



University of Manouba  
National School of Computer Sciences



REPORT OF  
THE DESIGN AND DEVELOPMENT PROJECT

Subject

---

3D Tumor Segmentation and Reconstruction  
using Machine Learning

---

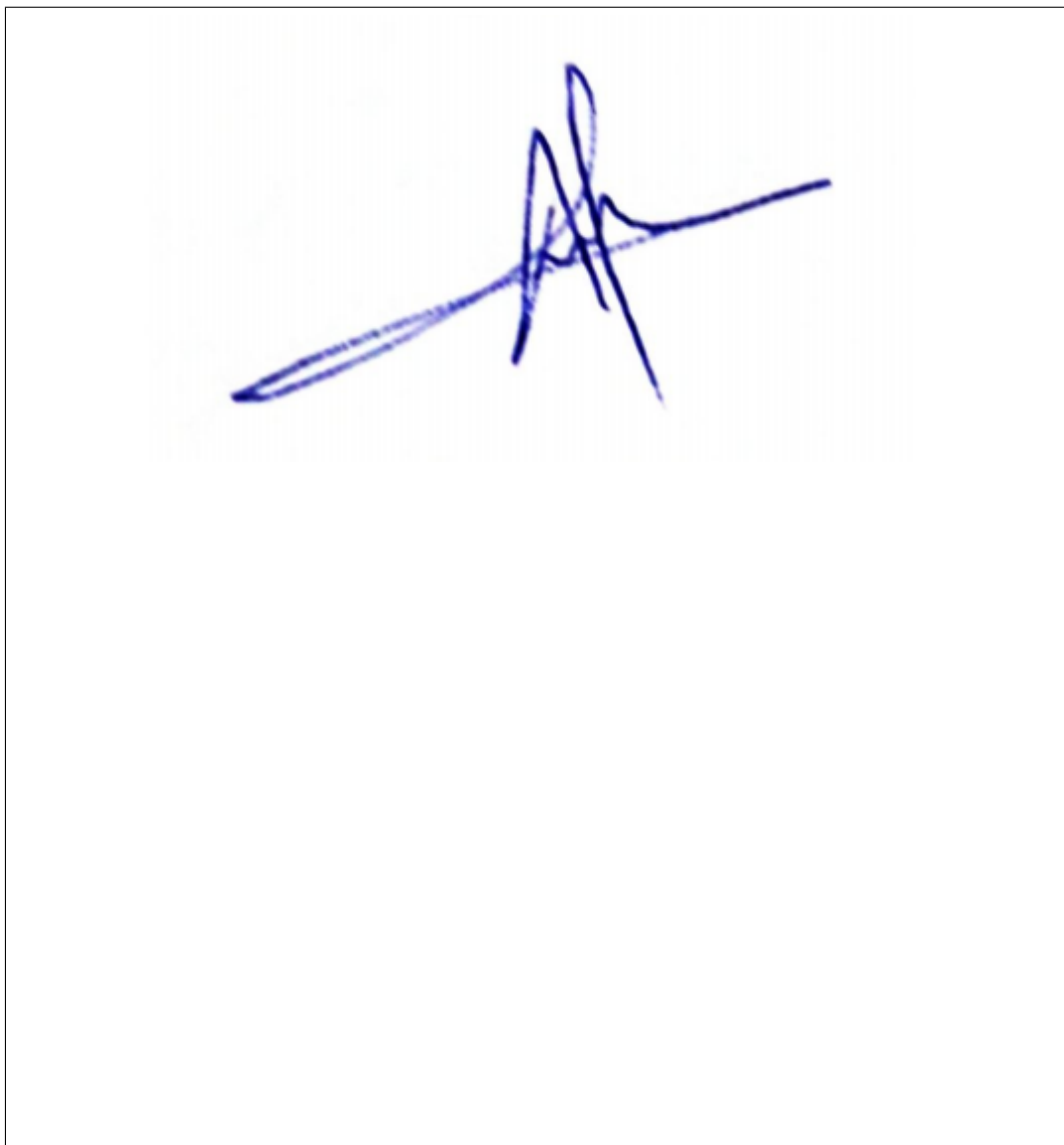
*Authors :*

Ahmed GHARBI      Hamza FRIGUI

*Supervisor :*

Dr. Mohamed Amine MEZGHICH

# Signature



# Acknowledgement

Foremost, we would like to express our sincere gratitude to our advisor Dr.Mohamed Amine Mezghich for the continuous support, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped us throughout all the stages of this project. Simply put, this project would not have been possible without the guidance and trust we received from him.

We would also like express our sincere gratitude to all of our teachers at the National School of Computer Science for their continuous help and treasurable training during our study years.

Finally, special thanks to the jury members who honored us by examining and evaluating this modest contribution.

# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Preliminary Study</b>	<b>3</b>
1.1 Presentation of the project . . . . .	3
1.2 Main Concepts . . . . .	3
1.2.1 Image segmentation . . . . .	3
1.2.2 3D Image reconstruction . . . . .	4
1.3 Problem setting . . . . .	4
1.3.1 Proposed solution . . . . .	4
1.4 Working Methodology . . . . .	5
1.4.1 Agile Methodology . . . . .	5
1.4.2 Scrum . . . . .	6
<b>2 State Of The Art</b>	<b>9</b>
2.1 Segmentation Methods . . . . .	9
2.1.1 HMRF-EM . . . . .	9
2.1.2 Deep Learning . . . . .	11
2.1.3 Level Set . . . . .	14
2.2 Reconstruction Methods . . . . .	16
2.2.1 Marching cubes . . . . .	16
<b>3 Requirements Analysis and Specification</b>	<b>18</b>
3.1 Requirements Analysis . . . . .	18
3.1.1 Actors . . . . .	18
3.1.2 Functional Requirements . . . . .	18
3.1.3 Non functional requirements . . . . .	19
3.1.4 Requirement Specification . . . . .	19
3.2 System Sequence Diagram . . . . .	20
<b>4 Design</b>	<b>25</b>
4.1 Global design . . . . .	25
4.1.1 Physical architecture . . . . .	25

---

4.1.2	Logical architecture . . . . .	26
4.2	Detailed design . . . . .	28
4.2.1	Package Diagram . . . . .	28
4.2.2	Class Diagram . . . . .	29
4.2.3	Sequence Diagram . . . . .	30
<b>5</b>	<b>Achievement</b>	<b>34</b>
5.1	Data Presentation . . . . .	34
5.2	Sprint1: Data Collection and Theoretical Study . . . . .	35
5.3	Sprint2: Design and Implementation of HMRF-EM and Level Set . . . . .	36
5.3.1	HMRF-EM . . . . .	36
5.3.2	Level Set . . . . .	36
5.4	Sprint3: Enhancing the performance of HMRF-EM and Level Set . . . . .	37
5.4.1	Enhancing the HMRF-EM performance . . . . .	37
5.4.2	Enhancing the Level Set performance . . . . .	43
5.5	Sprint4: Deep Learning and Marching Cubes Implementation . . . . .	44
5.6	Sprint5: Graphical User Interface . . . . .	46
5.7	Results . . . . .	48
	<b>Bibliographie</b>	<b>51</b>
	<b>Netographie</b>	<b>52</b>

# List of Figures

1.1	Proposed Solution . . . . .	5
1.2	Agile Principles **=[URL2 . . . . .	6
1.3	Scrum **=[URL3 . . . . .	7
1.4	To do list . . . . .	8
2.1	1st order neighbors . . . . .	11
2.2	2nd order neighbors . . . . .	11
2.3	U-net architecture(example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps.The arrows denote the different operations. . . . .	13
2.4	level set algorithm **=[URL4 . . . . .	16
2.5	Marching cubes input/output **=[URL6 . . . . .	17
3.1	Use case diagram . . . . .	20
3.2	Sequence diagram of the use case "Segment one slice using HMRF-EM" . .	21
3.3	Sequence diagram of the use case "Segment 3D MRI volumes using HMRF-EM" . . . . .	22
3.4	Sequence diagram of the use case "Segment 3D MRI volumes using Level Set"	23
4.1	One tier software architecture **=[URL5 . . . . .	26
4.2	two layer architecture . . . . .	27
4.3	Package Diagram . . . . .	28
4.4	Class Diagram . . . . .	29
4.5	Sequence Diagram -Level set- . . . . .	31
4.6	Sequence Diagram -HMRF-EM- . . . . .	33
5.1	The different tumor sub-regions **=[URL7 . . . . .	35
5.2	Before segmentation . . . . .	36
5.3	After segmentation . . . . .	36
5.4	Before segmentation . . . . .	37

5.5	After Segmentation . . . . .	37
5.6	Before fixing the adequate parameters . . . . .	38
5.7	After fixing the adequate parameters . . . . .	38
5.8	Ground truth . . . . .	38
5.9	Ground truth . . . . .	38
5.10	$k = 3$ . . . . .	39
5.11	$k = 5$ . . . . .	39
5.12	$k = 6$ . . . . .	39
5.13	Cropping tool . . . . .	39
5.14	After using the cropping tool . . . . .	40
5.15	The select-pixel tool . . . . .	40
5.16	Cropped flair modality . . . . .	41
5.17	HMRF-EM output . . . . .	41
5.18	After pixel selection . . . . .	41
5.19	t1ce modality . . . . .	42
5.20	after extraction . . . . .	42
5.21	Final result . . . . .	42
5.22	Flair modality . . . . .	43
5.23	Before fixing the parameters . . . . .	43
5.24	After fixing the adequate parameters . . . . .	43
5.25	Segmentation result . . . . .	44
5.26	Ground truth . . . . .	44
5.27	DL segmentation result . . . . .	45
5.28	Ground truth . . . . .	45
5.29	DL segmentation result . . . . .	45
5.30	Ground truth . . . . .	45
5.31	HMRF-EM interface . . . . .	46
5.32	Level Set interface . . . . .	47
5.33	Deep Learning interface . . . . .	47

# List of Tables

5.1	Dice score . . . . .	48
-----	----------------------	----



# List of Acronyms

**HMRF-EM** *Hidden Markov Random Field-Expectation Maximization*

**LS** *Level Set*

**DL** *Deep Learning*

**CNN** *Convolutional Neural Network*

**GUI** *Graphical User Interface*

**MVC** *Model view controller*

# General Introduction

The World Health Organization (WHO) states that around 400,000 people in the world are affected with the brain tumor and 120,000 people have died in 2016 . Glioblastoma is the most common form of brain cancer and is highly aggressive, with a 5 year survival rate of 5.3 aged 40 to 64 and median survival time of 331 to 529 days (depending on the course of treatment). In addition to high mortality rates, glioblastoma is very costly to treat, with a mean expenditure of over \$100,000 in the six months post surgery. Consequently, there exists a significant need to accurately diagnose gliomas and glioblastomas in their early stages.

Before the treatment of chemotherapy, radiotherapy, or brain surgeries, there is a need for medical practitioners to confirm the boundaries and regions of the brain tumor and determine where exactly it is located and the exact affected area. Manual segmentation of tumors is a challenging and time-consuming task. Moreover, there is a significant variability between segmentations produced by human experts. An accurate automatic or semi-automatic segmentation method could help in therapy planning and in monitoring of the tumor progression by providing the exact localization of tumor subregions and by precisely quantifying their volume.

Brain tumor segmentation is one of the most important and difficult tasks in many medical-image applications because it usually involves a huge amount of data. Artifacts due to patient's motion, limited acquisition time, and soft tissue boundaries are usually not well defined. There are large class of tumor types which have variety of shapes and sizes. They may appear in different sizes and types with different image intensities. Some of them may also affect the surrounding structures that change the image intensities around the tumor. Different tumor tissues (necrotic core, active rim, edema) can be imaged using multiple Magnetic Resonance (MR) sequences. For instance, T2-FLAIR sequence is suitable for detecting edema while T1-weighted MR images acquired after the injection of a gadolinium-based contrast product are suitable to detect active parts of the tumor core . These tumor tissues may be treated with different therapies and their analysis is important to assess the tumor characteristics, in particular its malignity

In this work, we try to shed the light on few promising methods used in medical image segmentation. We propose the implementation and adaptation of three different methods to the specific task of brain tumor segmentation. The first method is Level Set, an edge based method, the second one is HMRF-EM, a generative and modal based method, and the last one is a discriminative supervised method known as Convolutional Neural Network. Each method can then be used to segment a volume of MR images of patients suffering from brain tumor. The output of the segmentation is then used to reconstruct a 3D model of the tumor using a cutting-edge algorithm called Marching Cubes. Finally we can compare the performance of each algorithm to the ground-truth, a manual segmentation done by the experts of the field, and conclude the strengths and weaknesses of each algorithm. Our work is presented according to the following outline. In the first chapter, we conduct a preliminary study in which we introduce the main concepts and detail our working methodology, in the second chapter we introduce the algorithms used in this work and explain how they work, in the third, we make a list of the functional and non-functional requirements and we present the use case and the system sequence diagram related to our work. In the fourth chapter, we describe the project design and architecture. In the fifth and last chapter, we describe deeply each iteration done in our work and we present the graphical user interface. Finally we conclude our work by comparing the pros and cons of each method and then giving a sum up and some potential upgrades for our proposed software.

# Chapter 1

## Preliminary Study

This chapter will allow us to introduce our project. We will present the general context of our work , the main concepts and the result to hope for.

### 1.1 Presentation of the project

This project, titled "3D Tumor Segmentation and Reconstruction using Machine Learning", is done as part of the design and development project (PCD) of the second year of our engineering training. It is one of the modules of interest in our training that allows us to test our technical skills and project management. It is supervised by our teachers within four months during the second semester of the second year.

### 1.2 Main Concepts

In this section , we will focus on the main concepts of our project .

#### 1.2.1 Image segmentation

Image segmentation is an image processing operation that aims to gather pixels together according to predefined criteria. The pixels are thus grouped into regions, which constitute a partition of the image.

Segmentation is a crucial step in image processing. It can be grouped into 3 classes:

- Region-based segmentation : It relies on the search for areas with common attributes, either brightness or, more rarely, textures. The drawbacks of this method is determining the boundaries between regions.
- Edge-based segmentation : This approach seeks to exploit the fact that there is a detectable transition between two related regions. The difficulty lies in the presence of noise in the images , that is why there is a need to use filters to reduce noise.

- Classification segmentation ( or thresholding ) : This approach focus on the report that each pixel individually maintains with information calculated over the entire image , allowing to build n classes of intensity. Pixels belonging to the same class and being related form the regions of the image. The problem with this approach is that the quality of the solution depends on the initial set of clusters .

### 1.2.2 3D Image reconstruction

Refers to the technique of obtaining a three-dimensional representation of an object or a scene from a set of images taken from different points of view of the object or scene. One or more 2D representations of an object are available and it is desired to determine the coordinates of the elements visible on these representations in a reference of the real 3D space , that is the main idea of the 3D reconstruction.

## 1.3 Problem setting

In the medical field, specifically in radiology, segmentation, known to physicians as contouring, is a very important tool in the detection of tumors and the study of their evolution. Segmentation is a time consuming process. Moreover, there is a significant variability between segmentations produced by human experts. That is why the automatization of this operation is a necessity. In the next paragraph we will talk about our proposed solution .

### 1.3.1 Proposed solution

The solution we are going to present is the following:  
As mentioned before, there are several approaches that can be used. We chose to use three different approaches and compare their performances. The approaches are : Edge detection, Classification and Deep Learning.  
To do so ,we proceeded as follows:

- Implement the level set algorithm which is an edge detection approach. Segment every slices of the volume of the MR Image and compare the results of each slice with the ground truth.
- Implements the HMRF-EM algorithm which classifies the pixels of the MR images into clusters. After the segmentation ends , we will compare the volume and each slice with its equivalent in ground truth.
- We will use Convolutional Neural Networks to solve the segmentation task and just like the two previous approaches, we will compare the result of the segmentation with the ground truth segmentation.

