UNIVERSITY OF BUCHAREST

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

ARTIFICIAL INTELLIGENCE

Master's Thesis

APPLYING SEGMENT ANYTHING MODEL ON MEDICAL IMAGE SEGMENTATION

Student

Dragomir Elena Alexandra

Coordinated by

Conf. Dr. Dumitru-Bogdan Alexe

Asist. Drd. Ciprian-Mihai Ceaușescu

Bucharest, June 2024

Abstract

The fast development of Artificial Intelligence models and their performances in recent years increases the potential to enhance various fields, including medicine. Medical image segmentation is a critical task in the analysis and diagnosis of various medical conditions, particularly in identifying and delineating tumors. The scope of this thesis is to employ a segmentation model for brain tumors that could be further used in clinics. The experiments are based on the Segment Anything Model and its fine-tuning on the BraTS dataset. We propose three scenarios depending on the level of human contribution. For a full automatic pipeline, a YOLO detection model and SAM obtain Dice scores of 0.70, 0.75 and 0.79 for enhancing tumor, tumor core and whole tumor respectively. If we add a minimal intervention from clinicians in form of a bounding box with the region that should be segmented for a few patients with unclear tumors, the scores are increasing to 0.77, 0.85 and 0.79. If all bounding boxes are provided, the scores are 0.76, 0.85 and 0.82. These scores are approaching current state-of-the-art models, showing the potential of SAM in this area.

Rezumat

Dezvoltarea rapidă a modelelor de Inteligență Artificială și îmbunătățirea performanțelor acestora în ultimii ani au crescut semnificativ potențialul de a ajuta diverse domenii, inclusiv medicina. Segmentarea imaginilor medicale reprezintă o sarcină esențială în analiza și diagnosticarea diferitelor afecțiuni medicale, în special pentru identificarea și delimitarea tumorilor. Scopul acestei teze este de a folosi un model de segmentare pentru tumorile cerebrale care ar putea fi utilizat ulterior în clinici. Experimentele se bazează pe modelul Segment Anything și pe antrenarea suplimentară a acestuia pe setul de date BraTS. Propunem trei scenarii în funcție de nivelul de contribuție din partea medicilor pe care îl dorim. Pentru un pipeline complet automat, un model de detectare YOLO împreună cu SAM obțin scoruri Dice de 0.70, 0.75 și 0.79 pentru clasele ”enhancing tumor”, ”tumor core” și ”whole tumor”. Dacă adăugăm o intervenție minimă din partea clinicienilor sub forma unor fereste de ground truth pentru regiunea ce trebuie segmentată pentru câțiva pacienți unde detecția s-a dovedit a fi mai dificilă, scorurile cresc la 0.77, 0.85 și 0.79. Atunci când sunt furnizate toate ferestele de ground truth, scorurile obținute sunt de 0.76, 0.85 și 0.82. Aceste scoruri apropiate de state-of-the-art indică potențialul pe care îl are SAM în zona medicală.

Contents

List of Figures

List of Tables

1 Introduction

Image segmentation has been consistently an essential technique in the field of computer vision and image preprocessing. Haralick defined it in 1992 as ”the partition of an image into a set of non-overlapping regions whose union is the entire image” [1]. Its wide range of applications has made image segmentation a constant focus in the research field, with remarkable developments over time. An early method of image segmentation was thresholding, proposed by Otsu in 1975 [3], where objects were delineated from gray-level histograms using different thresholds. K-means clustering was also used for this problem by Dhanachandra in [4] and its scope was to delimit an area of interest from the background by grouping similar pixels in clusters. Nowadays, the techniques had evolved to advanced models such as U-Nets [5] or even the foundational model Segment Anything [26], which we discuss more in detail in Section 3.1.

Image segmentation is divided into several categories: semantic, instance and panoptic segmentation. Semantic segmentation refers to the process of labeling pixels with a set of object categories while instance segmentation has the scope to detect and label each object of interest from a particular image [2]. For example, with semantic segmentation we can detect different objects from an image as a mask, each pixel from a category labeled the same, while instance segmentation should consider different instances of an object with different labels. This can be better understand using Figure 1.1.

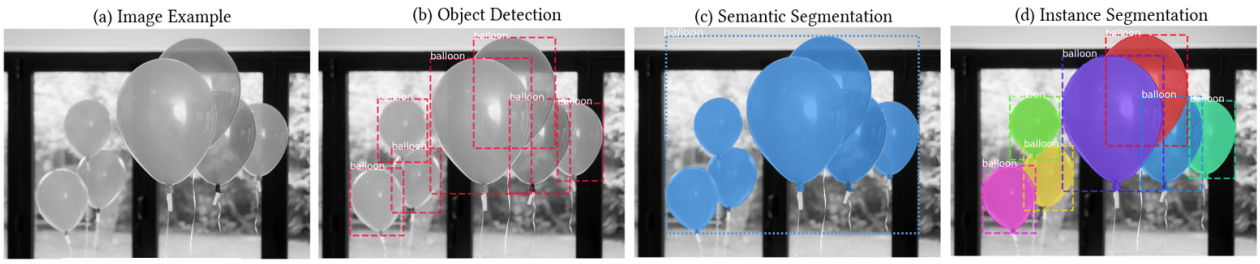

Figure 1.1: Object detection and segmentation types. *Example of object detection and different types of segmentation on an image (figure from [6])*

For the case of semantic segmentation we have all balloons colored the same, while in instance

segmentation we have the balloons colored individually. It should also be noted that both these methods are different than object detection, which only detects where an object is localized, as a bounding box, without the exact contour.

Panoptic segmentation refers to a combination of the other two mentioned above: all objects in an image should be labeled and different instances will correspond to different values [7].

As for the popular models used in the area, the development of deep learning created a new spectrum of image segmentation models that showed remarkable improvement regarding performance. In [2] there is a survey of the current advances of deep-learning-based segmentation methods including convolutional networks, encoder-decoder models, R-CNN, recurrent neural networks, generative models and adversarial training and attention-based models.

1.1 Medical image segmentation

Medical image segmentation primarily refers to the identification of organs or abnormalities from modalities like MRI, CT scans, and X-rays. These models can play a significant part for an efficient disease diagnosis and treatment, helping or even improving clinicians' diagnostics [8].

In a recent study [8] the authors present an analysis of the current image segmentations models in the medical field and they highlight that convolutional neural networks (CNNs) and U-Nets are dominating the field.

U-Nets were first proposed by Ronneberger in [5] and provided the winning solution for the ISBI cell tracking challenge in 2015, becoming state-of-the-art at the time [9]. The U-Net architecture features two major components: a contracting and an expanding path. The encoder extracts features similarly to regular convolutional networks, while the decoder upsamples and combines these features to learn localized classification information and create a detailed segmented output. The architecture is nearly symmetrical, resembling a U-shape, and primarily focuses on classifying the entire image into a single label. Its architecture can be better seen in Figure 1.2.

The introduction of U-Nets in 2015 significantly impacted medical imaging, leading to the development of numerous enhanced versions such as UNet++, Attention-UNet, TransUNet and

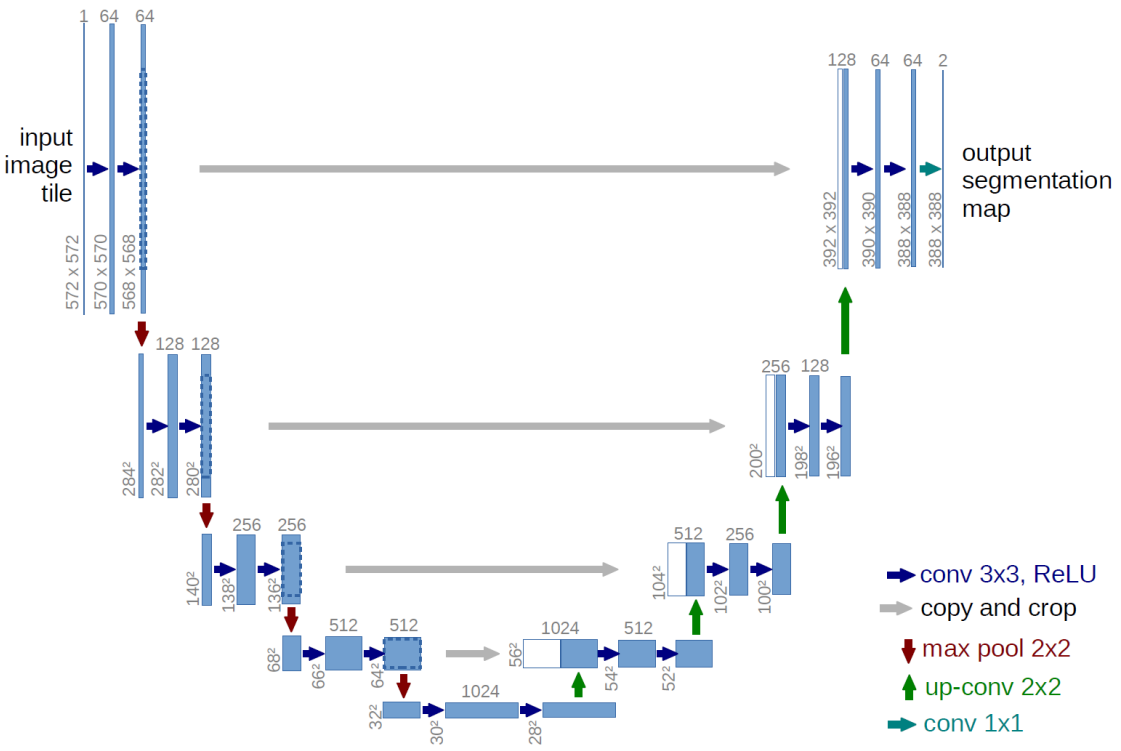

Figure 1.2: U-Net architecture. *Operations are represented by arrows, feature maps by blue boxes and the copies of the feature maps by white boxes (figure from [5])*

Swin-Unet. They are built upon the original architecture of U-Net while different techniques were added to improve efficacy depending on the specific task at hand. These techniques include attention mechanisms and transformer modules [8].

1.2 Structure of the thesis

However, the scope of this thesis does not cover the U-Net model but a new foundation model for image segmentation, Segment Anything, that we shall adapt for a medical task. Our primary goal is developing a pipeline to segment brain tumors, more specifically on the Brain tumor segmentation dataset (BraTS), presented in the next section. In Section 3.1 we show the architecture of the SAM model and its current applications in medical segmentation, along with the proposed method for a full pipeline that first detects the region of the tumor using YOLO and then segment it with the adapted SAM.

2 Brain Tumor Segmentation Dataset

BraTS [12–15] is a widely used dataset in the research community for detecting brain tumors. It is composed of MRI scans using several modalities: T1 weighted (T1), T1 weighted post contrast (T1ce), T2 weighted (T2) and Fluid Attenuated Inversion Recovery (FLAIR), each highlighting different regions relevant to a brain tumor by using different contrast solutions. In Figure 2.1 there is an example of how the four modalities are differentiated on an image.

