U-Net for Cerebral Cortical MR Image Segmentation

Chenchao Xu

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2018

Committee:

John Gennari

Linda Shapiro

Program Authorized to Offer Degree:
Department of Biomedical Informatics and Medical Education

University of Washington

Abstract

U-Net for Cerebral Cortical MR Image Segmentation

Chenchao Xu

Chair of the Supervisory Committee:
Associate Professor John Gennari
Biomedical Informatics and Medical Education

Cerebral cortex segmentation from three-dimensional structural Magnetic Resonance (MR) brain images plays an important role in measuring loss of cortical tissues for disorders such as Alzheimer's disease (AD). U-Net, a type of deep convolutional neural networks architecture, is a widely-used approach for biomedical image segmentation in recent years.

In this thesis, I implemented 2D/3D U-Net on MR images from 20 patients with labeled cerebral tissues and regions. A two-stage pipeline was designed for this task. In stage one, U-Net aims to generate a mask of grey matter to filter out other tissues in brain MRI images. In stage two, a similar U-Net architecture is used to label cerebral cortex sub-regions from images which only contains grey matter. Both 2D U-Net and 3D U-Net do not work for labeling gyri/sulci, and only achieve approximate 55% Dice overlap for labeling cortex regions. In contrast, the cortical segmentation package in FreeSurfer achieves over 90% Dice overlap for labeling gyri/sulci by using a graphical-based probabilistic estimation method with prior information.

I believe that the main reason of poor performance of 2D/3D U-Net is the loss of spacial information of pixels/voxels by cutting original MR images into small parts. The U-Net architecture does not seem to work well for handling high resolution 3D images with imbalanced number of classes. For feature work, researchers could create hybrid methods to combine deep neural networks with prior information to label cerebral cortical sub-regions.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would first like to express my gratitude to my advisor Prof. John Gennari from the Department of Biomedical Informatics and Medical Education, School of Medicine at University of Washington, Seattle. He is always patient to discuss with me on my work every week. He is really responsible. He spent his leisure time reading my writing and made plenty of useful comments. It is his technical and editorial advice that help me complete my master thesis and defense.

I would also like to thank to Prof. Linda Shapiro from the Department of Computer Science and Engineering, School of Engineering at University of Washington, Seattle. She is an expert in machine learning and computer vision. We set a meeting to discuss my work. She also introduced me to her PHD student Sachin Mehta. I have little background in brain MR images. Sachin helped me start and gave several good ideas.

I would also like to thank to Harkirat, Prof. Gennaris PHD student. Her work on Alzheimers Disease provids much background to me in this area.

And my thanks also go to Lora E. Brewsaugh, my master program advisor. She payed much attention to my procedure of defense and graduation. She is a responsible program advisor during my master study.

I would also like to thank to Mindboggle 101 data set for its public MR brain with ground truth.

Last but not least, I would like to thank my parents for their financial and emotional support for my oversea study in University of Washington, Seattle.

Author

Chenchao Xu

Chapter 1

INTRODUCTION

Image segmentation, which is an important task in clinical applications, is the first step in most computer-aided image analysis systems. It can significantly influence results of the whole system because other steps like 3D construction and disease prediction are based on accurate segmentation results. For patients suffering from Alzheimer's disease (AD) or mild cognitive impairment (MCI), the accurate segmentation of cerebral cortex from brain images can measure the loss of neurons to help make the decision which stage of disease they are in. Further segmentation of lobes or other sub-cortex regions may help find the change of areas corresponding to specific functions like vision and auditory. In recent years, deep convolutional neural networks have been widely used for biomedical image segmentation of various tasks for organs or tissues like liver [14], brain tumor [25] and vessel [49]. U-Net [66] is a type of deep convolutional neural networks architecture. It contains an encoding part for image analysis and a following decoding part to generate full-resolution segmentation for two-dimensional input images. With this thesis work, I try to implement 2D U-Net and 3D U-Net for labeling cerebral cortex sub-regions from brain MR images.

The brain is the most important and complex organ in human body. It takes responsibility of control, learning, language, sensory, memory, etc. [28] The brain can be divided into four main areas, cerebrum, cerebellum, brain stem and limbic system. The cerebrum, which is the largest area in the brain, is made up of cerebral cortex and several sub-cortex structures like basal ganglia and olfactory bulb. It is associated with high-level brain functions like action and thought. The cerebellum is also known as little brain. It has two hemispheres underneath cerebral hemispheres. The cerebellum takes responsibility of motor control and language. The brain stem consists of midbrain, pons and medulla. It controls several basic

body functions like breathing, heart beat and blood pressure. The brain stem also connects the rest of brain to the spinal cord so that the brain is able to send message to other organs in the motor and sensory systems. The limbic system consists of thalamus, hypothalamus, amygdala and hippocampus. It plays an important role in emotions and memory.

Among these parts, cerebral cortex is the most developed area in the human brain. Cerebral cortex in a human adult contains over 20 billion neocortical neurons. [61] It is the cerebral cortex that most strongly distinguishes mammals in the brain. Cerebral cortex is the outer layer (surface) of the cerebrum. It consists of two almost symmetric hemisphere, left cortex hemisphere and right cortex hemisphere. Corpus callosum beneath the cerebral cortex connects hemispheres to enable them communicate with each other. Cerebral cortex is greatly wrinkled and folded so that it can highly increase the surface area to contain more neuron beneath the skull. The ridge in the cortex is called gyrus and the groove is called sulci. Figure 1.1 shows that human cerebral cortex can be divided into four lobes based on gross topographical conventions. They are frontal, parietal, temporal, and occipital in one cortex hemisphere. These lobes play an important role in types of sensory information processing.

The frontal lobe is positioned in the front of the brain and constitutes approximately two thirds of the cerebral cortex. The frontal lobe is associated with motor control and language. In recent advanced studies [13], it is also considered to be relevant to cognitive process such as executive function, attention, personality, self-awareness etc. The parietal lobe is located behind the frontal lobe and central sulcus and above the occipital and temporal lobes. It can integrate different types of sensory perception from most parts of the body. [20] The somatosensory cortex in the parietal lobe is the major receptive area of the sense of touch. [10] Also, impulses from the skin like pain and warmth are sent to the parietal lobe through the thalamus. The temporal lobe is the second largest lobe in the cerebral cortex. [31] It is located beneath the lateral fissure. Due to it close to the ear, the temporal lobe plays an important role in processing auditory and speech signals. Besides, it is also associated to visual perception, language and emotion. The occipital lobe is the smallest lobe. It is

positioned at back of the cortex. The occipital lobe is regarded as the vision center of the brain [46]. It is responsible of processing visual signals and understanding the sense of vision such as color, movement and spatial position.

Since the cerebral cortex plays a key role in various brain functions such as motor control, sensory, language, attention and emotion, disorders of cerebral cortex could cause different types of behavioral and cognitive problems. Alzheimer's disease (AD) is the most common cause of dementia by losing structure or function of neurons in the brain [6]. AD leads to abnormal atrophy of neurons in cerebral cortex regions, comparing to normal reduction of neurons of aging. The atrophy begins in the medial temporal lobe and then spreads into other cerebral cortex areas. [11] There are a variety of symptoms of AD related to the atrophy of cerebral cortex such as memory loss, confusion with time, problems with writing and speaking, poor judgment, etc. [2]

According to Alzheimer's Disease (AD) Fact Sheet from National Institute of Aging [2], AD is estimated to be the third ranked cause of death for elderly people in the United States. The US population suffering from AD became over 5 million in 2016. AD highly reduces the quality of life of the elderly and increases the burden of medical resources. Although researchers across the world made great efforts to study AD in past decades, the cause of AD is still unclear now. [64] There is no significant prevention or medication for AD. If patients could be diagnosed in the early stage of AD, treatments might help delay the process of AD to keep the quality of life. [42] For example, mild cognitive impairment (MCI) which is a pre-stage of AD, leads to cognitive changes that are not enough to influence on daily life. [29] Patients, who are diagnosed as MCI, can receive treatments to delay the process of AD and dementia. Also, changes of brain areas could be used to do research on functions of corresponding brain regions.

Medical imaging can be an appropriate technique to see the change of brain area for AD research. Medical imaging refers to visualize internal structures of human body with radiology technologies like magnetic resonance (MR) imaging, ultrasound, X-ray computed tomography (CT) and elastography. This visualization technique can be used for research, clinical

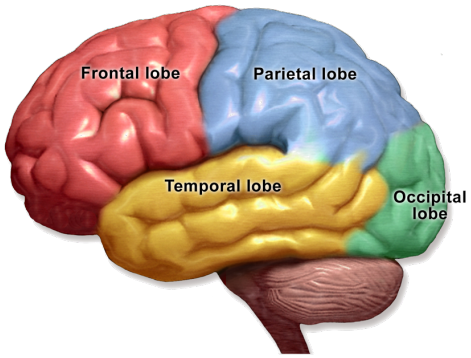

Figure 1.1: Four lobes of the cerebral cortex. [1]

use, medical education and diagnosis. Among various types of imaging modalities, CT and MR are two of them suitable for 3D brain structural imaging. CT uses computer-processed combinations of many X-ray to reconstruct each slice of a 3D image. It is radioactive and good for detecting bones and tumors in the head. While MR is non-radioactive and good for detecting soft tissues such as cortex, nerves and blood vessels.

In recent decades, MR imaging has been widely used for brain imaging in researches because of its high resolution and excellent soft-tissue delineation. MR is able to produce an image of a great range of intensity contrast for grey matter (nerve cell bodies), white matter (fibers) and other soft tissues in human brain. Traditionally, MR images analysis requests extracting information by hand from raw images such as Region of Interests (ROIs) labeling and organ segmentation [22]. Segmentation is used to measure volume, locate boundaries and classify tissues in the image. Radiologists and researchers have to spend much time on these tasks before further experiments. Computer-aided image analysis systems combined with MRI are designed to reduce the burden of manual labeling and has been a hot topic in biomedical image analysis [23].

In this thesis, I will try to implement deep convolutional neural network architectures to label cerebral cortex sub-regions from brain MR images.

Chapter 2
RELATED WORKS

Semantic segmentation in computer vision is a task of giving each pixel a meaningful label, such as assigning each pixel a label of white matter or grey matter in brain MR image segmentation. Compared to most two-dimensional natural images of three color channels, medical images are of high resolution, three-dimension and with only one channel of signals due to the methods of most medical imaging techniques like Magnetic Resonance Imaging and X-Ray Computed Tomography.

In this Chapter, I will briefly introduce four types of biomedical image segmentation approaches in the first section. Then I will describe several tasks and algorithms in brain segmentation. Last but not least, machine learning related segmentation methods will be introduced. I will also go into details for deep learning based methods which are the state-of-the-art architecture.

