are always used after convolutional layers, which are responsible for reducing the size of feature map. For example, the most commonly used max pooling returns the maximum value of the input area covered by filter. The pooling layers can decrease the required computational power as well as extract the dominant features.
Fully connected layers are the last layers of CNN architectures. They can learn non-linear combinations of features provided by convolutional layers and give the probabilities for each label. Different from convolutional layers in which neurons only receive input from part of previous layer, each neuron in fully connected layer takes input from all neurons from last layer. The fully connected layer provides 'voting' fro each label and gives prediction after a softmax layer.

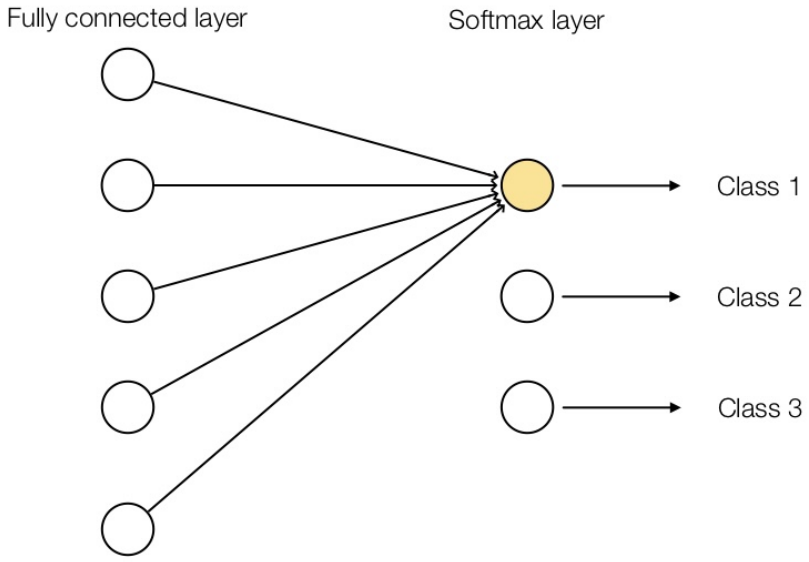


Figure 2.7: Illustration of fully connected layer

2.2.2 Fully Convolutional Network (FCN)

A fully convolutional network (FCN) proposed in 2015 [19] is a neural network which dose not contain any dense layer. It is equivalent to a typical convolutional neural network (CNN) in which fully connected layers are replaced by doing a 1x1 convolution through the entire region. It allows variable input image size, giving more flexibility than dense layer which needs a fixed input size.
Since the output layer provides a feature map which gets smaller and smaller with convolution, it is important to have upsampling in order to get an output of image size. In FCNs, the upsampling is done by transposed convolution, which is a reverse process of convolution. Figure 2.8 shows a transposed convolution process in which

the blue cubes are input and green cubes are related output. The correlation between input and output is presented by the arrow.

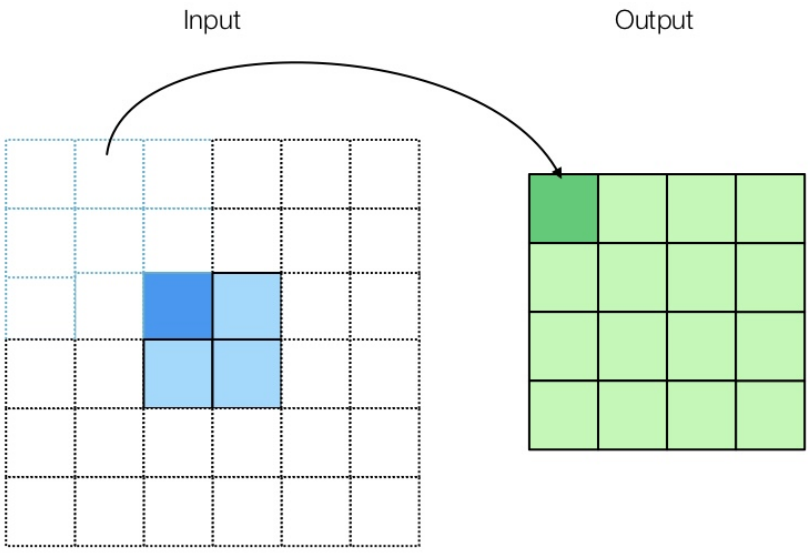


Figure 2.8: Illustration of a transposed convolution process. The blue cubes are input and green cubes are related output. The correlation between input and output is presented by the arrow.

2.2.3 The U-net and the encoder-decoder architecture

The encoder-decoder architecture is a general architecture for segmentation problem. Figure 2.9 shows a simple representation of the encoder-decoder architecture. The encoder can be a pre-trained classification network like VGG, and the decoder is to upsample the input features and provide pixel-level predictions.

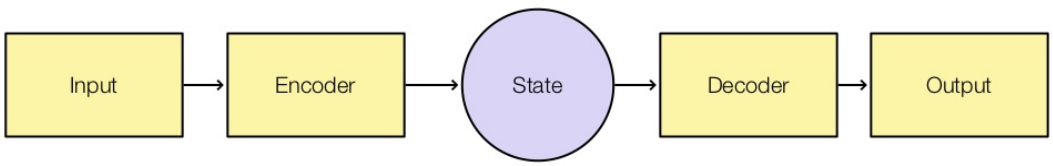


Figure 2.9: A simple presentation of the encoder-decoder architecture.

U-net [6] is a famous network with encoder-decoder architecture based on FCN and modified to get better segmentation result with smaller dataset. It is widely used in medical image analysis.

The U-net architecture consists of a downsampling path/encoder and an upsampling path/decoder like in Figure 2.10. In the left side of U-net, the regular convolutional layers and max pooling layers are applied, which decrease the size of input image while the depth increases. In the right side of U-net, transposed convolutional layers are applied, which is to perform upsampling and learn parameters through back propagation. It increases the input image to the original size and locates the information. In decoder path, there exist concatenations which concatenate the feature maps from encoder path with unsampled feature maps in order to get a better representation.

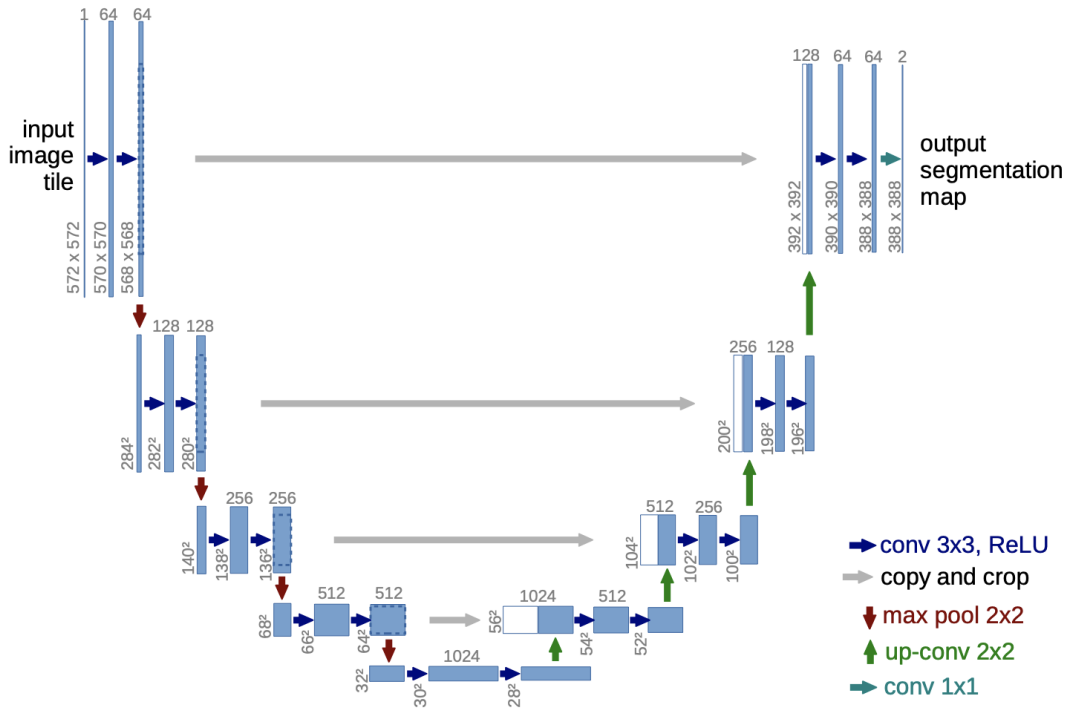


Figure 2.10: U-net structure proposed by Olaf Ronneberger et al [6]

2.3 Loss function

Loss function is used to evaluate an optimisation algorithm by calculating errors. Regression loss and classification loss are two major types of loss function. The regression loss predicts continuous values such as prices, and the classification loss provides predictions from a finite set of categorical values. The segmentation problem which has finite number of classes requires classification loss. The following sections illustrates two typical classification loss, cross-entropy loss and dice loss.

2.3.1 Cross-entropy loss

Cross-entropy loss is the most common used loss function in classification problems, which is also called log loss. Such loss function is logarithmic, which manages to penalizes heavily for wrong and confident predictions

$$CrossEntropyLoss = -(ylog(p) + (1-y)log(1-p)) \tag{2.4}$$

2.3.2 Dice coefficient

Dice coefficient is a statistic used to evaluate the similarities of two samples, which is widely used in medical image analysis.

2.3.2.1 Dice score

Dice score measures the overlap between the ground truth and prediction.

$$D = \frac{2|X \cap Y|}{|X| + |Y|} \tag{2.5}$$

where $|X \cap Y|$ is the overlap pixels between ground truth and predictions, $|X|$ and $|Y|$ are total number of pixels in both image. The dice score used in this project calculates the overlap of tumor pixels except background pixels. X denotes the ground truth tumor, and Y denotes the segmented tumor.

2.3.2.2 Dice loss

In medical image segmentation, it is common that the anatomy of interest consists of only a small region of the image. The training process risks stuck in local minimum which is strongly based on the background to predict. Dice loss layer is proposed by Fausto Milletari et al [20] to deal with imbalance classes.

$$D_{loss} = 1 - \frac{2|X \cap Y|}{|X| + |Y|} \tag{2.6}$$

The segmentation problem aims at maximizing of overlap for target region. It is straight forward using dice loss rather than cross-entropy loss.

2.4 Learning rate

Learning rate decay is a technique for training neural networks, which reduces learning rate during training according to a pre-defined learning rate schedules. This technique could help the model converge to a satisfying local minimum with less oscillation. The following sections explains two common learning rate scheduler, one is step decay, the other is exponential decay.
Learning rate for training neural networks needs to be carefully adjusted. The aim of adjusting learning rate is to reach fast convergence meanwhile keep the variance small in the end, i.e., the model would converge to a satisfying minimum value with small oscillation. The following sections explains some common learning rate parameter selection approaches, such as step decay and exponential decay.

2.4.1 Step decay

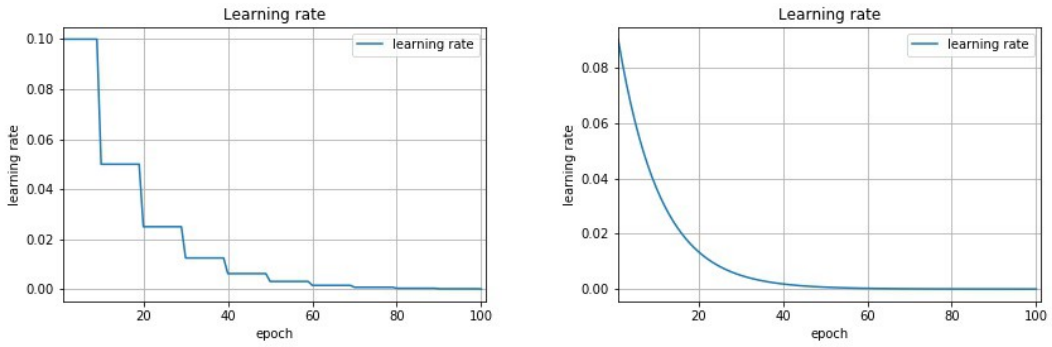
The most commonly used approach for setting learning rate is to let the value decay over steps, in which learning rate decreases when epoch increases.

2.4.2 Exponential decay

In the case of exponential decay, the learning rate decreases exponentially over the time. This is written as

$$LearningRate = LearningRate_0 \times e^{-kt} \tag{2.7}$$

Both learning rate decay methods could be implemented in keras with a **LearningRateScheduler** callback, and the examples of learning rate scheduler is in Figure 2.11



(a) Step decay scheduler

(b) Exponential decay scheduler

Figure 2.11: Illustration of different learning rate scheduler.

2.5 Transfer learning, refined training

For many task in machine learning area, state-of-the-art models have achieved to provide more and more accurate results. However, many of these models rely on huge amounts of accurate labeled data which are time-consuming as well as expensive.
Transfer learning is a machine learning method which stores knowledge learned from one domain and then applies it on another different but related domain. It enables to utilize knowledge from previous task. In the computer vision , the low-level features, such as shapes and edges, can be used in general tasks, which manage to transfer knowledge from one task to another.
The convolutional neural networks capture different features in each convolutional layer. The lower convolutional layer captures the low-level features, and the higher convolutional layer can capture more complex features. Generally, the last fully connected layer is used to score for respective tasks. When the pre-trained model is applied on a different task, it is important that the previously acquired feature would not lose.

The most common method is to take layers from a pre-trained model and freeze them in order to preserve the general features as in Figure 2.12. Then, new trainable layers are added to the top of the model which can help to adjust the old features to new scenarios. Only the new-added layers will be trained.

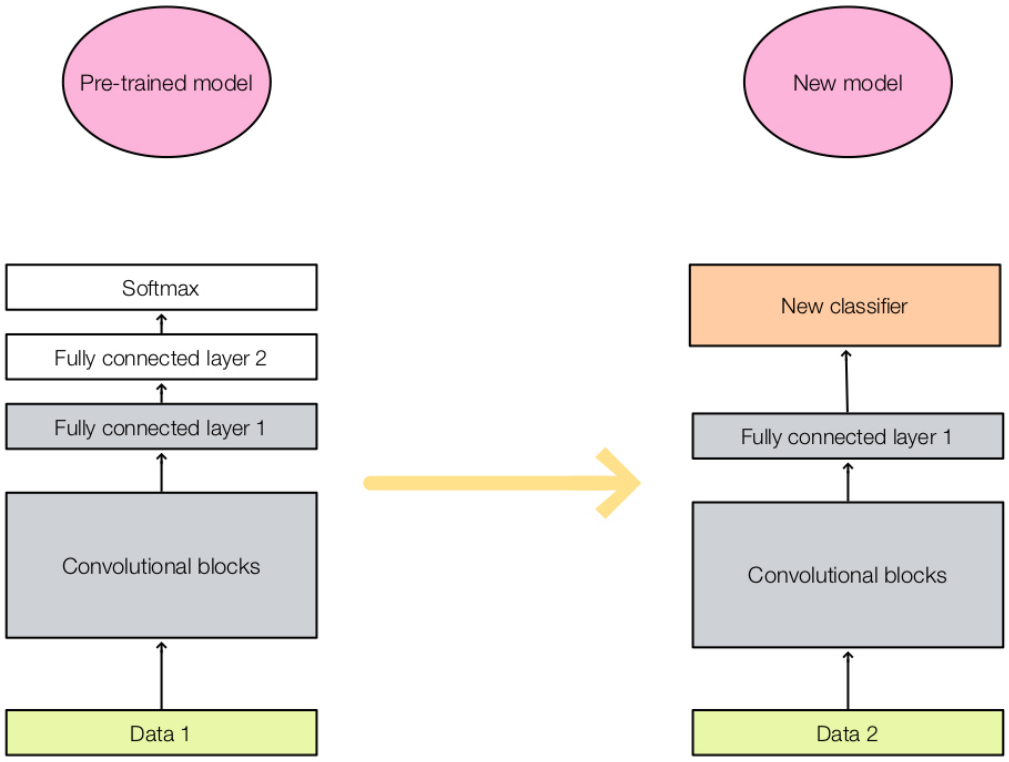


Figure 2.12: Transfer learning with pre-trained model. The gray blocks represent layers which are frozen during training

In refined training, no layers are fixed. The whole model is retrained on new data using the previous trained coefficients as the initial values. The parameters of gray blocks in Figure 2.12 would change with back propagation during training.