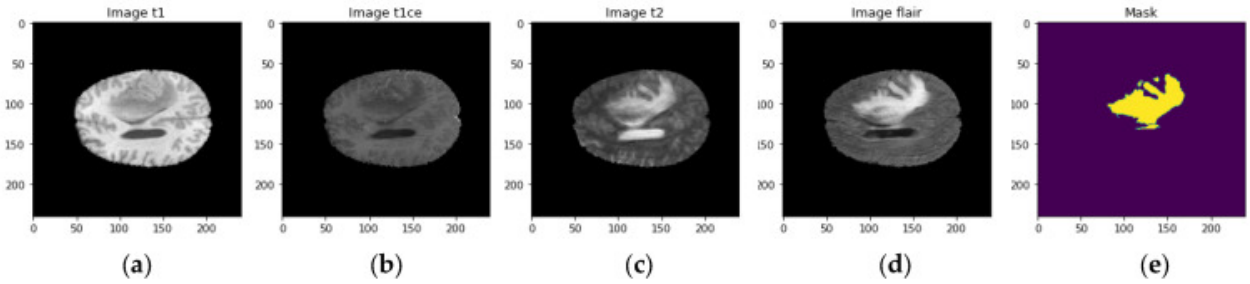

Figure 2.1: MRI modalities. *An image captured in the four available modalities: a. T1, b. T1ce, c. T2, d. FLAIR, e. ground truth mask (figure from [16])*

Since its first apparition in 2012, the BraTS dataset has enhanced and expended during the years. In 2012 there were included clinical and synthetic MRIs from only 20 patients, while in later years there were only clinical data and the number of patients increased up to 369 in 2020, which is the version on which we experimented in this thesis. We use the version provided from Kaggle [10] that contatins 369 MRIs fro training and 125 for validation. Since the validation data is not annotated, we use only the trainig data for train, validation and test. Each MRI contains 155 brain slices for each modality. An overview of a whole image is shown in Figure 2.2.

Each slice has an associated mask with the ground truth data that contains three labels for different tumor regions: GD-enhancing tumor (ET, label 4), peritumoral edema (ED, label 2) and the necrotic and non-enhancing tumor core (NCR/NET, label 1) [10]. The ET and NCR/NET are especially hyper-intense in T1 weighted post contrast method, while ED is more visible in FLAIR. In Figure 2.3 we have an example of a slice that contains all the tumor labels and the associated mask, represented with both T1ce and FLAIR modalities. It is also highlighted how the ED is more

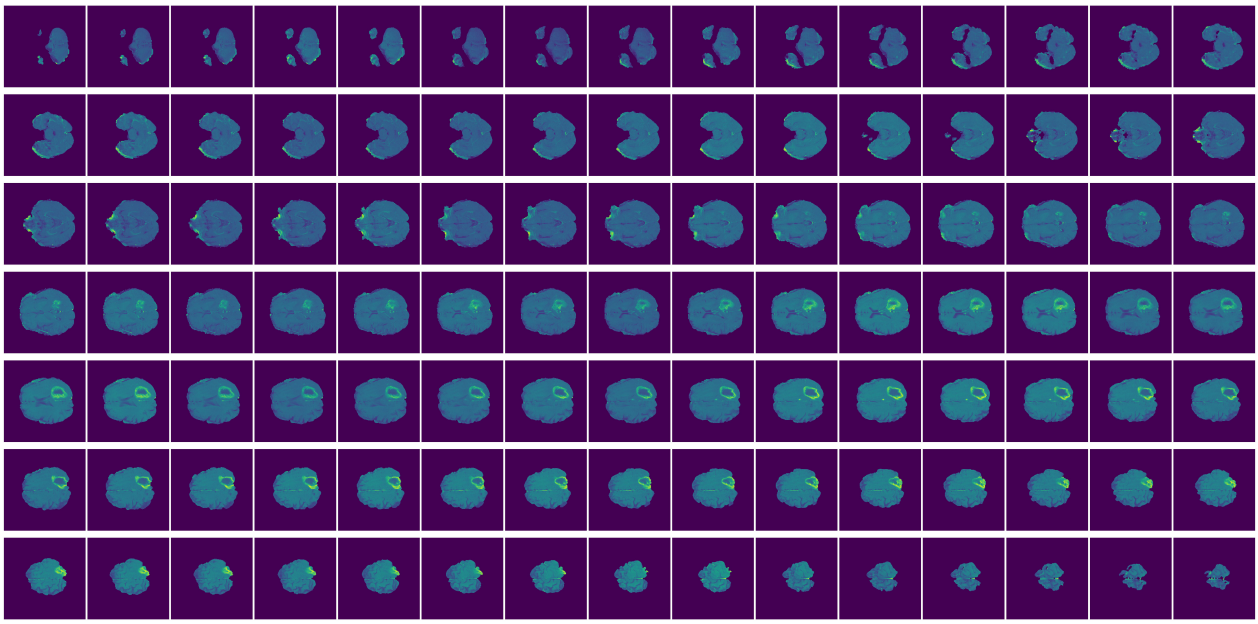

Figure 2.2: Example of MRI volume. *Subset of the slices from a MRI volume*

visible in FLAIR and ET in T1ce.

Usually, these subregions of the tumor are combined to form three significant tumor subparts: enhancing tumor (ET), tumor core (TC - ET and NCR/NET) and whole tumor (WT - all three labels) [11]. Most studies on BraTS evaluate the results on these three subregions.

The multimodal brain tumor image segmentation (BraTS) challenge was first introduced in 2012 during the international conference on Medical Image Computing and Computer Assisted Intervetion (MICCAI) [12; 17]. Ghaffari et al. provide an in-depth analysis of the models proposed in the challenge over the years in [17]. In 2012 the methods were primarily based on Random Forests classifiers, logistic regression, and Markov Random Field and the Dice scores (the main metric used in evaluation) varies from 0.14 to 0.7 for whole tumor and 0.09 to 0.37 for tumor core. In 2013 and 2014 the random forests were still frequently used, but Convolutional Neural Networks (CNNs) had started to see a growth in the area and therefore started to appear in the solutions of the challenge as well. Since the introduction of U-Nets by Ronneberger et al. in 2015 [5], they have served as the foundation of multiple solutions, achieving great results over the years.

In 2017, Kamnitsas et al. [18] proposed the winning solution using an Ensamble of Multiple

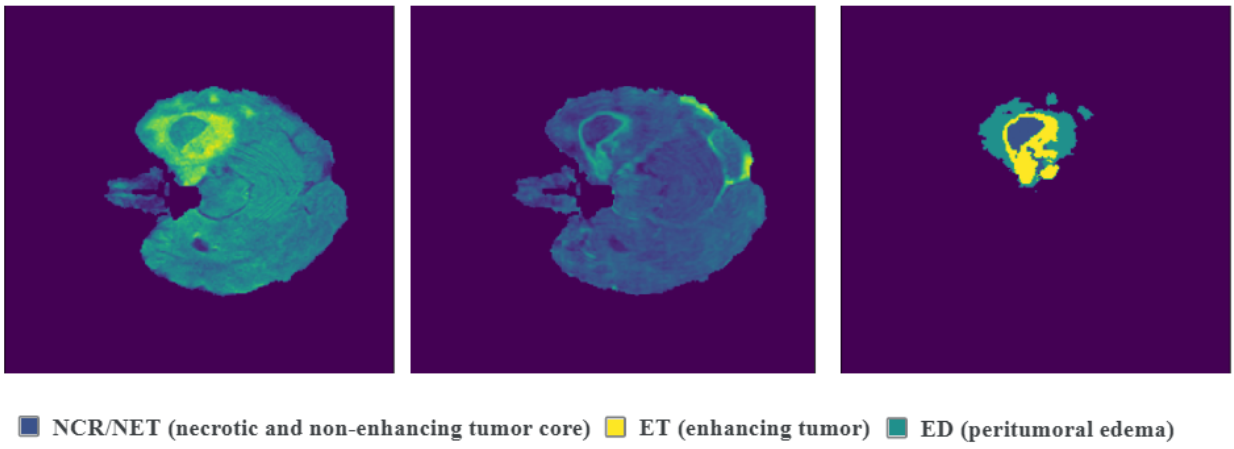

Figure 2.3: BraTS tumors on an image. *ET, ED and NCR tumors in one image. First image is captured using the modality FLAIR, second image is using T1ce and third image represents the ground truth mask*

Models and Architectures (EMMA), including the 3D convolutional networks DeepMedic, FC and U-Net and the Dice scores achieved were 0.90 for the whole tumor, 0.82 for the tumor core, and 0.75 for the enhancing tumor.

Myronenko [19] won first place in 2018 for the BraTS challenge with an asymmetric U-Net architecture. This model featured an expanded encoder to extract features of images and a compact decoder to reconstruct the label. The Dice scores were 0.91 for WT, 0.86 for TC and 0.82 for ET.

In 2019 Jiang et al. [20] submitted an innovative two-stage cascaded U-Net and won first place with Dice scores of 0.88, 0.83 and 0.83 for WT, TC and ET and Hausdorff distances of 4.61, 4.13 and 2.65.

From 2020 to 2022 the winning solutions were based on nnU-Net [21] and its derivatives or ensambles with DeepSeg and DeepSCAN (2022 solution [22]) while in 2023 the winning authors [22] added, besides the ensambles of models, an unconventional mechanism for data augmentation using generative adversarial networks (GANs).

All these solutions use all the modalities provided by the BraTS datasets (T1, T2, T1ce and FLAIR) by stacking the four images into a tensor with 4 input channels for the neural networks. When some of the modalities are missing, the systems may not perform as well. Some work was

done in [25] and [24] in this direction. Below, in Table 2.1 we can see dice score for each label for some combination of modalities available, tested on state-of-the-art models. The results are reported for the 2018 BraTS dataset. Zhang et al. [25] propose the mmFormer (Multimodal Medical Transformer), which combines a hybrid modality-specific encoder, an inter-modal Transformer and a decoder. As we can see in the table, it achieves the best scores for enhancing tumor. Kang et al. [24] present a Multimodal feature distillation with CNN - Transformer hybrid network (MCTSeg) that achieve state-of-the-art results for TC and WT.

