

**U-NET AND BOUNDARY PRESERVING BLOCK LEARNING
ALGORITHMS FOR 3D BRAIN TUMOR SEGMENTATION**

BY

SAMUEL TAN JOO WOON

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

**Faculty of Information and Communication Technology
(Kampar Campus)**

JANUARY 2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr Mogana a/p Vadiveloo who has given me this bright opportunity to engage in brain tumor segmentation project and also has given me a lot of assistance and advice along the way, especially during the time to find GPU for the model training. It is my first step to participate in medical image processing field that has help me to develop knowledge and get a hands-on experience in medical image segmentation. A million thanks to you.

Furthermore, I would also appreciate my parents and my family that gives me support throughout the course. Without them, I would never have the opportunity to breakthrough myself in my university life. Finally, I must say thank you to the COVID-19 pandemic that makes everyone stay at home during the MCO, it has help me to stay focus in my FYP without wasting time watching Tokyo Olympic at the mamak stall.

ABSTRACT

This project proposed an application that performs 3-dimensional (3D) brain tumor segmentation using deep learning model, U-Net and Boundary Preserving Block (BPB). Brain tumor segmentation is a crucial task to help the physicians to visualize the tumor region from the Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) and make the decision in the treatment plan. In order to accelerate the diagnosis process, many automatic and interactive image segmentation approaches have been proposed to outperform the brain tumor segmentation. The software such as ITK-SNAP, MIPAV, 3D Slicer and Olea Sphere® are the software that is currently available in the market that can perform the brain tumor segmentation. The problem lies with the ambiguous boundary of the brain tumor region in the image has caused the automatic segmented approaches in the mentioned software to produce under or over segmented tumor region. While the semi-automatic approaches are time consuming and labor intensive. The proposed system aims to visualize and perform automatic segmentation to the 3D brain tumor images. In this work, Qt and VTK framework is used to visualize the 3D brain tumor images written in C++ with Microsoft Visual Studio. By visualizing the brain tumor image, the user can view and interact with the 3D brain tumor images in three different planes in one user interface. In addition, deep learning model U-Net and BPB is implemented to perform the 3D brain tumor segmentation. Due to lack of GPU for training purpose, the size of dataset has been reduced to 30 images in the training set and 10 images in testing set, also the number of training epoch has been limited to 10. During the evaluation with testing set, the model of U-Net with the best result is in epoch 8 with IOU and DSC score of 0.06 and 0.1 respectively. The model of U-Net achieves IOU and DSC score of 0.02 and 0.04 respectively in epoch 6 with testing set.

TABLE OF CONTENTS

TITLE PAGE	i
REPORT STATUS DECLARATION FORM	ii
SUBMISSION SHEET FOR FYP	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xii
LIST OF ABBREVIATIONS	xiii
APPENDIX A	ix
A-1	ix
Chapter 1: Project Background	1
Introduction	1
1.1 Problem Statement	2
1.2 Project Objective	3
1.3 Project Scope	3
1.4 Impact and Contribution	4
1.5 Report Organization	4
Chapter 2: Literature Review	5
2.1 Objective of Literature Review	5
2.2 ITK-SNAP	5
2.2.1 User Interface	6
2.2.2 Feature and Functionalities	7
2.2.3 Advantages and Disadvantages	8
2.3 MIPAV	9
2.3.1 User Interface	9
2.3.2 Feature and Functionalities	10
2.3.3 Advantages and Disadvantages	11
2.4 3D Slicer	12
	vii

2.4.1	User Interface	12
2.4.2	Feature and Functionalities	13
2.4.3	Advantages and Disadvantages	14
2.5	Olea Sphere®	16
2.5.2	Feature and Functionalities	16
2.5.3	Advantages and Disadvantages	16
2.6	Critical Remark	17
Chapter 3: System Design		19
3.1	Introduction	19
3.2	System Design Diagram	19
3.2.1	Use Case Diagram	19
3.2.2	Activity Diagram	21
3.2.3	Wireframe	22
3.2.4	Flowchart of system architecture	23
3.3	Chapter Conclusion	23
Chapter 4: System Implementation		24
4.1	Overview	24
4.2	System Development Methodology	24
4.3	Hardware Setup	26
4.4	Software Setup	26
4.4.1	Visualization Module	26
4.4.2	Segmentation Module	27
4.5	Brain Tumor Dataset	28
4.6	Algorithm used	30
4.6.1	U-Net	30
4.6.2	Boundary Preserving Block (BPB)	31
4.7	Performance Metrics	33
4.8	Project Challenges and Timeline	34
4.8.1	Implementation issues and challenges	34
4.8.2	Timeline	35
4.9	Chapter Summary	37
Chapter 5: System Evaluation and Discussion		38
5.1	3D brain tumor images visualization	38
5.2	Segmentation Result	40
5.3	Chapter Conclusion	43
Chapter 6: Conclusion and Recommendation		44

6.1	Conclusion & Novelty	44
6.2	Future work	44
References		45
 APPENDIX A		 A-1
A.1	Final Year Project Weekly Report	A-1
A.2	Poster	A-7
 PLAGIARISM CHECK RESULT		
FYP 2 CHECKLIST		

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	3D MRI brain tumor images	1
Figure 1.2	Brain tumor region that has ambiguous boundary	2
Figure 2.1	Graphical User Interface (GUI) of ITK-SNAP	6
Figure 2.2	Interface to define parameters values for semi-automatic thresholding segmentation of ITK-SNAP	7
Figure 2.3	Result of 3D brain tumor segmentation using semi-automatic thresholding approach	8
Figure 2.4	MIPAV Graphical User Interface (GUI)	10
Figure 2.5	Result of mean-shift segmentation. The left image is the original image followed by the middle image segmented by mean shift at the first attempt while the right image is the image generated by mean shift at the third attempt with refined parameters values.	11
Figure 2.6	Graphical User Interface (GUI) of 3D Slicer	13
Figure 2.7	3D Brain Magnetic Resonance Imaging (MRI)	14
Figure 2.8	Under segmented 3D brain tumor region with ambiguous boundary using region-growing algorithm in 3D Slicer	14
Figure 3.1	Use Case Diagram	20
Figure 3.2	Activity Diagram	21
Figure 3.3	Wireframe of the proposed system	22
Figure 3.4	Flowchart of system architecture	23
Figure 4.1	Waterfall model	25
Figure 4.2	Example images of BraTS 2015 training dataset	29
Figure 4.3	2D U-Net architecture	31
Figure 4.4	3D U-Net architecture	31
Figure 4.5	Detail structure of Boundary Preserving Block	32

Figure 5.1	3D brain tumor image visualization interface of the proposed system	38
Figure 5.2	VTK visualization pipeline	39
Figure 5.3	Partial code of image visualization	40
Figure 5.4	The segmentation result with brats_tcia_pat319_0001 in testing set compared between original image, ground truth image, segmentation result (U-Net), segmentation result (U-Net with BPB) in (a), (b), (c) and (d) respectively	41
Figure 5.5	The segmentation result with brats_tcia_pat343_0001 in testing set compared between original image, ground truth image, segmentation result (U-Net), segmentation result (U-Net with BPB) in (a), (b), (c) and (d) respectively	42

LIST OF TABLES

Table Number	Title	Page
Table 2.1	Critical Remark of ITK-SNAP, MIPAV, 3D Slicer and Olea Sphere	18
Table 4.1	Project I timeline	35
Table 4.2	Project II timeline	36

LIST OF ABBREVIATIONS

<i>ASCO</i>	American Society of Clinical Oncology
<i>CNS</i>	Central Nervous System
<i>CT</i>	Computed Tomography
<i>MRI</i>	Magnetic Resonance Imaging
<i>3D</i>	Three Dimensional
<i>ROI</i>	Region of Interest
<i>XML</i>	Extensible Markup Language
<i>GUI</i>	Graphical User Interface
<i>BPB</i>	Boundary Preserving Block
<i>mha</i>	MetaImage File Format
<i>CPU</i>	Central Processing Unit
<i>GPU</i>	Graphics Processing Unit
<i>FCM</i>	Fuzzy c-means
<i>ABC</i>	Atlas Based Classification
<i>OS</i>	Operating System
<i>RAM</i>	Random Access Memory
<i>VRAM</i>	Video Random Access Memory
<i>ITK</i>	The Insight Toolkit
<i>VTK</i>	The Visualization Toolkit
<i>IDE</i>	Integrated Development Environment
<i>LGG</i>	Low Grade Glioma
<i>HGG</i>	High Grade Glioma

<i>ReLU</i>	Rectified Linear Unit
<i>SBE</i>	Shape Boundary-aware Evaluator
<i>IOU</i>	Intersection Over Union
<i>DSC</i>	Dice coefficient

Chapter 1: Project Background

Introduction

Brain tumor is a growth of a mass abnormal cell in the brain tissues causing inflammation and increased pressure on the nearby tissues. According to the report from Cancer.Net, a website from American Society of Clinical Oncology (ASCO), the average 5-year survival rate of malignant brain tumor for all ages is only 36% and 10-year survival rate for all ages is only 31% in United State (US). It is also estimated that 18,020 adults in the US will die from cancerous brain and Central Nervous System (CNS) tumor in year 2020 [1]. Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are the commonly used techniques to visualize the brain tumor region. The three-dimensional (3D) images taken from these scans will help the physicians to diagnose the disease. An example of this 3D brain tumor image is shown in Figure 1.1 where the tumor region is marked in green. However, the physicians often required to manually examine the 3D brain images from different angles to determine any abnormalities or tumors. This process is a cumbersome and time-consuming task to the physicians. Image segmentation is the most essential medical imaging process where it extracts the regions of interest (ROI) or tumors through semi-automatic and automatic approaches. It helps the physicians to accelerate the tumor diagnosis to stay focused on the treatment plan. [2]

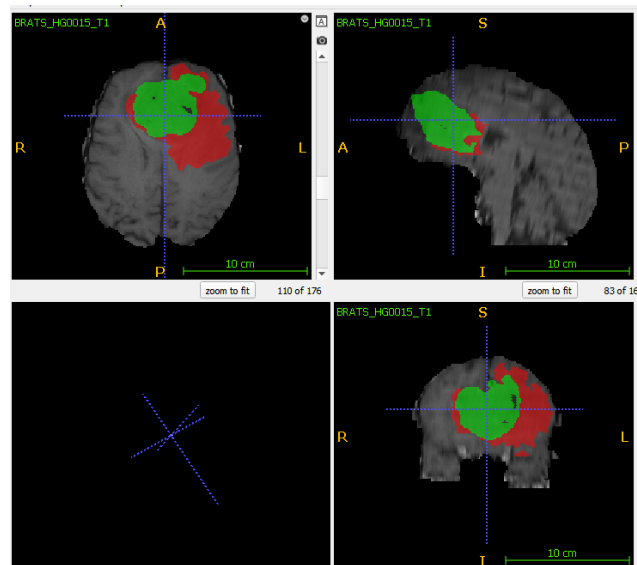


Figure 1.1: 3D MRI brain tumor images. The tumor regions are labeled in green.

1.1 Problem Statement

There are many semi-automatic and automatic segmentation methods that are currently available to help the physicians to speed up the segmentation process. The example of the software that performs the 3D brain tumor segmentation are ITK-SNAP [3], MIPAV [4], 3D Slicer [5] and Olea Sphere® [6]. Nevertheless, these softwares have their own weaknesses. The brain tumor region has ambiguous boundary separation between its region and the neighbouring region due to similar pixel intensity as shown in Figure 1.2. In this figure, the labelled tumor region has similar intensity with its neighbouring grey matter brain tissue regions. The existing software commonly use semi-automatic approaches to perform the 3D brain tumor segmentation. These approaches require users to draw or annotate the tumor regions or define the parameters, to manually perform the segmentation. As the 3D medical images are high in resolution, these semi-automatic segmentation approaches are time-consuming and labor intensiveness. In addition to that, since brain tumor suffer from ambiguous boundary, segmentation of these regions as region of interest (ROI) using the existing software becomes challenging.

In most cases, over-segmented tumor regions are generated where the tumor is segmented in many tiny regions which are difficult to be interpreted. While in some cases, under segmented tumor regions are produced.

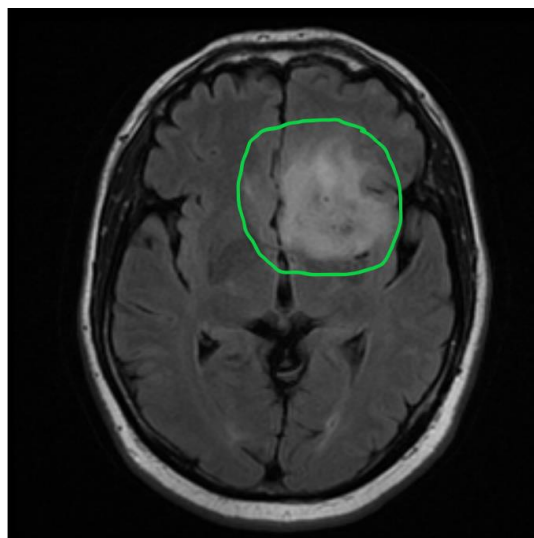


Figure 1.2: Brain tumor region (labelled in green) that has ambiguous boundary separation with grey matter regions.

1.2 Project Objective

The main objective is to develop an application to perform accurate segmentation of the 3D brain tumor images to some extent. To achieve the main objective, the sub-objectives are listed below:

1. To develop an application that is able to visualize the 3D brain tumor images in three different orthogonal views on a single graphical user interface (GUI) screen.