3

Deep learning Methods and Scheme in This Thesis Work

This thesis focuses on exploring a new labeling method for data in brain tumor segmentation task. It automatically draws the bounding boxes for tumor area and background area separately and discards the middle area between two bounding boxes. Deep learning model in this thesis is based on a multi-stream U-net structure from
The model is first trained with a large number of foreground-background (FG-BG) ellipse bounding box area data instead of annotated ground truth (GT) tumor data. Then it is refined by using a small number of GT labeled data.
Figure 3.1 shows the block diagram which presents the scheme of this thesis work. It starts with automatically drawing bounding boxes for tumor area and background area separately. Two ellipses are drawn automatically on the MRI scans for brain tumor, extracting the tumor pixels and background brain tissue pixels. Then, the tumor pixels and background pixels are combined in the deep learning model based on 2D U-net structure. Then the training model is refined with a small number of GT labeled tumor data. To compare the performance, we also use the same scheme trained on manually annotated GT tumor data . The two results are then compared to evaluate the performance of the proposed segmentation method.

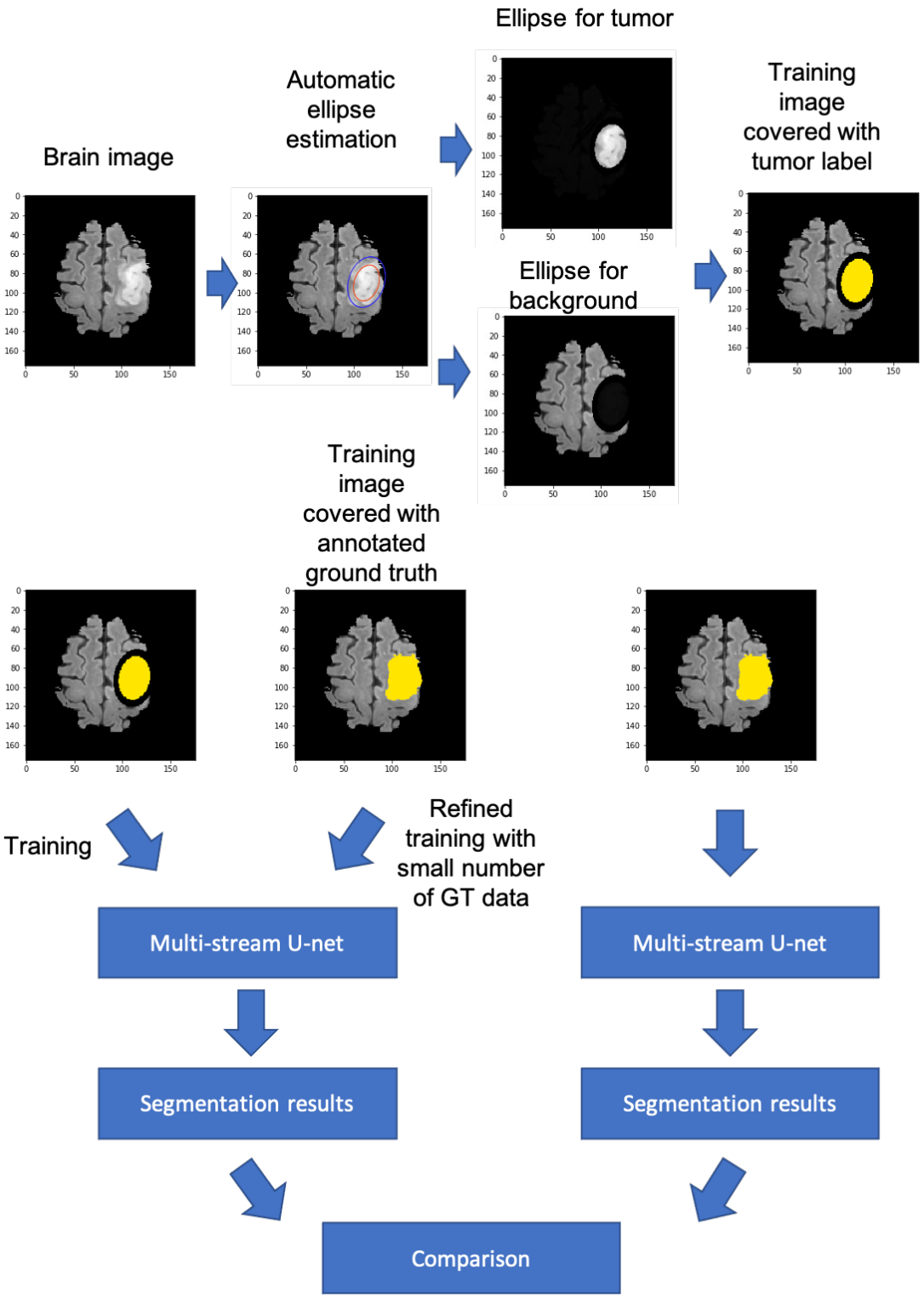


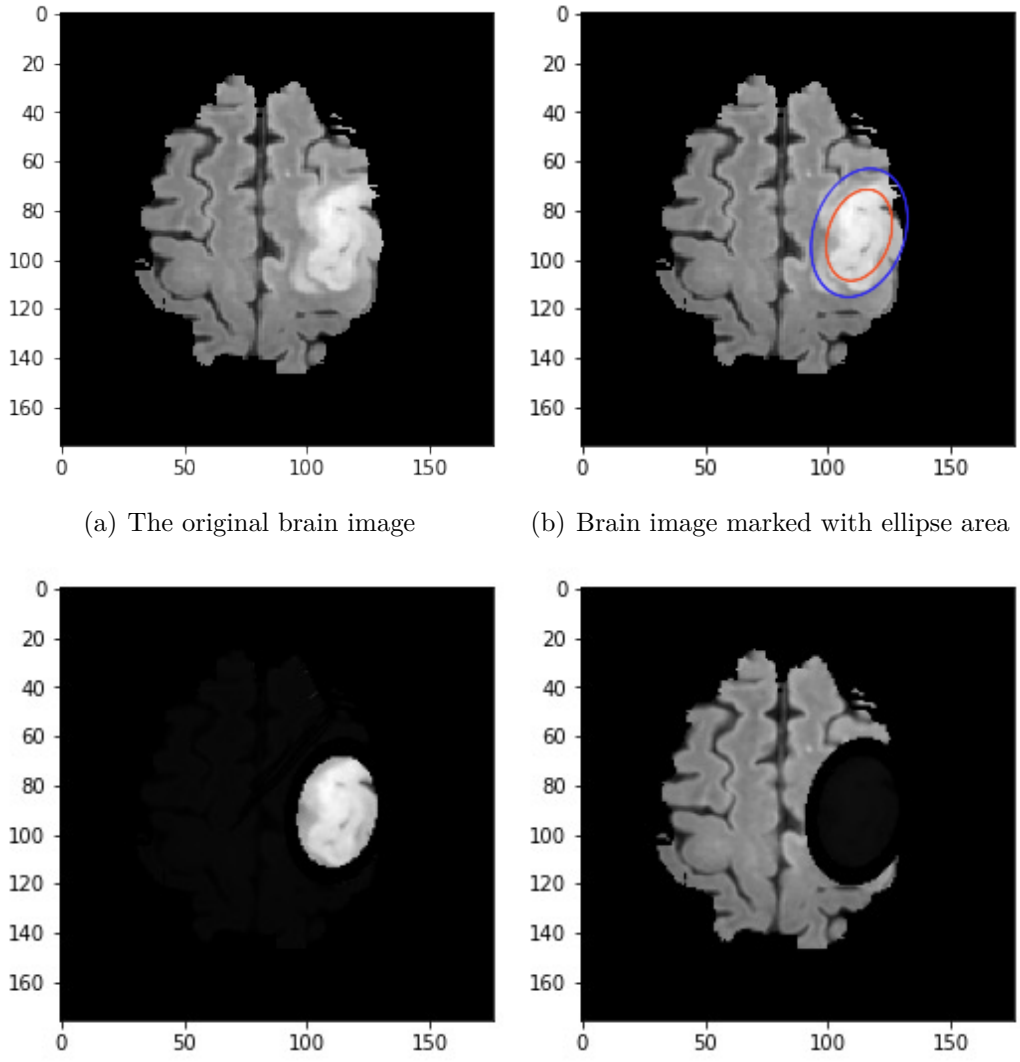
Figure 3.1: Pipeline of the DL scheme studied in this thesis work.

3.1 Automatic allocation of ellipse areas for foreground (FG) tumors and background (BG) normal brain tissues.

The deep learning approaches for brain tumor segmentation rely on large amounts of annotated MRI images. However, in practical situation, manual annotation of accurate tumor boundaries used for training is a time-consuming process and requires many medical experts.
This section proposed an automatically extraction of foreground-background bounding box areas without requiring GT labeling. This proposed labeling method does not require accurate boundaries of tumor, thus saves time for medical experts.

3.1.1 Bounding areas generation

Bounding areas in this labeling method is defined with ellipse shape. There are two ellipse drawn around tumor area. One is smaller than the actual tumor, which guarantees that all pixels inside small ellipse belong to the tumor. The other ellipse size is larger than that of the tumor, which guarantees that all pixels outside large ellipse belong to the background tissue. This FG-BG labeling methods does not require GT tumor boundaries, thus without requiring time consuming labeling from medical doctors. Figure 3.2 shows an illustration of the labeling method.
The ellipses are automatically generated by estimating the tumor area. Mathematically, an ellipse can be defined by its center, major axis length, minor axis length and orientation. There exists a Matlab function **regionprops** which can help to measure properties of image regions and return the desired measurements. Figure 3.3 shows how the Matlab function estimates a given area with ellipse.



(a) The original brain image

(b) Brain image marked with ellipse area

(c) Bounding box for tumor

(d) Bounding box for background

Figure 3.2: An illustration of how the bounding boxes are drawn for tumor and background. (a) The original brain image marked with 2 ellipse boxes. (b) Brain image marked with automatically estimated ellipse area. (c) Interior area from a small ellipse box is used as the foreground tumor area for training. (d) Exterior area from large ellipse box is used as the background area in training

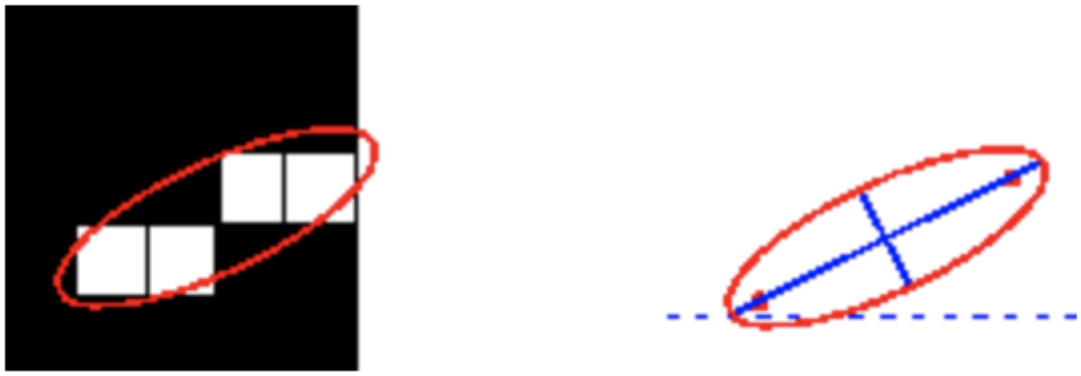
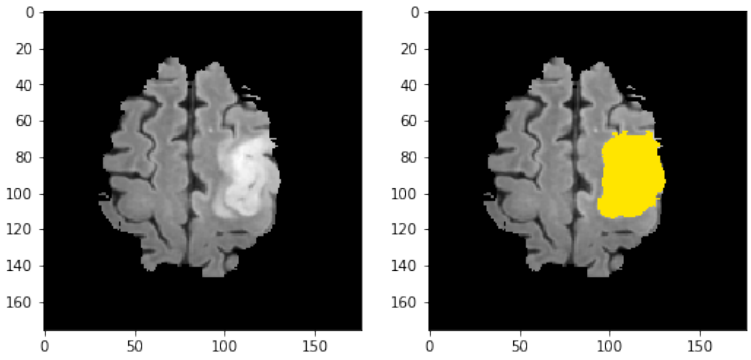


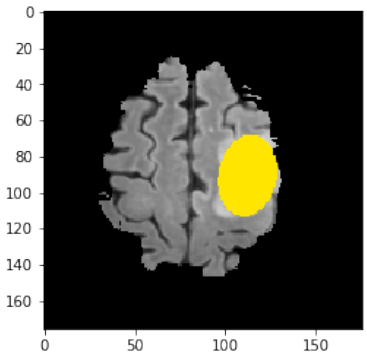
Figure 3.3: An ellipse estimation example from Matlab document.

3.1.2 Foreground-background labeled training data

Figure 3.4 presents the training data used after input with the proposed FG-BG areas. The proposed FG area does not fully cover the tumor and does not reflect the exact boundaries of tumor. The tumor area is specified by inside the small ellipse area, and the background tissues are specified by BG areas outside the large ellipse..



(a) Input training image (b) Tumor ground truth



(c) Tumor label

Figure 3.4: A training example of proposed labeling method

3.2 U-net scheme

The deep learning model in this thesis is based on the 2D U-net structure. It uses a multi-stream structure which gives each modality a separate channel, and combines outputs from different channels in the end for pixel-wise classification.

3.2.1 U-net structure for a single modality

For each modality channel, the U-net structure is the same. Figure 3.5 shows the details of U-net structure for a single modality. Besides the symmetric structure based on concatenation, it contains batch normalization layers in both encoder path and decoder path.
During training progress, the randomness from parameter initialization and input would accumulate in each layer, and the distribution of input for each layer would change since the previous layer would change its parameters through gradient descent. This phenomenon, which is called internal covariate shift, would make it complicated to train a neural network.
Internal covariate shift can be reduced by the batch normalization method. It applies a normalization step on input of each layer to guarantee the zero mean and unit variance. In deep learning, it is impractical to train on the entire data. Thus, the data would be split to batches and normalization is conducted on each batch.
In practice, using part of images in training could introduce unnatural features and bring difficulty in convergence. Adding more batch normalization layers may accelerate the training as well as stabilize the results.
The output block contains two extra convolutional layers and one softmax layer, which is shared in all channels for different modalities.