2.1 Biomedical Image Segmentation Methods

There are different ways to divide biomedical image segmentation methods into sub-groups. I classified recent segmentation approaches into four sub-types [50] [67]: manual segmentation, statistical models, deformable models and multi-atlas segmentation. Deep learning related approaches are not included in this part. I will go into detail of them in the last section of this Chapter.

2.1.1 Manual Segmentation

Before the emergence of robust automatic image segmentation algorithms, medical images are manually delineated by experts. They have to segment out regions of interests (ROIs)

from 2D slices with a user-guided interactive tool like ITK-Snap [21]. It might take an trained expert ten or more minutes to delineate organs or tissue from one 2D slice. More than 3 hours would be spent on a stack of 2D slices to finish the segmentation for one entire 3D medical image. [40] Image segmentation by hand requires a long time for processing and a number of trained experts. Furthermore, segmentation results vary from expert to expert in most cases and the quality of segmentation highly relies on both experience and performance of experts. With the prevalence of medical imaging, it becomes impossible to do manual segmentation for each medical image. These are reasons why semi-automatic and automatic methods are necessary to reduce the workload of medical image experts. Although manual labeling will be replaced with semi-automatic/automatic computer-aid segmentation tools in the future, it is still the gold standard for clinical diagnosis and research use.

2.1.2 Statistical Models

Statistical models learn anatomical variance by extracting prior shape or texture information from training data set (labeled images). This kind of prior information can not only improve segmentation accuracy but also reduce time complexity by constraining the search space. [24] Statistical models can be used to model the shape or the texture of training images. For modeling shape, images are all registered to a template image and then SM extracting the mean and variance of shape from training images. [16] Compared to modeling shape, modeling appearance takes texture into consideration besides image shape. Several extended statistical models may consider more information from the training data like context and local computer vision features.[17]

The key algorithm in statistical models is Expectation Maximization (EM) algorithm [57], which is an optimization methods for objective functions in pattern recognition. Image is set as a Gaussian mixture model. Pixels or key points in the image are consider as a multivariate Gaussian distribution. Intensity of each pixel is the observation value of the model. EM algorithm then uses Maximum Likelihood Estimates (MLE) to compute parameters of the Gaussian mixture model from observations. There are two basic steps in EM, the expectation

step, followed by the maximization step. In the expectation step, the posterior probability (estimation) of latent variables is calculated based on previous model parameters. In the maximization step, new model parameters are estimated with newly-computed probability of latent variables. These two successive steps iterate until convergence. [57]

2.1.3 Deformable Models

Deformable models (DM) in computer graphics are physical-based animation for domain-specific geometry in the image like curves or surfaces. [73] Captured geometry information is capable of representing shape, boundaries, etc and constraining the change of shape over space. And then the optimal algorithm is implemented on to-be-segmented images to get final segmentation. [55] Generally, DM methods convert the segmentation problem to delineate the object from an image like hippocampus from brain MR images.

The "snake" is a classic DM proposed by Kass et al. [39]. This semi-automatic approach starts with a given suitable initial contour. Then an objective energy function is minimized to get final contour of the object. The objective energy function consists of two relative terms, external force and internal force, to make the algorithm deformable over the object boundary. External force considers curve features of the model and local region relationship. Internal force focuses on smoothness and continuity of the boundary. In recent years, more and more DM models emerged for biomedical image segmentation. Based on the classic "snake" approach, researchers made some modifications for objective energy function or increase more prior information for DM methods. Shen et al. [69] implemented a shape deformable model to measure the size and shape of hippocampus. It captures prior retrieved landmarks information from atlases images. Wang et al. [75] combined the statistical model with DM to find boundary of the object as well as the correspondence of a subset of boundary points.

2.1.4 Multi-atlas Segmentation

Instead of developing model-based parametric algorithms for image segmentation, researchers have proposed atlas-based segmentation approaches based on image registration since the begin of this century. Rohlfing et al. [65] implemented a non-rigid registration to segment bee brain from confocal microscopy 3D images. Klein et al. constructed multi-atlases for brain image to automatically assign labels to cortical regions in human brain images. Multi-atlas segmentation (MAS), became popular for biomedical image segmentation because of its flexibility of capturing anatomical variance from training atlases.

Basically, there are four main steps for MAS: atlases construction, registration, atlas selection and label fusion. [37] Registration and label fusion are two key steps. MAS can be considered as an supervised learning method in machine learning. Atlases are labeled images which can be regarded as training data. These images are usually labeled experts by hand with a user-guided interactive visualization tool. Target images (unlabeled) are known as to-be-segmented novel images. They are used as validation data or test data to be predicted for segmentation results. For atlases construction, each atlas is treated equally to generate atlases in most methods. Yet, to build up robust atlases for MAS, we need to select high quality images from all manually-labeled images by using feature selection methods [63]. Also, selected atlases should cover adequate anatomical variance to improve the generalization of MAS. For registration, each atlas image is registered to the target image to compute the transformation map from an atlas to the target image. Since each atlas image is given label (ground truth), the label will be then mapped to the target image based on the transformation map to generate a segmentation of the target image. After doing previous process on each atlas, we will generate N labeled target images (label candidates), where N is the number of images in atlases. Many types of registration algorithm were developed like non-rigid deformable models by ANTS [7] and pixel-to-pixel dense alignment methods by SIFT-Flow [76]. Registration not only makes an important influence of segmentation results, but also becomes a bottleneck of running time because registration should be done N times

for one target image.

For atlas selection, only a subset of atlas will be chosen for label fusion to compute final segmentation. And some selection methods also assign each candidate a weight value to rank all label candidates. One reason for selecting atlas is that some bad candidates may misguide the next step. The other reason is to reduce running time and memory requirements of label fusion since complexity of most label fusion algorithm are more than linear. For label fusion, it aims to generate one label from selected atlas labels. Majority voting ,which is the naive method, assigns each pixel the most frequent label from atlas labels at that position. Other label fusion algorithms take use of image intensity, pair-wise similarity of labels, correlation structure, etc.

2.2 Human Cerebral Cortical Labeling by FreeSurfer

FreeSurfer [4] is a widely-used free software for analyzing structural and functional neuroimaging data. It provides various packages for MR brain image processing and analysis such as image Registration, cortical surface reconstruction, cortical segmentation, etc. For cortical segmentation, the definition of cortical sub-regions is based on Desikan-Killiany (DK) labeling protocol [21]. DK protocol was proposed for labeling human cortex regions into 34 sub-regions per hemisphere. Dr. Rahul S. Desikan manually labeled the sub-regions of cortex from 40 3D brain structural MR scans. He used a delineation method called 'sulcal' of tracing the boundary from the depth of one sulcus to another. This work was done in Department of Anatomy and Neurobiology, Boston University School of Medicine in 2006. Then these labeled scans were used to build atlases to label other new scans.

The DesikanKillianyTourville (DKT) protocol [45] is an extended version of DK labeling protocol. Compared to the original DK protocol, DKT protocol system is constructed from a big data set which is built upon 101 brain MR images and did several modifications to make the protocol easy to use. Three regions are removed from the original DK protocol, i.e. frontal and temporal poles and the banks of the superior temporal sulcus. Two poles region are eliminated because their regions are not continuous along gyri and sulci. The banks of

the superior temporal sulcus is eliminated because of unclear definition of its anterior and posterior. So DKT atlas is made up of 31 regions per hemisphere instead of 34 regions per hemisphere in DK protocol.

For the basic procedure of cortical segmentation, FreeSurfer implements a two-stage approach based on statistical models for labeling cortical regions in a structural MR brain image [27]. The first step is to construct atlas. Instead of registering each image to a selected manually-labeled image, FreeSurfer computes an affine registration for mapping each image to MNI305 atlas [15]. The MNI305 building method is coordinate system for brain linear mapping. It takes use of anatomical landmarks to generate a average MRI volume from all training images and then map each image to this average volume. The MNI305 atlas contains common anatomical information across all brain images to reduce the bias. Then a deformable model [68] is used to remove the skull. After that, three types of probabilities are computed at each point of image (pixel for 2D image, voxel for 3D image). The first prior information the probability distribution of label classes at each point. The second one contains the neighbor information which is the probability of the label class of one point given the label of its neighbors. The third one is the probability distribution function of value of each point. The second step of FreeSurfer's approach is for parcellation of cerebral cortex sub-regions. A first order anisotropic non-stationary Markov random field (MRF) is used to model parcellation units based on three types of prior information in training images. [27]

2.3 Deep Learning Methods for Semantic Image Segmentation

Machine learning is a key technique in Artificial Intelligence (AI). It has been widely-used in various research fields and industrial areas such as natural language processing, computer vision, computational biology, optical character recognition and self-driving systems. [51] The main concept of machine learning is to learn a robust statistical model from a large amount of data with some powerful algorithms and make predictions on other similar data. Machine learning tasks can be generally classified into two categories based on algorithms, i.e.

supervised learning and unsupervised learning. In supervised learning, input training data is given desired output (a.k.a. ground truth) to train a model to learn the transform map from input to output. Classification and regression are two examples of supervised learning tasks. The output of classification is discrete, while that of regression is continuous. In unsupervised learning, there is no desired output given for input training data. Unsupervised algorithms are asked to find hidden structures or features from unlabelled data by itself. Cluster, which is a classic task of unsupervised learning, aims to divide unlabelled input data into different groups.

Semantic image segmentation is a popular sub-domain in computer vision, especially for biomedical image segmentation. It involves assigning a class label for each pixel for a two-dimensional image or voxel for a three-dimensional image. In some traditional supervised machine learning methods for semantic segmentation, pixels in the image are treated as instances for classification. They are represented and described by computer vision features like Scale-invariant feature transform (SIFT) [53], Histograms of Oriented Gradient (HOG) [19], Local Binary Patterns (LBP) [59], etc. Features are then input into a classification model like Support Vector Machine (SVM) and Logistic Regression. The accuracy of segmentation highly depends on the quality of features and feature selection methods. In unsupervised learning methods like graphical models, each pixel or a group of pixels is usually encoded as a node with several features and all of them are connected in some ways to construct a graphical model such as Hidden Markov Models (HMM) [78] and Conditional Random Fields (CRF) [48]. Class label is set as a latent state for each node. Then an energy function is defined for optimization. For training, statistical inference methods like Loopy Belief Propagation [58] are used to optimize the energy function. Thousands or more of iteration is computed until converge during inference to assign a label for each pixel.