M	Flair / T1ce / T1 / T2	● ○ ○ ○	○ ● ○ ○	○ ○ ● ○	○ ○ ○ ●	● ● ○ ○	○ ○ ● ●	○ ● ● ○	● ○ ○ ●	○ ● ○ ●	● ○ ● ○	○ ● ● ●	● ● ○ ●	● ○ ● ●	● ● ● ○	Avg
ET	HeMIS	11.78	62.02	10.16	25.63	66.10	32.39	66.22	30.22	67.83	10.71	69.92	68.72	31.07	68.54	70.27	46.10
	U-HVED	23.80	57.64	8.60	22.82	68.36	24.29	61.11	32.31	67.83	27.96	67.75	68.93	32.34	68.60	69.03	46.76
	RobustSeg	25.69	67.07	17.29	28.97	70.30	32.01	69.06	33.84	69.71	32.13	70.10	70.88	**70.78**	70.78	71.13	51.02
	D^2-Net	8.10	66.30	8.10	16.00	64.80	16.50	70.70	17.40	68.70	9.50	68.30	66.40	19.40	65.70	68.40	42.30
	mmFormer	**39.33**	**72.60**	**32.35**	**43.05**	**75.05**	44.99	**74.04**	**47.52**	**74.51**	**42.96**	**74.75**	**75.67**	47.70	**75.47**	**77.61**	**59.85**
	MCTSeg	34.72	65.74	29.14	37.92	69.81	**69.64**	68.37	39.86	69.64	39.24	70.09	70.98	40.89	70.24	71.33	57.44
TC	HeMIS	26.06	65.29	37.39	57.20	71.49	60.92	72.46	57.68	76.64	41.12	78.96	77.53	60.32	76.01	79.48	62.57
	U-HVED	57.90	59.59	33.90	54.67	75.07	56.26	67.55	62.70	73.92	61.14	75.28	76.75	63.14	77.05	77.71	64.84
	RobustSeg	53.57	76.83	47.90	57.49	80.62	62.19	78.72	61.16	80.20	60.68	80.33	80.72	81.06	81.06	80.86	69.78
	D^2-Net	47.30	65.10	16.80	56.70	**80.80**	63.20	78.20	62.60	80.30	61.60	79.00	80.70	63.70	80.90	80.10	66.50
	mmFormer	**61.21**	75.41	56.55	**64.20**	77.88	69.42	78.59	**69.75**	78.61	65.91	80.39	79.55	71.52	79.80	**85.78**	72.97
	MCTSeg	61.05	**78.84**	**59.60**	63.97	**82.48**	**82.61**	**82.31**	69.26	**82.61**	**70.83**	**82.83**	**82.85**	**71.87**	**83.04**	82.96	**74.37**
WT	HeMIS	52.48	61.53	57.62	80.96	68.99	82.41	68.47	82.95	82.48	64.62	83.94	83.85	83.43	72.31	84.74	74.05
	U-HVED	84.39	53.62	49.51	79.83	85.93	81.56	64.22	87.58	81.32	85.71	82.32	88.09	88.07	86.72	88.46	79.16
	RobustSeg	85.69	74.93	70.11	82.24	88.51	84.78	77.18	88.28	85.19	88.24	86.01	89.27	88.73	88.73	89.45	84.39
	D^2-Net	84.20	42.80	15.50	76.30	87.50	80.10	62.10	87.90	84.10	87.30	80.90	88.80	88.40	87.70	88.80	76.20
	mmFormer	**86.10**	72.22	67.52	81.15	87.30	82.20	74.42	87.59	82.99	87.06	82.71	88.14	87.75	87.33	89.64	82.94
	MCTSeg	85.90	**75.20**	**73.92**	**82.70**	**88.88**	**86.69**	**79.96**	**89.37**	**86.69**	**88.61**	**87.33**	**90.14**	**89.83**	**89.68**	**90.31**	**84.91**

Table 2.1: SOTA for incomplete modalities. *Dice score for combinations of modalities on some state-of-the-art models. Black dots mark the available modality in each of the 14 experiments. (figure from [24])*

Segment Anything Model, the model used in this thesis, requires a single image as input and therefore we experimented with one modality for each training. We will see in later sections that comparing our results with state-of-the-art models for brain tumor segmentation (models that utilized all available modalities) is challenging. Nonetheless, our results are quite comparable to the top-performing models that use only a single modality, the ones in the table above.

3 Methods

3.1 Segment Anything Model (SAM)

Segment Anything is the first foundation model for image segmentation, released in April 2023 by Meta AI. Inspired by the way large language models are fundamentally transforming NLP with the capability of zero-shot or few-shot generalization, SAM was also built in a promptable manner and pre-trained on a vast dataset, SA-1B [26]. It includes 1 bilion masks annotated automatically by SAM during the training on 11 milion natural images.

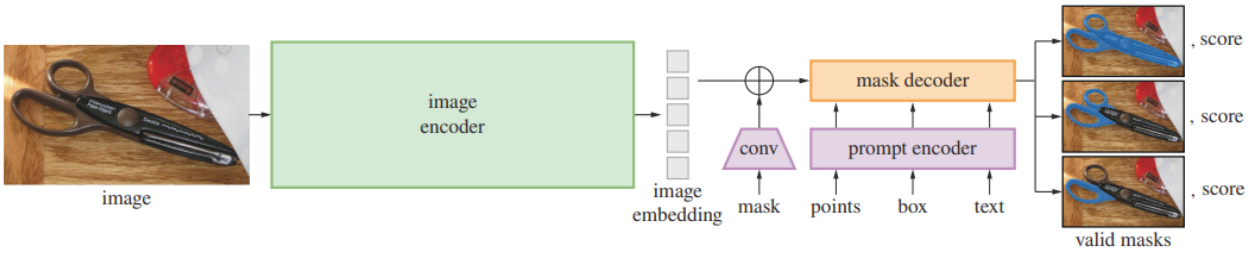

Figure 3.1: SAM architecture. *SAM oveview (figure from [26])*

In Figure 3.1 we have an overview of the SAM architecture. Shortly, the model has three main components: an image encoder, a prompt encoder (that can encode either points, box or text) and a lightweight mask decoder that uses the two information sources to predict a segmentation mask.

Image encoder: It is represented a pre-trained Vision Transformer (ViT) [28] that will return an image embedding.

Prompt encoder: Prompts should contain more information about what we specifically want to segment in an image. They can take various forms: text, a set of points indicating background or foreground, a bounding box that contains the desired object or a mask (typically from previous predictions). Positional encodings [29] are used to represent points and boxes, while the CLIP [30] text encoder is used for the text prompts. Mask are incorporated through convolutions and added to the image embedding.

Prompts' purpose is to make the task less ambiguous since there can be multiple correct way

to segment an image. For example, in Figure 3.2 we have four images and for each one we place a point in the foreground. The task is to return the mask for the object that contains the specified point. As we can see, SAM can return multiple masks for that single point. In the second image, the point can refer to either the whole person, only the bag or a small section of the bag, each being a valid mask, depending on the task.

Mask decoder: A modified version of a Transformer decoder block, as described in [31], is succeeded by a dynamic mask prediction head. This modification enables the decoder to use prompt self-attention and cross-attention, updating all embeddings from both the prompt to the image and the image to the prompt. The resulted image embeddings are upsampled and the tokens are mapped by a multilayer perceptron into a linear classifier that determines the probability of the mask's foreground [26]. A more detailed view of the decoder is observed in Figure 3.3.

Figure 3.2: Masks returned by SAM. *Example of how SAM can generate different masks from a single point (figure from [26])*

For training, the authors were inspired by [32] to do an interactive segmentation setup during training to mimic how the model will work in practice. First, they start by picking either a foreground point or a bounding box from the mask with equal probability. They uniformly sample points and the box is extracted with a random noise up to 20 pixels. The noise is added to balance between more real-life scenarios like instance segmentation or interactive segmentation. With these information, a first prediction is made and more points are chosen form the regions where errors occurred, subsequently: foreground if the point is a false negative and background otherwise, if false positive. Additionally, the

previous iteration's mask prediction is added to the prompt as an unthresholded mask. This process continued up to 16 times, without much additional computational time because of the lightweight mask decoder.

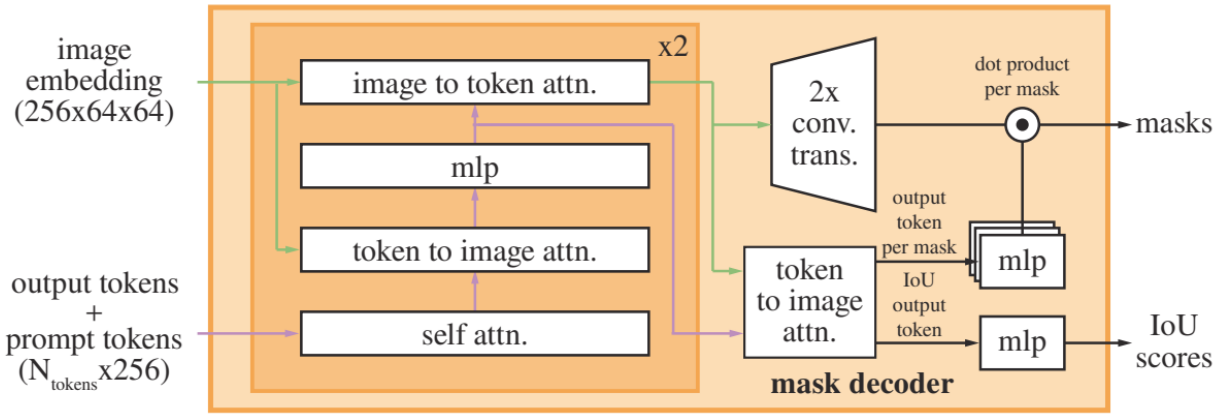

Figure 3.3: SAM mask decoder. *The lightweight mask decoder (figure from [26])*

The loss used was a linear combination of focal loss [33] and dice loss [34] in a ratio of 20:1 in favor of focal loss [26].

3.1.1 SAM in medical imaging tasks

Despite its novelty, extensive research has been done to adapt SAM (Segment Anything Model) for medical imaging problems. As of May 2024, over 150 papers have addressed this topic [35]. The authors of [35] present a detailed overview with the current adaptions of SAM and the derivative models that have emerged in the past year after SAM's release. Below, in Figure 3.4, it is a timeline with the methods appeared in 2023.

Given that there is a considerable difference between the natural images and medical ones, SAM struggles on the medical part and does not offer the same impressive zero-shot generalization. Many studies have evaluated its zero-shot performance on different types of datasets and modalities like CT scans, MRI, X-rays, revealing that SAM does not always give satisfactory results, even close to state of the art. A solution for that is therefore adapting the model for medical images by fine-tuning the entire architecture or specific components on a targeted dataset or across diverse data

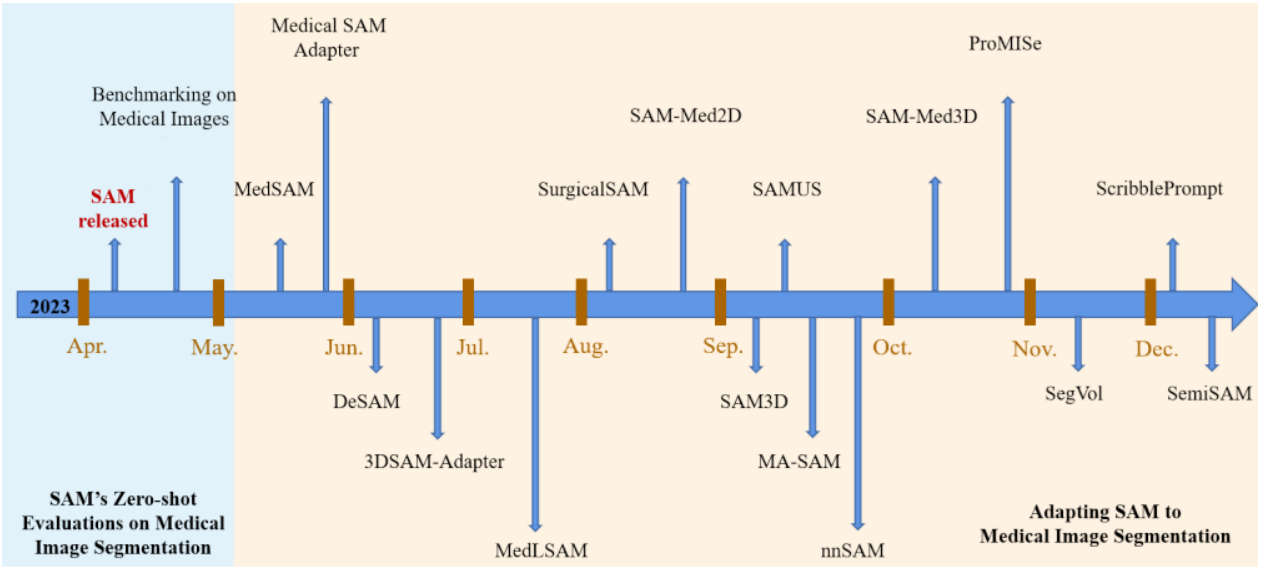

Figure 3.4: Applications of SAM in medicine. *A timeline with adaptions of SAM for medical image segmentation in 2023 (figure from [35])*

types.