After the segmentation operation ends and to better visualize our results , we would like to reconstruct the segmented volume . It will help the user to better estimate the success of the operation.

To do so we have chosen to implement the Marching cubes algorithm . The figure below will describe the process of our proposed solution.

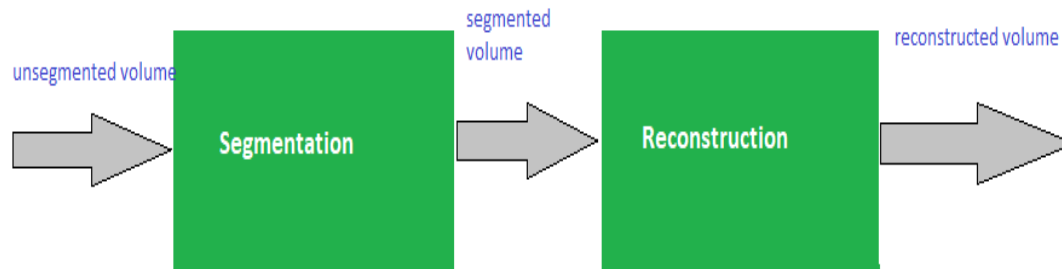


Figure 1.1: Proposed Solution

## 1.4 Working Methodology

In order to succeed in our project, we had to follow a working methodology. It allowed us to optimize our working time and share the tasks between the members of the group , so as to respect the time and constraints. So the best methodology to follow is one of the agile method obviously.

In this section we will present the agile methodology and we will talk about the Chosen methodology.

### 1.4.1 Agile Methodology

Agile methods are more pragmatic than traditional methods and rely on an iterative, incremental and adaptive development cycle.

The four basic values of the agile method are:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

One of the most used agile methods is Scrum and it is the method we choose to follow. In the next paragraph we will present Scrum and justify our choice .

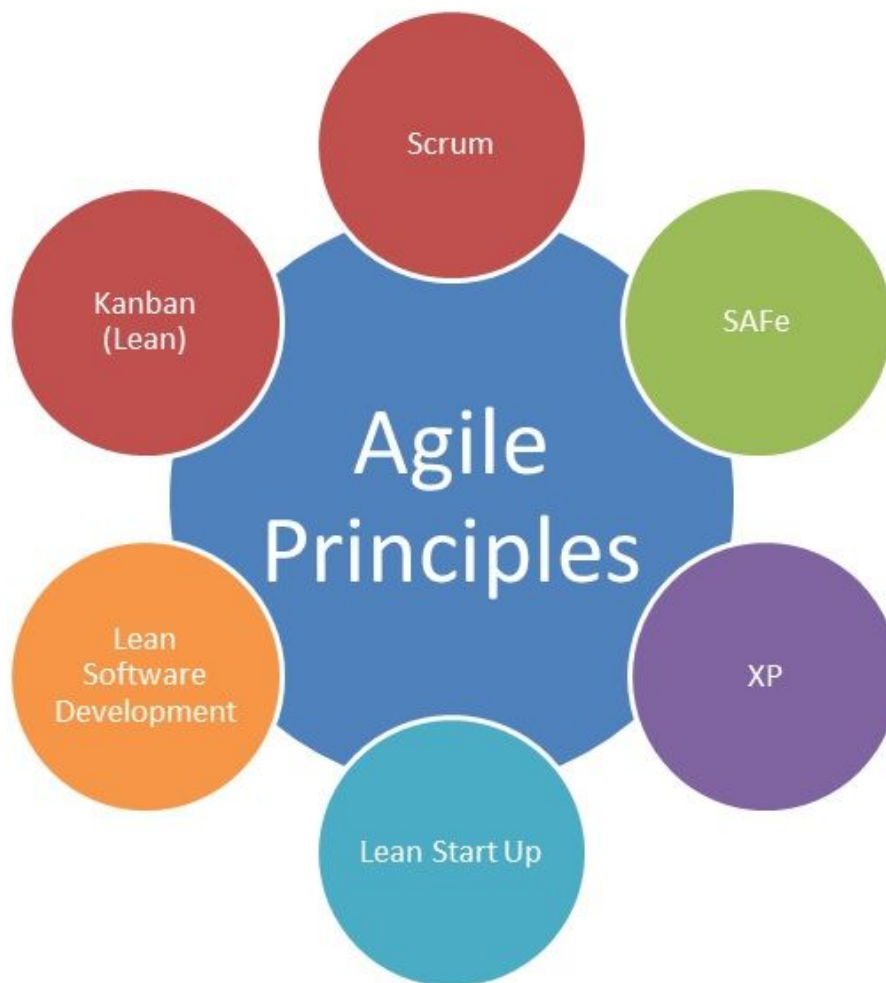


Figure 1.2: Agile Principles [URL1]

### 1.4.2 Scrum

Scrum is considered as a group of practices answering most of the recommendations of the Agile Manifesto. The framework is based on the division of a project in time boxes, called Sprints. Scrum starts with a user story in order to specify actions and requirements. In the figure below, we are going to describe the workflow of scrum methodology.



Figure 1.3: Scrum [URL2]

By matching the amount of work we have and the team capacity, we decided to choose the Scrum, because it is based on continuous cooperation between the members of the group via the Daily scrum, it allows the development team to synchronize their activities and plan the next 24 hours (every day, 15 minutes, same time, same place). During the daily meeting, we discuss the issues and understand the project-progress.

We also prefer scrum because we find that the Sprints will be very beneficial in the communication with our supervisor (a weekly meeting).

To do so, We used an online to-do list provided by a website called "Trello" the [U+FB01]gure below present a screen-shot of our list.



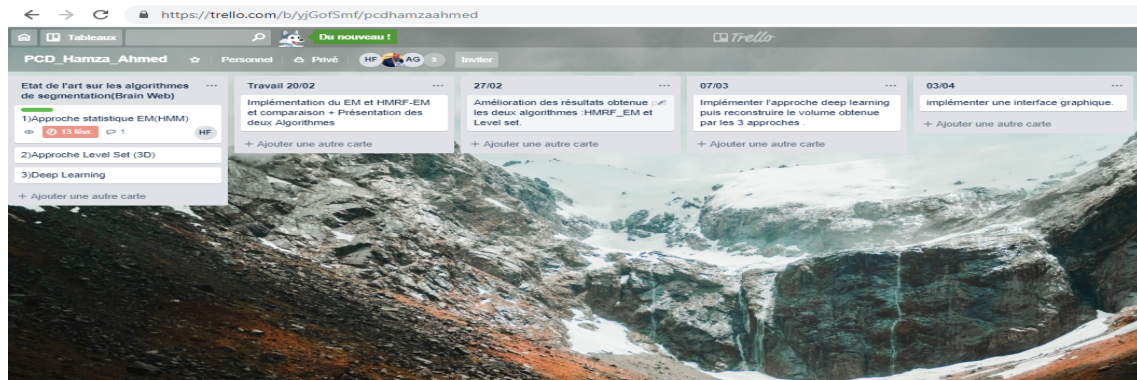


Figure 1.4: To do list

## Conclusion

This chapter included a presentation of our project and an overview of the problem setting. Also we justified the choice of our working methodology by presenting it and giving the reasons that made us chose it.

# Chapter 2

## State Of The Art

In this chapter, we will discuss the most pioneering methods in the domain of medical image segmentation and 3D reconstruction.

### 2.1 Segmentation Methods

Throughout the years, various methods were developed to segment brain tumors from MRI images. In this work, we will only cover three of them.

#### 2.1.1 HMRF-EM

##### 2.1.1.1 Introduction

Markov random field (MRF) theory provides a convenient and consistent way of modeling context dependent entities such as image pixels and other spatially correlated features. This is achieved through characterizing mutual influences among such entities using MRF probabilities. MRFs have been widely employed to solve vision problems at all levels. Most of the MRF models are for low level processing. These include image restoration and segmentation, surface reconstruction, edge detection, texture analysis, optical flow, shape from X, active contours, deformable templates, data fusion, visual integration, and perceptual organization. The use of MRFs in high level vision, such as for object matching and recognition, has also emerged in recent years. The interest in MRF modeling in computer vision is still increasing, as reflected by books as well as journals and conference papers published in recent years. It's worth mentioning that the first use of HMRF models for brain tumor segmentation backs to the 2000s with the emergence of the revolutionary paper of Zhang et al., 2001. [2]

### 2.1.1.2 Definition

In real images, regions are often homogeneous, neighboring pixels usually have similar properties (intensity, color, texture...) Markov Random Field (MRF) is a probabilistic model which captures such contextual constraints.

### 2.1.1.3 A Mathematical Approach

Segmentation can be seen as a pixel labelling task. First, we extract features from the input image. Each pixel  $s$  in the image has a feature vector. For the whole image, we have:

$$f = \left\{ \vec{f}_s, s \in S \right\}$$

Now, let's define the set of labels  $\Lambda$  :

- Each pixel  $s$  is assigned a label  $\omega_s \in \omega$
- For the whole image we have :  $\omega = \{\omega_s, s \in S\}$

Then, let's define a probability measure on the set of all possible labelings and select the most likely one.

$P(\omega | f)$  measures the probability of a labelling, given the observed feature  $f$ .

Our goal is to find an optimal  $\hat{\omega}$  labeling which maximizes  $P(\omega | f)$ .

This is called the Maximum a Posteriori (MAP) estimate:

$$\hat{\omega}^{MAP} = \operatorname{argmax}_{\omega \in \Omega} P(\omega | f)$$

Wishing to express the distribution of the possible states of our model, we use the Gibbs distribution:

$$P(\omega | f) = \frac{1}{|Z|} \exp(-U(\omega))$$

where  $U$  is the energy function of our MRF model.

In order to define the energy function of our MRF model, we need first to introduce some concepts.

#### the notion of Neighbors:

For each pixel, we can define some surrounding pixels as its neighbors. The labeling field  $X$  can be modeled as a Markov Random Field if:

- For all  $\omega \in \Omega$  :  $P(X = \omega) > 0$
- For every  $s \in S$  and  $\omega \in \Omega$  :  $P(\omega_s | \omega_r, r \neq s) = P(\omega_s | \omega_r, r \in N_s)$

#### The notion of Clique:

A subset  $C \subseteq S$  is called a clique if every pair of pixels in this subset are neighbors.

A clique containing  $n$  pixels is called  $n^{th}$  order clique, denoted by  $C_n$ .

#### the notion of Clique Potential:

For each clique  $c$  in the image, we can assign a value  $V_c(\omega)$  which is called clique potential

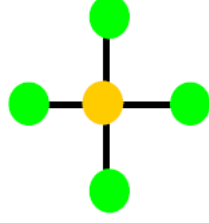


Figure 2.1: 1st order neighbors

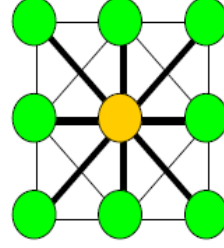


Figure 2.2: 2nd order neighbors

of  $c$ , where  $\omega$  is the configuration of the labeling field. For instance, the value of a second order clique is :

$$V_{c_2}(i, j) = \beta \delta(i, j) = \begin{cases} -\beta & \text{if } \omega_i = \omega_j \\ +\beta & \text{if } \omega_i \neq \omega_j \end{cases}$$

- In this model, Pixel labels are represented by Gaussian distributions:

$$P(f_s | \omega_s) = \frac{1}{\sigma_{\omega_s} \sqrt{2\pi}} e^{-(f_s - \mu_{\omega_s})^2 / 2\sigma_{\omega_s}^2}$$

Finally, we can define the energy function of our MRF model:

$$U(\omega) = \sum_s (\log(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2}) + \sum_{s,r} \beta \delta(\omega_s, \omega_r)$$

The energy function is represented by two parts: the cluster binding in the image and the neighborhood energy for all the cliques. Here,  $U(\omega)$  is the total energy of a label field  $\omega$ . In conclusion, to find the optimal  $\hat{\omega}$  labeling, we must maximize  $P(\omega | f)$  which would amount to minimizing the energy function  $U(\omega)$ . To estimate our model parameters automatically, we use the **Expectation-Maximization** algorithm.

## 2.1.2 Deep Learning

### 2.1.2.1 Introduction

A meaningful feature extraction is the keystone to apply machine learning algorithms for data analysis effectively. Commonly, domain-related features were mostly designed by human experts, which made it challenging for non-experts to exploit machine learning techniques. However, deep learning has alleviated such complications by absorbing the feature engineering step into a learning step. That is, instead of extracting features in a hand-designed manner, deep learning requires only a set of data with minor preprocessing, if necessary, and then discovers the informative representations in a self-taught manner and therefore allowing non-experts in machine learning to use efficaciously deep learning for their researches or applications, especially in medical image analysis.

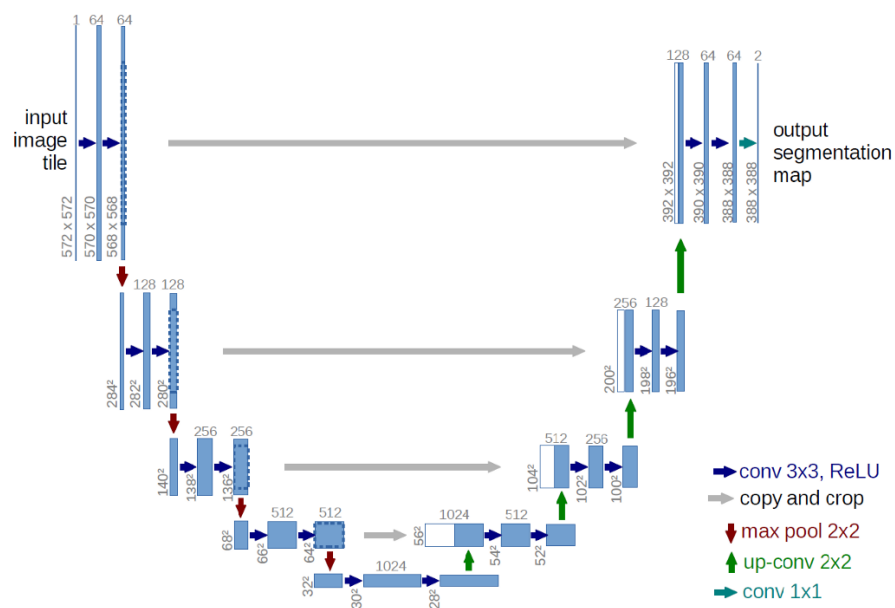
### 2.1.2.2 Deep Learning in Computer Vision

Record-breaking performances in a variety of artificial intelligence applications and challenges have been achieved thanks to the deep learning's characteristic of learning hierarchical feature representations solely from data. In particular, outstanding improvements in computer vision inspired its use to medical image analysis such as image segmentation, image registration, image fusion, image annotation, computer-aided diagnosis and prognosis, lesion and tumor detection, and microscopic imaging analysis, to name a few. The unprecedented success of deep learning arises mostly from the following factors:

- Advancements of high-tech central processing units (CPUs)
- Advancements of high-tech graphics processing units (GPUs)
- Availability of a huge amount of data (i.e., big data)
- Developments of learning algorithms

### 2.1.2.3 Deep Learning in Medical Image Segmentation

Multimodality magnetic resonance imaging is the primary method of screening and diagnosis for gliomas. Accurate segmentation of the tumor to determine features such as volume, spread, and location is critical to diagnosis and forming a course of treatment. Currently, tumor regions are segmented manually by radiologists, but advances in computer vision have made possible the ability to automate the segmentation process. Specifically, tumor segmentation algorithms based on convolutional neural networks (CNNs) have been shown to be at least as effective as other automated tumor segmentation methods. Several CNN-based methods have been proposed for brain tumor segmentation from multimodal MRI, including those based on segmenting individual MRI slices, volumetric segmentation, and CNNs combined with other statistical methods. Nearly all current architectures for brain tumor segmentation use a pixel-wise U-net approach. Ronneberger's U-net [1]. is a breakthrough towards automatic image segmentation and has been applied successfully in many tasks that require image-to-image transforms, for example, images to segmentation masks. Originally proposed in 2-D, many extensions, such as 3-D versions, exist. Below is the network architecture float



### 2.1.3 Level Set

#### 2.1.3.1 Introduction

The level set method is a digital technique for analyzing surfaces and shapes. Solving topology problems during the evolution of curves can be effectively done by the level set method, while the previous algorithms cannot deal with them. In recent years, there are many image segmentation algorithms based on level set method. The idea behind all level set algorithms is to represent the curve or surface in question at a certain time  $t$  as the zero level set (with respect to the space variables) of a certain function  $u(t, x)$ , the so called level set function. Thus the initial surface is the set  $x|u(0, x) = 0$ . Of course the method works for hyper surfaces of arbitrary dimensions, but for applications obviously only curves (two space variables) and surfaces (three space variables) are important.

