UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Diseño de un anotador semántico de imagen médica para gliomas asistido por redes neuronales convolucionales.

Trabajo Fin de Máster

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

AUTOR/A: Gil-Terrón Rodríguez, Francisco Javier

Tutor/a: Juan Císcar, Alfonso

Cotutor/a: Fuster i Garcia, Elies

Cotutor/a: García Gómez, Juan Miguel

CURSO ACADÉMICO: 2021/2022

Departamento de sistemas informáticos y computación

Design of a medical image semantic annotator for gliomas assisted by convolutional neural networks

Master's Final Project

Master's Degree in Artificial Intelligence,
Pattern Recognition and Digital Imaging

Author: Francisco Javier Gil-Terrón Rodríguez

Tutor: Alfons Juan Císcar

2º Tutor: Elies Fuster i Garcia

3º Tutor: Juan Miguel García Gómez

2021 - 2022

Resumen

El glioma de alto grado es un tumor del sistema nervioso central no resuelto hasta el momento. El seguimiento de los pacientes que sufren este tumor tras su resección requiere un seguimiento basado en imágenes por resonancia magnética. Para realizar este seguimiento es esencial la información clínica e histológica obtenida de los pacientes, aquella información sobre la estructura y características de los tejidos que permite adquirir conocimiento sobre el estado del paciente.

Aunque la segmentación de imágenes de resonancia magnética antes de cirugía se considera resuelta en el ámbito de la investigación mediante redes neuronales convolucionales, no ocurre los mismo para imágenes posteriores a la cirugía. Es necesario facilitar a los radiólogos una herramienta anotadora que permita generar imágenes de seguimiento segmentadas y anotadas clínica e histológicamente. Además, estas imágenes segmentadas pueden ser usadas para entrenar y adaptar futuros modelos automáticos de segmentación y hacer más eficiente este proceso.

En este proyecto diseñaremos un software de anotación de secuencias de seguimiento de imágenes de resonancia magnética. La segmentación ofrecida estará asistida por redes neuronales convolucionales o segmentaciones previas, permitiendo al radiólogo modificar la máscara de segmentación y añadir nuevos hallazgos según su criterio experto, con lo que se podrá adaptar los modelos de segmentación automática. En el presente estudio se han logrado desarrollar modelos de segmentación con resultados aceptables en estudios clínicos específicos, así como una mejora del error del 10% en los resultados obtenidos por modelos que han sido adaptados frente a los que no.

Así pues, el software permitirá al usuario de forma interactiva la anotación de segmentaciones semánticas y hallazgos en las imágenes, así como asociarlos a información clínica e histológica relevante.

Palabras clave: Reconocimiento de formas, Reconocimiento de imágenes, Aprendizaje automático, Biomedicina, Anotador semántico, Imagen médica, Gliomas, Redes neuronales convolucionales

Abstract

High-grade glioma is an unresolved tumor of the central nervous system. The follow-up of patients with this tumor after its resection requires a monitoring based on magnetic resonance imaging. Clinical and histological information obtained from patients is essential for this follow-up, that is, information about the structure and characteristics of the tissues that allows acquiring knowledge about the patient's condition.

Although the segmentation of pre-surgical magnetic resonance imaging is considered solved in the field of research using convolutional neural networks, the same is not true for post-surgery images. There is a need to provide radiologists with an annotator tool to generate segmented, clinically, and histologically annotated tracking images. Furthermore, these segmented images can be used to train and adapt future automatic segmentation models and make this process more efficient.

In this project we will design an annotation software for magnetic resonance imaging tracking sequences. The segmentation offered will be assisted by convolutional neural networks or segmentation priors, allowing the radiologist to modify the segmentation mask and add new findings according to their expert criteria, thus allowing the automatic segmentation models to be adapted. In the present study, it has been possible to develop segmentation models with acceptable results in specific clinical studies, as well as a 10% error improvement in the results obtained by models that have been adapted compared to those that have not.

Therefore, the software will allow the user to interactively annotate semantic segmentations and findings in the images, as well as to associate them with relevant clinical and histological information.

Keywords: Pattern recognition, Image recognition, Machine learning, Biomedicine, Semantic annotator, Medical imaging, Gliomas, Convolutional Neural Networks

Resum

El glioma d'alt grau és un tumor del sistema nerviós central no resolt fins al moment. El seguiment dels pacients que pateixen aquest tumor després de la seua resecció requereix un seguiment basat en imatges per ressonància magnètica. Per a realitzar aquest seguiment és essencial la informació clínica i histològica obtinguda dels pacients, aquella informació sobre l'estructura i característiques dels teixits que permet adquirir coneixement sobre l'estat del pacient.

Encara que la segmentació d'imatges de ressonància magnètica abans de cirurgia es considera resolta en l'àmbit de la investigació mitjançant xarxes neuronals convolucionals, no ocorre els mateix per a imatges posteriors a la cirurgia. És necessari facilitar als radiòlegs una eina anotadora que permeta generar imatges de seguiment segmentades i anotades clínica i histològicament. A més, aquestes imatges segmentades poden ser usades per a entrenar i adaptar futurs models automàtics de segmentació i fer més eficient aquest procés.

En aquest projecte dissenyarem un programari d'anotació de seqüències de seguiment d'imatges de ressonància magnètica. La segmentació oferida estarà assistida per xarxes neuronals convolucionals o segmentacions prèvies, permetent al radiòleg modificar la màscara de segmentació i afegir noves troballes segons el seu criteri expert, amb el que es podrà adaptar els models de segmentació automàtica. En el present estudi s'han aconseguit desenvolupar models de segmentació amb resultats acceptables en estudis clínics específics, així com una millora de l'error del 10% en els resultats obtinguts per models que han sigut adaptats enfront dels que no.

Així doncs, el programari permetrà a l'usuari de manera interactiva l'anotació de segmentacions semàntiques i troballes en les imatges, així com associar-los a informació clínica i histològica rellevant.

Paraules clau: Reconeixement de formes, Reconeixement d'imatges, Aprenentatge automàtic, Biomedicina, Anotador semàntic, Imatge mèdica, Gliomes, Xarxes neuronals convolucionals

Table of contents

Index of figures

Index of tables

1. Introduction

With the growing technological revolution, the use of artificial intelligence and machine learning techniques is becoming a macrotrend. In this way, their use has come to permeate a large part of the different sectors, being healthcare the one with the greatest expectations for human well-being. It is in this area, the health sector, where the current work is focused.

The healthcare sector benefits from the AI techniques, from automatic writing of cases and clinical reports through speech recognition, to helping healthcare professionals to detect diseases given certain clinical findings or symptoms.

In this case, the area of knowledge to be addressed will revolve around the automatic detection of pathological signs in medical imaging, field in which the future of medical imaging is already linked to artificial intelligence. The main applications currently being carried out in this area are the classification and semantic segmentation of tissues and organs, generally through the use of convolutional neural networks, standard within the field of computer vision. The current work will mainly deal with the segmentation of gliomas, a tumor located in the brain area.

The segmentation of this type of tumors will consider the different tissue states belonging to the tumor and its surrounding areas, i.e., it is a multi-class segmentation task.

In the present work we will try to develop models to perform this segmentation in an automatic way. In addition, these models will be incorporated directly into the annotation software and will allow their adaptation and specialization according to the user's needs.

1.1 Motivation

During the management of glioma tumors there are a multitude of critical steps such as surgical treatment planning, image-guided interventions, or tumor growth monitoring. These steps may benefit from accurate identification of the boundaries of brain tumor regions on minimally invasive to non-invasive medical imaging.

However, manual segmentation of lesions and their associated pathological tissues is an arduous and error-prone task. Due to the time required for this task even the most experienced professional is not exempt from making mistakes. This problem increases if we take into account that, in the case of gliomas, for a correct interpretation, several magnetic resonance imaging sequences must be analyzed in parallel. Moreover, in this topic it is common to generate longitudinal series of images for patient follow-up, so that each patient will be studied at different time points. Taken together, these facts make manual segmentation and volumetric studies of glioma an expensive, laborious, and often inaccessible task for humans that could take up to hours. This highlights the need for automated segmentation tools that can help facilitate this process.

Currently, the task of automatic glioma segmentation has already been addressed with machine learning techniques by creating automatic segmentation models. However, most of these models are not integrated into annotation platforms, so they are mostly not accessible to radiologists.

On the other hand, almost all of these models, in turn, have been developed to segment pre-surgical images, so their performance when applied to post-surgical cases drop considerably. This is mainly due to the fact that in the post-surgical context there is a great variation of the tumor over time. For this reason, it is necessary to study longitudinal series of medical images, i.e., it is necessary to analyze images of the patient at different time stages.

Moreover, even in those limited cases where models are included in annotation platforms, they are static models that do not take advantage of the radiologists' expert knowledge to improve.

Taking into account the aforementioned problems, we will seek to develop software that automates this segmentation process, in addition to working on a system with which the target user is already familiar and allows progressive learning with the feedback provided by the user.

1.2 Objectives

The main objective of the study is to provide an effective and usable framework for healthcare professionals, and more specifically radiologists, to simplify and streamline their tasks of medical image segmentation of post-surgical cases of high-grade gliomas.

From this objective, the following sub-objectives can be highlighted:

1. To create automatic segmentation models that streamline the segmentation process of the high-grade glioma pathological tissues, offering acceptable results for healthcare professionals.
2. Develop a system capable of adapting the segmentation models to post-surgical cases based on the radiologists' expert knowledge, taking advantage of previous segmentations for continuous improvement of the models.
3. To provide the research radiologist with a familiar and usable interactive environment that allows easy segmentation of post-surgical images.

1.3 Structure

After this introductory chapter, section 2 will provide the necessary information for a correct understanding of the work. For this purpose, a brief description of gliomas and Magnetic Resonance Imaging (MRI) will be given. It will also emphasize the state of the art of semantic segmentation and contextualize the project that is being performed with the current machine learning technologies used in this field.

In section 3 a survey of semantic image segmentation platforms will be made including their comparison and evaluation to find the one that best suits the requirements of our application. In addition, it will be explained with which technology the proposed solution will be integrated in the finally chosen platform.

Chapter 4 will then explain the methodology followed to obtain the segmentation models, including, among others, architecture, and training strategy, as well as the model adaptation strategy, which aims to fit a model to post-surgical cases.

Subsequently, section 5 will specify how the previous models will be integrated into the annotation platform specified in section 3 and will explain the workflow to be followed by the user when using our tool.

Chapter 6 will evaluate the performance of the system created, considering the results of the models and the adaptation of these models to other areas. In addition, feedback from an expert in the sector is provided for the validation of the tool.

Finally, chapters 7 and 8 present the conclusions of the project and allude to the future lines of development opened up by the work.

In addition, the reference to a short video with a demonstration of the software in operation will be included in the appendices at the end of the thesis.

2. Preliminaries

This chapter will introduce the basics of the knowledge area covered by this work and give the necessary background.

Chapter 2.1 will explain what a glioma is and its clinical outcomes, chapter 2.2 will introduce how they are obtained, and which will be the main magnetic resonance images to be used. Part 2.3 will briefly explain what machine learning is and finally, the current state of the art in semantic segmentation of medical images will be presented in section 2.4.

2.1 Brain tumors and gliomas

The human nervous system is anatomically and physiologically divided into the central nervous system and the peripheral nervous system. The brain, the organ on which we will focus, belongs to the first one [1]. This system is mainly composed of glial cells and neurons. Among the glial cells we could highlight astrocytes, which are the most abundant type of cells and perform a multitude of functions such as maintaining the pH of the central nervous system and transporting nutrients to the neurons., or oligodendrocytes, which are responsible for the protection of neuronal axons [2].

Gliomas are a type of tumor that come from within the central nervous system and represent one of the most common types of brain tumors, especially in adult patients. Gliomas originate from cells of the glial lineage and have varying degrees of severity.

The clinical classification of tumors has been in continuous evolution, since gliomas include a very wide-ranging group of neoplasms. The most accepted categorization is based on the classification of the biological behavior and histological aspect of the different brain tumors [3]. This grading divides brain tumors on a scale of 1 to 4 (I-IV) according to the malignancy, aggressiveness, and reproducibility of the tumor.

Because there are a multitude of biologically and structurally distinct tumors, slow-growing tumors, i.e., those labeled as Grade I or II, are commonly referred to as low grade gliomas (LGG), while those that are more aggressive (Grade III or IV) are considered high-grade gliomas (HGG).

High-grade gliomas are the most aggressive primary neoplasms of the central nervous system, although due to the heterogeneity of gliomas, some low-grade tumors, specifically grade II, can be as aggressive as other grade IV tumors [4].

Because of this heterogeneity, correctly studying both the extent of the tumor and the heterogeneity itself is crucial to make a correct diagnosis, analyze the response to treatment and monitor disease progression. This is done through the study of longitudinal series of the patient, i.e., from different points in time, which is the key point where the tool to be developed is focused.

Among gliomas, grade II and III astrocytic tumors and grade IV glioblastomas can be considered, being mainly the type of tumors that populate our dataset. Despite the fact that astrocytic tumors have similarities with glioblastomas, both have a high degree of heterogeneity in appearance, shape, and histology, although glioblastomas are more aggressive.

Finally, it is important to note the implications that these tumors have on the life expectancy of patients. Survival for patients with grade II or III astrocytic tumors varies significantly by histopathology. Grade II and III astrocytic tumors represent one of the most difficult-to-treat pathologies due to their propensity recur after initial treatment. Patients with grade II astrocytic tumors have a median survival ranging between 4.6 and 6.5 years, and a median time to malignant progression of 5–11.4 years. In turn, grade III astrocytic tumors have a 41-month median survival [5].

On the other hand, glioblastoma stands out as a lethal cancer that lacks satisfactory therapy. Patients with glioblastoma have an overall survival from 12-14 months [6], in the case of following standard treatment (which includes surgical resection followed by radiotherapy and chemotherapy), and 4 months in the opposite case. Even though efforts have been made in recent years to develop new and personalized treatments, no major differences have been noted in the course of patients.

2.2 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) has become a standard for the diagnosis of many neoplasms since it allows to obtain relevant information that could not be obtained in any other way without aggressive interventions on the patient. Despite this, interpreting the information these images facilitate may be a complicated task.

In general, for the purpose of Magnetic Resonance Imaging, the nuclear resonance of hydrogen nuclei is commonly used due to their high abundance in the human body. To create an MRI signal, energy is transferred by means of a radio frequency pulse perpendicular to the magnetic moment of the hydrogen nuclei [7].

After applying the energy, the nuclei return to the equilibrium position causing an event called relaxation, that can be observed in the T1 and T2 forms.

The first, T1, indicates longitudinal magnetization recovery due to the nuclei returning to the lower energy state by transferring energy to the environment. On the other hand, T2 concerns the transverse magnetization loss. In this case, this occurs because of the interactions between nuclei which results in the nuclei being subjected to different local magnetic fields.

In turn, different precession frequencies can be obtained by applying gradients along each orthogonal direction of the scanner. This results in a spatial variation of the magnetic field that allows the signal contribution of each voxel in the image to be separated.

In this way, varying both the gradients and the way and time in which radio frequency pulse sequences are applied leads to different image sequences. Thanks to this, different contrasts between tissues can be produces, for example, to tissues such as fat or water, making them appear darker in the images [8].

Thus, images with T1- or T2-dependent features are called T1-weighted (T1w) and T2-weighted (T2w) respectively, although in this work, we will be referred to them as T1 and T2 for simplicity.

In addition to these two types of images, the images used in this work also include two other magnetic resonance images used in the field of glioma detection: The fluid-attenuated inversion recovery (FLAIR) can be used in brain imaging to suppress the effects of cerebrospinal fluid and minimizes contrast between gray and white matter. This procedure produces strong T2-weighted images that highlight periventricular hyperintense lesions [9].

The other images, being the last type of image to be addressed in this work, is called gadolinium-enhanced T1-weighted (T1gd), which, as its name indicates, are essentially T1 images extracted from a patient who has been administered a contrast substance (i.e., Gadolinium) that concentrates mainly in the regions of the active tumor.

Figure 1 shows the visual differences between the types of images mentioned above, where the three main images that will be used for the purpose of segmenting gliomas are shown: T1gd, T2 and FLAIR.

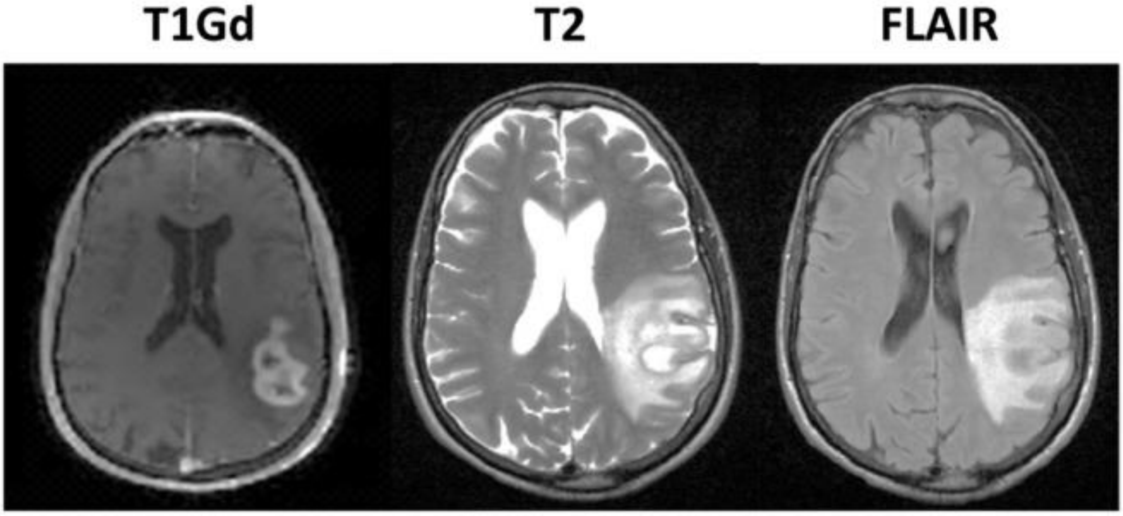


Figure 1 - Types of magnetic resonance imaging. Source: [10]

2.3 Machine learning

Machine learning is the discipline of artificial intelligence that involves the development of systems or applications capable of approaching solutions to specific tasks based on data or past experience [11].

Depending on the type of data used to train the model, machine learning can be divided mainly into:

- Supervised learning: Supervised learning is the most common technique and is based on a set of data $x \in \mathcal{X}$, which are the inputs, and a set of data $y \in \mathcal{Y}$, which are the outputs, also called labels or targets. Together they form what is called training set $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ and the performance is measured as a function of the output y that is predicted from the input x.
- Unsupervised learning: In the case of unsupervised learning there is only a set of inputs $\mathcal{D} = \{x_n : n = 1 : N\}$ without any observed output, therefore the model will estimate the unconditional distribution $p(x)$. Due to the absence of output, it avoids the need to label the data, which is usually a costly task.
- Reinforcement learning: In reinforcement learning, the system (also called agent) has to learn how to interact with a complex environment according to a defined policy, which describes the actions to take in response to each possible input. In this way, the policy will be modified

by the reward or punishment received as a function of the actions taken.

In this work we will focus on segmentation of medical images. Segmentation tasks are mainly of the supervised type, and more specifically, they are classification tasks. A classification problem consists of predicting the class or label given an input from among a set of mutually exclusive classes $\mathcal{Y} = \{1,2 \dots, C\}$. Due to the nature of the segmentation proposed in this work, which is semantic segmentation, this classification will be done at the pixel level, or at the voxel level transferred to the three-dimensional domain.