The 3D brain tumor images are visualized in three different orthogonal views using Microsoft Visual Studio on a single GUI screen. The three orthogonal views are axial, coronal, and sagittal planes. This implementation is to ease to user to view the 3D brain tumor regions effectively.

2. To develop an application that performs segmentation to the 3D brain tumor using deep learning model, UNet and boundary preserving segmentation framework, Boundary Preserving Block (BPB). In this work, UNet is chosen since it is more recent and popular technique that being used in medical imaging field to provide automatic segmentation without the interaction of human. At the same time, BPB is used to predict the boundary of the tumor region in the model to utilize the boundary information to provide better segmentation result.

1.3 Project Scope

The scopes of the proposed system are as follow:

1. The dataset being used in this project is limited to the three-dimensional (3D) grayscale medical images focusing on brain tumor images. These 3D brain tumor medical images would be in MetaImage (mha) file.
2. The rules derivation in the brain tumor region segmentation identification is based on the common understanding (or expert's opinion) on the datasets. Expert's opinion is required in the segmentation identification of the tumor regions in the 3D brain images.

1.4 Impact and Contribution

The main impact and contribution of this work is the development of an application that performs accurate segmentation of the 3D brain tumor images to some extent. The sub-contributions are as follows:

1. An application that visualizes the 3D brain tumor images in three different orthogonal views on a single graphical user interface (GUI) screen.
2. An application that delineates 3D brain tumor region using fully automated deep learning model, UNet and Boundary Preserving Block (BPB).

1.5 Report Organization

The details of this report are shown in the following chapters. In Chapter 2, existing software that perform the similar task will be introduced and the strength and the weakness of these software are reviewed. Then, the system design diagrams are discussed in Chapter 3. In Chapter 4, the details of system implementation are presented, and Chapter 5 is the system evaluation and result for this project. At last, Chapter 6 presented the conclusion and recommendation for this project.

Chapter 2: Literature Review

2.1 Objective of Literature Review

The purpose of literature review is to explore and evaluate different medical imaging software that enables visualization and/or segmentation of medical images in three-dimensional (3D) mainly focusing on brain tumor images. In addition, this literature review will list out the advantages and disadvantages of the reviewed software on the feature and functionalities as well as the user interface and user experience. By this, it would help to understand the challenges and difficulties that are currently faced by the medical imaging software in visualizing and segmenting 3D brain tumor images.

2.2 ITK-SNAP

ITK-SNAP [3] is an open-source software application that enables the user to view and perform segmentation on the three-dimensional (3D) brain tumor medical images. It is created in the early 21st century with the purpose to provide a tool that is easy to use for non-computer experts to learn and perform segmentation in the 3D brain tumor images [7] [8]. The software was produced after a long-time collaboration between Dr Paul Yushkevich, of the Penn Image Computing and Science Laboratory (PICS�), at the University of Pennsylvania, and Guido Gerig, PhD, of the Scientific Computing and Imaging Institute (SCI) at the University of Utah [3]. It is a software that is mainly focuses the problem of segmentation, while other features are only kept if it is necessary. The software is developed in C++, and it is supported in the major computer operating system such as Windows, Linux, and MacOS.

2.2.1 User Interface

The user interface of the ITK-SNAP is simple and straightforward to use, it supports to load multiple tumor images in various common file formats, such as DICOM and NIFTI. The user interface of the software is based on Qt, a C++ GUI framework, where it gives a modern graphical interface to the user. The program will display the images in three orthogonal views, axial, coronal and sagittal planes as shown in Figure 2.1. In Figure 2.1, the highlighted region is the tumor region, where it is performed by manual segmentation. In every plane of the images, there are two lines that intersect each other. The user can move the lines where it also moves the lines in other planes, so that the user able to navigate through the 3D volumes to evaluate the brain tumor and tissues regions.

The software is also able to display multiple images at the same time. It can be visualized in through *thumbnail* layout, where it only shows one image at a time. In addition, the user can change the image to display by clicking the thumbnail of another image or *tiled* layout, where multiple images are display side to side. The left side of the software provides a toolbox to the user to allow user to perform different operation to the image such as pan/zoom, drawing, etc.

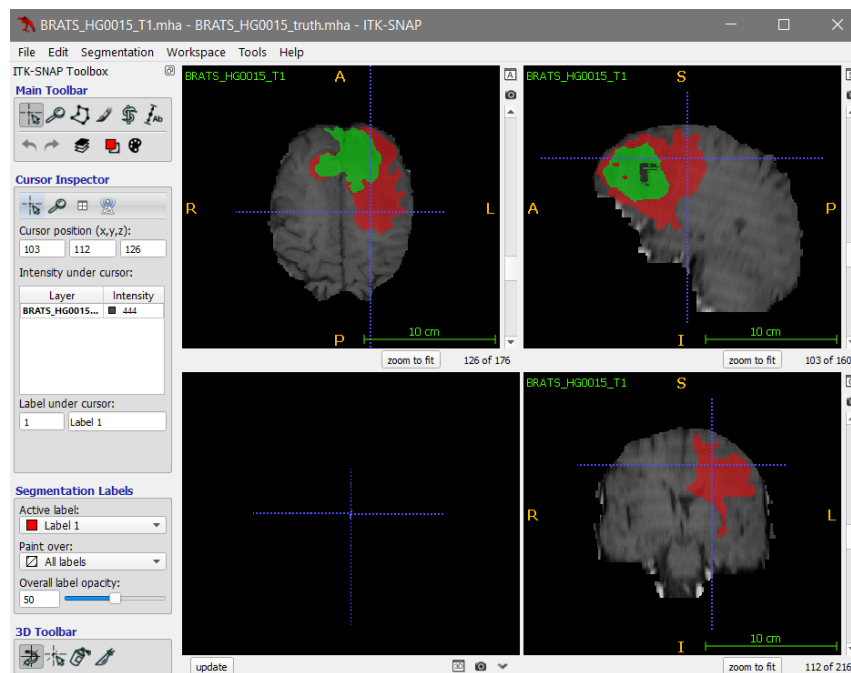


Figure 2.1: Graphical User Interface (GUI) of ITK-SNAP

2.2.2 Feature and Functionalities

In general, ITK-SNAP is a very specific software that mainly focusing on providing the functionality to perform segmentation to the user. It provides semi-automatic segmentation methods, where the user is required to provide parameters to allow the methods to perform the tumor region segmentation. These methods are categorized into thresholding, classification, clustering, edge detection, and active contour segmentation. In the Figure 2.2, it shows the step during the semi-automatic segmentation using thresholding, where the user is required to set the parameters values manually before proceeding to the next steps. The result of the segmentation is then shown in Figure 2.3, where the result in the tumor region was under segmented due to lack of expertise of the user.

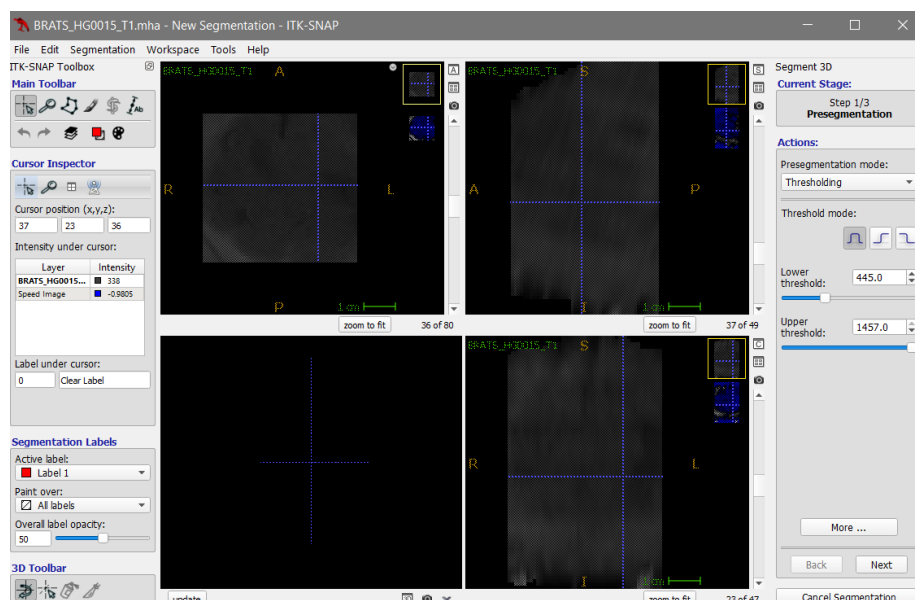


Figure 2.2: Interface to define parameters values for semi-automatic thresholding segmentation of ITK-SNAP

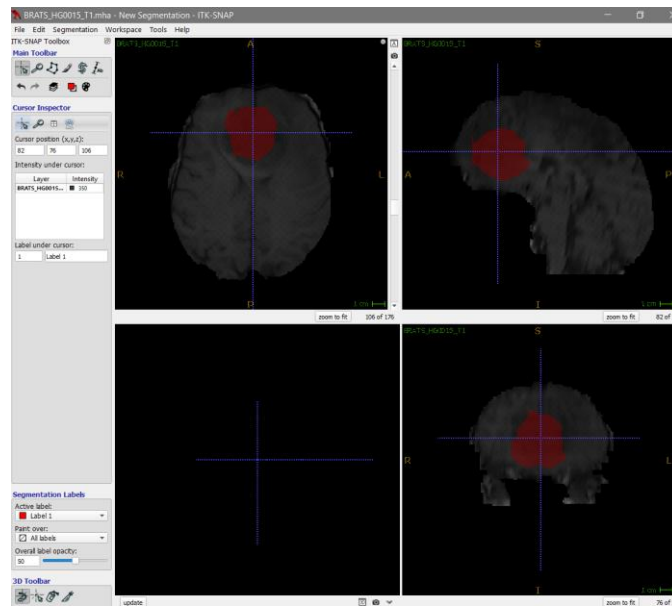


Figure 2.3: Result of 3D brain tumor segmentation using semi-automatic thresholding approach.

2.2.3 Advantages and Disadvantages

2.2.3.1 Advantages

The biggest advantage of this software is the user interface is very friendly for a first-time user even without instruction. Since the unrelated features of the software is keep in the minimum, the software is very easy to use for viewing and perform segmentation on the 3D brain tumor images using semi-automatic segmentation methods. Linked cursor is provided for the user to navigation through the 3D image easily. Besides that, the software also supports many different 3D medical images file format such as DICOM, GIPL and NifTI. The software also come with tutorial and video documentation to allow the user to learn the software in the short amount of time. It is free to use, open-source and runs in multiple platforms.

2.2.3.2 Disadvantages

The first disadvantages in the system to be discussed is the system only has a limited number of features. This is because the software is purposely designed to be easy to use, hence, the advance features have been comprised to keep the software simple and prevent feature creep. Next, the software is also unable to perform the segmentation of

the tumor regions automatically. It only supports semi-automatic or manual image segmentation which require the involvement of the user to perform the segmentation. Hence, the user must acquire the knowledge of the tumor region to carry out an accurate segmentation despite considering the tumor property where commonly it suffers from ambiguous boundary. In addition to that, these interactive methods are labor intensive and time consuming while performing segmentation on 3D medical images.

2.3 MIPAV

MIPAV or Medical Image Processing, Analysis, and Visualization [4] is a medical imaging software that allow the researcher and medical worker to perform quantitative analysis and visualization of medical images from different medical devices/scanners such as MRI, CT, PET or microscopy. The software is released by National Institutes of Health, it is created to support scientific research and clinical practice. It is a Java application and platform independence, it can be run in any platform that support Java including Windows, Linux and Mac Operating Systems (OS). The software support to view the images in different format such as DICOM, RAW, or even XML that follow MIPAV XML schema. MIPAV provides plug-in in Java, where the user can even develop their own plug-in to suit their needs or using official and unofficial plug-in to extend the feature and functionalities of the software [4].

2.3.1 User Interface

The user interface is split and separated into multiple windows such as main toolbar, output, and images (if there are images being opened) as shown in Figure 2.4. The interface can be confusing to the user as the user needs to navigate through different windows to check on the different parts of the software. However, it does provide many tools for the user to deal with the images. It also shows the usage of Graphics Processing Unit (GPU) and Central Processing Unit (CPU) to allow the user keep track on it.