Hinton et al. [35] proposed a powerful fast and greedy algorithm called Back Propagation for neural networks training in 2006. Back Propagation makes it possible to train deep neural networks in acceptable complexity of computing. In the recent decade, deep neural networks have been rapidly used in various application fields of machine learning. There are two

robust sub-types of deep neural networks designed for different tasks, i.e. Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). RNN is a dynamic network to train a model for sequential data like voice signal, video and gene sequence. CNN involves convolutional kernels to extract local feature from input data and it is popular in computer vision, signal processing, etc. RNN and CNN surpassed several traditional classic methods and made big progress in a huge number of domains. [51]

In computer vision, CNNs have been proven to be one of the best techniques for image segmentation, classification, etc. Fully convolutional networks (FCN) [52] is a supervised learning architecture designed for image semantic segmentation. Instead of assigning each pixel of the image a class to get a pixel-wise dense classification, FCN takes the whole image at a time as input to do end-to-end segmentation of natural images. For example, FCN would output an segmentation image map of $256 \times 256 \times N$ (N is the number of classes) if the output image size is 256×256. It usually consists of two main parts, encoding and decoding. The encoder, followed by the decoder, is a pretrained standard classification network like VGG-16 [70]. The decoder can be one or more deconvolutional layers or up-sampling layers to semantically convert features in the bottleneck layer onto the pixel space.

Nowadays, there are lots of papers published based on FCN. Ronneberger et al.[66] proposed a novel framework of convolutional neural networks, which is called U-Net, for biomedical image segmentation. The cropping step in U-Net concatenates feature maps of the encoding part to feature maps of decoding part. It can increase the resolution of segmentation results by adding more information of original images to the decoding part. Vijay et al.[8] added to the decoder with the pooling indices computed in the max-pooling step of the corresponding encoder. These added indices, then, combined with the sparse up-sample maps to be convolved with convolutional kernels to generate dense feature maps. Jegou et al.[38] extended DenseNets to deal with semantic segmentation. Dense blocks perform iterative concatenation of feature maps for fully convolutional network. This architecture performs well on urban scene benchmark data sets such as CamVidand Gatech. Chaurasia et al.[12] proposed a novel neural network architecture to make efficient use of computing resource for

semantic segmentation. It is well-performed on data sets with large-scale computation.

Among these extended architectures of FCN, U-Net has been proven to be one of the best frameworks for biomedical image segmentation. [66] It won the Grand Challenge for Computer-Automated Detection of Caries in Bitewing Radiography at ISBI (the IEEE International Symposium on Biomedical Imaging) 2015 and the Cell Tracking Challenge at ISBI 2015 on the two most challenging transmitted light microscopy categories by a large margin.

In this paper, I will implement 2D/3D U-Net for semantic image segmentation. Figure 3.1 shows an example of 2D U-Net. As a extended framework of FCN, U-Net takes in a whole image as input and generate a probability-based segmentation map. The depth of U-Net is how many up-sampling/down-sampling layers in the U-Net. In this example, the depth is four. A block is defined as several layers starting from a up-sampling/down-sampling layer to next up-sampling/down-sampling layer. A block consists of several successive convolutional layers followed by a up-sampling/down-sampling layer. The number of convolutional layers in one block is usually set as 2.

Basically, U-Net can be split into two parts, encoding (the left part) and decoding (the right part). In the left part, there are four neural network blocks and each of them contains two convolutional layers and a pooling layer. Two repeated convolutional layers use 3×3 kernel, zero padding and a rectified linear unit (ReLU) activation function. The pooling layer is a 2×2 max pooling layer which calculates the max value of a local 2×2 region for down-sampling. The right part which is symmetric to the left part consists four neural network blocks as well. Each block in the decoding part, except the final block, starts with a up-pooling layer and followed by two repeated convolutional layers. The up-pooling layer is a up-sampling layer to double previous feature map channels. Two convolutional layers, sometimes known as "up-convolution", in the right part are the same as the convolutional layers in the left part. The last block uses an extra $softmax$ layer to get the categorical distribution of all possible classes. The copy part between two sides is to copy feature maps from left side to right so that it can compensate the loss of information of the encoding part

in order to output segmentation of high resolution.

Besides the general description of U-Net in this Chapter, I will describe specific U-Net architectures used in this thesis in Chapter 3 .

Chapter 3

METHODS

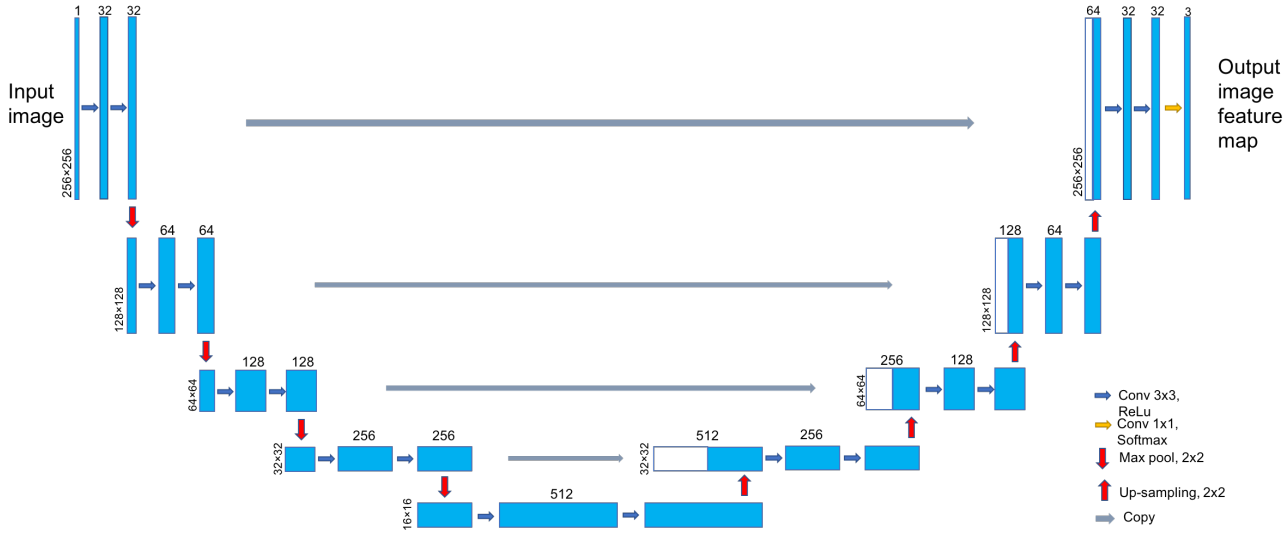

Figure 3.1: 2D U-Net Model Architecture Example

3.1 Basic Layers of U-Net

Figure 3.1 shows an example of 2D U-Net architecture. The input of the deep neural network is a batch of two dimensional images and the output is a segmentation map of the input. The dimension of segmentation map is three because the last layer is a $softmax$ layer to give each a pixel a categorical distribution of all possible classes. To get the final segmentation results, I can calculate max values on the $softmax$ axis. There are various layers in U-Net, such as convolutional layers, max pooling layers, up-sampling layers and concatenation(copy) layers [3].

Convolutional layer is the key feature in CNNs. Compared with fully-connect layer, it uses Parameter Sharing scheme to dramatically reduce the number of parameters in a deep neural network. For example, I assume that dimension of the input feature map is $50 \times 50 \times 3$ and dimension of the output feature map is $50 \times 50 \times 10$. A fully-connected layer uses $(50 \times 50 \times 3) \times (50 \times 50 \times 10) = 187.5M$ weights except weights of bias. Otherwise, a convolutional layer with a 3×3 kernel is able to take only $3 \times 3 \times 3 \times 10 = 270$ weights (excluding bias) to deal with the problem. The reason why the fully-connected layer uses so many parameters is that each neuron (pixel/voxel in computer vision) in the output feature map connects each neuron in the input feature map. The fully-connected layers always assigns an independent weight vectors for each neuron. While the convolutional layer makes an assumption that one local computing method for feature computing can be used on all pixels/voxels in the same feature map. Each neuron of the same channel in the output feature map shares the same weights in one kernel.

There are 4 main hyper-parameters of a convolutional layer, i.e. kernel size, depth, stride and padding. Kernel size is the size of convolutional window which determines how big the local region is considered to compute features. For two dimensional convolution, kernel size of 3×3, 5×5 or 7×7 is widely-used in most cases. Depth is also called as the number of convolutional kernels in the convolutional layer. It corresponds to the number of channels of the output feature map. Stride specifies the stride length of the convolution. A stride of 2 means that the convolutional kernel jumps 1 pixel/voxel in the specific axis when computing features. And it makes the output feature map half size of the input feature map. In most cases, a stride of 1 is used for a convolutional layer, which helps keep the same feature map size for input and output. Padding is designed for handling border pixels/voxels of the input map. In the original U-Net, authors uses no-padding for each convolutional layers. It makes the size of output feature map a little smaller $(N - k - 1)$ than that of the input, where N is the input size of one axis of a channel and k is the kernel size.

Max pooling layer is a kind of pooling layer. The pooling layer aims to down-sample the feature map so that it can help the CNN architecture reduce numbers of parameters in

following layers. It is usually inserted after several successive convolutional layers. These convolutional layers and one following pooling layer is usually called as a block in CNNs. There are different kinds of pooling operations such as average pooling and max pooling. Max pooling with a window size of 2×2 is popular in two dimensional CNNs. It keeps one largest value in the local 2×2 region and filters out rest 3 small ones. For example, we assume that the dimension of input feature map of a 2×2 max pooling layer is $256 \times 256 \times 10$. The dimension of output feature map would be $128 \times 28 \times 10$. The max pooling layer makes the feature map half size small in each channel than that of the input and it also keeps the depth (number of channels) of the feature map.

The up-sampling layer is a unique part in U-Net and other fully convolutional neural networks. The function of up-sampling layer is to map the feature into a high dimension. In the right side of U-Net, it is designed to reconstruct the dimension of the input image step by step to get the same scale segmentation results. Technically, the function of up-sampling layer is similar to rescaling/upsampling an image in image processing. It is usually made up of re-sampling and interpolation. And the concept of window size of up-sampling is the same as that of max-pooling. For example, we assume the dimension of input feature map of a 2×2 up-sampling layer is $128 \times 128 \times 10$. The dimension output feature map would be $256 \times 56 \times 10$. The max pooling layer makes the feature map big in each channel and it keeps the depth (number of channels) as well.

The concatenation layer is a type of merging layers to concatenate two or more input feature maps of the same dimension. In the U-Net, I use the concatenation layer to merge layers from the left side into layers of the right side. The dimension of right feature map is the same as the left after same numbers of pooling layers and up-sampling layers with a 2×2 local region. This operation aims to add more source information to the right side in order to get high resolution segmentation in the last layer. For example, we assume that we need concatenate two feature maps, the $128 \times 28 \times 10$ feature map from left side and the $128 \times 28 \times 20$ feature map from the right side. I would get a concatenated feature map of $128 \times 28 \times 30$.