A notable model whose scope was to serve as a universal medical image segmentation tool is MedSAM [36], which adapts SAM on a variety of data by compiling a diverse and extensive dataset with over one million medical image-mask pairs across 10 different modalities and 30 cancer types. The authors kept the original SAM architecture: image encoder, prompt encoder and a mask decoder. For prompts, they use the bounding boxes of tumors or organs that they aim to segment.

3.2 YOLO

YOLOv10 is the latest in the YOLO (You Only Look Once) [40] series of state-of-the-art object detection models, particularly known for its high speed and accuracy. However, we use the latest version of YOLO implemented via Ultralytics, YOLOv8 [39]. There are 5 available models, on different sizes: ”Nano”, ”Small”, ”Medium”, ”Large” and ”Extra Large” [41]. In Table 3.1 there are specified the sizes of each model. Due to the computational and time limitations, we used the ”Nano” version, which is the smallest but also the fastest of them.

YOLO model	Number of parameters
YOLOv8n	3.1 M
YOLOv8s	11.2 M
YOLOv8m	25.9 M
YOLOv8l	43.7 M
YOLOv8x	68.2 M

Table 3.1: YOLOv8 models number of parameters. *Number of parameters for "Nano", "Small", "Medium", "Large" and "Extra Large" versions of the YOLOv8 model*

We use the model "YOLOv8n" that was trained on the COCO dataset with 80 classes and fine-tune it on our BraTS dataset to detect each label of tumor: enhancing tumor (ET), peritumoral edema (ED) and the necrotic and non-enhancing tumor core (NCR/NET), as well as the concatenated regions: tumor core (ET and NET/NCR), whole tumor (all labels concatenated). Since there is a remarkable difference between the images from each modality, each highlighting different regions, we train two models of YOLO: one for FLAIR and one for T1ce.

In order to train YOLO we need to preprocess our data to have input images and "*.txt" files that contain the bounding boxes of the labels present in the image along with class, in the format "$class, x_{centre}, y_{centre}, width, height$", normalized and one per row. We extracted bounding boxes for the classes NCR, ED, ET, TC and WT and labeled them from 0 to 4 respectively. In Figure 3.5 there are shown the steps for this preprocessing. Since labels "tumor core" and "whole tumor" are the concatenations of the original labels form the dataset, their bounding boxes could coincide with some of their components' bounding boxes.

After several experiments, we observed that the default hyperparameters give the best results. The models were trained for 100 epochs. The metrics used for evaluating YOLOv8 models are precision, recall, mAP50 (mean average precision at IoU threshold of 50), map50-95 (mean average precision averaged over multiple IoU thresholds (0.5, 0.55, 0.6, ..., 0.95)) [39]. The results on test data for the two modalities can be seen in Table 3.2 and more insights about the precision, recall and F1 confidence in Figure 3.6. We are particularly interested in label WT from FLAIR and labels

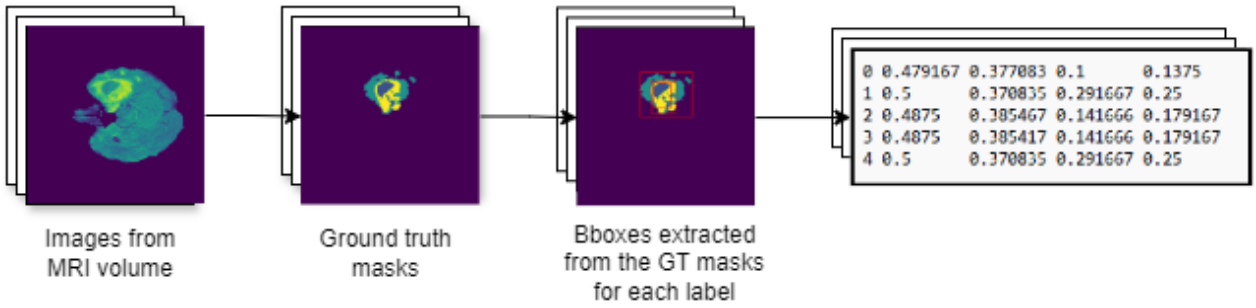

Figure 3.5: YOLO preprocessing. *From the ground truth masks we extract the bounding boxes for each label and save them, along with their class in a txt file*

ET and TC from T1ce because of how the labels are highlighted in different regions. ET and NCR/NET (that compose the tumor core - TC) are best seen in T1ce, while ED, which is the outer part of the whole tumor, is most visible in FLAIR.

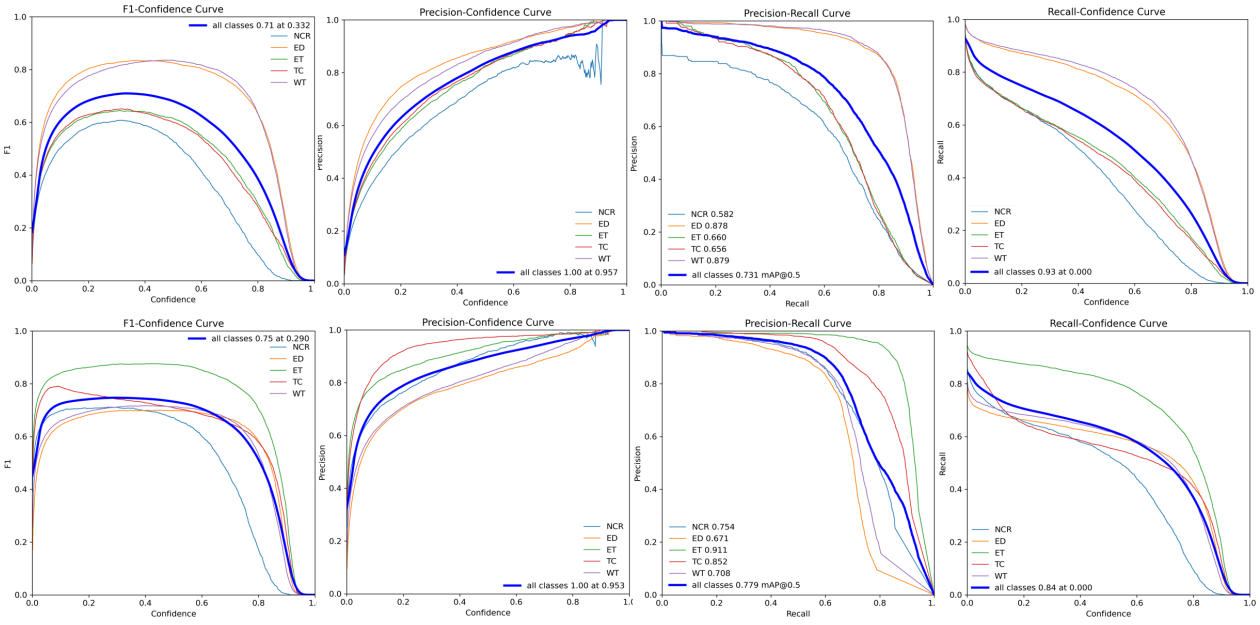

Figure 3.6: Metrics after fine-tuning YOLO on tumors in two modalities. *F1 confidence, precision confidence, recision-recall confidence and recall confidence for YOLO models fine-tuned on FLAIR (first row) and T1ce (second row)*

The plots from above show the scores of F1, precision and recall determined by the level of confidence and what is the optimal one in order to maximize the scores. For a confidence of around 0.3 F1 obtains the maximum score for both models, 0.71 in FLAIR and 0.75 in T1ce. The third

column shows the Precision-Recall curve and the value at which they would be equal: 0.73 for FLAIR and 0.78 for T1ce.

These plots are highlighting once more the boost in performance for the labels who are most visible in a specific modality. For example, on the first row, where we have plots corresponding to modality FLAIR, the outer lines in each graphs are ED and WT (their bounding boxes are almost identical in most cases) which suggest higher scores in terms of F1, precision and recall. The same happens in the second row for T1ce modality for labels ET and for TC in the precision-confidence plot.

Modality	Label	Precision	Recall	mAP50	mAP50-95
FLAIR	all	0.744	0.683	0.731	0.392
	NCR	0.649	0.566	0.582	0.214
	ED	0.83	0.834	0.878	0.566
	ET	0.713	0.586	0.66	0.308
	TC	0.734	0.582	0.656	0.301
	WT	**0.796**	**0.848**	**0.879**	**0.569**
T1ce	all	0.839	0.68	0.78	0.425
	NCR	0.825	0.623	0.754	0.339
	ED	0.76	0.645	0.671	0.311
	ET	**0.891**	**0.86**	**0.911**	**0.606**
	TC	**0.951**	**0.607**	**0.852**	**0.54**
	WT	0.767	0.666	0.709	0.33

Table 3.2: Results YOLO trainings. *Precision, recall, mAP50 and mAP50-95 on test data for fine-tuned YOLOv8*

We can observe that the precision is generally quite high while the recall is mostly at low values. This is reflected in the lack of predictions for multiple images. We have 25%, 49% and 11% of predictions missing for the labels ET in T1ce, TC in T1ce and WT in FLAIR, respectively. In Figure 3.7 we plot box plots to show the distribution of the IoU between the predictions (where, if multiple predictions were made, the one with the highest confidence for a specific label was selected) and the ground truth data from the test datasets for each label. We can observe here that the IQR (interquartile range, where the middle 50% of data is situated) lies approximately between 0.75 and 0.85 for each label. While there are a few outliers, the high IQR combined with the

minimal number of outliers indicates good precision overall for the returned predictions.

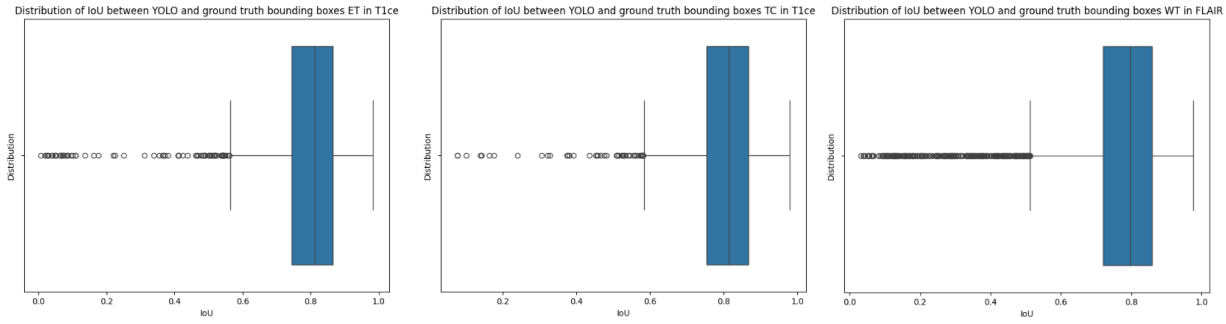

Figure 3.7: Box plots with IoU scores distribution between YOLO and ground truth data. *Label ET in T1ce (first plot), TC in T1ce (second plot) and WT in FLAIR (third plot)*

3.3 Proposed pipeline

The purpose of this thesis is to explore the adapting capabilities of SAM for a brain tumor segmentation task, using the BraTS 2020 dataset. We propose two training pipelines, presented in Figure 3.8. For a better adaption on a new dataset, it would be best to fine-tune the entire SAM architecture but, due to the computational complexity of this task, we only fine-tune the mask decoder while freezing both image encoder and prompt encoder as represented in Figure 3.8 with the snowflake symbol. For the prompt we use either the ground truth bounding boxes, either the ones obtained with the fine-tuned YOLO. Finally, along with the image embedding obtained with the image encoder, the mask decoder returns the mask of the detected tumor.