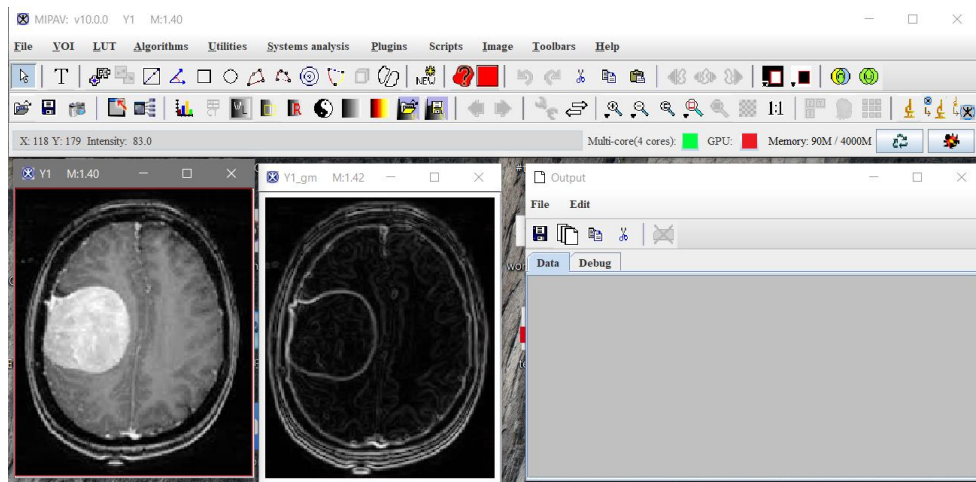


Figure 2.4: MIPAV Graphical User Interface (GUI)

2.3.2 Feature and Functionalities

MIPAV provides a ton of the feature to the user, all the software tools are implemented as plug-ins to the software package as developed at the Center for Information Technology, National Institutes of Health [4]. As for the image segmentation, the software provides many algorithms to the user such as pixel-based and region-based algorithm. The examples of the pixel-based algorithm include *thresholding* and *level-set* while region-based algorithm such as *mean-shift* and *watershed* are available in the software. However, this often comes with the issue similar to ITK-SNAP where the user is required to manually specify each and every tumor region or define parameters to perform the segmentation. This means for high-resolution 3D images to be segmented, this interactive method is going to be a tedious job and time-consuming. In addition to that, though these algorithms able to delineate the brain tumor region to some extent, however, they produce over-segmentation. In Figure 2.5, the left image is the original 3D brain tumor image. The middle image shows the tumor region with ambiguous boundary segmentation at the first attempt using mean shift algorithm with predefined parameters. At this first attempt, over segmented tumor regions were produced. While, at the third attempt, once the parameters values have been further refined, a good segmentation is obtained as shown in the right image of Figure 2.5. Hence, it can be concluded that the expertise is needed to define the parameters value to obtain a good tumor region segmentation.

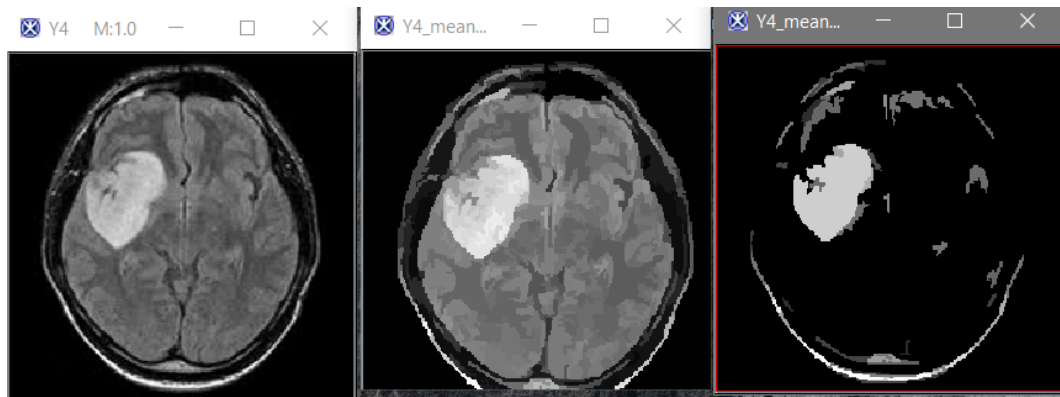


Figure 2.5: Result of *mean-shift* segmentation. The left image is the original image followed by the middle image segmented by mean shift at the first attempt while the right image is the image generated by mean shift at the third attempt with refined parameters values.

2.3.3 Advantages and Disadvantages

2.3.3.1 Advantages

MIPAV is a comprehensive software that enables the user to perform various operations in medical image processing. It provides a bundle of features and algorithms to let the user to choose from and to determine the best algorithm to be used based on the user needs or requirements. In term of 3D brain tumor image delineation, the software is able to manually segment the tumor region to some extent using the pixel and region-based algorithms that are available in the software. In addition to that, the software is free to use and run in any Java-enabled platform. The software also able to utilize multiple core and GPU to render the image faster and support various file format such as DICOM, RAW, JPEG, etc.

2.3.3.2 Disadvantages

The disadvantages of MIPAV are it may be hard to learn and steep learning curve due to the complexity of the system. The user needs to have technical knowledge on image processing to use the software. Moreover, the layout of the software is separated into different windows and causing inconvenience to the user. In term of result of 3D brain tumor segmentation, it provides many algorithms to delineate the tumor regions the result produced may be satisfactory to some extent despite there is a need for users to

manually specify the tumor region or define some parameters. However, the major issue is found that over-segmentation occurs while performing segmentation on brain tumor images with ambiguous boundary.

2.4 3D Slicer

3D Slicer [5] is a free open-source application that released under BSD license for medical image analysis, image processing and three-dimensional (3D) visualization. The software also can support advanced features such as automatic segmentation and image fusion. The software is built more than 20 years through the support from the US National Institutes of Health (NIH) and open-source community. It is firstly presented by David Gering in his MIT Master's thesis in 1999 for the software prototype. Then, Steve Pieper takes the roles of Chief Architect and the application experience continuous development under the leadership by Ron Kikinis since 1999. The software is later received the contribution from professional engineers that working together with the algorithm developer and application domain scientist, with the support from different organization including Kitware Inc., Isomics Inc. and GE Global Research, and the contribution of growing open-source community [9] [10]. It is written in C++, Python and using Qt as Graphical User Interface (GUI) framework, and it is natively designed to run in multiple platforms, including Windows, Linux and MacOS [5].

2.4.1 User Interface

The user interface used to develop using KWWidgets. Since version 4, Slicer has transited from KWWidget to Qt framework, the GUI is simple and straightforward. The toolbar is displayed on top of the interface while the images are loaded in the right region as shown in Figure 2.6. The left region is to insert any parameters for operation such as segmentation. The console is in the bottom region for python console. The software also comes with extension manger where the user can pick and install any extension to use additional feature. However, to perform some advanced operation such as segmentation, it can be confusing.

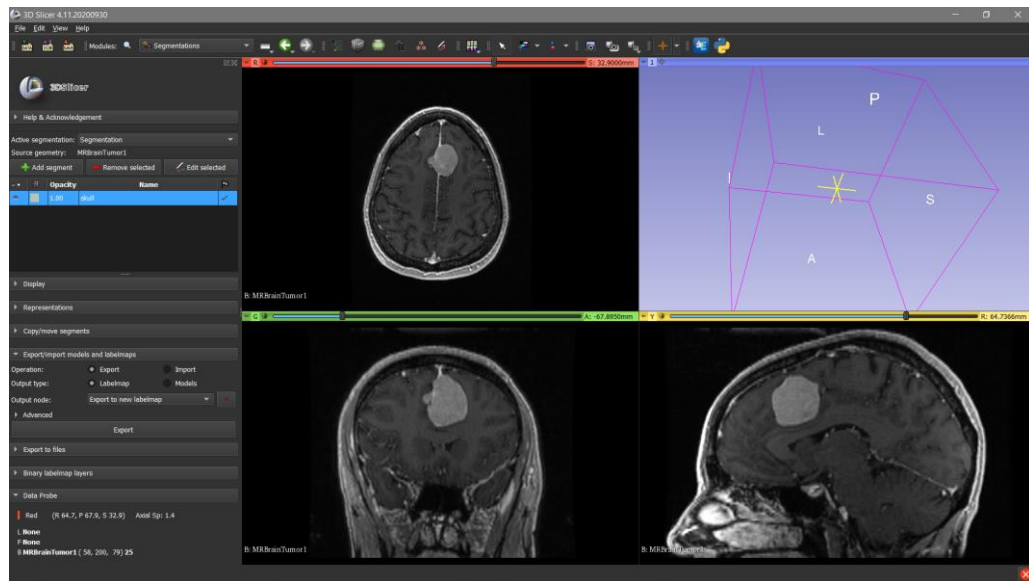


Figure 2.6: Graphical User Interface (GUI) of 3D Slicer.

2.4.2 Feature and Functionalities

3D slicer is only intended for medical research purpose only, it is not U.S. Food and Drug Administration [11] approved and not suitable for clinical use. However, it is a powerful software to visualization the images in two, three and four dimensions and has a variety of plug-ins to extend the application for even more features. The software supports DICOM standard image file and other various common file types such as NifTI, GIPL, PNG, etc. In addition, the application provides a built-in tool for segmentation using segmentation editor, the module gives several effects and tools to create segmentation. The segmentation algorithms that provided in 3D Slicers can be categorized into voxel-based method and region growing algorithm. The voxel-based method such as *otsu threshold* and *foreground masking (BRAINS)*, while region-based algorithm such as *region-growing* and *robust statistics* are default segmentation algorithm that available in the software. More algorithms such as *Fuzzy c-means (FCM)* and *Atlas Based Classification (ABC)* are available in the software through the extension. However, all these tools appear the same issue as ITK-SNAP and MIPAV where the user will also need to interact with the parameter manually to delineate the tumor region. Figure 2.7 show the original 3D brain tumor MRI image, while Figure 2.8 shows the segmented result using *region-growing* algorithm. As the segmentation

result, shown in Figure 2.8, an under segmentation occurred in segmenting brain tumor regions with ambiguous boundary.

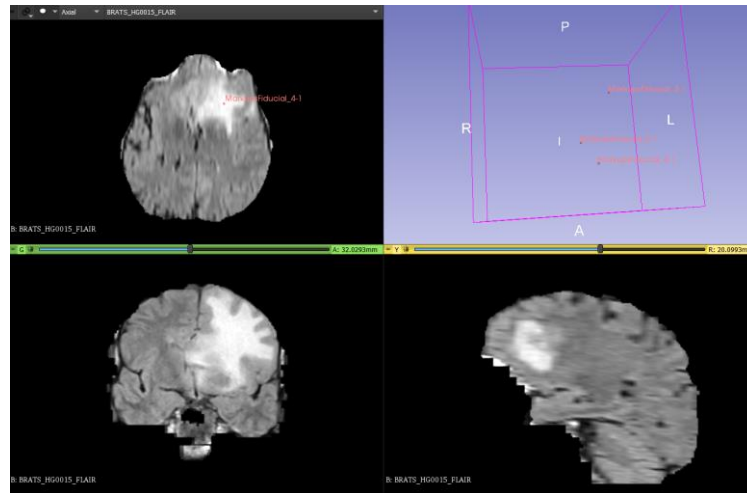


Figure 2.7: 3D Brain Magnetic Resonance Imaging (MRI)

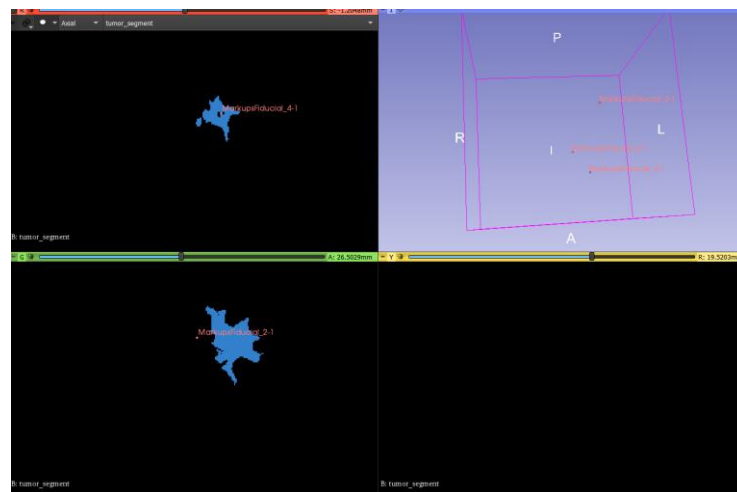


Figure 2.8: Under segmented 3D brain tumor region with ambiguous boundary using *region-growing algorithm in 3D Slicer*.

2.4.3 Advantages and Disadvantages

2.4.3.1 Advantages

One of the advantages of 3D slicer is its simplicity for extending features. An extension manager is provided in the software, where the user can select the extension that is useful to them without download the whole extension archive. Anyone can create the extension and share with the other user through the extension server. This makes

everyone can participate in the software development to make the software even better. Moreover, the software also provides a lot of flexibility to deal with brain tumor segmentation by providing large number of effects and interactive segmentation framework. Huge unknown elements and location of brain tumor sometimes make the fully automatic process difficult to achieve. As a result, interactive framework and single-subject analysis give more choice to the user to perform faster and more accurate segmentation [10]. In addition to that, the software is an open-source and freely available to the public and support various file format of the images.

2.4.3.2 Disadvantages

The disadvantage of the 3D slicer is the software is only intended for research purpose and not a medical product. It does not approve by any regulatory authority for daily clinical use. The software has multiple algorithms to perform 3D brain tumor segmentation such as *thresholding* and *Fuzzy c-means* (FCM) clustering algorithm [12]. The result of thresholding performs accurate the tumor segmentation. However, the user is required to select the threshold parameter value to perform the segmentation. As such, the user needs to judge based on experience or perform tentative segmentation before setting threshold value. While the FCM clustering algorithm requires high computational time to delineate a 3D brain tumor region. In FCM segmentation if the tumor region area to be segmented has smoother edges, the user will need less interaction to perform the segmentation. However, if the tumor region suffers from ambiguous boundary, then the number of interactions increases causing longer execution time and under segmentation might occur too [13].

2.5 Olea Sphere®

Olea Sphere® [6] is Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) medical imaging software that is provided by Olea Medical for image post-processing. It is an established software that is used by many leading institutions worldwide, both for clinical and research purposes. The software provides a complete CT and MRI solution for different cases such as neurology, oncology, cardiology etc. It features many components such as Olea Pulse, Olea Vision, etc to serve different image processing operation, including automatic 3D image rendering and image segmentation. The software is approved with constraints by United States Department of Veterans Affairs [14] to be used as medical software and able to run in different platform such as macOS, Windows and Linux. However, the software is closed source and license is required to use the software.