3.2 2D U-Net

Table 3.1: 2D U-Net Parameters

Convolutional Layer	Parameters
1^{st} layer in Encoding Block 1	$(3 \times 3 \times 1 + 1) \times 32 = 320$
2^{nd} layer in Encoding Block 1	$(3 \times 3 \times 32 + 1) \times 32 = 9248$
1^{st} layer in Encoding Block 2	$(3 \times 3 \times 32 + 1) \times 64 \approx 18K$
2^{nd} layer in Encoding Block 2	$(3 \times 3 \times 64 + 1) \times 64 \approx 37K$
1^{st} layer in Encoding Block 3	$(3 \times 3 \times 64 + 1) \times 128 \approx 74K$
2^{nd} layer in Encoding Block 3	$(3 \times 3 \times 128 + 1) \times 128 \approx 148K$
1^{st} layer in Encoding Block 4	$(3 \times 3 \times 128 + 1) \times 256 \approx 295K$
2^{nd} layer in Encoding Block 4	$(3 \times 3 \times 256 + 1) \times 256 \approx 590K$
1^{st} layer in Encoding-Decoding Block	$(3 \times 3 \times 256 + 1) \times 512 \approx 1.2M$
2^{nd} layer in Encoding-Decoding Block	$(3 \times 3 \times 512 + 1) \times 512 \approx 2.4M$
1^{st} layer in Decoding Block 1	$(3 \times 3 \times 512 + 1) \times 256 \approx 1.2M$
2^{nd} layer in Decoding Block 1	$(3 \times 3 \times 256 + 1) \times 256 \approx 590K$
1^{st} layer in Decoding Block 2	$(3 \times 3 \times 256 + 1) \times 128 \approx 295K$
2^{nd} layer in Decoding Block 2	$(3 \times 3 \times 128 + 1) \times 128 \approx 148K$
1^{st} layer in Decoding Block 3	$(3 \times 3 \times 128 + 1) \times 64 \approx 74K$
2^{nd} layer in Decoding Block 3	$(3 \times 3 \times 64 + 1) \times 64 \approx 37K$
1^{st} layer in Encoding Block 4	$(3 \times 3 \times 64 + 1) \times 32 \approx 18K$
2^{nd} layer in Encoding Block 4	$(3 \times 3 \times 32 + 1) \times 32 = 9248$
SoftMax layer in Encoding Block 4	$(1 \times 1 \times 32 + 1) \times 3 = 99$
Total Parameters	**6.9M**

I modify the auto-encoder structure segmentation network from U-Net (Figure 3.1) for brain lobes segmentation. The whole tasks are split into 2 stages. In stage one, I train

a U-Net to get a mask for filtering out brain tissues which I am not interested in such as white matter and some grey matter. In stage two, a similar U-Net architecture is trained to assign each pixel/voxel one specific brain lobe label. In stage 1, the size of input is modified to 256×256 and output size is modified to $256 \times 256 \times 3$ to fit our data set. There are totally 18 convolutional layers in the U-Net including the $softmax$ layer. For the decoding part in the U-Net, I use up-sampling layers instead of transpose convolutional layers. It can not only decrease the number of parameters, but also make the output segmentation much smoother. And for the convolutional kernel, I use *samepadding* to keep the same the size of feature map of each block. It can also help decrease the loss of border information in each convolutional step. For stage 2, the only difference from that of stage 1 is the number of channels of last $softmax$ layer, which is used to match the number of classes. I modify the number of channels as 11 in the softmax layer.

Compared to fully connected neural networks, one of the key advance of CNN is using convolutional kernels to share weights and reduce number of parameters. CNN network with small number of parameters can be trained easily and request few computing resources like memory. Table 3.1 shows the number of parameters of each convolutional layer in Figure 3.1. It can be computed by the following equation.

$$P_n = (d \times d \times c_{in} + b) \times c_{out} \tag{3.1}$$

where P_n is the number of parameters of the n^{th} convolutional layer, d is the size of convolutional kernel, c_{in} is the number of channels of input feature map, c_{out} is the number of channels of output feature map and b is the number of bias of each convolutional layer.

Take the second layer in Encoding Block 1 for example. The dimension of input feature map is $128 \times 128 \times 32$ where the number of input channels c_{in} is 32. The dimension of output feature map is $128 \times 128 \times 64$ where the number of output channels c_{out} is 64. The size of kernel d is 3×3. For bias, there are two options, no bias or one bias value. I used $b = 1$ here. Base on Equation 3.1, the number of parameters of the second layer in Encoding Block 2 can be calculated as $(3 \times 3 \times 32 + 1) \times 64 \approx 18K$.

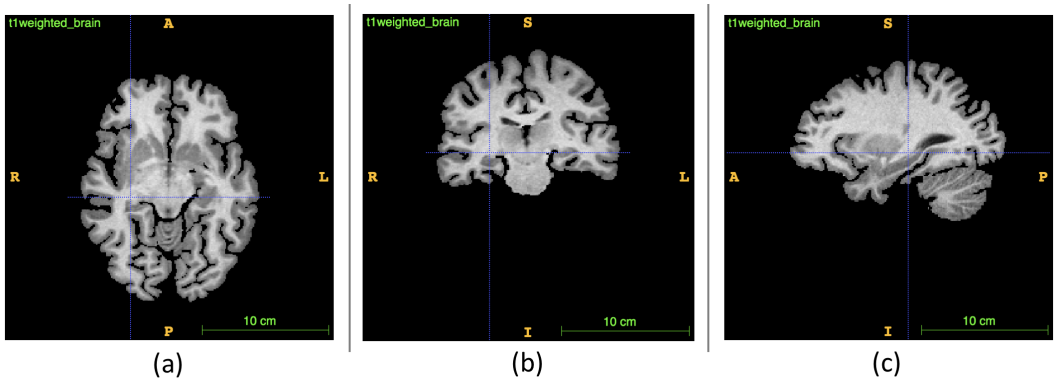

Figure 3.2: The Original 3D MR Image of OASIS-TRT-20-1[45]. (a) Horizontal plane. (b) Coronal plane. (c) Sagittal plane.

3.3 3D U-Net

I try to use 3D U-Net to take into consideration 3D spatial information and context for cortex segmentation. I also use the same two-stage U-Nets to segment brain lobes from MR images in 3D U-Net. Original 3D images are divided into small patches like $64 \times 64 \times 64$ patches for model input. Experiment procedure will be discussed in detail in the following implementation part. 3D U-Net architecture in Figure 3.3 is similar to the 2D one. It expands the 2D U-Net architecture into a 3D version.

Figure 3.3 illustrates the architecture of 3D U-Net and Table 3.2 shows number of parameters of 3D U-Net layer by layer. Instead of designing four blocks for both encoding and decoding part, I reduce the number of blocks of both size to three because it can decrease memory requirement to train a 3D U-Net. In implementation, 3D patches of $64 \times 64 \times 64$ with batch size of 16 requires over 90% memory on a NVIDIA GPU Tesla K80 (memory: 11G). If the deep neural network became deeper, I would have to reduce the batch size to keep the use of memory lower than 11G. While a batch size less than 8 would be too small to run a batch gradient descend during training. For layers in 3D U-Net, they are expanded from 2D to 3D such as $3 \times 3 \times 3$ convolutional kernels and $2 \times 2 \times 2$ max-pooling kernels.

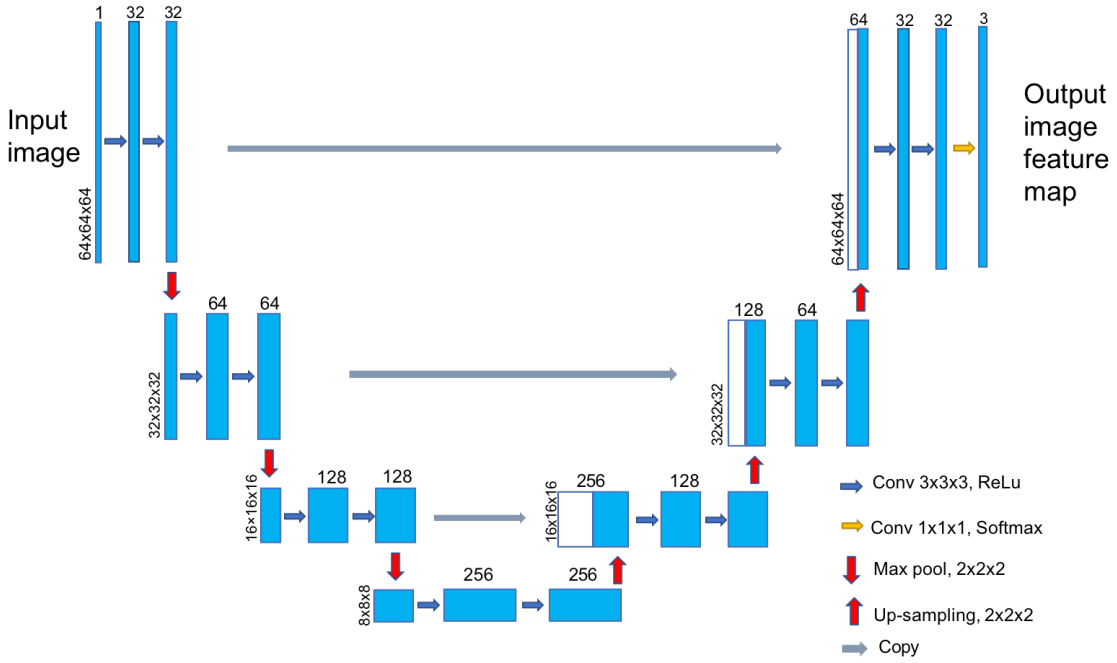

Figure 3.3: 3D U-Net Model Architecture Example

3.4 Loss Function and Evaluation Metric

In the implementation, the loss function is formed by weighted Dice Coefficient.

$$Loss = 1 - \sum_{i}^{l} W_i \frac{2|A_i \cap B_i|}{|A_i| + |B_i|}, \tag{3.2}$$

which shows that l is the number of labels for the image. W_i is the weight of the i^{th} label. It is created to deal with imbalance numbers of pixel labels. A_i is a 2D matrix of the i^{th} label prediction. B_i is a 2D matrix of the i^{th} label ground truth.

For evaluation, Dice Coefficient is used to evaluate the overlap between ground truth and semantic segmentation from the deep neural network.

$$Dice\ Coefficient = \frac{2|A_i \cap B_i|}{|A_i| + |B_i|}, \tag{3.3}$$

where A_i is a 2D matrix of the i^{th} label prediction. B_i is a 2D matrix of the i^{th} label ground truth.