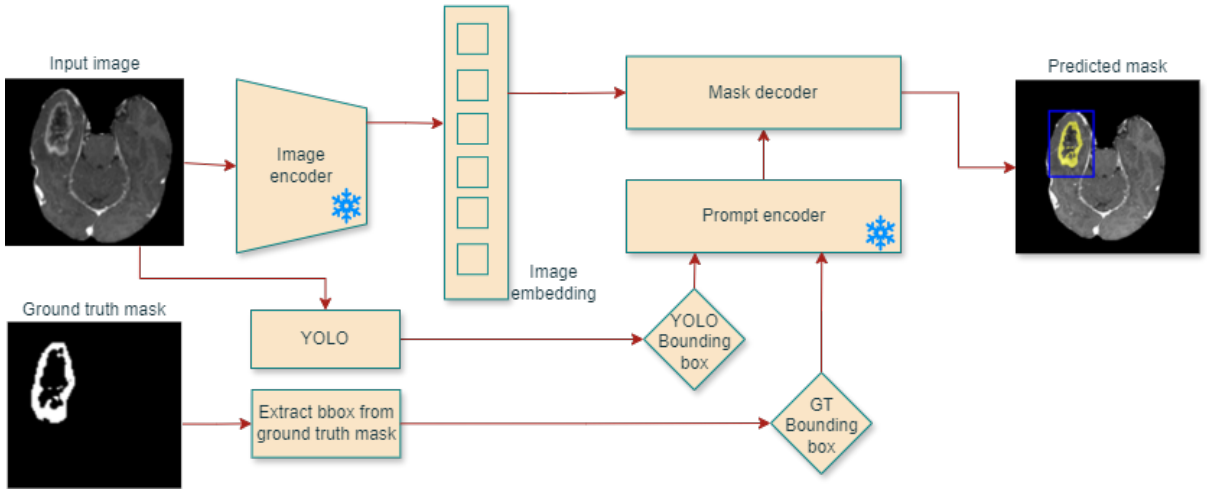

Figure 3.8: Proposed pipeline. *Architecture composed of SAM and YOLO with two options for the bounding box used by the prompt encoder: predicted by YOLO or extracted from ground truth*

4 Experimental evaluation

In this chapter we present the experiments with the two training pipelines and use the same for the testing part.

The main inspiration of this code is the GitHub repository of MedSAM [36]. The experiments highlight the differences of performance between the original SAM and our fine-tuned model on the BraTS dataset. Similar to MedSAM, we use the bounding box of the region of interest during training and testing. Since this information is not always accessible in real-life scenarios, we also added a detection part in the pipeline and compared the results between the fully and semi-supervised segmentantations.

4.1 Fine-tuning SAM decoder with ground truth bounding boxes

As previously mentioned, we fine-tune only the decoder part of the SAM architecture. The primarily used checkpoint is *'sam_vit_b_01ec64'*. There were some tests run with the checkpoint from MedSAM but the performance was bellow the original SAM. Trainings were performed for each label or combination of labels, individually.

First, the images were preprocessed and passed through the image encoder in order to obtain the image embeddings. We only selected the MRIs where there were more than 1000 pixels in total of the desired label and the slices with over 100 pixels so that we eliminate the regions that are too small and hard to detect and segment. During preprocessing, the images were normalized, converted to RGB format, resized to 256×256 from the original 240×240 and passed through SAM's preprocess function and image encoder that will eventually return the image embeddings. For each case of label we extract from the ground truth the mask corresponding to it. The embeddings are then saved along with the extracted masks.

4.1.1 Training setup

We evaluate our method using on the Dice score and Hausdorff distance.

Loss: In all our experiments we used Dice loss [34] and Hausdorff loss [37] and reached the conclusion that their linear combination with ratio 1:1 gives the best results. The authors of [34] propose this novel objective function based on the Dice coefficient D between two binary volumes defined as:

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \tag{4.1}$$

The sum runs over the N pixels of the binary segmentation volume $p_i \in P$ and the ground truth $g_i \in G$. In order to use it as an objective function, we make sure that it can be differentiated with respect to the j-th pixel of the prediction. If we denote the numerator of the fraction with f and the denominator with h we have:

$$\begin{aligned}\frac{\partial D}{\partial p_j} &= 2\,\frac{\frac{\partial f}{\partial p_j}h - \frac{\partial h}{\partial p_j}f}{h^2} \\ &= 2\,\frac{g_j\Big(\sum_i^N p_i^2 + \sum_i^N g_i^2\Big) - 2p_j\Big(\sum_i^N p_i g_i\Big)}{\Big(\sum_i^N p_i^2 + \sum_i^N g_i^2\Big)^2}\end{aligned} \tag{4.2}$$

Dice measures the overlap between the predicted segmentation and ground truth, without assigning weights to balance the two classes from the binary segmentation, background and foreground. This is especially useful in medical imaging since where the foreground (such as a tumor) is often significantly smaller than the background. The Dice loss inherently balances these classes, making it unnecessary to manually adjust weights in the loss function.

Hausdorff loss [37] derives from the average Hausdorff distance, differentiated in each point. It is a metric that measures the similarity between two sets of points considering the distance between the boundaries and it is defined as follows:

$$d_{AH}(P, G) = \frac{1}{|P|}\sum_{p\in P}\min_{g\in G} d(p, g) + \frac{1}{|G|}\sum_{g\in G}\min_{p\in P} d(p, g) \tag{4.3}$$

where P, G are the sets of the predicted and ground truth points, $|\cdot|$ represents the cardinal of the set and $d(\cdot, \cdot)$ is a metric of distance, Euclidean distance in our case. In Figure 4.1 we have a better representation of how the distance is measured. This metric is quite unstable since it can increase

drastically in case of an outlier, even if the rest of the points are close.

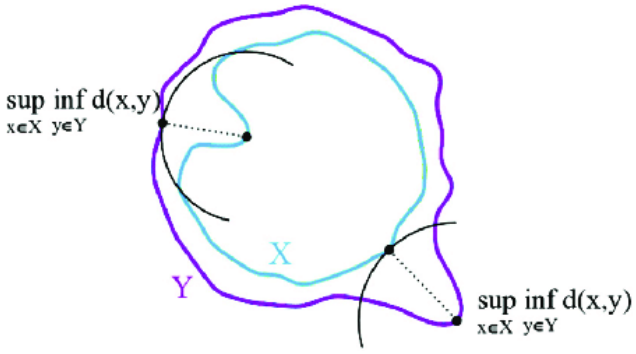

Figure 4.1: Hausdorff distance. *Figure from [38]*

Therefore, by combining these two metrics, Dice that maximizes the overlap between two segmentantions and Hausdorff that minimizes the distance between the boundaries, we should achieve a more balanced evaluation of segmentation performance that will capture a correct volume but also precise boundaries.

Optimizer: The optimizer used was Adam, with a learning rate of 10^{-5} and no weight decay.

Data: As previously mentioned, for labels of tumor ET and NCR the best modality to observe them is T1ce and for ED, FLAIR. We perform experiments on both these modalities for all the three clustered regions: enhancing tumor (ET), tumor core (TC - ET and NET/NCR) and whole tumor (WT - all three labels).

Training: During training we pass the bounding boxes obtained from the ground truth mask through the prompt encoder. Along with the image embeddings, these processed bounding boxes are used by the mask decoder to make predictions. As mentioned, we only compute gradients for the mask decoder. In order to speed up training and reduce memory, we use lower precision (float16) where possible by using automatic mixed precision (AMP). Batch size was set to 8. After training on 150 epochs we have the loss plots for train and validation data in Figure 4.2. For each label we saved the checkpoint with the lowest validation loss.

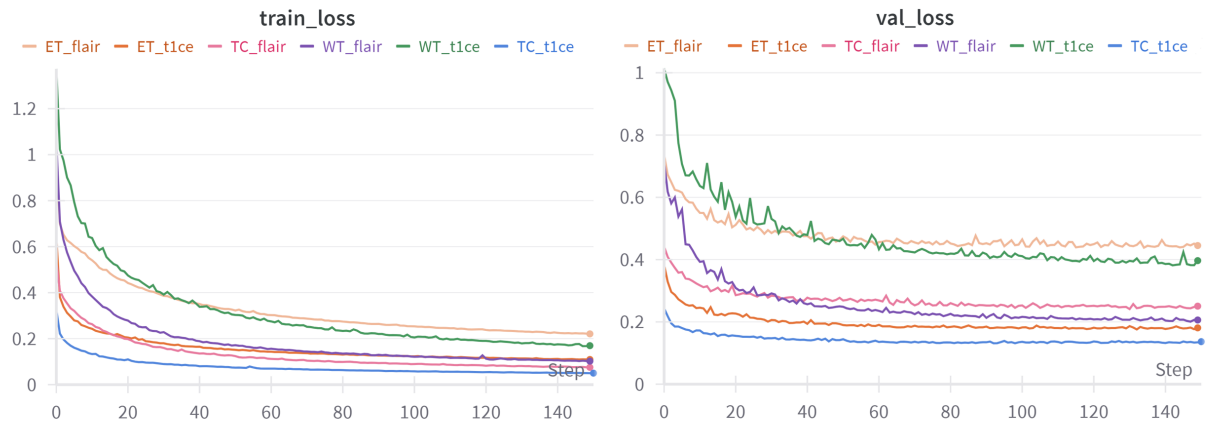

Figure 4.2: Losses for SAM fine-tuning with ground truth data. *Train and validation loss for training the labels ET, TC, WT with modalities T1ce and FLAIR*

Bounding boxes: As previously mentioned, SAM needs a prompt in order to return a prediction. For the training part we used the bounding box of the tumor. However, for the testing part we have two approaches. Currently we have access to the area of the tumor, as a bounding box, contrary to a real-life scenario. This information could eventually be given by a clinician and the model will then return the exact mask of the tumor, but we also provide a full pipeline that first detects the position of the tumor and then segment it using the fine-tuned SAM.

For the tumor detection part, we fine-tune the pretrained YOLOv8 model [39] to predict the bounding box for each tumor label. Training and testing data are the same images used for SAM training. Two YOLOv8 models were fine-tuned, one for each modality: FLAIR and T1ce, as we saw earlier in Section 3.2. A significant problem on this step is the lack of predictions for multiple images, especially for the class TC where, for 49% of data, YOLO does not return any prediction. For the ones where multiple predictions are made, it is chosen the one with the highest confidence for the specific label.