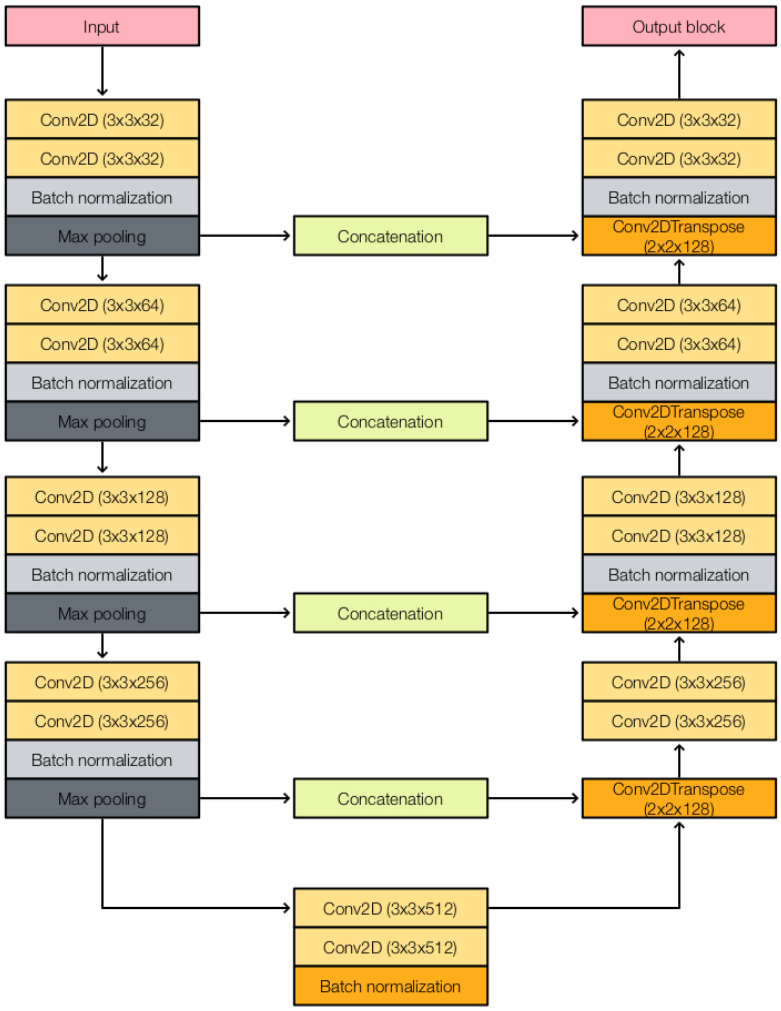


Figure 3.5: Structure and setting details for U-net in each channel.

3.2.2 Multi-stream fusion for multi-modalities

Figure 3.6 shows a flowchart of multi-stream U-net with 4 modalities. Outputs from different channels are concatenated, and sent into the output block. Number of channels for concatenation is depended on how many modalities the data contain. In this thesis, we use one multi-stream U-net with 4 modalities and one with 2 modalities. The first model contains T1, T1ce, T2, and FLAIR channels. The second model contains T1ce and FLAIR channels.

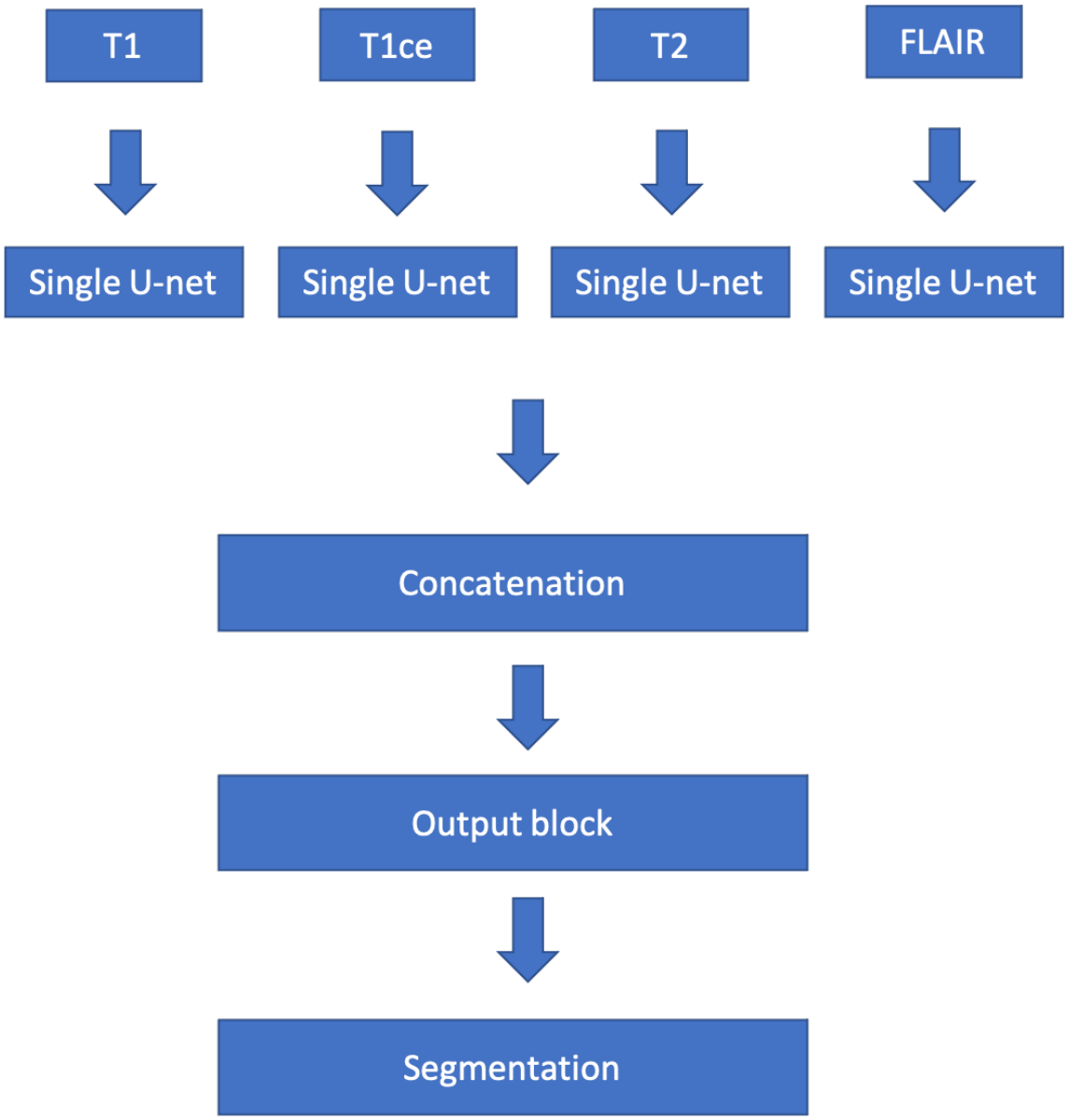


Figure 3.6: Multi-stream U-net structure for multi-modalities.

3.3 Experiments design

3.3.1 Case 1: Training multi-stream U-net based models using FG-BG bounding box areas data for tumor segmentation with 4 modalities (T1, T1ce, T2, FLAIR) and 2 modalities (T1ce, FLAIR) MRIs on two datasets

The main experiment in this thesis is to train a deep learning model with FG-BG ellipse area data, and prove that it is feasible for brain tumor segmentation without requiring manually annotated GT tumor data.
For MICCAI dataset, the tumor and background tissues are generated by FG-BG

ellipse area data according to the method described in Section 3.1. The deep learning models is based on the multi-stream 2D U-net structure in which the model details are described in Section 3.2.

Our experiments have been conducted on 2 datasets, one is MICCAI dataset consisting of 4 modalities (T1, T1ce, T2 and FLAIR MRI scans), and US dataset consisting of 2 modalities (T1ce and FLAIR MRI scans).

In 4-modality case, the model is first trained on a FG-BG labeled training set from MICCAI dataset with 4 modalities, which contains 171 patients. After the first step training, the model learns feature from tumor pixels and background pixels, but also learns some wrong features from the small non-tumor tissues inside ellipse bounding boxes. To overcome this problem, a second step is applied to refine the model by using 10% of annotated GT data from training set for training. Such refined training helps the model to learn more accurate features of tumors. The model performance is tested on test set from MICCAI data with 4 modalities.

Since US dataset is rather small, it is not sufficient for obtaining good training results. Hence, it uses model pre-trained by MICCAI dataset with 2 modalities and also refined with 10% of annotated GT data from training set. The model performance is tested on test set from MICCAI data with 2 modalities.

Then the model is refined again with 20% annotated GT data from US dataset in order to learn specific features of US dataset. Model after the second refined training is tested on the test set of US dataset.

3.3.2 Case 2: Adding weights to unbalanced classes for tumor and background in case 1 study

One of the major issues that lead to the difficulty of brain tumor segmentation is the unbalanced classes. Typically, the healthy tissues compose most part of the brain, while the tumors occupy a small area. Considering the semantic segmentation problem, the ratio of background pixels to tumor pixels is higher than 10, which means the neural network would tend to classify more pixels as background in order to get a better result from loss function. The typical method dealing with unbalanced classes is to introduce sample weighting in loss function. This method is to assign a higher weight to the minority classes and a lower weight to the majority classes, which allows the neural network to put more attention on the minority classes.

In some machine learning libraries like sklearn, they provide a "class weight" parameter helps estimate class weights for unbalanced dataset. The weights are based on the number of samples in each class, while with a larger value. The formula of weight estimation is

$$w_j = \frac{n_{samples}}{n_{classes} \times n_{samples_j}} \tag{3.1}$$

in which the w_j is the weight for the jth class, $n_{samples}$ is the sum of samples in the dataset, and $n_{samples_j}$ is the number of samples in jth class.

3.4 Criteria for performance evaluation

The segmentation performance is evaluated on test sets of 3 experiments.

- Using the model initially trained with 4-modality FG-BG data from MICCAI dataset, and refined trained with a small number of annotated GT data. The test performance is evaluated by applying the trained model on the MICCAI test subset.
- Using the model initially trained with 2-modality FG-BG data from MICCAI dataset, and refined trained with a small number of annotated GT data. The test performance is evaluated by applying the trained model on the MICCAI test subset.
- Using the model initially trained with 2-modality FG-BG data from MICCAI dataset and refined-trained with a small number of annotated GT data from MICCAI training set, following by further refined training on a small number of annotated GT data from US dataset. The test performance is evaluated by applying the trained model on the US test subset.

The evaluation is mainly based on the dice scores of tumor and the confusion matrices. Dice score used is described in Section 2.3.2. The dice score is based on the overlap of segmented tumor areas and the GT tumor areas. To further evaluate the performance, some segmented images are also included for visual observation. 5 slices in test set of MICCAI data are randomly selected to show the segmentation results from each model.

3.5 Comparison of performance

Performance comparison is conducted on the test results obtained from DL scheme trained by the proposed FG-BG approach, and by using the GT annotated tumors. Adding class weights to unbalanced classes is supposed to further improve the performance of the FG-BG approach. The performance of FG-BG approach, and FG-BG approach with class weights will be compared with the performance from the GT approach.

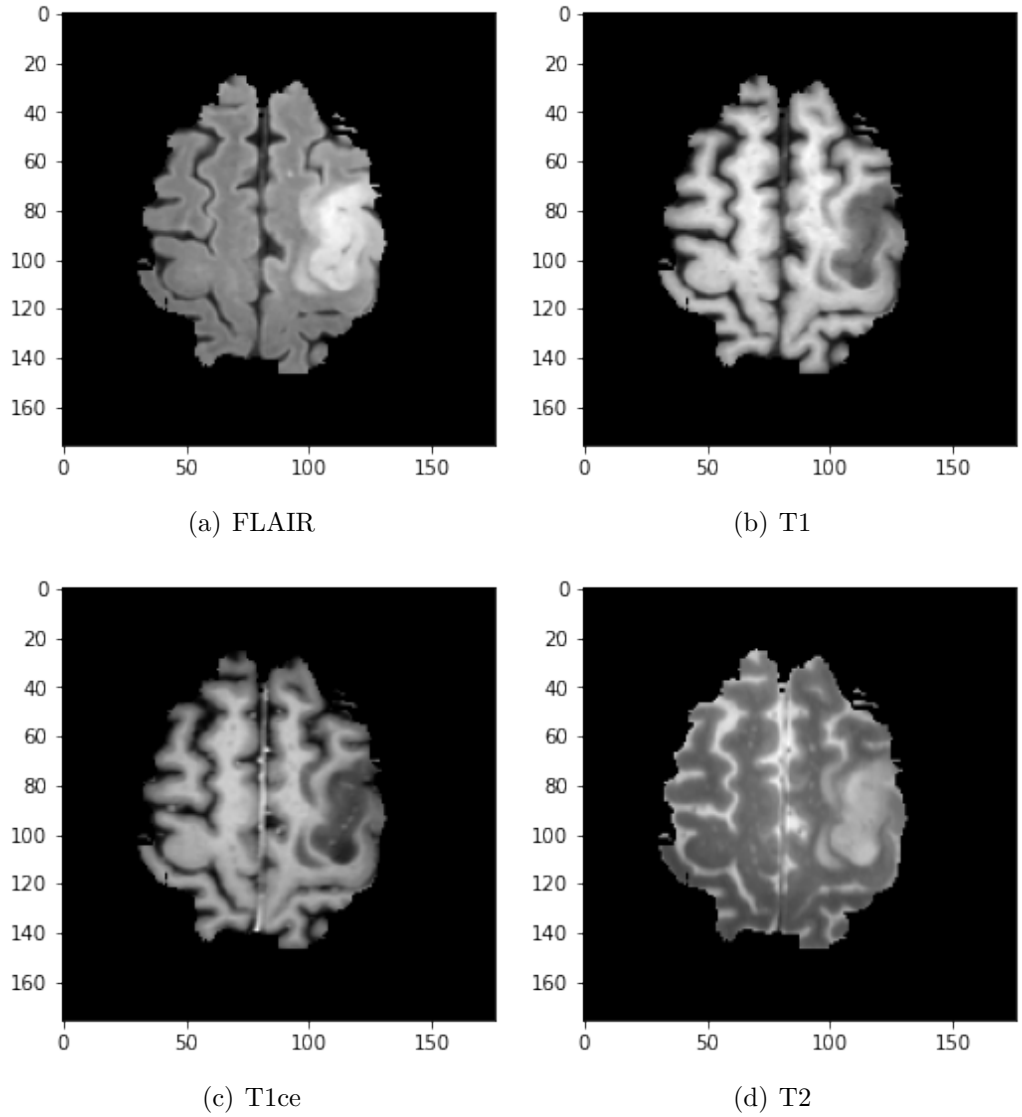
4

Results, Evaluation, and Comparison

4.1 Datasets used for experiments

4.1.1 MICCAI dataset

The MICCAI dataset was collected from Multimodal Brain Tumor Segmentation Challenge 2018. It consists of 4 modalities which are T1 weighted (T1), T1-weighted and contrast enhanced(T1ce), T2-weighted (T2) and T2 Fluid Attenuated Inversion Recovery (FLAIR). Figure 4.1 shows an example of patient scans in different modalities.
The dataset has been co-registered to the same anatomical template, interpolated to the same resolution and skull-stripped. Pixels in MICCAI dataset are classified with 4 classes with 0, 1, 2, 4 referring to background, the necrotic and non-enhancing tumor core, the peritumoral edema, and the enhancing tumor. In this thesis, we only focus on the whole tumor area, thus the data are simplified with background class noted as 0 and tumor class containing 1, 2, and 4 classes before noted as 1.
There are two folders, which are LGG(Lower Grade Glioma) and HGG(High Grade Glioma). The LGG folder contains 75 patients and the HGG folder contains 210 patients. For each patient, the scans are in NIfTI file and the 3D size is 155x240x240. 9 slices containing tumors are extracted from each 3D scan and the original slice size is 240x240.