At this point, it is worth to mention transfer learning, a type of learning that is characterized by being based on re-training with little data. Transfer learning consists of taking advantage of the structural similarities of data-rich tasks to extend the learning to similar data-poor tasks. In this way, a model can be adapted with a fine-tuning process, or domain adaptation if the data are not sufficiently similar.

In this case, transfer learning will be carried out to adapt generic models trained with cases of many patients to a specific one of a patient in order to be used in longitudinal sequences of the same patient.

2.4 State of the art in semantic segmentation

2.4.1 Semantic segmentation

In recent years, the use of machine learning has become a cross-cutting practice across all industries and tasks in complex data-related scenarios. This is mainly due to the rise of deep learning, which will be discussed in more detail in the next section. Deep learning is a category of machine learning that is achieving greater results with respect to classical methods in a wide variety of areas, including the healthcare sector, solving computer vision, natural language processing and speech recognition tasks, etc. In this context, semantic segmentation is no exception; in fact, deep learning almost entirely covers the state of the art for this task today.

As already mentioned, semantic segmentation consists, in essence, in trying to classify each of the pixels or voxels of an image according to its semantic information from a set of classes. The goal of this task is to distinguish different areas or regions of interest in an image in order to make it more understandable for further analysis [12] [13].

With the increasing intensification in the use of technologies, more and more applications require this segmentation, for example, biometric access technologies. In medical image analysis, semantic segmentation has positioned itself as a valuable practice for delimiting structures and tissues among other possible regions of interest.

Thanks to deep learning technologies, it has been possible to robustly segment images in the biomedical context, and more specifically, in the field of semantic segmentation of gliomas [14] [15], it has allowed the development of models capable of automatically and accurately classifying glioma tissues. Moreover, deep learning has in turn permeated radiology (and medicine in general), automating time-consuming tasks such as segmentation in 3D images and allowing guided interventions to be carried out. This has made it possible to speed up the work of healthcare personnel and reduce their intervention, which has led to improved patient monitoring, care, and diagnosis [16].

2.4.2 Deep learning and artificial neural networks

Deep learning is a specialization of machine learning that creates systems capable of learning features directly from data that follow an approach in which input data is continuously used to extend existing model knowledge that does not require hard-coded features or domain expertise. Considering the increase in computational power and the amount of data, this paradigm is generating models with exceptional performances. In this context, the use of this technology has positioned itself as the main solution for different areas of medical science, including the task of semantic segmentation.

Deep learning systems are based on the use of artificial neural networks with a multitude of layers bio-inspired in neural operations. These layers will be composed of operators called neurons, and the name deep learning is given due to the high number of layers used create functions of greater representation power.

The simplest case of these networks is called a multilayer perceptron, a case in which the neurons of one layer are all connected to those of the next layer and, as its name indicates, form a perceptron stack. In other words, each neuron will be an operator which applies a linear transformation to obtain a prediction $y = X \cdot W + b$ such that *X* is the input and *W* and *b* the parameters to be optimized.

During training, the performance of the model should be measured, i.e., how well the model has predicted the classes from the training inputs. For this purpose, a loss function reflecting this value is used. The most common loss function being Cross-Entropy, which measures the distance between the predicted labels and the actual labels or ground truth. The Cross-Entropy function is defined as $H(p,q) = -\sum_{y} p(y)\ log\ q(y)$ where p is the true class distribution and q is the predicted class distribution.

The training objective will be to optimize this loss function by iteratively adjusting the network weights by backpropagating a corrective error signal through the network [17]. This backpropagation of the error updates the network weights and is obtained by applying the chain rule to the network outputs against the parameters of each stage to obtain gradients in an efficient way.

During this process, first the ground truth is compared with the classification obtained by the network to calculate the gradients. Then the gradient is passed to an optimizing algorithm to optimize the training objective, where the standard approach is to use maximum likelihood estimation, by minimizing the negative log-likelihood. Finally, the weights of the network are updated for the next iteration through a parameter called learning ratio that determines the variation of the update.

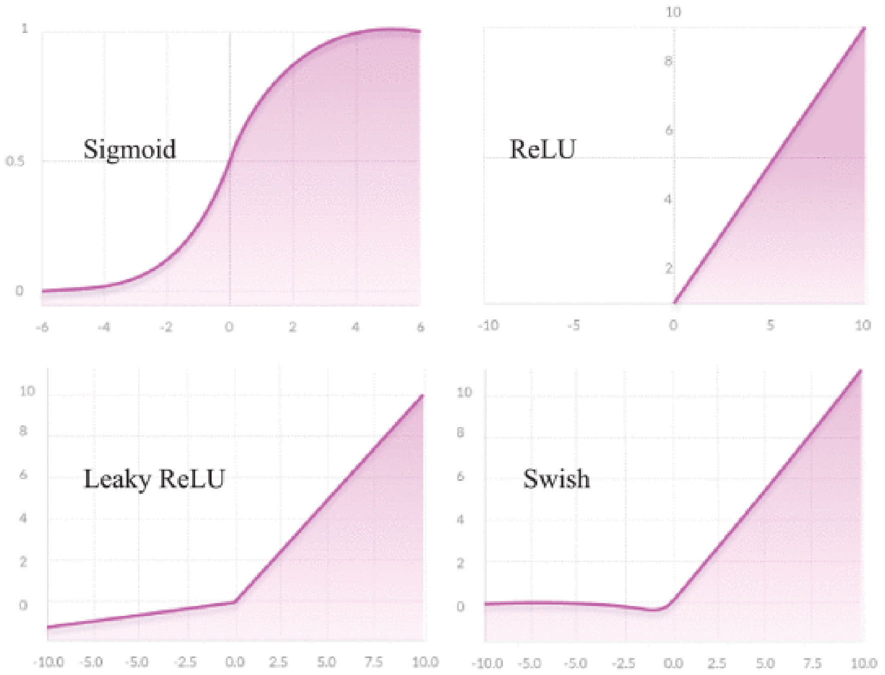


Figure 2 - Example of activation functions. Source: [18]

However, the perceptron is a deterministic version of logistic regression whose decision boundaries are linear, and therefore, so will be the decision boundaries of a neural network as defined above. Because of this linearity, its representational capacity is limited.

To deal with this limitation, so-called activation functions are used in each neuron, which produces non-linear transformations to the outputs of the neuron. These transformations allow weighting the importance of the input of each neuron for the prediction of the model and must be computationally efficient since they are computed across millions of neurons and require the calculation of the derivative of the function.

Some cases of these functions are shown in Figure 2. For example, the sigmoid activation function, which ranges from 0 to 1, is often suitable for models that predict probability as output.

However, these functions are computationally expensive and suffer from the vanishing gradient problem for very high or low input values since the derivative of the function in those regimes is close to 0 and therefore any gradient signal from higher layers will not be able to propagate back error to the previous layers. This vanishing gradient problem makes the network very insensitive to inputs in this range, making it difficult to train the model with gradient descent. In general, when training very deep models, the gradient tends to be very small, because the error signal passes through a series of layers that progressively decrease it. Non-saturating activation functions such as the ReLu function are used to solve this problem.

In addition to activation functions, there is another technique called batch normalization that improves the training of deep neural networks. Batch normalization is a widely adopted technique that allows faster and more stable training of deep neural networks. This technique makes the optimization landscape significantly smoother. This smoothness induces a more predictive and stable behavior of the gradients, allowing faster training [19].

Broadly speaking, Batch normalization is a mechanism that aims to stabilize the distribution (over a mini-batch) of inputs to a given network layer during training. This is achieved by augmenting the network with additional layers that set the first two moments (mean and variance) of the distribution of each activation to be zero and one respectively. Then, the batch normalized inputs are also typically scaled and shifted based on trainable parameters to preserve model expressivity. This normalization is applied before the non-linearity of the previous layer.

2.4.3 Convolutional neural networks

Having introduced what a neural network is and how it behaves, it should be noted that there are cases in which, due to the nature of the problem, in order to obtain adequate performance, it is necessary to take into account the contextual information of the values beyond the value itself. This is the case of working with images, where to obtain a correct representation, it is necessary to consider the spatial information provided by the values near each pixel or voxel.

To achieve this, a new type of operation is added to the network: convolutions, which is a fundamental building block and give name to convolutional neural networks. In the case of multilayer perceptron, each neuron in one layer is connected to all neurons in the following layers, which made the network susceptible to parameter overfitting. In contrast, convolutional neural networks can account for local connectivity. This alternative to the multilayer perceptron has proven effective, especially in computer vision tasks, where convolutional layer stacking has proven very efficient.

Convolution is a linear operation that consists of multiplying the weights with the input, similar to a traditional neural network. The convolution is performed between the input data and a matrix of weights, called a filter or mask, which is smaller than that of the inputs. In this way, an elementwise multiplication is performed between the filter and the input part of the filter size, resulting in a single value. The essential advantages of this type of network are that the filters do not need to be created by hand as their weights can be determined automatically by backpropagation training.

Although convolutional neural networks were originally devised to be applied to two-dimensional images, they can also be used on one-dimensional or even three-dimensional data. Figure 3 shows visually how the output of a convolution would be obtained in a two-dimensional space given an input and a mask or filter.

This operation is repeated systematically throughout the input image, resulting in an output matrix called a feature map. These feature maps allow the algorithm to discover hidden features and to express complex representations by simpler representations. This is because convolutions reduce the spatial dimensions of the images while increasing the depth of the images, while roughly maintaining the semantic complexity of their representation. In other words, convolutional neural networks use the hierarchical pattern of the data and assemble patterns at different special scale using smaller, simpler patterns.

Overall, the deeper layers of the model learn higher-order features, such as shapes, while the layers near the input learn low-level features, such as lines and edges.

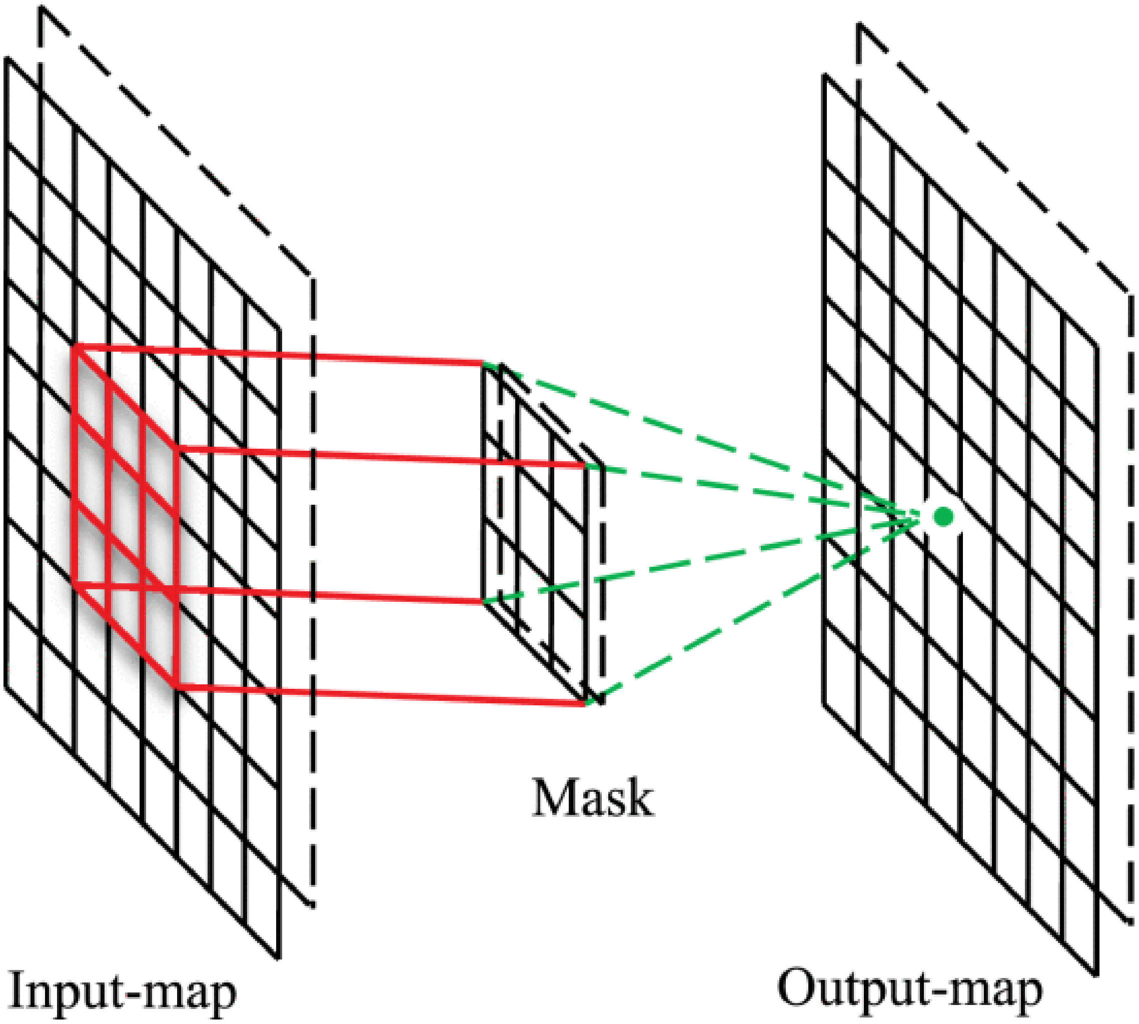


Figure 3 - Convolution operator. Source: [18]

2.4.4 U-net convolutional network structures

With what has been presented so far, the base architecture for semantic segmentation would be that of a fully convolutional neural network. This type of network creates by convolutions a final output in the form of a pixel or voxel classification map that corresponds one by one with those of the input, which can be seen in Figure 4. This is achieved by applying transposed convolutions or deconvolutions to the final feature map obtained through convolutions, so that the dimensions of the output match those of the input. This type of network is able to learn global and local features and contextual representations that can be used for semantic output prediction [20].

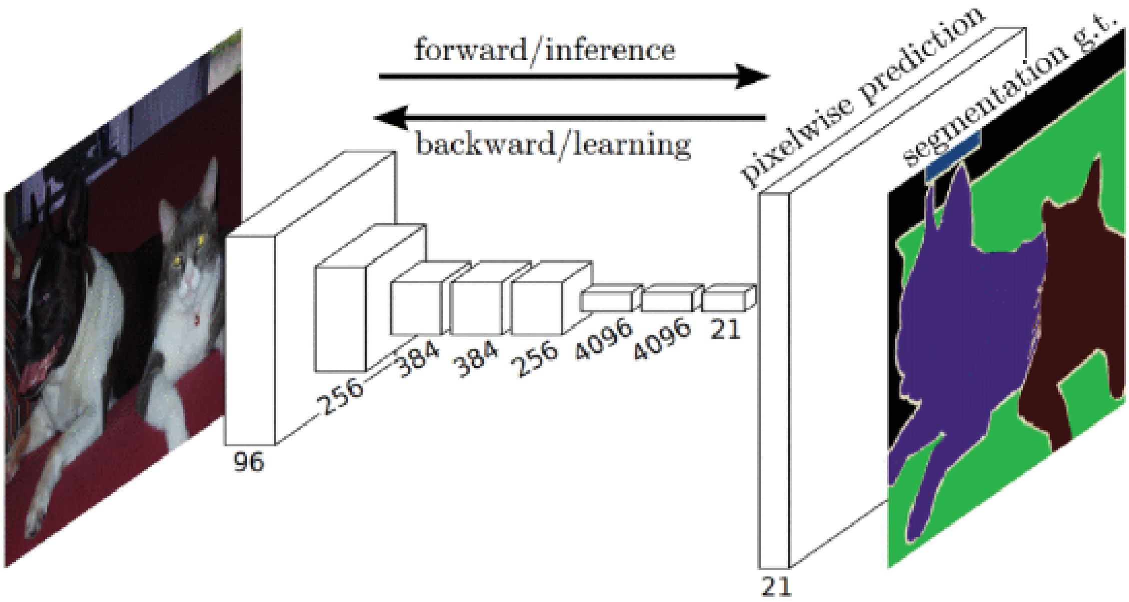


Figure 4 - Fully convolutional neural network. Source: [18]

Fully convolutional neural architectures have been used specifically for medical image segmentation, where they have obtained considerably satisfactory results.

Although, the main advantage of these networks is that they provide a comprehensive solution for semantic segmentation, they have a number of drawbacks that make them unsuitable for all tasks. Despite their success, the locality of the convolutional layers in these networks limits the ability to learn long-range spatial dependencies, in addition to their high computational cost and difficulty in adapting to three-dimensional images.

For this reason, a wide range of deep network architectures have been proposed for medical image segmentation, although it is true that it is the aforementioned Fully Convolutional Network and the U-Net that have revolutionized the current paradigm and have provided a path for subsequent models.

In the following, we will slightly deepen on the U-Net, networks that are widely used in image segmentation. U-Net currently represent the most prominent medical image segmentation model, although they are also used in other fields such as natural language processing or machine translation. This type of network is the one that has been used to create the models of this work, so in its respective section, section 4, this type of architectures is discussed in greater detail.

U-Nets are characterized as a type of model formed by convolutional layers with an encoder-decoder structure, i.e., in two stages. In essence, there will be in parallel a path or sequence of encoder layers and a sequence of decoder layers, as shown in Figure 5.

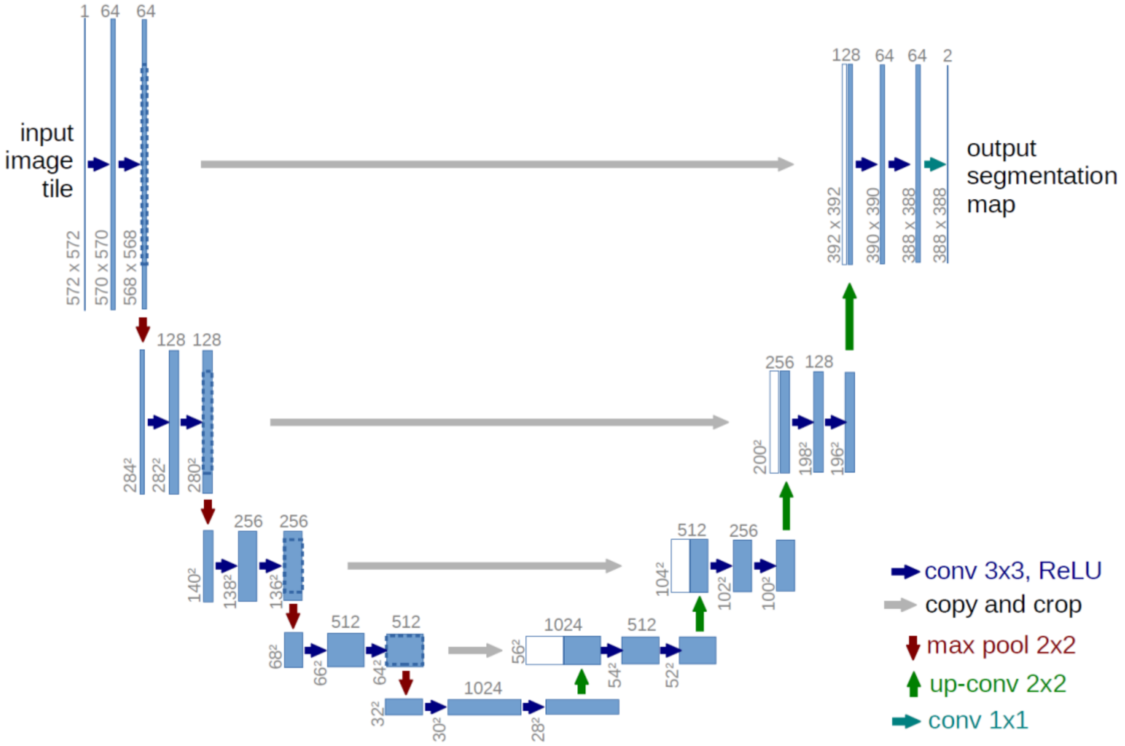


Figure 5 - U-Net architecture. Source: [21]

The encoding path is a convolution stack in which the input dimensions are reduced by the down-sampling operators. This helps to find information by exploring advanced features but, at the same time, causes a reduction in the size of the feature map with low resolution outputs.