#### 2.1.3.2 Definition

It is a generic numerical method for evolving fronts in an implicit form. It handles topological changes of the evolving interface and define problem in one higher dimension.

#### 2.1.3.3 A Mathematical Approach

The evolution of the surface in time is caused by forces or fluxes normal to the surface. The speed of point on the surface normal to the surface will be denoted by  $F(x)$  and is called the speed function. For points on the zero level set it is usually determined by physical models and in our case by the fluxes of certain gas species and subsequent surface reactions. The speed function  $F(x)$  generally depends on the time and space variables, and we assume at first that it is defined on the whole simulation domain and for the time interval considered.

Suppose we have surface  $\phi(x)$ . The c-level set of this surface is given by:

$$\{x|\phi(x) = c\} \quad (2.1)$$

Formally, we want to track the level curve at  $c=0$ , which is the zero level set of  $\phi$ .

$$\{x|\phi(x) = 0\} \quad (2.2)$$

As we are dealing with curve and surface evolution, we will parameterize our surface with a temporal variable  $t$  such that:

$$\phi(x(t), t) = 0 \quad (2.3)$$

We could think of that as the surface at time  $t$ , given the variable  $x$  at time  $t$ . Next, as we want to track the movement of the zero level curve of  $\phi$ , we will derive it with respect

to  $t$  i.e. we derive the equation of motion of  $\phi$ . Remember, the derivation of position is speed, and knowing the speed, we could model the movement of the surface.

$$\frac{\partial \phi(x(t), t)}{\partial t} = 0 \quad (2.4)$$

Using chain rule, we get:

$$\frac{\partial \phi}{\partial x(t)} \frac{\partial x(t)}{\partial t} + \frac{\partial \phi}{\partial t} = 0 \quad (2.5)$$

$$\nabla \phi X_t + \phi_t = 0 \quad (2.6)$$

As we state, the direction of the curve's movement is normal, which is  $\frac{\nabla \phi}{\|\nabla \phi\|}$ , also the speed function  $F$ . Hence, the speed vector is given by  $x_t = F \frac{\nabla \phi}{\|\nabla \phi\|}$

$$\nabla \phi F \frac{\nabla \phi}{\|\nabla \phi\|} = 0 \quad (2.7)$$

$$F \|\nabla \phi\| + \phi_t = 0 \quad (2.8)$$

we have the level set equation :

$$\phi_t = -F \|\nabla \phi\| \quad (2.9)$$

This gives us the speed of the surface evolution of  $\phi$ .

Knowing the initial value of  $\phi$  and the speed of evolution, we can solve the equation of motion. It is a Partial Differential Equation.

$$\frac{\partial \phi(x(t), t)}{\partial t} = \frac{\phi(x(t), t + \Delta t) - \phi(x(t), t)}{\Delta t} \quad (2.10)$$

$$\Delta t \phi_t = \phi(x(t), t + \Delta t) - \phi(x(t), t) \quad (2.11)$$

$$\phi(x(t), t + \Delta t) = \phi(x(t), t) + \Delta t F \|\nabla \phi\| \quad (2.12)$$

In the end

$$\phi' = \phi + \Delta t F \|\nabla \phi\| \quad (2.13)$$

So that is it. We just need to provide initial value for  $\phi$  and figure out the equation of force  $F$ , which depend on the system we are going to model.

### 2.1.3.4 Algorithm

This image summarizes and shows each step of the level set algorithm.



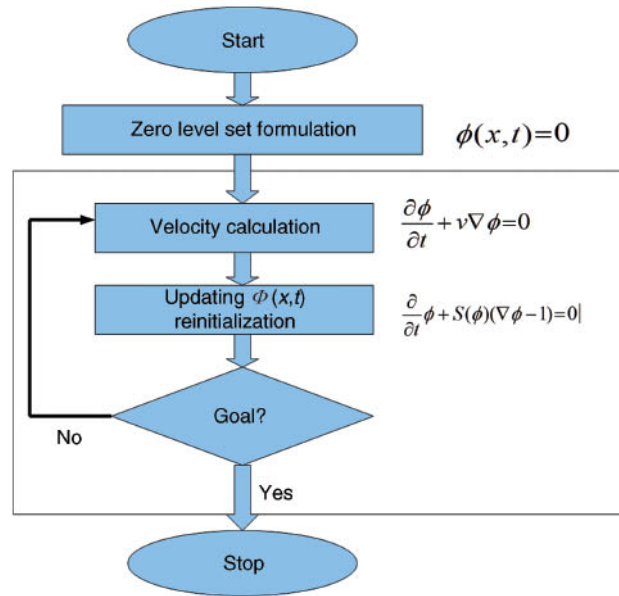


Figure 2.4: level set algorithm [URL3]

## 2.2 Reconstruction Methods

In this section we are going to talk about the reconstruction method that we used for reconstruct MRI images of segmented brain tumors .

### 2.2.1 Marching cubes

#### 2.2.1.1 Introduction

Volume visualization has been applied in a lot of problem domains and has become an important tool for exploring data and discovering knowledge. Marching cubes is a computer graphics algorithm, published in the 1987 SIGGRAPH proceedings by Lorensen and Cline, for extracting a polygonal mesh of an isosurface from a three-dimensional discrete scalar field ( called a voxel).

The Marching Cubes is a 3D extension of the Marching Squares 2D technique .

Medical visualizations ,such as CT and MRI scan data images,are one of the main application of this algorithm .

#### 2.2.1.2 Definition

Having an object, to determine if an arbitrary point is within the object, and bounds within which the object exists: The space within the bounds is divided into an arbitrary number of cubes.The corners of every cube is tested for whether they are inside the object. For every cube where some corners are inside and some corners are outside the object, the

surface must pass through that cube, intersecting the edges of the cube in between corners of opposite classification. Draw a surface within each cube connecting these intersections and you get your object .

### 2.2.1.3 Algorithm

The Marching cubes algorithm extract a 2D surface mesh from a 3D volume. It iterates across the volume, searching for regions that cross the level of interest. If such regions are found, Marching cubes generates triangulations and add them to an output mesh. The result is a set of vertices and triangular faces.

The algorithm requires a data volume and an isosurface value.

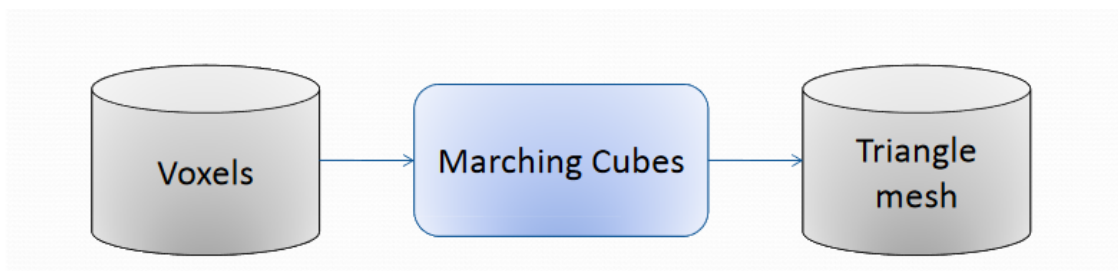


Figure 2.5: Marching cubes input/output [URL4]

The algorithm determines the intersection of the surface with the cube, moves or marches to the next cube. To find the surface intersection in a cube, if the data value in a vertex equals or exceeds the value of the surface that we are constructing, we give that vertex a one. Cube vertices with values below the surface receive a zero and are outside the surface. The surface intersects the cube edges where one of the vertex is outside the surface and all the others are inside. With this assumption, we can determine the topology of the surface in the cube.

## Conclusion

This chapter was a gentle introduction to the state of the art algorithms we used in our project. In the next chapter we will enumerate the different features offered by our application.

# Chapter 3

## Requirements Analysis and Specification

In order to guarantee the success of the project, a good analysis of the system is required. This chapter is dedicated to the analysis and specification of functional and non-functional requirements. We will finally present the main interactions between the user and the app using system sequence diagrams.

### 3.1 Requirements Analysis

#### 3.1.1 Actors

The main actors of our system are the radiologists who can use our system to get the result of the segmentation of brain tumors using three different approaches, compare the approaches, besides getting a 3D model of the tumor.

#### 3.1.2 Functional Requirements

The system must enable the user to:

- Upload MRI volumes.
- Visualize the data according to the specified axis and slice number
- Choose a segmentation method
- Carry out a slice or volume segmentation on the uploaded data
- Build a 3D model of the tumor out of the segmentation result
- Save the segmentation results

### 3.1.3 Non functional requirements

Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviours

- ★ **Performance:** The app must provide the services in a reasonable amount of time and the segmentation methods and the 3D reconstruction must be accurate.
- ★ **Reliability:** The application must be functional regardless of any circumstances
- ★ **Ergonomy:** The application must offer a user-friendly and ergonomic interface that must be easy to use
- ★ **Maintainability:** The code must be structured and understandable in order to ensure the extensibility of the application

### 3.1.4 Requirement Specification

#### 3.1.4.1 Use Case Diagrams

In order to illustrate the user's interaction with the system, we represent the use case diagram that shows the relationship between the user and the different use cases in which the user is involved.

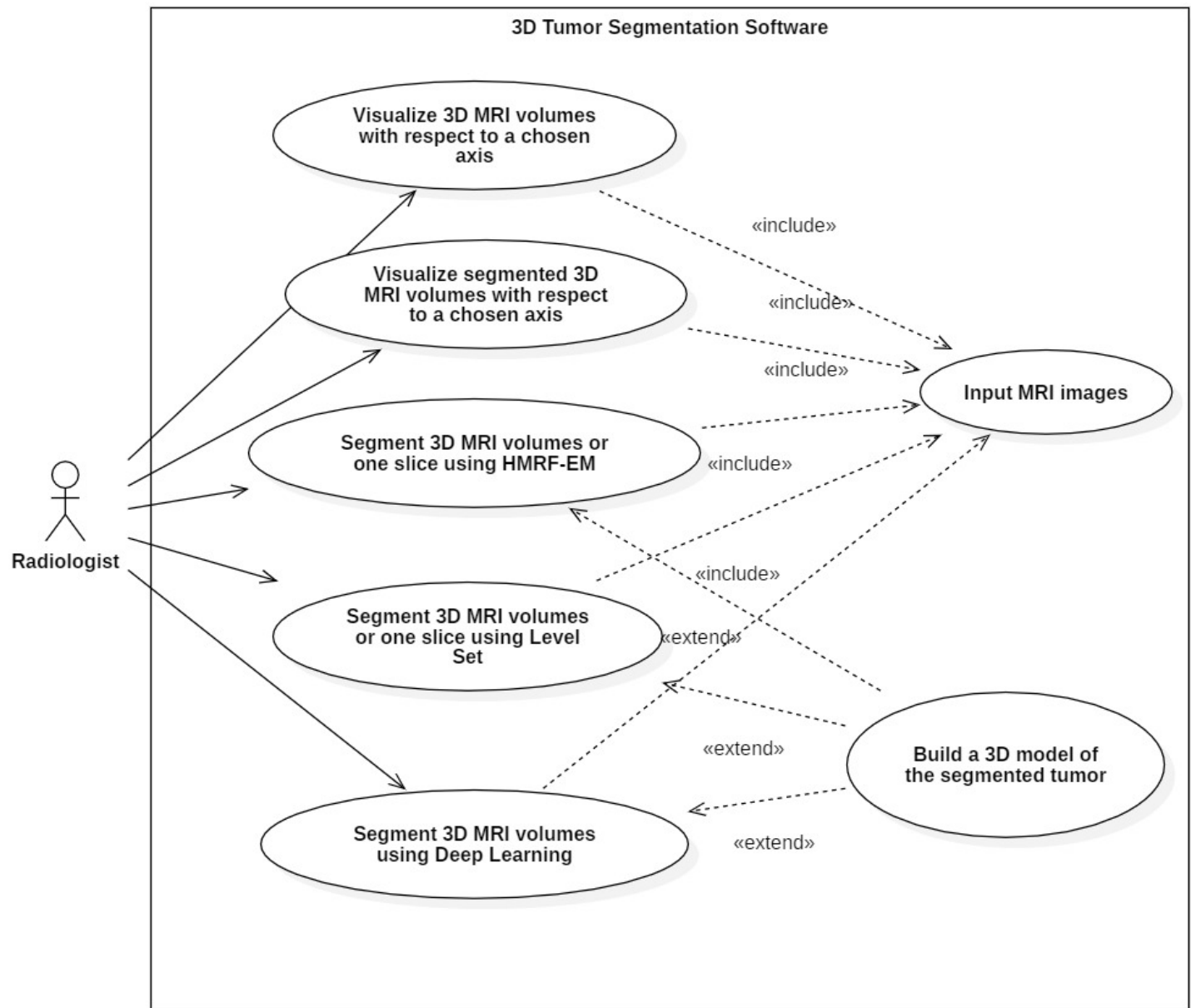


Figure 3.1: Use case diagram

## 3.2 System Sequence Diagram

In this section, we detail the different possible interactions with the user and the segmentation methods offered by the app from a temporal point of view.