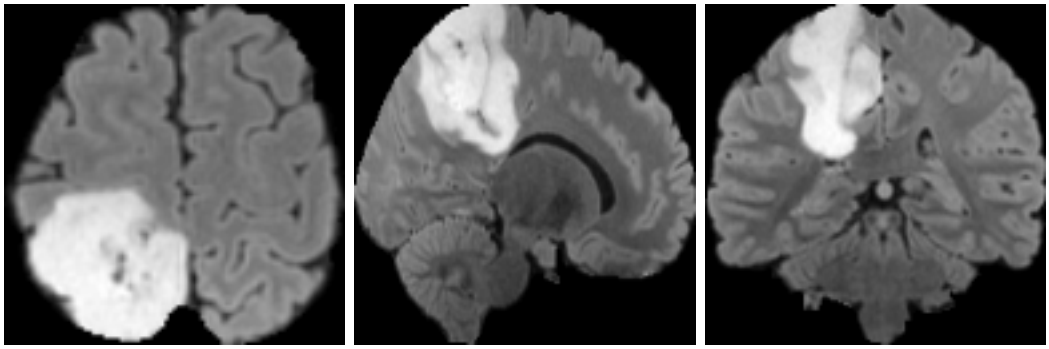
(a) FLAIR (b) T1

(c) T1ce (d) T2

Figure 4.1: An example of patient scans in different modalities

4.1.2 US dataset

The US dataset consists of two modalities which are T1ce and FLAIR with 75 patients. For each patient, it contains 15 MR image slices from 3 directions. The 2D image slices are in PNG format and the size of image is 128x128.



(a) Axial direction (b) Sagittal direction (c) Coronal direction

Figure 4.2: A patient example in US dataset with 3 directions.

Typically, number of slices for each direction in MRI scan varies from 150 to 210. The size of each modality for one patient in MICCAI dataset is 155x240x240. In order to reduce the size of data feed into neural network, we only select a few slices in each direction. The detailed information of MICCAI dataset and US dataset used in this thesis is in Table 4.1.

	T1	T1ce	T2	FLAIR	number of patients	number of slices per patient
MICCAI	Yes	Yes	Yes	Yes	285	9 (3 directions)
US	–	Yes	–	Yes	75	18 (3 directions)

Table 4.1: Data information of MICCAI dataset and US dataset used in this thesis.

4.2 Data pre-processing

4.2.1 Data cropping

The input size for each patient in MICCAI dataset is $155 \times 240 \times 240$, which contains a lot of background pixels. The first step of data pre-processing is to crop the slices into 176×176 in order to remove background pixels while retaining all brain pixels. This cropping is based on the center of each image thus has no effect on the relative position of brain.

4.2.2 Data normalization

Data normalization is an important process to prepare the data before training in image processing. In brain tumor segmentation, MRI scans with different scanner types or different scanning parameters have different intensity information. It has no effect in doctor diagnosis. However, when it comes to automatic segmentation, images without normalized intensity scale might fail to provide favorable results.
In this case, pixel values in each slice are normalized to zero mean and unit variance.

4.3 Environments

4.3.1 Cloud GPU platform

The experiments are first trained with Google Colab, which is a product provided by google research. It provides free GPUs with limited time. The Colab notebook is developed based on jupyter notebook, thus python codes written in jupyter notebook can be easily executed with Colab notebook. Figure 4.3 shows the interface of Colab notebook and presents how to enable GPU resources.

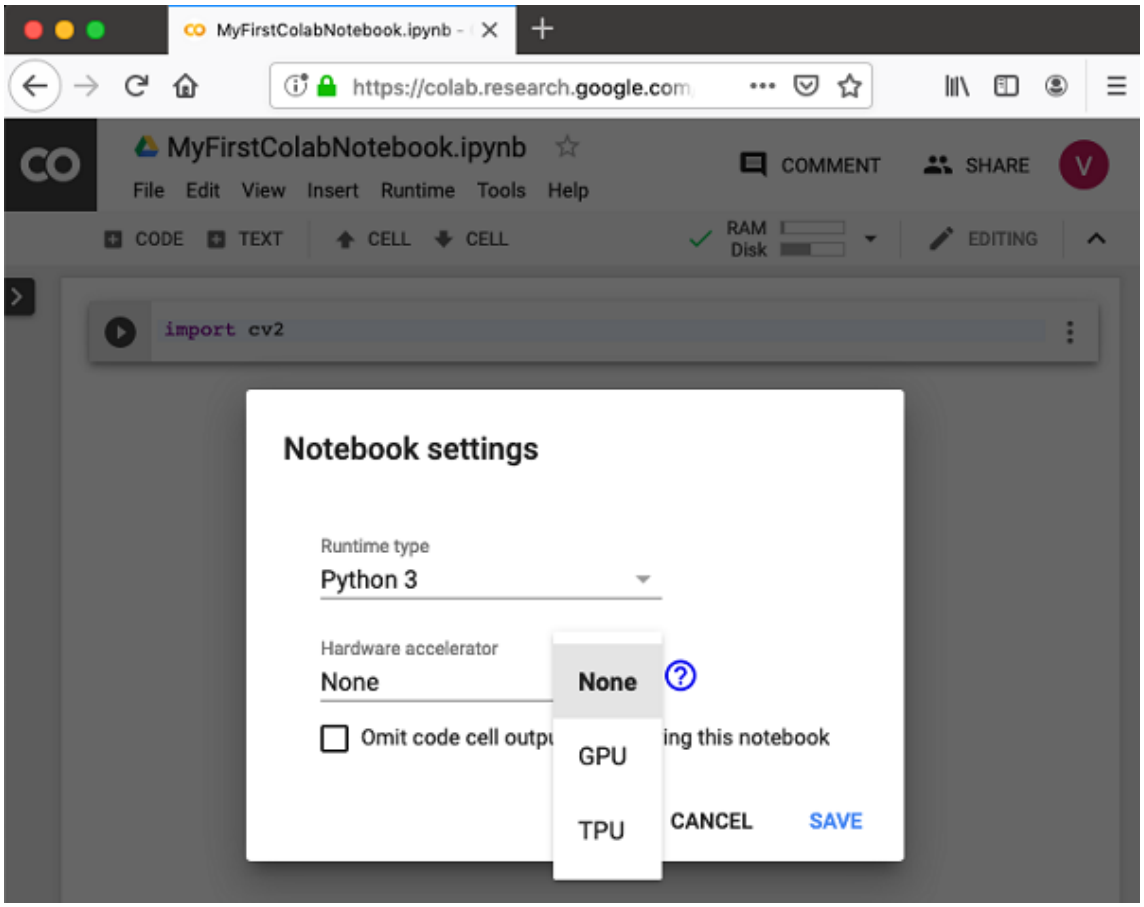


Figure 4.3: Google Colab interface of showing how to enable GPU resources.

However, the code is executed through a virtual machine which is extremely unstable and runs into crush down sometimes. It also has a restriction on GPU resources that is blind to users, which also brings a lot of trouble for training. The GPUs provided in Google Colab are assigned randomly, including NVIDIA Tesla K80, P100, T4, P4, and V100.
Another choice for cloud GPU platform is Matpool, which provides cloud services in artificial intelligence area. It provides multiple choices of GPUs as well as CPUs for renting and possesses different pre-defined environment settings for various kinds of usage. Compared with Google Colab, it costs rent but has much more stable performance. Figure 4.4 shows the interface of Matpool.

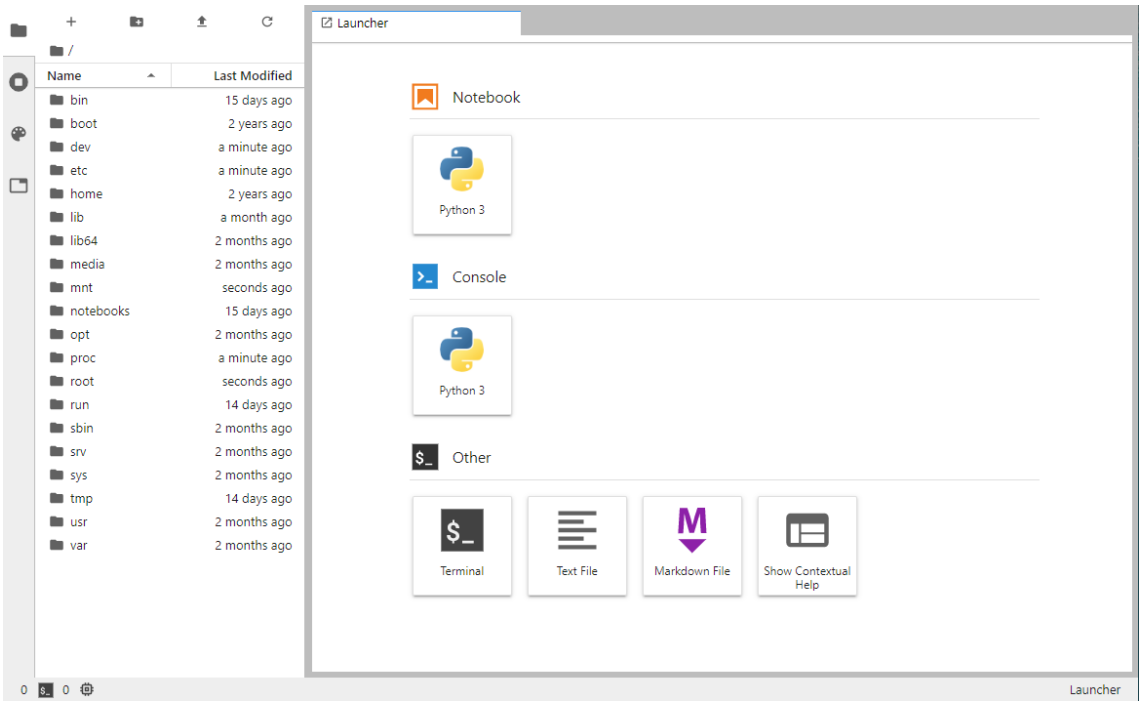


Figure 4.4: Matpool interface.

All results presented in this thesis are gained with Matpool, using NVIDIA GeForce RTX 2080 Ti. It has a Video RAM of 11GB with CPU 6× Xeon E5-2678 v3, and 62GB memory. It takes 26 seconds for each epoch while training models for each experiments, and 3 seconds for each epoch in the fine-tuning part.

4.3.2 Software and libraries

Models used in this thesis are build with Keras, which is an Application Programming Interface (API) integrated with machine learning platform TensorFlow. Keras stands out with its simpleness, flexibility, and power. It provides the sequential models as well as individual blocks for different layers.

4.4 Training information

4.4.1 Training, validation, and test subsets

For MICCAI dataset, we partition the 2D slices from 3D MR scans according to patients into three subsets, 60% for training, 20% for validation and 20% for test. Table 4.2 describes the partition details of MICCAI dataset. For US dataset, we only partition the data into training set for refined training and test set. Table 4.3 describes the partition details of US dataset.
Each experiment repeats for 5 times with different partition according to patients. The US dataset is used as a second test set for training with 2 modalities to see whether the models trained on MICCAI dataset can be generalized to data from different source. Slices in both dataset are adjusted to size of 176x176.

Table 4.2: Partition of 3D MR scans in the MICCAI dataset

Number of total patients	Training set	Validation set	Test set
285	171(60%)	57(20%)	57(20%)

Table 4.3: Partition of 3D MR scans in the US dataset

Number of total patients	Training set	Validation set	Test set
75	15(20%)	0	60(80%)

4.4.2 Training epochs

Experiments in foreground-background training consists of 2 steps. This case does not use early stopping, but has fixed training epochs for each step. In the first step of training, the tumor area is specified by the small ellipse area, which provides not so accurate features to the model. The model learns not only the tumor features but also learns an ellipse shape. In order to avoid model going too far away from the expected way, it is necessary to fix the training epochs. The second step training only takes a small number of annotated GT data without validation set. A fixed number of training epochs can also prevent the model from the over-fitting. Table 4.4 shows the fixed number of training epochs for each step.

	Number of training epochs
Step 1	70
Step 2	150

Table 4.4: Fixed training epochs for scheme trained with FG-BG areas.

4.4.3 Other issues

In order to reduce randomness in experiments, there introduces a random seed to initialize the random number generator. Figure 4.5 shows the settings for random seed.

```
import tensorflow as tf
import random as python_random

np.random.seed(49)
python_random.seed(49)
tf.random.set_seed(49)
```

Figure 4.5: Random seed settings for training process

4.5 Exploring settings for training scheme with foreground and background bounding boxes

4.5.1 Selection of ellipse box sizes

The foreground-background training scheme requires two ellipse-shaped bounding boxes. In order to define the size of bounding boxes, we introduce a parameter **p**, which represents the ratio of drawn ellipse axis length to the length estimated ellipse axis of tumor. The small ellipse should have a $p < 1$, and the large ellipse should have a $p > 1$.
We prefer that the small ellipse contains only tumor pixels, and pixels outside the large ellipse are background pixels. However, since the tumors are irregular, it is inevitable that the ellipse bounding boxes would introduce noise. The size of drawn ellipse (parameter **p**) would affect how much noise each label contains. Table 4.5 and Table 4.6 shows the relationship between the size of p and the introduced noise. In this thesis, we choose

$$p_{large} = 1.2 \tag{4.1}$$

$$p_{small} = 0.9 \tag{4.2}$$

Table 4.5: Relationship between the size of drawn large ellipse and background label noise in training set

p	True background pixels / labeled as background
1.1	0.9798
1.2	0.9935 (chosen)
1.3	0.9976
1.4	0.9990

Table 4.6: Relationship between the size of drawn small ellipse and tumor label noise in training set

p	True tumor pixels / labeled as tumor
0.6	0.9705
0.8	0.9460
0.9	0.9145 (chosen)
1.0	0.8428

4.5.2 Selection of refined training data size

Table 4.7 presents an illustration on how the size of refined training data of MICCAI dataset would affect the test results. In this thesis, 17 patients from training set are used to conduct the refine training, which take up about 10% of the training set. 10% can be viewed as a small enough subset in the original training set.

Table 4.7: Percentage of annotated GT tumor patients used in the refined training and its impact to the test performance on test set. The results in the table used refined training with 30 epochs

Number of patients in the refined training data / Total patients in the training set	Accuracy	Loss	Dice score
0 (no refined training)	0.9670	0.0308	0.8007
0.05 (9/171 patients)	0.9752	0.0227	0.8697
0.1 (17/171 patients) (chosen)	0.9765	0.0211	0.8795
0.2 (34/171 patients)	0.9782	0.0194	0.8926

The 2-modality models are also first refined with 10% of patients from MICCAI training set and then with 20% of US dataset. The refined training set in US data contain 15 patients.

4.6 Segmentation results from models trained with FG-BG ellipse area data

4.6.1 Segmentation results of 4-modality MICCAI dataset

4.6.1.1 Accuracy, loss, and dice score

Experiments are repeated 5 times by re-splitting the training, validation and testing subsets according to patients. Table 4.8 shows the results on MICCAI test subset in 5 runs, in terms of accuracy, loss and dice score. The average dice score is higher than 0.83 in 4-modality case which is acceptable.

run	accuracy	loss	dice score
1	0.9717	0.0252	0.8466
2	0.9750	0.0227	0.8212
3	0.9728	0.0243	0.8303
4	0.9744	0.0233	0.8422
5	0.9730	0.0244	0.8292
avg$\pm\sigma$	**0.9734$\pm$0.0013**	**0.0240$\pm$0.0010**	**0.8339$\pm$0.0103**

Table 4.8: Accuracy, loss, and dice score for FG-BG trained DL scheme on MICCAI test subset with 5 runs, 4-modality case

4.6.1.2 Confusion matrix

Table 4.9 shows the average confusion matrix from FG-BG trained DL scheme on MICCAI test subset, 4-modality case. It shows that over 85% of tumor pixels can be correctly labeled as tumor, which also proves that the proposed training procedure does not introduce many classification errors.