On the other hand, the symmetric decoding path uses transposed convolutions to perform accurate localization. At this point, the concatenation of feature maps associated with coding-decoding units of the same level is performed, so that it maps the coded state to an output sequence that ends up being a feature map with the same dimensions as the input (up-sampling). These connections between different sections of a network where the output of one layer is concatenated with another later in the network is called residual connection and allows to improve the gradient flow avoiding its loss along the training.

Although it is true that the U-Net provides a number of advantages over fully convolutional neural networks, without going any further, which is much more suitable for working in the three-dimensional domain, its training is usually dependent on the amount of data. Since in the medical field data is not usually an abundant resource, it will be important to resort to the artificial creation of samples from the available data, i.e., to apply the data augmentation technique correctly.

2.4.5 Metrics

To measure performance in segmentation tasks there are different metrics and there is currently some discussion about which ones best reflect segmentation results. Despite this, the most widely used and extended metric in the field of AI segmentation is the Dice coefficient or Sorensen-Dice index, a metric that will be used in this work to measure segmentation performance. The Dice is a statistical tool that measures the similarity between two datasets [22] and is the one used in BraTS benchmarks, which will be discussed in future sections.

This coefficient has the following formula where A and B are two sets:

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

In the scope of this work, automatic segmentation based on AI, to measure the results of a machine learning model, the ground truth and the output obtained by the model, which would be A and B respectively, are usually used as the two sets.

3. Annotation platforms

Image annotation consists, in essence, of assigning a series of labels to the image so that they reflect information displayed in the image. In the case of glioma annotation, the type of annotation (or segmentation) is semantic, i.e., the annotation is done on each of the pixels in the image. Thus, there will be a different label for each of the classes, in this case, the different structures that make up the glioma, and an additional one for the background or healthy tissue, so that each of the pixels of the final images will be annotated with one class or another.

3.1 Tool analysis

There are currently a large number of annotation platforms and tools focused in the area of computer vision, ranging from free of charge based on open-source development to pay-per-use applications.

There are additionally platforms with a generalist approach and those focused on specific fields, including medical imaging. Due to this variety of possibilities, there is no single platform that prevails in terms of annotation. However, some of these tools do stand out in terms of the number of users who use them both from a generalist and specific point of view.

In this work, a study of the main available platforms has been performed. These platforms will be evaluated in order to analyze which one best suit the needs of the project and use it as a reference.

It should be noted that the characteristics sought in the tool are, mainly, the following:

- The possibility of local execution avoiding the use of the cloud or external servers for clinical security reasons.
- Compatibility between different operating systems.
- Read medical images preferably in NIfTI and DICOM format.
- Visualize multiple channels simultaneously.
- Preferably be open-source software.

In addition, user orientation, flexibility, and the ability to integrate with other systems will be positively valued.

3.1.1 LabelImg

LabelImg is a native annotation platform compatible with any operating system written in Python and easy to install, from pip, docker or from source code. It has a simple and intuitive interface even for those less versed in the subject, although aesthetically it is outdated. It is open-source, multi-platform and works necessarily in local execution.

Since it is generalist in scope (see figure 6), it initially works only with standard image formats, but working locally it will be more efficient and safer than browser-based approaches. On the other hand, it is too simple for our needs, since it only allows us to open one or several standard images in a directory, annotate them, and export the annotations, besides having only two annotation formats: PascalVOC and YOLO.

It does not allow you to create projects or tasks or manage users. In addition, it does not have the functionality to segment at pixel level (semantic annotation), nor does it have segmentation automation and integration with machine learning techniques. In view of the above, although it is one of the most popular tools in image annotation, it is the option that is the furthest away depending on the project requirements.

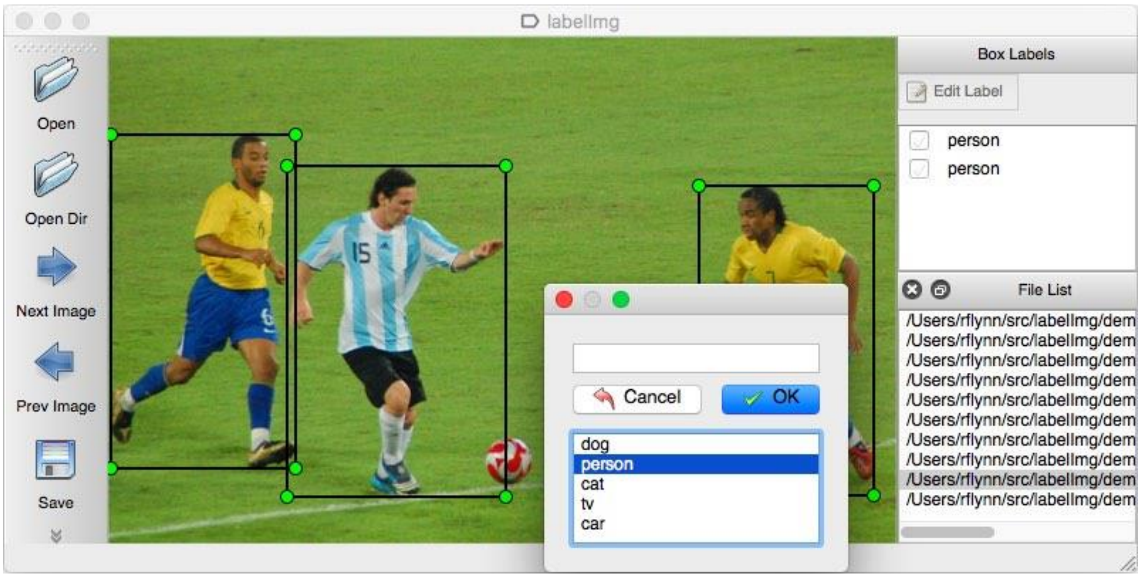

labelImg
Open
Open Dir
Next Image
Prev Image
Save
Box Labels
Edit Label
person
person
File List
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
Cancel
OK
dog
person
cat
tv
car

Figure 6 - LabelImg. Source: [23]

3.1.2 Computer Vision Annotation Tool (CVAT)

Computer Vision Annotation Tool is an alternative to LabelImg, also of generalist scope, although web-based, would be the natural evolution of the previous one in terms of functionalities since it is much more complete. Like the previous one, it is open-source, multi-platform and since it works on docker, it could be used locally or in a distributed way.

Like the previous one, it allows to annotate one or several images simultaneously and to import and export annotations, although in this case it offers many more annotation formats and functionalities, such as semantic annotation, creation of projects and subtasks with common classes depending on the project or user management. Its installation is simple and compatible with any operating system, although it requires docker for its operation and the fact that it runs over a browser could lead to problems in the medium and long term.

The interface is one of the most aesthetic found in this study, and although it is simple, it might not be very intuitive at the beginning due to the functionalities it offers. As in the previous case, it works with classic image formats, although it offers a DICOM to PNG format converter to be able to work with medical images.

This software is one step closer to what we are looking for in this work, although the absence of multichannel and the fact that the image display is so different from the usual one in the medical case renders it not suitable for our purpose.

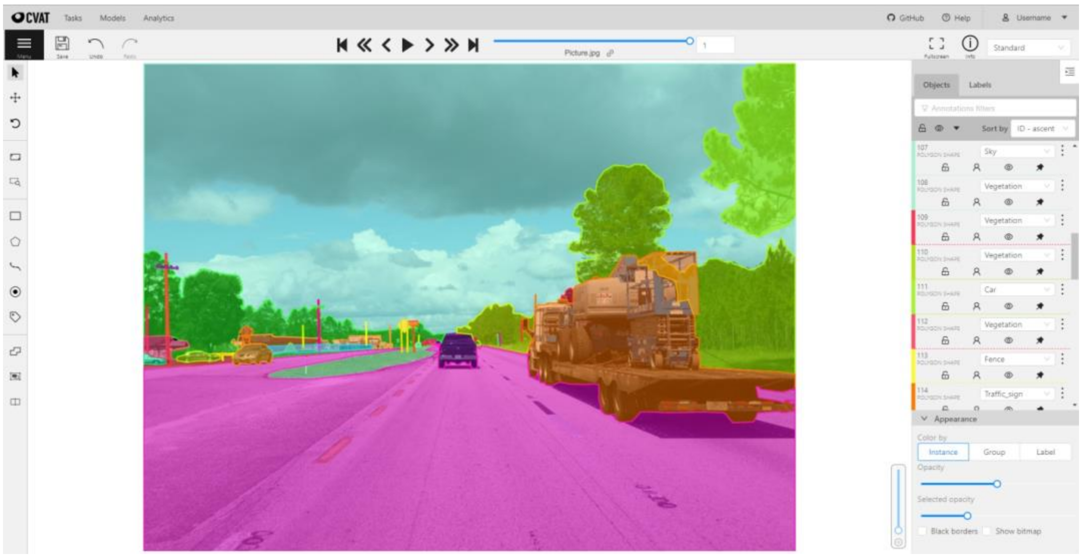


Figure 7 - Computer Vision Annotation Tool. Source: [24]

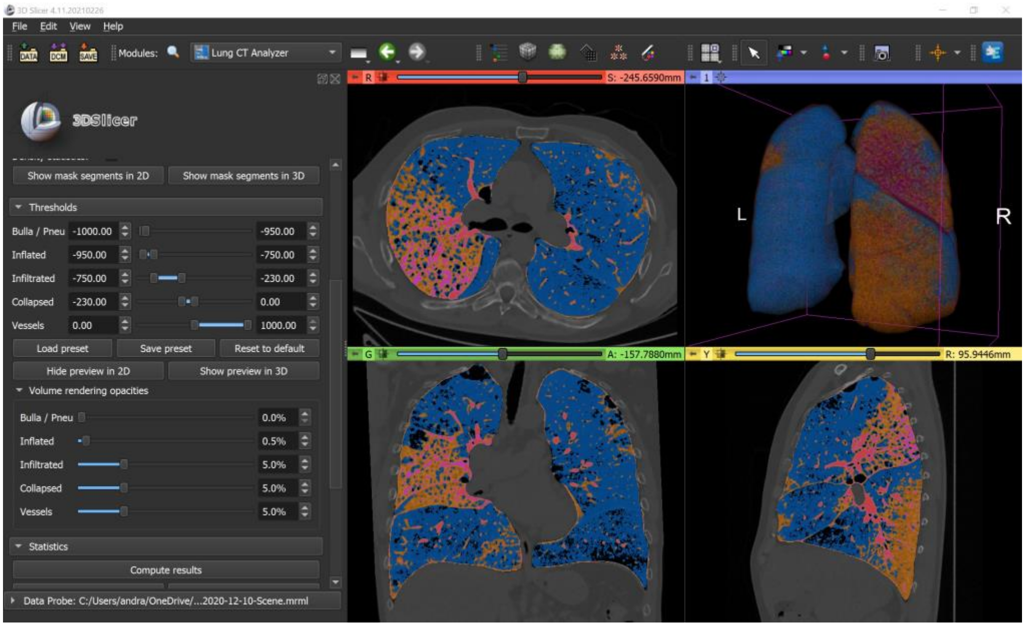
3.1.3 3D-Slicer

3D-Slicer is a tool already focused on the clinical and biomedical field and widespread in radiological community, which allows, among other features, to annotate medical images in their original format: DICOM and NIfTI.

The main advantage of this option is that it has a plug-in marketplace that would allow only the necessary additional features to be developed as a plugin on this already functional tool, although it would be somewhat of an entry barrier for some users, since it would require the use of this software. It is multi-platform and although it is not open-source, it would not be a problem when developing a plugin. Such a plugin could be hosted either distributed or local.

On the other hand, the main disadvantage of this software is that it is not exclusively focused on the segmentation task and has a wide variety of functionalities. This could be confusing for those users who are not familiar with 3D-Slicer, although it is true that, compared to similar alternatives, it has a fairly aesthetic interface.

Despite being an established and popular tool in the field of radiology, it has not been considered as a main option in order to find a more flexible one for the end user.



Modules: Lung CT Analyzer

Thresholds

Bulla / Pneu	-1000.00	-950.00
Inflated	-950.00	-750.00
Infiltrated	-750.00	-230.00
Collapsed	-230.00	0.00
Vessels	0.00	1000.00

Volume rendering opacities

Bulla / Pneu	0.0%
Inflated	0.5%
Infiltrated	5.0%
Collapsed	5.0%
Vessels	5.0%

Statistics

Data Probe: C:/Users/andra/OneDrive/...2020-12-10-Scene.mrml

S: -245.6590mm

A: -157.7880mm

R: 95.9446mm

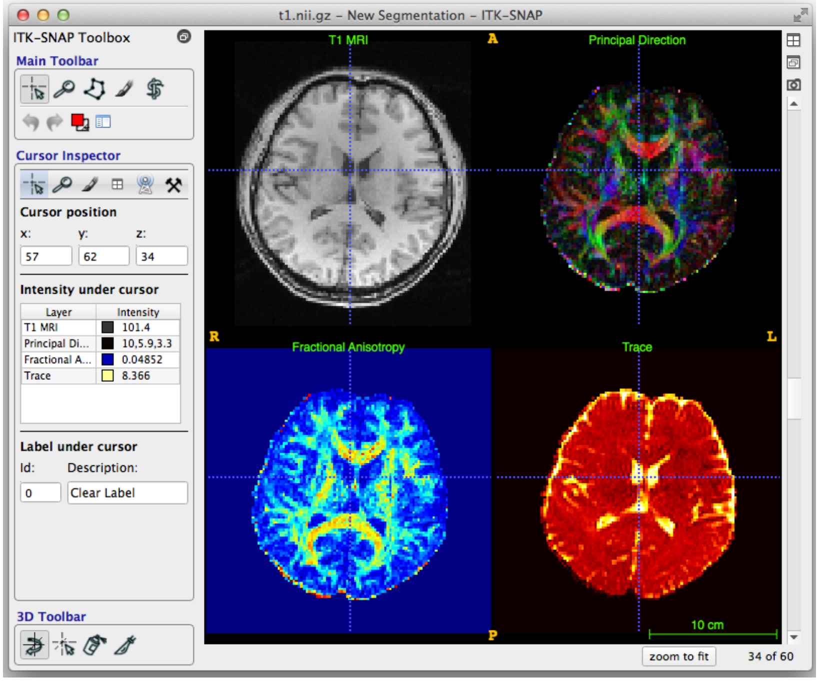
Figure 8 - 3D-Slicer. Source: [25]

3.1.4 ITK-Snap

Possibly, together with 3D-Slicer, the most used option for medical image annotation, since it is the main competitor of the previous option, but with focus on segmentation or semantic classification of images.

It is simpler to use for a radiologist than 3D-Slicer as it does not have other utilities unrelated to this practice and although its interface may be a bit clumsy, it is simple and intuitive. Although the inclusion of plugins is not yet implemented, it has a distributed segmentation service based on docker developed by third parties.

Since this is finally the annotation tool that has been chosen for the project, it will be discussed in more detail in the following section, where it will be explained what its functionalities are and how an automatic segmentation system based on deep learning is integrated into this software.



Cursor position

x: 57 y: 62 z: 34

Intensity under cursor

Layer	Intensity
T1 MRI	■ 101.4
Principal Di...	■ 10,5.9,3.3
Fractional A...	■ 0.04852
Trace	□ 8.366

Label under cursor

Id: 0 Description: Clear Label

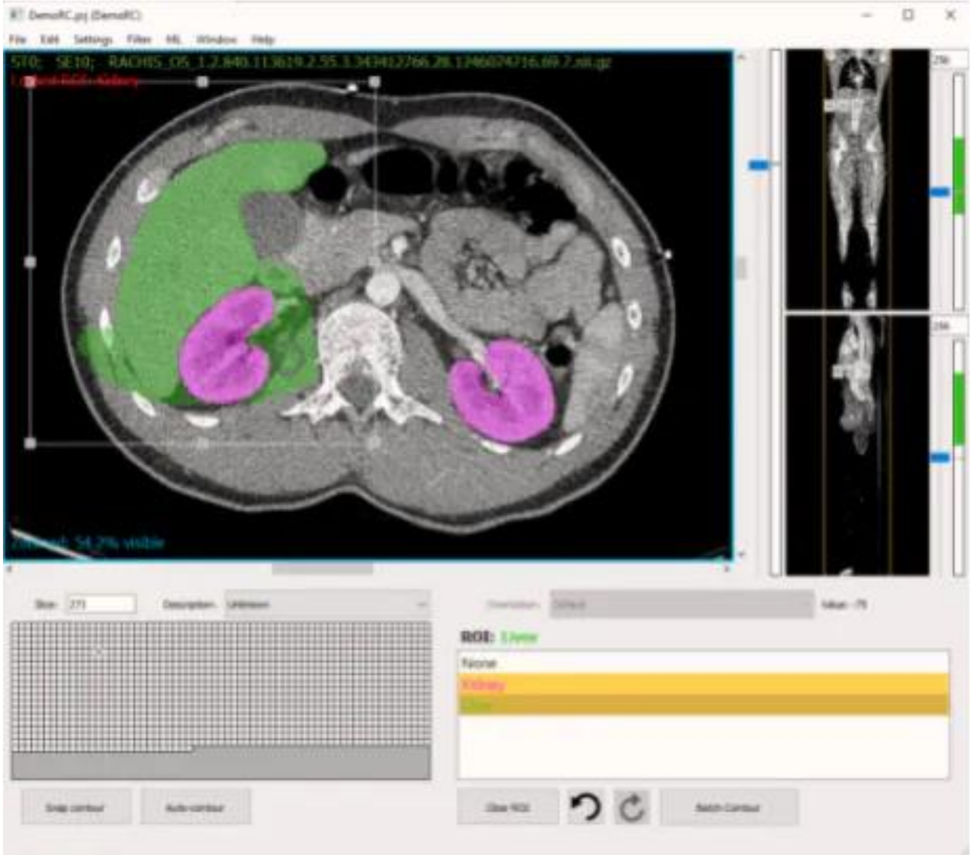
Figure 9 - ITK-Snap. Source: [26]

3.1.5 Ril-Contour

Native application focused on medical image segmentation compatible with NIfTI files. Closer alternative to LabelImg but focused on the scope of the current project usable via conda or from source code. This software is multi-platform and open-source, but its execution is necessarily limited to local.

However, it is little known, so the documentation is not too abundant (although the application is simple) and it seems that several bugs have been reported.

Ril-Contour was initially selected due to the proximity of the software to our approach. Written directly in Python and providing integration of automatic segmentation models with machine learning, it seemed a promising option to say the least, but after thorough evaluation, it had several shortcomings that were crucial for the proposed task, such as handling different channels simultaneously (i.e., different NIfTI files at the same time or NIfTIs composed of several channels).