Table 3.2: 3D U-Net Parameters

Convolutional Layer	Parameters
1^{st} layer in Encoding Block 1	$(3 \times 3 \times 3 \times 1 + 1) \times 32 = 896$
2^{nd} layer in Encoding Block 1	$(3 \times 3 \times 3 \times 32 + 1) \times 32 \approx 18K$
1^{st} layer in Encoding Block 2	$(3 \times 3 \times 3 \times 32 + 1) \times 64 \approx 55K$
2^{nd} layer in Encoding Block 2	$(3 \times 3 \times 3 \times 64 + 1) \times 64 \approx 111K$
1^{st} layer in Encoding Block 3	$(3 \times 3 \times 3 \times 64 + 1) \times 128 \approx 221K$
2^{nd} layer in Encoding Block 3	$(3 \times 3 \times 3 \times 128 + 1) \times 128 \approx 442K$
1^{st} layer in Encoding-Decoding Block	$(3 \times 3 \times 3 \times 128 + 1) \times 256 \approx 885K$
2^{nd} layer in Encoding-Decoding Block	$(3 \times 3 \times 3 \times 256 + 1) \times 256 \approx 1.8M$
1^{st} layer in Decoding Block 1	$(3 \times 3 \times 3 \times 256 + 1) \times 128 \approx 885K$
2^{nd} layer in Decoding Block 1	$(3 \times 3 \times 3 \times 128 + 1) \times 128 \approx 442K$
1^{st} layer in Decoding Block 2	$(3 \times 3 \times 3 \times 128 + 1) \times 64 \approx 221K$
2^{nd} layer in Decoding Block 2	$(3 \times 3 \times 3 \times 64 + 1) \times 64 \approx 111K$
1^{st} layer in Encoding Block 3	$(3 \times 3 \times 3 \times 64 + 1) \times 32 \approx 55K$
2^{nd} layer in Encoding Block 3	$(3 \times 3 \times 3 \times 32 + 1) \times 32 \approx 28K$
SoftMax layer in Encoding Block 3	$(1 \times 1 \times 1 \times 32 + 1) \times 3 = 99$
Total Parameters	**5.2M**

Chapter 4

U-NET IMPLEMENTATION

4.1 Data

Mindboggle-101 [45] is a free manually-labeled data set of MR images from 101 healthy subjects. About 20 papers related to Mindboggle-101 were published in recent years for brain visualization [41], brain shape analysis [9], etc. It aims to provide a large number of ground truth images of morphometric variation for clinical comparison and development of novel automatic registration/segmentation approaches. The DKT protocol is used to label cortical regions in Mindboggle-101 data set. For the experiment, training a model and tuning model parameters on entire 101 3D images require using GPUs for a long time. Due to limited computing resources, I randomly selected a subset called OASIS-TRT-20 from Mindboggle-101. It contains 20 T1-weighted brain MR images from healthy subjects. The age of 20 subjects ranges from 19 to 34 with mean of 23.4 and standard deviation of 3.9. MR images of young subjects might be limited for Alzheimers disease. Eight of them are men and the rest twelve are women. These three dimensional images are pre-processed and manually labeled according to the Desikan-Killiany-Tourville (DKT) protocol [21].

Figure 3.2 shows the pre-processed image of the first 3D MR image in OASIS-TRT-20 from three planes. Image intensity is normalized and skull is removed by FreeSurfer. There are 256, 256 and 160 slices for horizontal plane, coronal plane and sagittal plane, respectively. Table 4.1 shows labels of DKT protocol. There are totally 35 cortical labels per hemisphere for the original DKT protocol. Label number starts with 1 is for the left brain and 2 for the right brain. For example, label number "1006" is the left entorhinal and "2006" is the right entorhinal. Four labels have been already removed from the OASIS-TRT-20 data set, bankstss (1001, 2001), corpus callosum (1004, 2004), frontal pole (1032, 2032), and temporal

pole (1033, 2033) because they spanned the superior temporal sulcus fundus and the anterior boundary was ambiguous. In this paper, I do not label all 31 parts per hemisphere because it is hard problem to deal with by a volumn-based segmentation method like U-Net. Instead, I would like to figure out several main parts of grey matter. In table 4.1, thirty labels are grouped into 6 main parts and one label (insula) is removed according to the DKT protocol paper [21].

Table 4.1: Label Names of DKT protocol

Main Part of grey Matter	Label Name	Label Number
Temporal Lobe (medial aspect)	Entorhinal	1006, 2006
	Parahippocampal	1016, 2016
	Fusiform	1007, 2007
Temporal Lobe (lateral aspect)	Superior Temporal	1030, 2030
	Middle Temporal	1015, 2015
	Inferior Temporal	1009, 2009
	Transverse Temporal	1034, 2034
Frontal Lobe	Superior Frontal	1028, 2028
	Lateral Orbitofrontal	1012, 2012
	Medial Orbitofrontal	1014, 2014
	Precentral	1024, 2024
	Paracentral	1017, 2017
	Caudal Middle Frontal	1003, 2003
	Pars Opercularis	1018, 2018
	Pars Orbitalis	1019, 2019
	Pars Triangularis	1020, 2020
	Rostral Middle Frontal	1027, 2027
Parietal Lobe	Postcentral	1022, 2022
	Supramarginal	1031, 2031

	Superior Parietal	1029, 2029
	Inferior Parietal	1008, 2008
	Precuneus	1025, 2025
Occipital Lobe	Lingual	1013, 2013
	Pericalcarine	1021, 2021
	Cuneus	1005, 2005
	Lateral Occipital	1011, 2011
Cingulate Cortex	Isthmus Cingulate	1010, 2010
	Posterior Cingulate	1023, 2023
	Rostral Anterior Cingulate	1026, 2026
	Caudal Anterior Cingulate	1002, 2002
REMOVED	Bankstss	1001, 2001
	Corpus Callosum	1004, 2004
	Frontal Pole	1032, 2032
	Temporal Pole	1033, 2033
	Insula	1035, 2035

4.2 Two-stage 2D/3D U-Net

For both 2D U-Net and 3D U-Net, I implemented two U-Nets to handle the entire problem of labeling sub-regions of cerebral cortex. I call it a two-stage system. In stage one, it aims to remove white matter and other grey matter parts which are not included in 5 grey matter main parts. Stage one U-Net is a 3-class pixel/voxel classifier to generate a mask to segment 6 grey matter main parts from background and other tissues. In stage two, I designed a similar U-Net to set different labels for 6 grey matter main parts per hemisphere. The only difference of U-Net architecture in two stages is the depth of the last 1×1 convolutional layer which determines the number of classes for model output. In stage one, there are 3

classes. And there are 11 classes in stage two.

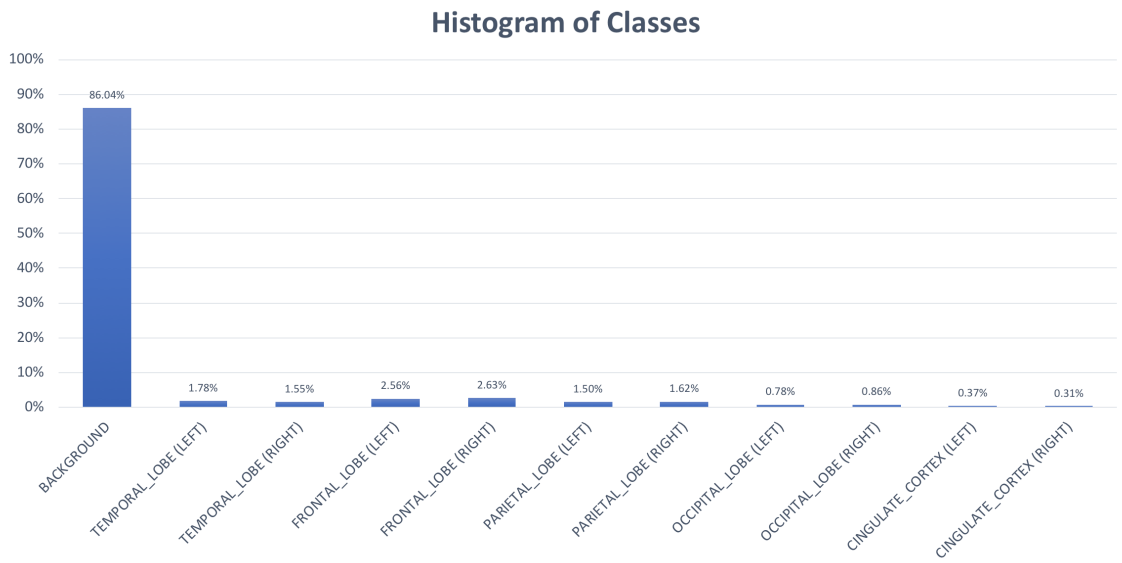

Figure 4.1: Histogram of labels in the training set

4.3 Prepare Data

For 2D U-Net, the first 18 subjects are used for training, the 19^{th} subject for validation and the last one for test. All 3D MR images were sampled on the horizontal plane to generate 2D slices. I did not include those 2D slices that contain only the background without any grey matter and white matter. The original size of 2D slice is 160×256. To make same the width and length of input images, I re-scaled 2D images to 256×256 with nearest interpolation. I also do normalization for input images. I calculated mean (1222) and standard deviation (323) of pixel values of grey matter and white matter in the training set. All pixel values of grey matter were subtracted by mean and then divided by standard deviation. And the pixel values of the background were set to 0.

For 3D U-Net, similar to that in 2D U-Net, first 18 subjects are used to generate training set, the 19^{th} subject for validation and the last one for test. The engineer-level challenge

is that an entire $256 \times 256 \times 160$ 3D image is too big for U-Net input because it will cost too much memory and time to train a CNN model. In this case, I cut the 3D images into several $64 \times 64 \times 64$ patches in both Stage 1 and Stage 2. I designed a pre-processing pipeline for generate patches for each training, validation and testing. First, I removed the border of 3D images on each plane. Second, these images were re-scaled to $256 \times 256 \times 256$ with interpolation. Third, each $256 \times 256 \times 256$ patch was cut into 64 $64 \times 64 \times 64$ patches as input for 3D U-Net. Consequently, there are 1152 3D patches ($64 \times 64 \times 64$) for training set, 64 3D patches for validation set and 64 patches for testing set.

	2D U-Net Stage 1	2D U-Net Stage 2	3D U-Net Stage 1	3D U-Net Stage 2
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	1e-4	1e-5	1e-4	1e-5
Mini Batch Size	32	32	16	16
Time per Batch	0.6 s	0.6 s	8 s	8 s
Time per Epoch	0.5 min	0.5 min	9.6 min	9.6 min
Epoch	30	30	10	20

Table 4.2: Training Parameters of 2D/3D U-Net

4.4 Training

For stage 1 of 2D U-Net and 3D U-Net, I used weighted Dice coefficient as loss function with the weight of 0.1, 0.6 and 0.3 for background, grey matter main parts per hemisphere and other brain tissue, respectively. The weighted loss function is used to compensate imbalanced number of samples from different classes. For 2D U-Net, since I removed 2D slices which contain only the background without any brain tissue, I finally got 1701 2D slices for training, 95 for validation and 137 for test. For 3D U-Net, I got 1152 3D patches for train, 64 for validation and 64 for test.

For stage 2 of 2D U-Net and 3D U-Net, I first generated the histogram of label. Figure 4.1 shows the label statistics of 1 background label and 10 types of pixel labels over the training set. The number of label 0 pixel is almost hundreds of times more than the number of other 10 labels. After weighted, Label 0 will be set a small weight, while the rest labels will be set a big weight. If Dice Coefficient were not weighted, there would be little 10 types of pixel labels regions in the segmentation results. The number of training, validation and test sample are the same as that in stage 1.