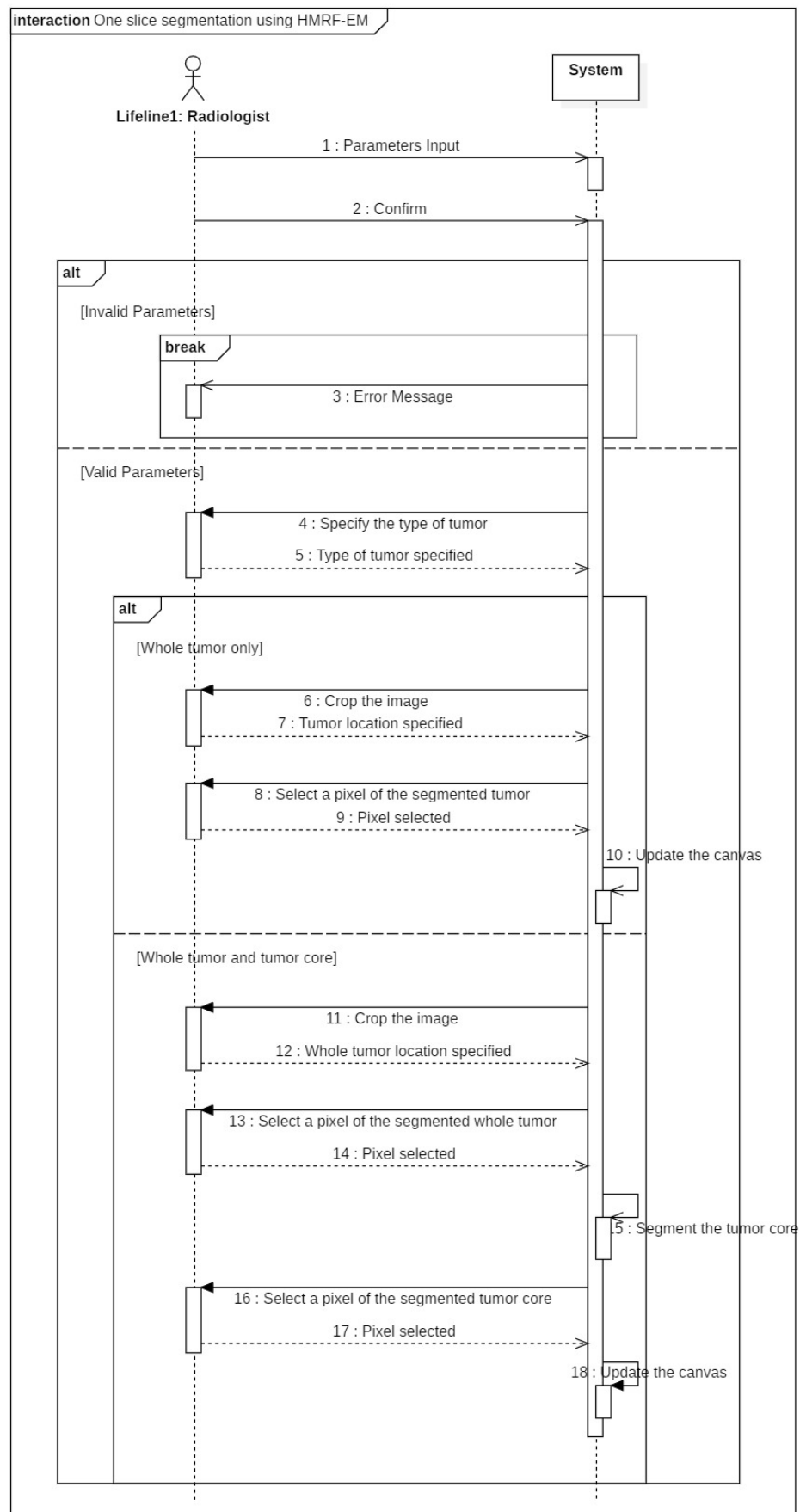


Figure 3.2: Sequence diagram of the use case "Segment one slice using HMRF-EM"

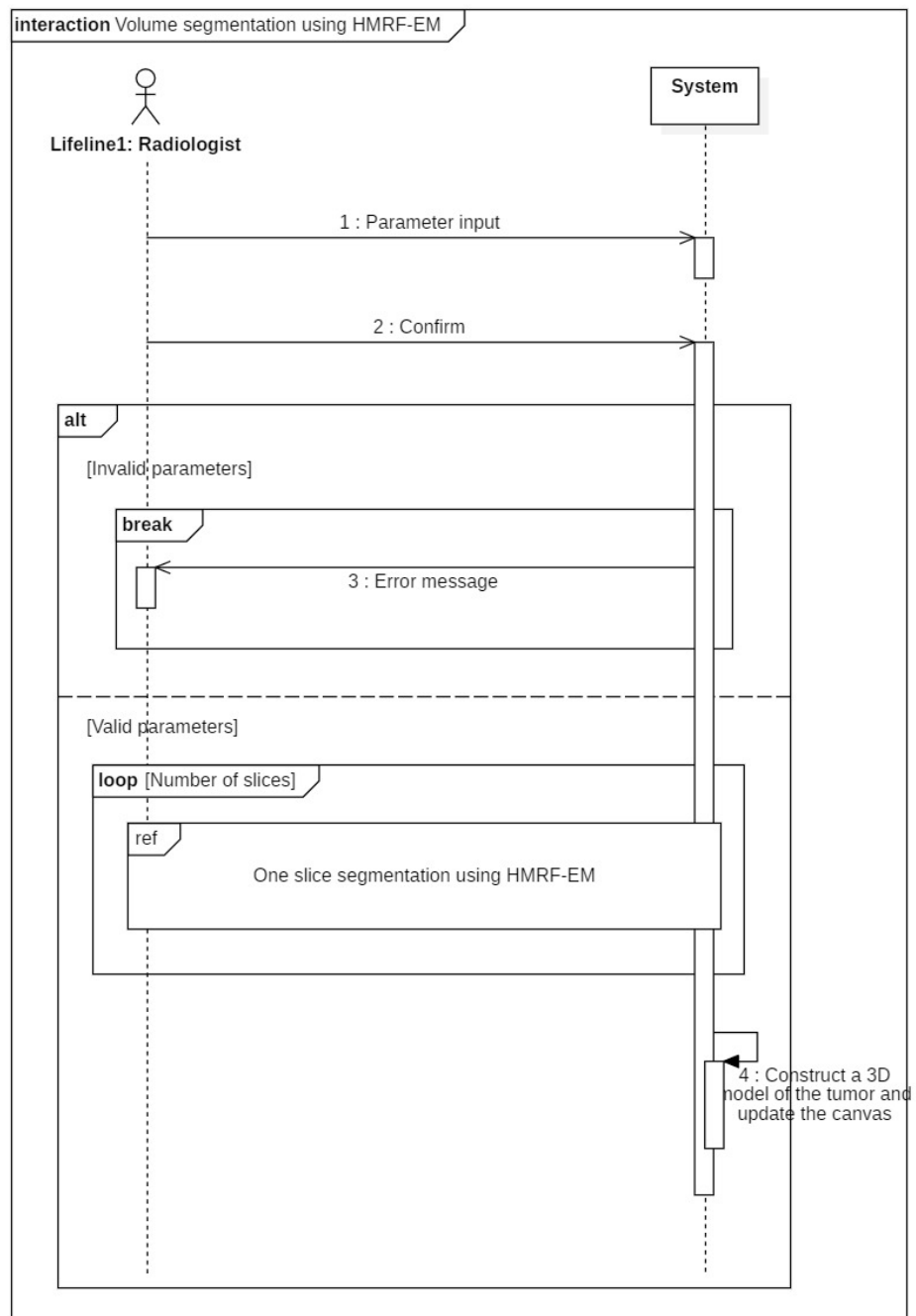


Figure 3.3: Sequence diagram of the use case "Segment 3D MRI volumes using HMRF-EM"

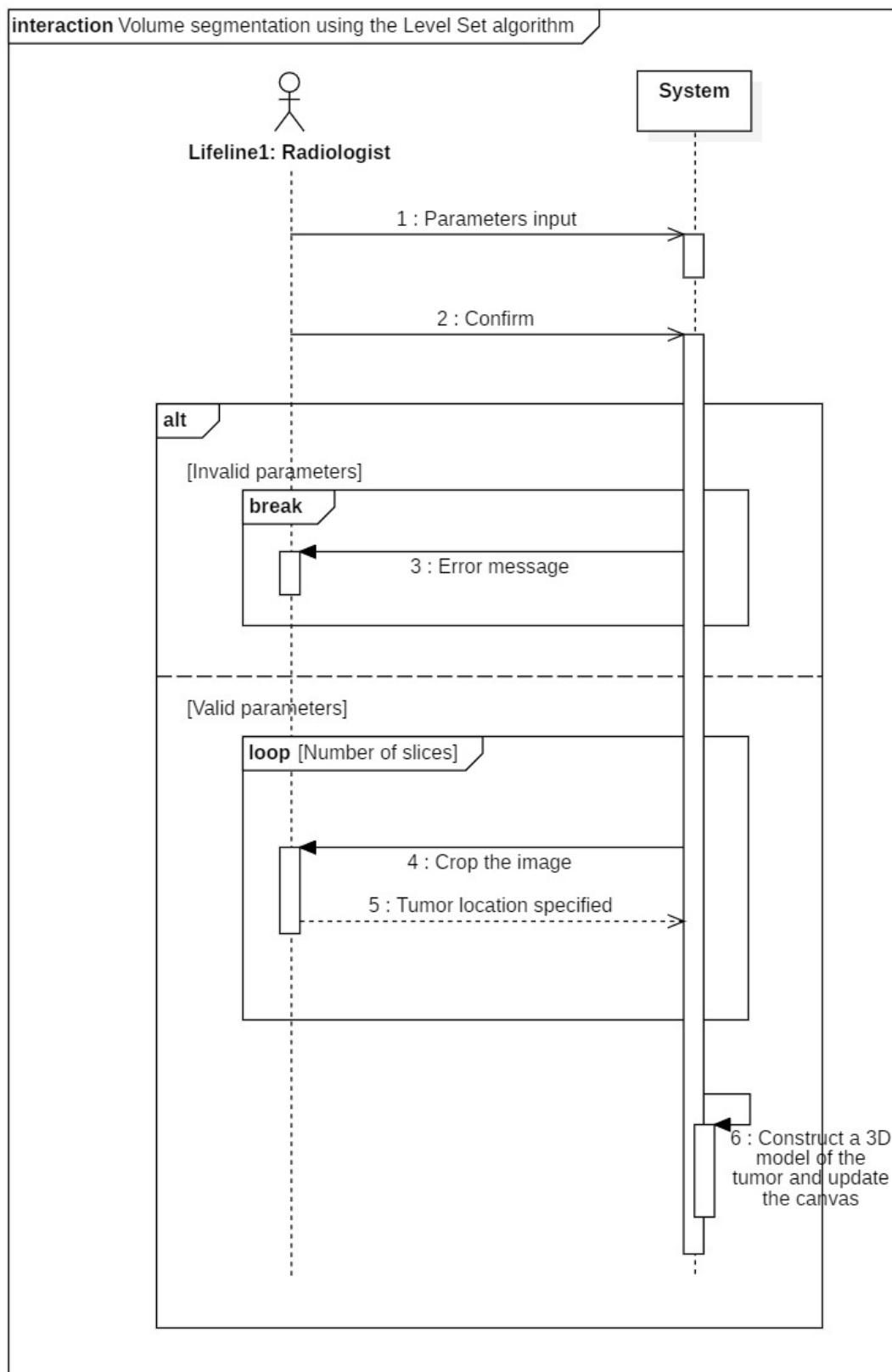


Figure 3.4: Sequence diagram of the use case "Segment 3D MRI volumes using Level Set"



Figure 2.2 and Figure 2.3 detail the interactions between the user and the system while using the HMRF-EM algorithm to segment one slice or a whole volume. The HMRF-EM algorithm we implemented is able to segment two of the Glioma sub-regions: the whole tumor visible in T2-FLAIR modality, and the tumor core visible in T1CE modality. In each slice segmentation, the different modalities of the slice will be displayed and then the radiologist have to specify whether the slice contains only a whole tumor region, or both whole tumor and tumor core sub-regions. After the segmentation of any of the sub-regions, the radiologist is asked to select one pixel containing the segmented sub-region. This step is necessary to keep the segmented region's label the same across all the slices, which will enable us later to reconstruct the 3D model of the segmented region.

Figure 2.4 details the interactions between the user and the system while using the Level Set algorithm. The Level Set algorithm is quite easy to use as it only requires the cropping of the region containing the tumor to help the algorithm. However, level set only allows the user to segment the whole tumor region.

The Deep Learning approach is the easiest to use. It only requires that the user upload the data and confirm, and then the algorithm will take care of detecting whether a certain slice contains a tumor, and segmenting the whole uploaded MRI volume. There are few interactions between the user and the system while using the Deep Learning approach, thus the non-necessity to accord a sequence diagram to it.

## Conclusion

Throughout this chapter, we identified the actors, specified the functional and non functional requirements of our application and we presented the use cases and the nominal scenarios of our project. The next chapter will be devoted to present the project design.

# Chapter 4

## Design

In order to achieve the appropriate result described in the requirements analysis and specification, In this chapter we will focus on the main architecture and the details of its components.

### 4.1 Global design

The architecture of a software describes its global structure, its skeleton. It describes the main elements that make up the software, as well as the flux of exchanges between these elements.

We can divide the design into two parts :Physical and Logical

#### 4.1.1 Physical architecture

Physical Architecture consists of One Tier, Two Tier, Three Tier and N-Tier architectures. In our case we used the 1-tier architecture where the Client, Server, and Database all reside on the same machine as you can see in the figure below .

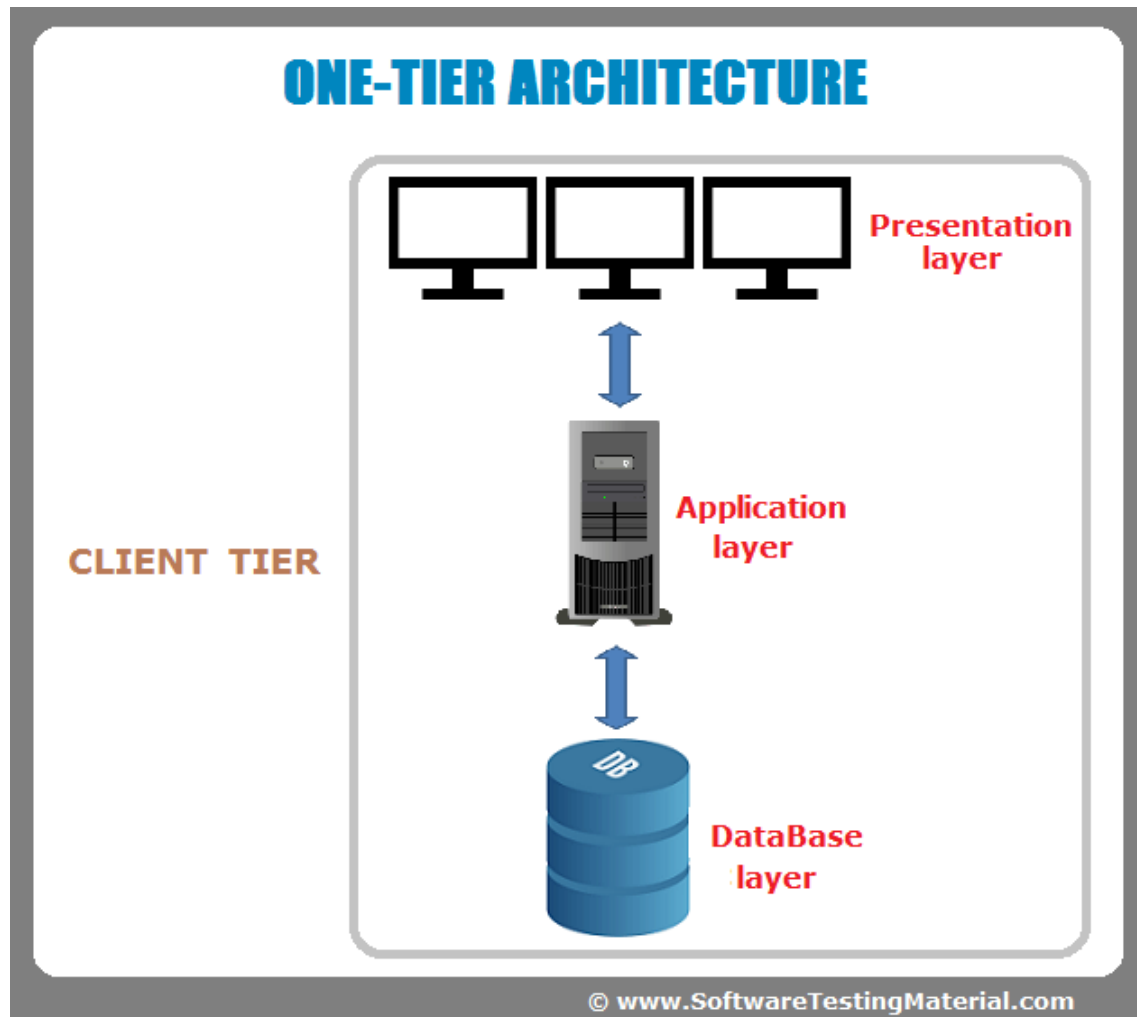


Figure 4.1: One tier software architecture [URL5]

We opt for this choice because we have developed a desktop application that does not require an external server. One Tier architecture is an ideal solution for application with low traffic demand which require effective resource utilization.