Slice: 271

Description: Unknown

ROI: Liver

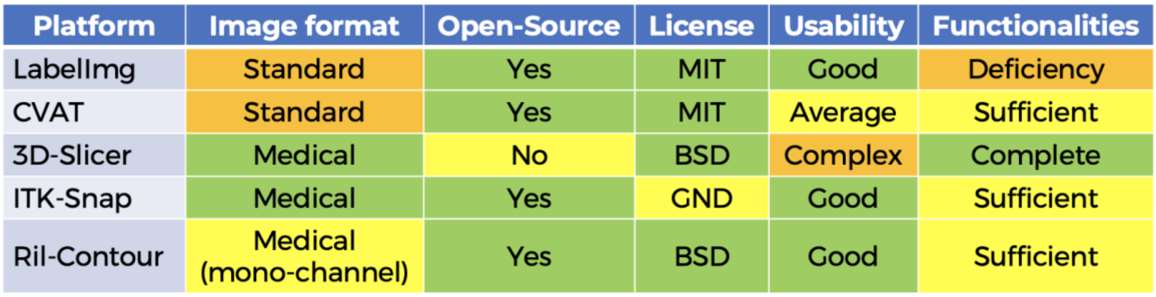
Figure 10 - Ril-Contour. Source: [27]

3.1.6 Summary

There are also other options for medical image segmentation that have not been included due to commercial licensing or proprietary code limitations, such as MedSeg, or that are under development and not usable today, such as MedTagger. Other options have been discarded due to their lack of compatibility, for example, Biomedisa; or that they are overshadowed by some of those already mentioned, such as MITK or ImageTagger.

Table 1 summarizes the characteristics of the applications analyzed according to the criteria defined at the beginning of this chapter, in addition to licensing, usability and functionalities. These characteristics have been color-coded according to the job requirements.

Table 1 - Platform summary. Source: Own elaboration.



Platform	Image format	Open-Source	License	Usability	Functionalities
LabelImg	Standard	Yes	MIT	Good	Deficiency
CVAT	Standard	Yes	MIT	Average	Sufficient
3D-Slicer	Medical	No	BSD	Complex	Complete
ITK-Snap	Medical	Yes	GND	Good	Sufficient
Ril-Contour	Medical (mono-channel)	Yes	BSD	Good	Sufficient

3.2 Selected platform

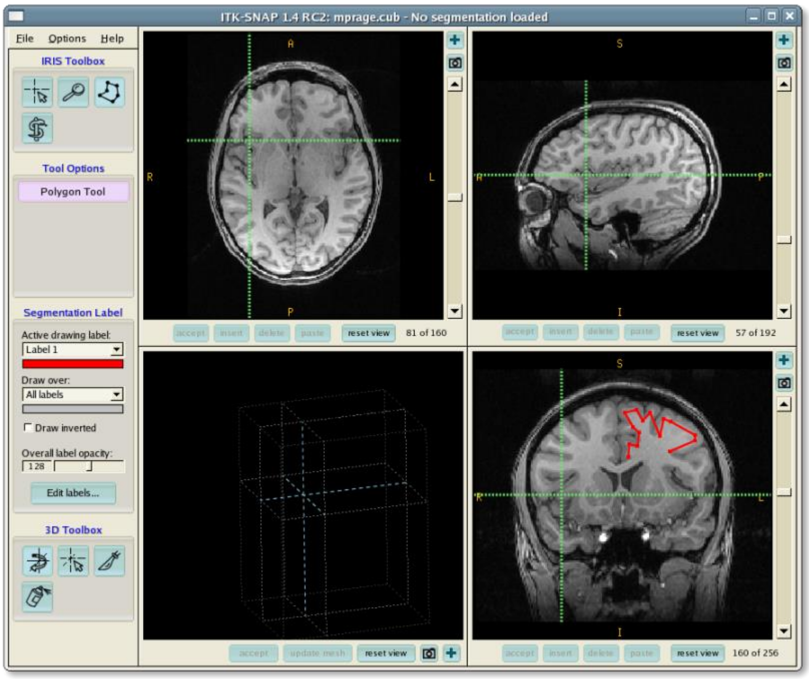
The annotation platform is an important factor for this project since most of the end-user interactions with the application will depend entirely on it and developing a completely ad-hoc platform is beyond the scope of the work.

After exploring the different alternatives, the option that has been chosen is ITK-Snap, as it is one of the most common applications in the field of medical segmentation with which radiologists are familiar.

3.2.1 ITK-Snap in detail

ITK-SNAP is a software application for segmenting structures in 3D medical images and that is easy to use and learn. ITK-SNAP is free, open source and cross-platform. [26]

For this application there is a service called Distributed Segmentation Service that allows encapsulating any operation using docker. Since ITK-Snap allows handling multi-channel files or several files at the same project, with this service we can carry out segmentations on several channels without modifying the application itself, and also on an environment that is friendly to the end user. ITK-Snap is multi-platform and open-source, and thanks to Distributed Segmentation Service, the segmentation execution could be done locally or in a distributed way.


ITK-SNAP 1.4 RC2: mprage.cub - No segmentation loaded

Active drawing label: Label 1

Draw over: All labels

☐ Draw inverted

Overall label opacity: 128

Figure 11 - ITK-Snap manual segmentation. Source: [26]

ITK-SNAP offers, among other functionalities, semi-automatic segmentation using active contour methods, as well as manual delineation and image navigation. In addition to these basic functions, ITK-SNAP offers many supporting utilities. Some of the main advantages of ITK-SNAP are:

- Support for many different 3D image formats, including NIfTI and DICOM.

- Support for concurrent and linked viewing and segmentation of multiple images.
- Support for color, multichannel and time-variant images.

Compared to other larger, open-source image analysis tools, ITK-SNAP's design focuses specifically, as stated above, on the image segmentation problem, and external or unrelated features are minimized. The design also emphasizes interaction and ease of use, and most of the development effort has been devoted to the user interface.

3.2.2 Distributed Segmentation Service

The basic idea of Distributed Segmentation Service is as follows: the system receives as input a set of NIfTI files (generally, those that were already open in the ITK-Snap workspace) and will give as output another set of files, for example, those that were given as input and additionally the performed segmentation. The system is indifferent to what happens between the inputs and outputs, in other words, it gives us the freedom to employ any framework of our choice, which will be discussed in more depth in the next chapter.

Thus, DSS allows images to be sent directly from ITK-Snap to external service providers in order to apply advanced image processing algorithms to the data, with just a few mouse clicks. When using DSS, the client communicates with a middleware server, a web-based application whose main server is https://dss.itksnap.org, although it also allows local execution of this server, a course of action followed by our proposal.

The DSS architecture is composed of three layers, as illustrated in Figure 12:

- Client: A command line tool or GUI that communicates with DSS over the web. Existing DSS clients are the ITK-Snap GUI and the itksnap-wt command line tool, included with ITK-Snap.
- Middleware: The middleware layer is a Python-based web application that orchestrates communication between various service providers and clients.
- Service: Layer where the algorithms provided by the various providers are executed as DSS services, mainly using instructions from the itksnap-wt command line tool. For the current project, the services would be based on ad-hoc algorithms for glioma segmentation.

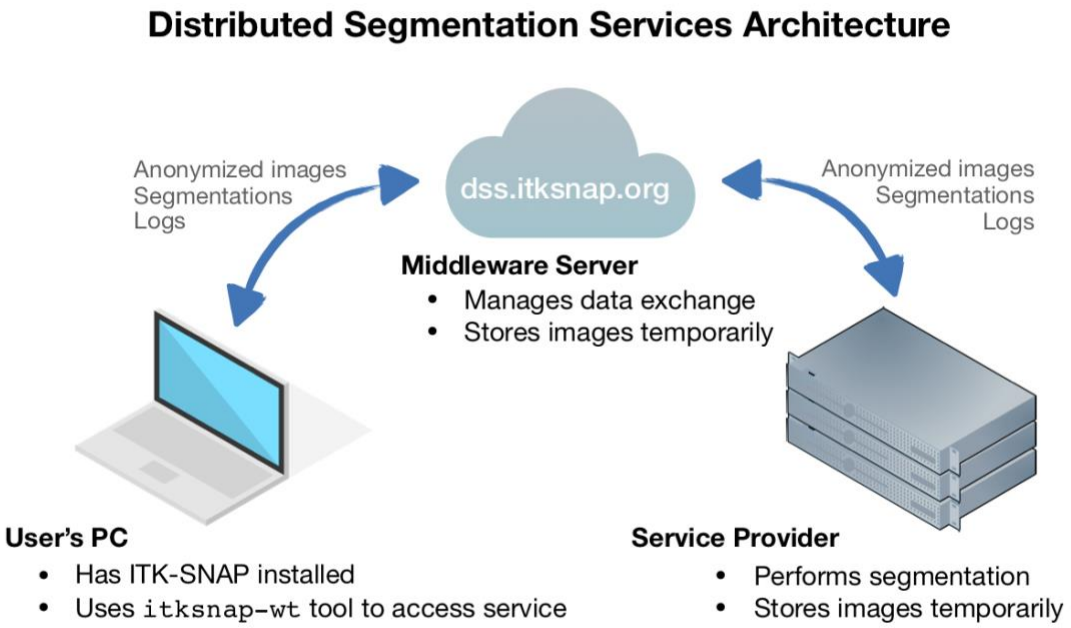


Figure 12 - DSS Architecture. Source: [28]

The DSS system is composed of three docker containers. The first one, the service itself, is the one explained in the immediately preceding paragraph, while the other two containers correspond to the middleware layer (one for the database and web application respectively). The client layer does not require a container since, as mentioned above, it is integrated in the ITK-Snap application itself.

Although the middleware layer has administrative functionalities that facilitate the registration of demanded services, user management, etc., they will not be too relevant for the current project, since it is intended to run these services locally and not through an external server that requires user control.

Figure 13 illustrates the DSS workflow from the time the user requests to use a service until the service is completed:

First, the user must submit a 'ticket' for a particular service with which to manage and monitor the image processing task. For this the user will be asked to provide the necessary images present in the current ITK-Snap workspace or to add external images to it.

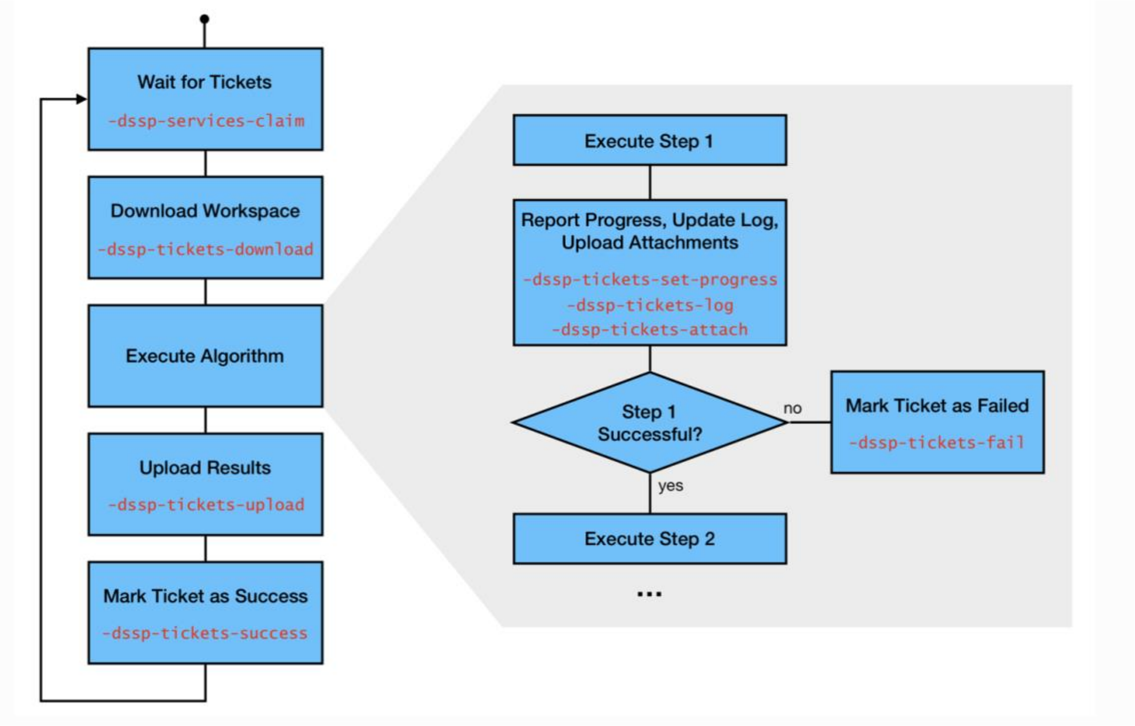


Figure 13 - DSS operation. Source: [28]

After this DSS will download the workspace with the images indicated by the user and will execute the algorithm to finally return the results of the algorithm, depending on what the particular service dictates.

With all the above mentioned in this section, thanks to the final choice of using ITK-Snap and Distributed Segmentation Service, therefore, made it easier to focus our efforts on modeling and how the models would adapt to user interactions.

4. Modeling

This chapter discusses the methodology followed to create the initial segmentation models from which new models will be adapted with the proposed software, as well as their adaptation. Thus, the data used for the creation of the models, the network architecture used, the training strategy and the technologies, both hardware and software, will be discussed, as well as the experimentation that will be performed to obtain these models and how they will be adapted.

4.1 Datasets

In the field of segmentation of this type of brain tumor, BraTS (Multimodal Brain Tumor Segmentation) Challenge [29][30] represents a reference to train glioma segmentation models and to compare the performance between different models.. All BraTS MRI scans include native T1 and post-contrast T1-weighted (T1gd) T2-weighted (T2), and T2 Fluid Attenuated Inversion Recovery (T2-FLAIR) volumes and were acquired with different clinical protocols and various scanners from multiple institutions.

In these datasets the following pathological tissues beyond healthy tissue are defined: "edema", " non-enhancing core", "necrotic core" and "enhancing core". These structures can be identified by the use of the different MRI sequences cited above.

"Edema" is mainly obtained from T2 images, where FLAIR is used to check the extent of edema and to discriminate it from ventricles or other fluid-filled structures.

For segmentation of the tumor core, where the other three tissue types are located, T1 and T1gd images are used. Within the core, the "enhancing core" is obtained by intensity thresholding with T1gd images including the gadolinium-enhancing tumor rim and excluding the necrotic center and vessels.

On the other hand, the "necrotic core" is composed of the low-intensity necrotic structures within the T1gd-visible enhancing rim, and finally the "non-enhancing core" structures were defined as the remaining part of the gross tumor core, i.e., after subtracting the "enhancing core" and the "necrotic core" structures. In the latest datasets (from year 2021), tissues corresponding to “non-enhancing core” are no longer considered, so that those previously belonging to this tissue will be reclassified to “necrotic-core” or “enhancing-core”.

Such tumor substructures meet specific radiological criteria and serve as identifiers to recognize regions of similar appearance by algorithms that process the image information, rather than providing a biological interpretation of the annotated image patterns. For example, "non-enhancing core" labels may also include normal enhancing vascular structures that are close to the tumor core, and "edema" may be the result of cytotoxic or vasogenic tumor processes, or previous therapeutic interventions [31].

Despite this distinction of structures, the metrics and results will be measured on three combinations of these, according to the metrics used in the BraTS Challenge, which are: complete or whole tumor, composed of the four structures, tumor core formed by the "necrotic core", "enhancing core " and "non-enhancing core"; and finally, the enhancing tumor that corresponds to the “enhancing core”. In Figure 14 the distinction and segmentation of the tumor tissues can be seen visually reflected.

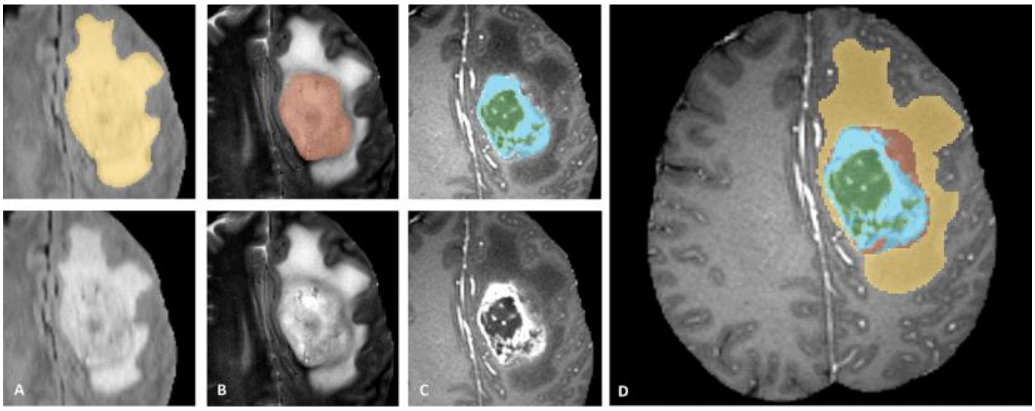


Figure 14 - Tumor Segmentation datasets 2017-2019. Source: [31]

From left to right: whole tumor visible on FLAIR (a), tumor core visible on T2 (b), enhancing tumor structures visible on T1Gd (blue), surrounding cystic/necrotic core components (green) (c). The segmentations are combined to generate the final labels of the tumor structures (d): edema (yellow), non-enhancing core (red), cystic/necrotic core (green), enhancing core (blue).

Figure 14 still has the distinction of the non-enhancing tumor class as can be seen. This classification corresponds to the 2017 and 2019 datasets that have been used. On the other hand, Figure 15 is the homonymous image of the 2021 dataset where this tissue type is no longer taken into account. In view of both figures, panels A-C represent the regions considered for the performance evaluation, while panel D shows the combined segmentations that produce the labels of the tumor subregions.

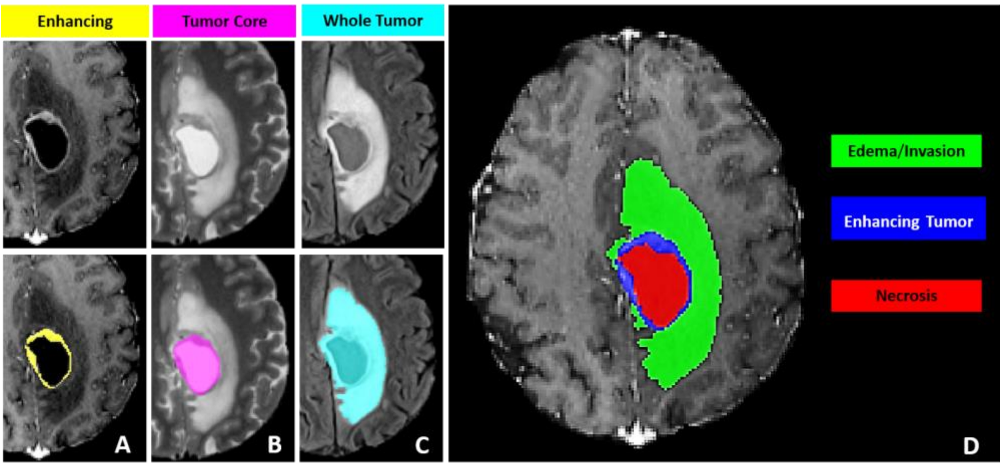


Figure 15 - Tumor Segmentation dataset 2021. Source: [32]

Table 2 summarizes the tumor structures that will be used to measure the results. In the rows are the different classes that are used during segmentation, and in the columns the structures that are used to obtain results, where it is indicated which of the above belong to that structure.

Table 2 - Structure summary. Source: own elaboration.

	Enhancing tumor	Tumor core	Whole tumor
Enhancing tumor	X	X	X
Necrotic and Non-enhancing tumor		X	X
Edema			X

All the imaging datasets have been segmented manually, by one to four raters, following the same annotation protocol (which may vary slightly between datasets from different years but remains consistent within each dataset), and their annotations were approved by experienced board-certified neuro-radiologists. Annotations comprise the GD-enhancing tumor (ET), the peritumoral edematous/invaded tissue (ED), and the necrotic tumor core (and non-enhancing tumor core if it is taken into account in the dataset in question) (NCR or NCR/NET).

The preprocessing steps performed in the BraTS datasets includes voxel isotropic resampling to a common resolution (1 mm^3), inter-patient registration to a common reference space using the same anatomical template and skull-tripping for cranium removal.