In order to solve (partially) this problem of missing labels, we perform, during inference, an additional search through the adjacent slices of a current image from the MRI for predictions. For example, we have Figure 4.3 with a batch of images with the ground truth bounding boxes and their predictions. The slices here are consecutively and we can notice that even though for the last slice

we do not have a prediction, we could take the previous one without much loss since successive slices are quite similar.

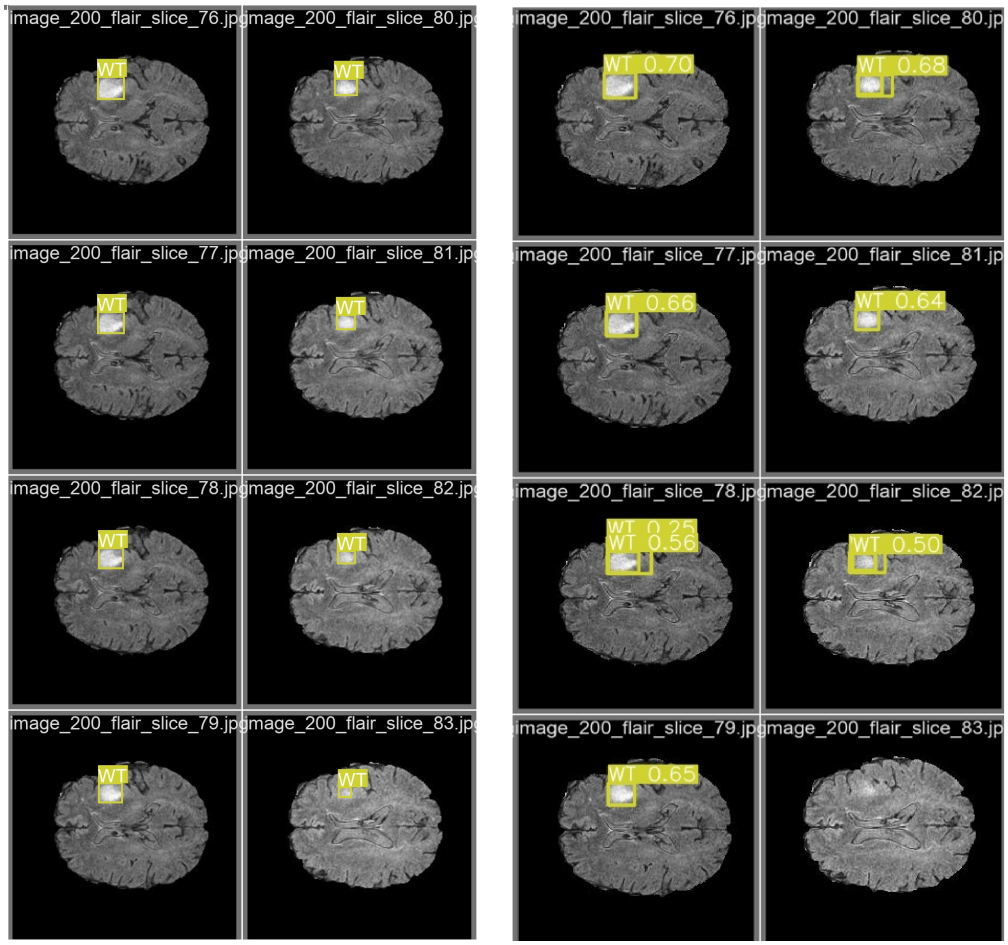

Figure 4.3: Example of predicted boxes by YOLO. *Slices 76-83 from an MRI volume: ground truth data (left) and the bounding boxes predicted and their confidence (right)*

We perform this search in a window of 20 slices (10 to the left and 10 to the right of the current image) and, if still no prediction is found, the respective image is skipped. The search starts from the index of the current image and then alternates between the next higher and lower indexes until it reaches the limits. In this way, we choose the image closest to the current one.

Some examples of slices that were skipped can be found in figure 4.4.

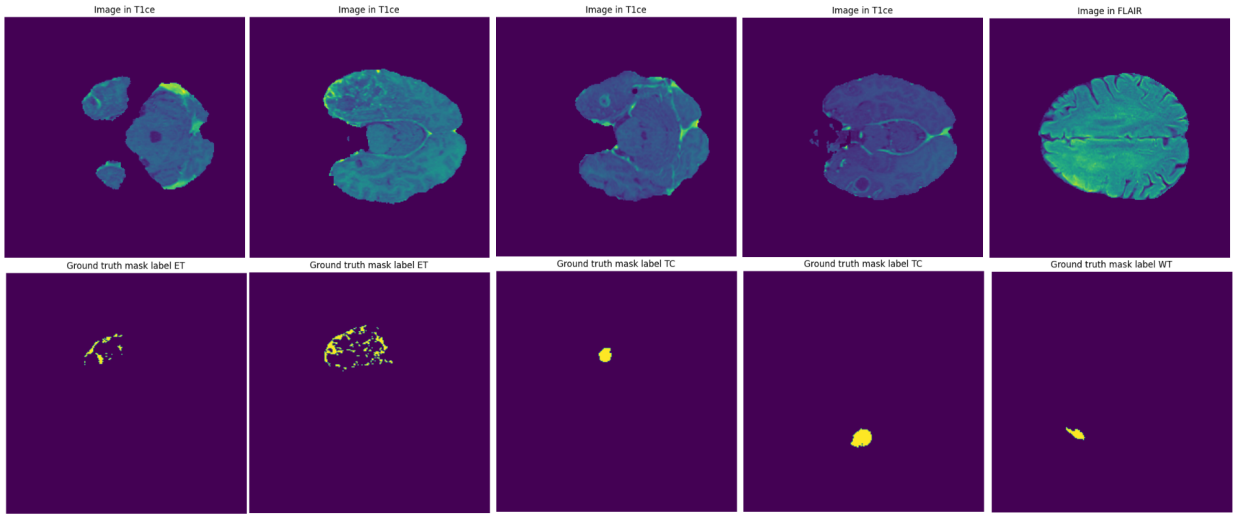

Figure 4.4: Example of slices without YOLO predictions. *Images from T1ce or FLAIR (first row) and the masks for the wanted labels (second row)*

4.1.2 Results

For the testing part, during inference, we run both the initial SAM checkpoint and the one after fine-tuning in order to see if the training improves the SAM model. First, we pass the images through the image encoder (which was frozen during training) to obtain image embeddings, encode the bounding box of the tumor with the prompt encoder and finally use the fine-tuned mask decoder to return the mask prediction. In Table 4.1 we can see the results for the what we described above. The testing was done with both ground truth boxes and with YOLO boxes.

Modality	Label	Ground truth bbox				YOLO bbox			
		Original SAM		Fine-tuned SAM		Original SAM		Fine-tuned SAM	
		DSC	Hausdorff	DSC	Hausdorff	DSC	Hausdorff	DSC↑	Hausdorff↓
FLAIR	ET	0.47	28	0.52	14	0.4	39	0.43	24
	TC	0.62	28	0.74	14	0.55	36	0.66	23
	WT	0.68	37	**0.82**	**18**	0.75	35	**0.73**	**27**
T1ce	**ET**	0.58	27	**0.76**	**12**	0.66	17	**0.72**	**14**
	TC	0.74	28	**0.85**	**11**	0.61	43	**0.73**	**31**
	WT	0.59	38	0.72	19	0.48	57	0.59	44

Table 4.1: Results on test data with fine-tuned SAM using ground truth bounding boxes. *Dice score and Hausdorff distance for labels ET, TC, WT and modalitites T1ce and FLAIR using during testing ground truth boxes or YOLO predicted boxes*

It can be observed that the best result for the WT label was achieved using the FLAIR modality while ET and TC perform best using T1ce. This can be attributed to the fact that the ET and NCR labels (that compose TC) are best highlighted in T1ce whereas ED (the addition to TC to form WT, which is the outer layer of the tumor, as seen in Figure 2.3) is most clearly visible in FLAIR. Below in Figure 4.5 we have an example of an image and the predicted results using ground truth bounding boxes and YOLO boxes for each label: ET, TC and WT.

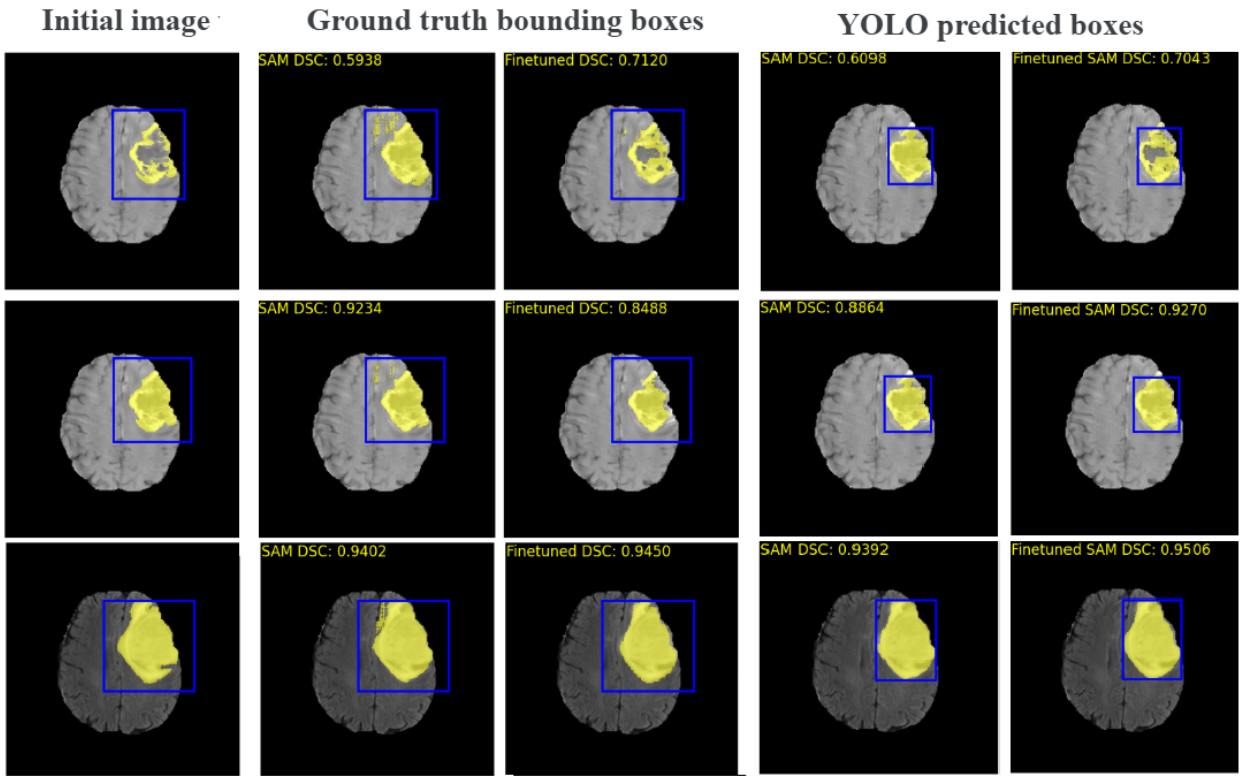

Figure 4.5: Example of predictions with fine-tuned SAM. *Predictions of an image for labels ET in T1ce (first row), TC in T1ce (second row) and WT in FLAIR (third row) using bounding boxes form ground truth and YOLO*

4.2 Fine-tuning SAM decoder with YOLO bounding boxes

4.2.1 Training setup

Another set of experiments adapt the previous training process and incorporate the predicted YOLO bounding boxes in train as well, in order to ensure consistency.