2.5.2 Feature and Functionalities

Olea Sphere® is a powerful software in MRI and CT imaging that approved by U.S. Food and Drug Administration (FDA) for commercial use. It features variety of functionalities including image viewing, image manipulation, image processing, and etc to the medical images that generated from MRI or CT systems. The feature of Olea Sphere® is extendable by installing plug-ins according to the user's need. As for the brain tumor segmentation, Olea Sphere® is able to perform automatic or manual background segmentation. However, the segmentation algorithm being used in the software is not mentioned. It is only present to the user with the name "Magic Wand" to perform automatic segmentation to the tumor region [15].

2.5.3 Advantages and Disadvantages

2.5.3.1 Advantages

The biggest advantages of Olea Sphere® is it is a professional commercial software that approved by medical authority to be use in clinical setting. The software comes with complete solution to serve different needs in medical image processing such as breastscape® for breast biopsy solution and neuroradiology for neurology imaging solution. The user also can extend the feature of the software through the use of

additional plugins. Olea Sphere® is performs fully automatic brain tumor segmentation and user does not need to acquire the knowledge of image processing to get the segmented result as it has been automated for parameter setting. From the case study provided Olea Medical, it shows that the software can delineate the desired brain tumor region correctly [16].

2.5.3.2 Disadvantages

As for the disadvantages for Olea Sphere®, the software is not free to use and closed source. This means that it is not known how Olea Sphere® is really dealing with the segmentation from the technical perspective and the users do not has much control over the segmentation setting such as the algorithm used. Furthermore, the evaluation on the brain tumor segmentation with ambiguous boundaries with the software is unknown as it is not free to use. The case report released by Olea Medical does not contain the image of brain tumor that has ambiguous boundaries in the tumor region.

2.6 Critical Remark

This section shows the summarization on the advantages and disadvantages of software that have reviewed in Section 2.2 to 2.5 as shown as Table 2.1. The table mainly shows the pros and cons of these software related to brain tumor segmentation. Both MIPAV and 3D Slicer support automatic segmentation, but with the complexity of the software, both software also become hard to learn. All three software needs user involvement to some extends when delineating brain tumor region that has ambiguous boundary. Over segmentation occurred with the automatic segmentation of brain tumor region with ambiguous boundary in MIPAV, while 3D Slicer suffers from longer processing time and under segmentation in brain tumor delineation. As for Olea Sphere, it is a professional software that used in clinical setting for brain tumor segmentation. However, the software is not free to use, so the evaluation with the software is unable to be performed.

Table 2.1 Critical Remark of ITK-SNAP, MIPAV, 3D Slicer and Olea Sphere

Software	Advantages	Disadvantages
ITK-SNAP	<ul style="list-style-type: none"> • User friendly • Free to use • Offers manual and semi-automatic segmentation 	<ul style="list-style-type: none"> • Limited number of functions • Not supported for fully automatic segmentation • User involvement and image processing knowledge is required to delineate brain tumor region with ambiguous boundary
MIPAV	<ul style="list-style-type: none"> • Lots of features • Tons of algorithm available to deal with delineation of brain tumor • Supports automatic segmentation 	<ul style="list-style-type: none"> • Hard to learn • Over-segmentation issue in brain tumor region with ambiguous boundary • Only for research purpose
3D Slicer	<ul style="list-style-type: none"> • Extendibility through plug-ins • Easy manual segmentation with interactive framework • Supports automatic segmentation 	<ul style="list-style-type: none"> • Hard to learn • Longer processing time in brain tumor delineation • Under segmentation might occur when dealing with tumor regions with ambiguous boundary
Olea Sphere	<ul style="list-style-type: none"> • Professional software for segmentation • Supports automatic segmentation 	<ul style="list-style-type: none"> • Not free • Closed source • Segmentation result on brain tumor with ambiguous boundary is unknown

Chapter 3: System Design

3.1 Introduction

In this section, various system design diagrams of the proposed system are discussed. This includes use case diagram, activity diagram, wireframe and flowchart of system architecture. This chapter will give an overview of the proposed system in diagram.

3.2 System Design Diagram

3.2.1 Use Case Diagram

The use case diagram of proposed system is shown in Figure 3.1. The user is able to view the 3D brain image in three orthogonal views, which are axial, coronal and sagittal plane and the user can interact with the slice of the image in each plane through the slider provided. Next, the user can perform the brain tumor segmentation with a click of button after the image is loaded and defined a seed point. The system will generate the segmented result and the user will then be able to view and save the result. The segmented result will display in three different planes allow the user to see through the slice of the image and the result can be save in MetaImage file format to the local drive.

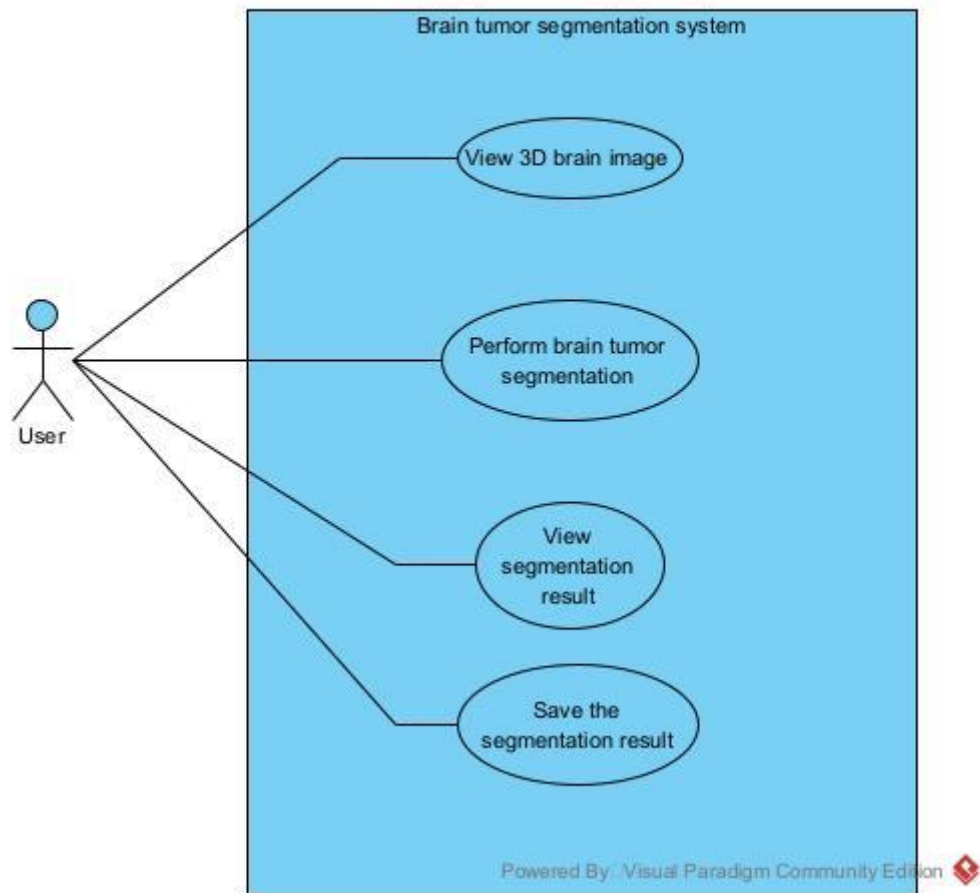


Figure 3.1. Use Case Diagram

3.2.2 Activity Diagram

The activity diagram of proposed system is shown in Figure 3.2. When the system is loaded, the system will display a single graphical user interface (GUI) screen that allow the user to load and display image in 3 different planes. To perform the segmentation, the user will first select a seed point. Then, the user can start the segmentation process. After the segmentation is done, the system will display the segmented result to the user to let the user view the segmented result. The user will then be able to save the segmented result to the local drive in MetaImage format.

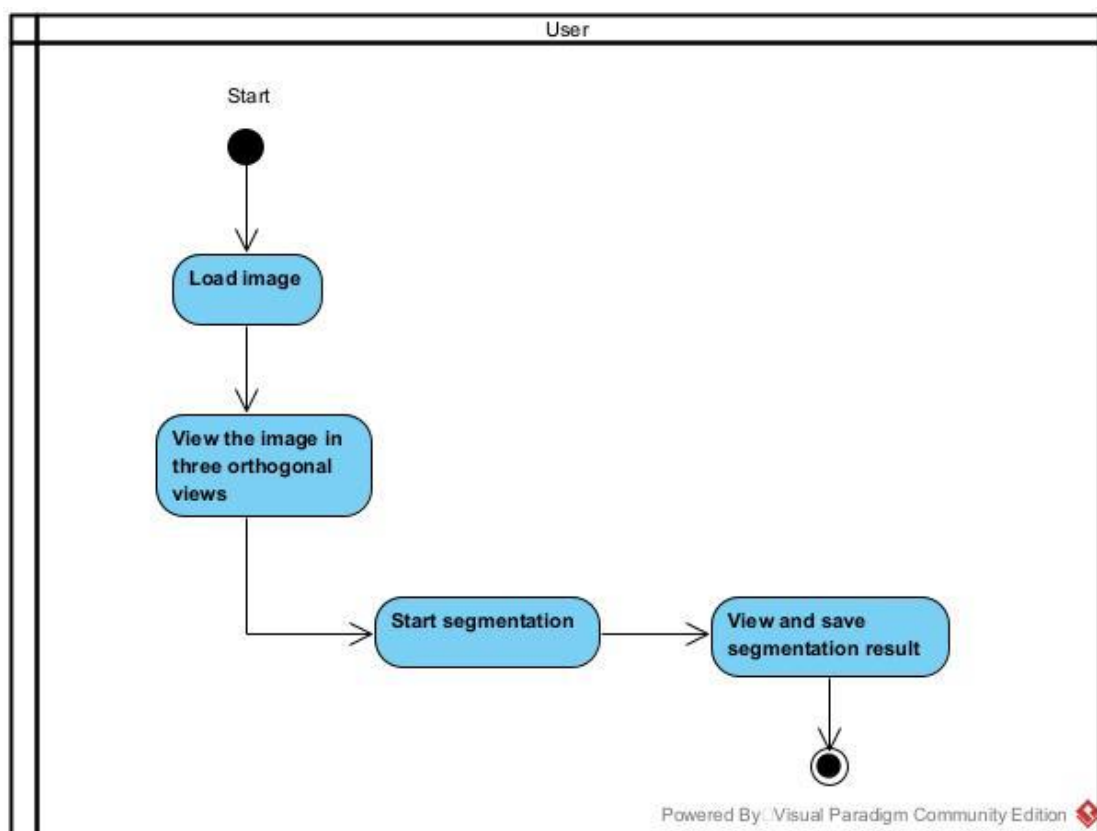


Figure 3.2. Activity Diagram

3.2.3 Wireframe

The wireframe of the proposed system is shown in Figure 3.3. The user will first load the image through the file tab. After the user loaded the image, the 3D image in three different plane and using the slicer to interact with the slice of the image. The segmentation button will start the segmentation process where with pretrained model, U-Net and Boundary Preserving Block (BPB). The segmented result will display in the main screen and the image is auto saved in the directory.

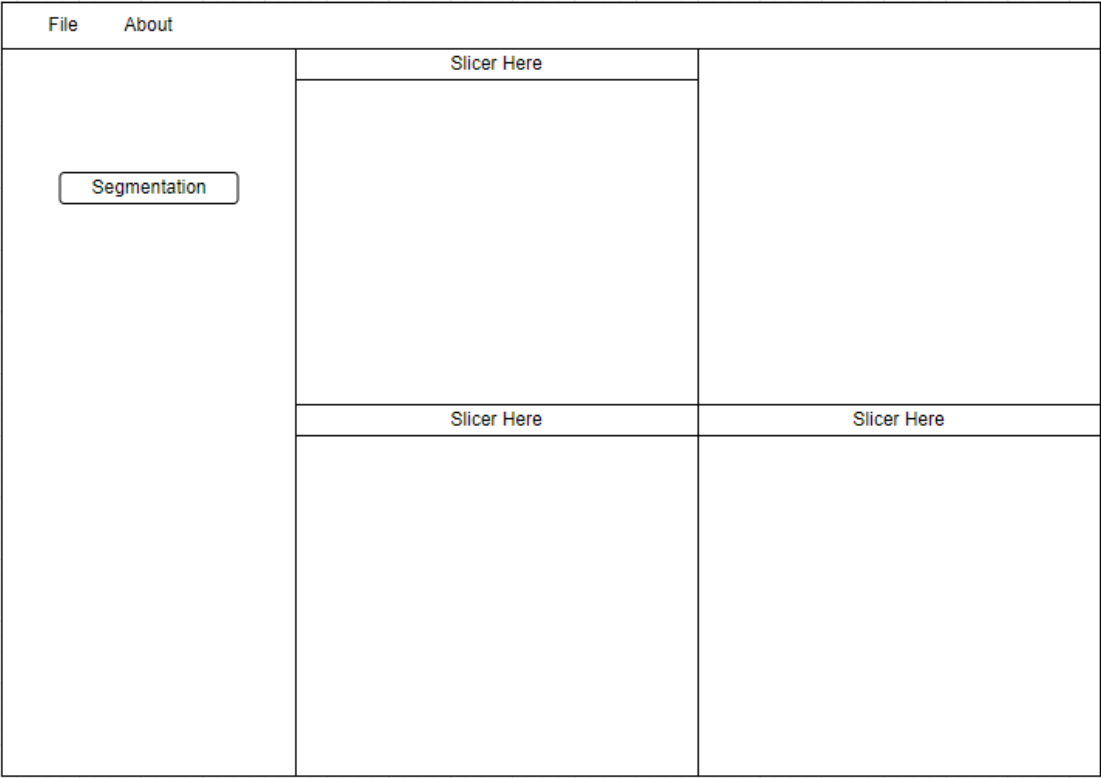


Figure 3.3. Wireframe of the proposed system.