#### 4.1.2 Logical architecture

MVC is a 3-layer logical architecture (Model ,View,Controller) and it is very popular for web applications. The MVC is effective in the case where we have a large database, it allows us to access this database in an efficient way thanks to the controller but in our case we do not have that kind of problem.

Instead of using MVC, we used a 2-layer architecture that is derived from MVC because we did not see the advantage of using a 3-layer architecture in our application (a desktop application with limited data).

The two layers we used are:

- Model :we will find components that defines the business logic ,such as algorithms that we used (in segmentation and reconstruction).
- View-Control:we will find the Graphic user interface ('GUI'). It contains components for displaying the interface and components responsible for coordinating with the Model layer.

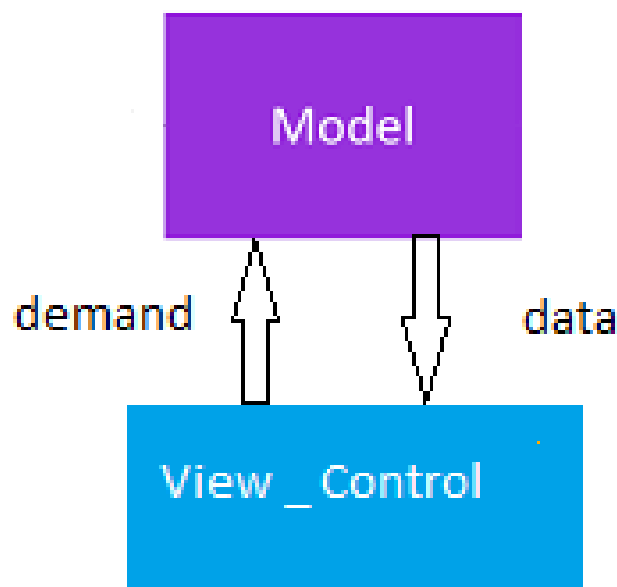


Figure 4.2: two layer architecture

We should notice that any change in the model may affect the execution of the application. That is why we used a View-Control layer, it acts as a coordinator and is responsible for the integrity of the Model.

In our application the Model contains the business logic : segmentation classes , reconstruction classes and classes responsible of preprocessing the data (a crucial task in our work).

The model layer provides the View-Control layer with the requested data.

The View-Control layer, which is in our case the HMI (Human -Machine Interface), will get the input from the user and send it to the model layer. After the execution , the View-Control layer will show the output to user.

## 4.2 Detailed design

In this section we will dive deeper in the design of the application. First ,we will begin by describing the package diagram ,it will confirm our logical architecture. Second ,we will do the class diagram , necessary to describe the relation between classes in each package. Finally , we will show a detailed sequences diagram presenting the process of executing the algorithm of Level Set and HMRF-EM .

### 4.2.1 Package Diagram

The package diagram will give us a global view on the connection between the two layers and the location of classes in packages without focusing on the relations between them (between classes) ,as you can see in the figure below.

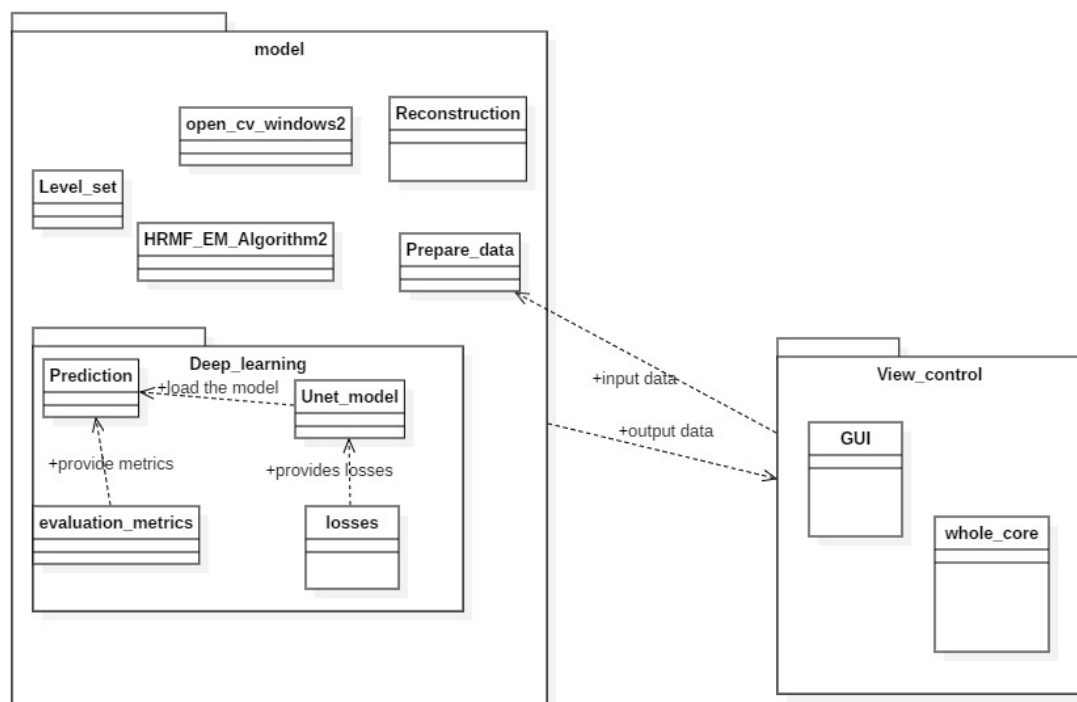


Figure 4.3: Package Diagram

### 4.2.2 Class Diagram

The class diagram will allow us to understand the relationships between classes belonging to the same package.

#### 4.2.2.1 Class Diagram -Model-

In this section , we will be focusing on the connection between the classes belonging to the Model package.

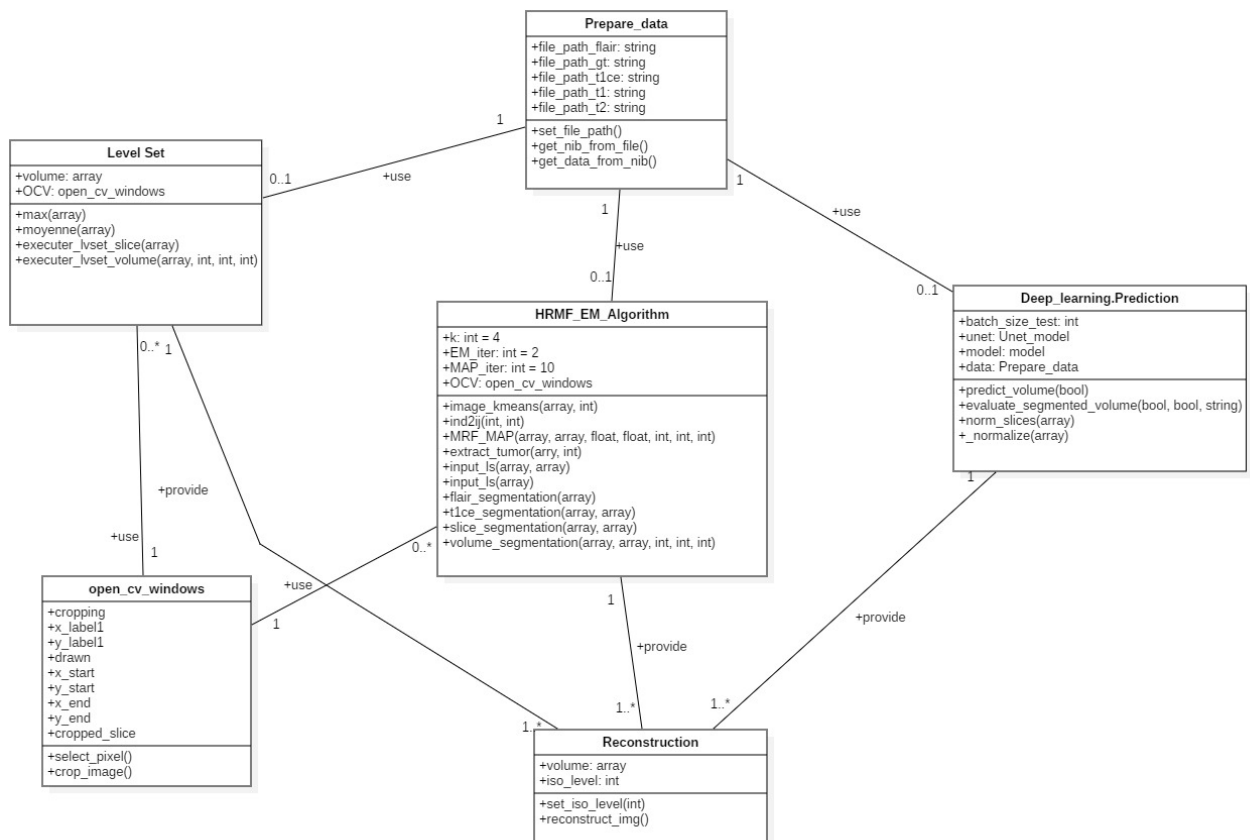


Figure 4.4: Class Diagram

We will describe the classes used in the diagram above (without focusing on the details).

- **Prepare\_data** :it is the class responsible of preprocessing the input data .
- **open\_cv\_windows** :it provides the algorithms with adequate data that will enable to have a good output.

- `Level_set` : it is the class that will enable us to execute the Level set algorithm.
- `HMRF_EM_Algorithm` : as indicated by its name , this class is responsible of execution HMRF\_EM algorithm.
- `Prediction` : it enables the prediction of a segmentation in the deep learning model.
- `Reconstruction` : after the segmentation of the whole volume , this class will let us reconstruct the volume in 3D.

### 4.2.3 Sequence Diagram

The sequence diagram models the relationship between objects in a sequential order. In this section , we chose to focus on two sequence diagrams :

- The first diagram will describe the operations of segmenting a slice with the Level set method.
- The second will explain how the slice segmentation works with the HMRF\_EM method and the interrelationships between the objects concerned.

#### 4.2.3.1 Level set sequences diagram

As mentioned in the paragraph above, we will detail the communication between the objects for the segmentation operation of the Level set method and the reconstruction. The different components that will interact are :

- a `Level set` object , considered as the main object of this scenario.
- an `open_cv_windows` object , we will use the `crop` method .
- a `Prepare_data` object , it will provide the `Level set` object with input data.
- a `Reconstruction` object , it will reconstruct the segmented volume.

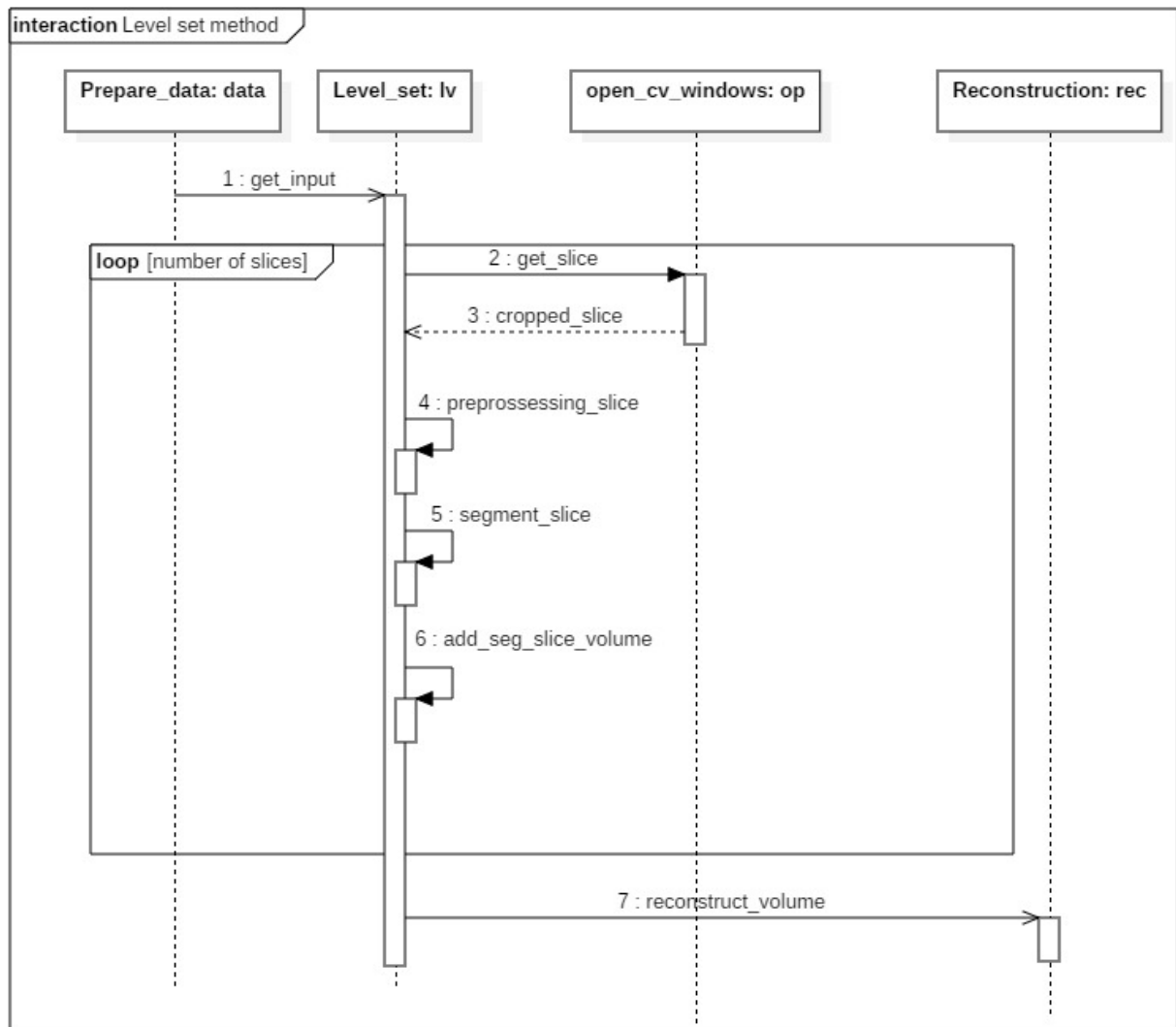


Figure 4.5: Sequence Diagram -Level set-



#### 4.2.3.2 HMRF\_EM sequence diagram

In this section, we will explain the communication between the objects contributing in the execution of the HMRF\_EM method and reconstructing of the volume.

The interacting objects are:

- a HMRE\_EM\_Algorithm object , considered as the main object of this scenario.
- an open\_cv\_windows object , we will use the crop and select\_pixel methods.
- a Prepare\_data object , it will provide the HMRF\_EM object with input data.
- a Reconstruction object , it will reconstruct the segmented volume.

## Conclusion

Throughout this chapter, we have specified our (physical and logical) architecture. We have detailed the connections between the packages and between the classes and we have specified the sequencing of the operations for the two methods: HMRF-EM and Level Set.