The setup for these experiments is similar to the previous one, the only difference being the prompt given in training. Earlier we used bounding boxes extracted from the ground truth masks. Now, we predict these bounding boxes with YOLO, like we did initially during testing.

We use the YOLO models trained before on FLAIR and T1ce modalities and employ the same procedure: if there is no prediction found for the current image, we search through the adjacent slices (a window of 20) and return the closest one found. If there is still no prediction found, we

use a default bounding box that encapsulates all the masks from the training data for the current label. Fundamentally, this bounding box has proven to be one that encloses the whole brain.

Bounding boxes prediction was done prior to training, the results being saved in the data loader. We have two methods here: considering the bounding boxes predicted by YOLO and the bounding boxes with a perturbation of maximum 20 pixels in each direction, as done during the fine-tuning of MedSAM as well [36].

The loss function used was once again the linear combination of Dice and Hausdorff and the optimizer was Adam with a learning rate of 10^{-5}. In the next section we present the results of the experiments with the combinations of modality-label that obtained earlier the best results: WT with FLAIR, ET and TC with T1ce.

4.2.2 Results

Similar to the previous trainings, we also run both the initial SAM and the fine-tuned one in order to compare them. First, we predict the bounding box of the current label using YOLO and apply the steps from training if there is no prediction found by searching through the adjacent slices for a prediction. If still none is found, we consider two cases: either use the default bounding box from training, either skip the slice. If the slice is used, the image goes through the image encoder, the bounding box is encoded with the prompt encoder and, along with the image embedding, return a prediction using the fine-tuned mask decoder. Table 4.2 shows the results for these experiments.

Modality	Label	Bbox noise in train and test	Default bbox in test	Original SAM		Fine-tuned SAM	
				DSC	Hausdorff	DSC	Hausdorff
T1ce	ET	No	No	0.66	17	**0.77**	**16**
	ET	No	Yes	0.59	32	0.70	23
	ET	Yes	Yes	0.52	45	0.69	28
	TC	No	No	0.81	14	**0.85**	**15**
	TC	No	Yes	0.58	52	0.75	32
	TC	Yes	Yes	0.53	62	0.73	33
FLAIR	WT	No	No	0.75	35	0.79	31
	WT	No	Yes	0.74	39	**0.79**	**32**
	WT	Yes	Yes	0.65	51	0.77	39

Table 4.2: Results on test data with fine-tuned SAM trained with YOLO predicted bounding boxes. *Dice scores and Hausdorff distances for labels ET, TC and WT*

For the labels ET and TC YOLO particularly struggles to find tumor regions in the images while in WT there only a few predictions missing and therefore the mean average Dice on all images is not affected as much. On the other hand, for ET and TC there is a significant difference between the case where we use the default bounding box and the one where we skip the slice. This is due to the fact that SAM requires a precise bounding box and cannot identify an object that it is only a small part of the region given in the prompt. An example of slice where a default bounding box was used can be seen in Figure 4.6. Here, the goal was for YOLO to predict TC and nothing was found.

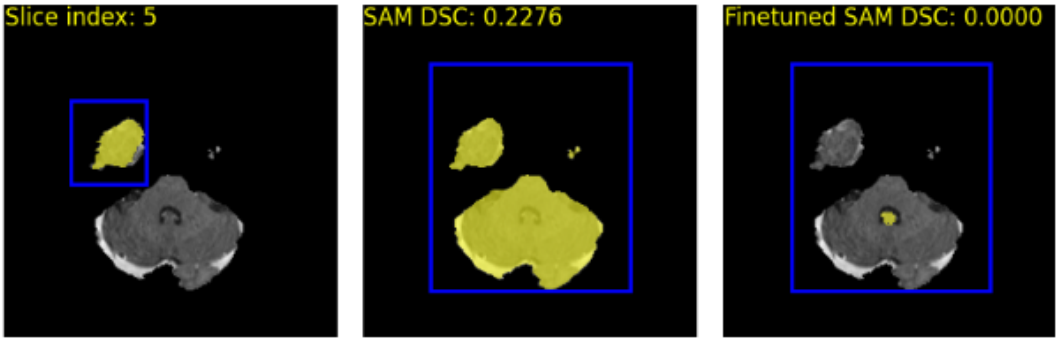

Figure 4.6: Prediction of SAM with a default bounding box. *Example slice with default bounding box. Ground truth mask and bounding box (first image), default bounding box and mask predicted by original SAM (second image), default bounding box and mask predicted by fine-tuned SAM (third image)*

For this particular patient, 33 out of 58 slices could not be identified using YOLO and therefore the default bounding box was used, the average dice score decreasing from 0.54 to 0.31. However, in Figure 4.7 we have examples where YOLO predicted a box close to truth.

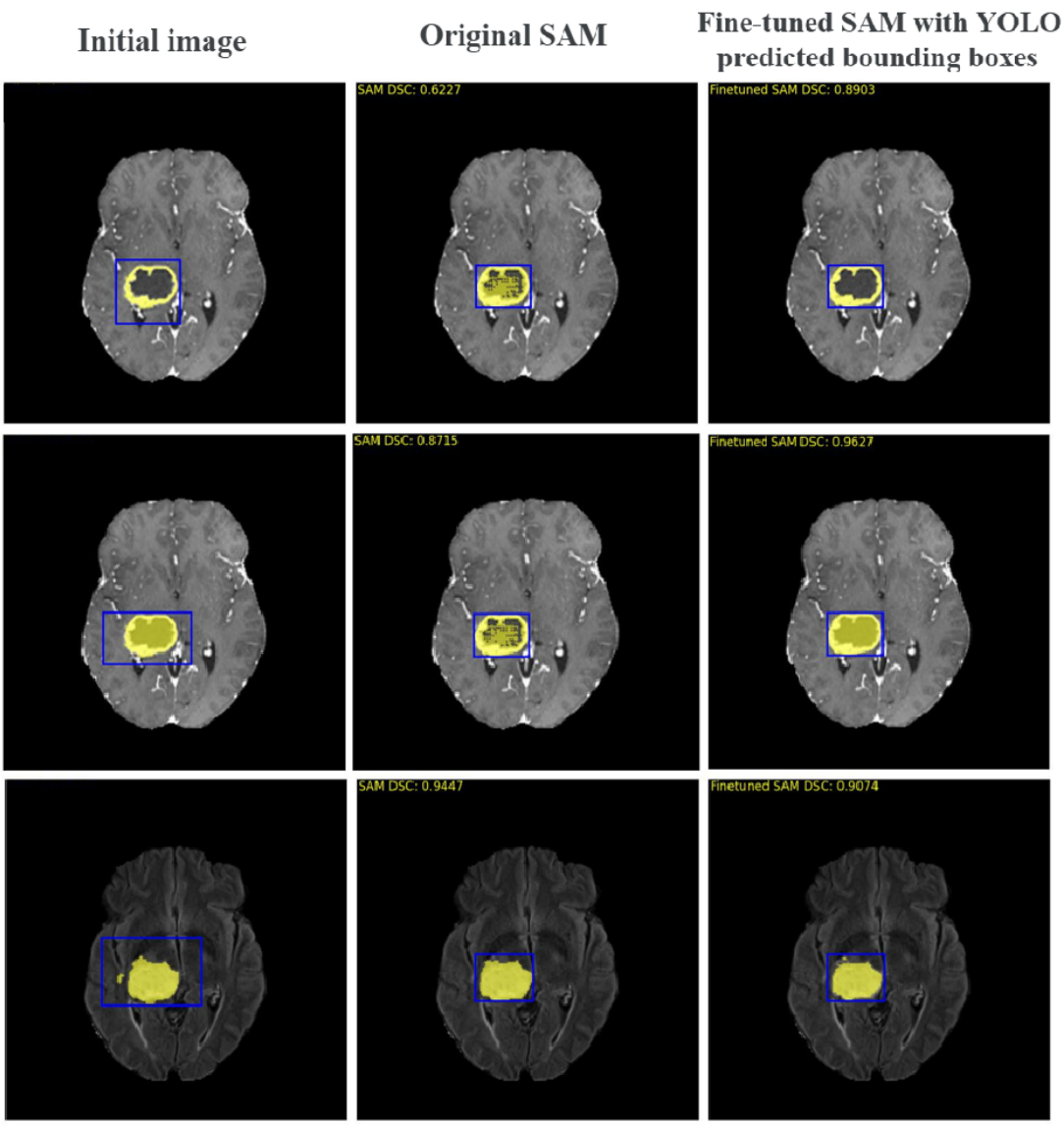

Figure 4.7: Example of predictions with fine-tuned SAM using YOLO bbox. *Predictions of an image for labels ET in T1ce (first row), TC in T1ce (second row) and WT in FLAIR (third row) using bounding boxes predicted by YOLO*

4.3 Discussions

In the previous section we experimented how the foundational model Segment Anything can adapt on a medical imagining task, more particularly on a brain tumor segmentation task. For each label (ET, TC and WT) we have a different model, trained on a different modality (T1ce and FLAIR), depending on where the tumor sections were most visible. Tables 4.1 and 4.2 showed that the best results are obtained when the bounding box of the tumor is given beforehand and adding an extra step of identifying the tumor in the image using YOLO decreases the accuracy and adds more instability to the pipeline since there are many images where there were no predictions found because of the blurriness of the tumor's outline.

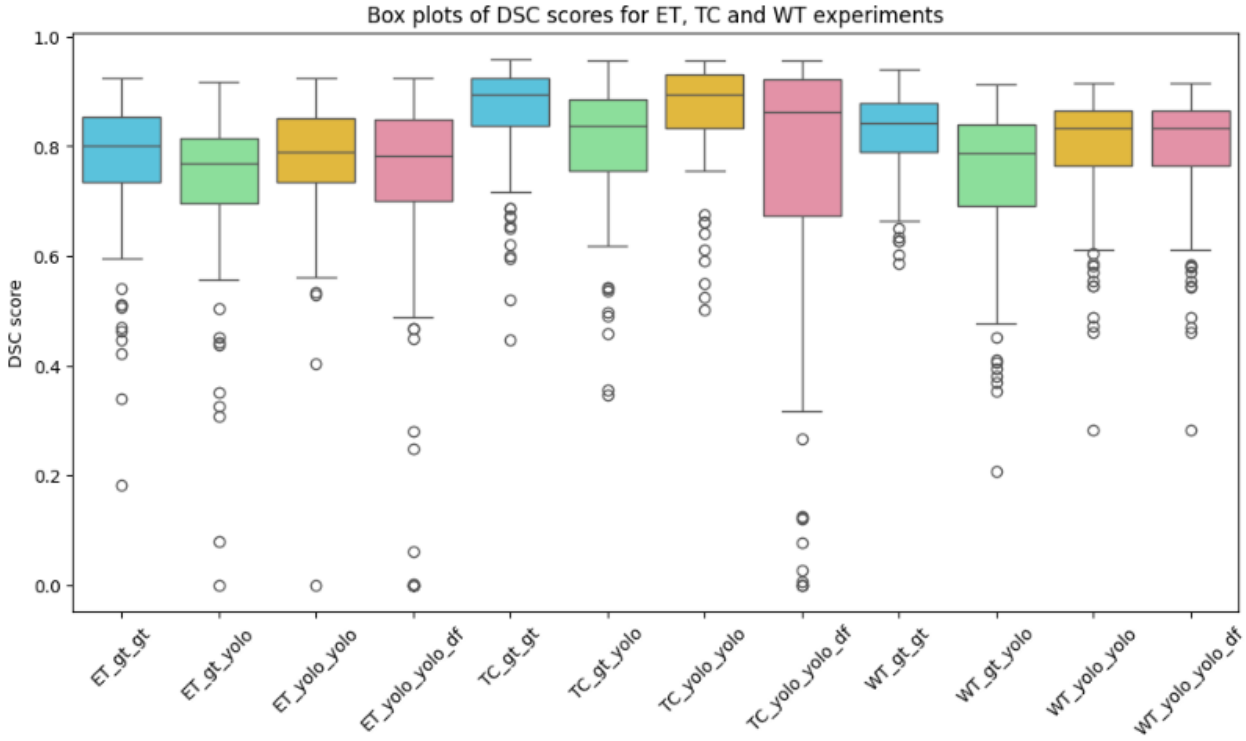

Figure 4.8: Box plot for the distribution of Dice scores among the experiments. *Box plots for labels ET, TC, WT using: ground truth in train and test (blue boxes), ground truth train and YOLO test (green boxes), YOLO train and test with skipping slices (yellow boxes), YOLO train and test with default box (pink boxes)*