3.2.4 Flowchart of system architecture

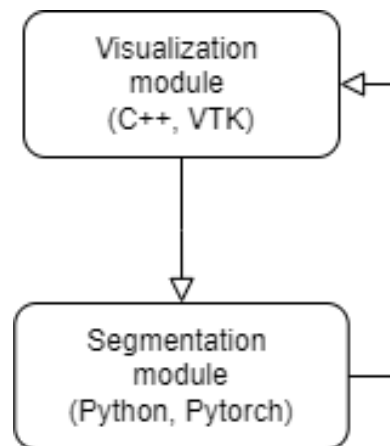


Figure 3.4. Flowchart of system architecture.

The proposed system is developed in two programming language and framework. The visualization module is developed in C++ using VTK framework. This module handles the visualization of 3D brain tumor image in three different orthogonal view. In segmentation module, it will receive the instruction from the visualization module, where it provides the information of image to segment. The segmentation module is developed in python using pytorch framework where the module is using deep learning model, U-net and Boundary Preserving Block (BPB) to perform segmentation task. After the segmentation work done, the module will then save the image into the predefined file path and the result will be display in visualization module.

3.3 Chapter Conclusion

In conclusion, this chapter show the overview design of the proposed system in the form of use case diagram, activity diagram, wireframe and flowchart of system architecture. The proposed system is mainly composed of two modules that perform the task of visualization and segmentation. The proposed system provides functionality for the user to perform segmentation of brain tumor in an easy way.

Chapter 4: System Implementation

4.1 Overview

This section will discuss the details of system implementation. The following section will be divided into six main sections: Section 4.2 System Development Methodology, Section 4.3 Hardware Setup, Section 4.4 Software Setup, Section 4.5 Brain Tumor Dataset, Section 4.6 Algorithm used, Section 4.7 Performance Metrics, 4.8 Project Challenges and Timeline and 4.9 Chapter Summary.

4.2 System Development Methodology

The project is developed using waterfall model [17] as the methodology. Waterfall model is a project management model where the system development process is separated into different phases: Requirements, Analysis, Design, Implementation, Verification and Maintenance. The phase of the process is in a linear and sequential manner as shown in Figure 4.1. In the requirement and analysis phases, users' requirements are gathered and analyzed. While, in the design phase, the users' requirements that were gathered and analyzed in the previous phases are studied and the system design is prepared. The system design will then be implemented in the implementation phase, where the system development is performed. Later in the verification phase, testing is performed on the developed system for debugging. In addition to that, in this phase, users will begin to review the system developed to make sure it meets their requirements. Lastly, the system will go into the maintenance phase, where the production team will continue applied patches to fix bug and enhance the product until the user is satisfied.

This project will go through the first five phases of the waterfall model which are requirement, analysis, design, implementation, and verification. In this project, there is no maintenance phase as the proposed system is only at the prototype level. In the requirement and analysis phases, this project will gather all the possible requirement of the system and analyze the existing software to gain insight on the possible outcome of the system. This includes studying how 3D CT/MRI image works, study how existing

software performs segmentation and getting the test dataset images for this project from The Medical Image Computing and Computer Assisted Intervention Society [18].

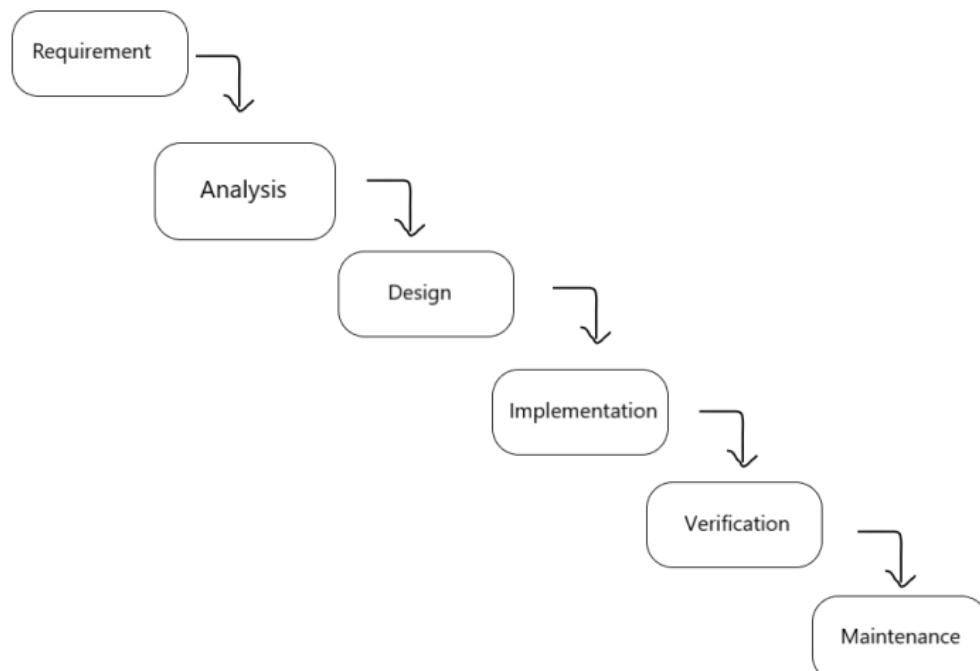


Figure 4.1. Waterfall model

Next, the design phase will further analyze the insight gain from the previous phases to design the overall architecture of the system. In this phase, the architecture of the system will be further broken down into specific parts and various technical problem such as Graphical User Interface (GUI) and wireframe design, the implementation details of U-Net and BPB that will be used to perform the segmentation, along with the programming language and libraries to be used are identified. Further in the implementation phase, the system will be developed according to the prototype design that has been developed in the design phase. The system will be developed into different small parts called units, which later will be integrated. Each unit developed will be tested for its functionality called Unit Testing. Lastly, in the verification phase, all the unit will be integrated and test as a fully working application, which is referred as Integration Testing. In this stage, the system will be tested to see whether it meets the software requirement that is defined during the requirement and analysis phases.

4.3 Hardware Setup

The proposed system is developed with the following hardware specification,

- Intel Quad Core CPU, i3-10105F
- NVIDIA GTX 1650
- 32 GB DDR4 RAM
- 1 TB Solid State Drive

Due to the high demand of GPU usage in various cloud service provider, the deep learning model training is done in local computer using CPU only. This is also due to the model training require powerful GPU with at least 28 GB VRAM, hence the GPU in local computer, GTX 1650 is not powerful enough to support the model training.

4.4 Software Setup

The software setup of the proposed system can be classified into two module, visualization and segmentation module. The software and library used to develop visualization module will be discussed first, then followed by the segmentation module.

4.4.1 Visualization Module

4.4.1.1 C++

C++ is a cross platform programming language created by Bjarne Stroustrup, as an extension to C language. It is a powerful general-purpose programming language that can be used to develop operating system, games, browser, etc.

4.4.1.2 Qt Framework

Qt is a free and open-source C++ based Graphical User Interface (GUI) framework. It supports the application to run in multiple platform such as Windows, Linux, and MacOS as well as mobile or embedded platform.

CHAPTER 4

4.4.1.3 VTK Library

VTK, or The Visualization Toolkit is an open-source library that allow visualization of 2D/3D computer graphics. It comes with a set of tools for rendering, interaction and plotting, the user able to control every single detail of rendering object that show in the screen. This made the library very powerful for commercial, and research and development purpose.

4.4.1.4 Visual Studio Code

Visual Studio Code is a free code editor that developed by Microsoft. It is able to run on Windows, Linux and MacOS. It is a powerful software that support lot of the features such as debugging, syntax highlighting, etc.

4.4.1.5 CMake

CMake is a tool used to control the compilation process with compiler independent method for C++. It generated the configuration file that can be used by various compiler and allow the source code can be compiled smoothly in any build environments.

4.4.1.6 Qt Creator

Qt Creator is a Qt Framework integrated development environment that allows the user to develop Qt application without the hassle of project configuration. It comes with a set of tools such as C++ compiler, Qt Designer, and code editor to speed up the development of Qt application.

4.4.2 Segmentation Module

4.4.2.1 Python

Python is a high level, general-purpose programming language that comes with powerful ecosystem of libraries and packages make it one of the most popular programming languages to be used. Python is easy to use and have been used in different settings such as software development, artificial intelligence, data science, automation, and etc.

4.4.2.2 PyTorch

PyTorch is one of the most popular open-source machine learning frameworks that mainly developed by Facebook's AI Research lab (FAIR). It is being used for artificial intelligence related work such as computer vision and natural language processing,

4.4.2.3 PyCharm

PyCharm is python Integrated Development Environment (IDE) developed by JetBrains. It comes with various features that make development in python easy and convenience and allows the developer to focus on development work without distracted by the environment setup issue.

4.4.2.4 Anaconda

PyCharm is one of the world's most popular machine learning and data science platform. It comes with thousands of packages and libraries that make data science and machine learning task to be carried out easily.

4.5 Brain Tumor Dataset

The dataset that being used in this project is BraTS 2015 [19] which are taken form The SICAS Medical Image Repository [20]. The dataset consists of 276 patients in the training dataset that divided into two categories according to the tumor types, which are Low Grade Glioma (LGG) and High-Grade Glioma (HGG). Each patient come with 4 modality of the image which are Flair, T1, T1c and T2, the dataset also come with ground truth images which can later be compared with the segmentation result. The image in this dataset is in MetaImage (mha) file format where makes the dataset having 1104 three-dimensional images in total excluding the ground truth.

The BraTS 2015 testing dataset will not be included in this project as the dataset does not provide ground truth images, The experiment for segmentation is conducted on 30 LGG images and 3 HGG images of different patient with intensity feature. Some of the

example image in the dataset is shown in Figure 4.2. All the image used in this project is Flair modality.

In this project, due to lack of computation power and resources, only 30 images in the dataset are selected to be used to train the deep learning model. The first 15 images in HGG and first 15 images in LGG is selected to make up the training set. The testing set used in this project contain of 10 images where it is randomly selected from the remaining images that is not included in the training set.

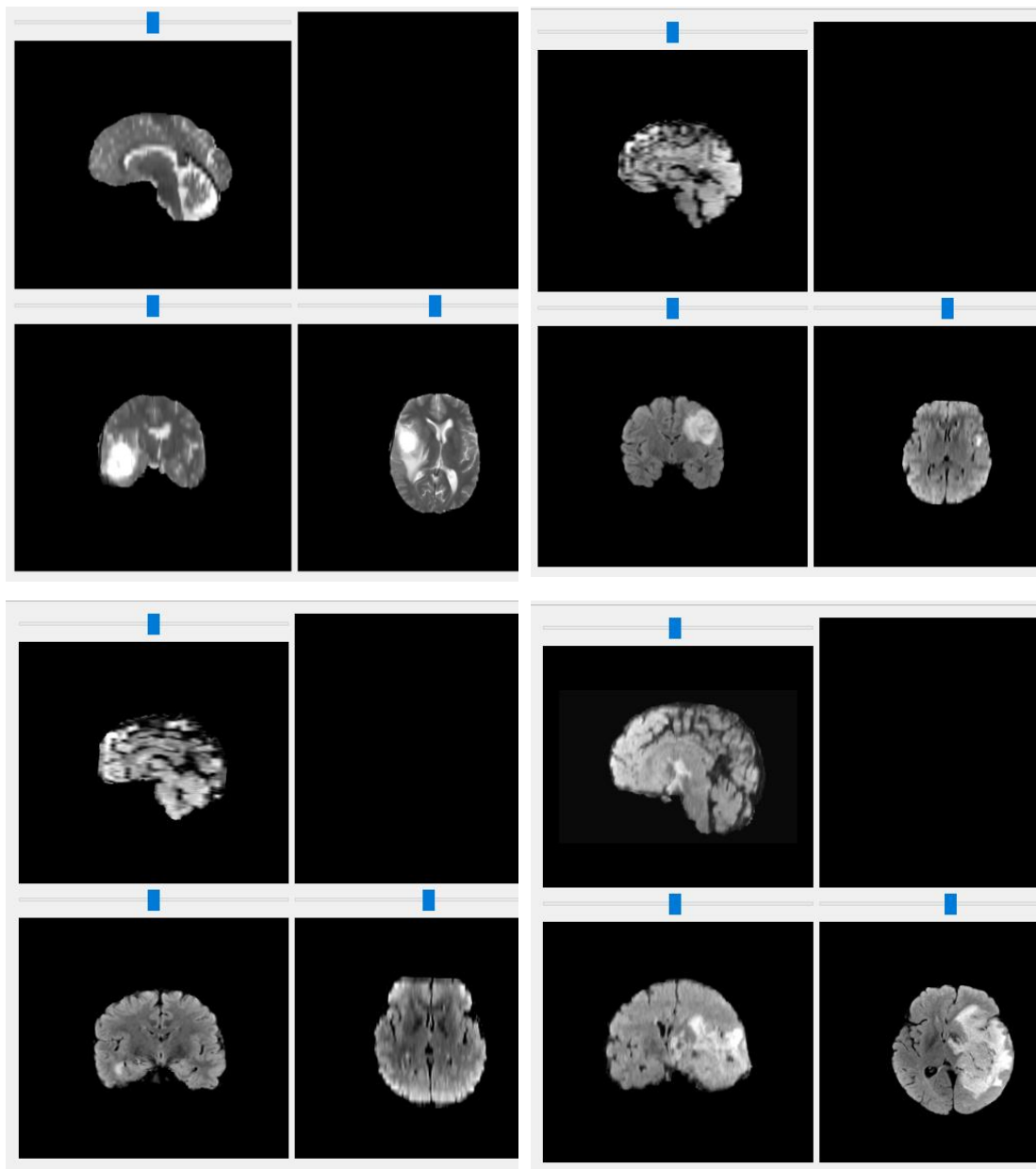


Figure 4.2. Example images of BraTS 2015 training dataset.