Table 4.2 lists parameters for training 2D/3D U-Net in each stage. These U-Nets were trained on one NVIDIA K80 GPU with 11 G memory. 2D/3D U-Net uses the Adam optimizer with a different learning rate for stage 1 ($1e-4$) and stage 2 ($1e-5$). The mini batch size of 2D U-Net is 32. While that of 3D U-Net is 16 because 3D U-Net requires much more memory during training than 2D U-net. It took 0.5 minute and 9.6 minutes to train 2D U-Net and 3D U-Net, respectively. The number of epoch is 30 for 2D U-Net of stage 1 and stage 2. And 3D U-Net of stage 1 was trained for 10 epochs and 3D U-Net of stage 2 was trained for 20 epochs. Figure 4.2 shows the weighted Dice coefficient loss of training set and validation set. I saved the model with the lowest loss in each U-Net. Except 3D U-Net of stage 2, other three U-Nets get convergence during training. The loss of 3D U-Net shows it starts overfitting after 16 epochs.

4.5 Results

Table 4.3 shows the Dice coefficient of U-Nets. Both 2D U-Net and 3D U-Net works well for labeling grey matter and white matter in stage 1. They both achieve approximate 95% Dice coefficient. Figure 4.3 and Figure 4.4 illustrate the comparison of prediction and ground truth. The prediction is similar to the ground truth. For stage 2, 2D U-Net and 3D U-Net only achieve approximate 55% Dice coefficient for labeling sub-regions of cerebral cortex. Figure 4.5 shows the Dice coefficient of 10 brain lobes in stage 2 for both 2D U-Net and 3D U-Net.

	2D U-Net Stage 1	2D U-Net Stage 2	3D U-Net Stage 1	3D U-Net Stage 2
Dice coefficient	0.9460	0.550 ± 0.046	0.9633	0.567 ± 0.079

Table 4.3: Dice coefficient of results on the test set

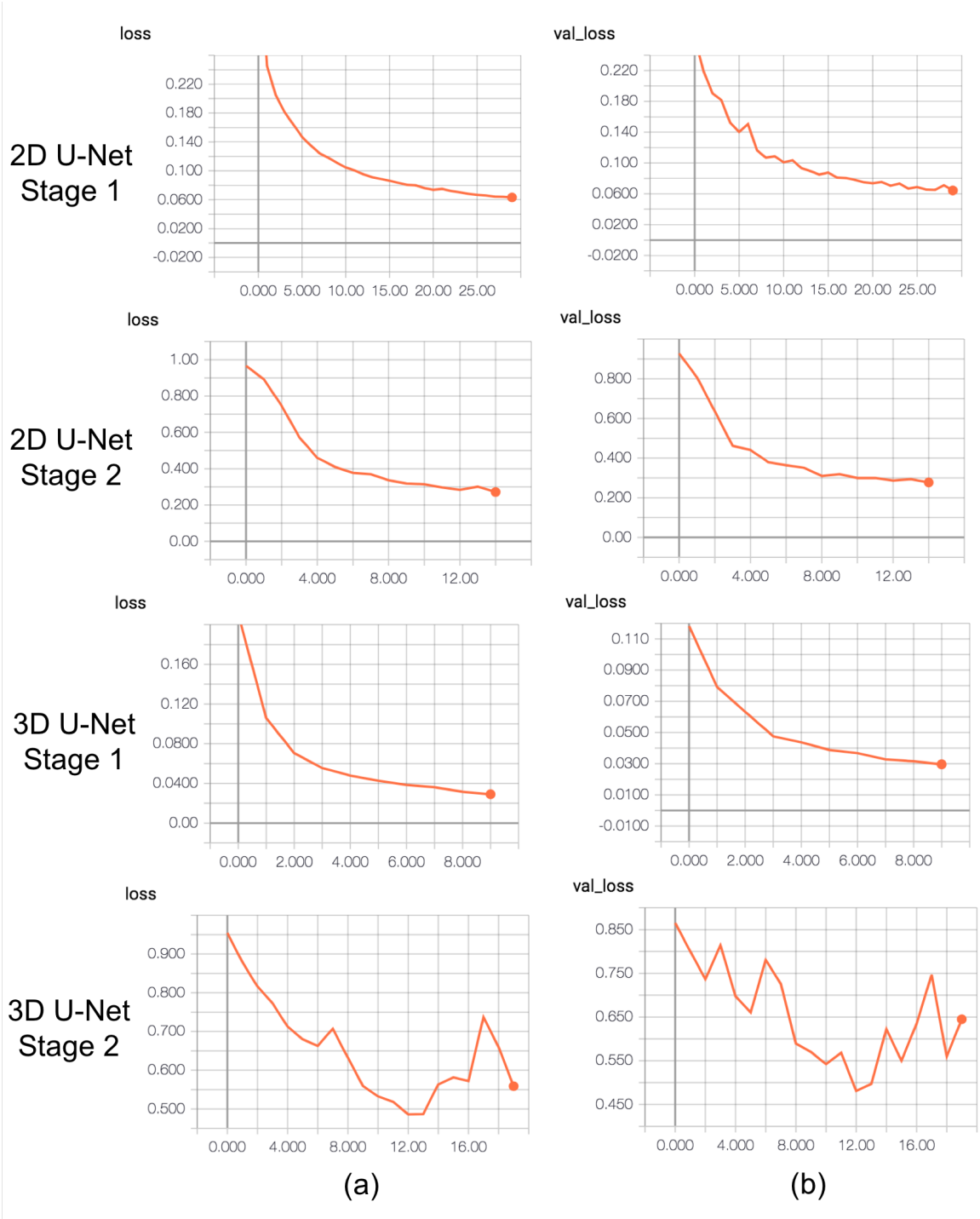

Figure 4.2: Weighted Dice Coefficient Loss Plot (X-axis: number of epochs, Y-axis: value of loss). (a) Training loss. (b) Validation loss

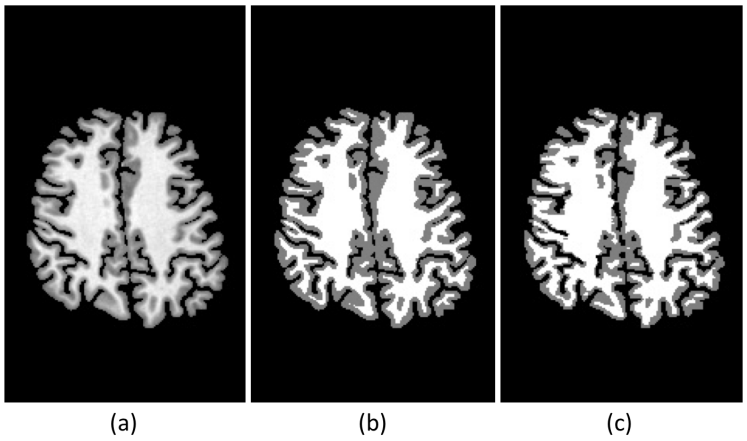

Figure 4.3: Results of 2D U-Net in stage 1 of horizontal plane on the test set. (a) Original image (b) Ground truth image (c) Prediction of 2D U-Net

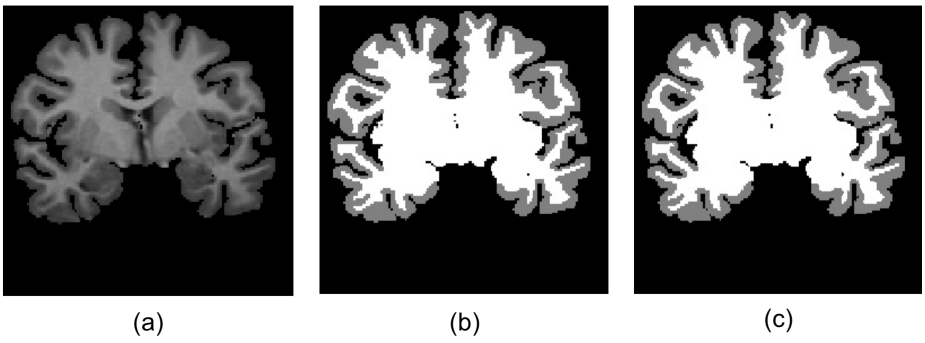

Figure 4.4: Results of 3D U-Net in stage 1 of coronal plane on the test set. (a) Original image (b) Ground truth image (c) Prediction of 3D U-Net

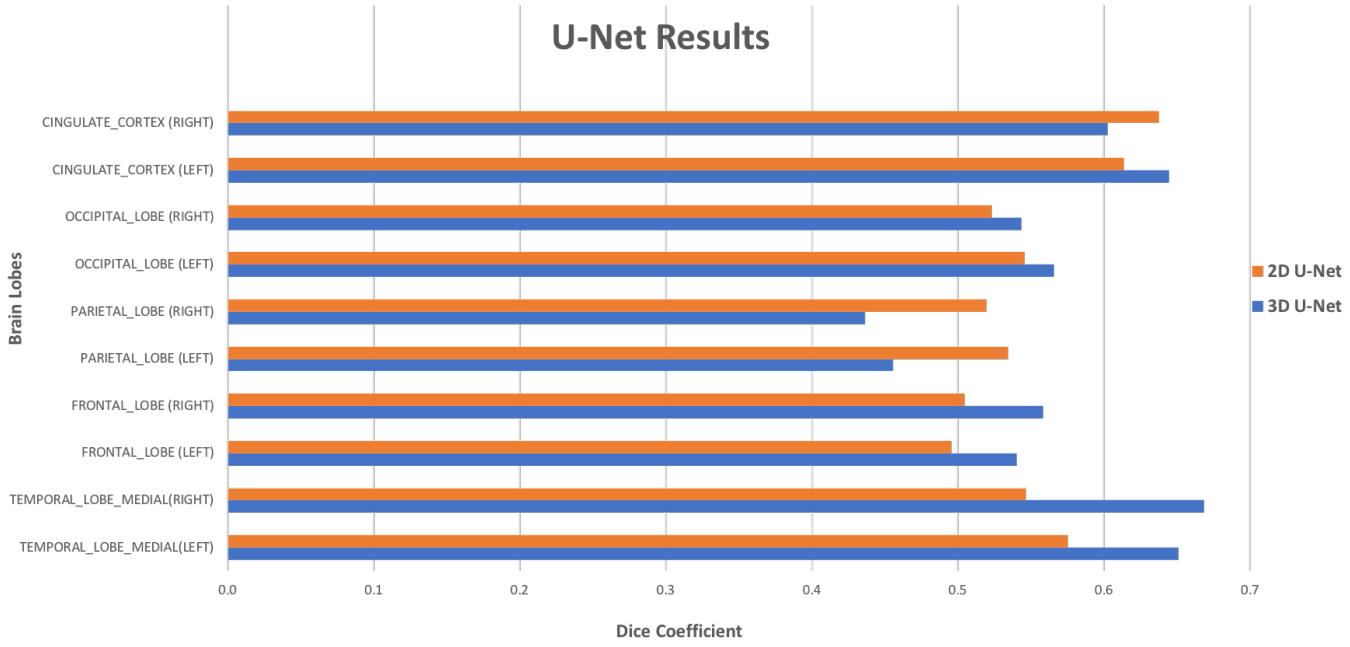

Figure 4.5: Dice overlap of each class of brain lobes by 2D/3D U-Net on test set

Chapter 5

DISCUSSION

In this chapter, I will discuss some challenges around MindBoggle data set, establishing ground truth, and in selecting parameters for the U-Net algorithm. I will first talk about data set used in the thesis and the corresponding manual labeling protocol. Then, I will mention details of U-Net architecture like parameters tuning and the comparison of 2D U-Net and 3D U-Net. In the last section, I will compare U-Net to the widely-used cerebral cortex segmentation package, FreeSurfer.