In Figure 4.8 we have box plots for the distribution of Dice scores for each label within four categories of experiments:

- train SAM with ground truth bounding boxes and test with ground truth bounding boxes (gt_gt label, blue plots)
- train SAM with ground truth bounding boxes and test with YOLO bounding boxes (gt_yolo label, green plots)
- train SAM with YOLO bounding boxes and test with YOLO bounding boxes, skipping slices without predictions (yolo_yolo label, yellow plots)
- train SAM with YOLO bounding boxes and test with YOLO bounding boxes, using default bounding box for slices without predictions (yolo_yolo_df label, pink plots)

The section colored in the box plots represents the IQR (interquartile range) where the middle 50% of data is situated, while the whiskers extend to a larger range, 1.5 times the IQR. The circles plotted outside the whiskers are the outliers and represent the points significantly different than the rest. A more compact plot indicates consistent performance while a wider one suggests more variability in the results. Therefore, we have again that the blue plots (corresponding to the ground truth boxes) offer the best results in terms of the distribution of Dice scores, followed by the yellow ones.

Each solution can be suitable depending on the available resources. If a clinician is available to provide bounding boxes, then we can have the best results using the ground truth in training and testing. For scenarios requiring a full pipeline without any human intervention, using YOLO for both training and testing (and with default bounding boxes where predictions are absent, pink plots) can be a viable option. A hybrid approach involves training and testing with YOLO, skipping images without predictions, and waiting for a clinician's input for the bounding box of the tumor, thus minimizing the clinician's contribution while still enhancing accuracy. This case is represented by the yellow boxes in the figure.

5 Conclusion

In conclusion, bounding boxes impact significantly the performance of the fine-tuned SAM model. In Table 5.1 we have a summary of the results presented in the thesis. Ground truth bounding boxes consistently give the highest accuracy, as indicated by the blue plots. However, when clinician input is limited or a fully automated pipeline is required, YOLO-predicted bounding boxes provide a viable alternative, as we can observe in the box plots of color yellow. The semi-supervised approach that combines the YOLO predictions with a clinician intervention when the images are unclear offers a balanced solution, ensuring robust segmentation while reducing the need for extensive manual input.

Training	ET		TC		WT	
	Dice	Hausdorff	Dice	Hausdorff	Dice	Hausdorff
GT GT	0.76	12	0.85	11	0.82	18
GT YOLO	0.72	14	0.73	31	0.73	27
YOLO YOLO	0.77	16	0.85	15	0.79	31
YOLO YOLO + default box	0.70	23	0.75	32	0.79	32

Table 5.1: Summary results obtained. *Dice scores and Hausdorff distances for labels ET, TC, WT for the four scenarios described early*

Each method has its advantages and can be chosen depending on the specific requirements and constraints of the medical imaging task at hand. The results are comparable to state-of-the-art models that use a single modality, presented in Table 2.1, even with fine-tuning limited to only the decoder part of SAM, suggesting the potential of this model in future challenges.

Bibliography

[1] Haralick, R.M. and Shapiro, L.G., 1992. Computer and robot vision (Vol. 1, pp. 158-205). Reading, MA: Addison-wesley.

[2] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N. and Terzopoulos, D., 2021. Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence, 44(7), pp.3523-3542. source

[3] Otsu, N., 1975. A threshold selection method from gray-level histograms. Automatica, 11(285-296), pp.23-27. source

[4] Dhanachandra, N., Manglem, K. and Chanu, Y.J., 2015. Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science, 54, pp.764-771. source

[5] Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18 (pp. 234-241). Springer International Publishing. source

[6] Ruiz, D.V., Salomon, G. and Todt, E., 2020. Can giraffes become birds? an evaluation of image-to-image translation for data generation. arXiv preprint arXiv:2001.03637. source

[7] Image segmentation detailed overview, 2023. SuperAnnotate source

[8] Yao, W., Bai, J., Liao, W., Chen, Y., Liu, M. and Xie, Y., 2024. From cnn to transformer: A review of medical image segmentation models. Journal of Imaging Informatics in Medicine, pp.1-19. source

[9] Siddique, N., Paheding, S., Elkin, C.P. and Devabhaktuni, V., 2021. U-net and its variants for medical image segmentation: A review of theory and applications. Ieee Access, 9, pp.82031-82057. source

[10] BraTS2020 Dataset, https://www.kaggle.com/datasets/awsaf49/brats20-dataset-training-validation/data

[11] Henry, T., Carré, A., Lerousseau, M., Estienne, T., Robert, C., Paragios, N. and Deutsch, E., 2021. Brain tumor segmentation with self-ensembled, deeply-supervised 3D U-net neural networks: a BraTS 2020 challenge solution. In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part I 6 (pp. 327-339). Springer International Publishing. source

[12] Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R. and Lanczi, L., 2014. The multimodal brain tumor image segmentation benchmark (BRATS). IEEE transactions on medical imaging, 34(10), pp.1993-2024. source

[13] Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann, J.B., Farahani, K. and Davatzikos, C., 2017. Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Scientific data, 4(1), pp.1-13. source

[14] Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T., Berger, C., Ha, S.M., Rozycki, M. and Prastawa, M., 2018. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. arXiv preprint arXiv:1811.02629. source

[15] Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., et al., 2017. Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-GBM collection. The Cancer Imaging Archive. DOI: 10.7937/K9/TCIA.2017.KLXWJJ1Q source

[16] Mostafa, A.M., Zakariah, M. and Aldakheel, E.A., 2023. Brain tumor segmentation using deep learning on MRI images. Diagnostics, 13(9), p.1562. source

[17] Ghaffari, M., Sowmya, A. and Oliver, R., 2019. Automated brain tumor segmentation using multimodal brain scans: a survey based on models submitted to the BraTS 2012–2018 challenges. IEEE reviews in biomedical engineering, 13, pp.156-168. source

[18] Kamnitsas, K., Bai, W., Ferrante, E., McDonagh, S., Sinclair, M., Pawlowski, N., Rajchl, M., Lee, M., Kainz, B., Rueckert, D. and Glocker, B., 2018. Ensembles of multiple models and architectures for robust brain tumour segmentation. In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers 3 (pp. 450-462). Springer International Publishing. source

[19] Myronenko, A., 2019. 3D MRI brain tumor segmentation using autoencoder regularization. In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4 (pp. 311-320). Springer International Publishing. source

[20] Jiang, Z., Ding, C., Liu, M. and Tao, D., 2020. Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task. In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 5th International Workshop, BrainLes 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 17, 2019, Revised Selected Papers, Part I 5 (pp. 231-241). Springer International Publishing. source

[21] Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J. and Maier-Hein, K.H., 2021. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods, 18(2), pp.203-211.

[22] Zeineldin, R.A., Karar, M.E., Burgert, O. and Mathis-Ullrich, F., 2022, September. Multimodal CNN networks for brain tumor segmentation in MRI: a BraTS 2022 challenge solution. In International MICCAI Brainlesion Workshop (pp. 127-137). Cham: Springer Nature Switzerland.

[23] Ferreira, A., Solak, N., Li, J., Dammann, P., Kleesiek, J., Alves, V. and Egger, J., 2024. How we won BraTS 2023 Adult Glioma challenge? Just faking it! Enhanced Synthetic Data Augmentation and Model Ensemble for brain tumour segmentation. arXiv preprint arXiv:2402.17317. source

[24] Kang, M., Ting, F.F., Phan, R.C.W., Ge, Z. and Ting, C.M., 2024. A Multimodal Feature Distillation with CNN-Transformer Network for Brain Tumor Segmentation with Incomplete Modalities. arXiv preprint arXiv:2404.14019. source

[25] Zhang, Y., He, N., Yang, J., Li, Y., Wei, D., Huang, Y., Zhang, Y., He, Z. and Zheng, Y., 2022, September. mmformer: Multimodal medical transformer for incomplete multimodal learning of brain tumor segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 107-117). Cham: Springer Nature Switzerland. source

[26] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y. and Dollár, P., 2023. Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4015-4026). source

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, ´ and Ross Girshick. Masked autoencoders are scalable vision learners. CVPR, 2022 source

[28] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. source

[29] Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. and Ng, R., 2020. Fourier features let networks learn high frequency functions in low dimensional domains. Advances in neural information processing systems, 33, pp.7537-7547. source

[30] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. and Krueger, G., 2021, July. Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PMLR. source

[31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30. source

[32] Sofiiuk, K., Petrov, I.A. and Konushin, A., 2022, October. Reviving iterative training with mask guidance for interactive segmentation. In 2022 IEEE International Conference on Image Processing (ICIP) (pp. 3141-3145). IEEE. source

[33] Li, Z., Chen, Q. and Koltun, V., 2018. Interactive image segmentation with latent diversity. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 577-585). source

[34] Milletari, F., Navab, N. and Ahmadi, S.A., 2016, October. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In 2016 fourth international conference on 3D vision (3DV) (pp. 565-571). Ieee. source

[35] Zhang, Y., Shen, Z. and Jiao, R., 2024. Segment anything model for medical image segmentation: Current applications and future directions. Computers in Biology and Medicine, p.108238. source

[36] Ma, J., He, Y., Li, F., Han, L., You, C. and Wang, B., 2024. Segment anything in medical images. Nature Communications, 15(1), p.654. source

[37] Ribera, J., Guera, D., Chen, Y. and Delp, E.J., 2019. Locating objects without bounding boxes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6479-6489). source

[38] Uccheddu, F., Servi, M., Furferi, R. and Governi, L., 2018. Comparison of mesh simplification tools in a 3d watermarking framework. In Intelligent Interactive Multimedia Systems and Services 2017 10 (pp. 60-69). Springer International Publishing. source

[39] https://docs.ultralytics.com/modes/

[40] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788). source

[41] Krishnakumar, M., 2024, Object detection and tracking with YOLOv8 https://wandb.ai/mukilan/wildlife-yolov8/reports/Object-detection-and-tracking-with-YOLOv8