4.6 Algorithm used

This section discusses the algorithm used to implement the segmentation module. The main part of the segmentation module consists of two important algorithm which are U-Net [21] and Boundary Preserving Block (BPB) [22].

4.6.1 U-Net

U-Net [21] is considered one of the most commonly used deep learning model for semantic segmentation task. The model is first proposed in 2015 that focus on biomedical image segmentation by using convolutional network. The model consists of two main parts, which are contracting path (left side) and expansive path (right side) as shown as Figure 4.3. The contracting path will extract the low-level feature of the images using the down sample convolution block. Each down sample convolution block consists of two 3x3 convolution layer, a rectified linear unit (ReLU) and followed by a 2x2 max pool layer. The number of feature channel will be doubled after each down sample convolution block and will go through the down sample convolution block 4 times. At the expansive path, up-sample convolution block that consists of one 2x2 up convolution, two 3x3 convolution layer and a ReLU. After the feature maps go through 2x2 up convolution layer in each up-sample convolution block, it will be concatenate with the corresponding feature maps in contracting path before feed to the 3x3 convolution layer. A 1x1 convolution layer is used in the final layer to map the feature maps into the number of classes.

In this project, the number of classes in the final layer is one as it performs binary segmentation task. Furthermore, the U-Net implementation has been modified to the 3D U-Net [23] as shown as Figure 4.4 to suit the project's need. The structure of the model is same as 2D U-Net with the different where the convolution layer has become 3D convolution layer. For example, the 3x3 convolution layer in the 2D U-net will become 3x3x3 convolution layer. Due to the nature of 3D images, this has made the model become big enough that required the computation instance to have high RAM (more than 28 GB) to train the model with CPU and required the GPU to have high VRAM (more than 28 GB) if used for accelerate training in this project.

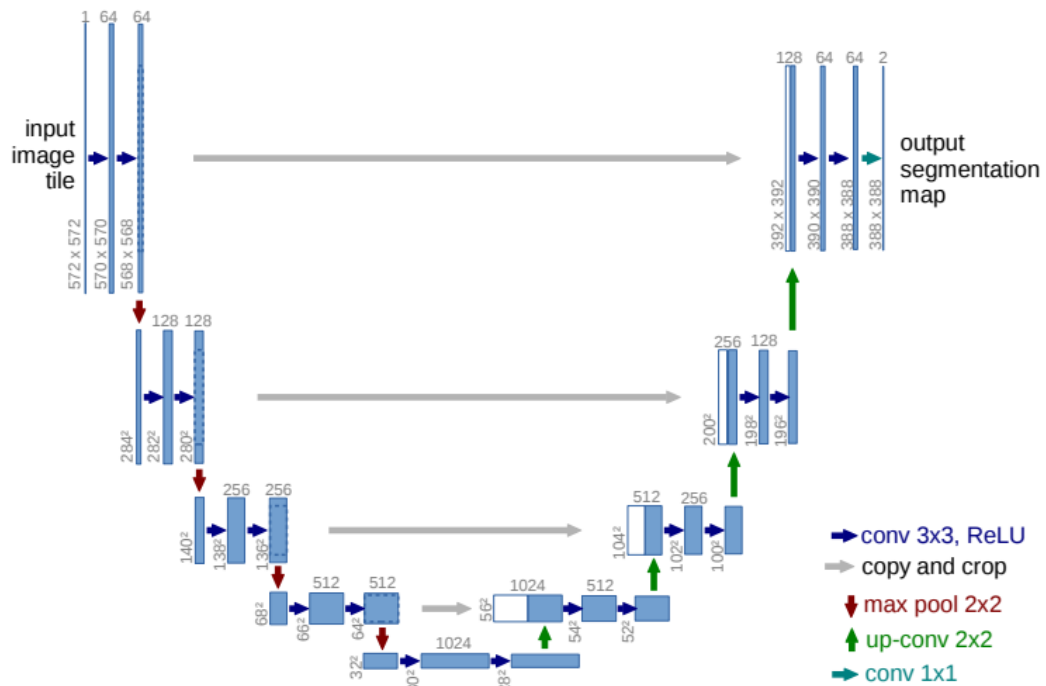


Figure 4.3. 2D U-Net architecture [21]

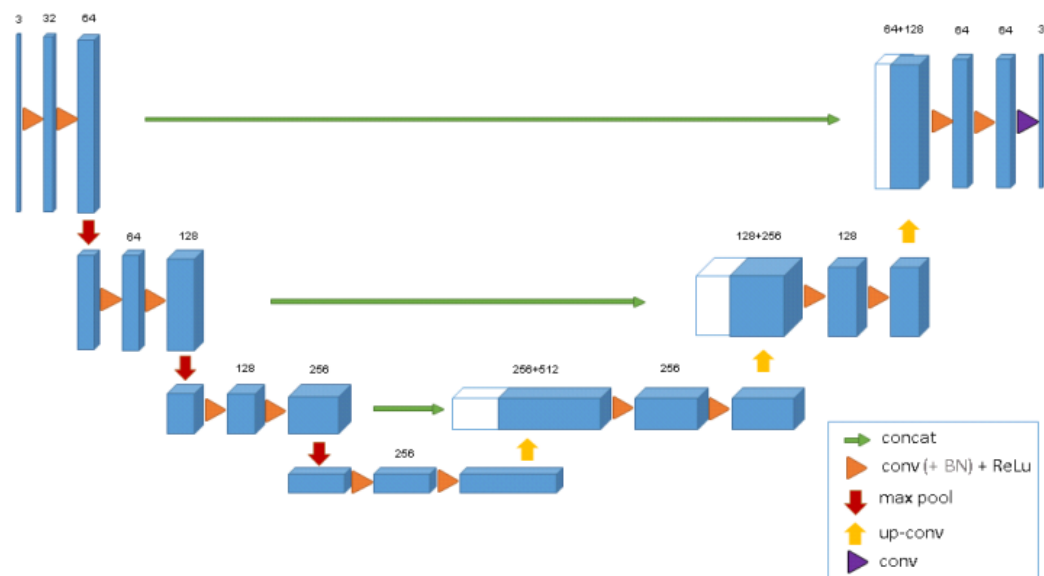


Figure 4.4. 3D U-Net architecture [23]

4.6.2 Boundary Preserving Block (BPB)

Boundary Preserving Block (BPB) [22] is one of the image segmentation methods that aim to preserving boundary of the target region to tackle the issue of ambiguous boundary in medical image. This method is first proposed in 2020 together with Shape

Boundary-aware Evaluator (SBE) [22]. However, this project only focuses in BPB and this method does not have any open-source implementation yet until the time of writing. BPB is included in the segmentation network to predicts the boundary of target region as shown as Figure 4.5. The formula for the BPB can be expressed as

$$v^i = f^i \oplus (f^i \otimes \hat{M}^i) \quad (4.1)$$

where v^i , f^i and \hat{M}^i denotes as output, input, key point map generated by boundary point map generator, while \oplus and \otimes denotes as element-wise addition and channel-wise multiplication. Boundary point map generator will generate the predicted boundary key point map using dilated convolution [24]. Dilated convolution is express as $d_r^s(\cdot)$, where s is the filter size $s \times s$ and r represent the dilation rate. The formula of boundary point map generator can be written as

$$\hat{M}^i = \sigma(d_1^1 \{ [d_1^1(f^i), d_1^3(f^i), d_2^3(f^i), d_4^3(f^i), d_6^3(f^i)] \}) \quad (4.2)$$

where σ denotes sigmoid function. In this project, the BPB has been modified to 3D version in each operation to suit the project's need. Furthermore, this project placed 3 BPB inside U-Net between the convolution layer, each BPB place after 2 convolution blocks.

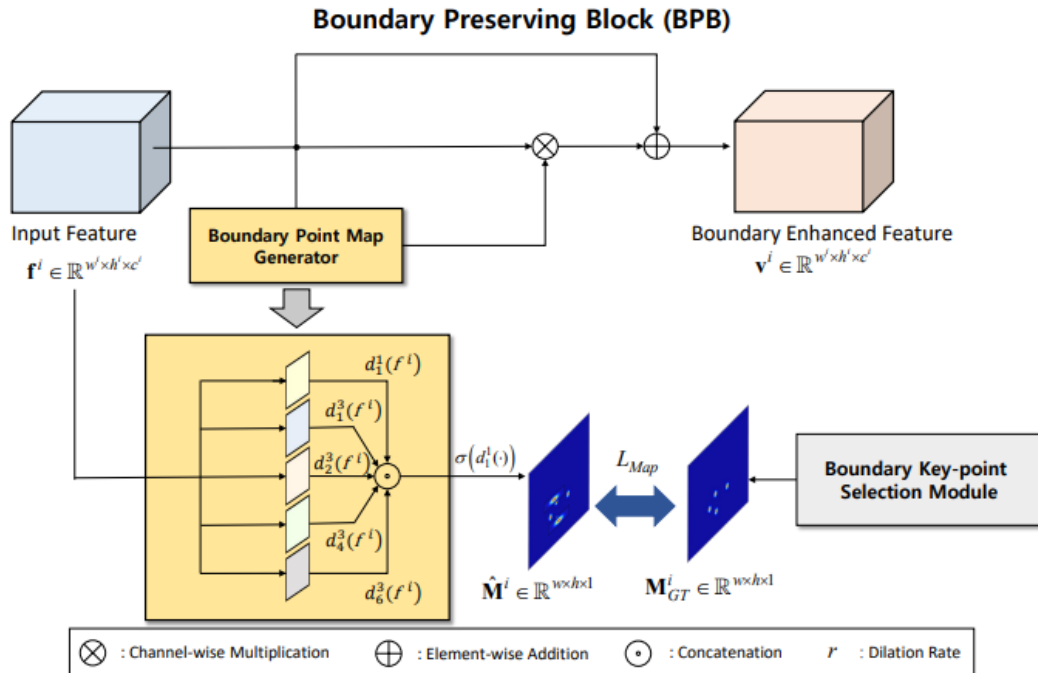


Figure 4.5 Detail structure of Boundary Preserving Block [22]

4.7 Performance Metrics

This section discusses the performance metrics used to evaluate the accuracy of the segmentation result in comparison to the ground truth. The performance metrics that are chosen to be used in this work are Intersection over union (IOU) [25] and Dice similarity coefficient (DSC) [26]. IOU is one of the most commonly used metrics to evaluate the performance of a segmentation result [27]. IOU is computed by taking two arbitrary shapes and calculate the ratio of overlapping region over the union region as shown in equation 4.3,

$$\text{IOU} = \frac{|A \cap B|}{|A \cup B|} \quad (4.3)$$

A is the segmented image and B is the ground truth image. The larger the value of IOU, the better is the segmentation result, as the segmented region has perfectly fit the ground truth region. While DSC or Dice coefficient [26] is another popular performance metric that has been used in measuring a segmentation result. This metric is commonly used to measure the accuracy of the segmentation result in comparison to the ground truth [28]. The DSC is as defined as in equation 4.4,

$$\text{DSC} = \frac{2|A \cap B|}{|A| + |B|} \quad (4.4)$$

A is the segmented image and B is the ground truth image. The larger the value of DSC, the better is the segmentation result. IOU tends to penalize the errors in smaller segments than DSC to measure the accuracy of the segmentation. The differences between these two-evaluation metrics are seen when the segmentation accuracy is evaluated on a large dataset.

4.8 Project Challenges and Timeline

4.8.1 Implementation issues and challenges

There are several issues and challenges have been faced during the implementation of this project. One of the biggest challenges has been encountered is lack of GPU to train the deep learning model. As training the 3D deep learning model would require a lot of time, GPU is one of the common methods used to speed up the training process. However, even after spending large amount of time contacting with university laboratory, cloud service provider included Microsoft, Google and Amazon, there is still no powerful GPU available to been used in the training process. Furthermore, Google Colab, one of the free machine learning platforms, does not have GPU that is powerful enough to enable the model training in this project. At the end, the model training is done on CPU with reduced version of training set. Although the size of training size has been greatly reduced from 220 images to 30 images, the training process still take up to 1 whole day to complete 10 epochs. Including the experiment time, the whole training process in this project takes up to 5 whole days to complete.

Furthermore, the implementation of BPB is built from scratch, as it is one of rather new technique, so there is no open-source implementation that can be referred to. Due to this reason, a lot of effort have been put in to learn and implement the BPB from the research paper [22].