Finally, the datasets used to generate different models have been those corresponding to the BraTS Challenge of 2017, 2019 and 2021, which have 285, 335 and more than 1.000 pre-surgical samples respectively in NIfTI format. Each dataset has the files of the previous BraTS edition to which new ones are added, with certain modifications and corrections as mentioned above. In continuity with what has been commented, the first two datasets distinguish between necrotic tumoral tissues and non-enhancing tumoral, while the last dataset only contemplates necrotic tissue.

4.2 Architecture

The network architecture used to create the base model for the glioma segmentation is the one proposed in [33]. This follows the structure of a U-net (widely considered the state of the art for image segmentation), more specifically, it is a residual inception U-net. This kind of network is based on Residual-Inception blocks with the objective of capturing features at different scales. For simplicity, the term simple block is used to denote that a convolution followed by an activation function ReLu, and Batch Normalization is being used.

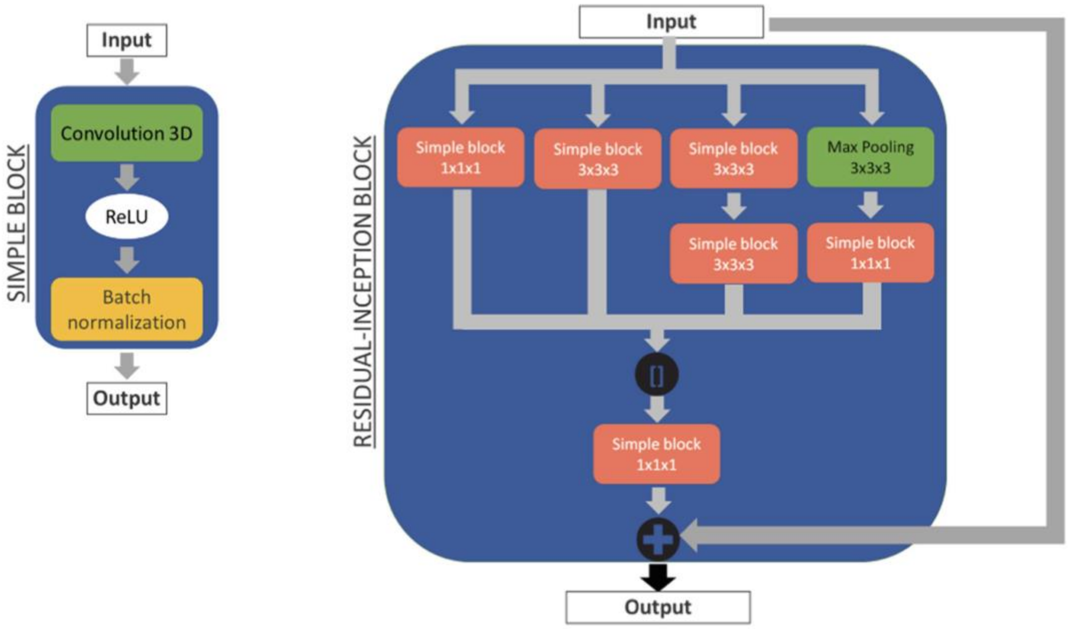


Figure 16 - Residual Inception Block. Source: [33]

These Residual-Inception blocks will be composed, as illustrated in Figure 16, with the structure:

- Simple block with kernel size 1.
- Simple block with kernel size 3.
- Two consecutive simple blocks with kernel size 3.

- Max Pooling with kernel size 3 (and stride 1) followed by a simple block with kernel size 1.

The output of these four channels will be concatenated and another simple block with kernel size 1 is applied after this to reduce the information. Lastly, a residual connection joins the input of the Residual-Inception block with the output. The number of filters used in each of these blocks will depend on the level of the network in which it is located, as described below.

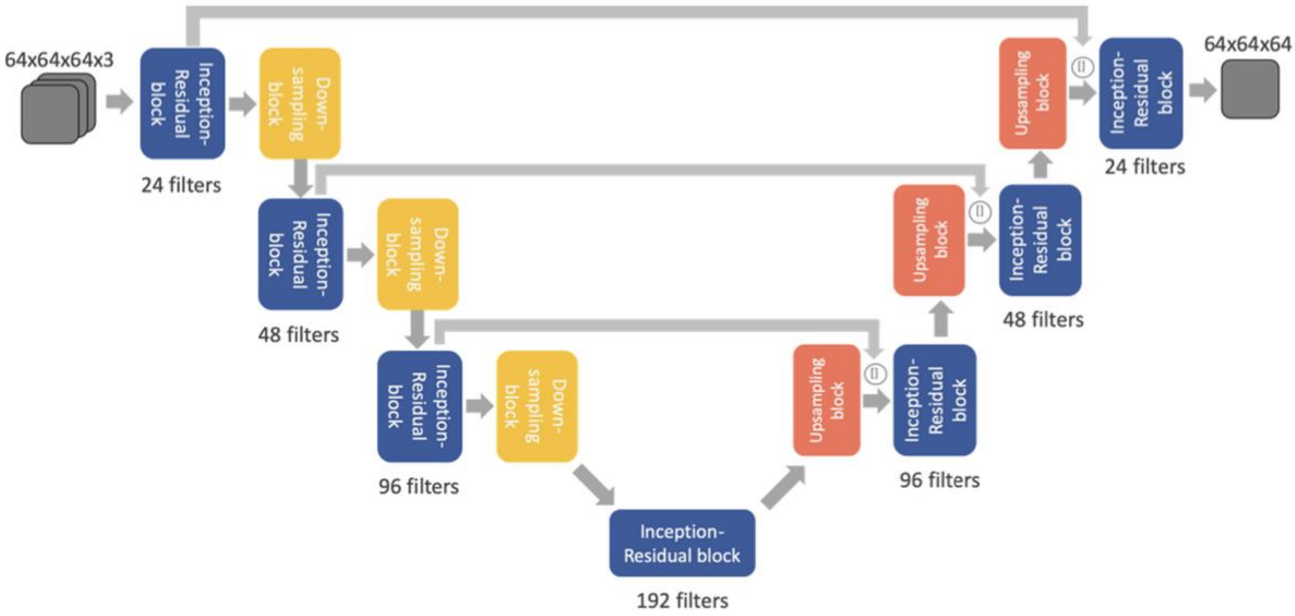


Figure 17 - Network architecture. Source: [33]

As can be seen in Figure 17, our final design consists of four depth levels, where, in the case of the encoder (down-sampling), convolutional blocks of kernel size 3 with stride 2, ReLu as activation function and Batch Normalization are used, and in the case of the decoder (up-sampling) deconvolutional or transposed convolution blocks with the same characteristics.

Each of these levels has, respectively, 24, 48, 96 and 192 filters, where a Residual-Inception block will be applied with the number of filters of its corresponding level.

Furthermore, following the U-net structure, long concatenation-skip connections between symmetric levels are used to improve the gradient flow through the training phase.

4.3 Training strategy

With respect to the preprocessing techniques already included in the dataset, briefly discussed in section 4.1, the following have been added in order to improve the training process.

Firstly, as specified in the datasets section, four 3D images of a patient at a given study are obtained from MRI scans using different radiological MRI sequences: T1, T1Gd, T2 and Flair. However, since the information provided by T1 may be somewhat redundant with the information provided by T1Gd, T1 will not be used to obtain the final models and has only been used to verify that it does not really provide significant information and that its inclusion causes minimal differences in the results. Thus, the remaining three channels will be used as input to the network: T1Gd, T2 and Flair.

Before stacking the volumes, z-score normalization has been applied to each of these channels on those voxels corresponding to the brain, i.e., excluding the background from the normalization. In addition, Gaussian noise sampled from a normal distribution with random parameters has been applied with a probability of 0.5 as data augmentation, and random MRI bias field artifacts has been added with the same probability creating intensity variations of very low frequency across the whole image.

One of the most important points to highlight about the strategy followed during training is that the process is based on patches instead of complete images, i.e., the images are divided into several fragments or patches that will be the input of the network instead of the complete images. This is done due to hardware limitations, because although powerful machines are available for training, the ultimate goal is that the models are portable and can be run on conventional machines. In this way, patches of size 64x64x64 will be extracted from the complete 3D images and considering the number of channels to be used (T1Gd, T2 and Flair), the input of the network will be 64x64x64x3.

The two main strategies followed to obtain these patches were: obtaining patches randomly from the complete images, i.e., with a uniform probability throughout the volume; and obtaining balanced patches by making the different classes equally likely to be the center of the extracted patch.

The loss function used for training is mainly Dice loss, but the weighted sum of Dice loss and cross entropy loss has also been used in certain cases to study whether it improved the results obtained with dice alone. As for the optimizing algorithm, Adam has been used with an initial learning rate of 1e-3 and learning rate scheduler with patience 10 and multiplicative factor 0.5.

Due to memory restrictions, a batch size of 24 has been used, where it must be considered that 12 patches are extracted per volume to increase the variability in each batch, i.e., each batch will contain patches extracted from two different volumes or samples.

4.4 Technology used

Among the technologies used to develop the models, three main ones have been used: PyTorch as a base, Monai and TorchIO.

The first one, which serves as a support for the following two, PyTorch [34], is a Python package that provides tensor computation with strong GPU acceleration and deep neural networks built on a tape-based autograd system. It is characterized by the fact that backpropagation uses dynamically created graphs on the fly, which allows the user to change the way the network behaves arbitrarily.

Monai [35] is an open-source framework written in Python and based on PyTorch and specialized in the field of deep learning in healthcare imaging, both in classification and segmentation tasks. It provides domain-optimized foundational capabilities for developing medical imaging training workflow in a native PyTorch paradigm. Monai has been used in this work to create the entire network architecture from scratch.

On the other hand, TorchIO [36] also is an open-source Python framework centered in deep learning for medical imaging that follows the design of PyTorch, but this one focuses on efficient loading, preprocessing, data augmentation (including intensity and spatial transforms) and patch-based sampling of 3D medical images. This last feature is the one that has been exploited from TorchIO, since it has been used mainly to extract patches from the images, although some data augmentation mechanisms have also been used.

Furthermore, the training process has been entirely run on Nvidia-docker. Nvidia-docker [37] is a container toolkit that allows users to build and run GPU accelerated docker containers running on Nvidia GPU equipped machines. The use of Nvidia-docker

The container has been run on an Ubuntu 18.04 machine and has been allocated 240 GB of RAM and a Tesla V100 SXM2 32GB GPU.

4.5 Experimentation

Once the network architecture to be used and the strategy to be followed for training the models have been defined, it is important to note that in no case is it a process that is already implemented in any of the libraries used. In other words, the network has been created manually according to the explained structure and the processes have been developed ad-hoc.

Based on the training strategy, different experiments have been performed using the three datasets discussed above and the results have been measured using the Dice coefficient. The results obtained for the described experiments will be presented in section 6. As mentioned in the datasets section, this metric will not be obtained from the classes resulting from the segmentation but will be made on combinations of those: enhancing tumor (which does match the class with the same name), tumor core (sum of enhancing tumor and necrotic/non-enhancing tumor) and whole tumor (sum of all classes, enhancing tumor, necrotic/non-enhancing tumor, and edema).

Furthermore, it is important to note that the tests designed are not intended to improve the state of the art, but rather the main objective of this stage is to obtain a suitable base model with which to start adapting new models, as will be described in the next subsection.

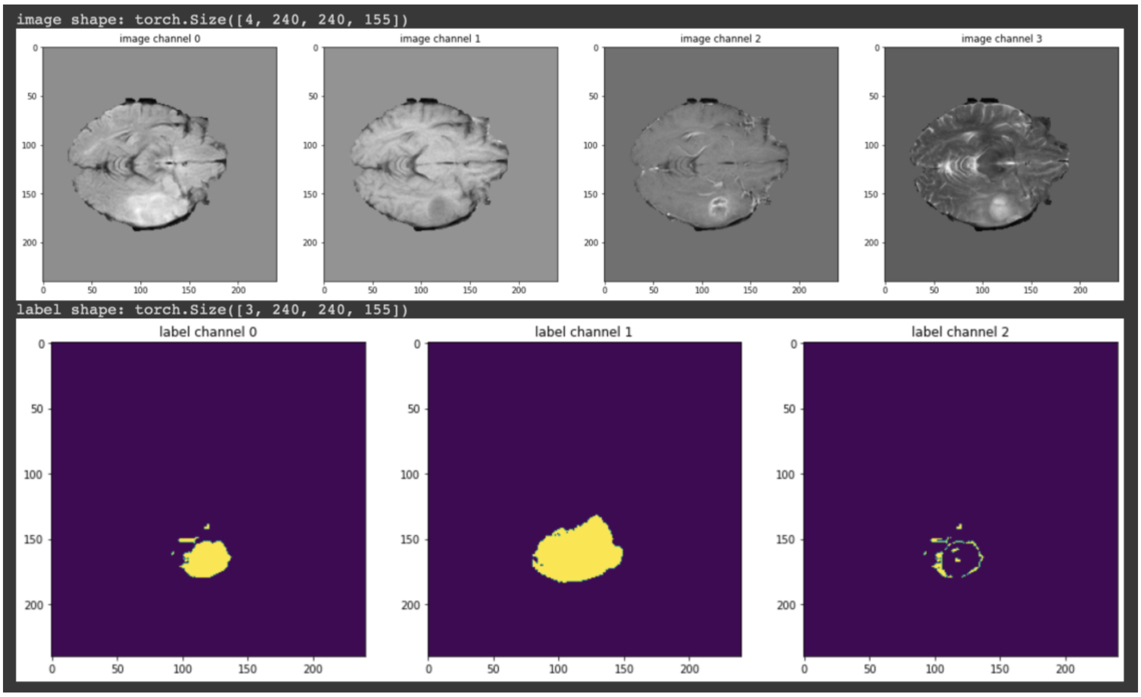
The first experiment proposed is to use the full images for training. It should be noted that this test is carried out in order to corroborate that the architecture used is suitable for this task and to contrast the results with some frame of reference (since they usually work with the full images). After this control experiment that will be performed on the three datasets, the experiments that will actually produce the models we will use, those based on images divided into patches, will begin.

As a test, it is proposed to use only labeled patches, i.e., patches that necessarily contain some class label, and thus not to train with patches that were 100% healthy tissue. It is also proposed to train with patches obtained sequentially from the images, i.e., each complete image is divided into as many patches as possible and passed sequentially to the network. These two alternatives are expected to be the worst performers as they are the furthest from the proposals used in the state of the art. If so, they will only be used in the first dataset.

The most relevant strategies for obtaining patches, random patches and balanced patches are proposed below. These strategies are described in the training strategy section. If the two previous ones produce worse results as expected, these tests will be replaced in the remaining datasets with random patches and balanced patches but applying a slight fine-tuning.

It is worth noting that, for the generation of patches, a distinction has been made between two approaches, multi-label, and multi-class. This is because initially the multilabel approach was followed since it is the one proposed by Monai, but obtaining balanced patches is not implemented in TorchIO for this approach, so we resorted to use the other one.

Figures 18 and 19 illustrate the difference between these two perspectives on the same sample. While the first, multi-label, obtains as output from the network a mask with three binary channels (one for each structure to be evaluated: enhancing tumor, tumor core and whole tumor), the multi-class obtains as output a mask with a single channel containing all the classes to be distinguished from which these structures (enhancing tumor, tumor core and whole tumor) can be obtained.

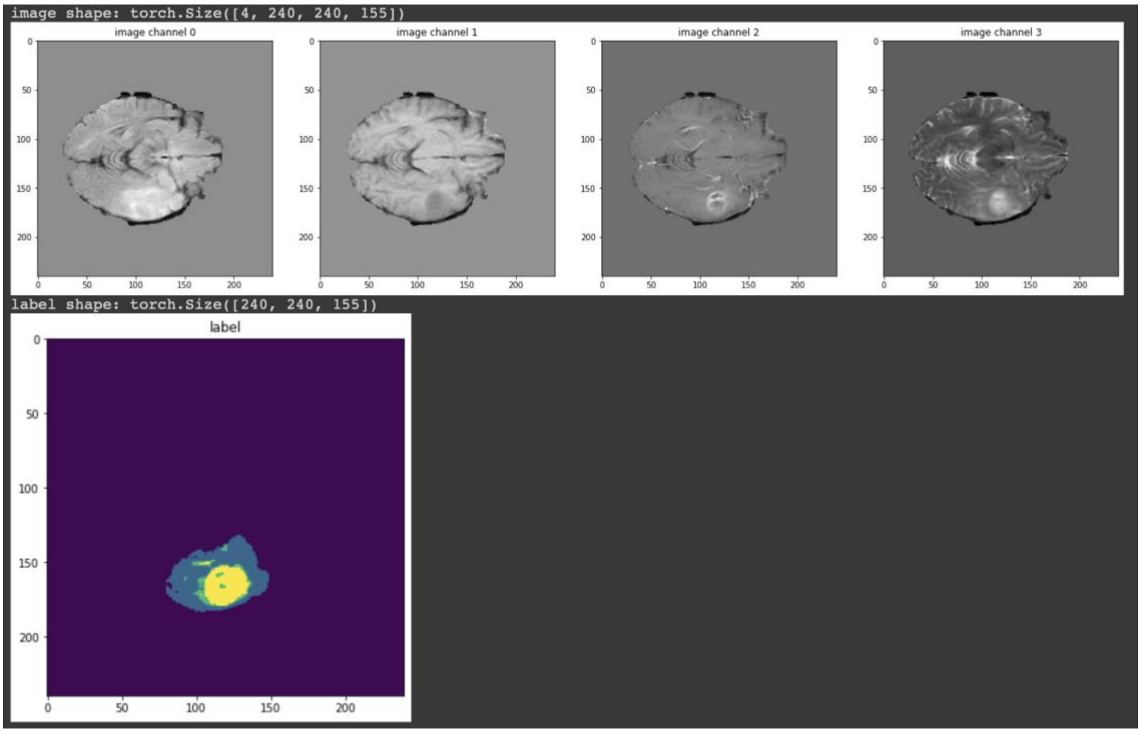


```
image shape: torch.Size([4, 240, 240, 155])
```

```
label shape: torch.Size([3, 240, 240, 155])
```

Figure 18 - Multi-label approach. Source: Own elaboration.

It should be noted that the purpose of considering and exploring different alternatives for extracting patches from the source volumes is to minimize the impact of working on the patched space instead of the original one.



```
image shape: torch.Size([4, 240, 240, 155])
```

```
label shape: torch.Size([240, 240, 155])
```

Figure 19 - Multi-class approach. Source: Own elaboration.

In order to check that the results are consistent and that the change of approach does not significantly affect the results, a test was repeated, namely the one based on random patches. This repetition has only been performed on the experiments of the first dataset, and the rest of the experiments have been performed directly from the multi-class approach, since with this approach both random and balanced patch extraction strategies can be obtained.

4.6 Model adaptation

It is important to remember that one of the objectives of the work is to achieve models capable of correctly segmenting postsurgical cases. This will require adapting the models trained with pre-surgical cases to provide better results when used to segment the longitudinal series of post-surgical stages of a patient. In other words, the models trained with the datasets described will be the basis with which to achieve this adaptation.

The retraining process is the one that can generate a great value for the end user, since at present, due to the fact that the models are trained with pre-surgical cases, there is a great variability in the results that these models give when applied to post-surgical cases. This process of fine-tuning the models would be within the domain of transfer learning, since we are adapting a model trained in one domain, that of pre-surgical cases, to work in another, that of post-surgical cases.

To retrain the model, a pre-trained model will be loaded, and data provided by the user radiologist will be used, either post-surgical cases segmented with our models or manually segmented by professionals. In case of having been segmented by our models, the user should correct the mask with the possible imperfections for the adaptation to be effective.