		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9796±0.0046**	0.1443±0.0216
	Tumor	0.0204±0.0046	**0.8557±0.0216**

Table 4.9: Average confusion matrix with standard deviation from FG-BG trained DL scheme on MICCAI test subset, 4-modality case

4.6.1.3 Randomly selected segmented results overlapped on brain images

Figure 4.6 shows the segmented results overlapped on brain images. The segmented results are from scheme trained by FG-BG bounding data and tested on MICCAI dataset, 4-modality case.

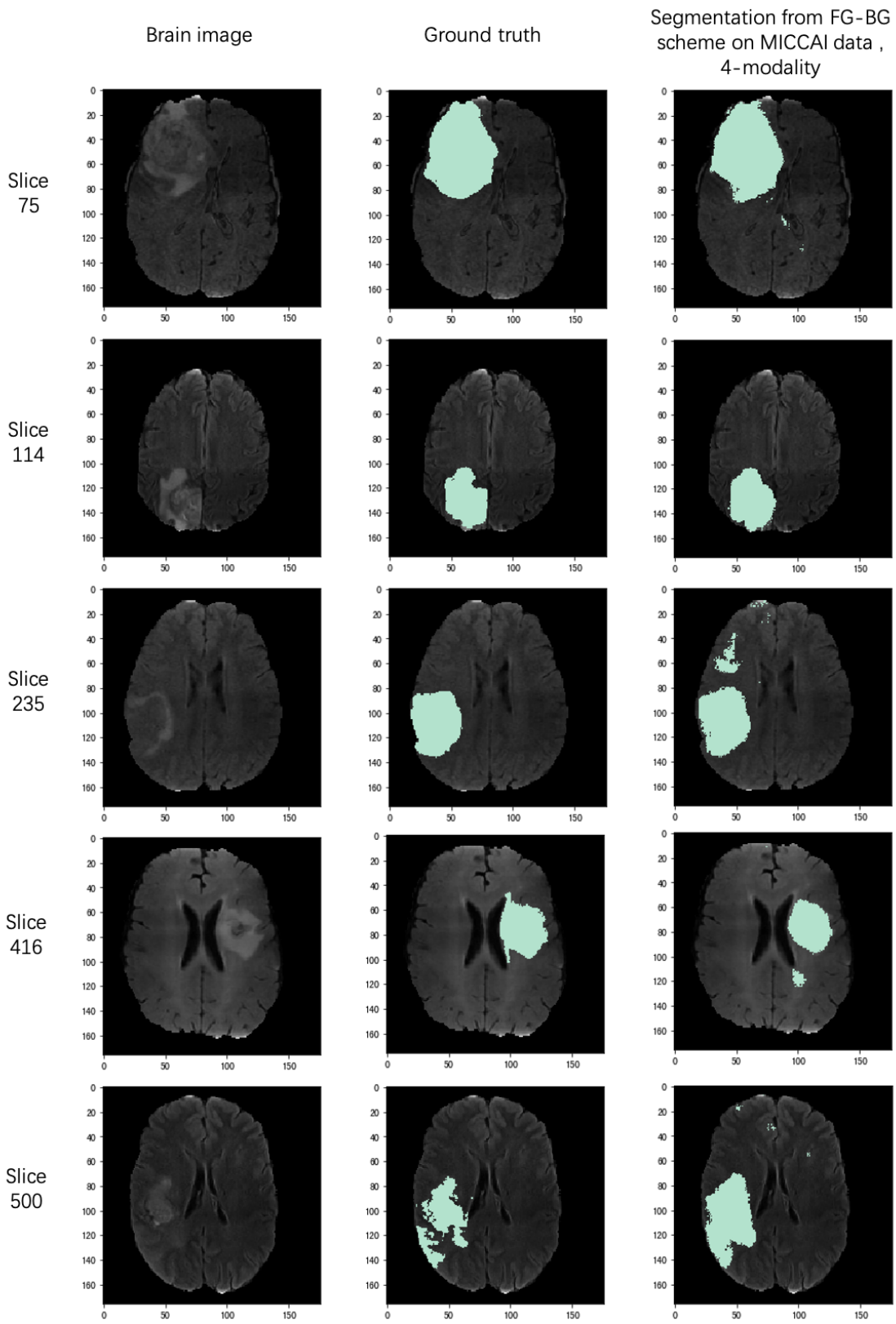
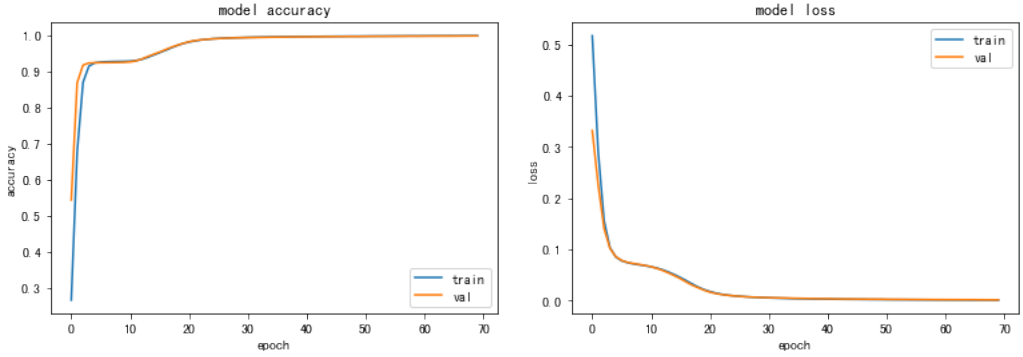


Figure 4.6: Randomly selected segmented results overlapped on brain images. The first column is the original brain images. The second column is the annotated GT overlapped on brain images. The third column is the segmented result from FG-BG scheme on MICCAI data overlapped in brain images, 4-modality case.

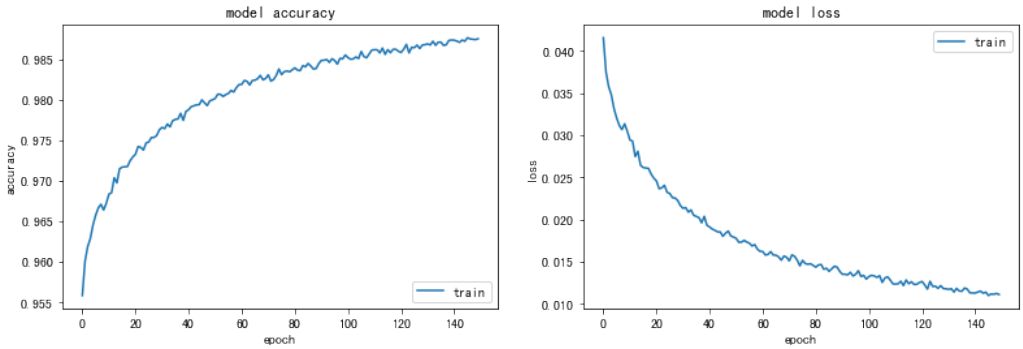
4.6.1.4 Convergence of the training: accuracy/loss in the training and validation as a function of epochs

Figure 4.7 (a) and (b) shows the first step of training on foreground-background labeled data. Figure 4.7 (c) and (d) shows the refined training on a small number of accurate labeled data.
Curves for the first step training shows that the model can reach convergence with FG-BG ellipse area data. For the refine training step, the models stops training before it reaches a convergence point, which is due to the consideration of lacking validation set, risking at encounter over-fitting issue. Extra experiments have been conducted to prove that increasing the fine-tuning epoch to 400 could not provide a better test result.



(a) Initial training accuracy for 4 modalities

(b) Initial training loss for 4 modalities



(c) Refined training accuracy for 4 modalities

(d) Refined training loss for 4 modalities

Figure 4.7: Training accuracy and loss curves for training with foreground and background bounding areas on MICCAI test set, 4-modality case.

4.6.2 Segmentation results of 2-modality MICCAI dataset

4.6.2.1 Accuracy, loss, and dice score

Table 4.10 shows the results on MICCAI test subset in 5 runs, in terms of accuracy, loss and dice score. The average dice score in 2-modality case is 0.81, which shows that 2-modality data can also be used to train a DL model with good efficiency but slightly reduce performance.

run	accuracy	loss	dice score
1	0.9665	0.0311	0.8130
2	0.9726	0.0247	0.8101
3	0.9668	0.0288	0.8106
4	0.9701	0.0213	0.8142
5	0.9719	0.0336	0.8100
avg$\pm\sigma$	**0.9696$\pm$0.0028**	**0.0279$\pm$0.0049**	**0.8116$\pm$0.0019**

Table 4.10: Accuracy, loss, and dice score for FG-BG trained DL scheme on MICCAI test subset with 5 runs, 2-modality case

4.6.2.2 Confusion matrix

Table 4.11 shows the average confusion matrix from FG-BG trained DL scheme on MICCAI test subset, 2-modality case. It shows that over 82% of tumor pixels can be correctly labeled as tumor.

		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9852$\pm$0.0028**	0.1747$\pm$0.0121
	Tumor	0.0148$\pm$0.0028	**0.8253$\pm$0.0121**

Table 4.11: Average confusion matrix with standard deviation from FG-BG trained DL scheme on MICCAI test subset, 2-modality case

4.6.2.3 Randomly selected segmented results overlapped on brain images

Figure 4.8 shows the segmented results overlapped on brain images. The segmented results are from scheme trained by FG-BG bounding data and tested on MICCAI dataset, 2-modality case.

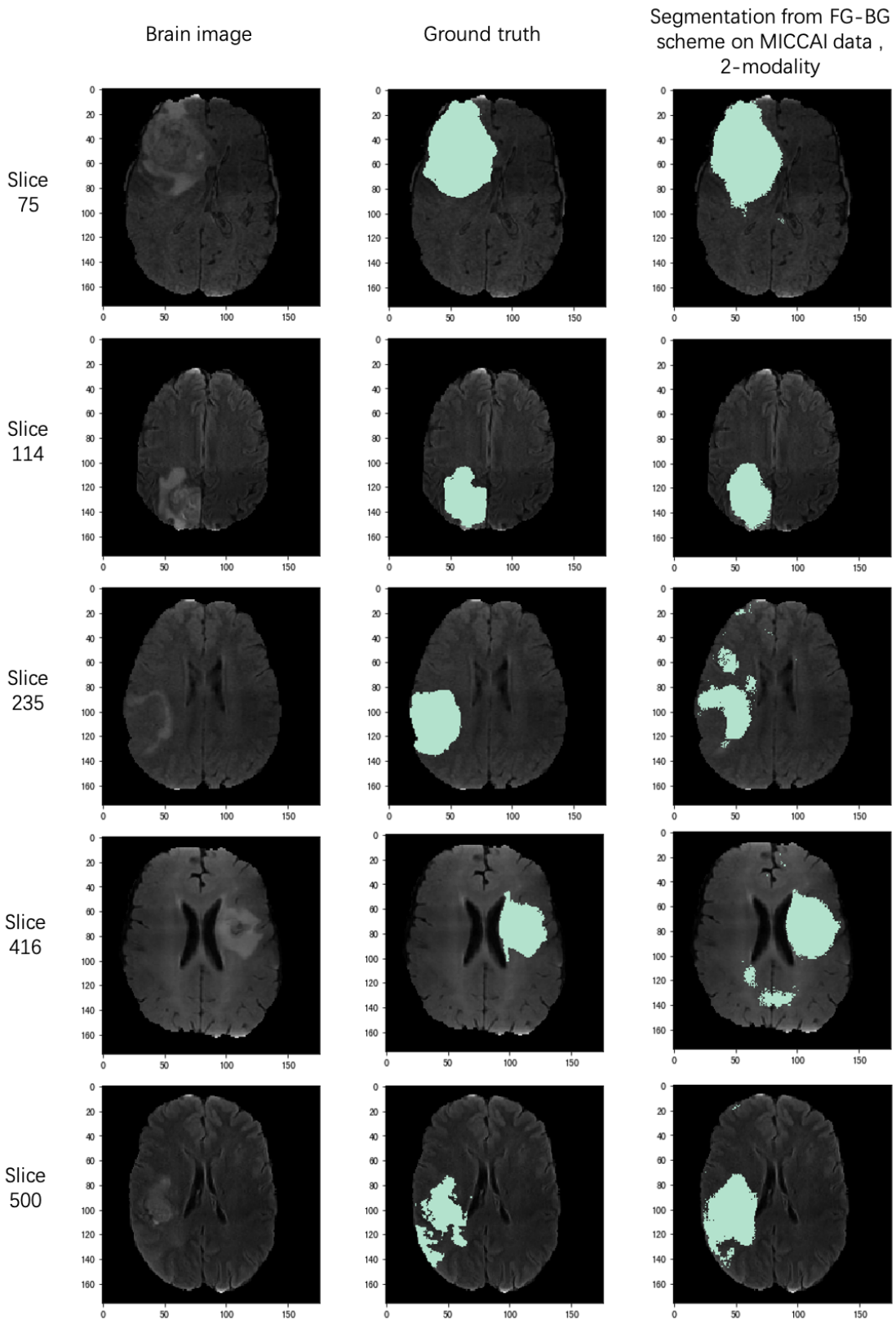
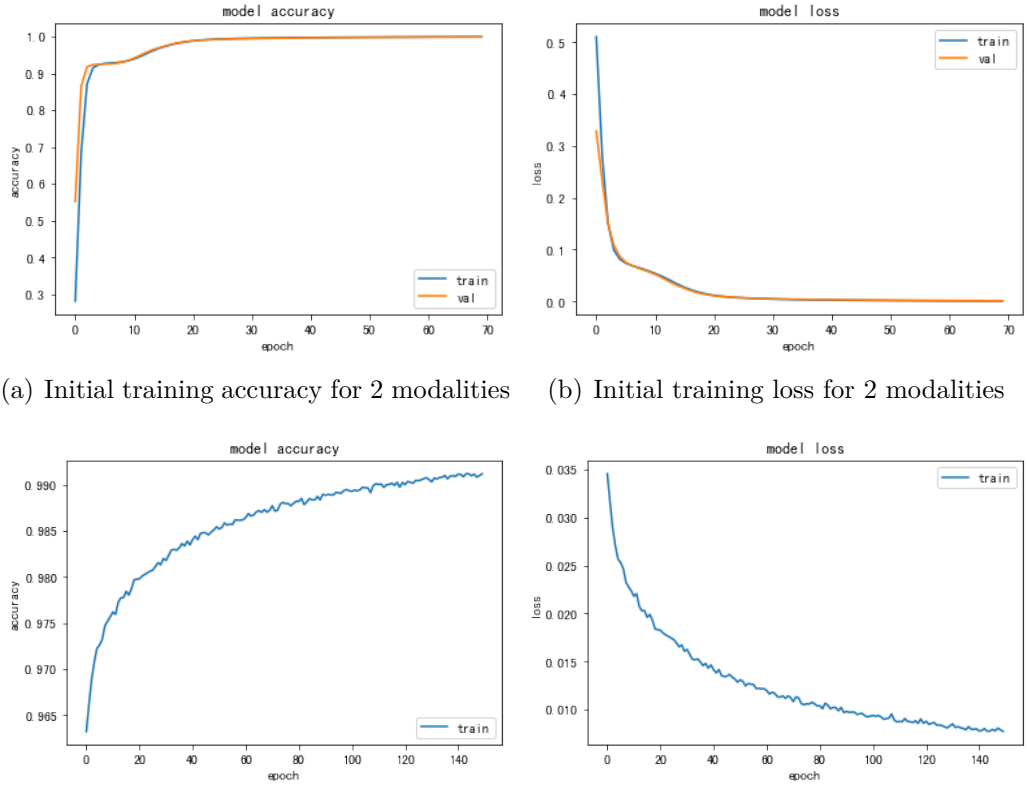


Figure 4.8: Randomly selected segmented results overlapped on brain images. The first column is the original brain images. The second column is the annotated GT overlapped on brain images. The third column is the segmented result from FG-BG scheme on MICCAI data overlapped in brain images, 2-modality case.