Next, the incompleteness of documentation of the VTK library is also one of the challenges faces in this project. The documentation of the library is outdated during the development of this project, so extra effort was required in order to search for the updated configuration and reference at other unofficial sources and through trial and error. This has affected the progress of development in visualization of the 3D images and consumed large amount of time.

Lastly, the challenge that has been encountered is the retrieval of datasets. The effort has been made to request the datasets from MICCAI. However, the MICCAI organization did not respond after several email had been sent, this lead to retrieve datasets from other research sources. After a long time of research, the most appropriate dataset finally had been found by applying the access to BraTS 2015 dataset [19] from The SICAS Medical Image Repository [20].

4.8.2 Timeline

Table 4.1 Project I timeline.

Task	Start	Due	Duration (Days)
1. Research and literature review	1 June 2021	13 June 2021	13
2. Requirement analysis	7 June 2021	13 June 2021	7
3. System Design	14 June 2021	20 June 2021	7
4. Implementation I: Visualization	21 June 2021	18 July 2021	28
5. Implementation II: Segmentation	19 July 2021	13 August 2021	26
6. Result Analysis and Experiment	14 August 2021	19 August 2021	6
7. Report Writing	20 August 2021	27 August 2021	8
8. Presentation	28 August 2021	9 September 2021	12

The Table 4.1 shows the project timeline table in FYP1. This semester is a long semester with 14 weeks, where the project has been started before the semester start. Before the semester start, some research was conducted to get the basics knowledge in the field of image processing and set up the development environment for the implementation phase. During the first two weeks of the semester, the system requirement and design has been completed. The implementation phase takes the longest time to apply the concept learned into the prototype. The experiment, report writing, and presentation will be conducted after implementation phase.

Table 4.2 Project II timeline

Task	Start	Due	Duration (Days)
1. Implement U-Net	24 January 2022	11 February 2022	14
2. Implement BPB	12 February 2022	25 February 2022	14
3. Contact and setup university server to use the GPU	26 February 2022	11 March 2022	14
4. Setup training environment in Azure, Google and Amazon (Request for quota increase and waiting for the response)	12 March 2022	5 April 2022	25
5. Move to reduced training set, train in CPU and perform experiment	6 April 2022	15 April 2022	10
6. Report Writing	16 April 2022	22 April 2022	7
8. Presentation	23 April 2022	28 April 2022	6

The Table 4.2 shows the project timeline table in FYP2. This semester is a long semester with 14 weeks, where the project has been continuing from where the FYP1 end. Before the semester start, some research was conducted to look for the alternative method that can perform segmentation, deep learning with U-Net was decided at the end. During the first four weeks of the semester, the implementation of U-Net and BPB has been done. However, the process of trying to find available GPU that can allow the model training has been last for more than one month. Due to lack of GPU, the model training is done in CPU in the end.

4.9 Chapter Summary

This chapter discussed the implementation details of the system where the visualization module is developed in C++, Qt and VTK, while the segmentation module is developed with Python and PyTorch. The dataset used in this project is BraTS 2015 with 30 images in training set and 10 images in testing set. The deep learning model used for segmentation in this project is U-Net and BPB. Furthermore, IOU and DSC score are used to evaluate the performance of the segmentation result. In this project, due to lack of GPU, the model training is not able to be trained well.

Chapter 5: System Evaluation and Discussion

5.1 3D brain tumor images visualization

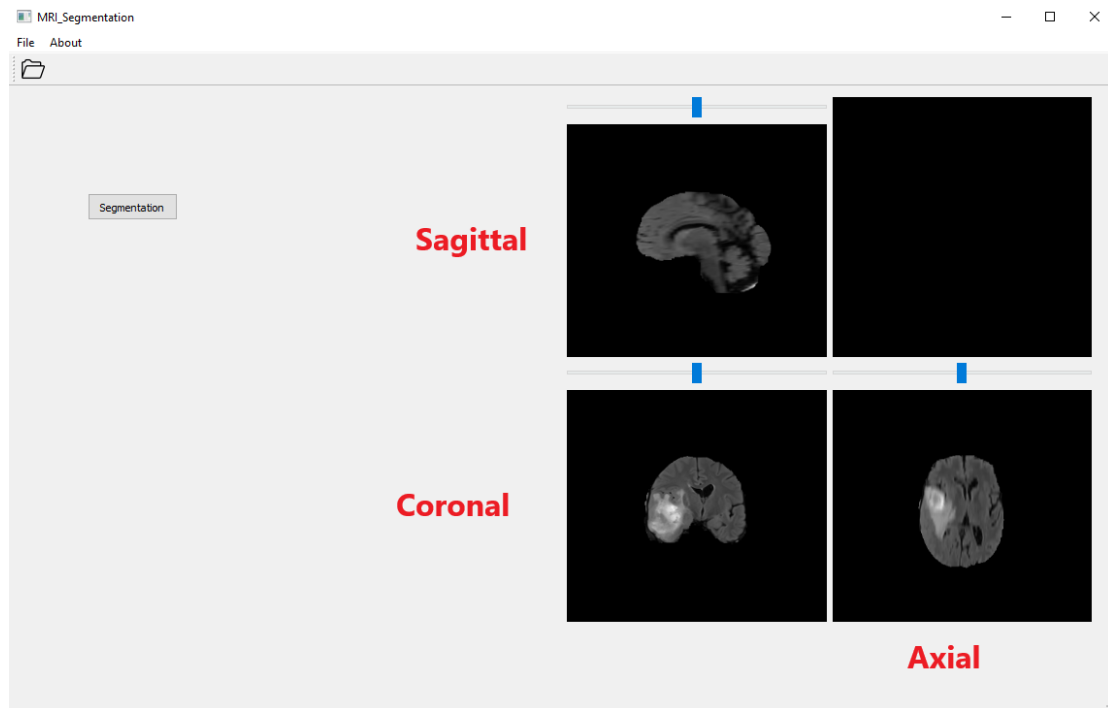


Figure 5.1. 3D brain tumor image visualization interface of the proposed system. The render window in the top left shows the sagittal plane of the 3D image, the bottom left, and bottom right show the coronal and axial plane respectively. The segmentation button will perform segmentation and save the segmentation result.

Figure 5.1 shows the 3D medical image visualization of the proposed system. The GUI shows the 3D MR brain tumor image of a patient in three different orthogonal views which are axial, coronal, and sagittal plane in a single window. This graphical user interface (GUI) is developed in Microsoft Visual Studio with Qt and VTK framework. The slider is provided in each plane to control the brain tumor image slices in horizontal, vertical, and longitudinal directions. Axial represents the horizontal while coronal shows the vertical and sagittal gives the longitudinal 3D brain tumors views, respectively.

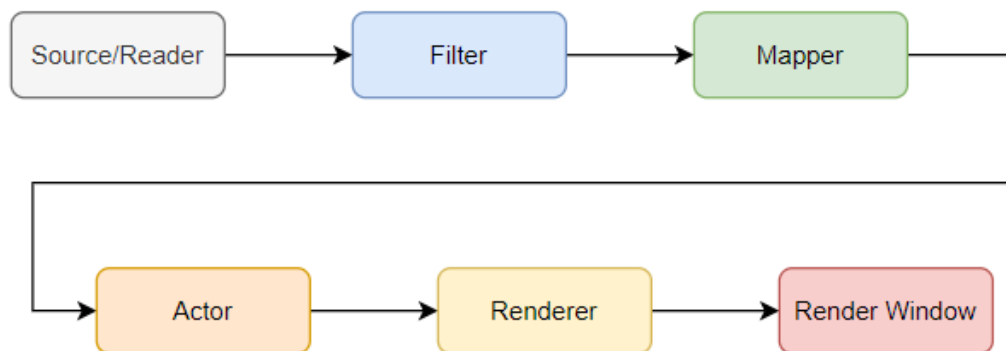


Figure 5.2. VTK visualization pipeline

The visualization process takes through a series of process as per required by the VTK. The VTK visualization pipeline can be divided into 6 phrases as shown as Figure 5.2. The application will first read the image data from the source and connect it to a mapper, then the mapper will map the source to an actor. The filter is optional depends on the task, filter will act as processing to the image and can be used in image segmentation or other image operation. Next, the actor will represent the image object in the renderer, the renderer will control the rendering processing for different actors such as the camera view and the lighting. Eventually, the render window will create a window to allow the renderer to visualize the object and display it to the user. The partial code that shows the VTK visualization pipeline is shown in Figure 5.3. The code loads the image file that selected by the user and displays it to the user through the VTK visualization pipeline.

```

reslice[i] = vtkSmartPointer<vtkImageReslice>::New();
reslice[i]->SetInputConnection(reader->GetOutputPort());
reslice[i]->SetOutputDimensionality(2);
reslice[i]->SetResliceAxes(resliceAxes[i]);
reslice[i]->SetInterpolationModeToLinear();

colorTable[i] = vtkSmartPointer<vtkLookupTable>::New();
colorTable[i]->SetRange(0, 1000);
colorTable[i]->SetValueRange(0.0, 1.0);
colorTable[i]->SetSaturationRange(0.0, 0.0);
colorTable[i]->SetRampToLinear();
colorTable[i]->Build();

colorMap[i] = vtkSmartPointer<vtkImageMapToColors>::New();
colorMap[i]->SetLookupTable(colorTable[i]);
colorMap[i]->SetInputConnection(reslice[i]->GetOutputPort());
colorMap[i]->Update();

imgActor[i] = vtkSmartPointer<vtkImageActor>::New();
imgActor[i]->SetInputData(colorMap[i]->GetOutput());

renderer[i] = vtkSmartPointer<vtkRenderer>::New();
renderer[i]->AddActor(imgActor[i]);
//renderer[i]->SetBackground(255.0, 1.0, 1.0);
renderWindow[i] = vtkSmartPointer<vtkGenericOpenGLRenderWindow>::New();
renderWindow[i]->AddRenderer(renderer[i]);
renderWindow[i]->Render();

```

Figure 5.3. Partial code of image visualization. The visualization pipeline of VTK can be seen in the Figure where the after the image is loaded into the system, the mapper will map the image reader to the actor. The renderer will then map the actor and display it in the render window.

5.2 Segmentation Result

The result of segmentation is compared to the ground truth images to measure the accuracy by using IOU [25] and Dice (DSC) coefficient [26]. The model has been done with 10 epochs for both U-Net and U-Net with BPB. As several experiment has been conducted, the following parameter has been found to give the best result in segmentation, training in U-Net used SGD optimizer with 0.00001 learning rate, while training in U-Net with BPB used Adam optimizer with 0.0001 learning rate. During the training stage, the best segmentation result in U-Net model is in epoch 8 with 0.09 IOU. The best segmentation result in U-Net with BPB model is in epoch 6 with 0.04 IOU. In testing set, the average IOU and DSC score of U-Net is 0.06 and 0.10 respectively, while, the average IOU and DSC score of U-Net with BPB is 0.02 and 0.04 respectively.

Some of the segmentation examples of testing set with brats_tcia_pat319_0001 and brats_tcia_pat343_0001 is show in Figure 5.4 and Figure 5.5 respectively.

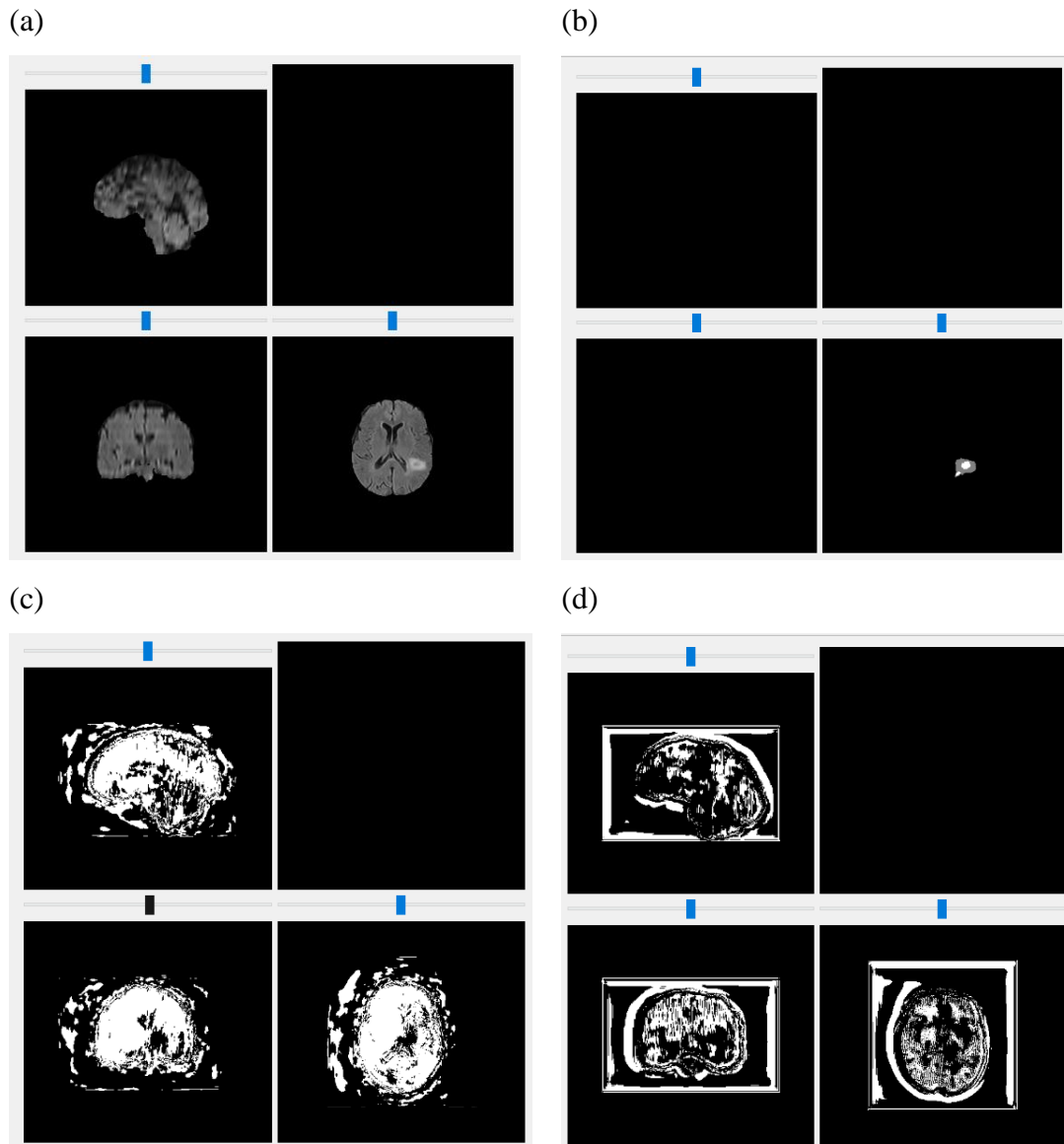


Figure 5.4. The segmentation result with brats_tcia_pat319_0001 in testing set compared between original image, ground truth image, segmentation result (U-Net), segmentation result (U-Net with BPB) in (a), (b), (c) and (d) respectively.