The parameterization of the adaptation should be carried out as accurately as possible since inadequate parameterization could lead to a worsening of the models. Thus, the strategy followed has been rather conservative in order to avoid model deterioration. Therefore, for retraining, the weights of the entire network are frozen except for those of the last layer and retrained by applying the same learning factor with which the network was trained to generate the model.

However, in order to test the improvement of adapting a model with respect to a base model, we have experimented with adapting a model to the cases of a specific center. That is, a model that has been trained with the samples from one center will be used to predict the segmentation masks of cases from a different center. This is due to the lack of availability of post-surgical cases. It is important to note that the differences present in the images from different centers can become significant, due to, among other things, the use of different machinery. Therefore, this adaptation task is not trivial. As in the case of the base models, the results of this test will be shown in the evaluation section.

5. Integration with ITK-Snap

This chapter will explain how the software developed allows the user to perform segmentations automatically and adapt the models created in section 4 to longitudinal series cases of a specific patient. For this purpose, the workflow to be followed by the radiologist using the Distributed Segmentation Service in ITK-Snap as explained in section 3.2 and how the models have been integrated within this service will be explained in detail.

5.1 Services creation

The developed models have been integrated into the chosen annotation platform (i.e., ITK-SNAP), thus allowing the user to perform automatic glioma segmentations directly within the annotator.

As mentioned in its respective section, this annotation platform includes Distributed Segmentation Service, a system that allows developers to publish different services in the cloud. However, our work will be based on the local execution of this service rather than in the cloud, being its inclusion to the cloud a future line of work.

As far as Distributed Segmentation Service (DSS) is concerned, none of its base components such as the administrative web application or the request database will be modified. Only the services proposed in the work will be added: automatic glioma segmentation from a given model and adaptation of a model to a patient.

The first thing to do to create a service is to generate a project in GitHub with a JSON file containing the information of that service. This information includes the title, description, or images needed to carry out the service. The git-hash code of that project will be used inside Distributed Segmentation Service so that the system generates the service according to what is established in the JSON file.

After creating the service, the system waits for a service to be called from the ITK-Snap interface. The way this happens has been modified from the original so that the system, once it is up and running, waits indefinitely until a call is made to any of the services created.

When requesting any of the services the user will be prompted for those images that were defined in the JSON file and the images will be sent to the docker where the Distributed Segmentation Service is running. Here, through Linux commands and itksnap-wt command line a series of actions will be executed and one result or another will be returned to the user. It is here where the bulk of the implementation of the services is located.

Thus, a Python script has been developed for each of the two services that will run Distributed Segmentation Service depending on the chosen service. The first, for glioma segmentation, will build the network proposed in section 4 and load a pre-trained model. It will then load the images requested by the service to perform the segmentation and the inference of the segmentation mask will be performed. The second script, for the model adaptation service, will also load the network structure and a pre-trained model. In this case, in addition to the input images for the network, the corresponding ground-truth will be requested to re-train the model and what will be created in this script is a new model through this process.

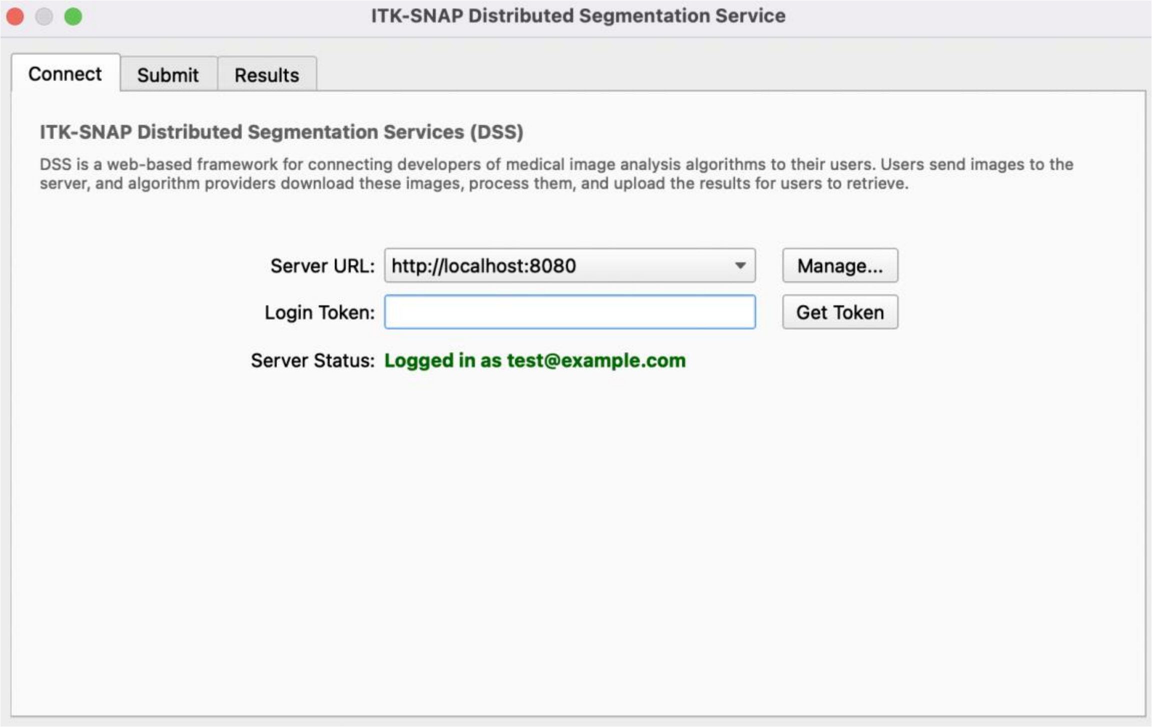
5.2 Segmentation service

This subsection discusses the first of the services, the automatic glioma segmentation service, and the workflow for using the system created before starting to adapt the models, which will be discussed in the next section.

First, once the service is running, the user must connect to the Distributed Segmentation Service from the ITK-Snap interface. In this case, the default server is modified since it is running locally. If the system is working correctly, it will be available to obtain a token to login and access the services, as illustrated in Figure 20.

Once the connection has been successfully established, the services tab will display, in this case, both the proposed segmentation and model adaptation services. As can be seen in Figure 21, the multichannel segmentation service is selected, as it is worth remembering that for the correct segmentation of gliomas several channels are necessary.

These channels are FLAIR, T1gd and T2, explained in section 2.1 and their usefulness for this task is specified in section 4.1. These three images are the ones requested to the user and can be loaded in an ITK-Snap workspace prior to the connection with Distributed Segmentation Service or loaded later.



ITK-SNAP Distributed Segmentation Services (DSS)

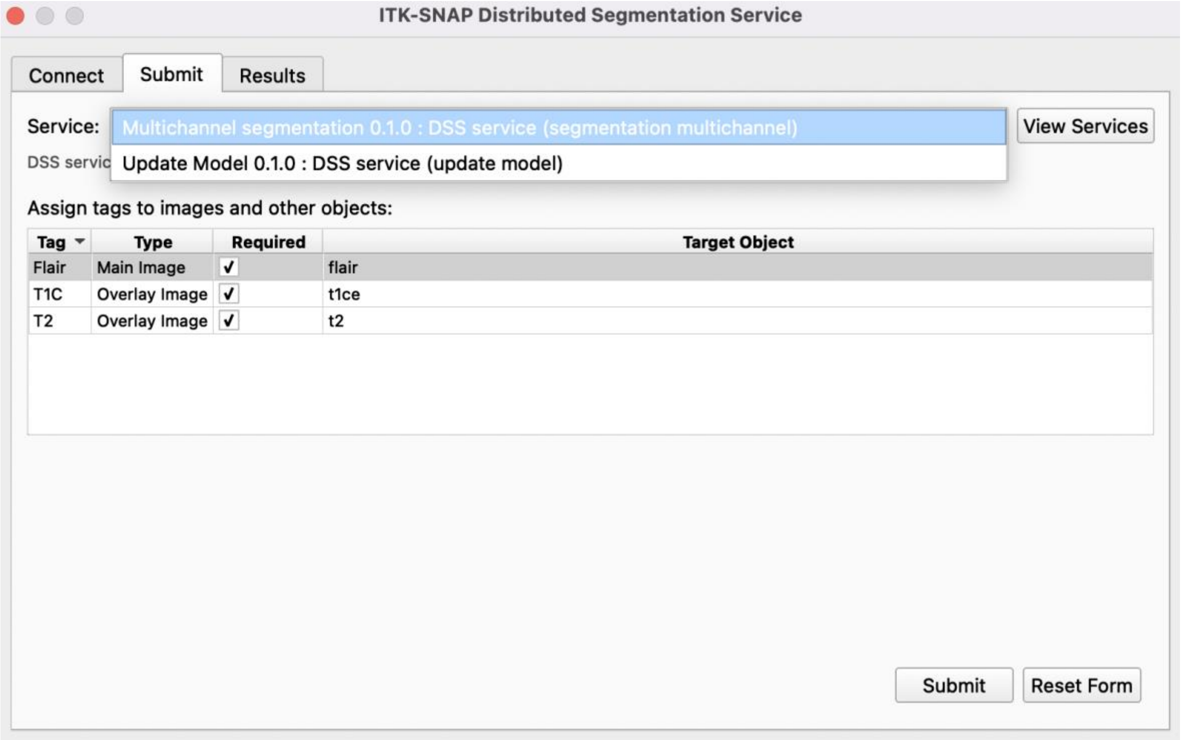
DSS is a web-based framework for connecting developers of medical image analysis algorithms to their users. Users send images to the server, and algorithm providers download these images, process them, and upload the results for users to retrieve.

Server URL: http://localhost:8080

Login Token:

Server Status: Logged in as test@example.com

Figure 20 - Connection to DSS. Source: Own elaboration.



Service: Multichannel segmentation 0.1.0 : DSS service (segmentation multichannel)

Update Model 0.1.0 : DSS service (update model)

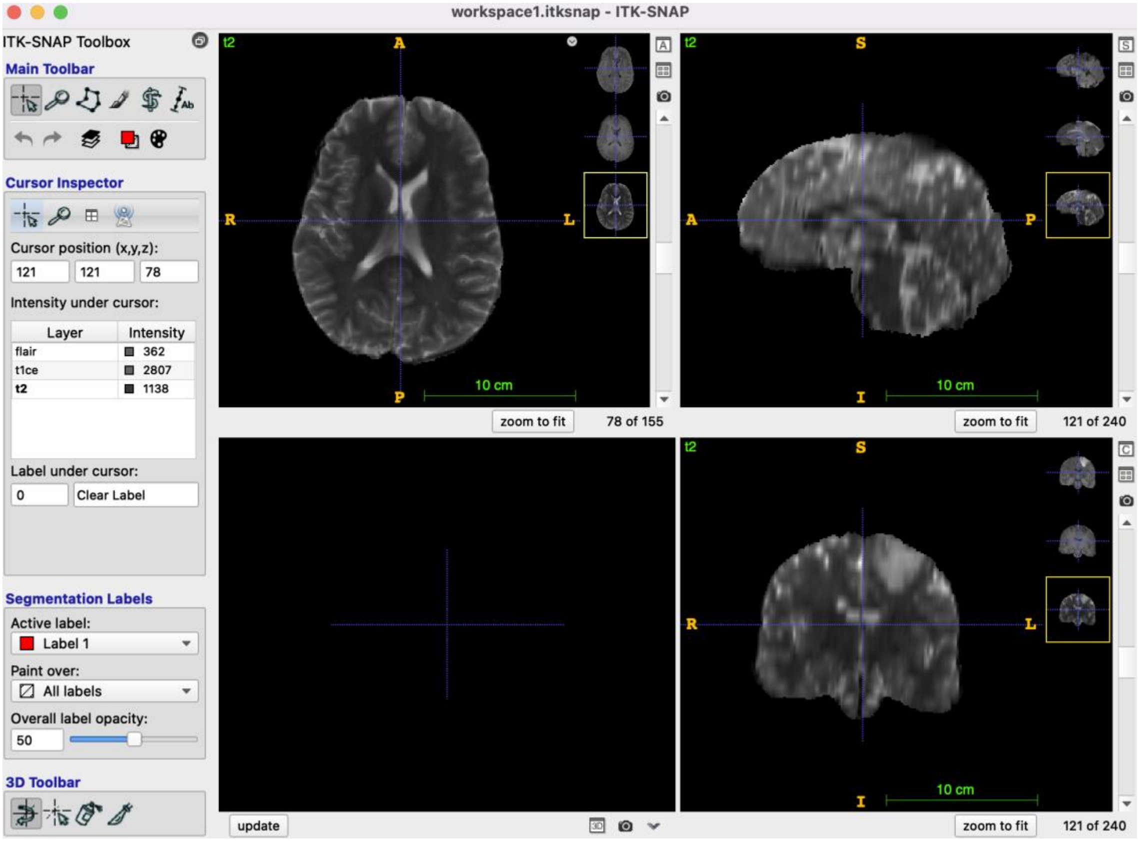
DSS servic

Assign tags to images and other objects:

Tag	Type	Required	Target Object
Flair	Main Image	✓	flair
T1C	Overlay Image	✓	t1ce
T2	Overlay Image	✓	t2

Figure 21 - Services in DSS. Source: Own elaboration.

It is important that these images are loaded correctly as requested by the system, since the order in which they are processed by the network will depend on this. Below, Figure 22 shows an example of an ITK-Snap workspace with these three images loaded, i.e., ready to run the automatic segmentation service.



workspace1.itksnap - ITK-SNAP

ITK-SNAP Toolbox

Cursor position (x,y,z): 121 121 78

Intensity under cursor:

Layer	Intensity
flair	■ 362
t1ce	■ 2807
t2	■ 1138

Label under cursor: 0 Clear Label

Active label: Label 1

Paint over: All labels

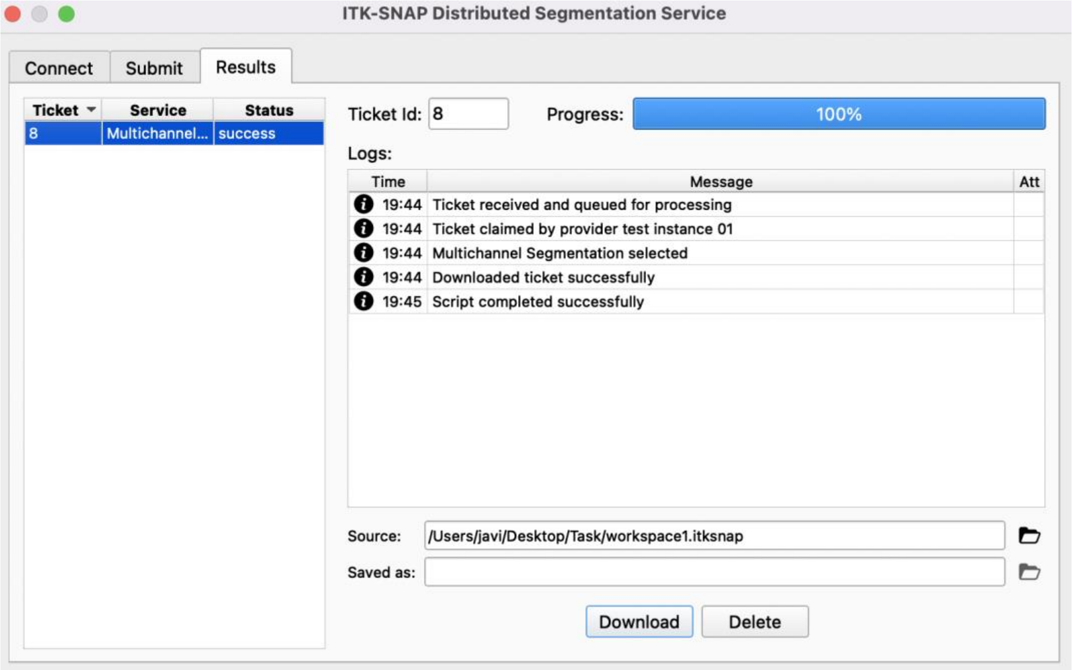
Overall label opacity: 50

Figure 22 - ITK-Snap workspace. Source: Own elaboration.

After uploading the images and submitting the task, the user will be automatically taken to the results tab where the status of the task can be viewed in real time.

This is where the user will be notified of any possible warnings that may occur during the execution of the task, as well as the percentage of execution remaining (determined by the developer of the service). Distributed Segmentation Service gives the developer freedom to make all kinds of communications to the user through this tab, if deemed necessary.

Following the execution trace shown in the figures of this section, Figure 23 shows the results tab after the automatic segmentation service has been completed.



Ticket	Service	Status
8	Multichannel...	success

Ticket Id: 8

Progress: 100%

Logs:

	Time	Message	Att
ℹ	19:44	Ticket received and queued for processing	
ℹ	19:44	Ticket claimed by provider test instance 01	
ℹ	19:44	Multichannel Segmentation selected	
ℹ	19:44	Downloaded ticket successfully	
ℹ	19:45	Script completed successfully	

Source: /Users/javi/Desktop/Task/workspace1.itksnap

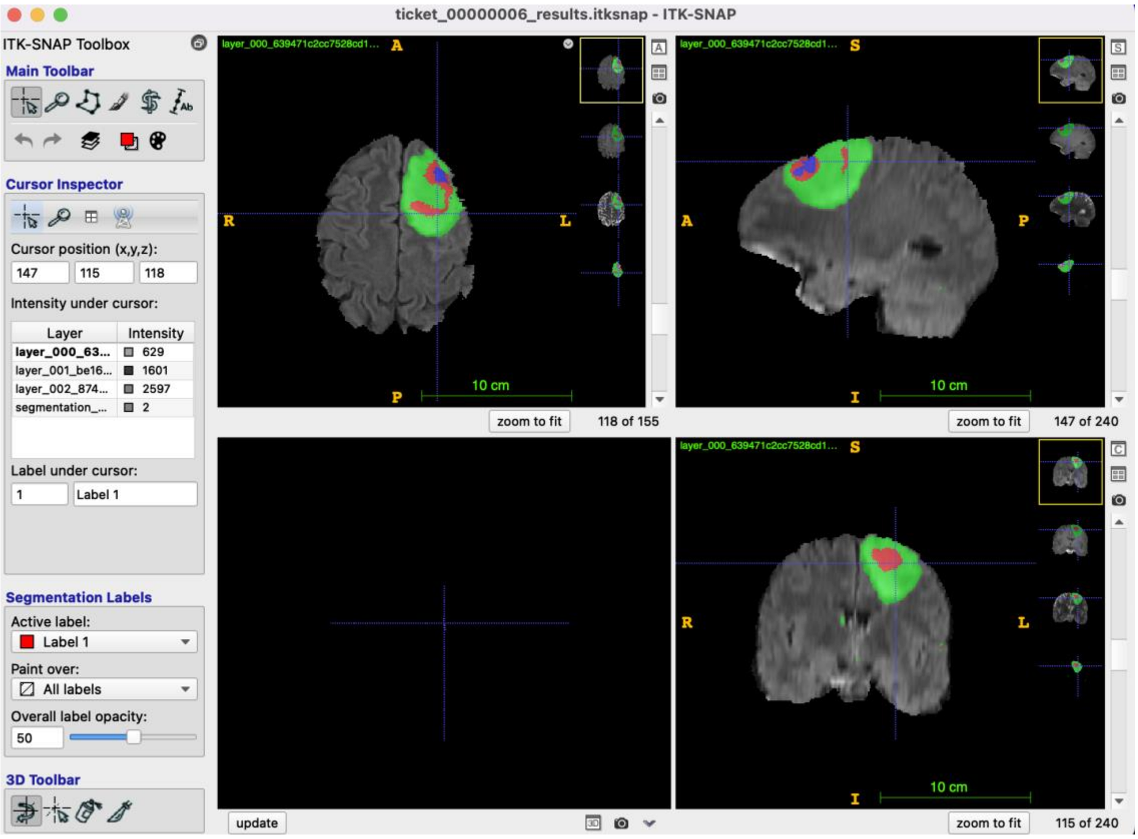
Saved as:

Figure 23 - Results of the service. Source: Own elaboration.