5.1 DKT Protocol Labeling & MindBoggle Data Set

DKT protocol was developed based on original FreeSurfer's DK labeling protocol. FreeSurfer's DK classifier was trained on 40 brains manually labelled by one expert.[21] It is difficult to make sure the quality of ground truth (DK labels) if no extra experts do the inspection. It is mentioned in the DKT protocol labeling paper [45] that 40 brains were pre-labeled by FreeSurfer's DK classifier and then further manually edited by two experts. One expert edited brains first and the other one inspected all edits. This manual labeling procedure is able to reduce errors made by the first experts. I think this is a big advantage of DKT protocol.

The MindBoggle data set provides 101 labeled MR brain images for cerebral cortex segmentation. Although it contains a large number of brain images with ground truth, it is not suitable to train a model for processing images from patients with cerebral cortex disorders like AD. One reason is that all 101 images in the MindBoggle data set are acquired from healthy individuals. Since cerebral cortex changes for those who suffer from cerebral cortex disorders [26] like cortex thickness reduction, these images from healthy individuals

may not be able to contain this kind of anatomical variance. The other reason is that the age distribution of these healthy individuals is centered around 20 to 40. And the max age is 61. There are normal age-related morphometric changes for human cerebral cortex. [36] Most patients with AD are over 75. [34] Therefore, the MindBoggle data set may not be appropriate for the older age group.

5.2 U-Net Parameters Tuning

As a type of deep neural networks architecture, U-Net has lots of parameters to consider for both architecture design and parameters tuning during training. In architecture design, I used up-sampling layers instead of de-convolutional layers for the decoding part. Up-sampling is a reverse of pooling in CNNs in order to reconstruct feature maps into a high-resolution matrix. I used the simplest way for up-sampling, a combination of resampling and interpolation. There is no parameters to learn in up-sampling layers. In contrast, the de-convolutional layer, whose parameters can be learned, is a reverse of convolutional layer. The de-convolutional layer can be considered as an up-sampling step followed by a convolutional step. The reason why I used up-sampling layer in the U-Net is to reduce the number of parameters. It can help prevent overfitting since I have limited computing resources, and not so many training 2D slices and 3D patches.

Since the training of stage 2 is a much more difficult task than that of stage 1, I took efforts to tune parameters for U-Net in stage 2. During the training of U-Net, the most important parameter is the optimizer and its corresponding learning rate. Generally, Adam optimizer [43] and Stochastic Gradient Descent with momentum [71] are two widely-used optimizers for training deep neural networks because of fast convergence. I tried these two optimizers and did not find significant difference. They are both able to achieve convergence within 30 epochs for 2D U-Net and 20 epochs for 3D U-Net. I chose the latest optimizer, Adam, as the one for further parameters tuning. The default learning rate of Adam optimizer is 0.001. Take the 2D U-Net for example. I tried three different learning rates, 0.01, 0.001, and 0.0001. A learning rate of 0.01 is too big to achieve the similar loss of a learning rate of

0.001 and 0.0001. The loss of 0.0001 achieves a 2% lower loss than that of 0.001. Although a learning rate 0.0001 takes more time to train, it can converge in 30 epochs. Another important parameter in training is how to set weights for the weighted Dice coefficient loss. Setting different weights for classes aims to deal with extremely imbalance classes of this task. If all classes were assigned the same weight, U-Net would predict almost each pixel/voxel as the background class because nearly 90 percent pixels/voxels are background. For lobe classes, I tried the same weight for them. The result of prediction shows that no pixel/voxel is predicted as cingulate cortex (left/right) and occipital lobes (left/right). It may be caused by much smaller number of pixels/voxels of these lobes in the image than other lobes. I solved this problem by decreasing the weights of other lobes and increasing the weights of cingulate cortex (left/right) and occipital lobes (left/right).

5.3 2D U-Net vs. 3D U-Net

I used similar procedures for cerebral cortex segmentation of 2D U-Net and 3D U-Net. In stage 1, I aim to delineate grey matter (cerebral cortex) to filter white matter and background for further segmentation. The prediction of stage will be used as a mask to get final results. These three tissue types, grey matter, white matter and background, all have has differences in intensity. Consequently, both 2D U-Net and 3D U-Net perform well in stage 1.

In stage 2, 2D U-Net and 3D-Net are unable to achieve good performance. For 2D U-Net, I need to cut original 3D images into slices on one of three anatomical planes. Sagittal plane separates left from right for the brain. Slices of sagittal plane would not work for lobe (left/right) segmentation because it is hard to figure out which 2D slice is from left brain or right. I finally choose to extract 2D slices on the transverse plane because the cerebral cortex boundary of most slices is much longer than that on the coronal plane. The main limitation of 2D U-Net is that it only takes use of context information on one plane. 2D U-Net does not consider context information within planes and position of the 2D slice in the original 3D image. For 3D U-Net, I aim to take more context information into consideration. However, it is difficult to train a U-Net for original 3D images ($256 \times 256 \times 256$) on a standard GPU

with 11 G memory. So I have to cut the original image into small 3D patches ($64 \times 64 \times 64$). Although 3D U-Net considers local context in the 3D patch, it still loses the position of the 3D patches in the original image and global context information.

For computational complexity, I ran 2D U-Net and 3D U-Net on the same GPU, NVIDIA Tesla K80 (RAM 11G). I used same architecture for stage 1 and stage 2 in 2D/3D U-Net except the dimension of final $softmax$ layer. I also used the same mini batch size and number of epochs for both stages, as shown in Table 4.2 As a result, the time complexity of training 2D/3D U-Net in stage 1 and stage 2 would be almost the same. It takes 20 minutes to train 2D U-Net, and 4.8 seconds for prediction on all slices of one subject. While it takes 3 hours to train 3D U-Net, and 32 seconds for prediction on all small patches of one subject. Since training a model is a one-time work, a long-time training is acceptable. I care more about prediction time because prediction will be done multiple times for to-be-segmented images. In conclusion, 3D U-Net is less efficient than 2D U-Net.

5.4 U-Net vs. FreeSurfer

FreeSurfer contains a popular software package for cerebral cortex segmentation. It provides a robust procedure based on a graphical-based probabilistic information estimation (PIE) method [27] to divide human cortex to sub-regions (cortical sulci and gyri labeling). The main idea of PIE is to combine geometric information and neuroanatomical conventions to build a probabilistic inference model for cortex segmentation. There are three key steps, image alignment, atlas construction and parcellation.

FreeSurfer's DKT classifier achieves over 90% Dice coefficient cortical for labeling sulci and gyri which is a harder task than lobes labeling I did in this thesis. 2D/3D U-Net does not work on sulci and gyri labeling and also not work well on lobes segmentation, 55% Dice coefficient. There are three possible reasons.

The first one is the imbalance of classes. Cerebral cortex accounts for about 15 percent volume of brain MR image after background border is removed. Lobes are part of cerebral cortex. Some small lobes like the left/right occipital lobe only accounts for less than 1

percent volume. Assigning different weights for classes is a potential solution to deal with imbalanced classes. However, the weighted Dice coefficient loss function is very sensitive to weight choosing in this case. An unsuitable choice might lead to no labels assigned to some classes in prediction.

The second one is the limitation of U-Net architecture. Kernels in the U-Net learn various information from training images such as intensity, local texture and context information. However, it does not contain any prior information for cerebral cortex regions. In contrast, FreeSurfer's approach uses three types of prior information for its graphical-based PIE method. There are the probability distribution of label classes at each point, neighbor information and the probability distribution function of value of each point. Since images have been already aligned to a template, these prior information and geometric information contain main features of sulci and gyri.

In addition, 2D U-Net and 3D U-Net have their own limitations, respectively. In contrast to Freesurfer's method of building a graphical model for original 3D images, U-Net cannot use original 3D images as input because of limited memory of standard GPUs. The 2D U-Net and the 3D U-Net in this thesis use tricky ideas to reduce dimension or size of the model input. Consequently, these tricky ideas would lead to information loss of original 3D images. The 2D U-Net architecture loses context information between planes after extracting 2D slices from original 3D images. And the 3D U-Net is unable to capture context information between patches after cutting original 3D images into small 3D patches. Also, 2D U-Net and 3D U-Net both lose the original position of each pixel/voxel in orignal 3D images. The information of position is important in this case. Sulci/gyri or lobes are similar in texture and intensity. However, their position in the brain would be an important feature to distinguish different sulci/gyri or lobes. In future works, researchers might focus on the combination of CNNs with position information for cerebral cortex segmentation.

The third one is the limit number of training data. In the experiment, I have limited computing resources so that I only used a sub-set (20 brains) of the whole Mindboggle data set of 101 MR brains. If more training images were include for 2D/3D U-Net training, higher

performance would be achieved.

Chapter 6

SUMMARY

In this thesis, I tried to use a deep convolutional neural networks architecture called U-Net to label sub-regions of human cerebral cortex from 3D MR images. The measurement of volume and changes in cerebral cortex can help do research on cerebral cortex disorders like Alzheimer's disease. For U-Net implementation, I tried both 2D U-Net and a extended version called 3D U-Net. I implemented a two-stage pipeliment for both 2D U-Net and 3D U-Net. In stage one, a three-class classifier is trained by U-Net to generate a mask of grey matter for next stage. Both 2D U-Net and 3D U-Net achieve approximate 95% of grey matter for stage one. In stage two, 2D U-Net and 3D U-Net achieve only approximate 55% for labeling ten lobes of cerebral cortex. The main reason of poor performance of 2D/3D U-Net is the loss of global position information of pixels/voxels by cutting original into small parts (i.e. 2D slices, 3D small patches). In the feature work, researchers could create hybrid methods to combine deep neural networks architectures with prior information and spacial information to label cerebral cortical sub-regions. The source code of this thesis is now available on GitHub. (`https://github.com/BarryccXu/Unet-for-Biomedical-image-segmentation`)

BIBLIOGRAPHY

[1] *Lobes figure reference*. http://www.thinkfirst.org/youth-lesson10.