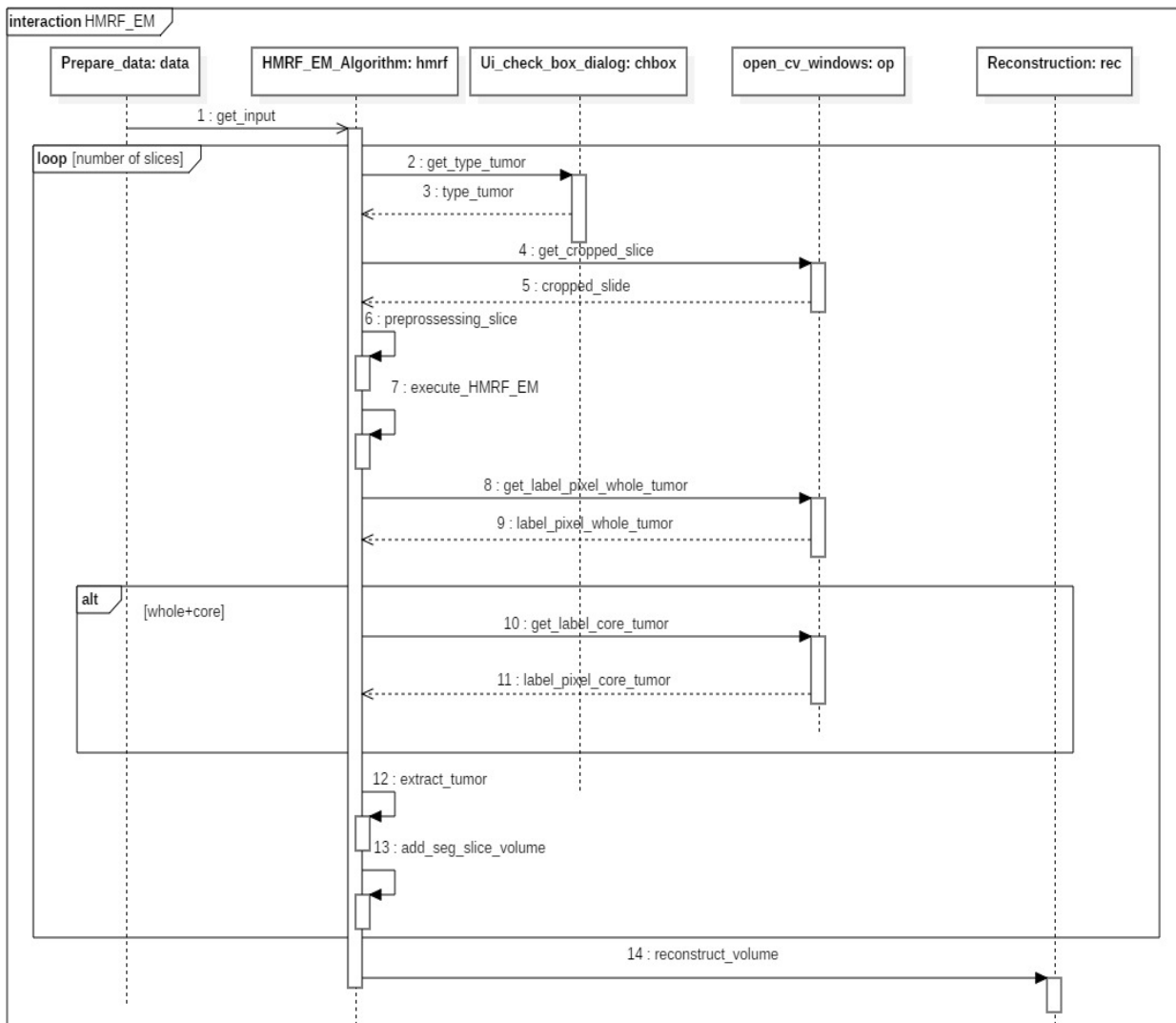


Figure 4.6: Sequence Diagram -HMRF-EM-

# Chapter 5

## Achievement

In this chapter, we will enumerate and detail every sprint of our work and we will discuss the difficulties we encountered and how we managed to solve them.

### 5.1 Data Presentation

Before addressing the sprints, we think that it is crucial to present the data we worked on.

The Multimodal Brain Tumor Segmentation (BraTS) is a challenge held annually since 2012 in conjunction with MICCAI conference. Each year, the challenge committee present an updated version of last year’s dataset. To get access to this dataset, we had to create an account in CBICA’s Image Processing Portal and wait for its approval. Once our account was approved, we had to fill a form to request the data. We finally managed to get the BraTS’2017 and BraTS’2018 datasets. The BraTS 2018 consists of 220 brains with high-grade and 54 brains with low grade gliomas. All BraTS datasets, share four MRI modalities namely; T1, T1CE, T2, FLAIR. All images are skull stripped. In addition to the multimodal MRI scans, for each brain, a ground truth segmentation is also provided. All the imaging datasets have been segmented manually, by one to four raters, following the same annotation protocol, and their annotations were approved by experienced neuro-radiologists.

The tumor structures are grouped in 3 different tumor regions:

- 1-The complete tumor region (also called whole tumor)
- 2-The core tumor region
- 3-The enhancing tumor region

Figure 5.1 shows the different tumor sub-regions. The whole tumor (yellow) visible in FLAIR modality. The tumor core (red) visible in T1CE. The enhancing tumor region (light blue and green) is visible in T2 modality.

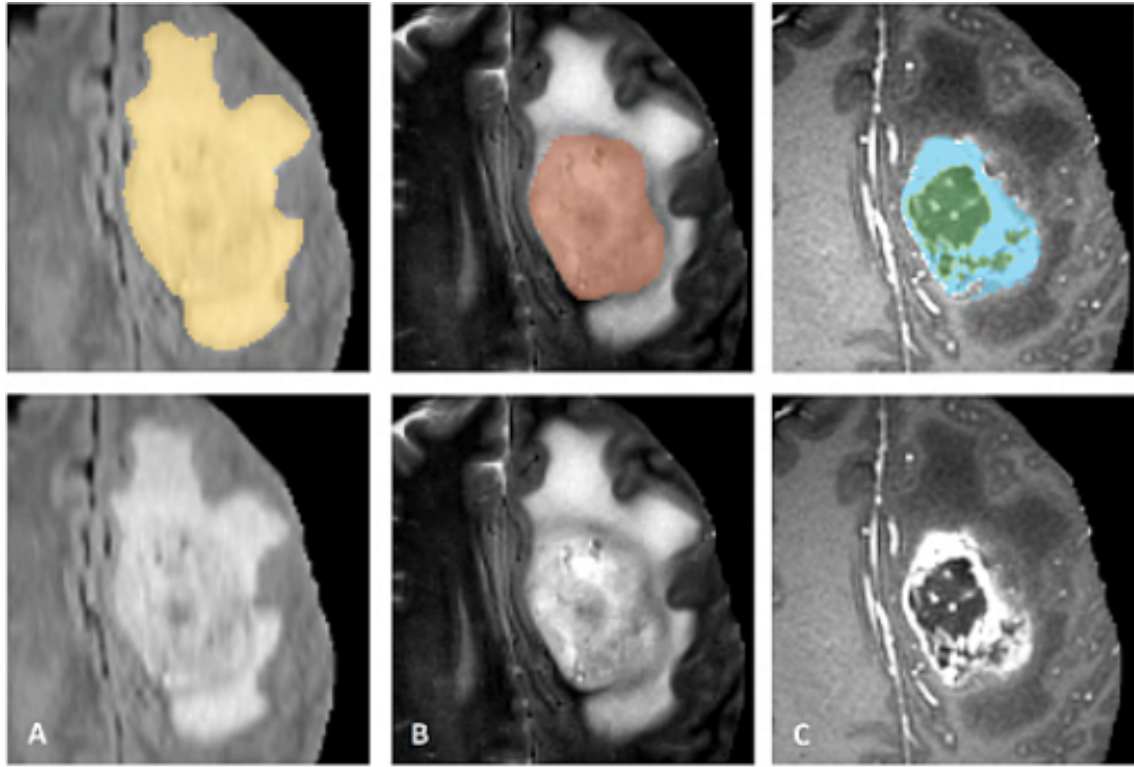


Figure 5.1: The different tumor sub-regions [URL6]

## 5.2 Sprint1: Data Collection and Theoretical Study

In the beginning, the choice of the tumor type to work on was not clear. We considered working on cardiac tumors, liver tumors or brain tumors. It turned out that working on cardiac tumors was pretty hard since two of the tumor segmentation methods we intended to use (HMRF-EM and Level Set) are intensity-based segmentation methods. Cardiac MRIs are characterized by weak or blurred object boundaries which makes it difficult for our algorithms to properly delineate the regions of interests. Regarding liver MRIs, they also suffered from the same problem, besides the lack of datasets containing manually segmented tumors that we could have used to compare the results we obtained with the ground truth segmentation. We finally decided to focus on brain tumors since the BraTS datasets offer 4 different modalities, which are perfect for our algorithms. However, a new challenge arose with this choice: we now have to segment 3 different sub-regions instead of one.

In this sprint, a large amount of time was devoted to the understanding of the math behind our algorithms. A solid understanding of the algorithms is essential to implement them successfully and to adapt them to the specific task of brain tumor segmentation. The HMRF-EM and Level Set algorithms were the most time consuming approaches to

understand. Regarding the Deep Learning approach, it did not take us much time because we had a general background on the matter.

## 5.3 Sprint2: Design and Implementation of HMRF-EM and Level Set

In this section, we will talk about the process we followed to implement the algorithms.

### 5.3.1 HMRF-EM

The HMRF-EM algorithm was particularly hard to implement. Searching the web, we found a handful of research papers talking about the model and its applications, but none of them offered a detailed algorithm. We also did not find any library that supported the HMRF-EM. One research paper in particular has helped us enormously in this task. The research paper offered a MATLAB implementation of the HMRF-EM algorithm. Since we need a licence to use MATLAB, we tested the algorithm on Octave, which is an open source software featuring a high-level programming language, primarily intended for numerical computations and mostly compatible with MATLAB. Though there is a library called Oct2Py that allows us to seamlessly call M-files and Octave functions from Python, we decided to not use the code, because when we tested the algorithm in Octave, the segmentation was really slow. The segmentation of one single image could sometimes take 10 to 15 minutes. Therefore, we decided to implement the algorithm from scratch in Python. We first tested it on regular RGB images. Here is an example of a dog image before and after the segmentation using HMRF-EM.



Figure 5.2: Before segmentation

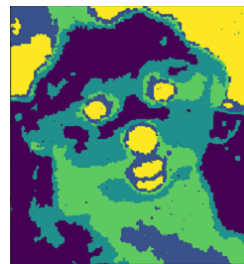


Figure 5.3: After segmentation

### 5.3.2 Level Set

We first tried to implement our own version of the algorithm. The output of the implemented version was acceptable. However, one issue led us to use Chan-Vese model for

active contours implemented in Scikit-Image ( an open-source image processing library for the Python programming language): our implemented version was very slow compared to the Scikit-Image implementation.

When testing the Level Set algorithm on regular images, we found mixed results. Sometimes the segmentation was good, especially when the images has clear boundaries, but other times, the segmentation was unacceptable. Here is an example:



Figure 5.4:  
Before  
segmentation



Figure 5.5: After  
Segmentation

When testing both of the algorithms on the MR brain images, the results were inaccurate and sometimes unacceptable. Thus, we decided to devote the next sprint to adapt the algorithms to our task and to improve the performance of each model.

## 5.4 Sprint3: Enhancing the performance of HMRF-EM and Level Set

First, we will begin with the HMRF-EM algorithm.

### 5.4.1 Enhancing the HMRF-EM performance

One particular problem we faced while working on the HMRF-EM algorithm was finding the adequate hyperparameters. We had to find the best filter to use to reduce the noise before starting the segmentation process, the best parameters to the filter and the best parameters for the algorithm (number of clusters, number of MAP iterations, number of EM iterations, number of neighbors). An exhaustive search to find these parameters was not an option since the HMRF-EM algorithm takes approximately 90 seconds to segment one single slice. Figure 4.6, figure 4.7 and figure 4.8 show the result of the segmentation of one slice before finding the adequate parameters (on the left), after finding the adequate parameters (in the middle) and the ground truth segmentation (on the right)

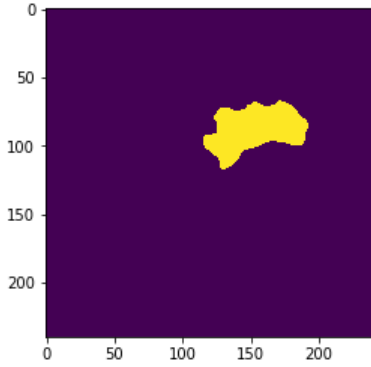


Figure 5.6: Before fixing the adequate parameters

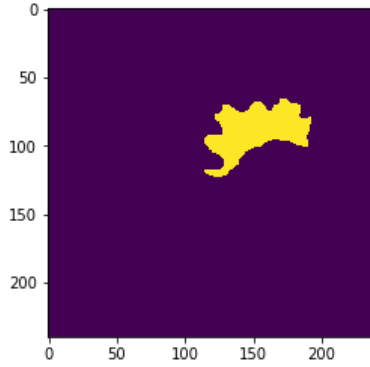


Figure 5.7: After fixing the adequate parameters

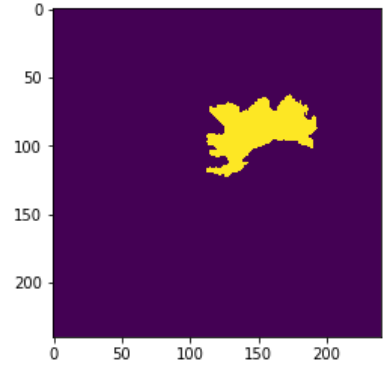


Figure 5.8: Ground truth

Fixing the number of clusters  $k$  was without a doubt the hardest part of this sprint. the number of clusters depends heavily on the input. Some slices required only  $k = 4$  clusters to get the tumor segmented, others required up to  $k = 7$  to get a good result. In the following example, we show the problem with fixing the number of clusters  $k$ . The figure 5.9 represents the ground truth segmentation of a slice.

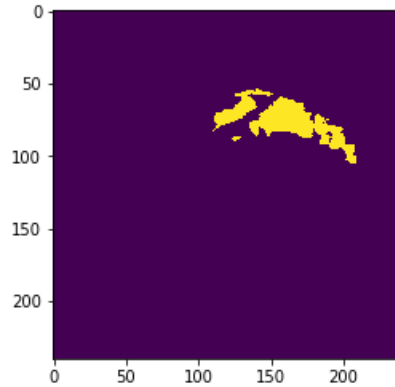
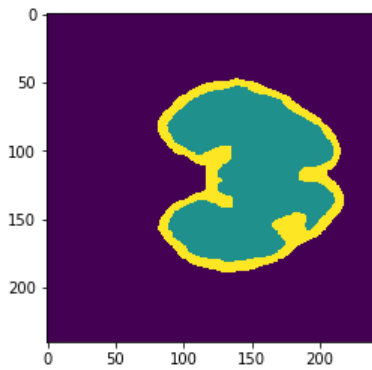
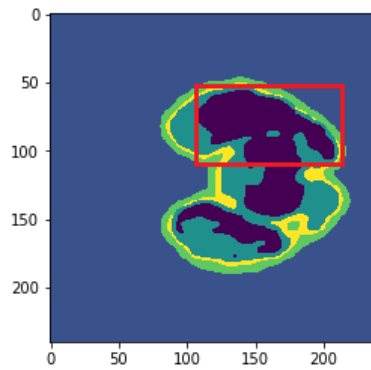
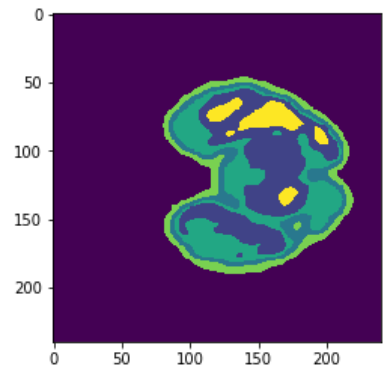


Figure 5.9: Ground truth

Figure 5.10 , figure 5.11 and figure 5.12 represent the result of segmentation (before post-processing the output) using different values of  $k$ .