4.6.2.4 Convergence of the training: accuracy/loss in the training and validation as a function of epochs

Figure 4.9 (a) and (b) shows the first step of training on foreground-background labeled data. Figure 4.9 (c) and (d) shows the refined training on a small number of accurate labeled data. The training curves prove that model trained with 2-modality data can successfully reach convergence.



(a) Initial training accuracy for 2 modalities

(b) Initial training loss for 2 modalities

(c) Refined training accuracy for 2 modalities

(d) Refined training loss for 2 modalities

Figure 4.9: Training accuracy and loss curves for training with foreground and background bounding areas on MICCAI test set, 2-modality case.

4.6.3 Segmentation results of 2-modality US dataset

Table 4.12 presents the test results on US dataset before and after the second refined training (i.e. on the 20% patients with annotated GT tumors in the US data). Table 4.13 shows the average confusion matrix on US dataset after the second refined training.

The average dice score can reach 0.77 before the second refined training, which proves that model trained with MICCAI dataset maintains the ability of generalising to new dataset. After refined training, the average dice score increases to 0.90.

run	Before refined training with US data			After refined training with US data		
	loss	dice score	accuracy	loss	dice score	accuracy
1	0.0301	0.7558	0.9665	0.0102	0.9070	0.9884
2	0.0233	0.7938	0.9743	0.0096	0.9114	0.9891
3	0.0243	0.7833	0.9712	0.0094	0.9098	0.9894
4	0.0288	0.7654	0.9695	0.0100	0.9083	0.9885
5	0.0270	0.7757	0.9705	0.0103	0.9095	0.9885
avg $\pm\sigma$	**0.0267** **± 0.0031**	**0.7748** **± 0.114**	**0.9704** **± 0.0020**	**0.0099** **± 0.0008**	**0.9092** **± 0.0027**	**0.9888** **± 0.0009**

Table 4.12: Test results on US dataset, trained with FG-BG bounding areas. The pre-trained model contains a first step training on a large number of FG-BG bounding data and the first refined training on a small number of annotated GT data from MICCAI dataset. The table shows the the test results before and after the second refined training with annotated GT US dataset.

		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9928±0.0015**	0.1238±0.0021
	Tumor	0.0072±0.0015	**0.8762±0.0021**

Table 4.13: Average confusion matrix with standard deviation from FG-BG trained DL scheme on US test subset, 2-modality case

4.7 Segmentation results from models trained with FG-BG ellipse area data, adding weights on unbalanced classes

4.7.1 Segmentation results of 4-modality MICCAI dataset

4.7.1.1 Accuracy, loss, and dice score

Table 4.14 presents accuracy, loss, and dice score of 5 runs on MICCAI data test subset in 4-modality case. Comparing with dice scores in previous case, the model performance is slightly improved by adding weights to unbalanced classes. The average dice score in 4-modality case increases from 0.8339 to 0.8407.

run	accuracy	loss	dice score
1	0.9730	0.4413	0.8406
2	0.9757	0.4408	0.8392
3	0.9758	0.4378	0.8420
4	0.9727	0.4420	0.8387
5	0.9720	0.4420	0.8372
avg$\pm\sigma$	**0.9724$\pm$0.0032**	**0.4413$\pm$0.0054**	**0.8407$\pm$0.0018**

Table 4.14: Accuracy, loss, and dice score for FG-BG trained DL scheme on MICCAI test subset with 5 runs adding weights for unbalanced classes, 4-modality case

4.7.1.2 Confusion matrix

Table 4.15 shows the average confusion matrices in MICCAI data test subset.

		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9846$\pm$0.0009**	0.1612$\pm$0.0008
	Tumor	0.0154$\pm$0.0009	**0.8388$\pm$0.0008**

Table 4.15: Average confusion matrix with standard deviation from FG-BG trained DL scheme on MICCAI test subset adding weights for unbalanced classes, 4-modality case

4.7.1.3 Randomly selected segmented results overlapped on brain images

Figure 4.10 shows the segmented results overlapped on brain images. The segmented results are from scheme trained by FG-BG bounding data, adding weights on unbalanced classes, 4-modality case.

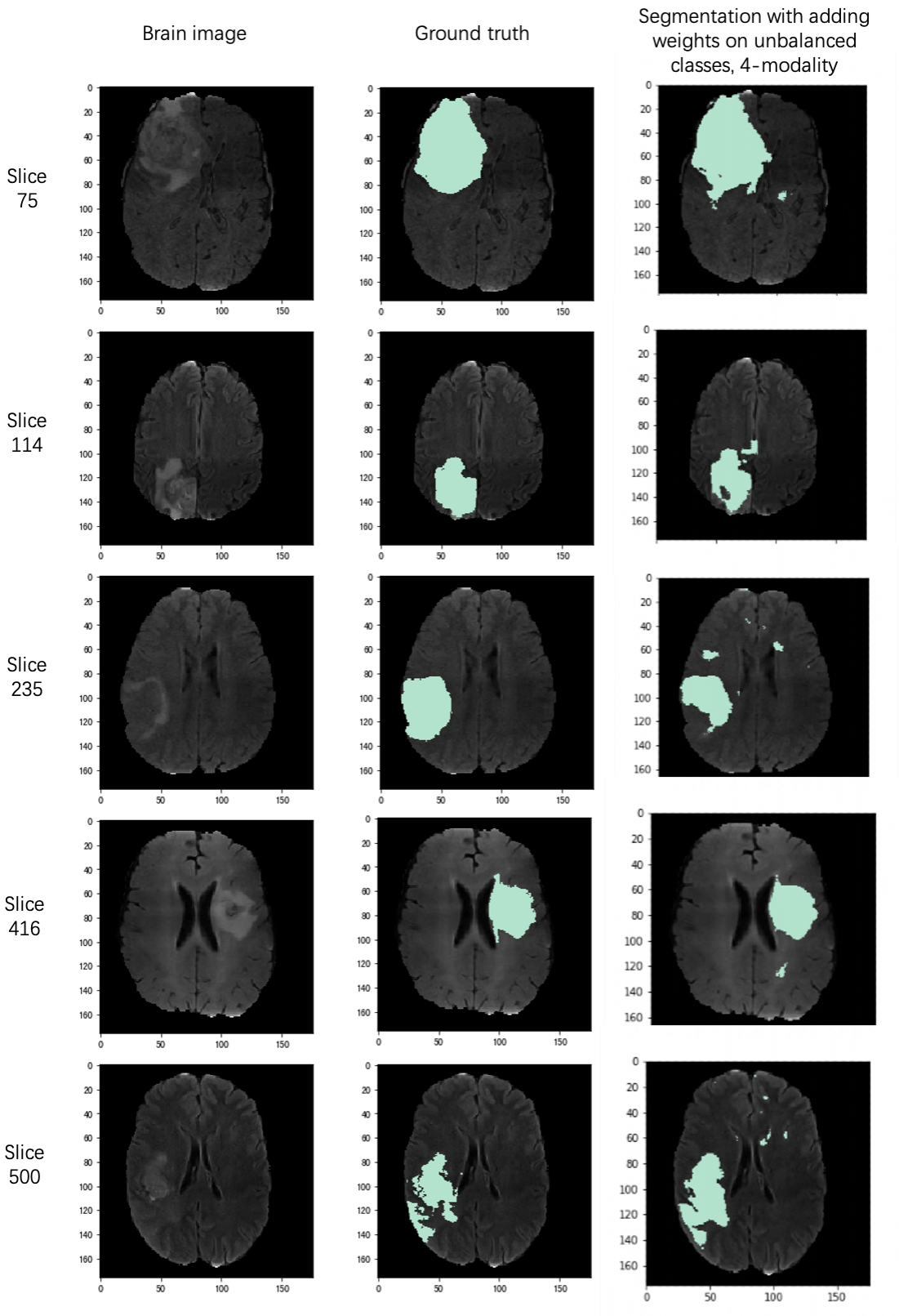
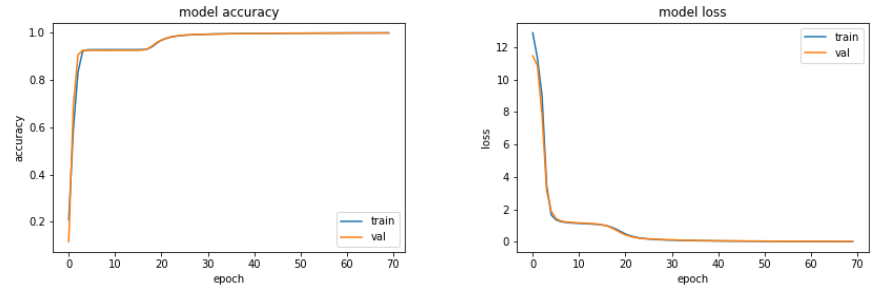


Figure 4.10: Randomly selected segmented results overlapped on brain images. The first column is the original brain images. The second column is the annotated GT overlapped on brain images. The third column is the segmented results on MICCAI data overlapped in brain images, adding weights on unbalanced classes, 4-modality case.

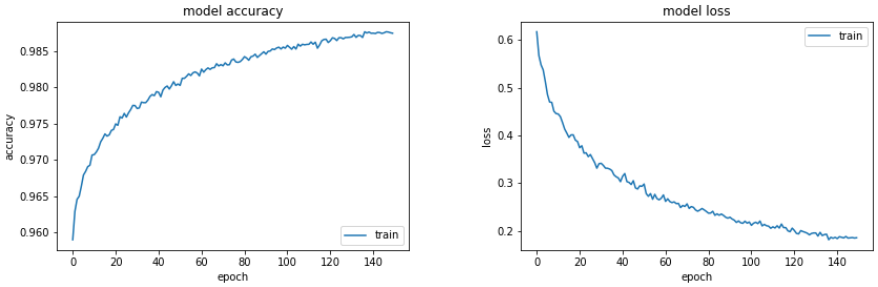
4.7.1.4 Convergence of the training: accuracy/loss in the training and validation as a function of epochs

Figure 4.11 presents training curves for each experiment in this case, from which it could be concluded that these models can successfully converge.



(a) Initial training accuracy for 4 modalities

(b) Initial training loss for 4 modalities



(c) Refined training accuracy for 4 modalities

(d) Refined loss for 4 modalities

Figure 4.11: Training accuracy and loss curves for training with FG-BG bounding areas on MICCAI test set adding weights on unbalanced classes, 4-modality case.

4.7.2 Segmentation results of 2-modality MICCAI dataset

4.7.2.1 Accuracy, loss, and dice score

Table 4.16 presents accuracy, loss, and dice score of 5 runs on MICCAI data test subset in 2-modality case.

run	accuracy	loss	dice score
1	0.9699	0.4851	0.8202
2	0.9705	0.4844	0.8224
3	0.9664	0.4878	0.8192
4	0.9680	0.4876	0.8220
5	0.9662	0.4891	0.8184
avg$\pm\sigma$	**0.9686$\pm$0.0021**	**0.4859$\pm$0.0036**	**0.8200$\pm$0.0017**

Table 4.16: Accuracy, loss, and dice score for FG-BG trained DL scheme on MICCAI test subset with 5 runs adding weights for unbalanced classes, 2-modality case

4.7.2.2 Confusion matrix

Table 4.17 shows the average confusion matrices in MICCAI data test subset. Comparing with previous case, the added weights improve the true positive value for tumor class from 0.8253 to 0.8267.

		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9816±0.0006**	0.1733±0.0024
	Tumor	0.0184±0.0006	**0.8267±0.0024**

Table 4.17: Average confusion matrix with standard deviation from FG-BG trained DL scheme on MICCAI test subset adding weights for unbalanced classes, 2-modality case

4.7.2.3 Randomly selected segmented results overlapped on brain images

Figure 4.12 shows the segmented results overlapped on brain images. The segmented results are from scheme trained by FG-BG bounding data, adding weights on unbalanced classes, 2-modality case.

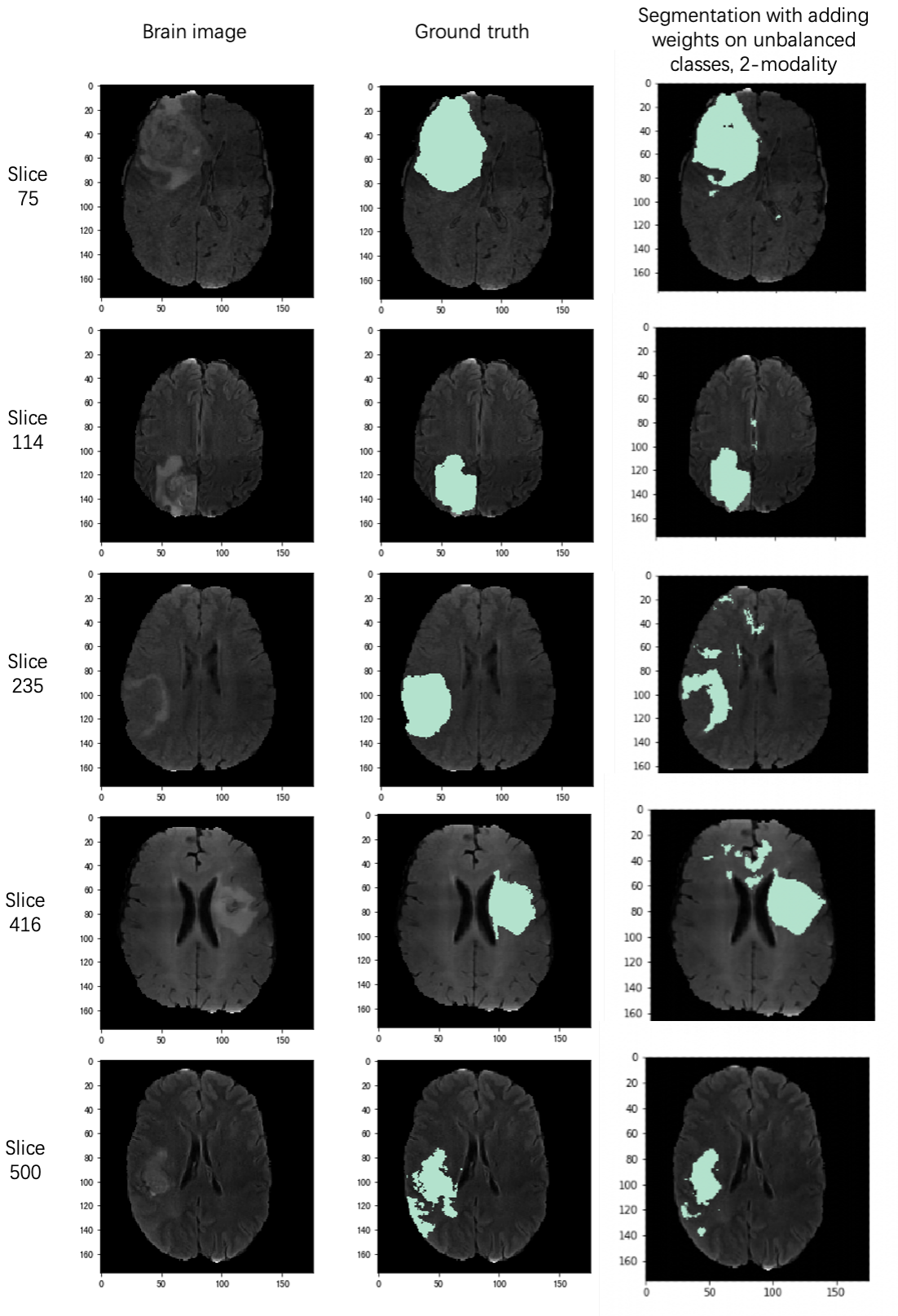
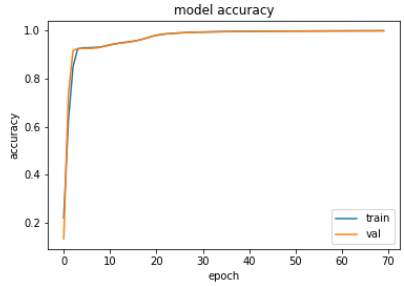


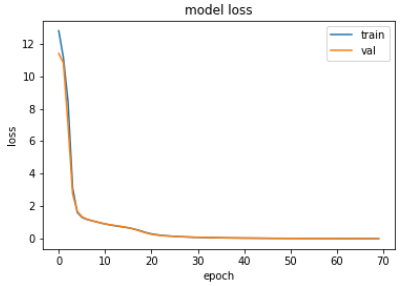
Figure 4.12: Randomly selected segmented results overlapped on brain images. The first column is the original brain images. The second column is the annotated GT overlapped on brain images. The third column is the segmented results on MICCAI data overlapped in brain images, adding weights on unbalanced classes, 2-modality case.