[2] *NIH Alzheimer's disease*. https://www.nia.nih.gov/health/alzheimers.

[3] *Stanford CS231n Notes*. http://cs231n.github.io/convolutional-networks/.

[4] *FreeSurfer*, 2013. https://surfer.nmr.mgh.harvard.edu/.

[5] Steven E Arnold, Bradley T Hyman, Jill Flory, Antonio R Damasio, and Gary W Van Hoesen. The topographical and neuroanatomical distribution of neurofibrillary tangles and neuritic plaques in the cerebral cortex of patients with alzheimer's disease. *Cerebral cortex*, 1(1):103–116, 1991.

[6] Alzheimer's Association et al. 2017 alzheimer's disease facts and figures. *Alzheimer's & Dementia*, 13(4):325–373, 2017.

[7] Brian B Avants, Nicholas J Tustison, Gang Song, Philip A Cook, Arno Klein, and James C Gee. A reproducible evaluation of ants similarity metric performance in brain image registration. *Neuroimage*, 54(3):2033–2044, 2011.

[8] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[9] Forrest Sheng Bao, Satrajit S Ghosh, Joachim Giard, Ramin V Parsey, and Arno Klein. Brain shape analysis for predicting treatment remission in major depressive disorder. In *41st annual meeting for the society for neuroscience*, 2011.

[10] James W Bisley and Michael E Goldberg. Attention, intention, and priority in the parietal lobe. *Annual review of neuroscience*, 33:1–21, 2010.

[11] David F Cechetto and Nina Weishaupt. *The Cerebral Cortex in Neurodegenerative and Neuropsychiatric Disorders: Experimental Approaches to Clinical Issues*. Academic Press, 2017.

[12] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. *arXiv preprint arXiv:1707.03718*, 2017.

[13] Cèline Chayer and Morris Freedman. Frontal lobe functions. *Current neurology and neuroscience reports*, 1(6):547–552, 2001.

[14] Patrick Ferdinand Christ, Florian Ettlinger, Felix Grün, Mohamed Ezzeldin A Elshaera, Jana Lipkova, Sebastian Schlecht, Freba Ahmaddy, Sunil Tatavarty, Marc Bickel, Patrick Bilic, et al. Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks. *arXiv preprint arXiv:1702.05970*, 2017.

[15] D Louis Collins, Peter Neelin, Terrence M Peters, and Alan C Evans. Automatic 3d intersubject registration of mr volumetric data in standardized talairach space. *Journal of computer assisted tomography*, 18(2):192–205, 1994.

[16] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In David Hogg and Roger Boyle, editors, *BMVC92*, pages 9–18, London, 1992. Springer London.

[17] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[18] Joseph T Coyle, Donald L Price, and Mahlon R Delong. Alzheimer's disease: a disorder of cortical cholinergic innervation. *Science*, 219(4589):1184–1190, 1983.

[19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[20] Derek Denny-Brown and RA Chambers. The parietal lobe and behavior. *Research Publications of the Association for Research in Nervous & Mental Disease*, 1958.

[21] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[22] Ivana Despotović, Bart Goossens, and Wilfried Philips. Mri segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, 2015, 2015.

[23] Kunio Doi. Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Computerized medical imaging and graphics*, 31(4-5):198–211, 2007.

[24] J Dolz, L Massoptier, and M Vermandel. Segmentation algorithms of subcortical brain structures on mri for radiotherapy and radiosurgery: a survey. *IRBM*, 36(4):200–212, 2015.

[25] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In *Annual Conference on Medical Image Understanding and Analysis*, pages 506–517. Springer, 2017.

[26] Bruce Fischl and Anders M Dale. Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences*, 97(20):11050–11055, 2000.

[27] Bruce Fischl, André Van Der Kouwe, Christophe Destrieux, Eric Halgren, Florent Ségonne, David H Salat, Evelina Busa, Larry J Seidman, Jill Goldstein, David Kennedy, et al. Automatically parcellating the human cerebral cortex. *Cerebral cortex*, 14(1):11–22, 2004.

[28] Richard SJ Frackowiak. *Human brain function*. Academic press, 2004.

[29] Serge Gauthier, Barry Reisberg, Michael Zaudig, Ronald C Petersen, Karen Ritchie, Karl Broich, Sylvie Belleville, Henry Brodaty, David Bennett, Howard Chertkow, et al. Mild cognitive impairment. *The Lancet*, 367(9518):1262–1270, 2006.

[30] Panteleimon Giannakopoulos, Patrick R Hof, Jean-Pierre Michel, José Guimon, and Constantin Bouras. Cerebral cortex pathology in aging and alzheimer's disease: a quantitative survey of large hospital-based geriatric and psychiatric cohorts. *Brain Research Reviews*, 25(2):217–245, 1997.

[31] Pierre Gloor and Alan H Guberman. The temporal lobe & limbic system. *Canadian Medical Association. Journal*, 157(11):1597, 1997.

[32] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. Mapping the structural core of human cerebral cortex. *PLoS biology*, 6(7):e159, 2008.

[33] Lei He, Zhigang Peng, Bryan Everding, Xun Wang, Chia Y Han, Kenneth L Weiss, and William G Wee. A comparative study of deformable contour methods on medical image segmentation. *Image and vision computing*, 26(2):141–163, 2008.

[34] Liesi E Hebert, Paul A Scherr, Julia L Bienias, David A Bennett, and Denis A Evans. Alzheimer disease in the us population: prevalence estimates using the 2000 census. *Archives of neurology*, 60(8):1119–1122, 2003.

[35] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[36] Peter R Huttenlocher. Morphometric study of human cerebral cortex development. *Neuropsychologia*, 28(6):517–527, 1990.

[37] Juan Eugenio Iglesias and Mert R Sabuncu. Multi-atlas segmentation of biomedical images: a survey. *Medical image analysis*, 24(1):205–219, 2015.

[38] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1175–1183. IEEE, 2017.

[39] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, volume 259, page 268, 1987.

[40] Michael R Kaus, Simon K Warfield, Arya Nabavi, Peter M Black, Ferenc A Jolesz, and Ron Kikinis. Automated segmentation of mr images of brain tumors. *Radiology*, 218(2):586–591, 2001.

[41] Anisha Keshavan, Arno Klein, and Ben Cipollini. Interactive online brain shape visualization. *Research Ideas and Outcomes*, 3:e12358, 2017.

[42] Zaven S Khachaturian. Diagnosis of alzheimer's disease. *Archives of neurology*, 42(11):1097–1105, 1985.

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[44] Arno Klein, Brett Mensh, Satrajit Ghosh, Jason Tourville, and Joy Hirsch. Mindboggle: automated brain labeling with multiple atlases. *BMC medical imaging*, 5(1):7, 2005.

[45] Arno Klein and Jason Tourville. 101 labeled brain images and a consistent human cortical labeling protocol. *Frontiers in neuroscience*, 6:171, 2012.

[46] Heinrich Klüver. Visual functions after removal of the occipital lobes. *The Journal of Psychology*, 11(1):23–45, 1941.

[47] Ron Kohavi. Glossary of terms. *Machine Learning*, 30:271–274, 1998.

[48] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[49] Paweł Liskowski and Krzysztof Krawiec. Segmenting retinal blood vessels with deep neural networks. *IEEE transactions on medical imaging*, 35(11):2369–2380, 2016.

[50] Jin Liu, Yi Pan, Min Li, Ziyue Chen, Lu Tang, Chengqian Lu, and Jianxin Wang. Applications of deep learning to mri images: a survey. *Big Data Mining and Analytics*, 1(1):1–18, 2018.

[51] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[52] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[53] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[54] Deborah C Mash, Donna D Flynn, and Lincoln T Potter. Loss of m2 muscarine receptors in the cerebral cortex in alzheimer's disease and experimental cholinergic denervation. *Science*, 228(4703):1115–1117, 1985.

[55] Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: a survey. *Medical image analysis*, 1(2):91–108, 1996.

[56] Bruce L Miller and Jeffrey L Cummings. *The human frontal lobes: Functions and disorders*. Guilford Publications, 2017.

[57] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[58] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.

[59] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[60] RE Passingham. The frontal lobes and voluntary action. oxford psychology series. 1993.

[61] DP Pelvig, Henning Pakkenberg, AK Stark, and Bente Pakkenberg. Neocortical glial cell numbers in human brains. *Neurobiology of aging*, 29(11):1754–1762, 2008.

[62] Olivier Potvin, Louis Dieumegarde, Simon Duchesne, Alzheimer's Disease Neuroimaging Initiative, et al. Freesurfer cortical normative data for adults using desikan-killiany-tourville and ex vivo protocols. *NeuroImage*, 156:43–64, 2017.

[63] Pavel Pudil, Jana Novovičová, and Josef Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994.

[64] Christiane Reitz and Richard Mayeux. Alzheimer disease: epidemiology, diagnostic criteria, risk factors and biomarkers. *Biochemical pharmacology*, 88(4):640–651, 2014.

[65] Torsten Rohlfing, Robert Brandt, Randolf Menzel, and Calvin R Maurer. Segmentation of three-dimensional images using non-rigid registration: Methods and validation with application to confocal microscopy images of bee brains. In *Medical Imaging 2003: Image Processing*, volume 5032, pages 363–375. International Society for Optics and Photonics, 2003.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[67] Mohammed A-M Salem, Alaa Atef, Alaa Salah, and Marwa Shams. Recent survey on medical image segmentation. *Computer Vision: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*, page 129, 2018.

[68] Florent Ségonne, Anders M Dale, Evelina Busa, Maureen Glessner, David Salat, Horst K Hahn, and Bruce Fischl. A hybrid approach to the skull stripping problem in mri. *Neuroimage*, 22(3):1060–1075, 2004.

[69] Dinggang Shen, Scott Moffat, Susan M Resnick, and Christos Davatzikos. Measuring size and shape of the hippocampus in mr images using a deformable shape model. *Neuroimage*, 15(2):422–434, 2002.

[70] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[71] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[72] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[73] Demetri Terzopoulos and Kurt Fleischer. Deformable models. *The visual computer*, 4(6):306–331, 1988.

[74] G Waldemar, B Dubois, M Emre, J Georges, IG McKeith, M Rossor, P Scheltens, P Tariska, and B Winblad. Recommendations for the diagnosis and management of alzheimer's disease and other disorders associated with dementia: Efns guideline. *European Journal of Neurology*, 14(1), 2007.

[75] Yongmei Wang and Lawrence H Staib. Boundary finding with correspondence using statistical shape models. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 338–345. IEEE, 1998.

[76] Yan Xu, Chenchao Xu, Xiao Kuang, Hongkai Wang, Eric I Chang, Weimin Huang, Yubo Fan, et al. 3d-sift-flow for atlas-based ct liver image segmentation. *Medical physics*, 43(5):2229–2241, 2016.

[77] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

[78] Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.