As seen, when the values of  $k$  is too low, the algorithm is unable of detecting the tumor. With value  $k = 5$ , HMRF-EM could detect the tumor (the red rectangle in figure 5.11) but it also considered some benign part of the tumor as malignant. One way of fixing this problem is by augmenting the value of  $k$  to force the algorithm to consider the benign part as another cluster. However, this solution did not work for all the slices. In figure 5.12 we can see how augmenting the value of  $k$  did not solve the problem.

Figure 5.10:  $k = 3$ Figure 5.11:  $k = 5$ Figure 5.12:  $k = 6$ 

To solve this problem properly, we decided to implement a cropping tool that will help the algorithm to localize the tumor. Before every segmentation, the different modalities are shown to the radiologist and a window pops up containing the flair modality. In this window, the radiologist can easily detect where the tumor is located, and is asked to manually crop the image, leaving only the part containing the tumor. The cropping tool allows us to enhance significantly the performance of HMRF-EM, fix the value of  $k = 4$  to all the slices, and to gain in terms of processing time since the HMRF-EM will only be applied to the cropped area. Figure 5.13 shows the implemented cropping tool.

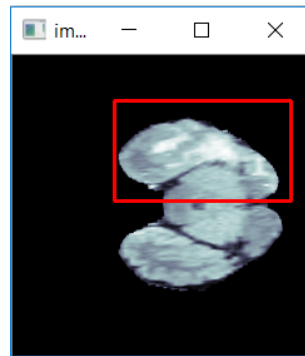


Figure 5.13: Cropping tool

The result of the segmentation after using the cropping tool is shown in the figure 5.14



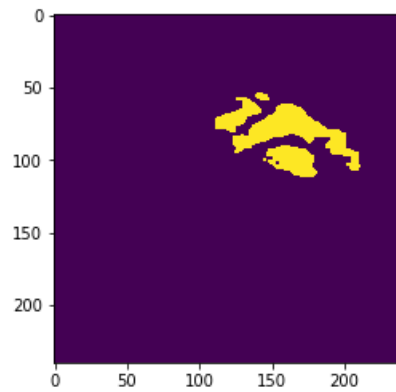


Figure 5.14: After using the cropping tool

One last problem we encountered with the HMRF-EM was the value of the pixels characterizing the tumor. As seen previously in figure 5.11 and figure 5.12, random values are attributed to the pixels of a certain cluster, which makes the task of isolating the tumor in the post-processing phase hard. In addition, this issue could cause serious problems later in the reconstruction phase. That's why we had to implement a simple tool that shows the result just after the clustering, asking the radiologist to simply select one pixel that belongs to the cluster representing the tumor. Figure 5.15 shows this tool. The red point in the middle represent the selected pixel.

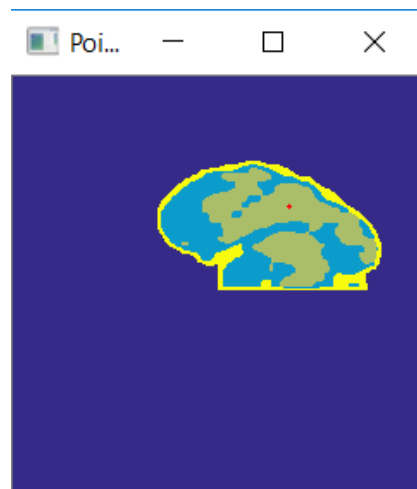


Figure 5.15: The select-pixel tool

One final thing that probably should be mentioned, is that we are using the HMRF-EM algorithm to detect only the whole tumor. For the tumor core, we are actually using the OpenCV implementation of the Expectation Maximization algorithm that estimates the parameters of the multivariate probability density function in the form of a Gaussian mixture distribution. That being said, the OpenCV version does not follow the HMRF model aforementioned and thus does not take into consideration the notion of pixel neighbourhood. In reality, we tried to use the HMRF-EM algorithm to segment the tumor core, but we found out that the OpenCV version of EM performed better in this task, while the HMRF-EM performed better at the task of whole tumor segmentation. Maybe the reason behind this is the fact that HMRF-EM consider that images are often homogeneous and neighboring pixels usually have similar properties, whereas, the tumor core region is relatively heterogeneous. To sum up the whole process of segmenting the tumor using the HMRF-EM approach, we will briefly describe the steps in order:

1- After cropping the flair modality, we apply the HMRF-EM algorithm on the cropped part to extract the whole tumor. Using the pixel-selection tool, we obtain the figure 5.18

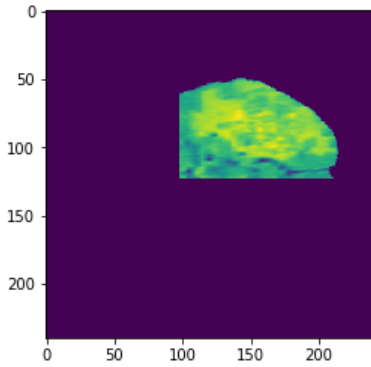


Figure 5.16: Cropped flair modality

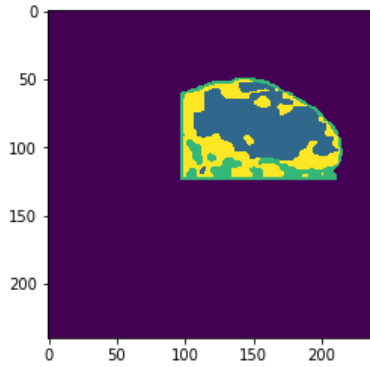


Figure 5.17: HMRF-EM output

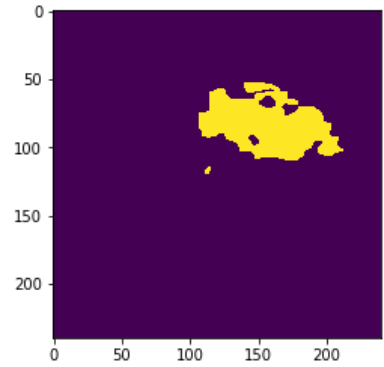


Figure 5.18: After pixel selection

2- Next, using the t1ce modality (used to extract the tumor core), we set the value of all the pixels except the yellow ones in figure 5.18 to 0.

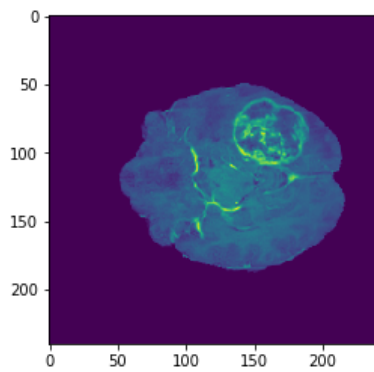


Figure 5.19: t1ce modality

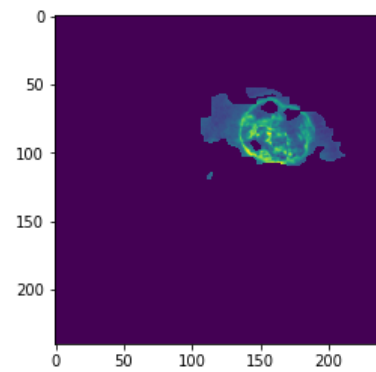


Figure 5.20: after extraction

3- We then apply the OpenCV EM algorithm on the image in figure 5.20 to extract the tumor core and we apply again the select-pixel tool to keep track of the value of each cluster. We finally get the result in figure 5.21.

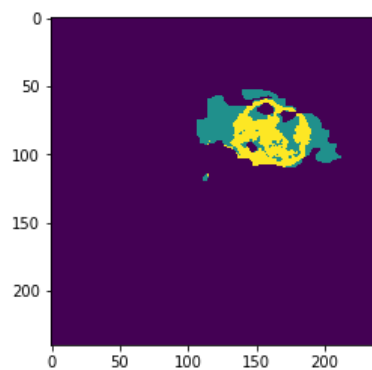


Figure 5.21: Final result

Next, we will talk about enhancing the performance of the Level Set algorithm.

### 5.4.2 Enhancing the Level Set performance

We encountered many problems we've already dealt with during the HMRF-EM performance enhancement.

Figure 5.22 shows the FLAIR modality that we are using to extract the whole tumor, while figure 5.23 shows the result of the segmentation using Level Set before fixing the parameters.

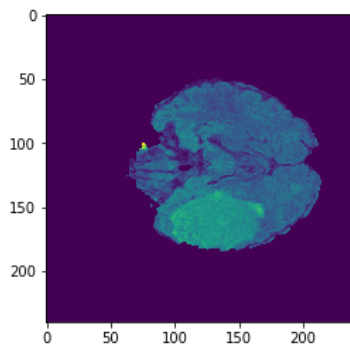


Figure 5.22: Flair  
modality

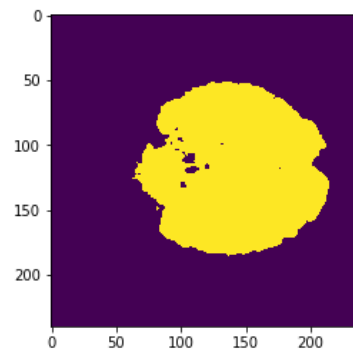


Figure 5.23:  
Before fixing the  
parameters

After fixing the adequate parameters, we found the following result shown in Figure 5.24

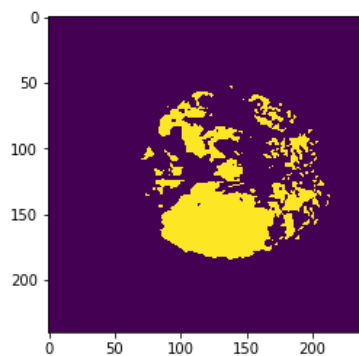


Figure 5.24: After fixing the adequate parameters

As seen, the Level Set algorithm managed to detect the tumor but left a lot of noise. This was the moment when we decided to use the cropping tool that enhanced greatly the result of the segmentation. The following 2 figures show the result of the segmentation after using the cropping tool (on the left) and the ground truth (on the right).

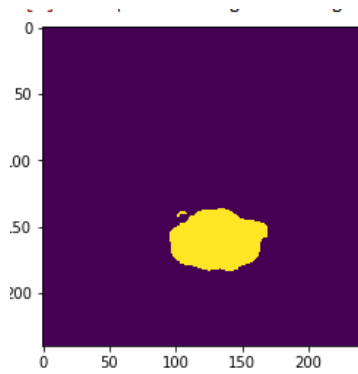


Figure 5.25:  
Segmentation  
result

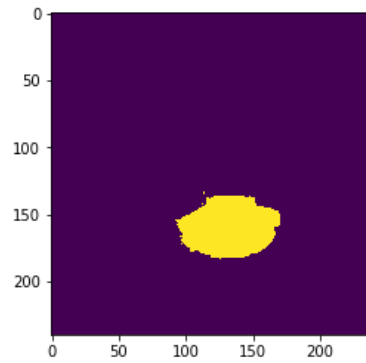


Figure 5.26:  
Ground truth

## 5.5 Sprint4: Deep Learning and Marching Cubes Implementation

At this stage, we took care of addressing the Deep Learning approach and the reconstruction algorithm Marching Cubes. Regarding the Deep Learning approach, we used a pre trained U-Net model which is a convolutional neural network that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg, Germany. Dozens of U-Net versions were implemented for the specific task of brain tumor segmentation. Many reasons led us to work with a pre-trained model. First of all, implementing the U-Net architecture and training the model would take us an extensive amount of time. Secondly, training the model would require an immense computational power. The AWS services we are provided with would have not been enough. Last but not least, The pre-trained model we borrowed is the fruit of years of research. Our model could have not compete with its outstanding performance.

Our work consisted of finding the adequate model, understanding the code behind it, debug it, deploy it to our project and test it on several MRI images. It should be noted that this CNN model is capable of segmenting the 3 different aforementioned sub-regions of the brain tumor. Figure 5.27 shows the result obtained by this model, compared to the ground truth in figure 5.28 .

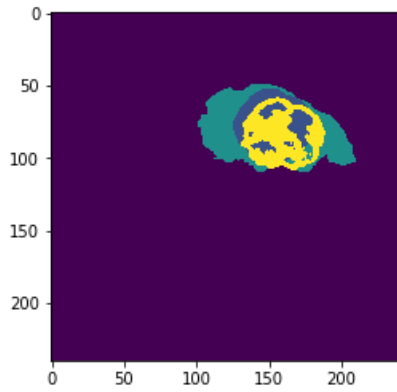


Figure 5.27: DL  
segmentation  
result

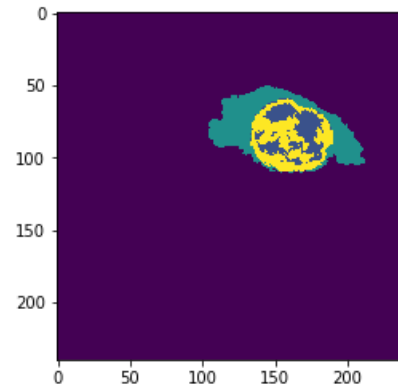


Figure 5.28:  
Ground truth

As for the Marching Cubes algorithm, we used the implementation of the Scikit-Image library. We fixed the adequate parameters of the algorithm and tested it on the outputs of the 3 different approaches. Figure 5.29 shows the result of reconstructing the tumor from the output of the Deep Learning approach, while figure 5.30 shows the reconstruction using the ground truth arrays.

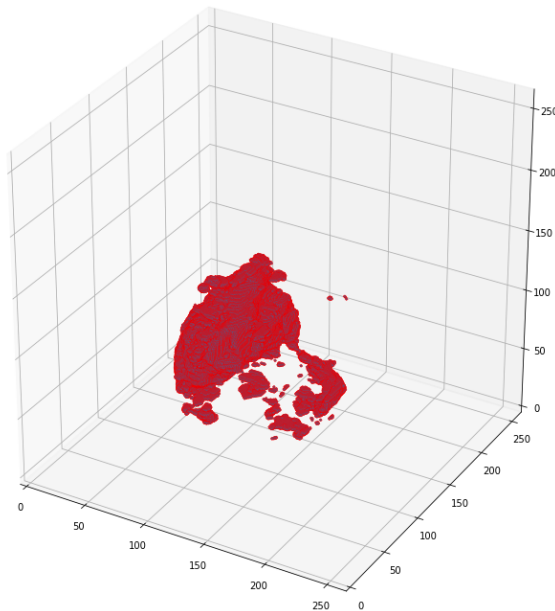


Figure 5.29: DL  
segmentation  
result

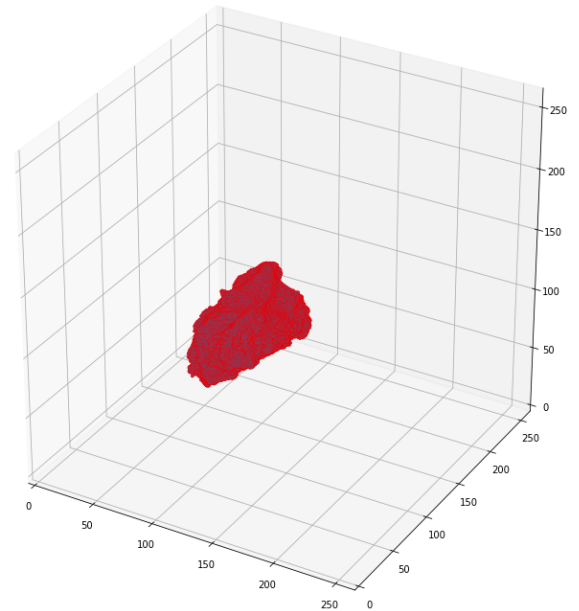


Figure 5.30:  
Ground truth

## 5.6 Sprint5: Graphical User Interface

In this sprint, we designed a graphical user interface using PyQt which is a GUI widgets toolkit. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI library. In this phase, we established the link between the Model components and the View\_Control components. The logical components have been gradually integrated and multiple integration tests have been done to ensure the smooth functioning of the application. Next, we present the interfaces. Each algorithm requires a different input parameters.