4.7.2.4 Convergence of the training: accuracy/loss in the training and validation as a function of epochs

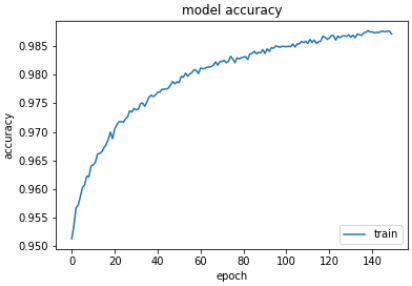
Figure 4.13 presents training curves for each experiment in this case, from which it could be concluded that the models can successfully converge.



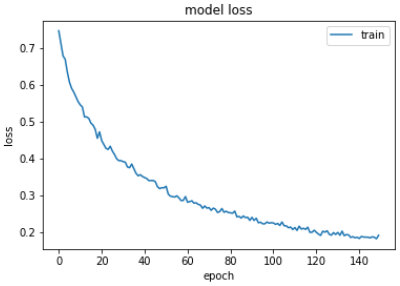
(a) Initial training accuracy for 2 modalities



(b) Initial training loss for 2 modalities



(c) Refined training accuracy for 2 modalities



(d) Refined loss for 2 modalities

Figure 4.13: Training accuracy and loss curves for training with FG-BG bounding areas on MICCAI test set adding weights on unbalanced classes, 2-modality case.

4.7.3 Segmentation results of 2-modality US dataset

Table 4.18 presents the test results on US dataset before and after the second refined training. Table 4.19 shows the average confusion matrix after the second refined training.

run	Before refined training with US data			After refined training with US data		
	loss	dice score	accuracy	loss	dice score	accuracy
1	0.0275	0.7604	0.9696	0.0090	0.9082	0.9907
2	0.0210	0.7987	0.9776	0.0084	0.9126	0.9912
3	0.0218	0.7881	0.9741	0.0081	0.9113	0.9915
4	0.0266	0.7701	0.9705	0.0082	0.9092	0.9908
5	0.0222	0.7802	0.9737	0.0080	0.9107	0.9907
avg $\pm\sigma$	**0.0243** **± 0.0019**	**0.7795** **± 0.0130**	**0.9736** **± 0.0024**	**0.0087** **± 0.0003**	**0.9104** **± 0.0021**	**0.9910** **± 0.0004**

Table 4.18: Test results on US dataset, trained with FG-BG bounding areas, adding weights on unbalanced classes. The pre-trained model contains a first step training on a large number of FG-BG bounding data and the first refined training on a small number of annotated GT data from MICCAI dataset. The table shows the the test results before and after the second refined training with annotated GT US dataset.

		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9958±0.0027**	0.1153±0.0034
	Tumor	0.0042±0.0027	**0.8847±0.0034**

Table 4.19: Average confusion matrix with standard deviation from FG-BG trained DL scheme on US test subset, after adding weights for unbalanced classes, 2-modality case

4.8 Performance comparison: segmented results from scheme trained with FG-BG ellipse bounding area vs. annotated ground truth tumor area.

4.8.1 Comparison on dice scores and accuracy in MICCAI and US test sets

Table 4.20 and Figure 4.14 compares the dice scores in MICCAI data test set and US data test set between scheme trained with FG-BG bounding data and scheme trained with annotated GT data. Compared with models trained with annotated

GT data, dice scores from models trained with FG-BG bounding data decrease about 6% which is rather small.

Table 4.21 and Figure 4.15 compares the average accuracy in MICCAI data test set and US data test set between scheme trained with FG-BG bounding data and scheme trained with annotated GT data.

Table 4.22 and Figure 4.16 compares the average accuracy of tumor class in MICCAI data test set and US data test set between scheme trained with FG-BG bounding data and scheme trained with annotated GT data.

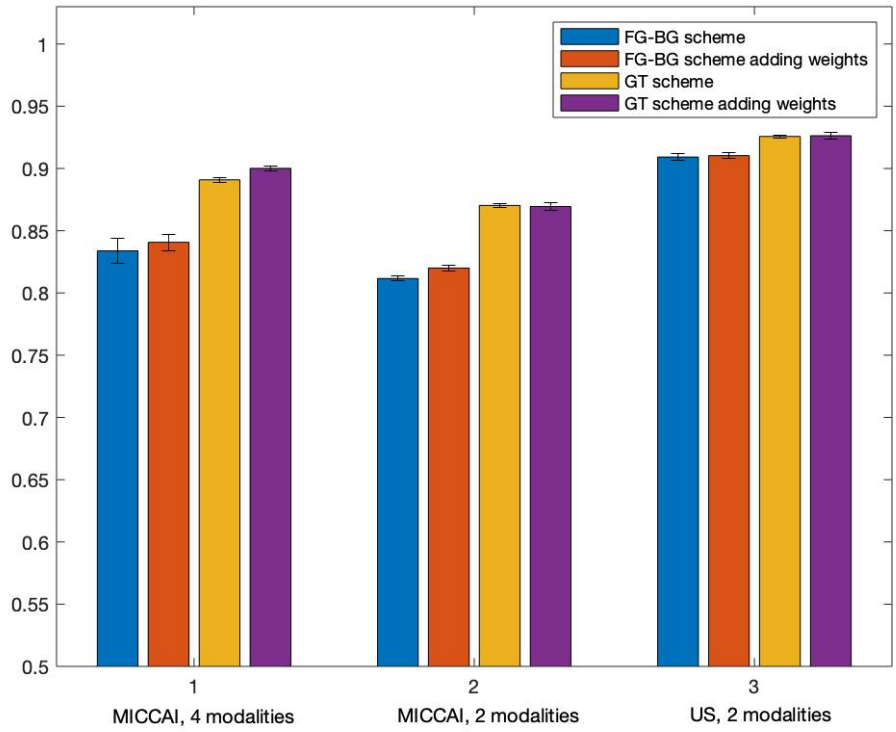


Figure 4.14: Comparison of dice scores in bar plot format. The first cluster contains dice results in MICCAI 4-modality case. The second cluster contains dice results in MICCAI 2-modality case. The second cluster contains dice results in US 2-modality case. It compares the average dice scores in FG-BG scheme and GT scheme before and after adding weights on unbalanced classes. The value in each bar is specified in Table 4.20.

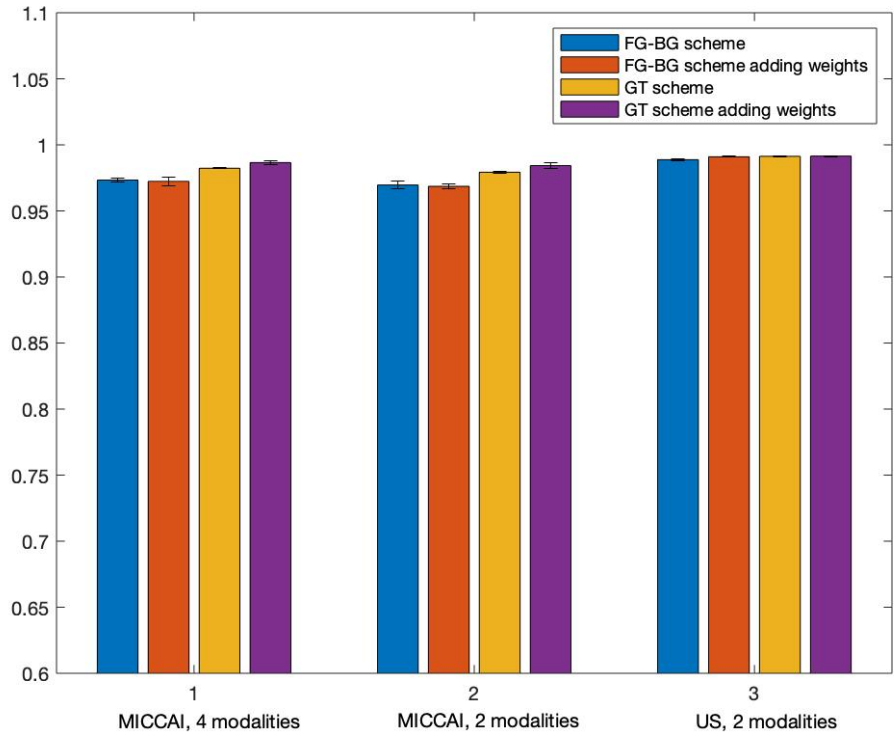


Figure 4.15: Comparison of average accuracy in bar plot format. The first cluster contains dice results in MICCAI 4-modality case. The second cluster contains dice results in MICCAI 2-modality case. The second cluster contains dice results in US 2-modality case. It compares the average accuracy in FG-BG scheme and GT scheme before and after adding weights on unbalanced classes. The value in each bar is specified in Table 4.21

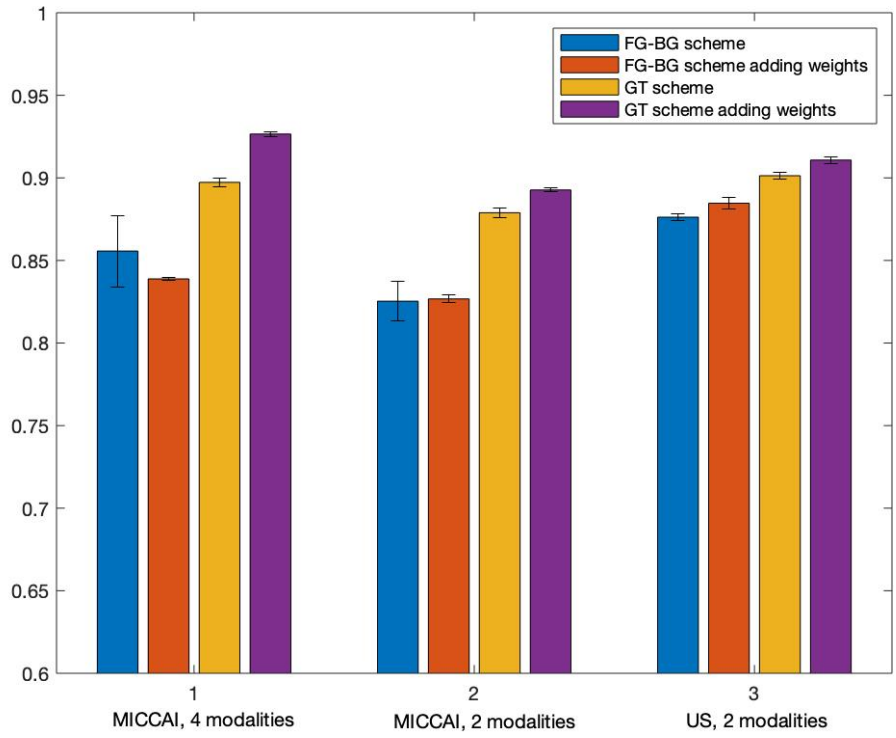


Figure 4.16: Comparison of average tumor accuracy in bar plot format. The first cluster contains dice results in MICCAI 4-modality case. The second cluster contains dice results in MICCAI 2-modality case. The second cluster contains dice results in US 2-modality case. It compares the average accuracy in FG-BG scheme and GT scheme before and after adding weights on unbalanced classes. The value in each bar is specified in Table 4.22

	MICCAI 4-modality case	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.8339±0.0103	0.8407±0.0006
Trained with annotated GT data	0.8909±0.0018	0.9001±0.0018
	MICCAI 2-modality case	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.8116±0.0019	0.8200±0.0024
Trained with annotated GT data	0.8703±0.0017	0.8695±0.0031
	US 2-modality case refined twice	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.9092±0.0027	0.9104±0.0021
Trained with annotated GT data	0.9257±0.0011	0.9263±0.0024

Table 4.20: Comparison of average dice scores between FG-BG scheme and GT scheme on MICCAI test set and US test set.

	MICCAI 4-modality case	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.9734±0.0013	0.9724±0.0032
Trained with annotated GT data	0.9823±0.0003	0.9866±0.0012
	MICCAI 2-modality case	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.9696±0.0028	0.9686±0.0021
Trained with annotated GT data	0.9793±0.0007	0.9843±0.0020
	US 2-modality case refined twice	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.9888±0.0009	0.9910±0.0004
Trained with annotated GT data	0.9913±0.0001	0.9915±0.0003

Table 4.21: Comparison of average accuracy between FG-BG scheme and GT scheme on MICCAI test set and US test set.

	MICCAI 4-modality case	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.8557±0.0216	0.8388±0.0008
Trained with annotated GT data	0.8973±0.0028	0.9266±0.0015
	MICCAI 2-modality case	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.8253±0.0121	0.8267±0.0024
Trained with annotated GT data	0.8789±0.0027	0.8927±0.0013
	US 2-modality case refined twice	
	before adding weights	after adding weights
Trained with FG-BG bounding data	0.8762±0.0021	0.8847±0.0034
Trained with annotated GT data	0.9013±0.0019	0.9108±0.0021

Table 4.22: Comparison of average tumor accuracy between FG-BG scheme and GT scheme on MICCAI test set and US test set.

4.8.2 Comparison on confusion matrices

Table 4.23 and Table 4.24 compare the average confusion matrices on MICCAI data test subset between scheme trained with foreground-background bounding area and scheme trained with annotated ground truth area. And Table 4.25 compares the average confusion matrices on US data test set.

The true positive classification for background pixels remains almost the same. Classification performance on tumor pixels compromises a bit with data labeled with foreground and background bounding areas.