What this service has done is, under the Distributed Segmentation Service system, run the Python script using the convolutional neural network created in section 4, with a pre-trained mode. The model used in this case is the one that has obtained the best results in the evaluation phase, although the use of one model or another is easily configurable, as will be discussed in the following section.

The three channels used for glioma segmentation mask inference will be obtained directly from the ITK-Snap workspace above. It should be recalled that the network is patch-based, so the final volume composed of the three channels will be divided into 64x64x64 patches. Since the input images need not be of such a size that a perfect patch division is made, overlapping patches will most likely occur. In other words, two consecutive patches would contain voxels corresponding to the same area of the original image. In case such an overlap exists, the average of the overlapping sections of the network output will be calculated in order to obtain the most probable label for those voxels in question.

When the segmentation task is finished, the user can download the new ITK-Snap workspace including the three images (FLAIR, T1gd and T2) and the glioma segmentation mask. Figure 24 shows the example for this run trace.



ticket_00000006_results.itksnap - ITK-SNAP

Cursor position (x,y,z): 147 115 118

Intensity under cursor:

Layer	Intensity
layer_000_63...	629
layer_001_be16...	1601
layer_002_874...	2597
segmentation_...	2

Label under cursor: 1 Label 1

Active label: Label 1

Paint over: All labels

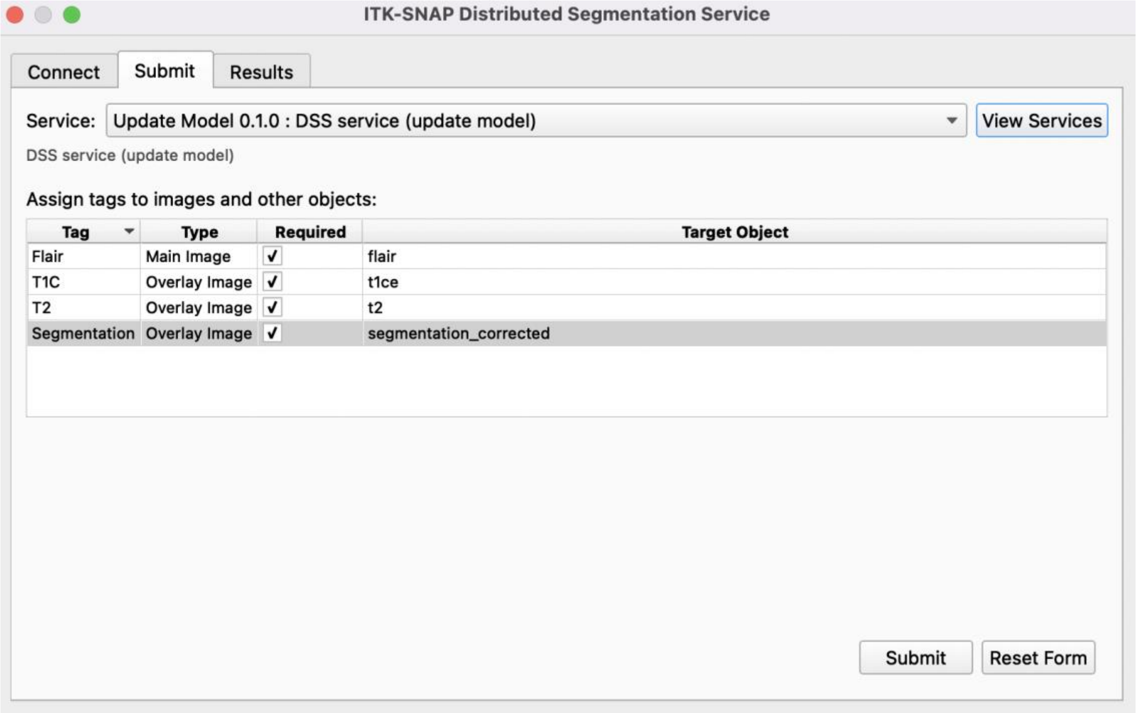
Overall label opacity: 50

Figure 24 - Segmentation completed. Source: Own elaboration.

5.3 Model adaptation service

The main idea of this service is that the radiologist performs an automatic segmentation of a case, for example, the one shown in Figure 24 above, and corrects the segmentation mask to retrain the model and improve its performance for future cases of longitudinal post-surgical series.

As can be seen in Figure 25, in this case the model adaptation service will request, once again, the three images corresponding to FLAIR, T1gd and T2, plus the segmentation mask for that case that may be an automatic or corrected mask. As with the automatic segmentation service, the user will be progressively notified of the status of the task.



Service: Update Model 0.1.0 : DSS service (update model)

DSS service (update model)

Assign tags to images and other objects:

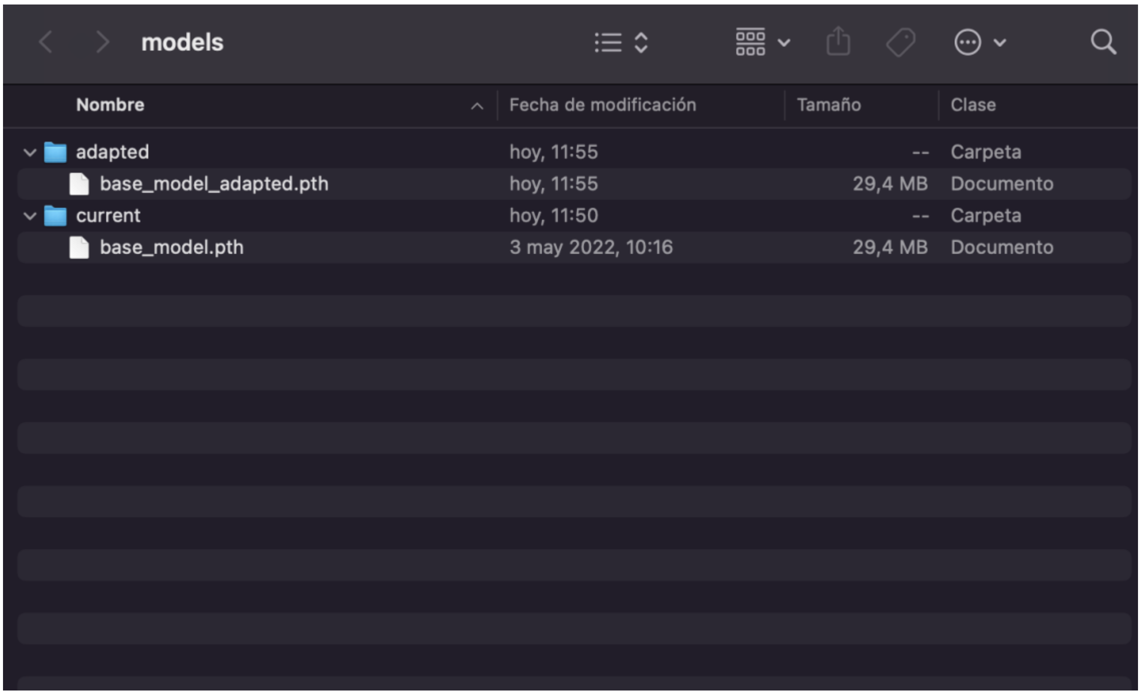
Tag	Type	Required	Target Object
Flair	Main Image	✓	flair
T1C	Overlay Image	✓	t1ce
T2	Overlay Image	✓	t2
Segmentation	Overlay Image	✓	segmentation_corrected

Figure 25 - Update model service. Source: Own elaboration.

The model adaptation service, like the segmentation service, will run a Python script under Distributed Segmentation Service. Thus, any base model will be retrained considering the latter case corrected by the radiologist. The retrained or adapted models can be subjected to this process again with other cases so that the model learns progressively and suits the radiologist's needs. This implies that a model could be adapted to only one patient, or to a set of patients if desired by the user.

The way the user can manage the models is a simple directory system. A directory called 'current' will contain the model with which the segmentation service is being performed, so there must be only one model in this directory (otherwise the user will be notified through the application). Once the model adaptation service is executed, the new model created will appear in another directory called 'adapted'.

The user is free to rename these models (e.g., to identify different patients) and apply the segmentation with the desired model by simply placing it in the 'current' folder. Figure 26 shows these directories after having executed the previous trace, where it can be seen how the model 'base_model' (which was used for the segmentation in the previous section) has generated a new model 'base_model_adapted' after finishing the adaptation service.



Nombre	Fecha de modificación	Tamaño	Clase
adapted	hoy, 11:55	--	Carpeta
base_model_adapted.pth	hoy, 11:55	29,4 MB	Documento
current	hoy, 11:50	--	Carpeta
base_model.pth	3 may 2022, 10:16	29,4 MB	Documento

Figure 26 - Model directory. Source: Own elaboration.

It should be noted that the case presented as an example for model adaptation is not representative of the use case for longitudinal series, but instead a pre-surgical case has been segmented in order to visually show how the system works. This is mainly because longitudinal post-surgical series of patients are not available.

6. Evaluation

This chapter will try to evaluate the developed software from its different perspectives, considering the different sub-objectives set out at the beginning of the study. First, chapter 6.1 will show the results provided by the base models created, as outlined in section 4. Next, section 6.2 will show the improvement provided by the model adaptation as described in section 4.6. Finally, an attempt will be made to evaluate the software from the end-user's perspective based on their feedback.

6.1 Base models

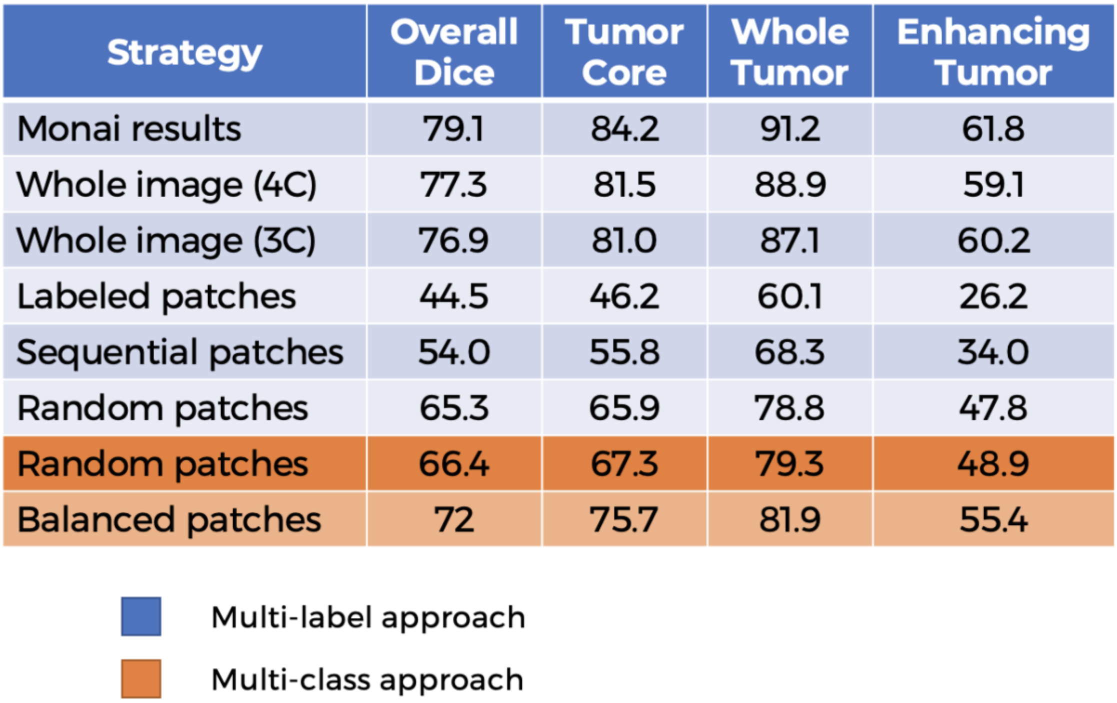
This section presents the results obtained from the experiments described in section 4.5. All tables measure the results using the Dice coefficient, and in the rows will have the different tests or strategies that have been followed for each experiment. The columns will have the results for each of the different structures taken into account, as well as the overall Dice.

6.1.1 2017 BraTS Dataset

Table 3 presents the results, obtained with the 2017 dataset, where the first entry reflects those obtained by Monai as a reference, so the validation set used in this dataset is the same as the one used by Monai. These values provided by Monai, while not the best in the current literature, are presented as some of the highest.

The first experiments consist of using the whole images without dividing into patches, using all channels, and discarding T1 (4C and 3C respectively). This method is only used in order to check that the architecture provided results that did not depart from the state of the art and to confirm that the variation in the results when including the T1 channel was minimal. The models that will actually be used, as previously mentioned, will be based on images divided into patches, not complete images.

Table 3 - 2017 Dataset results. Source: Own elaboration.



Strategy	Overall Dice	Tumor Core	Whole Tumor	Enhancing Tumor
Monai results	79.1	84.2	91.2	61.8
Whole image (4C)	77.3	81.5	88.9	59.1
Whole image (3C)	76.9	81.0	87.1	60.2
Labeled patches	44.5	46.2	60.1	26.2
Sequential patches	54.0	55.8	68.3	34.0
Random patches	65.3	65.9	78.8	47.8
Random patches	66.4	67.3	79.3	48.9
Balanced patches	72	75.7	81.9	55.4

🟦 Multi-label approach

🟧 Multi-class approach

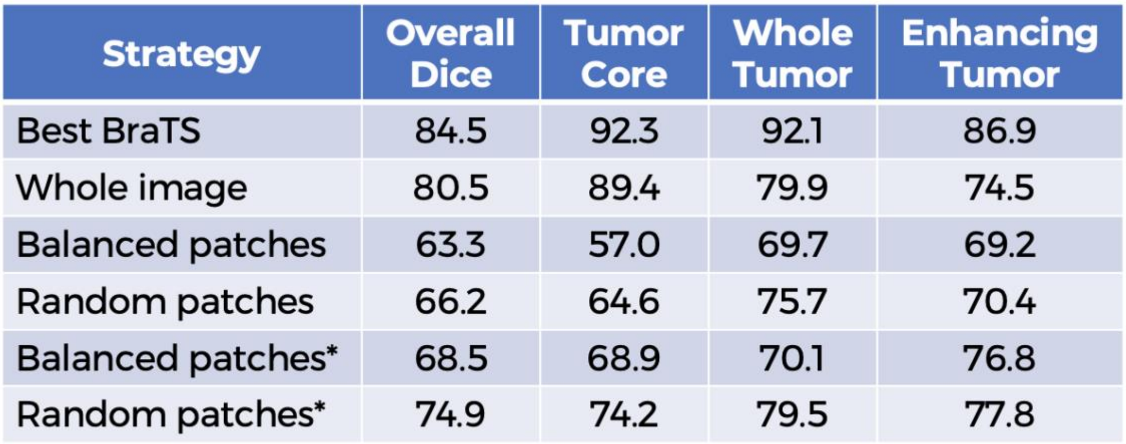
As mentioned in section 4.5 where the experimentation to be performed was described, this first dataset is the only one that will contain the control experiment where 4 images are used instead of 3. It will also be the only one that will have the paradigm shift from multi-label and multi-class, since in view of these results the difference is not only small, but it is superior for the multi-class case. Therefore, all the following tests were carried out from this multi-class perspective.

6.1.2 2019 BraTS Dataset

With this second dataset, the 2019 dataset, it is important to note that, as mentioned in the section on datasets, there are differences in the criteria for performing the segmentation. Therefore, there is an important difference in the evaluation metrics of the enhancing tumor structures.

Table 4 shows the results obtained for this dataset, once again, showing first reference values, which in this case correspond to one of the best results obtained in the BraTS challenge of that year. For this reason, the validation set used in this dataset is the official one published in the BraTS Challenge. In continuity with the previous results, the first experiment performed was with the complete volumes without dividing them into patches to check that we are within the state of the art (emphasize that in this case, the reference is not average values, but the best ones for this task).

Table 4 - 2019 Dataset results. Source: Own elaboration.



Strategy	Overall Dice	Tumor Core	Whole Tumor	Enhancing Tumor
Best BraTS	84.5	92.3	92.1	86.9
Whole image	80.5	89.4	79.9	74.5
Balanced patches	63.3	57.0	69.7	69.2
Random patches	66.2	64.6	75.7	70.4
Balanced patches*	68.5	68.9	70.1	76.8
Random patches*	74.9	74.2	79.5	77.8

In order to obtain the results of this dataset, the less functional approaches used in the previous case have been taken into account, as discussed in Section 4.5, so that only the balanced patches and random patches strategies (using the multi-class approach) have been experimented with.

Since, consequently, the number of tests executed with this dataset would be smaller, a small fine-tuning has been attempted in the experiments marked with “*”, in which the weighted sum of Dice loss and cross entropy loss has been used as loss function instead of only using Dice loss, as it was done in the rest of the cases.

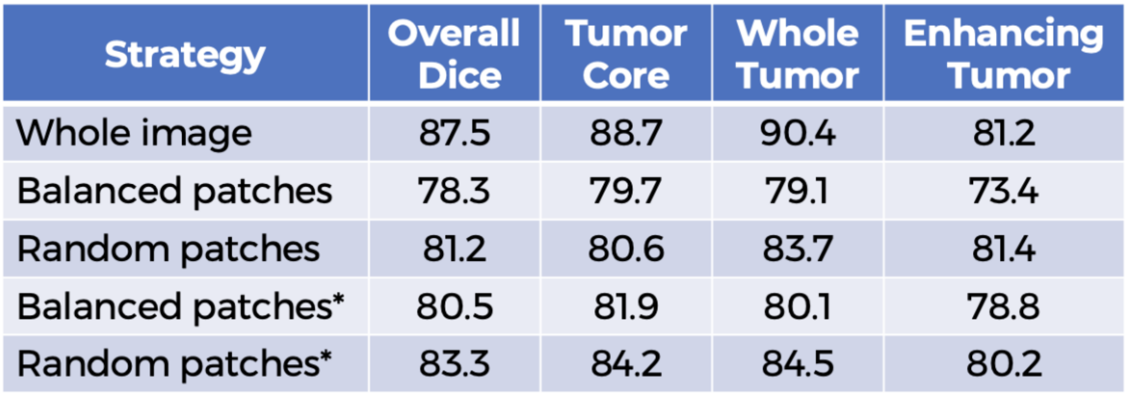
As can be seen, for this dataset, better results were obtained with a strategy based on random patches than with balanced patches, which may be due to these differences in criteria between the datasets when segmenting the different classes. In any case, this does not mean that using balanced patches is a bad decision, but rather that the way in which the balancing is produced (explained in the datasets section) may not be the best for this task.

6.1.3 2021 BraTS Dataset

Finally, Table 5 shows the results obtained for the last dataset used, the one corresponding to the BraTS challenge 2021. In this case, only the own results without reference frame are provided because the challenge is still in the participation phase, so the results have not been published in any case. Because of this, a validation set extracted from the training set has been used.

Therefore, we will take advantage of the timing of this work with that of the challenge to participate with the models created.

Table 5 - 2021 Dataset results. Source: Own elaboration



Strategy	Overall Dice	Tumor Core	Whole Tumor	Enhancing Tumor
Whole image	87.5	88.7	90.4	81.2
Balanced patches	78.3	79.7	79.1	73.4
Random patches	81.2	80.6	83.7	81.4
Balanced patches*	80.5	81.9	80.1	78.8
Random patches*	83.3	84.2	84.5	80.2

As can be seen, the experiments carried out are the same as those performed to obtain the results of the 2019 datasets. Once again, those marked with ('*') were those in which Dice weighted sum and cross entropy were used as loss function instead of only Dice.