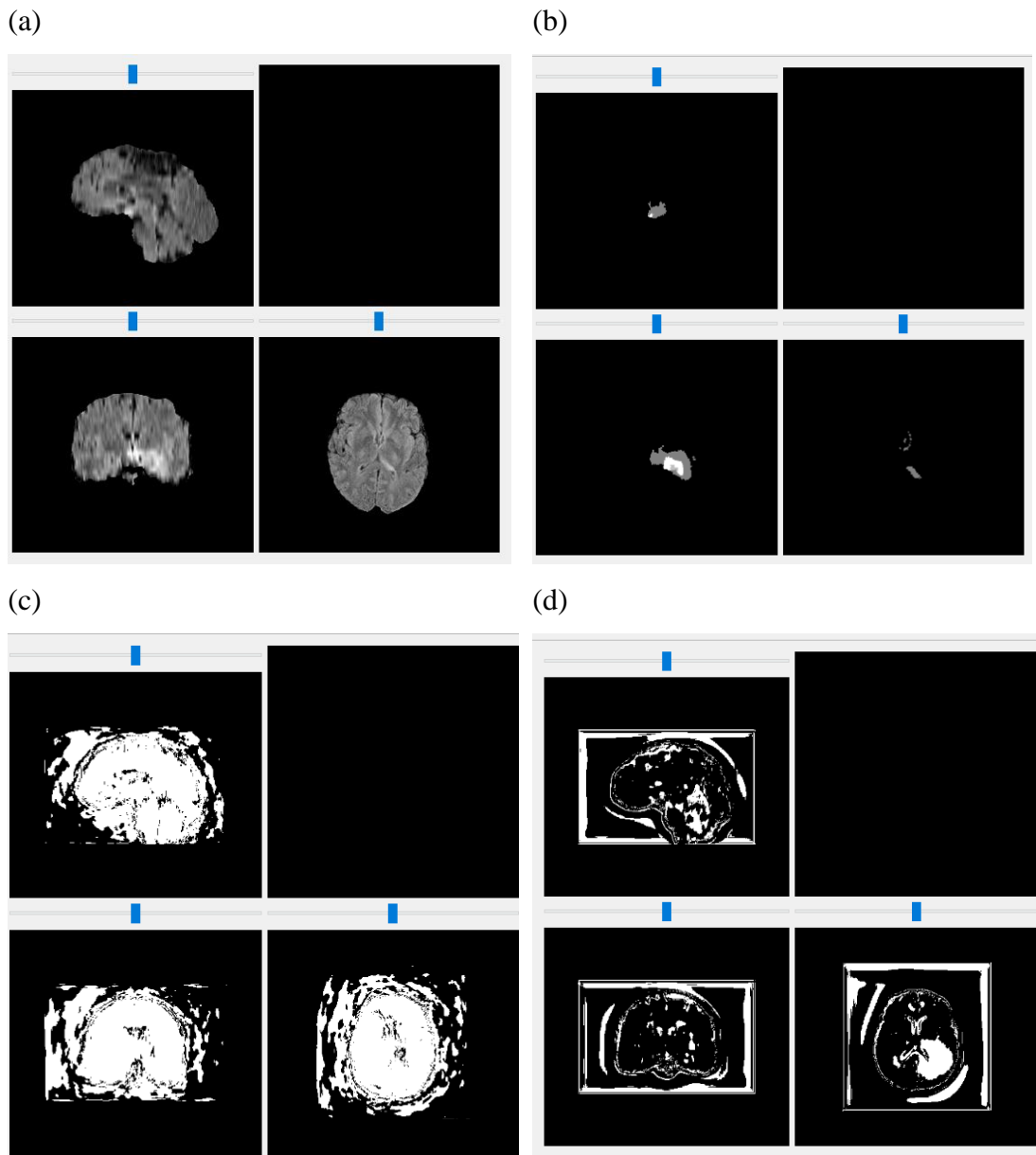


Figure 5.5. The segmentation result with brats_tcia_pat343_0001 in testing set compared between original image, ground truth image, segmentation result (U-Net), segmentation result (U-Net with BPB) in (a), (b), (c) and (d) respectively.

Due to lack of GPU, this has caused insufficient time and dataset used for training, with only 30 images and 10 epochs is not enough to train the model to produce good result. From the Figure 5.4 and 5.5, the model does not recognize and not able to segment the target region correctly. Furthermore, using BPB alone might also not able to produce good segmentation result as the method is first proposed together with SBE.

5.3 Chapter Conclusion

This chapter has discussed the visualization and segmentation module in this project. The visualization module is able to visualize the 3D brain tumor images in three different orthogonal views on a single graphical user interface (GUI) screen and a slider is provided in each orthogonal views to allow the user to interact with the slice of the image. In segmentation module, U-Net gives 0.06 IOU and 0.10 DSC in testing set, while U-Net with BPB gives 0.02 IOU and 0.04 DSC score in testing set.

Chapter 6: Conclusion and Recommendation

6.1 Conclusion & Novelty

In conclusion, this project delivered a 3D brain tumor image segmentation software using deep learning model, U-Net and BPB. In the existing software, the segmentation result causing over/under segmentation for the brain tumor image with ambiguous boundaries. By developing the proposed software, it aims to provide a simple to use segmentation software to the user for fully automated brain tumor segmentation.

The proposed software is developed using Microsoft Visual Studio with Qt and VTK framework and able to visualize the 3D brain tumor image in three different orthogonal views which are axial, coronal, and sagittal plane on a single graphical user interface (GUI) screen. Moreover, in this project, deep learning model, U-Net and BPB has been selected to evaluate the performance of the segmentation. The BraTS 2015 training dataset [21] with 30 images with flair modality has been selected as training set and 10 images randomly selected has been used as testing set. The performance metrics that have been used to evaluate the result in this experiment are IOU and DSC score. From the experiment results, it is shown that the average IOU and DSC score in testing set of U-Net is 0.06 and 0.1 respectively, while the average IOU and DSC score in testing set of U-Net with BPB is 0.02 and 0.04 respectively.

6.2 Future work

The greatest problem in this project is the accuracy of the segmentation result, the trained model in this project is not able to produce good segmentation result. This is because lack of GPU to be used in the training process, train the model with more epoch and larger dataset is necessary to achieve better result. Moreover, the original paper of BPB [22] was proposed BPB together with SBE, implement BPB and SBE in the model at the same time might get better segmentation result.

References

- [1] Cancer.Net, "Doctor-Approved Patient information from ASCO," 2020. [Online]. Available: <https://www.cancer.net>.
- [2] E. Ilunga–Mbuyamba, Juan Gabriel Avina–Cervantes, Jonathan Cepeda–Negrete, Mario Alberto Ibarra–Manzano and Claire Chalopin, "Automatic selection of localized region-based active contour models using," *Computers in Biology and Medicine*, vol. 91, pp. 69 - 79, 2017.
- [3] ITK-SNAP, "ITK-SNAP," 2018. [Online]. Available: <http://www.itksnap.org/>.
- [4] C. f. I. Technology, "Medical Image Processing, Analysis and Visualization," 2007. [Online]. Available: <https://mipav.cit.nih.gov>.
- [5] 3. Slicer, "3D Slicer," 2020. [Online]. Available: <https://www.slicer.org>.
- [6] O. Medical, "Olea Medical," 2021. [Online]. Available: <https://www.olea-medical.com/en>.
- [7] Paul A. Yushkevich, Yang Gao and Guido Gerig, "ITK-SNAP: An interactive tool for semi-automatic segmentation of multi-modality biomedical images," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, 2016.
- [8] 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*, vol. 31, no. 3, pp. 1116-1128, 2006.
- [9] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper and R. Kikinis, "3D Slicer as an image computing platform for the Quantitative Imaging Network," *Magnetic resonance imaging*, vol. 30, no. 9, p. 1323–1341, 2012.
- [10] R. Kikinis and S. Pieper, "3D Slicer as a tool for interactive brain tumor segmentation," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, 2011.
- [11] FDA, "U.S. Food and Drug Administration," 2020. [Online]. Available: <https://www.fda.gov/home>.

REFERENCES

- [12] J. C. Bezdek, R. Ehrlich and W. Full, "FCM: The fuzzy c-means clustering algorithm.," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191-203, 1984.
- [13] Y. Wang, H. Wang, K. Shen, J. Chang and J. Cui, "Brain CT image segmentation based on 3D slicer," *Journal of Complexity in Health Sciences*, vol. 3, no. 1, pp. 34 - 42, 2020.
- [14] U. D. o. V. Affairs, "Olea Shpere," 2019. [Online]. Available: <https://www.oit.va.gov/Services/TRM/ToolPage.aspx?tid=11574#>.
- [15] Vital, "Olea Tumor Basic," 2021. [Online]. Available: <https://www.vitalimages.com/wp-content/uploads/vpmc-13988a-olea-tumor-basic.pdf>.
- [16] A. Vanbinst, "Brain Tumor: Low-Grade Glioma," 2021. [Online]. Available: <https://www.olea-medical.com/wp-content/uploads/2021/10/Case-Report-019-Olea-Sphere-Brain-Tumor-Low-Grade-Glioma.pdf>.
- [17] W. Royce, "Managing the Development of Large Software Systems," *IEEE WESCON*, vol. 26, pp. 328-388, 1970.
- [18] MICCAI, "The Medical Image Computing and Computer Assisted Intervention Society," 2020. [Online]. Available: <http://www.miccai.org>.
- [19] B. H. Menze and et al., "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," *IEEE Trans Med Imaging*, vol. 34, no. 10, pp. 1993-2024, 2015.
- [20] Swiss Institute for Computer Assisted Surgery, "The SICAS Medical Image Repository," 2021. [Online]. Available: <https://www.smir.ch/>.
- [21] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015.
- [22] H. J. Lee, J. U. Kim, S. Lee, H. G. Kim and Y. M. Ro, "Structure Boundary Preserving Segmentation for Medical Image With Ambiguous Boundary," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [23] O. Cicek, A. Abdulkadir, S. S. Lienkamp, T. Brox and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, 2016.
- [24] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," in *International Conference on Learning Representations*, 2015.

REFERENCES

- [25] M. LEVANDOWSKY and D. WINTER , "Distance between Sets," *Nature*, vol. 234, pp. 34-35, 1971.
- [26] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology*, vol. 26, pp. 297-302, 1945.
- [27] R. Hamid , T. Nathan , J. Gwak, A. Sadeghian, I. Reid and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [28] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 29, 2015.
- [29] Microsoft, "Microsoft," 2020. [Online]. Available: <https://www.microsoft.com/en-my>.
- [30] T. G. Project, "The GNU Project," 2020. [Online]. Available: <https://www.gnu.org/gnu/thegnuproject.en.html>.

POSTER

U-NET AND BOUNDARY PRESERVING BLOCK LEARNING ALGORITHMS FOR 3D BRAIN TUMOR SEGMENTATION



INTRODUCTION

This project proposed a system that performs 3D brain tumor image segmentation using deep learning model, U-Net and Boundary Preserving Block (BPB)

OBJECTIVE

To developed application that is able to:

1. Visualize the 3D brain tumor image
2. Perform segmentation with deep learning model, U-Net and Boundary Preserving Block (BPB)

METHODOLOGY

Visualization is developed with C++, Qt and VTK.

The segmentation module has been done with:

- U-Net and BPB developed with Python and PyTorch
- Model train with BraTS 2015 training set 30 images, testing set 10 images.
- Model train with 10 epoch
- U-Net is using SGD optimizer, learning rate of 0.00001
- U-Net with BPB is using Adam optimizer, learning rate of 0.0001
- IOU and DSC score as performance metrics

PROBLEM STATEMENT

Poor segmentation result in ambiguous boundary image with automatic segmentation in existing software

RESULT (DSC)

Result with testing set

U-Net: 0.1

U-Net with BPB: 0.04

The model is not able produce good segmentation result due to lack of GPU in model training