FG-BG scheme		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9846±0.0009**	0.1612±0.0008
	Tumor	0.0154±0.0009	**0.8388±0.0008**
GT scheme		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9886±0.0005**	0.0734±0.0025
	Tumor	0.0114±0.0005	**0.9266±0.0015**

Table 4.23: Comparison of average confusion matrix between scheme trained with foreground-background bounding area and scheme trained with annotated ground truth area, 4-modality case, MICCAI test subset.

FG-BG scheme		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9816±0.0006**	0.1733±0.0024
	Tumor	0.0184±0.0006	**0.8267±0.0024**
GT scheme		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9850±0.0012**	0.1094±0.0015
	Tumor	0.0150±0.0012	**0.8906±0.0015**

Table 4.24: Comparison of average confusion matrix between scheme trained with foreground-background bounding area and scheme trained with annotated ground truth area, 2-modality case, MICCAI test subset.

FG-BG scheme		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9858±0.0027**	0.1153±0.0034
	Tumor	0.0042±0.0027	**0.8847±0.0034**
GT scheme		Actual classes	
		Background	Tumor
Predicted classes	Background	**0.9973±0.0014**	0.1073±0.0013
	Tumor	0.0027±0.0014	**0.8927±0.0013**

Table 4.25: Comparison of average confusion matrix between scheme trained with foreground-background bounding area and scheme trained with annotated ground truth area, 2-modality case, US test subset.

4.8.3 Comparison on visual segmented results

Concluding from the randomly selected segmented results in Figure 4.17 and Figure 4.18, the FG-BG scheme managed to give reasonable segmentation results for tumor but with more noise. It tends to label more background pixels as tumor in most cases.

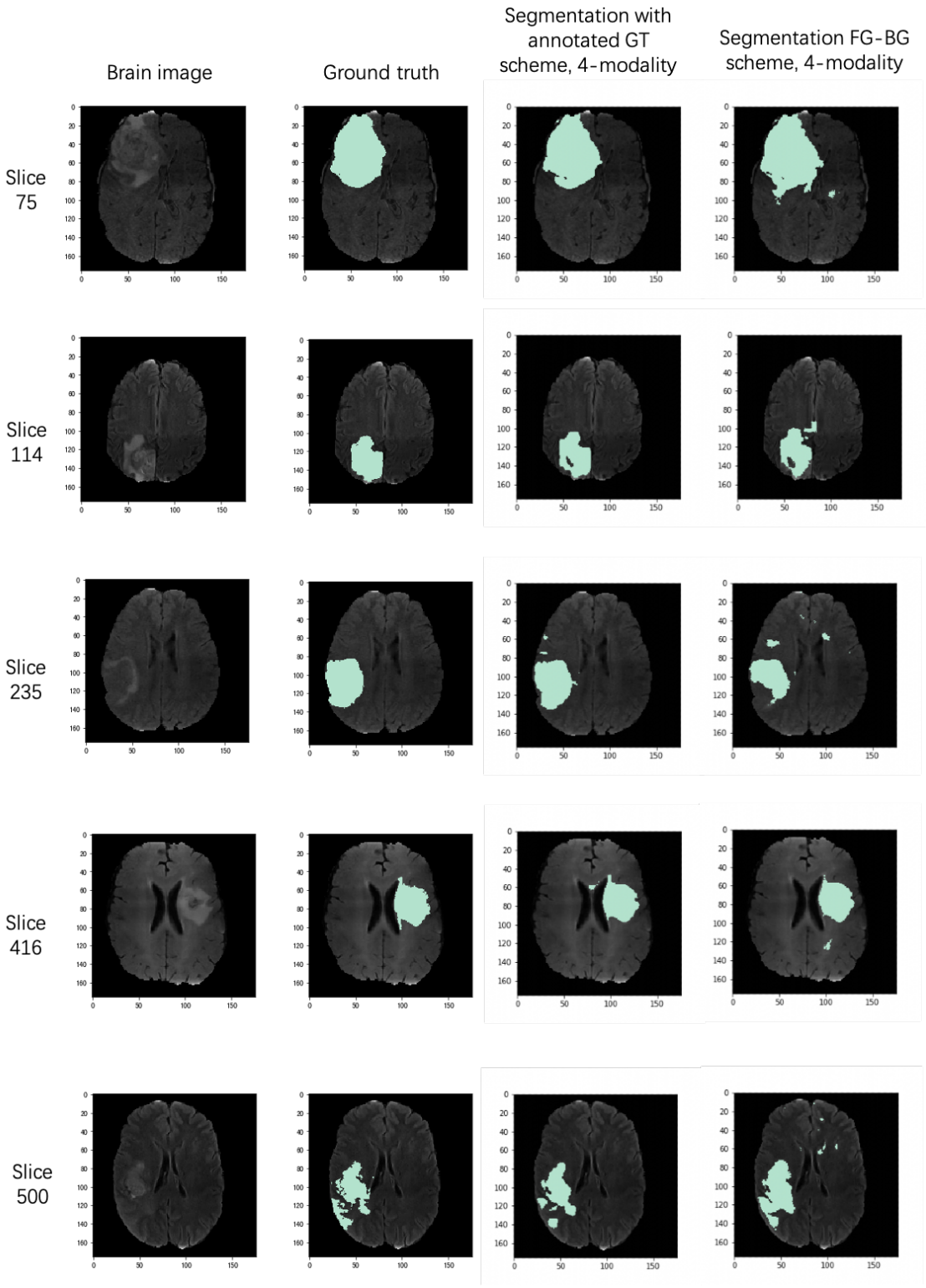


Figure 4.17: Randomly selected segmented results overlapped on brain images. The first column is the original brain images. The second column is the annotated GT overlapped on brain images. The third column is the segmented results from GT scheme, 4-modality. The forth column is the segmented results from FG-BG scheme, 4-modality, MICCAI test subset.

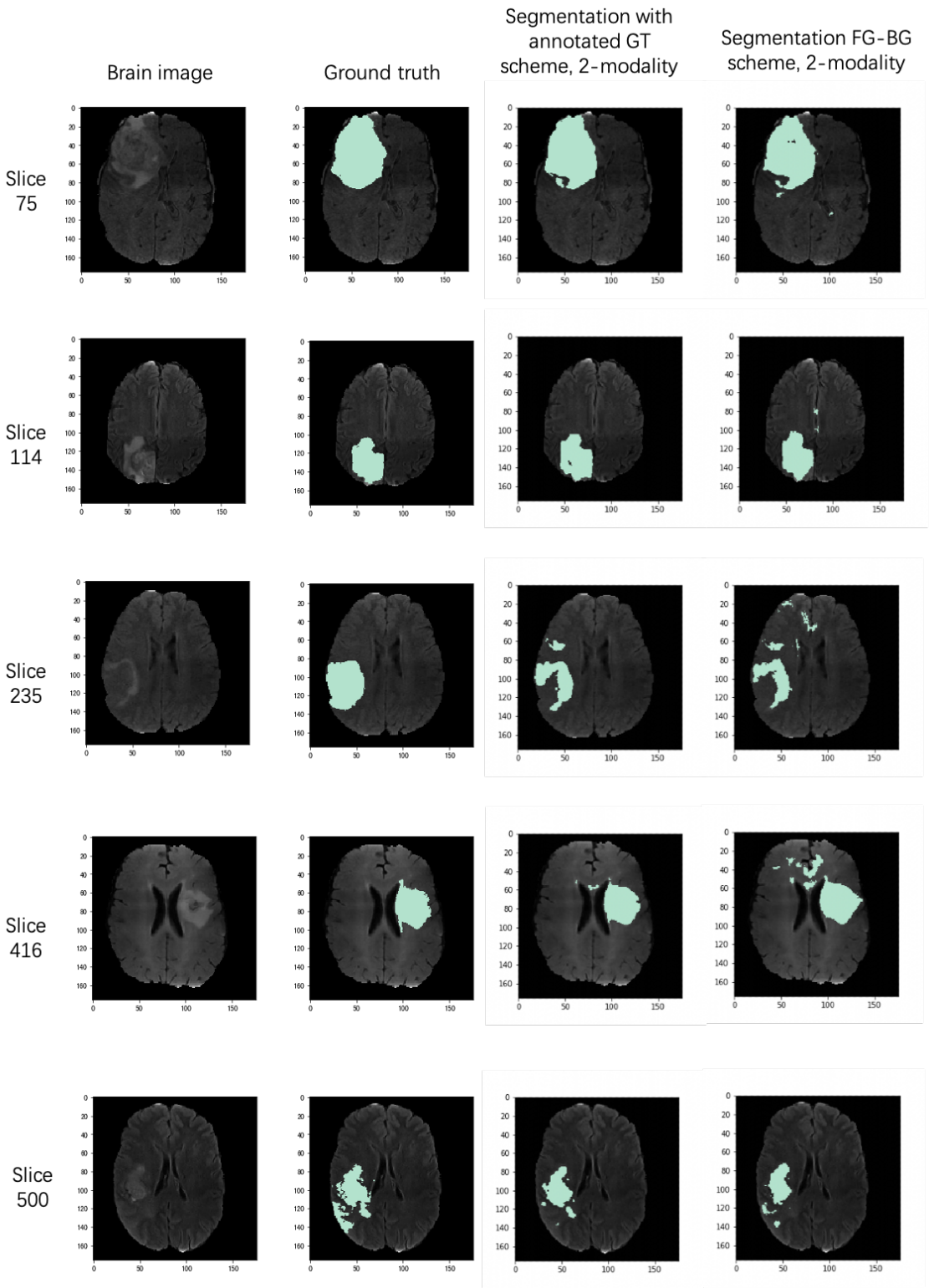


Figure 4.18: Randomly selected segmented results overlapped on brain images. The first column is the original brain images. The second column is the annotated GT overlapped on brain images. The third column is the segmented results from GT scheme, 2-modality. The forth column is the segmented results from FG-BG scheme, 2-modality, MICCAI test subset.

5
Discussion

5.1 Results

This thesis has been focused on exploring training deep learning model with foreground-background data instead of annotated ground truth data. It proposed a method of labeling tumor area and background area with bounding boxes and discarding the detailed boundaries of tumor. The model is first trained with a large number of foreground-background labeled data and then refined with a small number of accurate data. In order to adjust dataset with different situations, the models are trained with 4-modality data and 2-modality data from MICCAI dataset separately. Results from each model proves that such labeling method and training procedure can successfully give pixel-wise results. The prediction examples are randomly selected and distinctly present a pleasing prediction results.
This thesis also conduct for handling unbalanced classes issue. It assign different weights on tumor and background classes during training. This case compares different models from previous cases with and without compensation for unbalanced classes, and both test dice scores and confusion matrices give obvious improvements. The weighted dice loss function achieves to focus neural network on tumor area thus increases the accuracy of tumor pixels classification.

5.2 Limitations and future work

This thesis only aims at exploring a new labeling method, not pursuing perfect segmentation results, thus maintain lots of limitations. There is still a lot of work that need to be explored.
Experiments in this thesis are based on a 2D mult-istream U-net and feature- level fusion. Although single stream U-net is rather standardized, multi-stream network and fusion for medical image segmentation can still be improved in many ways. Due to the time limitation of this thesis work, we have not exploited these options.
The proposed method uses two ellipse areas for foreground tumor area and background normal tissue area. With the help of a small number of annotated ground truth tumor boundary data manually drawn by medical experts, we are able to generate good segmentation results. In our tests, the results were further improved by adding weights to the unbalanced classes. Further improvement can be made on how to generate best estimate on the weights.

6

Conclusion

Brain tumor segmentation is an important task in medical image analysis. Deep learning methods for brain tumor segmentation usually require large amounts for annotated ground truth data for supervised learning. Usually tumor annotation requires medical experts and is time consuming. . To overcome this problem, this thesis proposed a novel method, where the input for supervised training of a multi-stream U-net are the foreground-background areas specified by 2 ellipse bounding boxes, i.e., foreground tumor area by the interior area of small ellipse, and background normal tissue area by the exterior area of the larger ellipse. The DL network initially trained by FG-BG areas is then refined by further training from a small number of patients (e.g. 10%) with annotated ground truth tumors (marked by medical experts). Since the US dataset is very small, we used FG-BG trained DL network by 2 modality MICAII dataset followed by a further refined training using 20% data (i.e. 15 patients) in US dataset. We also studied an approach to further improve the segmentation results by adding weights to the unbalanced classes during the training.
Experiments conducted on 2 datasets (i.e., MICCAI and US) have shown that very good performance (average accuracy 0.9734 on 5 runs for MICCAI dataset and average accuracy 0.9888 on 5 runs for US dataset). Experiments have also shown that adding weights to imbalanced classes further improved the performance (average accuracy 0.9724 on 5 runs for MICCAI dataset, dice score 0.9910 on 5 runs for US dataset). Discussion on further improvement is also included.

Bibliography

[1] Department of computing, imperial college london. http://www.imperial.ac.uk/computing/, May 2021.

[2] Muhammad Khan, Muhammad Rashid, Muhammad Sharif, Kashif Javed, and Tallha Akram. Classification of gastrointestinal diseases of stomach from wce using improved saliency-based method and discriminant features selection. *Multimedia Tools and Applications*, 78, 10 2019.

[3] Sang-gil Lee, Jae Seok Bae, Hyunjae Kim, Jung Hoon Kim, and Sungroh Yoon. Liver lesion detection from weakly-labeled multi-phase CT volumes with a grouped single shot multibox detector. *CoRR*, abs/1807.00436, 2018.

[4] A. Arnab, Shuai Zheng, Sadeep Jayasumana, B. Romera-Paredes, Måns Larsson, Alexander Kirillov, Bogdan Savchynskyy, C. Rother, F. Kahl, and Philip H. S. Torr. Conditional random fields meet deep neural networks for semantic segmentation. 2017.

[5] Jose Dolz, Christian Desrosiers, and Ismail Ben Ayed. Ivd-net: Intervertebral disc localization and segmentation in MRI with a multi-modal unet. *CoRR*, abs/1811.08305, 2018.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[7] Brain tumor: Introduction. https://www.cancer.net/cancer-types/brain-tumor/introduction, February 2021.

[8] Brain tumors. https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Brain-Tumors, February 2021.

[9] Understanding brain tumors. https://braintumor.org/brain-tumor-information/understanding-brain-tumors/, February 2021.

[10] Chung-Ming Chen, Yi-Hong Chou, Norio Tagawa, and Younghae Do. Computer-aided detection and diagnosis in medical imaging. *Computational and mathematical methods in medicine*, 2013:790608, 09 2013.

[11] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[12] R. Nock and F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458, 2004.

[13] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.

[14] Nils Plath, Marc Toussaint, and Shinichi Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *ICML*, pages 817–824, 2009.

[15] Yan Shen and Mingchen Gao. Brain tumor segmentation on MRI with missing modalities. *CoRR*, abs/1904.07290, 2019.

[16] Mohammad Havaei, Nicolas Guizard, Nicolas Chapados, and Yoshua Bengio. Hemis: Hetero-modal image segmentation. *CoRR*, abs/1607.05194, 2016.

[17] Rihuan Ke, Aurélie Bugeau, Nicolas Papadakis, Peter Schütz, and Carola-Bibiane Schönlieb. A multi-task u-net for segmentation with lazy labels. *CoRR*, abs/1906.12177, 2019.

[18] Sajad Ranjbar, Fereidoon Moghadas Nejad, Hamzeh Zakeri, and Amir H. Gandomi. 3 - computational intelligence for modeling of asphalt pavement surface distress. In Pijush Samui, Dookie Kim, Nagesh R. Iyer, and Sandeep Chaudhary, editors, *New Materials in Civil Engineering*, pages 79–116. Butterworth-Heinemann, 2020.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[20] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *CoRR*, abs/1606.04797, 2016.