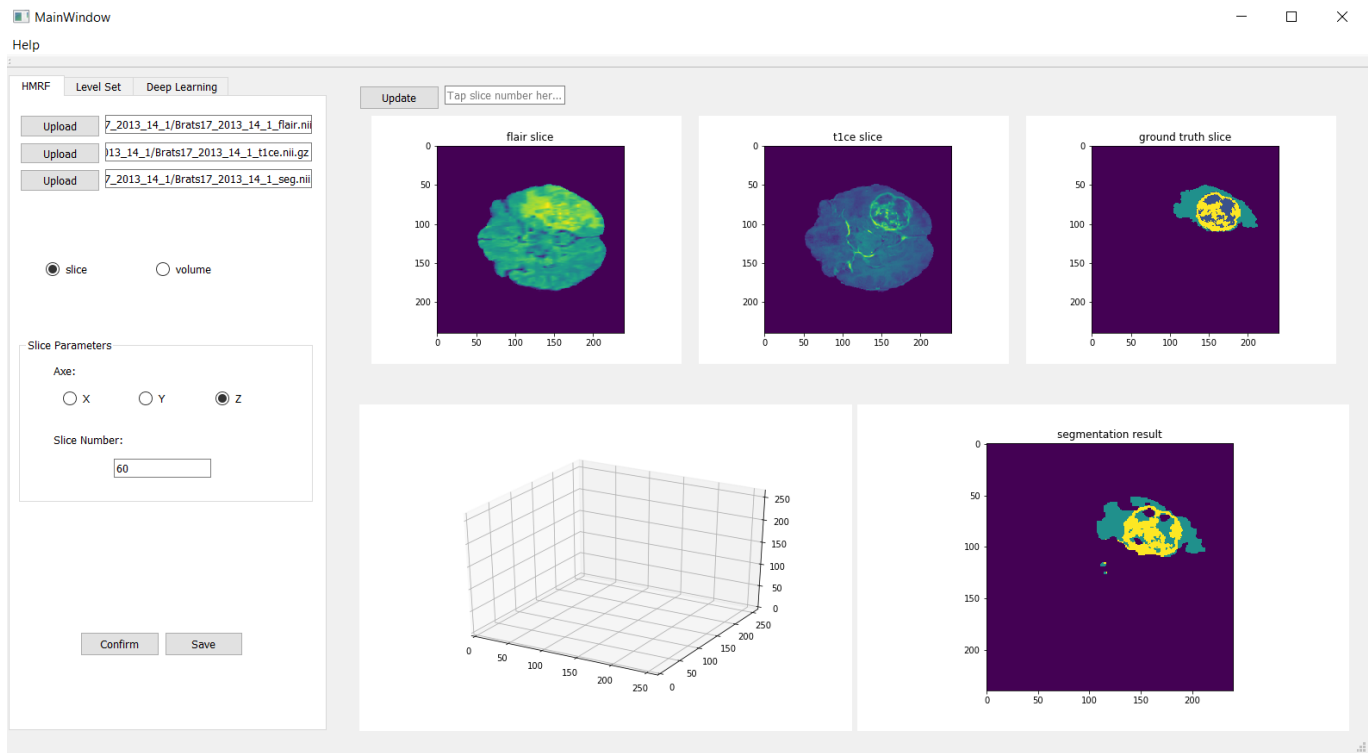


Figure 5.31: HMRF-EM interface

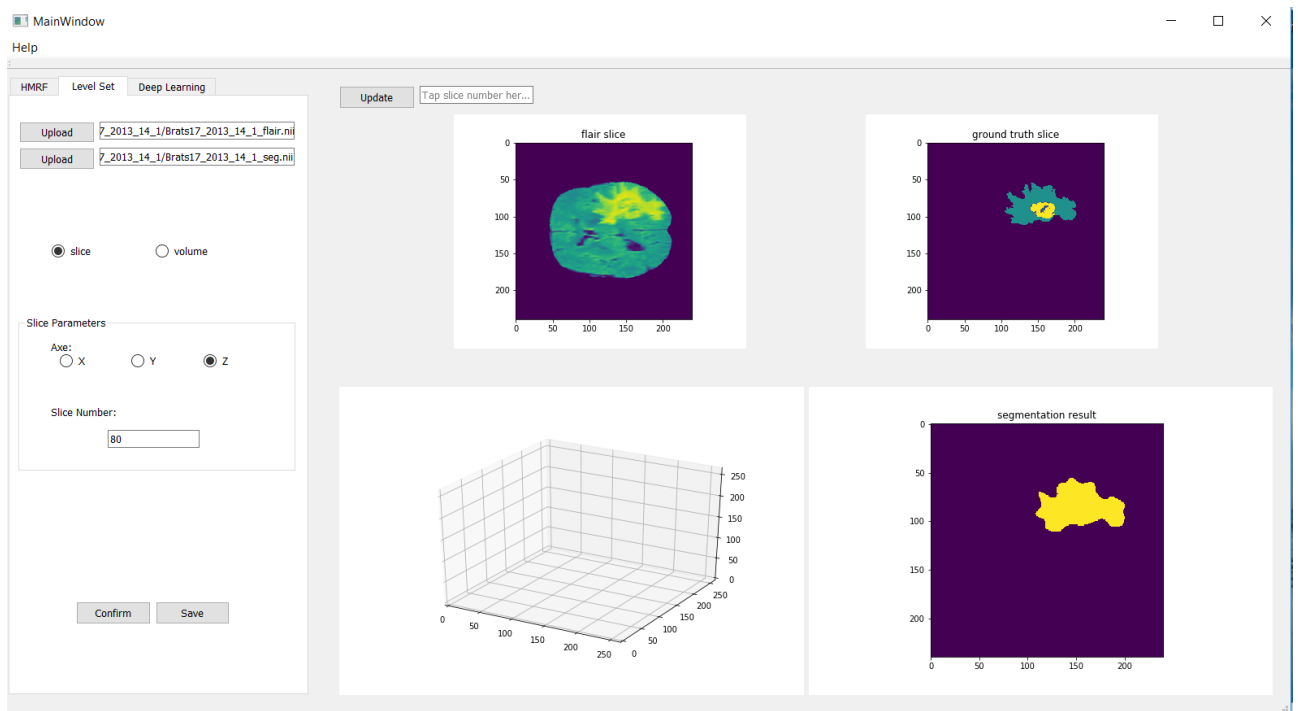


Figure 5.32: Level Set interface

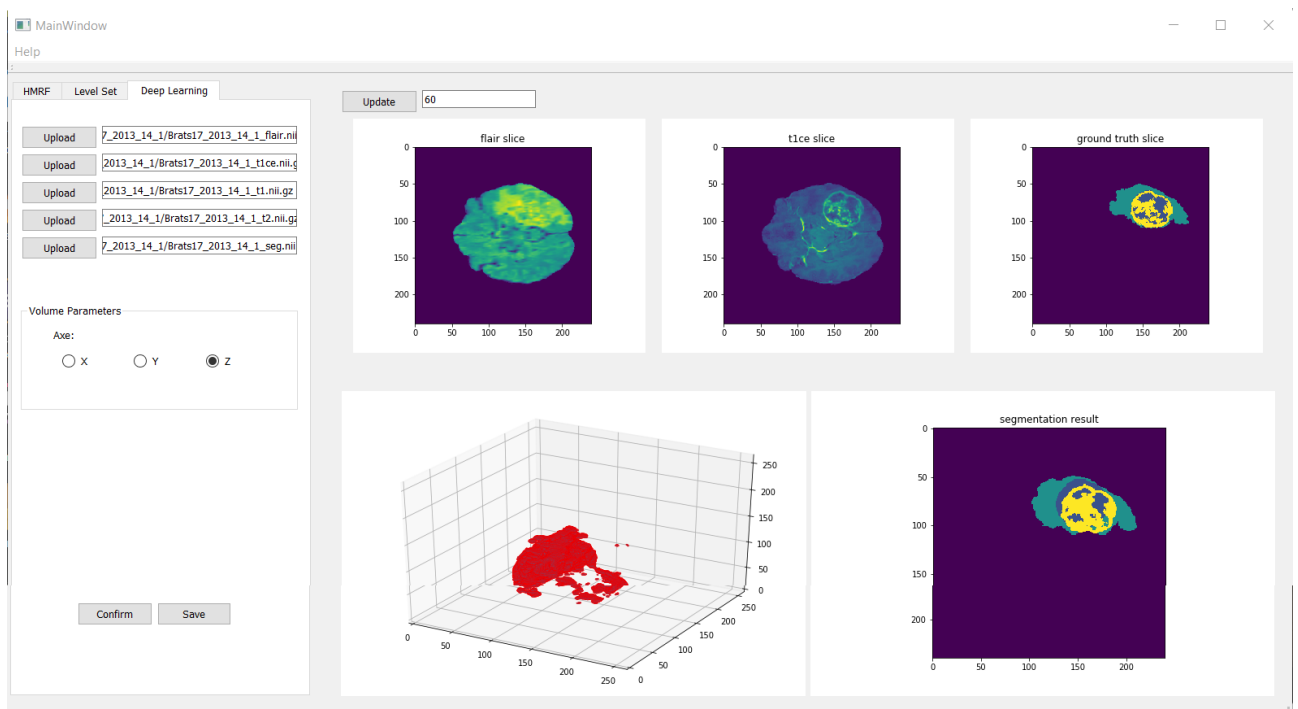


Figure 5.33: Deep Learning interface



## 5.7 Results

To compare the performance of the different algorithms, we used the Dice Score. The Dice score is often used to quantify the performance of image segmentation methods. It is a measure of how similar two images are. If  $A$  is the ground truth image, and  $B$  is the

$$\text{Dice score} = \frac{2 \cdot |A \cap B|}{2 \cdot |A \cap B| + |B \setminus A| + |A \setminus B|}$$

segmentation result of a method then:

- $|A \cap B|$  represents the number of true positives: it is the total number of pixels which have the value 1 in both A and B. So it is the intersection of the regions of ones in A and B. It is the same as using the AND operator on A and B.
- $|B \setminus A|$  represents the number of false positives: it is the number of pixels which appear as 1 in B but zero in A.
- $|A \setminus B|$  represents the number of false negatives: it is the number of pixels which appear as 1 in A but zero in B.

We randomly took 10 slices from different MRI volumes and calculated the mean Dice score.

	HMRf-EM	Level Set	Deep Learning
Whole tumor	0.78	0.71	0.86
Tumor core	0.82	X	0.72
Enhancing tumor	X	X	0.62

Table 5.1: Dice score

## Conclusion

The Deep Learning approach seems to be the most promising approach in the field of brain tumor segmentation. It is able to segment the 3 different sub-regions of the tumor with outstanding results and in a very short time (it takes around 55 seconds to segment a MRI volume consisting of 150 slices, each slice is 240x240 pixels using a Core i7 CPU chip and Nvidia GTX1060 GPU). However, it is not easy to apply this approach, because the data collection and annotation process can be very expensive and challenging. The HMRf-EM is the slowest of the 3 approaches. It showed good results on the whole tumor segmentation task and performed surprisingly well in the task of tumor core segmentation. The cropping might have something to do with this surprising result. One

---

disadvantage of this method is that it is a semi-automatic method that requires the intervention of the user.

The Level Set method performed the worst among the 3 approaches. We could only use it to segment the whole tumor. However, it should be noted that it is a very fast method, and can perform really well in other tasks.

# General Conclusion

The work presented in this report was carried out in the framework of the second year's design and development project at the National School of Computer Science. Throughout the five chapters, we have detailed the various steps that led to the realization of the project. Indeed, in the first chapter, we started with a preliminary study in which we defined the main concepts, set the problem and presented our working methodology. The second chapter was dedicated to the explanation of the algorithms we used. In the third chapter we specified the functional and non functional requirements to have a better understanding of the application we are developing. Later in the fourth chapter, we covered the global design, explaining the physical and logical architecture we followed and then detailed the design using numerous UML diagrams. Finally, in the last chapter, we detailed each sprint we've been through, the different steps of the implementation, the difficulties we encountered and the solutions we came up with.

It is important to mention that building this project was not an easy task. During our work, we faced numerous difficulties but we never doubted that this work will be achieved successfully. This has been our first experience in the domain of computer vision and to put it simply, it was an amazing experience.

As every project can be enhanced and expanded, various perspectives are at hand for our project. We can mention the inclusion of other prominent segmentation methods. Moreover, we probably can turn the HMRF-EM and Level Set algorithms into automatic methods that would not require the intervention of the user by training a neural network model to detect the existence or the non existence of a tumor and another one to make the crop operation automatic. This would result in a hybrid method that would be interesting to see it at work. Last but not least, a more sophisticated 3D model could be generated using improved versions of Marching Cubes or other tools available in the Unity 3D modules.

# Bibliography

- [1] Fischer P. Brox T. Ronneberger, O. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, October 2015.
- [2] Smith S. Zhang Y, Brady M. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Trans Med Imaging*, january 2001.

# Netographie

- [URL1] <http://http://www.sywane.com/methodologie.php/>, accessed 12/02/-2019.
- [URL2] <https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/>, accessed 12/02/2019.
- [URL3] <https://www.degruyter.com/view/j/bams.2015.11.issue-1/bams-2014-0017/bams-2014-0017.xml/> ,accessed 23/02/2019.
- [URL4] <http://www.cs.upc.edu/~virtual/SGI/docs/1..pdf/> ,accessed 24/02/2019.
- [URL5] <https://www.softwaretestingmaterial.com/software-architecture//> ,accessed 01/04/2019.
- [URL6] <https://www.med.upenn.edu/sbia/brats2018/tasks.html/> ,accessed 12/03/2019.

---

**ملخص —** هذا العمل هو جزء من مشروع التصميم والتطوير في المدرسة الوطنية لعلوم الإعلامية ، مشروعنا ، الذي اقترحه الدكتور محمد أمين مزغيش ، يتمثل في إنشاء تطبيق يسمح بتجزئة أورام الدماغ باستخدام ثلاث طرق ثم إعادة إنشاء الصورة في ثلاثي الأبعاد .  
**الكلمات الرئيسية :** تجزئة , إعادة إنشاء الصورة , أورام .

---

**Résumé —** Ce travail s'inscrit dans le cadre du projet de conception et de développement à l'École Nationale des Sciences de l'Informatique.

Notre projet, proposé par Dr.Mohamed Amine Mezghich ,consiste à créer une application bureau qui permet la segmentation de tumeur cérébrale en utilisant trois approches puis reconstruire l'image en 3D.

**Mots clés :** Segmentation , Reconstruction , Tumeur.

---

---

**abstract—** This work is part of the design and development project at the National School of Computer Sciences. Our project, proposed by Dr. Mohamed Amine Mezghich, consists of creating a desktop application that allows the segmentation of brain tumors. using three approaches and then reconstruct the image in 3D.

**Key words :** Segmentation , Reconstruction , Tumor.

---