In view of these results, it can be seen that they are substantially better than in the cases of the previous datasets. This is mainly due to the large increase in the number of samples available in this dataset. Again, the strategy of training with randomly obtained patches across volumes has yielded better results than training with balanced patches. Considering the amount of data that have been used, it could be deduced that indeed the strategy in which the patches are balanced is not the best for this task, so for future work we will develop a different one that manages to consistently improve the random strategy.

Considering the results achieved, the model obtained with random patches with weighted sum of Dice loss and cross entropy loss as loss function has been used as the base model for the annotator. However, as explained in the ITK-Snap integration section, it should be noted that the system is flexible when using any other model. In addition, it is important to remember that the models obtained in this section must be portable in order to be integrated into the annotation platform. That is why the division into patches is made and why the decrease in results obtained due to not using complete images can be assumed.

6.2 Adapted models

The results of the test intended to demonstrate the effectiveness of model fitting will be shown below. For this purpose, as mentioned in the model adaptation section, a model trained with the cases obtained in several centers will be adapted to segment cases from a different center.

For this experiment, the 2019 dataset was used because the distinction of the center from which each case comes was simpler than in the rest. Thus, of the 335 cases contained in the dataset, 100 corresponding to a given center were divided into 70 for model retraining and 30 for test evaluation. Of the remaining 225 samples belonging to different centers other than the previous one, they were also divided into 70 to maintain the same number of retraining samples as the previous case and a base model was created with the remaining 155.

Thus, two trials will be performed: retraining the base model trained with the 155 cases from different centers with the 70 cases, both from the test center and the others, which will be progressively added to observe the variation of performance with the number of additional samples. In both cases, the same test set of 30 cases belonging to the target center of the adaptation will be used.

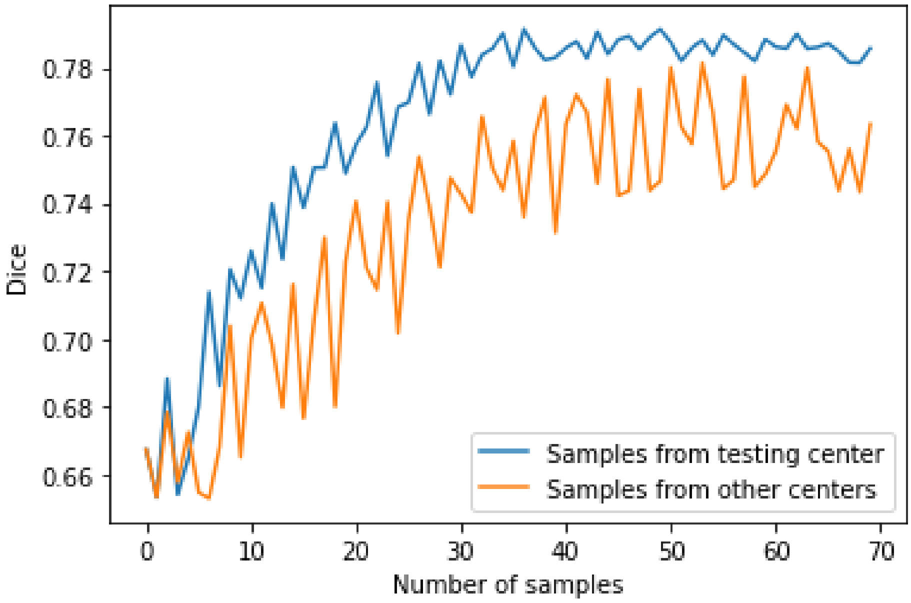


Figure 27 - Adaptation of the model to a center. Source: Own elaboration.

As expected, if the base model is retrained with more training samples, the results will improve regardless of the center from which the images come because more cases can be learned. However, this improvement is superior if the model is retrained directly with cases from the center from which the segmentations are trying to be predicted. The results of this test can be seen in Figure 27.

In view of these results, and considering as error 1 - Dice, the adaptation of the model to the cases of a specific center has had an average error improvement of 10%. These results reinforce the hypothesis that the specialization of the models to the different areas of use or protocols of each center can help to improve the models over time.

6.3 Final user validation

In order to validate the tool created throughout this work, Karoline Skogen, PhD senior neuroradiologist in charge of the review and editing of the segmentations in the ImPRESS clinical trial (ClinicalTrials.gov Identifier: NCT03951142) was consulted.

Her team already had the possibility of using deep learning to obtain segmentation masks of gliomas in postoperative cases, since, as a professional in the field, she states that the use of automatic segmentation does indeed make the task of segmenting brain tumors easier. However, the models they used are in-house models developed at Oslo University Hospital, which have not been trained with post-chirurgical samples. Not having post-chirurgical samples to train the models is the main reason why substantial errors are generated during their segmentation. For this reason, this utility was scored with a value of 5 on a Likert scale of 0 to 10. This score is given to the models they are currently working with because there is a wide dispersion in the results obtained, i.e., there are cases where the results are quite good and others where they are not.

This is one of the areas that the current proposal seeks to reinforce through the adaptation of models, allowing the models to learn from the corrections, especially in cases where they do not work too well, in order to improve progressively.

As for our tool, Karoline considered that the choice of ITK-Snap as the base annotation platform in which to include our segmentation support methods is appropriate. Furthermore, she stated that the system proposed in this work can be useful for brain tumor segmentation support and that it has a suitable approach.

The usability of the prototype was rated 7 on the same scale as above. This score is due to the possibility that the radiologist may not have used ITK-Snap before, and although it is not complicated to learn, it could entail an additional effort at the beginning.

As mentioned before, the tool is executed locally and not in a distributed way. Because of this, if the machine on which it is being executed does not have a GPU, both the inference of segmentation masks and model adaptation will necessarily be performed on the CPU. In this case, the automatic segmentation from the original MRI images could take up to minutes. In the tests performed, this time ranged from 2 to 7 minutes depending on the case and the model being used. For model fitting, this time was slightly longer, although it is worth noting that the retraining settings allow the time that could be taken to be greatly adjusted. In any case, a time of approximately 5 minutes was considered reasonable for the task, both for automatic segmentation and model fitting.

On the other hand, the idea of managing the models manually through directories, as explained in the previous section, was initially designed as an alternative to the functional software, until another alternative could be found to automate the process. However, this possibility of self-managing the re-trained models seemed to be useful, so it could persist in future versions.

In addition, the need to implement the system in a secure environment for clinical data was stated. Since the current version is a local execution version, this will depend directly on the machine on which the user installs the system but should be carefully considered for possible future versions in distributed environments.

7. Conclusions

With all that has been said throughout the study, it could be considered that the objective set at the beginning of the study has been achieved. Based on the feedback received, and the results seen in the evaluation, it has been possible to create a tool that can be useful for health professionals in the task of glioma segmentation. However, the measure to which the sub-objectives that were set out at the beginning of the work have been achieved or not should be qualified.

As for the first of the subjective, the creation of automatic glioma segmentation models, not one but 3 datasets have been explored for their development. In view of the results presented in section 6.1, we have achieved models close to the state of the art that, although inferior to these, are within an acceptable range for use by healthcare personnel and, above all, are portable models that could be used in practically any minimally up-to-date machine.

With regard to the second objective, concerning model adaptation, we have managed to develop a system capable of adapting pre-trained models to a specific area. The results presented in section 6.2 show how a model can be improved by using data from a given context. In addition, it should be noted that based on the feedback obtained, this is a utility that is declared of high interest for radiologists, since it could be of great use in allowing them to automatically segment post-surgical cases and obtain acceptable results.

In terms of providing a friendly and simple environment for the radiologist, i.e., the third and last sub-objective raised, the choice of ITK-Snap as the annotation platform seems to have been a success due to its simplicity. Even if the radiologist has not used this software before, learning to use it does not pose a real problem, as discussed in the section on validation by the end user. Furthermore, considering the portability of the tool and that the system runs locally, there will be cases where inference and/or retraining is performed on the CPU and not on the GPU. In that case the execution could take up to 5-10 minutes depending on the case, which is within a reasonable range of time that the radiologist would be willing to wait. Thus, as discussed in section 6.3, it seems that the tool could indeed be useful for healthcare professionals and that the approach is correct in respect to its usability.

In another order of ideas, relating the current work with the Sustainable Development Goals (SDGs) proposed by the United Nations, there is an inherent link between the work and SDG number 3, health, and well-being, due to the nature of the work. In addition, goal number 9, industry, innovation, and infrastructure, would also be present, as the work seeks to bring a novel solution to the state of the art by creating something new.

In short, the developed tool can be useful in the context of specific clinical studies where radiologists perform volumetric segmentations, which is our target. However, it could even go further, since the improvement of segmentation models based on manual correction of masks can provide models with margins of error close to those of an experienced radiologist. Thus, work such as this may result in models reliable enough to be used in regular clinical practice.

8. Future works

After completing the work that was proposed to be developed throughout this master's thesis, different lines of future work are opened with which the tool could be improved.

For example, the fact that the current proposal is for local execution, as mentioned above, could lead to the execution being done on the CPU, taking up to minutes. Although this does not necessarily have to happen, since it is not uncommon for radiologists to work on a workstation equipped with GPU, there is room for improvement. This could be solved by using the same solution on the cloud instead of local execution, always being able to take advantage of the use of GPUs. This alternative would also remove the restriction of using patches (if the hardware resources were sufficient) and would allow the use of networks based on more complex architectures, since for the current work excessively complex networks could not be used because their utilization on CPUs would take too long. Although it should be noted that a cloud solution would also open up a series of new problems, especially related to data security, since the data is sensitive due to its medical nature, as well as making it difficult for the user to manage the models created with the tool.

On the other hand, other network architectures could be explored, for example, by including attention models or even experimenting with vision transformers. However, as discussed above, it should be noted that if the local execution environment is maintained, the network complexity would have to be taken into consideration.

In addition, we could also try to optimize the parameterization used when adjusting the pre-trained models with segmentations corrected by the radiologist. We could even create a system that dynamically modifies which layers are frozen and which are not depending on certain variables in order to improve the performance of the models as much as possible.

Finally, it is important to note that all these proposals will continue to be studied beyond this master's thesis in a process of continuous improvement of the tool created, considering a possible future doctoral thesis together with the BDSLab team of the UPV.

Bibliography

[1] N. M. Borden, S. E. Forseen, C. (Medical E. C. Stefan, and A. J. E. Moore, "Imaging anatomy of the human brain : a comprehensive atlas including adjacent structures," Accessed: Jun. 17, 2022. [Online]. Available: https://books.google.com/books/about/Imaging_Anatomy_of_the_Human_Brain.html?hl=es&id=-RyDCgAAQBAJ.

[2] E. C. Holland, "Progenitor cells and glioma formation," *Curr. Opin. Neurol.*, vol. 14, no. 6, pp. 683–688, 2001, doi: 10.1097/00019052-200112000-00002.

[3] D. N. Louis *et al.*, "The 2016 World Health Organization Classification of Tumors of the Central Nervous System: a summary," *Acta Neuropathol.*, vol. 131, no. 6, pp. 803–820, Jun. 2016, doi: 10.1007/S00401-016-1545-1.

[4] R. Chen, M. Smith-Cohn, A. L. Cohen, and H. Colman, "Glioma Subclassifications and Their Clinical Significance," *Neurotherapeutics*, vol. 14, no. 2, pp. 284–297, Apr. 2017, doi: 10.1007/S13311-017-0519-X/TABLES/2.

[5] M. S. Berger, S. Hervey-Jumper, and W. Wick, "Astrocytic gliomas WHO grades II and III," *Handb. Clin. Neurol.*, vol. 134, pp. 345–360, Jan. 2016, doi: 10.1016/B978-0-12-802997-8.00021-9.

[6] L. Gately, S. A. McLachlan, J. Philip, J. Ruben, and A. Dowling, "Long-term survivors of glioblastoma: a closer look," *J. Neurooncol.*, vol. 136, no. 1, pp. 155–162, Jan. 2018, doi: 10.1007/S11060-017-2635-1.

[7] M. A. Flower, *Webb's Physics of Medical Imaging, Second Edition*. CRC Press, 2016.

[8] A. Oppelt, "Imaging systems for medical diagnostics : fundamentals, technical solutions and applications for systems applying ionizing radiation, nuclear magnetic resonance and ultrasound," *Publicis Pub, Erlangen*, p. 996, 2005.

[9] R. Kates, D. Atkinson, and M. Brant-Zawadzki, "Fluid-attenuated Inversion Recovery (FLAIR)," *Top. Magn. Reson. Imaging*, vol. 8, no. 6, p. 389???396, Dec. 1996, doi: 10.1097/00002142-199612000-00005.

[10] A. Hawkins-Daarud, R. C. Rockne, A. R. A. Anderson, and K. R. Swanson, "Modeling tumor-associated edema in gliomas during anti-angiogenic therapy and its impact on imageable tumor," *Front. Oncol.*, vol. 3 APR, p. 66, 2013, doi: 10.3389/FONC.2013.00066/BIBTEX.

[11] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[12] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, “Review the state-of-the-art technologies of semantic segmentation based on deep learning,” *Neurocomputing*, vol. 493, pp. 626–646, Jul. 2022, doi: 10.1016/J.NEUCOM.2022.01.005.

[13] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, “A review of semantic segmentation using deep neural networks,” *Int. J. Multimed. Inf. Retr.*, vol. 7, no. 2, pp. 87–93, Jun. 2018, doi: 10.1007/S13735-017-0141-Z/FIGURES/3.

[14] E. J. van Kempen *et al.*, “Performance of machine learning algorithms for glioma segmentation of brain MRI: a systematic literature review and meta-analysis,” *Eur. Radiol.*, vol. 31, no. 12, pp. 9638–9653, Dec. 2021, doi: 10.1007/S00330-021-08035-0/TABLES/2.

[15] Z. Jiang, C. Ding, M. Liu, and D. Tao, “Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11992 LNCS, pp. 231–241, 2020, doi: 10.1007/978-3-030-46640-4_22/TABLES/4.

[16] M. P. Schilling *et al.*, “Automated Annotator Variability Inspection for Biomedical Image Segmentation,” *IEEE Access*, vol. 10, pp. 2753–2765, 2022, doi: 10.1109/ACCESS.2022.3140378.

[17] G. Chartrand *et al.*, “Deep learning: A primer for radiologists,” *Radiographics*, vol. 37, no. 7. Radiographics, pp. 2113–2131, Nov. 01, 2017, doi: 10.1148/rg.2017170077.

[18] M. Z. Khan, M. K. Gajendran, Y. Lee, and M. A. Khan, “Deep Neural Architectures for Medical Image Semantic Segmentation: Review,” *IEEE Access*, vol. 9, pp. 83002–83024, 2021, doi: 10.1109/ACCESS.2021.3086530.

[19] S. Santurkar, D. Tsipras, A. Ilyas, and A. M. c A. Mit, “How Does Batch Normalization Help Optimization?”

[20] A. Hatamizadeh *et al.*, “UNETR: Transformers for 3D Medical Image Segmentation,” *Proc. - 2022 IEEE/CVF Winter Conf. Appl. Comput. Vision, WACV 2022*, pp. 1748–1758, Mar. 2021, doi: 10.1109/WACV51458.2022.00181.

[21] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[22] A. D. Yao, D. L. Cheng, I. Pan, and F. Kitamura, “Deep learning in neuroradiology: A systematic review of current algorithms and approaches for the new wave of imaging technology,” *Radiol. Artif. Intell.*, vol. 2, no. 2, Mar. 2020, doi: 10.1148/RYAI.2020190026.

[23] KeitoTobi1, “tzutalin/labelImg: LabelImg is a graphical image annotation tool and label object bounding boxes in images,” *GitHub*, 2017. https://github.com/tzutalin/labelImg (accessed Apr. 28, 2022).

[24] Intel Corporation, “openvinotoolkit/cvat: Powerful and efficient Computer Vision Annotation Tool (CVAT),” 2021. https://github.com/openvinotoolkit/cvat (accessed Apr. 28, 2022).

[25] BWH and 3D Slicer contributors, “3D Slicer image computing platform | 3D Slicer,” 2022. https://www.slicer.org/ (accessed Apr. 28, 2022).

[26] P. A. Yushkevich *et al.*, “User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability,” *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, Jul. 2006, doi: 10.1016/j.neuroimage.2006.01.015.

[27] K. A. Philbrick *et al.*, “RIL-Contour: a Medical Imaging Dataset Annotation Tool for and with Deep Learning,” *J. Digit. Imaging*, vol. 32, no. 4, pp. 571–581, Aug. 2019, doi: 10.1007/s10278-019-00232-0.

[28] P. Yushkevich, “ITK-SNAP Distributed Segmentation Service (DSS),” *ITK-snap*, 2018. https://alfabis-server.readthedocs.io/en/latest/ (accessed Apr. 28, 2022).

[29] S. Bakas *et al.*, “Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge,” *Sandra Gonzlez-Vill*, vol. 124, Nov. 2018, doi: 10.48550/arxiv.1811.02629.

[30] S. Bakas *et al.*, “Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features,” *Sci. data*, vol. 4, Sep. 2017, doi: 10.1038/SDATA.2017.117.

[31] B. H. Menze *et al.*, “The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS),” *IEEE Trans. Med. Imaging*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015, doi: 10.1109/TMI.2014.2377694.

[32] U. Baid *et al.*, “The RSNA-ASNR-MICCAI BraTS 2021 Benchmark on Brain Tumor Segmentation and Radiogenomic Classification,” 2021, Accessed: Jun. 13, 2022. [Online]. Available: http://arxiv.org/abs/2107.02314.

[33] A. Crimi, S. B. Eds, and G. Goos, “Brainlesion : Glioma , Multiple Sclerosis , Stroke and Traumatic Brain Injuries Lecture Notes in Computer Science,” no. May, pp. 327–337, 2020, doi: 10.1007/978-3-030-46640-4.

[34] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[35] M. Consortium, “MONAI: Medical Open Network for AI.” Feb. 16, 2021, doi: 10.5281/ZENODO.6114127.

[36] F. Pérez-García, R. Sparks, and S. Ourselin, “TorchIO: A Python library for efficient loading, preprocessing, augmentation and patch-based

sampling of medical images in deep learning," *Comput. Methods Programs Biomed.*, vol. 208, Mar. 2021, doi: 10.1016/j.cmpb.2021.106236.

[37] NVIDIA and D. Merkel, "Containers For Deep Learning Frameworks :: NVIDIA Deep Learning Frameworks." https://docs.nvidia.com/deeplearning/frameworks/user-guide/index.html (accessed May 25, 2022).

Appendices

Below is a link to a video uploaded to YouTube with a practical demo of the work performed.

https://www.youtube.com/watch?v=z3tIEXMclu4

In the video you can see how, first of all, an ITK-Snap workspace is opened where the three images corresponding to Flair, T1gd and T2 are open.

After this, the ITK-Snap Distributed Segmentation Service is opened and connected to the localhost, where our service is running on docker. Then the multichannel segmentation task is selected, which will return another workspace with the segmentation mask for that particular case.

Finally, the user selects the model adaptation service, where the corrected mask will be added. In this case, this mask has been added to the one that was already available a priori in order not to spend more time in the demo correcting the mask obtained by the initial model.

As you can see on the left side, at the end of the script a new model has been generated in the 'adapted' directory, as a result of the adaptation of the base model to this case. In order to use this new model, it will be enough to drag this model to the 'current' folder, and the segmentation service will automatically take this model for future segmentations (or even new adaptations).